Fixpoint Checks and Computations for Behavioural Metrics and Games

Von der Fakultät für Ingenieurswissenschaften,
Abteilung Informatik und Angewandte Kognitionswissenschaften der
der Universität Duisburg-Essen

zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
(Dr. rer. nat.)

genehmigte Dissertation

von

Richard Otto Eggert
aus
Köln

1. Gutachter: Prof. Dr. Barbara König
2. Gutachter: Prof. Dr. Giorgio Bacci

Tag der mündlichen Prüfung: 28. November 2023

Abstract

Lattice theory is a well studied area of mathematics and finds many applications, for example in system verification. By Knaster-Tarksi, any monotone function $f\colon L \to L$ on a complete lattice L admits a least and a greatest fixpoint. It often occurs that one is specifically interested in such a least or greatest fixpoint.

This thesis provides a framework which can be used to verify if some fixpoint $a \in L$ of f is indeed the least/greatest fixpoint of f. Additionally, we are able to derive lower/upper bounds for least/greatest fixpoints. This theory will be embedded into a (gs-)categorical framework which allows us to create a tool for fixpoint verification.

Additionally, there is interest in computing these least/greatest fixpoints as no general method exists which always yields an exact computation. To this end, we provide a generalization of strategy iteration which allows one to compute least/greatest fixpoints.

Throughout this thesis we provide a wide range of applications where these methods can be applied. These include a variety of state-based system - termination probability of Markov chains, bisimilarity for transition systems and behavioural metrics for labeled Markov chains, metric transition systems and probabilistic automata - and three two-player-games - discounted mean-payoff games, simple stochastic games and energy games.

Zusammenfassung

Die Verbandstheorie ist ein gut untersuchtes Gebiet der Mathematik und findet viele Anwendungen, beispielsweise in der Systemverifikation. Nach Knaster-Tarski hat jede monotone Funktion $f\colon L \to L$ auf einem vollständigen Verband L einen kleinsten und einen größten Fixpunkt. Oftmals ist man speziell an solch einem kleinsten oder größten Fixpunkt interessiert.

Diese Arbeit stellt eine Theorie zur Verfügung, welche benutzt werden kann, um zu ermitteln, ob ein gegebener Fixpunkt $a \in L$ von f in der Tat der kleinste/größte Fixpunkt von f ist. Zusätzlich kann man untere/obere Schranken für den kleinsten/größten Fixpunkt ermitteln. Diese Theorie wird in ein (gs-)kategorisches Framework eingebettet, welches es uns ermöglicht, ein Werkzeug für solche Fixpunktverifizierungen zu erstellen.

Darüber hinaus besteht Interesse an der Berechnung dieser kleinsten/größten Fixpunkte, da es keine allgemeine Methode gibt, die immer eine exakte Berechnung liefert. Zu diesem Zweck stellen wir eine Verallgemeinerung von Strategie-Iteration bereit, die es ermöglicht, kleinste/größte Fixpunkte zu berechnen.

In dieser Arbeit stellen wir einige Anwendungen vor, für die diese Methoden angewendet werden können. Dazu gehören verschiedene zustandsbasierte Systeme - Terminierungswahrscheinlichkeit für Markow-Ketten, Bisimularität für Transitionssysteme und Verhaltensabstände für beschriftete Markow-Ketten, metrische Transitionssysteme und probabilistische Automaten - und drei Zwei-Spieler-Spiele - diskontierte Mean-Payoff-Spiele, einfache stochastische Spiele und Energie-Spiele.

Acknowledgement:
First of all, I would like to thank my supervisor *Barbara König* as I rarely encountered such a decent human being. I can not imagine I will ever find a supervisor I would prefer over her. She is extremely calm and knowledgeable which was indispensable in guiding me in the right direction.

Second, I would like to thank my collaborators *Paolo Baldan* and *Tommaso Padoan* as I really enjoyed working with them and this thesis would not be possible without their contributions. Additionally, I thank *Timo Matt* as he is mainly responsible for creating the tool UDEfix. He is a very sharp student and a pleasure to work with.

Next, I am really thankful for having such great coworkers throughout my time at the university. It was usually a blast to spend time with them and they were all very accomodating at all times.

Lastly, I thank my parents, my friends and the dragon.

Contents

1 | Introduction

Fixpoints appear in almost every area of mathematics - algebra, analysis, numeric, optimization, computer science, etc. To this end, many fixpoint theorems were discovered, e.g. fixpoint theorems by Banach, Browder-Göhde-Kirk, Borel and Gödel, to name a few. A handful of fixpoint iterations stemming from these discoveries can be used in different settings, e.g. Newton's method or more specifically the Bellman-Ford algorithm which computes shortest paths can be seen as fixpoint iterations. In this thesis we are mainly interested in fixpoints over complete lattices.

Lattices - a set L induced with a (partial) order that has all infima and suprema - first showed up in the middle of the nineteenth century in algebraic logic and number theory. Lattices resurfaced in the end of the 1920's in the field of algebra, mainly studied by Birkhoff [Bir40]. Knaster and Tarski added to this theory and provided a characterization of least and greatest fixpoints for a monotone function $f\colon L \to L$ over a complete lattice L [KT, Tar55].

This thesis provides a method to verify if some given fixpoint $a \in L$ of f is the least/greatest fixpoint (i.e. $a = \mu f / a = \nu f$) as well as deriving lower/upper bounds for a least/greatest fixpoint. Additionally, we present a generalization of strategy iteration which exactly computes these extreme fixpoints.

In order to make these methods effective we require a non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ where Y is some finite set and $\mathbb{M}$ a complete MV-chain. We remark that $\mathbb{M}^Y$ denotes the set of functions mapping from Y to $\mathbb{M}$. Complete MV-chains are complete lattices enriched with a strong algebraic structure (e.g. the interval $[0,1]$ with truncated addition) and non-expansiveness basically means that the distance between any two elements does not increase after applying f.

Such fixpoints are ubiquitous in computer science; we are in particular interested in applications in concurrency theory and games, such as bisimilarity [San11], behavioural metrics [DGJP04, vB17, CvBW12, BBKK18] and two-player-games [Con90, BCD+11, ZP96].

1.1. General Ideas

We are given a non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ and some fixpoint $a\colon Y \to \mathbb{M}$ of f. How can we detect if this fixpoint a is the least fixpoint of f or not?

To this end, we will derive a function - called approximation - on the powerset of Y. Any fixpoint of this function corresponds to what we call a "vicious cycle". Elements in such a vicious cycle intuitively convince each other that their value is higher/lower than it actually is.

Now, given some fixpoint $a \in \mathbb{M}^Y$ of a non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$, it holds that a is the least fixpoint of f if and only if the system does not contain any vicious cycle. Additionally, we provide an (incomplete) proof rule which allows one to derive lower bounds for the least fixpoint. This theory is worked out in Chapter 3.

As we will see, non-expansive functions enjoy good closure properties which allows one to assemble a more complicated function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ from a few basic functions. This - together with the fact that our approximation theory can be cast into a gs-categorical framework - forms the basis for the tool UDEfix we will present in Chapter 4 which performs fixpoint checks for user created functions.

Next, we are also interested in finding a general method to compute μf as no general method exists which yields an exact computation. Therefore, we derive a generalization of strategy iteration which progressively improves the strategy until an optimal one is found. This optimal strategy corresponds to μf. This generalization of strategy iteration is worked out in Chapter 5.

Dual methods for greatest fixpoints will be presented as well.

1.2. Related Literature

It is well-known that several computations in the context of system verification can be performed by various forms of fixpoint iteration and it is worthwhile to study such methods at a high level of abstraction, typically in the setting of complete lattices and monotone functions. Going beyond the classical results by Tarski [Tar55], combination of fixpoint iteration with approximations [CC00, BKP20] and with up-to techniques [Pou07] has proven to be successful. Here we treat a more specific setting, where the carrier set consists of functions from a finite set into an MV-chain and the fixpoint functions are non-expansive (and monotone), and introduce a novel technique to perform fixpoint checks and obtain upper bounds for greatest and lower bounds for least fixpoints. Such techniques are applicable to a wide range of examples and so far they have been studied only in quite specific scenarios, such as in [BBL+21, Fu12, KKKW18]. The paper [KUK+22] presents somewhat related algorithms regarding reachability analysis.

We view gs-monoidal categories [CG99] as a means to compositionally build monotone non-expansive functions on complete lattices, for which we are interested in the least (or greatest) fixpoint.

Fixpoint equations also arise in the context of coalgebra [Rut00], a general framework for investigating behavioural equivalences for systems that are parameterized – via a functor – over their branching type (labelled, non-deterministic, probabilistic, etc.). Here in particular we are concerned with coalgebraic behavioural metrics [BBKK18], based on a generalization of the Wasserstein lifting [Vil09]. Such liftings require the notion of predicate liftings, well-known in coalgebraic modal logics [Sch08b], lifted to a quantitative setting [BKP18].

Strategy iteration is used in many different application domains with fairly similar underlying ideas and we believe that it is fruitful to provide a general definition of the technique, clarifying and solving several issues on this level. We propose both a strategy iteration approaching the least fixpoint from below as well as from above.

There is an extremely wide literature on strategy iteration, often also referred to as policy iteration or strategy improvement (for an overview see [GTW02]). Several quasi-polynomial algorithms have been recently devised for parity games [CJK+17, JL17, Leh18], while the existence of a polynomial algorithm is still an important open problem. This has been generalized to finite lattices by [HS21].

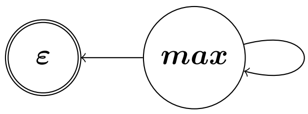
Various papers on strategy iteration focus on lower bounds [Fri11, Fea10]. This thesis, rather than concentrating on complexity issues, provides a general framework capturing strategy iteration in a general lattice theoretical setting. A work similar in spirit is [ABdMS21] which proposes a meta-algorithm GSIA such that a number of strategy improvement algorithms for simple stochastic games arise as instances, along with a general complexity bound. Differently from ours, this paper focuses on simple stochastic games and iteration from below. However, it allows for the parametrisation of the algorithm on a subset of edges of interest in the game graph, which is not possible in our approach, and so it can provide interesting suggestions for further generalisations.

Given their generality, we believe that the algorithms proposed in this thesis have the potential to be applicable to a variety of other settings.

We will discuss additional related literature in various chapters.

1.3. A Short Example

Imagine a game played on the following game graph:



The game starts by placing a pebble in one of the two states. The aim of the player - called Max - is to maximize his payoff. When the pebble is in state ε, Max obtains payoff $\varepsilon \in (0,1)$. Whenever the pebble is in state **max** the player can choose where to move the pebble - either to state **max** or state ε. If the pebble never reaches state ε player Max obtains a payoff of 0 and he can obtain a maximal payoff of 1.

As it often occurs the payoff player Max is able to obtain in any state can be derived as the least fixpoint of the following function $\mathcal{V}\colon [0,1]^{\{\mathbf{max},\boldsymbol{\varepsilon}\}} \to [0,1]^{\{\mathbf{max},\boldsymbol{\varepsilon}\}}$, defined as

$$\mathcal{V}(a)(v) = \begin{cases} \varepsilon & \text{if } v = \boldsymbol{\varepsilon} \\ \max\{a(\mathbf{max}), a(\boldsymbol{\varepsilon})\} & \text{if } v = \mathbf{max} \end{cases}$$

for $a\colon \{\mathbf{max}, \boldsymbol{\varepsilon}\} \to [0,1]$. It is immediate that player Max is able to obtain payoff ε whenever the pebble is in either state which corresponds to the least fixpoint $\mu\mathcal{V}$ of $\mathcal{V}$, i.e. $\mu\mathcal{V}(\boldsymbol{\varepsilon}) = \mu\mathcal{V}(\mathbf{max}) = \varepsilon$.

We briefly remark that $\mathcal{V}$ is in fact non-expansive and the interval $[0,1]$ is a complete MV-chain (with additional operators). Also, the above game is a simple stochastic game which we will discuss throughout this thesis.

Now, the function $\mathcal{V}$ admits multiple fixpoints, e.g. the greatest fixpoint $\nu\mathcal{V}$ of $\mathcal{V}$ is given by $\nu\mathcal{V}(\boldsymbol{\varepsilon}) = \varepsilon$ and $\nu\mathcal{V}(\mathbf{max}) = 1$. In the case of $\nu\mathcal{V}$, it is a fixpoint because state **max** convinces itself that its payoff is 1, as

$$\mathcal{V}(\nu\mathcal{V})(\mathbf{max}) = \max\{\nu\mathcal{V}(\mathbf{max}), \nu\mathcal{V}(\boldsymbol{\varepsilon})\} = \max\{1, \varepsilon\} = 1$$

which does not make sense since moving the pebble in a cycle yields a payoff of 0. This is what we call a "vicious cycle" as states in this cycle (in this case only state **max**) convince each other that their payoff is higher than it actually is. The theory in Chapter 3 enables us to detect these vicious cycles.

In the above example, it is rather clear, that the optimal (positional) strategy (fixing the successor of each state of player Max) for player Max is to move the pebble to state $\boldsymbol{\varepsilon}$ whenever it is in state **max**. This strategy - called C - corresponds to the least fixpoint $\mu\mathcal{V}$ of $\mathcal{V}$ in the sense that the strategy-induced function $\mathcal{V}_C\colon [0,1]^{\{\mathbf{max},\boldsymbol{\varepsilon}\}} \to [0,1]^{\{\mathbf{max},\boldsymbol{\varepsilon}\}}$, defined as

$$\mathcal{V}_C(a)(v) = \begin{cases} \varepsilon & \text{if } v = \boldsymbol{\varepsilon} \\ a(\boldsymbol{\varepsilon}) & \text{if } v = \mathbf{max} \end{cases}$$

for $a\colon \{\mathbf{max}, \boldsymbol{\varepsilon}\} \to [0,1]$, has as least fixpoint $\mu\mathcal{V}_C = \mu\mathcal{V}$. In Chapter 5 we derive a generalization of strategy iteration.

1.4. Applications

As we have discussed, our theories require a non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$. This might seem rather restrictive but as we will see many applications can be cast into this framework.

To this end, we will analyze the following applications in this thesis:

- Termination probability for Markov chains
- Bisimilarity for transition systems
- Behavioural distances for labeled Markov chains
- Behavioural distances for metric transition systems
- Behavioural distances for probabilistic automata
- Discounted mean-payoff games
- Simple stochastic games

- Energy games

There exist applications that can not be cast into our framework, e.g. parity games or mean-payoff games - as their solution does not correspond to a suitable fixpoint operator.

Quite interestingly, we were able to find an application in a seemingly completely different area - weakest preexpectations for (probabilistic) programs [MMKK18]. Here, given a postexpectation g and a (probabilistic) program C, we aim to derive the weakest preexpectation $wp[\![C]\!](g)$, i.e. what is the expected value of g after executing C. We are able to embed this setting into our framework with some restrictions and show that the function $wp[\![C]\!]$ is non-expansive for standard programs C. It gets even more interesting when the program C contains a *while*-loop. In this case, the weakest preexpectation corresponds to a least fixpoint and there exists interest in obtaining lower bounds for this warranted preexpectation. To this end, our theory provides an (incomplete) proof rule.

1.5. Structure

Most content presented in this thesis can be found in the original publications. A few additional applications were added and some parts are worked out in more detail. Also, we reorganized some contents to a more fitting position. For partial reading, see Figure 1.1 for a dependency graph for this thesis.

We build the mathematical foundation of the theories discussed earlier in Chapter 2. We first introduce the notations and elementary definitions (Section 2.1) which are mostly standard. Then we discuss the theory of linear programming (Section 2.2), complete lattices and MV-chains (Section 2.3), category theory (Section 2.4) and coalgebraic behavioural metrics (Section 2.5). Afterwards we introduce the applications we will use to illustrate our theories: state-based systems (Section 2.6) and two-player games (Section 2.7). There is very few original research within this chapter - all proofs that are written out can be seen as such.

In Chapter 3 we discuss the theory from the paper "Fixpoint Theory - Upside Down" [BEKP23b]. After discussing the theory for approximating the propagation of increases (Section 3.2) we derive the main proof rules we derived for greatest fixpoints (Section 3.3). After discussing the dual view for least fixpoints (Section 3.4) we tend to the (de)composition of functions and their approximations (Section 3.5). Next, we will discuss the approximations for the applications introduced in Chapter 2 (Section 3.6). Here, one can find some extensions to the original paper as not all applications were discussed in the original publications.

Chapter 4 tends to embedding the theory from Chapter 3 into a categorical setting based on the paper "A Monoidal View on Fixpoint Checks" [BEK+23b]. We can (partially) extend the theory to an infinite domain (Section 4.2) before describing the categories our functions and their approximations live in (Section 4.3). This allows us to find approximations for general predicate liftings (Section 4.4) and generalize the approximation of the Wasserstein lifting (Section 4.5). Afterwards we show that our previously derived categories are gs-monoidal (Section 4.6) which allowed us to create the tool UDEfix (Section 4.7) where the user can input his or her functions.

Lastly, in Chapter 5 we aim to generalize strategy iterations as in the paper "A Lattice-Theoretical View of Strategy Iteration" [BEKP23a]. Here, we derive two generalized strategy iterations - one approaching from above and one from below to compute some least fixpoint (Section 5.2). We also discuss the dual view for greatest fixpoints (Section 5.2.6) which was not present in the original publication. Afterwards, we instantiate these strategy iterations to the applications from Chapter 2 (Section 5.3) and perform a few runtime comparisons (Section 5.4). A handful of applications were not discussed in the original paper and the runtime comparison for discounted mean-payoff games is new as well.

We end with a conclusion and suggest future work (Chapter 6).

Proofs that were deemed too extensive and/or less central can be found in the appendix (Chapter A).

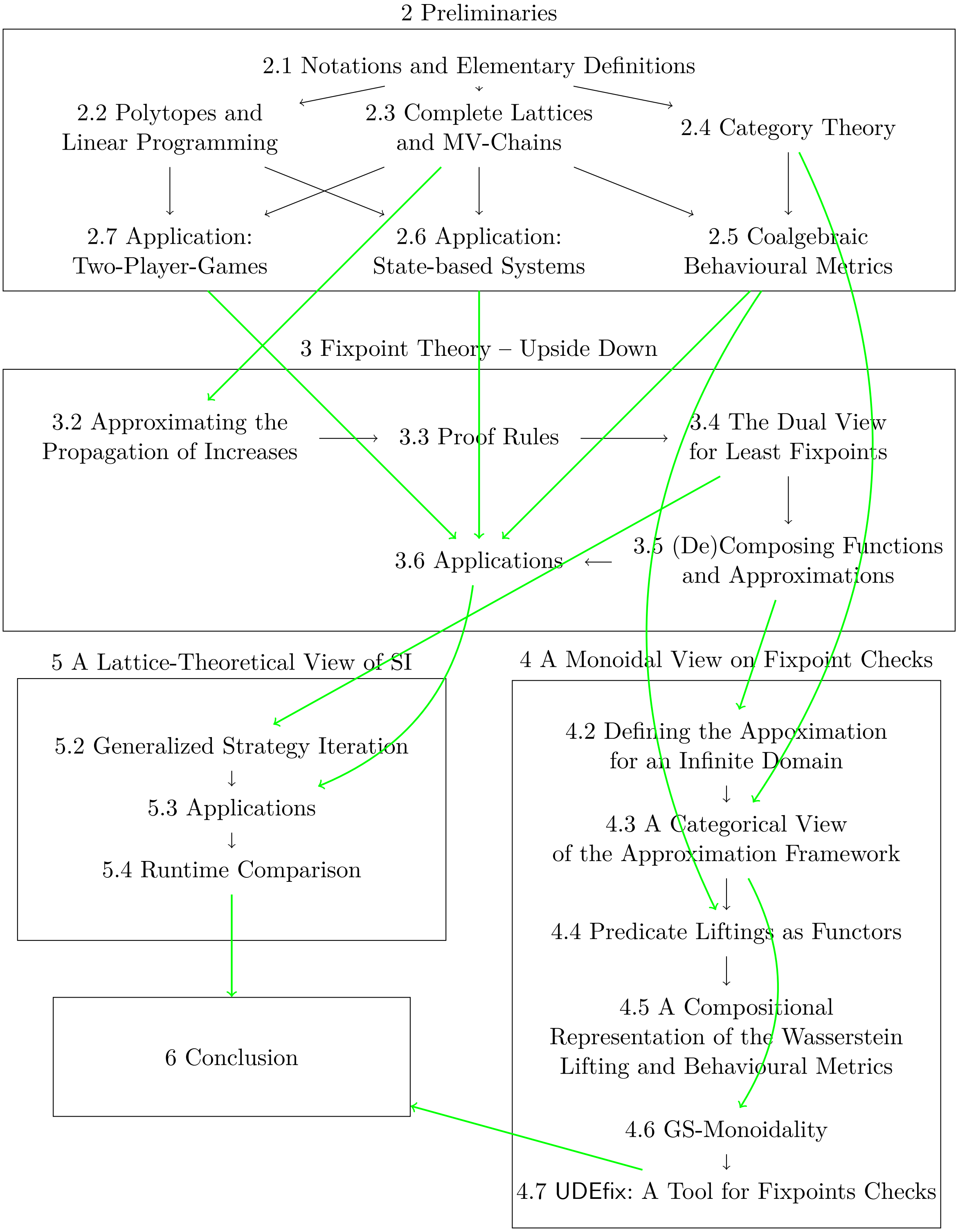


Fig. 1.1.: Dependency graph for this thesis, an arrow pointing from one section to another indicates that the arrow-emitting section should be read before for a complete understanding

Fixpoint Checks and Computations for Behavioural Metrics and Games

This thesis is based on the following original publications:

- **Chapter 3:**
  - *Conference Paper:* [BEKP21a]

    Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint Theory – Upside Down. In *Proc. of FOSSACS '21*, pages 62-81. Springer, 2021. LNCS/ARCoSS 12650.

  - *Journal Paper:* [BEKP23b]

    Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint Theory – Upside Down. *Logical Methods in Computer Science*, 19(2:15), June 2023. Selected Papers of the 24th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2022).

  - *ArXiv-Version:* [BEKP21b]

    Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint theory – Upside Down, 2021. arXiv:2101.08184

- **Chapter 4:**
  - *Conference Paper:* [BEK+23b]

    Paolo Baldan, Richard Eggert, Barbara König, Timo Matt, and Tommaso Padoan. A Monoidal View on Fixpoint Checks. In *Proc. of ICGT '23*, pages 3-21. Springer, 2023. LNCS 13961.

  - *ArXiv-Version:* [BEK+23a]

    Paolo Baldan, Richard Eggert, Barbara König, Timo Matt, and Tommaso Padoan. A Monoidal View on Fixpoint Checks, 2023. arXiv:2305.02957

- **Chapter 5:**
  - *Early Ideas:* [EK19]

    Richard Eggert and Barbara König. Computing Coalgebraic Behavioural Metrics On-The-Fly. In CALCO Early Ideas '19, 2019.

- *Conference Paper:* [BEKP23a]

  Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. A Lattice-Theoretical View of Strategy Iteration. In *Proc. of CSL '23*, volume 252 of *LIPIcs*, pages 17:1-17:17. Schloss Dagstuhl - Leibniz Center for Informatics, 2018.

- *ArXiv-Version:* [BEKP22]

  Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. A Lattice-Theoretical View of Strategy Iteration, 2022. arXiv:2207.09872

2 | Preliminaries

In this chapter we give an overview of the preliminaries needed for the theories in later chapters. After giving a short overview on elementary definitions and notations used throughout this thesis in Section 2.1, we briefly explain linear programming in Section 2.2 as a handful of linear programs arise throughout this thesis. Next, we introduce complete lattices and MV-chains in Section 2.3. These algebraic structures form the basis of the later introduced "Upside-Down"-theory in Chapter 3. Afterwards we give a short overview on the essential definitions of category theory in Section 2.4 which will be used to define coalgebraic behavioural metrics in Section 2.5. These allow for a quantitive measurement of the difference in behaviour of two states in a system. Lastly, we introduce a few example systems used to illustrate and apply the later theories. These are state based systems (Section 2.6) and two-player games (Section 2.7) which - due to their simplicity - yield excellent illustrations of the later theories.

2.1. Notations and Elementary Definitions

We briefly introduce the elementary definitions and notations used throughout this thesis. Most definitions are well-known and the notations are rather standard.

Sets. We use standard notations: $\mathbb{N}$ is the set of natural numbers (including 0), $\mathbb{Z}$ the set of integers, $\mathbb{Q}$ is the set of rational numbers and $\mathbb{R}$ is the set of real numbers. We denote $\mathbb{R}^+$ as the set of non-negative real numbers (including 0). Additionally, we write $\mathbb{N}^\infty = \mathbb{N} \cup \{\infty\}$ and $\mathbb{R}^\infty = \mathbb{R} \cup \{-\infty, \infty\}$.

Furthermore, given sets X and Y, we write $x \in X$ ($x \notin X$) to denote that x is (not) an element of X. A map or mapping or function f between two set X and Y, written as $f\colon X \to Y$, assigns exactly one element $f(x) \in Y$ to each element in $x \in X$. $X^Y = \{f \mid f\colon Y \to X\}$ denotes the set of all mappings from Y to X.

Given some set Y and a subset $Y' \subseteq Y$, the characteristic function $\chi_{Y'}\colon Y \to \{0,1\}$ is defined for $y \in Y$ as follows:

$$\chi_{Y'}(y) = \begin{cases} 1 & \text{if } y \in Y' \\ 0 & \text{otherwise} \end{cases}.$$

We use typical λ-calculus notation, i.e. $\lambda x.y$ describes the function which maps each x to some constant y.

The cross product $X = X_1 \times \cdots \times X_n$ between sets $X_1, \ldots X_n$ is the set of tuples given by

$$X = \{(x_1, \ldots, x_n) \mid x_1 \in X_1, \ldots, x_n \in X_n\}$$

We also define projection maps to the n-th component in the usual way, i.e. $\pi_i : X^n \to X$ where $X^n = X \times \cdots \times X$ (n-times) with $\pi_i(x_1, \ldots, x_n) = x_i$ for $i = 1, \ldots, n$. $|X|$ denotes the cardinality of the set X. In a similar fashion, for functions $f_i : X_i \to Y_i$, $i = 1, 2, \ldots, n$, we define $f_1 \times \ldots f_n : X_1 \times \ldots X_n \to Y_1 \times \ldots Y_n$ for $(x_1, \ldots, x_n)$ $(x_i \in X_i)$ as

$$(f_1 \times \ldots f_n)(x_1, \ldots, x_n) = (f_1(x_1), \ldots f_n(x_n)).$$

Definition 2.1.1 (powerset)**.** *Given some set X, we denote by $\mathcal{P}(X)$ the* ***powerset*** *of X. We define*

$$\mathcal{P}_f(X) := \{Y \in \mathcal{P}(X) \mid |Y| < \infty\}$$

as the ***finite powerset*** *of X. If X is finite it holds that $\mathcal{P}(X) = \mathcal{P}_f(X)$.*

Example 2.1.2. *Let $X = \{x, y, z\}$. Then $|X| = 3$ and*

$$\mathcal{P}(X) = \mathcal{P}_f(X) = \{\varnothing, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$$

where $\varnothing$ denotes the empty set.

We will frequently be working with probability distributions. To this end, it makes sense to define the support of a function, since we will mostly be working with finitely supported probability distributions.

Definition 2.1.3 (support)**.** *Given some set X and a function $f : X \to \mathbb{R}$. We define the* ***support*** *$supp(f)$ of f as*

$$supp(f) = \{x \in X \mid f(x) \neq 0\}.$$

Definition 2.1.4 (probability distribution)**.** *Given some set X, a* ***probability distribution*** *over X is some map $p : X \to [0, 1]$, such that $\sum_{x \in X} p(x) = 1$. We write $\mathcal{D}(X)$ for the* ***set of probability distributions*** *over X, i.e. $\mathcal{D}(X) = \{p : X \to [0, 1] \mid \sum_{x \in X} p(x) = 1\}$.*
We denote $\mathcal{D}_f(X)$ the ***set of probability distributions with finite support****, i.e.*

$$\mathcal{D}_f(X) = \{p : X \to [0, 1] \mid \sum_{x \in X} p(x) = 1 \text{ and } supp(p) < \infty\}.$$

If X is finite $\mathcal{D}(X) = \mathcal{D}_f(X)$ holds.

Example 2.1.5. *Let* $X = \{x, y, z\}$*. Then* $p\colon X \to [0,1]$ *with* $p(x) = 0.5$*,* $p(y) = 0.5$ *and* $p(z) = 0$ *is an element of* $\mathcal{D}(X)$*.*

When describing a probability distribution we usually omit the elements outside the support.

Relations. For a binary relation $R \subseteq X \times Y$, we might write xRy instead of $(x, y) \in R$ to indicate that x and y are related. We need a few equivalence relations in this thesis.

Definition 2.1.6 (equivalence relation)**.** *A relation* $R \subseteq X \times X$ *is an* ***equivalence relation*** *if it is*

- reflexive: *for all* $x \in X$ *it holds that* $(x, x) \in R$
- symmetrical: *for all* $x, y \in X$ *it holds that if* $(x, y) \in R$ *then* $(y, x) \in R$
- transitive: *for all* $x, y, z \in X$ *it holds that if* $(x, y) \in R$ *and* $(y, z) \in R$ *then* $(x, z) \in R$

Given an equivalence relation R on some set X, the equivalence class of an element $a \in S$, denoted by $[a]$ is given by $[a] = \{x \in X \mid xRa\}$. The equivalence classes form a partition of X and the set of equivalence classes is called the quotient set, denoted by X/R.

Example 2.1.7. *Let* $X = \{x, y, z\}$ *and* $R = \{(x, x), (y, y), (z, z), (x, y), (y, x)\}$*. Then* $R \subseteq X \times X$ *is an equivalence relation and* $X/R = \{\{x, y\}, \{z\}\}$ *is the quotient set.*

Metrics and Norms. We list the properties of a metric and a pseudometric.

Definition 2.1.8 (pseudometric, metric)**.** *Given some set* X*, a map* $d\colon X \times X \to \mathbb{R}$ *is a* ***pseudometric*** *if the following axioms hold for all* $x, y, z \in X$*:*

1. Reflexivity: $d(x, x) = 0$
2. Symmetry: $d(x, y) = d(y, x)$
3. Triangle inequality: $d(x, y) \leq d(x, z) + d(z, y)$

A pseudometric $d\colon X \times X \to \mathbb{R}$ *is a* ***metric*** *if additionally it holds for all* $x, y \in X$*:*

- Positiv definite: $d(x, y) = 0$ *if and only if* $x = y$

If d *is a (pseudo)metric on* X*, we call* (X, d) *a* ***(pseudo)metric space****. If* d *has codomain* $[0, \top]$ *for some* $\top > 0$*, we call* (X, d) *a* $\top$*-(pseudo)metric space.*

Pseudometrics allow for differing elements to have distance 0 which will prove useful. Note that any metric is a pseudometric. Throughout this thesis we will mostly be working with 1-(pseudo)metric spaces. Most people are familiar with the euclidian and discrete metric.

Example 2.1.9. $([0,\top], d_e)$ *is a* $\top$*-metric space where*

$$d_e(a,b) = |a-b|$$

for $a,b \in [0,\top]$*. We call* d_e *the euclidian metric.*

$([0,\top], d')$ *is a* $\top$*-metric space where*

$$d'(a,b) = \begin{cases} 1 & \text{if } a \neq b \\ 0 & \text{otherwise} \end{cases}$$

for $a,b \in [0,\top]$*. We call* d' *the discrete metric.*

We can define non-expansive functions between metric spaces.

Definition 2.1.10 (non-expansive, isometry (metric spaces))**.** *Let* (X, d_X) *and* (Y, d_Y) *be* $\top$*-metric spaces for some* $\top > 0$*. We call a function* $f\colon X \to Y$ ***non-expansive*** *if* $d_Y \circ (f \times f) \leq d_X$*. If equality holds,* f *is called an* ***isometry****.*

Lastly, we define a norm:

Definition 2.1.11 (norm)**.** *Given some vector space*[1] V *over* $\mathbb{R}$*. A mapping* $\|\cdot\|\colon V \to \mathbb{R}^+$*,* $x \mapsto \|x\|$ *is a* ***norm*** *if it satisfies the following axioms for all* $x,y \in V$ *and* $\alpha \in \mathbb{R}$*:*

1. Definite: *If* $\|x\| = 0$ *then* $x = 0$
2. Absolute Homegeneity: $\|\alpha \cdot x\| = |\alpha| \cdot \|x\|$
3. Triangle inequality: $\|x+y\| \leq \|x\| + \|y\|$

We can interpret functions $x\colon Y \to \mathbb{R}$ for some finite set $Y = \{y_1, \ldots, y_n\}$ as vectors $(x(y_1), \ldots, x(y_n)) \in \mathbb{R}^n$ where we can define a norm on.

Example 2.1.12. *Most mathematicians are familiar with the p-norm, defined for* $1 \leq p < \infty$ *and* $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$ *as*

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}.$$

[1] See [Hal93] for the definition of a vector space.

For $p = \infty$ we obtain the norm

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|.$$

2.2. Polytopes and Linear Programming

We will give a short overview on linear programming based on [KV12]. Linear programming is not crucial for understanding the main theories this thesis provides but linear programs frequently arise throughout. To this end, linear programming can be used to obtain an exact computation of the least fixpoint of some functions. See [KV12] for a deeper look into linear programming.

As it is standard, any vector $x \in \mathbb{R}^n$ is a column vector and $x^T \in \mathbb{R}^n$ (the transposed vector) a row vector. A matrix $A \in \mathbb{R}^{m \times n}$ with m rows and n columns and can be written as

$$A = \begin{pmatrix} a_1^T \\ \vdots \\ a_m^T \end{pmatrix}$$

for vectors $a_i \in \mathbb{R}^n$ $(i = 1, \dots, m)$. The product $y^T x$ of two vectors $x, y \in \mathbb{R}^n$ is given by

$$y^T x = \sum_{i=1}^{n} y_i \cdot x_i.$$

Definition 2.2.1 (halfspace/polyhedron)**.** *Let $b \in \mathbb{R}$ and $a \in \mathbb{R}^n$ with $a \neq 0$. The set*

$$P(a, b) = \{x \in \mathbb{R}^n \mid a^T x \le b\}$$

is an n-dimensional ***halfspace****. A* ***polyhedron*** *is the intersection of finitely many halfspaces, written as*

$$P(A, b) = \{x \in \mathbb{R}^n \mid Ax \le b\}$$

for a vector $b \in \mathbb{R}^m$ and a matrix $A \in \mathbb{R}^{m \times n}$ where $n \ge m$.

Without restriction, we assume $rank(A) = m$ (otherwise one can just remove redundant rows from A and b). We note that $\{x \in \mathbb{R}^n \mid Ax \le b,\ A_{eq}x = b_{eq}\}$ ($A_{eq} \in \mathbb{R}^{k \times n}$, $b_{eq} \in \mathbb{R}^k$) is a polyhedron as well, precisely we obtain the polyhedron

$$P(\begin{pmatrix} A \\ A_{eq} \\ -A_{eq} \end{pmatrix}, \begin{pmatrix} b \\ b_{eq} \\ -b_{eq} \end{pmatrix}).$$

Any polyhedron is convex, i.e. for all $x, y \in P(A, b)$ and $\lambda \in [0, 1]$ it holds that $\lambda \cdot x + (1 - \lambda) \cdot y \in P(A, b)$. We now give a characterization of the vertices of a polyhedron.

Definition 2.2.2 (basic solution/vertex). *Given some polyhedron $P(A,b)$, $r \in \mathbb{N}$ and an index set $I = \{i_1, \dots, i_r\} \subseteq \{1, \dots, n\}$ with $i_1 < i_2 < \dots < i_r$. We define*

$$x_I := (x_{i_1}, \dots, x_{i_r})^T \text{ and } A[I] := \begin{pmatrix} a_{1,i_1} & \dots & a_{1,i_r} \\ \vdots & & \vdots \\ a_{m,i_1} & \dots & a_{m,i_r} \end{pmatrix}.$$

*$x \in \mathbb{R}^n$ is a **basic solution** for the indices $\mathcal{B} = \{b_1, \dots, b_m\} \subseteq \{1, \dots, n\}$ if*

1. $Ax = b$
2. $x_i = 0$ *for all* $i \notin \mathcal{B}$
3. $A[\mathcal{B}]$ *is a regular matrix*[2]

*x is a **vertex** of $P(A,b)$ if x is a basic solution.*

The vertices of some polyhedron $P(A,b)$ are exactly the vertices of $P(A,b)$ in a geometrical sense - hence the name.

A bounded polyhedron is called a polytope which can be characterized via its finite set of vertices.

Definition 2.2.3 (polytope). *A **polytope** is a bounded polyhedron, i.e. there exists some $R > 0$ such that for all $x \in P(A,b)$ it holds*

$$\|x\| \le R.$$

Let $\{x_1, \dots, x_k\}$ be the (finite) set of vertices of some polytope $P(A,b)$. We have

$$P(A,b) = conv(x_1, \dots, x_n) = \{\alpha_1 \cdot x_1 + \dots + \alpha_k \cdot x_k \mid \alpha_1, \dots, \alpha_k \ge 0 \text{ and } \sum_{i=1}^{k} \alpha_i = 1\},$$

i.e. $P(A,b)$ can be written as the convex combination ($conv$) of its vertices.

Definition 2.2.4 (linear program). *A **linear program** (LP) is the following optimization problem:*

$$y^* = \min \; c^T x$$
$$\textit{such that } x \in P(A,b)$$

where $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$ and $P(A,b)$ is a polyhedron.

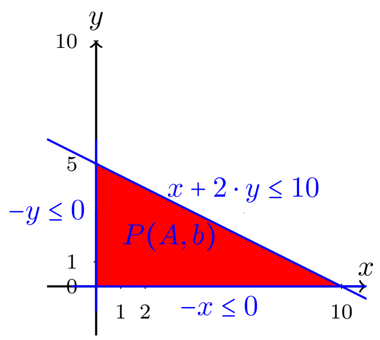
[2] A matrix $A \in \mathbb{R}^{n \times n}$ is regular if $rank(A) = n$ or, equivalently, if there exists some matrix $B \in \mathbb{R}^{n \times n}$ with $A \cdot B = B \cdot A = I_n$ where $I_n \in \mathbb{R}^{n \times n}$ denotes the unit matrix of size n.

Now, exactly one of the following three cases holds for any LP:

1. The LP is unsolvable because $P(A,b) = \emptyset$. For example: $A = (1,-1)^T$ and $b = (0,1)^T$.
2. The LP is unbounded, i.e. there exist infinitely many solutions with arbitrary high y^*. For example: $A = -1$, $b = 0$, $c = 1$.
3. The LP has an optimal solution $y^* = c^T x^*$ where x^* is a vertex of $P(A,b)$.

Whenever $P(A,b)$ is a non-empty polytope we can find the optimal solution at one of the finitely many vertices of $P(A,b)$. We note that in any LP min can easily be replaced by max since $\min c^T x = -\max(-c)^T x$.

Example 2.2.5. *Let* $A = \begin{pmatrix} 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{pmatrix}$, $b = (10,0,0)^T$ *and* $c = (-3,-4)^T$. *The polytope* $P(A,b)$ *has three vertices:* $x_1 = (0,5)^T$, $x_2 = (10,0)^T$ *and* $x_3 = (0,0)^T$ *(see the picture below). As one can see, the optimal solution* $y^* = -30$ *is attained at vertex* x_2.



Linear programs are widely studied and there exist a handful of methods which solve linear programs. For example, the ellipsoid method [KV12] computes the solution of any LP in polynomial time w.r.t. n and m.

The most common and practical method is the Simplex algorithm [KV12]. Here, we start at some vertex of $P(A,b)$ and if the optimal solution is not obtained we move to some different vertex. Since any vertex can not be visited twice we will at some point find an optimal vertex x^* of $P(A,b)$ and obtain the solution y^*. It has to be noted that the Simplex algorithm has exponential runtime (as the number of vertices is exponential w.r.t. n and m) in theory but proves very useful in practice. There exist a handful of modifications to the Simplex algorithm.

We refer to [KV12] for more details on linear programming, algorithms for solving linear programs and a handful of applications.

2.3. Complete Lattices and MV-Chains

The functions we aim to analyse in later parts are mainly of the form $f: L \to L$ and $f: \mathbb{M}^Y \to \mathbb{M}^Y$, where Y is some finite set, L is a complete lattice and $\mathbb{M}$ a complete MV-chain. In this section we introduce both algebraic structures.

2.3.1. Complete Lattices

We start with the more common lattice theory [DP02]. Lattice theory is a branch of mathematics that studies partially ordered sets and their properties. They arise in many areas of mathematics, including algebra, topology and computer science.

To begin, we define an order.

Definition 2.3.1 (order)**.** *A* ***(partial) order*** *over a set L is a binary relation $\sqsubseteq \subseteq L \times L$, such that*

- $\sqsubseteq$ *is reflexive: for all $l \in L$ it holds $l \sqsubseteq l$*
- $\sqsubseteq$ *is transitive: for all $l_1, l_2, l_3 \in L$; if $l_1 \sqsubseteq l_2$ and $l_2 \sqsubseteq l_3$ then $l_1 \sqsubseteq l_3$ holds*
- $\sqsubseteq$ *is anti-symmetric: for all $l_1, l_2 \in L$; if $l_1 \sqsubseteq l_2$ and $l_2 \sqsubseteq l_1$ then $l_1 = l_2$ holds*

An order is ***total****, if for all $l_1, l_2 \in L$ either $l_1 \sqsubseteq l_2$ or $l_2 \sqsubseteq l_1$ (or both) holds.*

A (partially) ordered set $(L, \sqsubseteq)$ is often denoted simply as L, omitting the order relation. The dual (partially) ordered set is given by $(L, \sqsupseteq)$, simply reversing the order.

Given $x, y \in L$, with $x \sqsubseteq y$, we denote by $[x, y]$ the interval $\{z \in P \mid x \sqsubseteq z \sqsubseteq y\}$.

Definition 2.3.2 (upper/lower bound)**.** *Let L be a (partially) ordered set and $Y \subseteq L$ a subset of L.*

- *An* ***upper bound*** *of Y is some element $ub \in L$, such that $y \sqsubseteq ub$ for all $y \in Y$. $lub \in L$ is the* ***least upper bound*** *(or* ***join****) of Y if $lub \sqsubseteq ub$ for all upper bounds ub of Y. A least upper bound may not exist. If it exists, we denote $\bigsqcup Y$ for the least upper bound of Y.*
- *A* ***lower bound*** *of Y is some element $lb \in L$, such that $lb \sqsubseteq y$ for all $y \in Y$. $glb \in L$ is the* ***greatest lower bound*** *(or* ***meet****) of Y if $lb \sqsubseteq glb$ for all lower bounds lb of Y. A greatest lower bound may not exist. If it exists, we denote $\bigsqcap Y$ for the greatest lower bound of Y.*

We write $l_1 \sqcup \cdots \sqcup l_n = \bigsqcup\{l_1, \ldots, l_n\}$ and $l_1 \sqcap \cdots \sqcup l_n = \bigsqcap\{l_1, \ldots, l_n\}$ for $l_1, \ldots, l_n \in L$. Next, we can already define complete lattices.

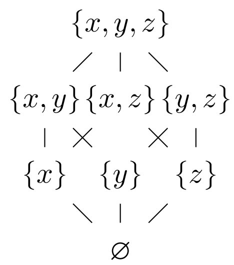


Fig. 2.1.: Hasse-Digramm for $X = \{x, y, z\}$, non-transitive elements in order relation are connected.

Definition 2.3.3 (complete lattice)**.** *A (partially) ordered set L is a **complete lattice** if every subset $Y \subseteq L$ has a least upper bound $\bigsqcup Y$ and a greatest lower bound $\bigsqcap Y$. We denote $\bot = \bigsqcap L$ and $\top = \bigsqcup L$.*

We will see that complete lattices have desirable properties and are fundamental for the theories in later chapters.

Example 2.3.4. *We name a few complete lattices used throughout this thesis. We denote $\leq$ to be the standard order on the reals.*

1. $(\{0, 1, 2, \dots, k\}, \leq)$ *with $k \in \mathbb{N}$ is a complete lattice.*
2. $(\mathbb{N}^\infty, \leq)$ *is a complete lattice.*
3. $([k_1, k_2], \leq)$ *where $k_1, k_2 \in \mathbb{R}$ is a complete lattice.*
4. $(L^Y, \sqsubseteq)$ *is a complete lattice whenever $(L, \sqsubseteq')$ is a complete lattice and Y some finite set. The (partial) order $\sqsubseteq$ is defined for $a, b \in L^Y$ as: $a \sqsubseteq b$ iff $a(y) \sqsubseteq' b(y)$ for all $y \in Y$.*
5. *Given some set X, $(\mathcal{P}(X), \subseteq)$ is a complete lattice. See Figure 2.1 for an example.*

For a better understanding we name two counterexamples

- $(\mathbb{N}, \leq)$ *is not a complete lattice. We have $\bigsqcup \mathbb{N} = \infty \notin \mathbb{N}$.*
- $(\mathbb{Q}, \leq)$ *is not a complete lattice. Let*

$$Y = \{x \in \mathbb{Q} \mid x^2 < 2\},$$

then $\bigsqcup Y = \sqrt{2}$ on $(\mathbb{R}, \leq)$, but $\sqrt{2} \notin \mathbb{Q}$.

Quantales. A quantale is a complete lattice equipped with a binary multiplication operator.

Definition 2.3.5 (quantale)**.** *A **quantale** $(L, *)$ is a complete lattice L with an associative binary operator $*\colon L \times L \to L$ called its multiplication satisfying the following distributive property*

$$l * (\bigsqcup_{i \in I} l_i) = \bigsqcup_{i \in I} (l * l_i) \text{ and } (\bigsqcup_{i \in I} l_i) * l = \bigsqcup_{i \in I} (l_i * l)$$

*for all $l, l_i \in L$ where I is any index set. The quantale is **unital** if it has an identity element e for its multiplication, i.e.*

$$l * e = l = e * l$$

*for all $l \in L$. A quantale $(L, *)$ is called **commutative** if the operator $*$ is commutative, i.e. $l_1 * l_2 = l_2 * l_1$ for all $l_1, l_2 \in L$.*

Complete Distributive Lattices. As we will be using this property later on, the lattice $([0, 1], \leq)$ is a complete distributive lattice.

Definition 2.3.6 (complete distributive lattice)**.** *A complete lattice L is **completely distributive** if, for any doubly indexed family $\{x_{j,k} \mid j \in J, k \in K_j\} \subseteq L$ we have*

$$\bigsqcup_{j \in J} \bigsqcap_{k \in K_j} x_{j,k} = \bigsqcap_{f \in F} \bigsqcup_{j \in J} x_{j,f(j)}$$

where $F = \{f\colon J \to \bigcup_{j \in J} K_j \mid f(j) \in K_j\}$, also called the set of choice functions. Complete distributivity is a self-dual property, i.e. dualizing the above statement yields the same class of complete lattices.

2.3.2. Monotone Functions on Complete Lattices

We now consider endofunctions over complete lattices.

Definition 2.3.7 (fixpoints)**.** *Let $f\colon L \to L$ be a function and $(L, \sqsubseteq)$ a complete lattice. We define*

- *The set of **fixpoints** of f: $Fix(f) = \{l \in L \mid f(l) = l\}$.*
- *The set of **pre-fixpoints** of f: $Pre(f) = \{l \in L \mid f(l) \sqsubseteq l\}$.*
- *The set of **post-fixpoints** of f: $Pre(f) = \{l \in L \mid l \sqsubseteq f(l)\}$.*

Additionally, we denote

- *The **greatest fixpoint** of f: $gfp(f) = \bigsqcup Fix(f) =: \nu f$*
- *The **least fixpoint** of f: $lfp(f) = \bigsqcap Fix(f) =: \mu f$*

Fixpoints, especially least/greatest fixpoints, are an essential focus of this thesis. In the following we provide a few fixpoint theorems. These hold for monotone functions.

Definition 2.3.8 (properties of functions)**.** *Let $(L_1, \sqsubseteq_1)$ and $(L_2, \sqsubseteq_2)$ be a (partially) ordered sets. A function $f\colon L_1 \to L_2$ is*

- ***monotone** if for all $l_1, l_2 \in L_1$, whenever $l_1 \sqsubseteq_1 l_2$ then $f(l_1) \sqsubseteq f(l_2)$ holds.*
- ***additive** if for all $l_1, l_2 \in L_1$ it holds $f(l_1 \sqcup l_2) = f(l_1) \sqcup f(l_2)$.*
- ***multiplicative** if for all $l_1, l_2 \in L_1$ it holds $f(l_1 \sqcap l_2) = f(l_1) \sqcap f(l_2)$.*

Knaster-Tarski's Theorem is at the heart of lattice theory providing a characterization of the least and greatest fixpoint.

Theorem 2.3.9 (Knaster-Tarski [Tar55])**.** *Let L be a complete lattice and $f\colon L \to L$ a monotone function. Then the following holds:*

$$\nu f = \bigsqcup Post(f) \text{ and } \mu f = \bigsqcap Pre(f).$$

In Figure 2.2 we see a schematical representation of a complete lattice L with the set of fixpoints, pre-fixpoints and post-fixpoints of some monotone function $f\colon L \to L$. Note that the intersection of the set of pre- and post-fixpoints is the set of fixpoints.

Now, to compute a least or greatest fixpoint of a function we can use Kleene iteration. First we define:

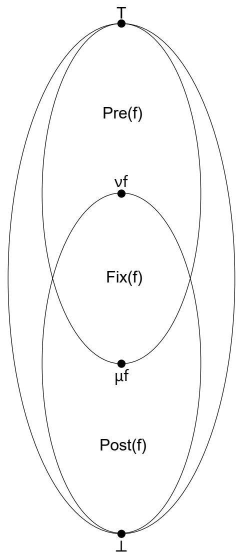


Fig. 2.2.: Complete Lattice L with pre-/post-/fixpoints of some function $f\colon L \to L$

Definition 2.3.10 (chains/Scott-(co)continuity)**.** *Let L be a (paritally) ordered set.*

- *A* ***descending chain*** *is a finite or infinite sequence $l_0, l_1, l_2, \dots$ (denoted by $(l_n)_n^{\downarrow}$) such that $l_{i+1} \sqsubseteq l_i$ for all $i \geq 0$. A function $f\colon L \to L$ is* ***Scott-continuous*** *if for all descending chains $(l_n)_n^{\downarrow}$ it holds $f(\bigsqcap_n l_n) = \bigsqcap_n f(l_n)$.*
- *An* ***ascending chain*** *is a finite or infinite sequence $l_0, l_1, l_2, \dots$ (denoted by $(l_n)_n^{\uparrow}$) such that $l_i \sqsubseteq l_{i+1}$ for all $i \geq 0$. A function $f\colon L \to L$ is* ***Scott-cocontinuous*** *i for all ascending chains $(l_n)_n^{\uparrow}$ it holds $f(\bigsqcup_n l_n) = \bigsqcup_n f(l_n)$.*

Now we present Kleene's Theorem which gives the blueprint to a simple algorithm capable of computing least and greatest fixpoints.

Theorem 2.3.11 (Kleene [NNH10])**.** *Let L be a complete lattice and $f\colon L \to L$ a monotone function. If f is Scott-continuous, respectively Scott-cocontinuous, then*

$$\nu f = \bigsqcap_{i=0}^{\infty} f^i(\top), \text{ respectively } \mu f = \bigsqcup_{i=0}^{\infty} f^i(\bot).$$

The above theorem gives rise to a simple iteration which approaches the least/greatest fixpoint of some monotone function $f\colon L \to L$. Given some $f\colon L \to L$ over a complete lattice L. We obtain the following iteration:

1. *Initiate:* $f^{(0)} = \bot$ **or** $f^{(0)} = \top$, $i = 0$
2. *Iterate:* $f^{(i+1)} = f(f^{(i)})$, $i > 0$

According to Theorem 2.3.11 the above iteration converges towards μf **or** νf. In some instances, if L satisfies the descending/ascending chain condition[3], there exists some $i \in \mathbb{N}$ with $f^{(i)}(\bot) = \mu f$ / $f^{(i)}(\top) = \nu f$. In general this may not be the case. Thus, when implementing the above iteration, we need to terminate the iteration at some point. For example, we stop if two successive iterations deviate by less than some small value ε (in some norm). We obtain:

1. *Initiate:* $f^{(0)} = \bot$ **or** $f^{(0)} = \top$, $i = 0$
2. *Iterate:*
   a) $f^{(i+1)} = f(f^{(i)})$, $i = i + 1$
   b) If $\|f^{(i)} - f^{(i-1)}\| > \varepsilon$: GOTO step 2(a),
      Else: STOP - Output the approximation $f^{(i)}$ of μf **or** νf.

The above procedure will be referred to as Kleene iteration and we will frequently use it to compute (approximate) least or greatest fixpoints. After terminating at iteration i it holds $\|f^{(i)} - \mu f\| < \varepsilon$ but it is unclear how many iteration this will take.

Example 2.3.12. *We do a simple Kleene iteration on the complete lattice* $([0,1], \leq)$*. Let* $f\colon [0,1] \to [0,1]$ *be given by* $f(x) = \frac{x+1}{2}$*. Clearly,* f *is monotone on* $[0,1]$ *and we proceed as follows:*

- $f^{(0)} = \bot = 0$
- $f^{(1)} = f(f^{(0)}) = f(0) = \frac{1}{2}$
- $f^{(2)} = f(f^{(1)}) = f(\frac{1}{2}) = \frac{3}{4}$
- $f^{(3)} = f(f^{(2)}) = f(\frac{3}{4}) = \frac{7}{8}$
- $\vdots$
- $f^{(i)} = \frac{2^i - 1}{2^i}$

The iteration converges towards $\mu f = 1$.

Almost all functions considered throughout this thesis are Scott-(co)continuous and whenever we apply Kleene iteration the function of interest is Scott-continuous/-cocontinuous (see Remark 2.3.26).

[3] e.g. every descending/ascending chain in L becomes stationary

Abstract Interpretation and Galois Connections. We briefly describe what abstract interpretation entails and define a Galois connection which can be used to obtain such an abstract interpretation. The main idea of abstract interpretation is to find a sound approximation of computer programs. These are based on monotone functions over complete lattices. Abstract interpretation can be viewed as a partial execution of a computer program which gains information about its semantics without performing all the calculations.

Definition 2.3.13 (Galois connection)**.** *Given two complete lattices* $(C, \sqsubseteq_1)$ *and* $(A, \sqsubseteq_2)$*. A pair* (α, γ) *of monotone functions* $\alpha\colon C \to A$ *and* $\gamma\colon A \to C$ *is a* ***Galois connection*** *if the following hold:*

- *for all* $c \in C$ *it holds* $c \sqsubseteq_1 \gamma(\alpha(c))$*,*
- *for all* $a \in A$ *it holds* $\alpha(\gamma(a)) \sqsubseteq_2 a$*.*

We refer to α as the abstraction assigning some abstract value to each element in L and γ as the conretization which assigns concrete values to each abstract element.

The following property holds for a Galois connection.

Lemma 2.3.14 ([CC77])**.** *Given two complete lattices* $(C, \sqsubseteq_1)$ *and* $(A, \sqsubseteq_2)$ *and a Galois-connection* (α, γ) *between them. For* $c \in C$ *and* $a \in A$ *it holds that*

$$\alpha(c) \sqsubseteq_2 a \Leftrightarrow c \sqsubseteq_1 \gamma(a).$$

Galois connections are at the heart of abstract interpretation [CC77, CC00]. In particular, when $\langle \alpha, \gamma \rangle$ is a Galois connection, given $f^C : C \to C$ and $f^A : A \to A$, monotone functions, if $f^C \circ \gamma \sqsupseteq \gamma \circ f^A$, then $\nu f^C \sqsupseteq \gamma(\nu f^A)$ (pointwise extension of the order). If the equality $f^C \circ \gamma = \gamma \circ f^A$ holds, a condition sometimes referred to as γ-*completeness*, then greatest fixpoints are preserved along the connection, i.e., $\nu f^C = \gamma(\nu f^A)$ [BKP20].

In Section 3.2 we will define a Galois-connection which lies at the heart of the theory in Chapter 3.

2.3.3. MV-Chains

The theories we will introduce later require a special kind of lattice enriched with a strong algebraic structure. The properties of MV-algebras fit our needs as the theory in Chapter 3 does not work on complete lattices. See [Mun07] for a nice tutorial on MV-algebras.

Definition 2.3.15 (MV-algebra)**.** *An MV-Algebra is a tuple* $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *where*

- M *is a set*
- $\oplus\colon M \times M \to M$ *is a binary operator on* M
- $0 \in M$ *is the neutral element in* M
- $\overline{(\cdot)}\colon M \to M$ *maps each element to its complement*

fulfilling the following properties:

1. $(M, \oplus, 0)$ *is a commutative monoid, i.e. for all* $x, y, z \in M$ *it holds*
   a) *Associativity:* $(x \oplus y) \oplus z = x \oplus (y \oplus z)$
   b) *Neutral element:* $x \oplus 0 = 0 \oplus x = x$
   c) *Commutativity:* $x \oplus y = y \oplus x$
2. $\overline{\overline{x}} = x$
3. $x \oplus \overline{0} = \overline{0}$
4. $\overline{(\overline{x} \oplus y)} \oplus y = \overline{(\overline{y} \oplus x)} \oplus x$

We denote $1 = \overline{0}$, $x \otimes y = \overline{\overline{x} \oplus \overline{y}}$ and $x \ominus y = x \otimes \overline{y} = \overline{\overline{x} \oplus y}$. Thus we can reformulate axioms 3 and 4:

3. $x \oplus 1 = 1$
4. $(y \ominus x) \oplus x = (x \ominus y) \oplus y$

For an MV-algebra $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$, we define its dual MV-algebra as $\mathbb{M}^{op} = (M, \otimes, 1, \overline{(\cdot)})$.

MV-algebras are endowed with a natural order.

Definition 2.3.16 (natural order)**.** *Let* $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *be an MV-algebra. The natural order on* $\mathbb{M}$ *is defined, for* $x, y \in M$*, by* $x \sqsubseteq y$ *if* $x \oplus z = y$ *for some* $z \in M$*. When* $\sqsubseteq$ *is total,* $\mathbb{M}$ *is called an MV-chain.*

The natural order gives an MV-algebra a lattice structure where $\bot = 0$, $\top = 1$, $\bigsqcup\{x, y\} = x \sqcup y = (x \ominus y) \oplus y$ and $\bigsqcap\{x, y\} = x \sqcap y = \overline{\overline{x} \sqcup \overline{y}} = x \otimes (\overline{x} \oplus y)$. We call the MV-algebra complete, if it is a complete lattice. This is not true in general, e.g. $([0, 1] \cap \mathbb{Q}, \oplus, 0, \overline{(\cdot)})$ with natural order $\leq$.

Given an MV-chain $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ with (total) natural order $\sqsubseteq$ then the reversed order $\sqsubseteq_{op}=\sqsupseteq$ is the (total) natural order in the dual MV-algebra $\mathbb{M}^{op} = (M, \otimes, 1, \overline{(\cdot)})$ as

$$\begin{aligned} x \sqsubseteq_{op} y &\Leftrightarrow \exists z:\ x \otimes z = y \Leftrightarrow \exists z:\ \overline{\overline{x} \oplus \overline{z}} = y \\ &\Leftrightarrow \exists z:\ \overline{y} = \overline{x} \oplus \overline{z} \Leftrightarrow \exists z':\ \overline{y} = \overline{x} \oplus z' \\ &\Leftrightarrow \overline{x} \sqsubseteq \overline{y} \Leftrightarrow y \sqsubseteq x \end{aligned}$$

for all $x, y \in M$.

Throughout later chapters we will exclusively work with complete MV-chains. We are mainly interested in the following two complete MV-chains.

Example 2.3.17. $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *are complete MV-chains for the following instances:*

1. *Let* $k \in \mathbb{R}$ *with* $k > 0$*. Set* $M = [0, k]$*,* $x \oplus y = \min\{x + y, k\}$ *(truncated addition) and* $\overline{x} = k - x$ *for* $x, y \in M$*. We obtain* $x \ominus y = \max\{x - y, 0\}$ *(truncated substraction) and* $x \otimes y = \max\{x + y - k, 0\}$*.*
2. *Let* $k \in \mathbb{N}$ *with* $k > 0$*. Set* $M = \{0, 1, 2, \dots, k\}$*,* $x \oplus y = \min\{x + y, k\}$ *(truncated addition) and* $\overline{x} = k - x$ *for* $x, y \in M$*. We obtain* $x \ominus y = \max\{x - y, 0\}$ *(truncated substraction) and* $x \otimes y = \max\{x + y - k, 0\}$*.*

We next review some properties of MV-algebras. They are taken from or easy consequences of properties in [Mun07] and will be used sparingly throughout this thesis. The proof is original research and a contribution made in this thesis.

Lemma 2.3.18 (properties of MV-algebras)**.** *Let* $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *be an MV-algebra. For all* $x, y, z \in M$ *it holds*

1. $x \oplus \overline{x} = 1$
2. $x \sqsubseteq y$ *iff* $\overline{x} \oplus y = 1$ *iff* $x \otimes \overline{y} = 0$ *iff* $y = x \oplus (y \ominus x)$
3. $x \sqsubseteq y$ *iff* $\overline{y} \sqsubseteq \overline{x}$
4. $\oplus$, $\otimes$ *are monotone in both arguments,* $\ominus$ *monotone in the first and antitone in the second argument.*
5. *if* $x \sqsubset y$ *then* $0 \sqsubset y \ominus x$*;*
6. $(x \oplus y) \ominus y \sqsubseteq x$
7. $z \sqsubseteq x \oplus y$ *if and only if* $z \ominus x \sqsubseteq y$*.*
8. *if* $x \sqsubset y$ *and* $z \sqsubseteq \overline{y}$ *then* $x \oplus z \sqsubset y \oplus z$*;*
9. $y \sqsubseteq \overline{x}$ *if and only if* $(x \oplus y) \ominus y = x$*;*
10. $x \ominus (x \ominus y) \sqsubseteq y$ *and if* $y \sqsubseteq x$ *then* $x \ominus (x \ominus y) = y$*.*
11. *Whenever* $\mathbb{M}$ *is an MV-chain,* $x \sqsubset y$ *and* $0 \sqsubset z$ *imply* $(x \oplus z) \ominus y \sqsubset z$

Proof. See Appendix: Lemma A.1.1. □

We are explicitly interested in sets of functions of the kind $\mathbb{M}^Y$ where $\mathbb{M}$ is a (complete) MV-chain and Y is a finite set. We extend $\oplus$, $\ominus$ and $\otimes$ to $\mathbb{M}^Y$ pointwise, e.g. for $a, b \in \mathbb{M}^Y$, we write $a \oplus b$ for the function defined by $(a \oplus b)(y) = a(y) \oplus b(y)$ for all $y \in Y$.

We will also consider distributions over MV-chains.

Definition 2.3.19 (distributions over MV-chains)**.** *Given an MV-chain* $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *with natural order* $\sqsubseteq$ *and some finite set* Y*, we call a function* $p : Y \to \mathbb{M}$ *a distribution when for all* $y \in Y$*, it holds*

$$\overline{p(y)} = \bigoplus_{y' \in Y \setminus \{y\}} p(y').$$

$\mathcal{D}_{\mathbb{M}}(Y)$ *denotes the set of distributions.*

Assume that $\mathbb{M}$ is endowed with an additional operation $\odot$ such that $(\mathbb{M}, \odot, 1)$ is a commutative monoid, for $x, y \in \mathbb{M}$, $x \odot y \sqsubseteq x$, and $x \odot y = 0$ iff $x = 0$ or $y = 0$, and $\odot$ weakly distributes over $\oplus$, i.e., for all $x, y, z \in \mathbb{M}$ with $y \sqsubseteq \overline{z}$, $x \odot (y \oplus z) = x \odot y \oplus x \odot z$. For $p \in \mathcal{D}_{\mathbb{M}}(Y)$ and $a : Y \to \mathbb{M}$, the *average sum* is given by

$$\bigoplus_{y \in Y} \big(p(y) \odot a(y)\big).$$

The usual probability distributions (see Definition 2.1.4) arise as a special case of $\mathcal{D}_{\mathbb{M}}(Y)$ with $\mathbb{M} = [0, 1]$ where $\odot$ is the standard multiplication.

Also note that in the characterization of the average sum, the operation $\odot$ is necessarily monotone. In fact, if $y \sqsubseteq y'$ then, by Lemma 2.3.18(2), we have $y' = y \oplus (y' \ominus y)$. Therefore $x \odot y \sqsubseteq (x \odot y) \oplus \big(x \odot (y' \ominus y)\big) = x \odot (y \oplus (y' \ominus y)) = x \odot y'$, where the second passage holds by weak distributivity.

It must be remarked that every complete MV-algebra is a quantale with respect to $\oplus$ and the inverse of the natural order.

Lemma 2.3.20 (complete MV-algebras are quantales)**.** *Let* $\mathbb{M}$ *be a complete MV-algebra. Then* $(\mathbb{M}, \oplus, \sqsupseteq)$ *is a unital and commutative quantale.*

Proof. See Appendix: Lemma A.1.2. □

2.3.4. Non-expansive Functions on MV-Chains

Next, for functions $f : \mathbb{M}^Y \to \mathbb{M}^Z$ (Y and Z are finite sets) we aim to define non-expansiveness. This is an essential property used throughout this thesis. For defining non-expansiveness it is convenient to introduce a norm, which can be seen as an adaptation of the standard ∞-norm.

Definition 2.3.21 (norm)**.** *Let $\mathbb{M}$ be an MV-chain and let Y be a finite set. Given $a \in \mathbb{M}^Y$ we define its norm as $\|a\| = \max\{a(y) \mid y \in Y\}$.*

$\|\cdot\|$ is clearly monotone, i.e. if $a \sqsubseteq b$ then $\|a\| \sqsubseteq \|b\|$, and has the standard properties of a norm:

Lemma 2.3.22. *Let $\mathbb{M}$ be an MV-chain and let Y be a finite set. Then $\|\cdot\|: \mathbb{M}^Y \to \mathbb{M}$ satisfies, for all $a, b \in \mathbb{M}^Y$ and $\delta \in \mathbb{M}$*

1. $\|a \oplus b\| \sqsubseteq \|a\| \oplus \|b\|$
2. $\|\delta \otimes a\| = \delta \otimes \|a\|$
3. $\|a\| = 0$ *implies that a is the constant 0.*

Proof. Concerning (1), let $\|a \oplus b\|$ be realised on some element $y \in Y$, i.e., $\|a \oplus b\| = a(y) \oplus b(y)$. Since $a(y) \sqsubseteq \|a\|$ and $b(y) \sqsubseteq \|b\|$, by monotonicity of $\oplus$ we deduce that $\|a \oplus b\| \sqsubseteq \|a\| \oplus \|b\|$.

Concerning (2), note that

$$\begin{aligned}
\|\delta \otimes a\| &= \max\{\overline{\overline{\delta} \oplus \overline{a(y)}} \mid y \in Y\} \\
&= \overline{\min\{\overline{\delta} \oplus \overline{a(y)} \mid y \in Y\}} \\
&= \overline{\overline{\delta} \oplus \min\{\overline{a(y)} \mid y \in Y\}} \\
&= \overline{\overline{\delta} \oplus \overline{\max\{a(y) \mid y \in Y\}}} \\
&= \overline{\overline{\delta} \oplus \overline{\|a\|}} \\
&= \delta \otimes \|a\|
\end{aligned}$$

Finally, point (3) is straightforward, since 0 is the bottom of $\mathbb{M}$. ☐

We now define non-expansiveness which is a crucial property we often require.

Definition 2.3.23 (non-expansive)**.** *Let $f: \mathbb{M}^Y \to \mathbb{M}^Z$ be a function where $\mathbb{M}$ is an MV-chain and Y, Z are finite sets. We say that f is* ***non-expansive*** *if for all $a, b \in \mathbb{M}^Y$ it holds that $\|f(b) \ominus f(a)\| \sqsubseteq \|b \ominus a\|$.*

Note that $(a, b) \mapsto \|a \ominus b\|$ is the supremum lifting of a directed version of Chang's distance [Mun07]. Moreover, when $\mathbb{M} = \{0, 1\}$, i.e., $\mathbb{M}$ is the two-point boolean algebra, the two notions coincide.

We note that this definition of non-expansiveness is different from Definition 2.1.10 but we will exclusively refer to non-expansive functions as in the definition above (unless stated otherwise).

It is easy to see that all non-expansive functions on MV-chains are monotone

Lemma 2.3.24 (non-expansiveness implies monotonicity). *Let $\mathbb{M}$ be an MV-chain and let Y, Z be finite sets. Every non-expansive function $f : \mathbb{M}^Y \to \mathbb{M}^Z$ is monotone.*

Proof. Let $a, b \in \mathbb{M}^Y$ be such that $a \sqsubseteq b$. Therefore, by Lemma 2.3.18(2), $a(y) \ominus b(y) = 0$ for all $y \in Y$, hence $a \ominus b = 0$. Thus $\|f(a) \ominus f(b)\| \sqsubseteq \|a \ominus b\| = 0$. In turn this implies that for all $z \in Z$, $f(a)(z) \ominus f(b)(z) = 0$. Hence Lemma 2.3.18(2), allows us to conclude $f(a)(z) \sqsubseteq f(b)(z)$ for all $z \in Z$, i.e., $f(a) \sqsubseteq f(b)$, as desired. □

The next lemma provides a useful equivalent characterisation of non-expansiveness.

Lemma 2.3.25 (characterization of non-expansiveness). *Let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a monotone function, where $\mathbb{M}$ is an MV-chain and Y, Z are finite sets. Then f is non-expansive iff for all $a \in \mathbb{M}^Y$, $\theta \in \mathbb{M}$ and $z \in Z$ it holds $f(a \oplus \theta)(z) \ominus f(a)(z) \sqsubseteq \theta$.*

Proof. See Appendix: Lemma A.1.3. □

As we will show in Section 3.5, non-expansive functions enjoy good closure properties (closure under composition and closure under disjoint union) which will prove very useful.

Remark 2.3.26. *For a finite set Y it holds that any non-expansive function $f\colon [0,1]^Y \to [0,1]^Y$ is continuous, thus Scott-continuous/cocontinuous. Additionally, for a non-expansive function $f\colon \{0,1,\dots,k\}^Y \to \{0,1,\dots,k\}^Y$ ($k \in \mathbb{N}$) the set of functions $\{0,1,\dots,k\}^Y$ is finite which guarantees Scott-continuity/cocontinuity of f.*

2.4. Category Theory

In this section we give a short overview on category theory. We will mainly list the important definitions and examples we will use throughout this thesis. We start by defining a category.

Definition 2.4.1 (category). *A **category** $\mathbb{A}$ consists of*

- *a collection $ob(\mathbb{A})$ of **objects***
- *for each $A, B \in ob(\mathbb{A})$, a collection $\mathbb{A}(A, B)$ of **arrows** or **maps** or **morphisms** from A to B*
- *for each $A, B, C \in ob(\mathbb{A})$, a function*

$$\begin{aligned} \mathbb{A}(B, C) \times \mathbb{A}(A, B) &\to \mathbb{A}(A, C) \\ (g, f) &\mapsto g \circ f \end{aligned}$$

  *called **composition***

- *for each $A \in ob(\mathbb{A})$, an element id_A of $\mathbb{A}(A, A)$, called the **identity** of A*

satisfying the following axioms:

- ***associativity:** for each $f \in \mathbb{A}(A, B)$, $g \in \mathbb{A}(B, C)$ and $h \in \mathbb{A}(C, D)$, we have*

$$(h \circ g) \circ f = h \circ (g \circ f)$$

- ***identity laws:** for each $f \in \mathbb{A}(A, B)$, we have*

$$f \circ id_A = f = id_B \circ f$$

We provide a few categories used throughout this thesis.

Example 2.4.2. *The following are well-defined categories:*

1. *Category* **Set***:*
   - *objects in* **Set** *are all sets*
   - *arrows in* **Set** *between objects $A, B \in ob($***Set***$)$ are maps from A to B, i.e. $arr(A, B) = B^A$*
   - *composition of arrows is the usual function composition*
   - *identity id_A for some $A \in ob(A)$ is the usual identity, i.e. $id_A(a) = a$ for all $a \in A$*
2. *Category **PMet** (see Definitions 2.1.8, 2.1.10):*
   - *objects in **PMet** are $\top$-pseudometric spaces (X, d) for some fixed $\top > 0$*
   - *arrows are non-expansive functions (see Definition 2.1.10) between these spaces*
   - *composition of arrows is the usual function composition*
   - *identities are the (isometric) identity functions*

3. *Product category $\mathbb{A} \times \mathbb{B}$ of categories $\mathbb{A}$ and $\mathbb{B}$:*
   - *objects in $\mathbb{A} \times \mathbb{B}$ are ordered pairs (A, B) with $A \in ob(\mathbb{A})$ and $B \in ob(\mathbb{B})$*
   - *arrows are ordered pairs $((A \xrightarrow{f} A'), (B \xrightarrow{g} B'))$ of arrows in $\mathbb{A}$ and $\mathbb{B}$*
   - *composition of arrows is defined componentwise by composition in $\mathbb{A}$ and $\mathbb{B}$*
   - *identities are the componentwise identities*

At the heart of category theory lie maps between categories, named functors.

Definition 2.4.3 (functor)**.** *Let $\mathbb{A}$ and $\mathbb{B}$ be categories. A **functor** $F: \mathbb{A} \to \mathbb{B}$ consists of:*

- *a map*

$$ob(\mathbb{A}) \to ob(\mathbb{B}),$$

  written as $A \mapsto F(A)$; we might write FA for $F(A)$

- *for each $A, A' \in \mathbb{A}$, a function*

$$\mathbb{A}(A, A') \to \mathbb{B}(F(A), F(A')),$$

  written as $f \mapsto F(f)$

satisfying the following axioms:

- *$F(f' \circ f) = F(f') \circ F(f)$ whenever $A \xrightarrow{f} A' \xrightarrow{f'} A''$ are arrows in $\mathbb{A}$*
- *$F(id_A) = id_{F(A)}$ whenever $A \in ob(\mathbb{A})$*

Again, we provide a handful of functors used throughout this thesis.

Example 2.4.4. *The following are well-defined functors:*

1. *The powerset functor $\mathcal{P}: \mathbf{Set} \to \mathbf{Set}$ maps*
   - *a set $X \in ob(\mathbf{Set})$ to its powerset $\mathcal{P}(X) \in ob(\mathbf{Set})$*
   - *an arrow $f \in \mathbf{Set}(X, Y)$ to $\mathcal{P}f \in \mathbb{S}{\approx}(\mathcal{P}(X), \mathcal{P}(Y))$ defined for $U \in \mathcal{P}(X)$ as $\mathcal{P}f(U) = \{f(u) \mid u \in U\} \in \mathcal{P}(Y)$*
2. *The finite powerset functor $\mathcal{P}_f: \mathbf{Set} \to \mathbf{Set}$ maps*
   - *a set $X \in ob(\mathbf{Set})$ to its finite powerset $\mathcal{P}_f(X) \in ob(\mathbf{Set})$*
   - *an arrow $f: X \to Y$ to $\mathcal{P}_f f: \mathcal{P}_f(X) \to \mathcal{P}_f(Y)$ defined for $U \in \mathcal{P}_f(X)$ as $\mathcal{P}_f f(U) = \{f(u) \mid u \in U\} \in \mathcal{P}_f(Y)$*
3. *The distribution functor $\mathcal{D}: \mathbf{Set} \to \mathbf{Set}$ maps*

- *a set* $X \in ob(\mathbf{Set})$ *to its set of probability distributions* $\mathcal{D}(X) \in ob(\mathbf{Set})$
- *an arrow* $f: X \to Y$ *to* $\mathcal{D}f: \mathcal{D}(X) \to \mathcal{D}(Y)$ *defined for* $p \in \mathcal{D}(X)$ *and* $y \in Y$ *as* $\mathcal{D}f(p)(y) = \sum_{x \in f^{-1}(\{y\})} p(x) \in \mathcal{D}(Y)$

4. *The finitely supported distribution functor* $\mathcal{D}_f: \mathbf{Set} \to \mathbf{Set}$ *maps*
   - *a set* $X \in ob(\mathbf{Set})$ *to its set of probability distributions* $\mathcal{D}_f(X) \in ob(\mathbf{Set})$ *with finite support*
   - *an arrow* $f: X \to Y$ *to* $\mathcal{D}_f f: \mathcal{D}_f(X) \to \mathcal{D}_f(Y)$ *defined for* $p \in \mathcal{D}_f(X)$ *and* $y \in Y$ *as* $\mathcal{D}_f f(p)(y) = \sum_{x \in f^{-1}(\{y\})} p(x) \in \mathcal{D}_f(Y)$
5. *The forgetful functor* $U: \boldsymbol{PMet} \to \mathbf{Set}$ *maps*
   - *a pseudometric space* $(X, d) \in ob(\boldsymbol{PMet})$ *to its underlying set* $X \in \mathbf{Set}$
   - *an arrow* $f: (X, d_X) \to (Y, d_Y)$ *to itself (*$f: X \to Y$*)*
6. *The swap functor* $X: \mathbb{A} \times \mathbb{B} \to \mathbb{B} \times \mathbb{A}$ *(*$\mathbb{A}$ *and* $\mathbb{B}$ *are categories) maps*
   - (A, B) *to* (B, A) *for objects* $A \in ob(\mathbb{A})$ *and* $B \in ob(\mathbb{B})$
   - *an arrow* $((A \xrightarrow{f} A'), (B \xrightarrow{g} B'))$ *to* $((B \xrightarrow{g} B'), (A \xrightarrow{f} A'))$
7. *The diagonal functor* $\Delta: \mathbb{A} \to \mathbb{A} \times \mathbb{A}$ *(*$\mathbb{A}$ *is a category) maps*
   - A *to* (A, A) *for an object* $A \in ob(\mathbb{A})$
   - *an arrow* $f: A \to A'$ *to* $((A \xrightarrow{f} A'), (A \xrightarrow{f} A'))$

We will sparingly use bifunctors which we spare to generalize further.

Definition 2.4.5 (bifunctor)**.** *Let* $\mathbb{A}$*,* $\mathbb{A}'$ *and* $\mathbb{B}$ *be categories. A* ***bifunctor*** F *is a functor* $F: \mathbb{A} \times \mathbb{A}' \to \mathbb{B}$ *where* $\mathbb{A} \times \mathbb{A}'$ *is the product category.*

Natural transformations can be seen as transformations between functors which respect the internal structures.

Definition 2.4.6 (natural transformation)**.** *Let* $\mathbb{A}$ *and* $\mathbb{B}$ *be categories and let* $\mathbb{A} \overset{F}{\to} \mathbb{B}$ *and* $\mathbb{A} \overset{G}{\to} \mathbb{B}$ *be functors. A* ***natural transformation*** *is a family* $\big(F(A) \overset{\alpha_a}{\to} G(A)\big)_{A \in ob(\mathbb{A})}$ *of arrows in* $\mathbb{B}$ *such that for every arrow* $A \overset{f}{\to} A'$ *in* $\mathbb{A}$*, the square*

$$\begin{array}{ccc} F(A) & \overset{F(f)}{\to} & F(A') \\ \alpha_A \downarrow & & \downarrow \alpha_{A'} \\ G(A) & \underset{G(f)}{\to} & G(A') \end{array}$$

commutes. The maps α_A *are called components of* α*.*

2.4.1. GS-Monoidal Categories

For the compositional modelling of graphs and graph-like structures it has proven useful to use the notion of monoidal categories [Mac71], i.e., categories equipped with a tensor product.

Definition 2.4.7 (monoidal category)**.** *A strict* ***monoidal category*** $\mathbb{C}$ *is a tuple* $(\mathbb{C}_0, \otimes, e)$ *where* $\mathbb{C}_0$ *is a category,* $e \in ob(\mathbb{C}_0)$ *is a distinguished object and* $\otimes\colon \mathbb{C}_0 \times \mathbb{C}_0 \to \mathbb{C}_0$ *is a bifunctor, satisfying the following axioms:*

- $(f \otimes g) \otimes h = f \otimes (g \otimes h)$ *for all* $f, g, h \in arr(\mathbb{C}_0)$
- $f \otimes id_e = id_e \otimes f = f$ *for all* $f \in arr(\mathbb{C}_0)$

It has long been realized that monoidal categories can have additional structure such as braiding or symmetries.

Definition 2.4.8 (symmetric monoidal category)**.** *A strict* ***symmetric monoidal category*** $\mathbb{C}$ *is a tuple* $(\mathbb{C}_0, \otimes, e, \rho)$ *where* $(\mathbb{C}_0, \otimes, e)$ *is a strict monoidal category and* $\rho\colon \otimes \Rightarrow \otimes \circ X\colon \mathbb{C}_0 \times \mathbb{C}_0 \to \mathbb{C}_0$ *is a natural transformation (where* X *is the functor that swaps its two arguments - see Example 2.4.4) such that* $\rho_{e,e} = id_e$*, and satisfying the following axioms:*

- $\rho_{a \otimes b, c} = (\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c})$ *for all* $a, b, c \in ob(\mathbb{C}_0)$
- $id_a \otimes id_b = \rho_{b,a} \circ \rho_{a,b}$ *for all* $a, b \in ob(\mathbb{C}_0)$

We are mostly interested in so called gs-monoidal categories [CG99, GH97], called s-monoidal in [Gad96]. These are symmetric monoidal categories, equipped with a discharger and a duplicator. Note that "gs" originally stood for "graph substitution" and such categories were first used for modelling term graph rewriting.

Definition 2.4.9 (gs-monoidal category)**.** *A strict* ***gs-monoidal category*** $\mathbb{C}$ *is a tuple* $(\mathbb{C}_0, \otimes, e, \rho, \nabla, !)$ *where* $(\mathbb{C}_0, \otimes, e, \rho)$ *is a strict symmetric monoidal category and* $!\colon Id_{\mathbb{C}_0} \Rightarrow e\colon \mathbb{C}_0 \to \mathbb{C}_0$ *(discharger),* $\nabla\colon Id_{\mathbb{C}_0} \Rightarrow \otimes \circ \Delta\colon \mathbb{C}_0 \to \mathbb{C}_0$ *(duplicator) are two tranformations (*Δ *is the diagonal functor, see Example 2.4.4), such that* $!_e = \nabla_e = e$ *and satisfying the following axioms:*

- Coherence Axioms:
  - $(id_a \otimes \nabla_a) \circ \nabla_a = (\nabla_a \otimes id_a) \circ \nabla_a$ *for all* $a \in ob(\mathbb{C}_0)$
  - $id_a = (id_a \otimes !_a) \circ \nabla_a$ *for all* $a \in ob(\mathbb{C}_0)$
  - $\nabla_a = \rho_{a,a} \circ \nabla_a$ *for all* $a \in ob(\mathbb{C}_0)$
- Monoidality Axioms:
  - $\nabla_a \otimes \nabla_b = (id_a \otimes \rho_{b,a} \otimes id_b) \circ \nabla_{a\otimes b}$ *for all* $a, b \in ob(\mathbb{C}_0)$
  - $!_a \otimes !_b = !_{a\otimes b} \circ id_e$ *for all* $a, b \in ob(\mathbb{C}_0)$

Gs-monoidal categories have been shown to be suitable for specifying term rewriting (see e.g. [CG99, GH97]). In essence gs-monoidal categories describe graph-like structures with dedicated input and output interfaces, operators for disjoint union (tensor), duplication and termination of wires, quotiented by the axioms satisfied by these operators. Particularly useful are gs-monoidal functors that preserve such operators and hence naturally describe compositional operations.

We will be able to derive two gs-monoidal categories and a gs-monoidal functor in Chapter 4 which forms the basis for the tool we describe in Section 4.7.

For a better overview we assemble all axioms fulfilled of a gs-monoidal category $\mathbb{C}$ below. All the non-obvious axioms are also depicted as string diagrams in Figure 2.3.

1. functoriality of tensor:
   - $(g \otimes g') \circ (f \otimes f') = (g \circ f) \otimes (g' \circ f')$
   - $id_{a\otimes b} = id_a \otimes id_b$
2. monoidality:
   - $(f \otimes g) \otimes h = f \otimes (g \otimes h)$
   - $f \otimes id_e = f = id_e \otimes f$
3. naturality:
   - $(f' \otimes f) \circ \rho_{a,a'} = \rho_{b,b'} \circ (f \otimes f')$

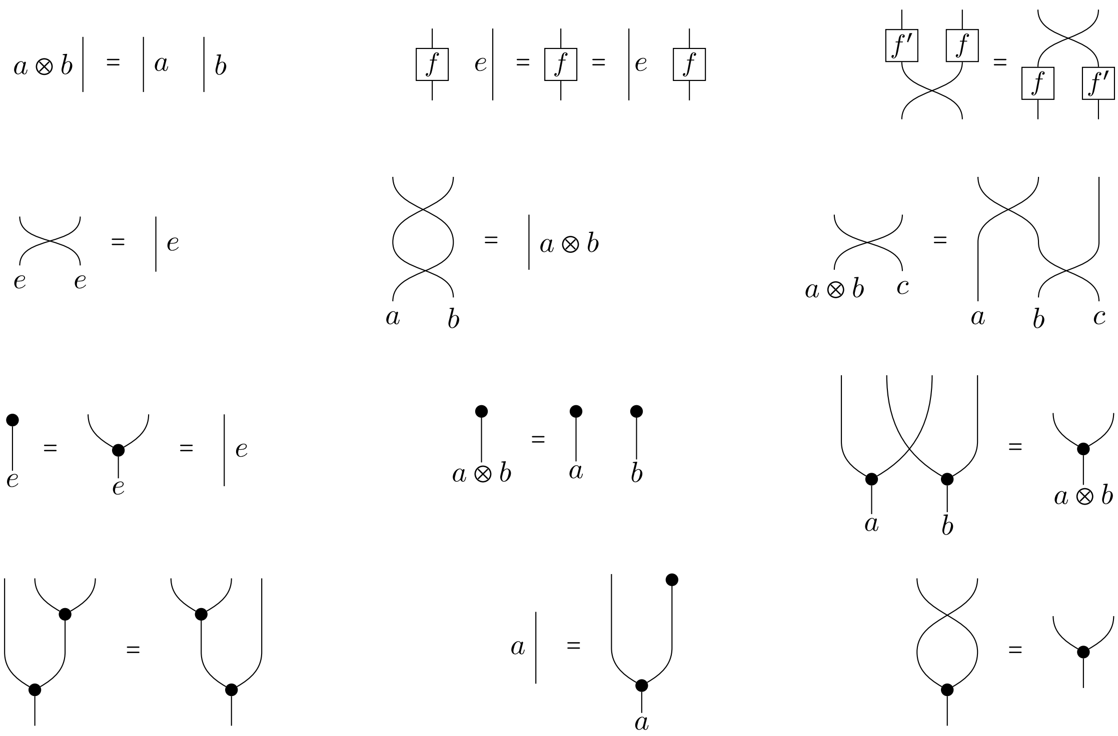


Fig. 2.3.: String diagrams of the axioms satisfied by GS-monoidal categories.

4. symmetry:
   - $\rho_{e,e} = id_e$
   - $\rho_{b,a} \circ \rho_{a,b} = id_{a\otimes b}$
   - $\rho_{a\otimes b,c} = (\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c})$
5. gs-monoidality:
   - $!_e = \nabla_e = id_e$
   - coherence axioms:
     - $(id_a \otimes \nabla_a) \circ \nabla_a = (\nabla_a \otimes id_a) \circ \nabla_a$
     - $id_a = (id_a \otimes !_a) \circ \nabla_a$
     - $\rho_{a,a} \circ \nabla_a = \nabla_a$
   - monoidality axioms:
     - $!_{a\otimes b} = !_a \otimes !_b$
     - $(id_a \otimes \rho_{a,b} \otimes id_b) \circ (\nabla_a \otimes \nabla_b) = \nabla_{a\otimes b}$
       (or, equivalently, $\nabla_a \otimes \nabla_b = (id_a \otimes \rho_{b,a} \otimes id_b) \circ \nabla_{a\otimes b}$)

Additionally, we can define a gs-monoidal functor between gs-monoidal categories.

Definition 2.4.10 (gs-monoidal functor)**.** *A functor $\#: \mathbb{C} \to \mathbb{D}$ is **gs-monoidal** if the following holds:*

1. *$\mathbb{C}$ and $\mathbb{D}$ are gs-monoidal categories*
2. *monoidality:*
   - $\#(e) = e'$
   - $\#(a \otimes b) = \#(a) \otimes' \#(b)$
3. *symmetry:*
   - $\#(\rho_{a,b}) = \rho'_{\#(a),\#(b)}$
4. *gs-monoidality:*
   - $\#(!_a) = !'_{\#(a)}$
   - $\#(\nabla_a) = \nabla'_{\#(a)}$

where the dashed operators are from the category $\mathbb{D}$, the others from $\mathbb{C}$.

2.5. Coalgebraic Behavioural Metrics

The presentation of this section is based on [BBKK18]. We start with a quick motivation.

2.5.1. Motivation

Consider the labeled Markov chain in Figure 2.4. An intuitive understanding of such a system is that in each state the system chooses a transition (indicated by the arrows) to another state using the probabilistic information which is given by the numbers on the arrows. Additionally we assign a label to each state (colors). We will formally introduce labeled Markov chains in Section 2.6.3.

Probabilistic bisimilation is a well known concept [TvB17] and two states of a system are bisimilar if their behaviour can not be differentiated by an outside observer of the system. States with different labels (colors) are deemed to behave differently, e.g. the behaviour of states u and z is different.

Assume we have $\varepsilon = 0$ in Figure 2.4, then any observer of the system could not differentiate the behaviour of states x and y. Whenever we are in either state, a coin is tossed to select the successor, either u or z. However, let $\varepsilon > 0$ be some very small number then the behaviour of states x and y can be observed to be different. However, since ε is very small, the behaviour of states x and y is very similar. This motivates a quantification of the behavioural difference of two states instead of just saying they behave the same or not.

As one can see, states x and y have the same label and thus their behaviour can just be differentiated by the probabilities with which they determine their successor. Therefore, to measure the behavioural distance in this example, we need to compare the probability distributions p_x and p_y on the set of states by which the successor is chosen (for x, respectively y). Immediately, it is not clear at all, how to measure the behavioural distance of two probability distributions. We approach the problem on a high-level: Given a pseudometric on the set of states, we lift this pseudometric to the set of pseudometrics on probability distributions.

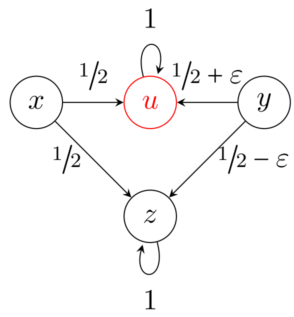


Fig. 2.4.: Example of a labeled Markov chain, $\varepsilon > 0$ is some small value

2.5.2. Predicate Liftings

Let $F\colon \mathbf{Set} \to \mathbf{Set}$ be an endofunctor on the category **Set** (see Example 2.4.4). An ($F-$)coalgebra is just a mapping $\alpha\colon X \to F(X)$ (written as FX). Note that FX might be an infinite set even if X is finite. Intuitively, α specifies a transition system whose branching type is given by F.

In general, we can lift a predicate $a\colon Y \to [0,1]$ to a predicate $a^F\colon FY \to [0,1]$ for some set Y [LPSS12]. Predicate liftings ([Pat03, Sch08b]) for arbitrary quantales have been studied, for instance, in [BKP18]. Since complete MV-algebras are quantales and we aim to show non-expansiveness of some liftings, we define predicate liftings over some complete MV-algebra $\mathbb{M}$.

Definition 2.5.1 (predicate lifting)**.** *Given a functor $F\colon \mathbf{Set} \to \mathbf{Set}$, a* ***predicate lifting*** *is a family of functions $\tilde{F}_Y\colon \mathbb{M}^Y \to \mathbb{M}^{FY}$ (where Y is a set), such that for $g\colon Z \to Y$, $a\colon Y \to \mathbb{M}$ it holds that $(Fg)^*(\tilde{F}_Y(a)) = \tilde{F}_Z(g^*(a))$.*

That is, predicate liftings must commute with reindexings (see Definition 3.5.3). The index Y will be omitted if clear from the context. Such predicate liftings are in one-to-one correspondence to so called evaluation maps $ev_F\colon F\mathbb{M} \to \mathbb{M}$.[4]

[4]This follows from the Yoneda lemma, see e.g. [Mac71].

Definition 2.5.2 (evaluation function/evaluation functor). *Let F be an endofunctor on* **Set**. *An* ***evaluation function*** ev_F *for F is a function*

$$ev_F: F\mathbb{M} \to \mathbb{M}.$$

Given ev_F, we define the corresponding lifting to be $\tilde{F}(a) = ev_F \circ Fa: FY \to \mathbb{M}$, where $a: Y \to \mathbb{M}$.

We will exclusively consider well-behaved liftings [BBKK18, BKP18].

Definition 2.5.3 (well-behaved). *A lifting $\tilde{F}: \mathbb{M}^Y \to \mathbb{M}^{FY}$ is* ***well-behaved*** *if*

- $\tilde{F}$ *is monotone*
- $\tilde{F}(0_Y) = 0_{FY}$ *where 0 is the constant 0-function*
- $\tilde{F}(a \oplus b) \sqsubseteq \tilde{F}(a) \oplus \tilde{F}(b)$ *for $a, b: Y \to \mathbb{M}$*
- *F preserves weak pullbacks (see [BBKK18])*

We aim to have not only monotone, but non-expansive liftings.

Lemma 2.5.4. *Let $ev: F\mathbb{M} \to \mathbb{M}$ be an evaluation map and assume that its corresponding lifting $\tilde{F}: \mathbb{M}^Y \to \mathbb{M}^{FY}$ is well-behaved. Then $\tilde{F}$ is non-expansive iff for all $\delta \in \mathbb{M}$ it holds that $\tilde{F}\delta_Y \sqsubseteq \delta_{FY}$, where δ is seen as the constant δ-predicate on Y, respectively FY.*

Proof. See Appendix: Lemma A.1.4. □

2.5.3. Lifting to Pseudometric Spaces

As it was the original goal, given a functor F: **Set** → **Set** and some pseudometric $d: X \times X \to \mathbb{M}(= [0,1])$ on X, we want to lift d to a pseudometric $d^F: FX \times FX \to \mathbb{M}$ on FX, i.e. we aim to find a functor $\bar{F}$: **PMet** → **PMet** with $\bar{F}(X, d) = (FX, d^F)$. It makes much more sense to work with pseudometrics instead of metrics as they allows different elements of X to have a distance of 0 whenever their behaviour is the same. We note that it is possible to lift any $\top$−pseudometric (see [BBKK18]).

Definition 2.5.5 (lifting to pseudometric spaces)**.** *Let U be the forgetful functor from Example 2.4.4. A functor $\bar{F}\colon \boldsymbol{PMet} \to \boldsymbol{PMet}$ is called a **lifting** of a functor $F\colon \mathbf{Set} \to \mathbf{Set}$ if the following diagram commutes.*

$$\begin{array}{ccc} \boldsymbol{PMet} & \xrightarrow{\bar{F}} & \boldsymbol{PMet} \\ {\scriptstyle U}\downarrow & & \downarrow{\scriptstyle U} \\ \mathbf{Set} & \xrightarrow[F]{} & \mathbf{Set} \end{array}$$

In this case, for any pseudometric space (X,d), we denote by d^F the pseudometric on FX which we obtain by applying $\bar{F}$ to (X,d).

Now, the next step is to construct such a lifting $\bar{F}$, in particular the distance function d^F. We will exclusively work with the Wasserstein lifting which has rather desirable properties.

2.5.4. The Wasserstein Lifting

We build on [BBKK18], where an approach is proposed for canonically defining a behavioural pseudometric for coalgebras of a functor $F\colon \mathbf{Set} \to \mathbf{Set}$, that is, for functions of the form $\xi\colon X \to FX$ where X is a set. Intuitively ξ specifies a transition system whose branching type is given by F. Given such a coalgebra ξ, the idea is to endow X with a pseudo-metric $d_\xi\colon X \times X \to \mathbb{M}$ - called the behavioural distance - defined as the least fixpoint of the map $d \mapsto d^F \circ (\xi \times \xi)$ where $\_^F$ lifts a metric $d\colon X \times X \to \mathbb{M}$ to a metric $d^F\colon FX \times FX \to \mathbb{M}$. Here we focus on the so-called Wasserstein lifting and show how approximations of the functions involved in the definition of the pseudometric can be determined.

To introduce the Wasserstein distance we require the definition of a coupling.

Definition 2.5.6 (coupling)**.** *Let $F\colon \mathbf{Set} \to \mathbf{Set}$ be a functor. Given a set X and $t_1, t_2 \in FX$, we call an element $t \in F(X^2)$ such that $F\pi_i(t) = t_i$, $i = 1, 2$, a **coupling** of t_1 and t_2 (with respect to F). We write $\Gamma_F(t_1, t_2)$ for the set of all these couplings.*

The main idea of the Wasserstein lifting is to couple two elements $t_1, t_2 \in FX$ in an 'optimal' way.

Definition 2.5.7 (Wasserstein distance)**.** *Let $F\colon \mathbf{Set} \to \mathbf{Set}$ be a functor with evaluation function ev_F. For every pseudometric space (X,d) the* ***Wasserstein distance*** *on FX is the function $d^{\downarrow F}\colon FX \times FX \to [0,\top]$ given by*

$$d^{\downarrow F}(t_1,t_2) \coloneqq \inf\{\tilde{F}d(t) \mid t \in \Gamma_F(t_1,t_2)\}$$

for all $t_1,t_2 \in FX$.

$\tilde{F}$ is a predicate lifting as in Section 2.5.2, i.e. $\tilde{F}(d) = ev_F \circ Fd\colon F(X\times X) \to \mathbb{M}$ where $d\colon X \times X \to \mathbb{M}$.

If $\tilde{F}$ is well-behaved, it holds that that $d^{\downarrow F}$ is a pseudometric on FX and the Wasserstein lifting $\bar{F}(X,d) = (FX, d^{\downarrow F})$ is a functor on **PMet** (see [BBKK18]).

In order to make the theories in later chapters effective we will need to restrict to a subclass of liftings.

Definition 2.5.8 (finitely coupled lifting)**.** *We call a lifting $\tilde{F}$* finitely coupled *if for all X and $t_1,t_2 \in FX$ there exists a $\Gamma'_F(t_1,t_2) \subseteq_{fin} \Gamma_F(t_1,t_2)$*[5]*, which can be computed given t_1,t_2, such that $\inf_{t\in\Gamma_F(t_1,t_2)} \tilde{F}d(t) = \min_{t\in\Gamma'_F(t_1,t_2)} \tilde{F}d(t)$.*

Observe that whenever the infimum above is a minimum, there is trivially a finite $\Gamma'(t_1,t_2)$. We however ask that there is an effective way to determine it. This last property holds for the two liftings we will consider next. These aim to lift pseudometrics to the powerset and the set of probability distributions, respectively.

Remark 2.5.9. *Given some coalgebra $\xi\colon X \to FX$, the behavioural distance of the Wasserstein lifting is given as the least fixpoint d_ξ of the map $d \mapsto d^{\downarrow F} \circ (\xi \times \xi)$. This map may omit multiple fixpoints. To fix this problem, one can introduce a discount factor $\lambda \in (0,1)$ and thus the map $d \mapsto \lambda \cdot (d^{\downarrow F} \circ (\xi\times\xi))$ is a contraction which omits one unique fixpoint d_ξ, i.e. for all $x,y \in X$, we have*

$$d_\xi(x,y) = \lambda \cdot d_\xi^{\downarrow F}(\xi(x),\xi(y)).$$

This is done sometimes in the literature (e.g. in [BBL⁺21, BBLM17]) to easy the computation of d_ξ. Throughout this thesis we will not be working with a discount factor.

2.5.5. The Hausdorff Lifting

Consider the functor $\mathcal{P}_f$. A coalgebra $\alpha\colon X \to \mathcal{P}_f(X)$ assigns a finite powerset $\alpha(x)$ to each $x \in X$. We now derive the Wasserstein Lifting.

$C \in \mathcal{P}_f(X\times X)$ is a coupling of $A,B \in \mathcal{P}_f(X)$, i.e. an element of $\Gamma_{\mathcal{P}_f}(A,B)$, if

- for all $a \in A$ there exists some $b \in B$ with $(a,b) \in C$

[5]It means $|\Gamma'_F(t_1,t_2)| < \infty$ and $\Gamma'_F(t_1,t_2) \subseteq \Gamma_F(t_1,t_2)$

- for all $b \in B$ there exists some $a \in A$ with $(a,b) \in C$
- $C \subseteq A \times B$

Note that, whenever $A = B = \varnothing$ we obtain the valid coupling $C = \varnothing$ and thus $\Gamma_{\mathcal{P}_f}(A,B) \neq \varnothing$ but whenever $A = \varnothing \neq B$ or $A \neq \varnothing = B$ then $\Gamma_{\mathcal{P}_f}(A,B) = \varnothing$.

Since A and B are finite, it is immediately clear, that C and $\Gamma_{\mathcal{P}_f}(A,B)$ are finite sets.

Example 2.5.10. *Let* $X = \{a,b,c\}$ *and* $A, B \in \mathcal{P}_f(X)$ *be given by*

$$A = \{a,b\}, \ \ B = \{b,c\}.$$

We have seven valid couplings $C_1, C_2, C_3, C_4, C_5, C_6, C_7 \in \mathcal{P}_f(X \times X)$ *given by*

$$\begin{aligned}
&C_1 = \{(a,c),(b,b)\}, \ \ C_2 = \{(a,b),(b,c)\},\\
&C_3 = \{(a,c),(b,b),(a,b)\}, \ \ C_4 = \{(a,c),(b,b),(b,c)\},\\
&C_5 = \{(a,c),(a,b),(b,c)\}, \ \ C_6 = \{(b,b),(a,b),(b,c)\},\\
&C_7 = \{(a,c),(b,b),(a,b),(b,c)\}
\end{aligned}$$

We define the evaluation map $ev_{\mathcal{P}_f}: \mathcal{P}_f\mathbb{M} \to \mathbb{M}$ for $U \in \mathcal{P}_f\mathbb{M}$ as

$$ev_{\mathcal{P}_f}(U) = \max_{u \in U} u, \ \text{where} \ \max_{u \in \varnothing} u = 0,$$

which is well-behaved [BBKK18] and the corresponding predicate lifting is non-expansive.

Lemma 2.5.11. *The predicate lifting* $\tilde{\mathcal{P}}_f: \mathcal{P}_f\mathbb{M} \to \mathbb{M}$ *is non-expansive.*

Proof. Given some set Y and $\delta \in \mathbb{M}$. We have $\tilde{\mathcal{P}}_f(\delta_Y) = \delta_{\mathcal{P}_f(Y)\setminus\{\varnothing\}} \sqsubseteq \delta_{\mathcal{P}_f(Y)}$. By Lemma 2.5.4, $\tilde{\mathcal{P}}_f$ is non-expansive. □

We obtain the following lifting - called the Hausdorff distance (in the dual sense):

Definition 2.5.12 (Hausdorff distance (dual))**.** *Given some coalgebra* $\alpha: X \to \mathcal{P}_f(X)$ *for a set* X *and the evaluation map* $ev_{\mathcal{P}_f}$ *as defined above. Given a pseudometric* d *on* X *and* $x, y \in X$*, we obtain the Wasserstein distance - named the* ***Hausdorff distance*** *(in the dual sense)*

$$\begin{aligned}
\mathcal{H}(d)(\alpha(x),\alpha(y)) &:= d^{\downarrow\mathcal{P}_f}(\alpha(x),\alpha(y))\\
&= \inf_{C \in \Gamma_{\mathcal{P}_f}(\alpha(x),\alpha(y))} \max_{(a,b)\in C} d(a,b)\\
&= \min_{C \in \Gamma_{\mathcal{P}_f}(\alpha(x),\alpha(y))} \max_{(a,b)\in C} d(a,b)
\end{aligned}$$

where $\min_{C\in\varnothing} \max_{(a,b)\in C} d(a,b) = 1$.

We can replace the inf with min since $\Gamma_{\mathcal{P}_f}(\alpha(x),\alpha(y))$ is a finite set. We call a coupling **optimal** if it attains the minimum.

Now, for any $A, B \in \mathcal{P}_f(X)$, one can construct the set $\Gamma_{\mathcal{P}_f}(A,B)$ or the set of minimal couplings $\Gamma'_{\mathcal{P}_f}(A,B)$.

Definition 2.5.13 (minimal couplings for $\mathcal{P}_f$)**.** *A coupling $C \in \Gamma_{\mathcal{P}_f}(A,B)$ is **minimal** if there exists no coupling $C' \in \Gamma_{\mathcal{P}_f}(A,B)$ with $C' \subset C$. We denote $\Gamma'_{\mathcal{P}_f}(A,B)$ as the set of minimal couplings of A, B.*

It is rather immediate that

$$\min_{C\in\Gamma_{\mathcal{P}_f}(A,B))}\max_{(a,b)\in C} d(a,b) = \min_{C\in\Gamma'_{\mathcal{P}_f}(A,B)}\max_{(a,b)\in C} d(a,b)$$

since for any coupling $C \notin \Gamma'_{\mathcal{P}_f}(A,B)$ it holds that there exists a coupling $C' \in \Gamma'_{\mathcal{P}_f}(A,B)$ with $C \subset C'$ and thus

$$\max_{(a,b)\in C} d(a,b) \geq \max_{(a,b)\in C'} d(a,b).$$

Therefore the minimum is attained at some $C \in \Gamma'_{\mathcal{P}_f}(A,B)$

There is a constructive way to obtain $\Gamma'_{\mathcal{P}_f}(A,B)$. Without loss of generality, let $A = \{a_1,\dots,a_n\}$ and $B = \{b_1,\dots b_m\}$ with $n \geq m$. Then any map $\rho\colon A \to B$ with $\bigcup_{a\in A}\rho(a) = B$ specifies a minimal coupling, i.e. $C_\rho = \{(a_1,\rho(a_1)),\dots(a_n,\rho(a_n))\}$. Now, $\Gamma'_{\mathcal{P}_f}(A,B) = \{C_\rho \mid \rho\colon A \to B : \bigcup_{a\in A}\rho(a) = B\}$. From this, it is clear that $|C| = n(=\max\{n,m\})$ for any minimal coupling.

Example 2.5.14. *We revisit Example 2.5.10. One can see that C_1 and C_2 are the only minimal couplings. Given the pseudometric d on X, given by $d(a,b) = 1$ and $d(a,c) = d(b,c) = 0$ (all other distances are given since d is a pseudometric) then we obtain*

$$\begin{aligned}\mathcal{H}(d)(A,B) &= \min_{i=1,\dots,7}\max_{(x,y)\in C_i} d(x,y) = \min\{0,1,1,0,1,1,1\} = 0\\ &= \min_{i=1,2}\max_{(x,y)\in C_i} d(x,y) = \min\{0,1\} = 0\end{aligned}$$

implying that C_1 and C_4 are optimal couplings and C_1 is the only optimal and minimal coupling.

The Hausdorff distance we derived is usually referred to as the dual characterization. The function $\mathcal{H}$ has an equal primal characterization due to Mémoli [Mém11], also observed in [BBKK18].

Definition 2.5.15 (primal characterization of $\mathcal{H}$)**.** *Given $A, B \in \mathcal{P}_f(X)$ and some pseudometric d on X. The* ***primal characterization of the Hausdorff distance*** *is given by*

$$\mathcal{H}(d)(A,B) = \max\{\max_{a\in A}\min_{b\in B} d(a,b), \max_{b\in B}\min_{a\in A} d(a,b)\}.$$

To clarify, it holds

$$\mathcal{H}(d)(A,B) = \max\{\max_{a\in A}\min_{b\in B} d(a,b), \max_{b\in B}\min_{a\in A} d(a,b)\} = \min_{C\in\Gamma_{\mathcal{P}_f}(A,B)} \max_{(a,b)\in C} d(a,b)$$

for any $A, B \in \mathcal{P}_f(X)$ and pseudometric d on X.

2.5.6. The Kantorovich Lifting

We note up front that the here defined Kantorovich lifting refers to the instantiation of the Wasserstein lifting for the functor $\mathcal{D}_f$. The name is sometimes used in a different setting (cf. [BBKK18]).

Consider the functor $\mathcal{D}_f$. A coalgebra $\alpha\colon X \to \mathcal{D}_f(X)$ assigns a probability distribution $\alpha(x)$ with finite support to each $x \in X$. We now derive the Wasserstein lifting.

A probability distribution $\omega \in \mathcal{D}_f(X \times X)$ is a coupling of $p, q \in \mathcal{D}_f(X)$, i.e. an element of $\Gamma_{\mathcal{D}_f}(p,q)$, if for all $x \in X$:

$$p(x) = \sum_{y\in X} \omega(x,y) \text{ and } q(x) = \sum_{y\in X} \omega(y,x).$$

Example 2.5.16. *Let $X = \{x, y\}$ and $p, q \in \mathcal{D}_f(X)$ be given by*

$$p(x) = p(y) = 0.5 \text{ and } q(x) = 0.25,\ q(y) = 0.75.$$

A valid coupling $\omega \in \mathcal{D}_f(X \times X)$ is the following:

$$\omega(x,x) = 0.25,\ \omega(x,y) = 0.25,\ \omega(y,y) = 0.5.$$

As one can see, there usually are infinitely many couplings as is the case in the above example, i.e. all convex combinations of ω and ω' ($\omega'(x,y) = 0.5$, $\omega(y,x) = 0.25$, $\omega'(y,y) = 0.25$). However, given two probability distributions $p, q \in \mathcal{D}_f(X)$ the set of couplings $\Gamma_{\mathcal{D}_f}(p,q)$ forms a convex polytope $\Omega(p,q)$ ([TvB17, PC20]). We denote the finite set of vertices of $\Gamma_{\mathcal{D}_f}(p,q)$ as $\Omega_V(p,q)$ (cf. Section 2.2).

We define the evaluation map $ev_{\mathcal{D}_f}\colon \mathcal{D}_f[0,1] \to [0,1]$ for $u \in \mathcal{D}_f[0,1]$ as

$$ev_{\mathcal{D}_f}(u) = \sum_{x\in[0,1]} x \cdot u(x) \qquad \text{(expectation)}$$

which is well-behaved [BBKK18] and the corresponding predicate lifting is non-expansive.

Lemma 2.5.17. *The predicate lifting* $\tilde{\mathcal{D}}_f \colon \mathcal{D}_f[0.1] \to [0,1]$ *is non-expansive.*

Proof. Given some set Y and $\delta \in \mathbb{M}$. We immediately have $\tilde{\mathcal{D}}_f(\delta_Y) = \delta_{\mathcal{D}_f(Y)}$. By Lemma 2.5.4, $\tilde{\mathcal{D}}_f$ is non-expansive. □

We obtain the following lifting:

Definition 2.5.18 (Kantorovich distance)**.** *Given some coalgebra* $\alpha \colon X \to \mathcal{D}_f(X)$ *for a set* X *and the evaluation map* $ev_{\mathcal{D}_f}$ *as defined above. Given a pseudometric* d *on* X *and* $x, y \in X$*, we obtain the Wasserstein distance - named the* ***Kantorovich distance***

$$\begin{aligned}\mathcal{K}(d)(\alpha(x),\alpha(y)) &\coloneqq d^{\downarrow\mathcal{D}_f}(\alpha(x),\alpha(y))\\ &= \inf_{\omega\in\Gamma_{\mathcal{D}_f}(\alpha(x),\alpha(y))}\sum_{x,y\in X}\omega(x,y)\cdot d(x,y)\\ &= \min_{\omega\in\Omega_V(\alpha(x),\alpha(y))}\sum_{x,y\in X}\omega(x,y)\cdot d(x,y).\end{aligned}$$

We note that the infimum is attained at some vertex of $\Gamma_{\mathcal{D}_f}(\alpha(x),\alpha(y))$ ([TvB17]) allowing us to replace the inf with min since any polytope only has finitely many vertices. A coupling that attains the minimum is called **optimal**.

There exists a dual characterization of the Kantorovich distance [BBKK18] which does not prove useful to us.

Example 2.5.19. *We revisit the previous example. We have* $\Omega_V(p,q) = \{\omega_1, \omega_2\}$ *where*

$$\omega_1(x,x) = 0.25,\ \omega_1(x,y) = 0.25,\ \omega_1(y,y) = 0.5$$

and

$$\omega_2(x,y) = 0.5,\ \omega_2(y,x) = 0.25,\ \omega_2(y,y) = 0.25.$$

Given the pseudometric d *on* X*, given by* $d(x,y) = 1$ *(all other distances are given since* d *is a pseudometric), we obtain*

$$\mathcal{K}(d)(p,q) = \min_{i=1,2}\sum_{x,y\in X}\omega_i(x,y)\cdot d(x,y) = \min\{0.25, 0.5\} = 0.25$$

implying that ω_1 *is an optimal coupling.*

In general, given two probability distributions $p, q \in \mathcal{D}_f(X)$ and a pseudometric d on X, the Kantorovich distance can be obtained as the solution of the following linear program:

$$\mathcal{K}(d)(p,q) = \min\sum_{x,y\in X} d(x,y)\cdot\omega_{x,y}$$

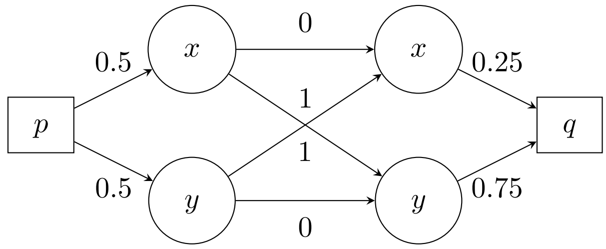


Fig. 2.5.: Transportation plan

$$\begin{aligned} \sum_{y \in X} \omega_{x,y} &= p(x) && \forall x \in X \\ \sum_{x \in X} \omega_{x,y} &= q(y) && \forall y \in X \\ \omega_{x,y} &\geq 0 && \forall x, y \in X \end{aligned}$$

There exists some ω^* with $\mathcal{K}(d)(p,q) = \sum_{x,y \in X} d(x,y) \cdot \omega^*_{x,y}$ and $\omega^* \in \Omega_V(p,q)$. Thus ω^* is an optimal coupling and a vertex of $\Gamma_{\mathcal{D}_f}(p,q)$. Since $\Gamma_{\mathcal{D}_f}(p,q)$ is a convex polytope, it is not surprising that linear programming can be used to compute the Kantorovich distance.

One can construct $\Omega_V(p,q) = \Gamma'_{\mathcal{D}_f}(p,q)$ as the set of basic solutions (vertices, see Definition 2.2.2) of the polyhedron $P(A,b)$ resulting from the linear program above. Specifying A and b is too colloquial so we spare doing it.

From a practical point of view, to compute $\mathcal{K}(d)(p,q)$ (from Example 2.5.19) one has to solve the transportation plan in Figure 2.5 [Vil09, PC20]: We have supply 0.5 for both states x and y on the left (given by p) and need to transport the supply to fulfill the demand of 0.25 for state x and 0.75 for state y on the right (given by q). The transport of each unit from supplier to demander yields some cost. We aim to nurse supply and demand with the cheapest transport.

As is clear in our example, one should avoid transporting goods from x to y and vice versa. Thus 0.25 from supplier x should be transported to consumer x and 0.5 units from supplier y should be transported to consumer y. The remaining 0.25 supply of x need to be transported to consumer y with a cost of 1 per unit and distance, resulting in a total cost of $0.25 \cdot 0 + 0.5 \cdot 0 + 0.25 \cdot 1 = 0.25$ for transporting the goods. One can see that we had $\mathcal{K}(d)(p,q) = 0.25$.

2.6. Application: State-based Systems

The theories we will introduce in later chapters can be applied to several of problems found in the literature. State-based systems yield easy examples to this end. We will introduce a handful of different systems and discuss the problems arising from them. In particular, the solution to these problems can be found by computing an extreme (least or greatest) fixpoint of some underlying function. Thus, we will derive the approximations to these functions in Chapter 3 and apply strategy iteration to compute the solutions in Chapter 5.

In particular we aim to compute the termination probability of Markov chains (Section 2.6.1), the bisimilarity for transition systems (Section 2.6.2), behavioural distances for labeled Markov chains (Section 2.6.3), metric transition systems (Section 2.6.4) and probabilistic automata (Section 2.6.5). We note up front that all systems are some sort of probabilistic automata. For sake of simplicity and to ease the understanding we will discuss them all on their own.

2.6.1. Termination Probability of Markov Chains

Markov chains, named after their 'inventor' Andrey Andreyevich Markov, have been studied for over 100 years [Beh14]. These systems are very common and frequently used in practice to simulate probabilistic processes. They are rather simple in nature; we have states which transition according to some probability distribution. We add a subset of terminal states to the definition.

Definition 2.6.1 (Markov chain)**.** *A* ***Markov chain*** *is a tuple* $\mathsf{MC} = (S, T, \eta)$ *consisting of a finite set of states* S*, a subset of terminal states* $T \subseteq S$ *and a transition function* $\eta\colon S \smallsetminus T \to \mathcal{D}(S)$*.*

Assume we are given some Markov chain $\mathsf{MC} = (S, T, \eta)$. Intuitively, $\eta(s)(s')$ denotes the probability of transitioning from state $s \in S \smallsetminus T$ to state $s' \in S$. Given some state $s \in S$, we want to determine the termination probability of s, i.e. the probability of eventually reaching any terminal state from s.

As a concrete example, take the Markov chain given in Figure 2.6. We have $S = \{x, y, z, u\}$ and $T = \{u\}$, i.e. u is the only terminal state. Additionally, the transitions $\eta(x), \eta(y), \eta(z) \in \mathcal{D}(S)$ are given as follows (omitting elements outside the supports):

$$\eta(x)(x) = \eta(x)(y) = \eta(x)(u) = 1/3 \text{ and } \eta(y)(z) = \eta(z)(y) = 1.$$

The termination probability of state u is 1 (as it is a terminal state) and the termination probability of states y and z is 0 as they will just run in a cycle, thus never reaching the only terminal state u. Now, when we are in state x, we will eventually leave state x and either move to state y (thus never reaching a terminal state) or to the terminal state u. Since $\eta(x)(y) = \eta(x)(u)$ the termination probability of state x is $1/2$.

The termination probability of a Markov chain arises as the least fixpoint of the following operator [BK08].

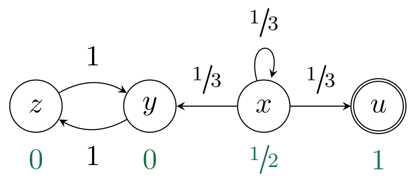


Fig. 2.6.: Example of a Markov chain, termination probabilities are given in green

Definition 2.6.2 (fixpoint operator - termination probability for Markov chains)**.** *Given some Markov chain* $\mathsf{MC} = (S, T, \eta)$*, we define the function* $\mathcal{T}\colon [0,1]^S \to [0,1]^S$ *for* $t \in [0,1]^S$ *and* $s \in S$ *as*

$$\mathcal{T}(t)(s) = \begin{cases} \sum_{s' \in S} \eta(s)(s') \cdot t(s') & \textit{if } s \in S \setminus T \\ 1 & \textit{otherwise} \end{cases}$$

It is easy to see that $\mu\mathcal{T}(y) = \mu\mathcal{T}(z) = 0$, $\mu\mathcal{T}(u) = 1$ and $\mu\mathcal{T}(x) = 1/2$ for the Markov chain in Figure 2.6. Note however, that $\mathcal{T}$ usually omits more than one fixpoint, e.g. $\nu\mathcal{T}(x) = \nu\mathcal{T}(y) = \nu\mathcal{T}(z) = \nu\mathcal{T}(u) = 1$ for the system in Figure 2.6 (in fact $t \equiv 1$ is always a fixpoint for any Markov chain).

Computation of $\mu\mathcal{T}$**.** Since $[0,1]^S$ is a complete lattice (see Example 2.3.4) we can use Kleene iteration (from below) to obtain an approximation of $\mu\mathcal{T}$. Kleene iteration usually does not yield an exact computation. An exact computation can be obtained by solving the following linear program

$$\begin{aligned} \min \sum_{s \in S} t_s & \\ t_s = 1 & \qquad \forall s \in T \\ t_s = \sum_{s' \in S} \eta(s)(s') \cdot t_{s'} & \qquad \forall s \in S \setminus T \end{aligned}$$

It is well-known that the function $\mathcal{T}$ can be tweaked in such a way that it has a unique fixpoint, coinciding with $\mu\mathcal{T}$, by determining all states which can not reach any terminal state and setting their value to zero [BK08]. We denote this set of states by NT which can easily be computed: To check if $s \in S$ lies in NT, one can simply compute the set of reachable states from s. If no terminal state is reachable it holds that $s \in NT$. By computing NT beforehand, $\mu\mathcal{T}$ can be obtained by solving the following linear system of equations

$$\begin{aligned} t_s = 1 & \qquad \forall s \in T \\ t_s = 0 & \qquad \forall s \in NT \end{aligned}$$

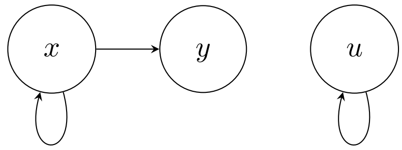


Fig. 2.7.: Example of a transition system

$$t_s = \sum_{s' \in S} \eta(s)(s') \cdot t_{s'} \qquad\qquad \forall s \in S \setminus (T \cup NT)$$

Kleene iteration from above can also be applied to compute the unique fixpoint we obtain when tweaking $\mathcal{T}$ such that the values of states in NT are set to 0.

Termination probability is a special case of the considerably more complex simple stochastic games that will be introduced in Section 2.7.3, where the trick of modifying the function is not applicable.

Looking Ahead. Termination probability will serve as the running example for the theory we will introduce in Chapter 3. In Section 3.6.1 we formally derive the approximation of $\mathcal{T}$ which allows us to detect whether some fixpoint of $\mathcal{T}$ is the least fixpoint or not. Additionally, this will allow us to derive lower bounds for $\mu\mathcal{T}$, which is useful in proving that some state terminates with at least some given probability.

Strategy iteration (Chapter 5) is not applicable to compute $\mu\mathcal{T}$ - at least in a sensible way.

2.6.2. Bisimilarity for Transition System

We now consider transition systems in their most basic form. Here, we only have states and directed edges connecting these states.

Definition 2.6.3 (transition system)**.** *A **transition system** is a tuple $\mathsf{TS} = (S, \eta)$ consisting of a finite set of states S and a transition function $\eta\colon S \to \mathcal{P}(S)$. We write $s \to s'$ for $s' \in \eta(s)$.*

The definition above can be expanded; we could assign labels to edges and/or states. These systems could still be analyzed by the later theories but for simplicity we stick with the definition above.

In Figure 2.7 we are given a very easy example of a transition system. We have $S = \{x, y, u\}$ and the transition function $\eta(x) = \{x, y\}$, $\eta(y) = \emptyset$ and $\eta(u) = \{u\}$.

Given some transistion system (S, η), we are interested in finding out whether two states $x, y \in S$ have the same observable behaviour, i.e. an outside observer can not distinguish two states by their behaviour. For the transition system in Figure 2.7, it is imminent that states x and y behave differently since x has a successor and y does not. The same holds for states y and u. Now, state x can transition to state y which has no successors whereas u can never reach any state without successor. Thus their behaviour differs from the viewpoint of an outside observer of the system. This motivates the definition of a bisimulation.

Definition 2.6.4 (bisimulation (transition system))**.** *Let* $\mathit{TS} = (S, \eta)$ *be a transition system. A relation* $R \subseteq S \times S$ *is a* ***bisimulation*** *if the following holds for all* $s, t \in S$ *with* sRt*:*

- *if* $s \to s'$ *then there exists some* $t \to t'$ *with* $s'Rt'$
- *if* $t \to t'$ *then there exists some* $s \to s'$ *with* $s'Rt'$

Two states $s, t \in S$ are bisimilar, written $s \sim t$ if they are related by some bisimulation. The relation $\sim$ is the union of all bisimulations - called bisimilarity. It holds that $\sim$ is a bisimuation itself and an equivalence relation (i.e. reflexive, symmetric and transitive). Bisimilar states have the same observable behaviour. In Figure 2.7 we have $\sim = \{(x,x),(y,y),(u,u)\}$.

Intuitively, two states are bisimilar if each transition from either state can be mirrored by the other states, i.e. it has a transition to a bisimilar state.

The bisimulation $\sim$ of any transition system can be computed as the greatest fixpoint of the following operator.

Definition 2.6.5 (fixpoint operator - bisimilarity for transition systems)**.** *Given a transition system* $\mathit{TS} = (S, \eta)$*. We define the function* $\mathcal{B}\colon \{0,1\}^{S\times S} \to \{0,1\}^{S\times S}$ *for* $a \in \{0,1\}^{S\times S}$ *and* $s, t \in S$ *as*

$$\mathcal{B}(a)(s,t) = \min\{\min_{s'\in\eta(s)} \max_{t'\in\eta(t)} a(s',t'), \min_{t'\in\eta(t)} \max_{s'\in\eta(s)} a(s',t')\}.$$

with $\min\emptyset = 1$ *and* $\max\emptyset = 0$*.*

It holds that $\nu\mathcal{B}(s,t) = 1$ if and only if $s \sim t$[6]. Again, this fixpoint is not unique, i.e. $\mu\mathcal{B} \equiv 0$ for the transition system in Figure 2.7. One can exactly compute $\nu\mathcal{B}$ via Kleene iteration (from above) since we only assign values in $\{0,1\}$ to states.

We quickly remark that $\mathcal{B}$ is very similar to the Hausdorff lifting $\mathcal{H}$ only that all min and max are swapped. This similarity will prove useful later on.

Computation of $\mu\mathcal{B}$. Bisimilarity for a transition system is usually obtained via partitioning refinement [SR11]. Here, we start with the relation $\sim_0 \subseteq S \times S$ containing all pairs of states and then iteratively compute $\sim_{k+1} \subseteq S \times S$ as follows: Let $s, t \in S$ then

[6] $\mathcal{B}(a)(s,t) = 1$ iff $\forall s \to s'\ \exists t \to t' : a(s',t') = 1$ and vice versa.

$s \sim_{k+1} t$ iff for $s \to s'$ there exists $t \to t'$ with $s' \sim_k t'$ and vice versa. Whenever $\sim_k = \sim_{k+1}$ for some $k \in \mathbb{N}$, we have $s \sim t$ iff $s \sim_k t$.

Kannelakis and Smolka describe a more efficient way to split a partitioning ([SR11]) which Page and Tarjan improve on [PTB85, SR11].

The Ehrenfeucht-Fraisse game is a two-player game (players are called attacker and defender) played on a transition system and two states are bisimilar iff there exists a winning strategy for the defender (see [Bos92] for details).

Also, two states are bisimilar iff they satisfy the same modal formulas [GO07].

Looking Ahead. $\mathcal{B}$ has a dual representation that we will detail in Section 3.6.7. For this dual representation we will derive the approximation of $\mathcal{B}$ which allows us to detect whether some fixpoint of $\mathcal{B}$ is the greatest fixpoint or not. Additionally, we can find upper bounds for $\nu\mathcal{B}$ which witnesses that two states are bisimilar without computing the whole bisimulation relation.

Strategy iteration is applicable to compute $\nu\mathcal{B}$ and we will detail this procedure in Section 5.3.6.

2.6.3. Behavioural Distances for Labeled Markov Chains

Labeled Markov chains are very similar to Markov chains. Instead of having terminating states we assign a label to each state. This is the most basic definition of a labeled Markov chain, e.g. see [BBLM17] where they additionally have terminating states and exit probabilities.

Definition 2.6.6 (labeled Markov chain)**.** *A* ***labeled Markov chain*** *is a tuple* $\mathsf{LMC} = (S, \eta, L, \ell)$ *consisting of a finite set of states* S*, a transition function* $\eta\colon S \to \mathcal{D}(S)$*, a set of labels* L *and a labeling function* $\ell\colon S \to L$*.*

Any labeled Markov chain can be seen as a Markov chain (with $T = \varnothing$) over some metric space (L, d_L). Throughout this thesis, we will simply have some finite set of labels L and $d_L\colon L \times L \to \{0,1\}$ will be the discrete metric assigning a distance of one to any two differing elements.

Consider the labeled Markov chain in Figure 2.8. Here, $S = \{x,y,z,u\}$ and $L = \{\mathit{black}, \mathit{red}\}$, where we have the discrete metric $d_L(\mathit{black}, \mathit{red}) = 1$ on L. Now, $\ell(x) = \ell(y) = \ell(z) = \mathit{black}$, $\ell(u) = \mathit{red}$; $\eta(u)(u) = \eta(z)(z) = 1$, $\eta(x)(u) = \eta(x)(z) = 1/2$ and $\eta(y)(u) = 1/2 + \varepsilon$, $\eta(y)(z) = 1/2 - \varepsilon$ (omitting elements outside the supports).

For labeled Markov chains we are interested in investigating the difference in behaviour of its states (as motivated in Section 2.5.1). Now, as one can see, states z and u behave differently since their labels differ. On the other hand, the behaviour of states x and y is very similar, only differing by some small value ε. However, their behaviour is observably different. As motivated in Section 2.5.1 we would like to measure this difference in behaviour. In particular, we want states having the same observable behaviour to have a

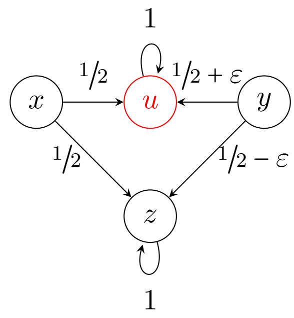


Fig. 2.8.: Example of a labeled Markov chain: labels are denoted by colours, $\varepsilon > 0$ is some small value

behavioural distance of 0. Thus, similar to transition systems, we define a probabilistic bisimulation.

Definition 2.6.7 (probabilistic bisimulation (labeled Markov chain))**.** *Let $\mathsf{LMC} = (S, \eta, L, \ell)$ be a labeled Markov chain. An equivalence relation $R \subseteq S \times S$ is a **proabilistic bisimulation** if the following holds for all $s, t \in S$ with sRt:*

- $\ell(s) = \ell(t)$
- *for all equivalence classes $C \in S/R$, we have $\eta(s)(C) = \eta(t)(C)$*

where $\eta(s)(C) = \sum_{c \in C} \eta(s)(c)$.

Two states $s, t \in S$ are bisimilar, written $s \sim t$ if they are related by some probabilistic bisimulation. The relation $\sim$ is the union of all bisimulations - called bisimilarity. It holds that $\sim$ is a bisimulation itself and an equivalence relation. Bisimilar states have the same observable behaviour.

To put the above notion into easy simple words, two states s and t are bisimilar if they have the same label and they transition to any set (equivalence class) of bisimilar states with the same probability. A probabilistic bisimulation can be computed via partitioning refinement [GVdV18].

The following operator computes the behavioural distance between all states in a labeled Markov chain.

Definition 2.6.8 (fixpoint operator - behavioural distances for labeled Markov chains). *Given a* $\mathsf{LMC} = (S, \eta, L, \ell)$. *We define the function* $\Delta\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ *for* $d \in [0,1]^{S\times S}$ *and* $s, t \in S$ *as*

$$\Delta(d)(s,t) = \max\{d_L(\ell(s), \ell(t)), \mathcal{K}(d)(\eta(s), \eta(t))\}$$

which simplifies to

$$\Delta(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{K}(d)(\eta(s), \eta(t)) & \text{otherwise} \end{cases}$$

whenever d_L *is the discrete metric.*

The least fixpoint of Δ yields the behavioural distance for a labeled Markov chain. It holds $s \sim t$ iff $\mu\Delta(s,t) = 0$ [TvB17]. For example, in Figure 2.8 we obtain the pseudometric $\mu\Delta(x,u) = \mu\Delta(y,u) = \mu\Delta(z,u) = 1$, $\mu\Delta(x,y) = \varepsilon$, $\mu\Delta(x,z) = 1/2$ and $\mu\Delta(y,z) = 1/2 - \varepsilon$ (all other distances are given since $\mu\Delta$ is a pseudometric).

Remark 2.6.9. *As it is the case for the other state based systems, we can directly derive the behavioural distance via the Wasserstein lifting from Section 2.5.4 by setting* $\mathbb{M} = [0,1]$ *and using the functor* $FX = \Lambda \times \mathcal{D}_f(X)$*, where* Λ *is a fixed set of labels*[7]*. We observe that couplings of* $(a_1, p_1), (a_2, p_2) \in FX$ *only exist if* $a_1 = a_2$ *and – if they do not exist – the Wasserstein distance is the empty infimum, hence* 1*. If* $a_1 = a_2$*, couplings correspond to the usual probabilistic couplings of* p_1, p_2 *from Section 2.5.6 and the least fixpoint of* $\mathcal{W}(d) = (\xi \times \xi) \circ d^{\downarrow F}$[8] *equals the behavioural metric, as explained in Section 2.5.4, and the least fixpoint of* Δ.

Computation of $\mu\Delta$. Once again, one can approximate $\mu\Delta$ via Kleene iteration (from below). We described how to compute the Kantorovich distance in Section 2.5.6.

The literature gives a handful of interesting algorithms which compute $\mu\Delta$. The paper [BBLM17] gives a linear program whose solution yields $\mu\Delta$ directly. They also present an on-the-fly algorithm which is similar to the strategy iterations we will describe in Chapter 5 but adds some problem specific optimizations. The paper [TvB17] adds on to this and derives a (partial) policy (strategy) iteration. They also handle the problem that Δ usually does not have one unique fixpoint. They show that the function $\Lambda\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ defined for $d\colon S \times S \to [0,1]$ and $s, t \in S$ as

$$\Lambda(d)(s,t) = \begin{cases} 0 & \text{if } s \sim t \\ \Delta(d)(s,t) & \text{otherwise} \end{cases}$$

has a unique fixpoint which equals $\mu\Delta$. Another approach which remedies this problem

[7] Liftings of multifunctors were analyzed in [BBKK18].

[8] ξ specifies the labeled Markov chain, i.e. $\xi(x) = (a, p)$.

is to introduce some discount factor $\lambda \in (0,1)$ and modify Δ as follows:

$$\Delta_\lambda(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \lambda \cdot \mathcal{K}(d)(\eta(s),\eta(t)) & \text{otherwise} \end{cases}$$

This is sometimes done in the literature [BBLM17]. As a contraction, Δ_λ has a unique fixpoint. To the two functions above with a unique fixpoint Kleene iteration from above is applicable as well.

Looking Ahead. In Section 2.6.5 we will discuss that any labeled Markov chain can be seen as an instance of a probabilistic automata.

In Section 3.6.3 we derive the approximation of Δ.

In Section 5.3.4 we describe how strategy iteration can be applied to compute $\mu\Delta$. We will also show that the policy and paritial policy iterations in [TvB17] can be seen as instances of our strategy iteration.

2.6.4. Behavioural Distances for Metric Transition Systems

Metric transition systems are transition systems (see Section 2.6.2) where we additionally assign some weight to each state. These were first introduced and studied in [dAFS09].

Definition 2.6.10 (metric transition system)**.** *A* ***metric transition system*** *is a tuple* $\mathit{MTS} = (S,\eta,L,\ell)$ *consisting of a finite set of states* S*, a transition function* $\eta\colon S \to \mathcal{P}(S)$ *and a labeling function* $\ell\colon S \to L$ *for some set of labels* L*.*

Metric transition systems are transition systems over some metric space (L,d_L). We will have $L = [0,1]$ and the metric $d_L\colon L \times L \to [0,1]$ is given as $d_L(x,y) = |x-y|$ for $x,y \in L$ (Euclidian metric). Thus, $d_L(\ell(s),\ell(t)) = |\ell(s) - \ell(t)|$ for states $s,t \in S$.

Bisimulation is defined very similar to transition systems without weights. We just need to guarantee that bisimilar states have the same label/weight.

Definition 2.6.11 (bisimulation (metric transition system))**.** *Let* $\mathit{MTS} = (S,\eta,L,\ell)$ *be a metric transition system. An equivalence relation* $R \subseteq S \times S$ *is a* ***bisimulation*** *if the following holds for all* $s,t \in S$ *with* sRt*:*

- $\ell(s) = \ell(t)$ *(iff* $d_L(s,t) = 0$*)*
- *if* $s \to s'$ *then there exists some* $t \to t'$ *with* $s'Rt'$
- *if* $t \to t'$ *then there exists some* $s \to s'$ *with* $s'Rt'$

Two states $s,t \in S$ are bisimilar, written $s \sim t$ if they are related by some bisimulation. The relation $\sim$ is the union of all bisimulations. $\sim$ is a bisimulation itself and an equivalence relation. Bisimilar states have the same observable behaviour.

The introduction of weights allows for a quantitative reasoning regarding the observable behaviour of states. The following operator computes the behavioural distance between states in a given metric transition system.

Definition 2.6.12 (fixpoint operator - behavioural distances for metric transition systems)**.** *Given a metric transition system* $\mathsf{MTS} = (S, \eta, L, \ell)$*, we define the function* $\mathcal{J}\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ *for* $d \in [0,1]^{S\times S}$ *and* $s, t \in S$ *as*

$$\mathcal{J}(d)(s,t) = \max\{d_L(\ell(s), \ell(t)), \mathcal{H}(d)(\eta(s), \eta(t))\}$$

which simplifies to

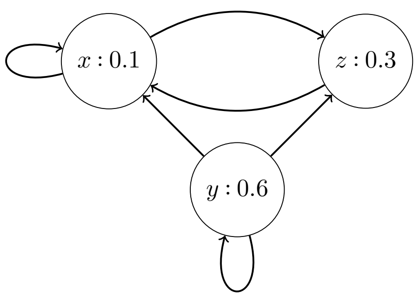
$$\mathcal{J}(d)(s,t) = \max\{|\ell(s) - \ell(t)|, \mathcal{H}(d)(\eta(s), \eta(t))\}$$

when d_L *is the euclidian metric.*

The least fixpoint of $\mathcal{J}$ yields the behavioural distance function for a metric transition system. We have $s \sim t$ iff $\mu\mathcal{J}(s,t) = 0$ ([BBL+21], as metric transition systems are special probabilistic automata). Again, $\mathcal{J}$ usually does not have a unique fixpoint.

Computation of $\mu\mathcal{J}$ can be done via Kleene iteration (from below) which in fact yields an exact computation after a finite number of iterations since the number of values pairs of states can take is bounded by the cardinality of $\{d_L(\ell(s), \ell(t)) \mid s, t \in S\}$, which is finite. We discussed the computation of the Hausdorff-distance $\mathcal{H}(d)$ in Section 2.5.5. The literature is rather scarce when it comes to computing behavioural distances for metric transition systems directly. We will see that metric transition systems are a form of probabilistic automata which are studied more frequently.

Example 2.6.13. *We consider the metric transition system depicted below, where the metric space of labels is the real interval* $L = [0,1]$ *with the Euclidean distance* $d_L(x,y) = |x - y|$*.*



Here, $\eta(x) = \{x, z\}$*,* $\eta(y) = \{x, y, z\}$ *and* $\eta(z) = \{x\}$*. Additionally we have* $\ell(x) = 0.1$*,* $\ell(y) = 0.6$ *and* $\ell(z) = 0.3$ *resulting in* $d_L(\ell(x), \ell(y)) = 0.5$*,* $d_L(\ell(x), \ell(z)) = 0.2$ *and* $d_L(\ell(y), \ell(z)) = 0.3$*. The least fixpoint of* $\mathcal{J}$ *is a pseudo-metric* $\mu\mathcal{J}$ *given by* $\mu\mathcal{J}(x,y) = \mu\mathcal{J}(y,z) = 0.5$ *and* $\mu\mathcal{J}(x,z) = 0.2$*. Since* $\mu\mathcal{J}$ *is a pseudo-metric, the remaining entries are fixed:* $\mu\mathcal{J}(u,u) = 0$ *and* $\mu\mathcal{J}(u,v) = \mu\mathcal{J}(v,u)$ *for all* $u, v \in \{x, y, z\}$*.*

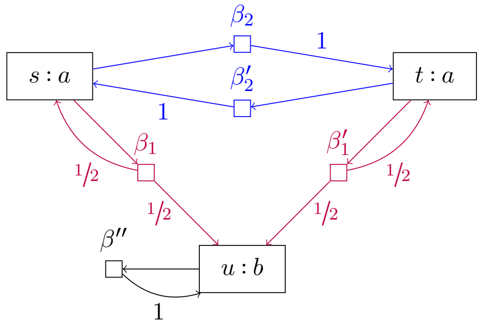


Fig. 2.9.: A probabilistic automaton.

Looking Ahead. In the next section on probailistic automata we will discuss that any metric transistion system can be seen as an instance of a probabilistic automata.

In Section 3.6.5 we will derive the approximation of $\mathcal{J}$.

In Section 5.3.5 we show how one can apply strategy iteration to compute $\mu\mathcal{J}$.

2.6.5. Behavioural Distances for Probabilistic Automata

When combining branching systems with probabilities we obtain probabilistic automata.

Definition 2.6.14 (probabilistic automaton)**.** *A **probabilistic automaton** is a tuple $\mathsf{PA} = (S, \eta, L, \ell)$ consisting of a finite set of states S, a transition function $\eta\colon S \to \mathcal{P}_f(\mathcal{D}(S))$ and a labeling function $\ell\colon S \to L$ for a set of labels L.*

Consider the probabilistic automaton in Fig. 2.9 with state space $Y = \{s, t, u\}$, labels $\ell(s) = \ell(t) = a$ and $\ell(u) = b$ and probability distributions $\beta_1, \beta_2, \beta_1', \beta_2', \beta''$ as indicated. For instance, from state s, there are two possible transitions β_1 which with probability $1/2$ goes to u and with probability $1/2$ stays in s, and β_2 which goes to t with probability 1.

Any probabilistic automaton can be seen as a branching Markov chain over some metric space (L, d_L). As we have for labeled Markov chains, $d_L\colon L \times L \to \{0, 1\}$ will be the discrete metric assigning a distance of one to any two differing elements (as it is in [BBL+21]). In fact, any labeled Markov chain can be seen as an instance of a probabilistic automata where for each state there is only one transition to some probability distribution over S, i.e. $\eta\colon S \to \mathcal{D}(S)$.

On the other hand, any metric transition system can be seen as a special kind of probabilistic automaton as well (with generic distances on labels). Given a state $s \in S$, let β_s denote the Dirac distribution, assigning probability 1 to s and 0 to all other states. Then we can "transform" the transition relation $\eta\colon S \to \mathcal{P}(S) = \mathcal{P}_f(S)$ into $\eta'\colon S \to \mathcal{P}_f(\mathcal{D}(S))$, defining $\eta'(s) = \{\beta_t \mid t \in \eta(s)\}$.

Bisimilarity for probabilistic automata combines probabilistic bisimilarity with branching bisimilarity:

Definition 2.6.15 (bisimulation (probabilistic automaton))**.** *Let* $\mathsf{PA} = (S, \eta, L, \ell)$ *be a probabilistic automaton. An equivalence relation* $R \subseteq S \times S$ *is a probabilistic bisimulation if the following holds for all* $s, t \in S$ *with* sRt*:*

- $\ell(s) = \ell(t)$
- *if* $s \to p_s$ *then for all equivalence classes* $C \in S/R$ *there exists some* $t \to p_t$ *with* $p_s(C) = p_t(C)$
- *if* $t \to p_t$ *then for all equivalence classes* $C \in S/R$ *there exists some* $s \to p_s$ *with* $p_s(C) = p_t(C)$

Again, we have $p(C) = \sum_{c \in C} p(c)$*.*

Two states $s, t \in S$ are bisimilar, written $s \sim t$ if they are related by some bisimulation. The relation $\sim$ is the union of all bisimulations. $\sim$ is a bisimulation itself and an equivalence relation. Bisimilar states have the same observable behaviour.

Combining Hausdorff- and Kantorovich-liftings, we obtain the following operator which yields the behavioural distance for probabilistic automata.

Definition 2.6.16 (fixpoint operator - behavioural distances for probabilistic automata)**.** *Given a probabilistic automaton* $\mathsf{PA} = (S, \eta, L, \ell)$*, we define the function* $\mathcal{M}\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ *for* $d \in [0,1]^{S\times S}$ *and* $s, t \in S$ *as*

$$\mathcal{M}(d)(s,t) = \max\{d_L(\ell(s), \ell(t)), \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t))\}$$

which simplifies whenever d_L *is the discrete metric to*

$$\mathcal{M}(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t)) & \text{otherwise} \end{cases}.$$

The least fixpoint of $\mathcal{M}$ yields the behavioural distance function. Again, this fixpoint is usually not unique. It holds that $\mu\mathcal{M}(s,t) = 0$ iff $s \sim t$ [BBL⁺21].

Example 2.6.17. *Consider the probabilistic automaton in Figure 2.9 with state space* $Y = \{s, t, u\}$*, labels* $\ell(s) = \ell(t) = a$ *and* $\ell(u) = b$ *and probability distributions* $\beta_1, \beta_2, \beta_1', \beta_2', \beta''$ *as indicated.*

In order to explain how the function $\mathcal{M}$*, resulting from the combination of Hausdorff and Kantorovich lifting, works, let us consider the pseudometric* $d(s,t) = 1/2$*,* $d(s,u) = d(t,u) = 1$*. This is not the least fixpoint of* $\mathcal{M}$*, since the distance of states* s, t *is clearly* 0 *as the two states exihibit the same behaviour.*

We now illustrate how to compute $\mathcal{M}(d)(s,t)$*. We obtain* $\mathcal{M}(d)(s,u) = \mathcal{M}(d)(t,u) = 1$ *and, since* $\ell(s) = \ell(t) = a$*, we have*

$$\mathcal{M}(d)(s,t) = \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)).$$

where $\delta(s) = \{\beta_1, \beta_2\}$ *and* $\delta(t) = \{\beta_1', \beta_2'\}$.

It is not difficult to see that the vertices of the coupling polytope $\Omega(\beta_1, \beta_1')$ *are* $\Omega_V(\beta_1, \beta_1') = \{\omega_1, \omega_2\}$ *with*

$$\omega_1(s,t) = 1/2,\ \omega_1(u,u) = 1/2 \quad \textit{and} \quad \omega_2(s,u) = 1/2,\ \omega_2(u,t) = 1/2$$

and $\omega_i(x,y) = 0$, $i \in \{1,2\}$, *for every other pair* $(x,y) \in S \times S$. *Then the Kantorovich lifting is determined as follows:*

$$\mathcal{K}(d)(\beta_1, \beta_1') = \min\{\sum_{x,y \in S} d(x,y) \cdot \omega_1(x,y), \sum_{x,y \in S} d(x,y) \cdot \omega_2(x,y)\} = \min\{1/4, 1\} = 1/4.$$

Similarly we can obtain $K(d)(\beta_1, \beta_2') = 1/2$, $K(d)(\beta_2, \beta_1') = 1/2$, $K(d)(\beta_2, \beta_2') = 1/4$.

In order to conclude the computation via the Hausdorff lifting, note that the minimal set-couplings of $\delta(s) = \{\beta_1, \beta_2\}$ *and* $\delta(t) = \{\beta_1', \beta_2'\}$ *are*

$$R_1 = \{(\beta_1, \beta_1'), (\beta_2, \beta_2')\} \qquad R_2 = \{(\beta_1, \beta_2'), (\beta_2, \beta_1')\}$$

and any other set-coupling includes R_1 *or* R_2. *Then we obtain*

$$\begin{aligned}\mathcal{M}(d)(s,t) &= \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) \\ &= \min\{\max_{(x,x') \in R_1} \mathcal{K}(d)(x,x'), \max_{(x,x') \in R_2} \mathcal{K}(d)(x,x')\} \\ &= \min\{\max\{\mathcal{K}(d)(\beta_1, \beta_1'), \mathcal{K}(d)(\beta_2, \beta_2')\}, \max\{\mathcal{K}(d)(\beta_1, \beta_2'), \mathcal{K}(d)(\beta_2, \beta_1')\}\} \\ &= \min\{\max\{1/4, 1/4\}, \max\{1/2, 1/2\}\} = \min\{1/4, 1/2\} = 1/4.\end{aligned}$$

We observe that other kinds of probabilistic automata, e.g., those originally introduced by Rabin [Rab63], where transitions rather than states are labelled and some states are marked as final, i.e., the transition relation is of the kind $\eta : S \to \{0,1\} \times \mathcal{D}(S)^L$, or their non-deterministic variant, can be easily cast into our framework.

Computation of $\mu\mathcal{M}$. $\mu\mathcal{M}$ can be computed via Kleene iteration (from below) to obtain an approximation. The paper [BBL+21] presents an on-the-fly algorithm which computes $\mu\mathcal{M}$. They also show how to obtain the bisimilarity distance as the solution of a simple stochastic game.

Looking Ahead. As we will see, this on-the-fly algorithm can be seen as an instance of the strategy iterations we propose and we will show this relation in Section 5.3.7. This algorithm encounters the problem that it gets stuck at any fixpoint of $\mathcal{M}$. Similar to what we will be doing, the authors compute a self-closed relation which gives the set of states whose values can be reduced in a fixpoint that is not the least. We will derive the approximation of $\mathcal{M}$ in Section 3.6.5 and show that this self-closed relation can be seen as an instance of our theory. In fact, this paper gave us motivation to search for a generalization.

2.7. Application: Two-Player-Games

In this section we introduce a few games - in particular two-player-games - played on directed graphs. These are discounted mean-payoff games (Section 2.7.2), simple stochastic games (Section 2.7.3) and energy games (Section 2.7.4). These three games are closely related and can in fact be reduced into one another (see [CF11] for a nice overview). They are however different enough to illustrate different aspects of the theories in later chapters. Due to their rather simple nature they prove very useful to this end. Especially strategy iteration is most intuitive when it comes to playing a game.

2.7.1. General Definitions

In this first part we give the general framework and definitions which can be instantiated to all three examples. We start by defining a graph.

> **Definition 2.7.1** (graph)**.** *A **graph** $G = (V, E)$ consists of a finite set of vertices V and a finite set of edges $E \subseteq V \times V$. We denote $n = |V|$ to be the number of vertices and $m = |E|$ the number of edges.*

We will exclusively consider **directed** graphs. $succ(v)$ denotes the **set of successors** of state $v \in V$, i.e. $succ(v) = \{v' \in V \mid (v, v') \in E\}$.

The games we aim to analyze, begin by placing a pebble on some state $v_0 \in V$. We have two players - Max and Min - which both control a subset of states where they choose where the pebble moves to. Some states are controlled by the environment where the pebble moves out of control of either player. This motivates the following definition.

> **Definition 2.7.2** (two-player-graph)**.** *A **two-player-graph** is a directed graph $G = (V, E)$ where the set of vertices V is given as the union of disjoint sets $V_{Max}, V_{Min}, V_{Env}$, i.e. $V = V_{Max} \cup V_{Min} \cup V_{Env}$. Additionally, it holds that $succ(v) \neq \emptyset$ for all $v \in V_{Max} \cup V_{Min}$.*

States in V_{Max} are controlled by player Max, states in V_{Min} are controlled by player Min and states in V_{Env} are controlled by the environment. The environment is non-biased and can for example move the pebble according to some probability distribution.

A play on a graph is defined as follows.

> **Definition 2.7.3** (play)**.** *Given some directed graph $G = (V, E)$. Any finite or infinite sequence $pl^{v_0} = v_0, v_1, v_2, \dots$ of states where $(v_i, v_{i+1}) \in E$ for all $i \geq 0$, is called a **play** in G starting from v_0. Such a play might be finite or infinite. Pl_G^v denotes the (possibly infinite) set of plays in G starting from v.*

A play in G results in some payoff which player Max obtains from player Min. Thus, the games we consider are zero-sum games (see [Mor94]). Player Max wants to play in a way which maximizes this payoff, whereas player Min wants to minimize it.

For each of the three games we will discuss, both players Max and Min have optimal positional strategies, i.e. it is optimal for them to move the pebble to the same fixed successor when in a state controlled by them and either player does not need to consider the history of the game. This is at least the case as long as the other player plays optimally which he has no benefit of deviating from.

Definition 2.7.4 (positional strategy)**.** *A* ***positional strategy*** *for player Max is a mapping* $\sigma\colon V_{\mathsf{Max}} \to V$ *where* $(v, \sigma(v)) \in E$ *for all* $v \in V_{\mathsf{Max}}$*.* Σ *denotes the set of all strategies of player Max.*
A ***positional strategy*** *for player Min is a mapping* $\tau\colon V_{\mathsf{Min}} \to V$ *where* $(v, \tau(v)) \in E$ *for all* $v \in V_{\mathsf{Min}}$*.* Π *denotes the set of all strategies of player Min.*

Since $|V|$ and $|E|$ are finite, the number of positional strategies for both players are finite as well.

Strategies induce new graphs where edges not chosen in a strategy are removed. It proves convenient to define them as follows.

Definition 2.7.5 (strategy-induced graph)**.** *Given a two-player-graph* $G = (V, E)$ *and strategies* $\sigma \in \Sigma$ *and* $\tau \in \Pi$ *for players Max and Min. We define the following* ***strategy-induced graphs***

- $G_\sigma = (V, E_\sigma)$ *where* $E_\sigma = E \setminus \{(v, v') \mid v \in V_{\mathsf{Max}}, \sigma(v) \neq v'\}$*, i.e. we remove all edges not chosen by the strategy* σ*.*
- $G_\tau = (V, E_\tau)$ *where* $E_\tau = E \setminus \{(v, v') \mid v \in V_{\mathsf{Min}}, \tau(v) \neq v'\}$*, i.e. we remove all edges not chosen by the strategy* τ*.*
- $G_{\sigma,\tau} = (V, E_{\sigma,\tau})$ *where* $E_{\sigma,\tau} = E_\sigma \cap E_\tau$*, i.e. we remove all edges not chosen by the strategies* σ *and* τ*.*

Strategy-induced graphs are again two-player-graphs. Playing by a positional strategy is like playing on the respective strategy-induced graph. A play on a strategy-induced graph is defined as follows.

Definition 2.7.6 (strategy-induced play)**.** *Given a two-player-graph $G = (V, E)$ and strategies $\sigma \in \Sigma$ and $\tau \in \Pi$ for players* Max *and* Min*. A **strategy-induced play** is a play*

- *pl^v_σ in G_σ starting from $v \in V$, i.e. $pl^v_\sigma \in Pl^v_{G_\sigma}$,*
- *pl^v_τ in G_τ starting from $v \in V$, i.e. $pl^v_\tau \in Pl^v_{G_\tau}$,*
- *$pl^v_{\sigma,\tau}$ in $G_{\sigma,\tau}$ starting from $v \in V$, i.e. $pl^v_{\sigma,\tau} \in Pl^v_{G_{\sigma,\tau}}$.*

It will prove convenient to assign a payoff to pairs of strategies and a given starting vertex, instead of plays.

Definition 2.7.7 (payoff-function)**.** *Given some two-player-graph $G = (V, E)$ and strategies $\sigma \in \Sigma$ and $\tau \in \Pi$ for players* Max *and* Min*. A **payoff-function** is a map $P: V \times \Sigma \times \Pi \to \mathbb{R}^\infty$, assigning some real payoff to each state and pair of strategies.*

Such a payoff-function needs to be instantiated to the game at hand. Nevertheless, and convenient for our purposes, it allows one to define optimal strategies.

Definition 2.7.8 (optimal strategies)**.** *Given some two-player-graph $G = (V, E)$ and a payoff-function $P: V \times \Sigma \times \Pi \to \mathbb{R}^\infty$. Strategies $\sigma^* \in \Sigma$ and $\tau^* \in \Pi$ for players* Max *and* Min *are **optimal** (w.r.t. P), if for all $\sigma \in \Sigma$, for all $\tau \in \Pi$ and for all $v \in V$ the following holds:*

$$P(v, \sigma, \tau^*) \le P(v, \sigma^*, \tau^*) \le P(v, \sigma^*, \tau).$$

For all three games we consider, optimal positional strategies exist for both players.

The payoff which is obtained when both players play optimally starting from state $v \in V$ is what we aim to compute. We interpret this as the solution of a game.

Definition 2.7.9 (solution)**.** *Given some two-player-graph $G = (V, E)$, a payoff-function $P: V \times \Sigma \times \Pi \to \mathbb{R}^\infty$ and optimal strategies $\sigma^* \in \Sigma$ and $\tau^* \in \Pi$ for players* Max *and* Min*. The map $v^*: V \to \mathbb{R}^\infty$ defined as*

$$v^*(v) = P(v, \sigma^*, \tau^*)$$

*is called the **solution** of G (w.r.t. P).*

For all three games we can find the solution by computing the least fixpoint of an endofunction over a complete lattice which allows us to apply the theories in later chapters. This is sadly not the case for closely related two-player-games like parity games or mean-payoff games.

2.7.2. Discounted Mean-Payoff Games

We start by describing discounted mean-payoff games - first introduced in [ZP95]. These are very similar to the better known mean-payoff games [ZP95] which sadly do not fit into our framework as one can not obtain a solution as a fixpoint of some function. As we will see the fixpoint function, which computes the solution of a discounted mean-payoff game, has a unique fixpoint which is in general not the case for the other two-player-games we will consider.

Definition 2.7.10 (discounted mean-payoff game)**.** *A* ***discounted Mean-Payoff Game*** *is a tuple* $\Gamma_M = (G, w, \lambda)$ *where* $G = (V, E)$ *is a two-player-graph with* $V_{\mathsf{Env}} = \emptyset$*. Additionally, we are given some discount factor* $\lambda \in (0,1)$ *and some map* $w\colon E \to \mathbb{Z}$*, assigning a weight* $w(e)$ *to each edge* $e \in E$*. We define* $W = \max\{|w(e)| \mid e \in E\}$*.*

A discounted mean-payoff game is played as follows:

- A pebble is placed on some initial state $v \in V$.
- If the pebble is in some state $v \in V_{\mathsf{Max}}$, player Max moves the pebble to some state $v' \in succ(v)$.
- If the pebble is in some state $v \in V_{\mathsf{Min}}$, player Min moves the pebble to some state $v' \in succ(v)$.

Given positional strategies σ and τ for players Max and Min, $(G_\sigma, w_\sigma, \lambda)$, $(G_\tau, w_\tau, \lambda)$ and $(G_{\sigma,\tau}, w_{\sigma,\tau}, \lambda)$ are discounted mean-payoff games as well. Here, $w_\sigma\colon E_\sigma \to \mathbb{Z}$, $w_\tau\colon E_\tau \to \mathbb{Z}$ and $w_{\sigma,\tau}\colon E_{\sigma,\tau} \to \mathbb{Z}$ are defined as $w\colon E \to \mathbb{Z}$ on existing edges.

There also exists a unique infinite play $pl^v_{\sigma,\tau}$ in $G_{\sigma,\tau}$ starting from $v \in V$ (as each state has only one successor). We obtain the following payoff-function.

Definition 2.7.11 (payoff-function of a discounted mean-payoff game)**.** *Given some discounted mean-payoff game* $\Gamma_M = (G, w, \lambda)$*. We define the payoff-function* $P_M\colon V \times \Sigma \times \Pi \to [-W, W]$ *for strategies* $\sigma \in \Sigma$*,* $\tau \in \Pi$ *and* $v_0 \in V$ *as follows:*

$$P_M(v_0, \sigma, \tau) = (1-\lambda) \cdot \sum_{i=0}^{\infty} \lambda^i \cdot w(v_i, v_{i+1})$$

where $pl^{v_0}_{\sigma,\tau} = v_0, v_1, v_2, \ldots$ *is the unique play starting from* v_0 *in* $G_{\sigma,\tau}$*.*

It is immediate that $-W \leq P_M(v_0, \sigma, \tau) \leq W$: For any play $pl^{v_0}_{\sigma,\tau} = v_0, v_1, v_2, \ldots$ we have

$$\begin{aligned} -W &= \frac{1-\lambda}{1-\lambda} \cdot (-W) = (1-\lambda) \cdot \sum_{i=0}^{\infty} \lambda^i \cdot (-W) \\ &\leq (1-\lambda) \cdot \sum_{i=0}^{\infty} \lambda^i \cdot w(v_i, v_{i+1}) = P_M(v_0, \sigma, \tau) \end{aligned}$$

$$\leq (1-\lambda)\cdot\sum_{i=0}^{\infty}\lambda^i\cdot W = \frac{1-\lambda}{1-\lambda}\cdot W = W$$

by geometric series.

For discounted mean-payoff games optimal positional strategies exist for both players.

Lemma 2.7.12. *For discounted mean-payoff games the following holds: There exist optimal positional strategies $\sigma^* \in \Sigma$ and $\tau^* \in \Pi$ for both players Max and Min.*

Proof. See [ZP95]. □

The solution $v_M^*: V \to \mathbb{R}$, $v_M^*(v) = P_M(v, \sigma^*, \tau^*)$, of some discounted mean-payoff game Γ_M is obtained as the unique fixpoint of the following function.

Definition 2.7.13 (fixpoint operator for discounted mean-payoff games)**.** *Given some discounted mean-payoff game $\Gamma_M = (G, w, \lambda)$, we define the operator $\mathcal{L}: [-W,W]^V \to [-W,W]^V$ for $v \in V$ and $a: V \to [-W,W]$ as follows:*

$$\mathcal{L}(a)(v) = \begin{cases} \max_{(v,u)\in E}(1-\lambda)\cdot w(v,u) + \lambda\cdot a(u), & v \in V_{\mathsf{Max}} \\ \min_{(v,u)\in E}(1-\lambda)\cdot w(v,u) + \lambda\cdot a(u), & v \in V_{\mathsf{Min}} \end{cases}.$$

[ZP95] show that $\mathcal{L}$ has a unique fixpoint.

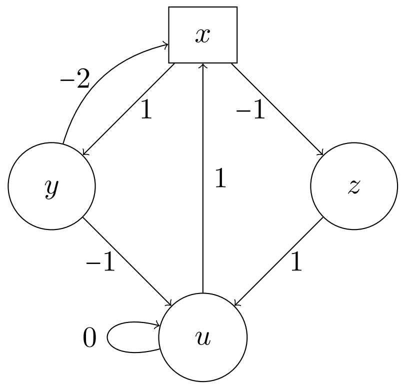
Lemma 2.7.14. *Given some discounted mean payoff game $\Gamma_M = (G, w, \lambda)$. We have $\mu\mathcal{L} = v_M^*$, i.e. the least (and unique) fixpoint $\mu\mathcal{L}$ of $\mathcal{L}$ coincides with the solution v_M^* of Γ_M.*

Proof. See [ZP95]. □

Computation of $\mu\mathcal{L}$. Note that $[-W,W]^V$ is a complete lattice (see Example 2.3.4). Thus we can compute $\mu\mathcal{L} = \nu\mathcal{L}$ via Kleene iteration (from above and below). This yields an approximation.

The literature is rather scarce when it comes to discounted mean-payoff games since they arise from the more common and complex mean-payoff games.

Example 2.7.15. *We are given the following discounted mean-payoff game $\Gamma_M = (G, w, \lambda)$. Squares belong to player Max whereas circles belong to player Min.*



We have $V = \{x, y, z, u\}$ and $\lambda = 0.5$. Edges and their weights are to be taken from the picture ($W = 2$). The following table displays the computation of $\mu\mathcal{L}$ via Kleene iteration (from below):

	$a^{(0)}$	$a^{(1)}$	$a^{(2)}$	$a^{(3)}$	$a^{(4)}$	$a^{(5)}$	...	$\mu\mathcal{L}$
x	-2	$-1/2$	$-1/2$	$-1/8$	$-1/8$	$-1/32$	...	0
y	-2	-2	$-5/4$	$-5/4$	$-17/16$	$-17/16$	...	-1
z	-2	$-1/2$	0	$1/4$	$3/8$	$7/16$	...	$1/2$
u	-2	-1	$-1/2$	$-1/4$	$-1/8$	$-1/16$	...	0

The iteration converges towards $\mu\mathcal{L}$. To perform Kleene iteration from above, just replace $a^{(0)} \equiv W$.

Looking Ahead. Since $\mathcal{L}$ has a unique fixpoint there is no need to derive an approximation.

We will show how to apply our strategy iterations to discounted mean-payoff games in Section 5.3.1 and perform a short runtime comparison in Section 5.4.1.

2.7.3. Simple Stochastic Games

Simple stochastic games, first introduced in [Con92], are well known and studied. Here, the game contains probabilistic transitions and sink states which yield some payoff.

Definition 2.7.16 (simple stochastic game)**.** *A **simple stochastic game** is a tuple $\Gamma_S = (G, p, c)$ where $G = (V, E)$ is a two-player-graph with $V_{\mathsf{Env}} = V_{\mathsf{Av}} \cup V_{\mathsf{Sink}}$. We have $V_{\mathsf{Av}} \cap V_{\mathsf{Sink}} = \emptyset$ and $succ(v) \neq \emptyset$ for all $v \in V_{\mathsf{Av}}$. Additionally, we are given some map $p\colon V_{\mathsf{Av}} \to \mathcal{D}(V)$, assigning a probability distribution $p(v)$ to each state $v \in V_{\mathsf{Av}}$[9] and some map $c\colon V_{\mathsf{Sink}} \to [0, 1]$, assigning some payoff to each state $v \in V_{\mathsf{Sink}}$.*

A simple stochastic game is played as follows:

[9]We have $p(v)(v') > 0$ iff $v' \in succ(v)$.

- A pebble is placed on some initial state $v \in V$.
- If the pebble is in some state $v \in V_{\mathsf{Max}}$, player Max moves the pebble to some state $v' \in succ(v)$.
- If the pebble is in some state $v \in V_{\mathsf{Min}}$, player Min moves the pebble to some state $v' \in succ(v)$.
- If the pebble is in some state $v \in V_{\mathsf{Av}}$, the pebble moves to state $v' \in succ(v)$ with probability $p(v)(v')$.
- If the pebble is in some state $v \in V_{\mathsf{Sink}}$, the game ends and player Max obtains payoff $c(v)$ from player Min.

Note that the pebble may never reach a sink state and thus the game never ends. In that case player Max obtains payoff 0 from player Min.

Given positional strategies σ and τ for players Max and Min, (G_σ, p, c), (G_τ, p, c) and $(G_{\sigma,\tau}, p, c)$ are simple stochastic games as well. We can derive the expected payoff of a game starting in $v \in V$:

Definition 2.7.17 ((expected) payoff-function of a simple stochastic game)**.** *Given some simple stochastic game $\Gamma_S = (G, p, c)$. We define the (expected) payoff-function $P_S\colon V \times \Sigma \times \Pi \to [0,1]$ for strategies $\sigma \in \Sigma$, $\tau \in \Pi$ and $v \in V$ as follows:*

$$P_S(v, \sigma, \tau) = \sum_{v' \in V_{Sink}} p^v_{\sigma,\tau}(v') \cdot c(v').$$

where $p^v_{\sigma,\tau}(v')$ denotes the probability of reaching sink state v' starting from $v \in V$ in $G_{\sigma,\tau}$.

It is immediate that $0 \le P_S(v, \sigma, \tau) \le 1$ since

$$0 \le \sum_{v' \in V_{\mathsf{Sink}}} p^v_{\sigma,\tau}(v') \cdot 0 \le \sum_{v' \in V_{\mathsf{Sink}}} p^v_{\sigma,\tau}(v') \cdot c(v') \le \sum_{v' \in V_{\mathsf{Sink}}} p^v_{\sigma,\tau}(v') \cdot 1 \le 1$$

since $\sum_{v' \in V_{\mathsf{Sink}}} p^v_{\sigma,\tau}(v') \le 1$ for all $v \in V$.

For simple stochastic games optimal positional strategies exist for both players.

Lemma 2.7.18. *For simple stochastic games the following holds: There exist optimal positional strategies σ^* and τ^* for both players Max and Min.*

Proof. See [Con92]. □

The solution $v^*_S\colon V \to [0,1]$, $v^*_S(v) = P_S(v, \sigma^*, \tau^*)$ of a simple stochastic game Γ_S coincides with the least fixpoint of the following function.

Definition 2.7.19 (fixpoint operator for simple stochastic games)**.** *Given some simple stochastic game $\Gamma_S = (G, p, c)$, we define the operator $\mathcal{V}\colon [0,1]^V \to [0,1]^V$ for $a\colon V \to [0,1]$ and $v \in V$ as follows:*

$$\mathcal{V}(a)(v) = \begin{cases} \max_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

Lemma 2.7.20. *Given some simple stochastic game $\Gamma_S = (G, q, p)$. We have $\mu\mathcal{V} = v_S^*$, i.e. the least fixpoint $\mu\mathcal{V}$ of $\mathcal{V}$ coincides with the solution v_S^* of Γ_S.*

Proof. See [KRSW20]. □

Note that $[0,1]^V$ is a complete MV-chain (see Example 2.3.4) thus we can use Kleene iteration (from below) to approximate $\mu\mathcal{V}$ (see Example 2.7.22). $\mathcal{V}$ may omit more than one fixpoint thus Kleene iteration from above is not applicable.

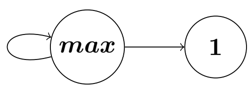
Stopping Simple Stochastic Games. When it comes to algorithms which solve simple stochastic games, the literature mainly considers stopping simple stochastic games.

Definition 2.7.21 (stopping)**.** *A simple stochastic game $\Gamma_S = (G, p, c)$ is called **stopping** if for all strategies $\sigma \in \Sigma$ and $\tau \in \Pi$ of players Max and Min and all $v \in V$ we have*

$$\sum_{v' \in V_{\mathsf{Sink}}} p^v_{\sigma,\tau}(v') > 0.$$

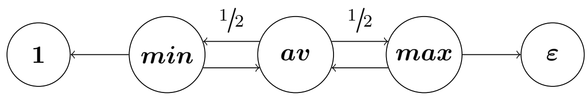
In a stopping simple stochastic game, we will eventually reach a sink state and thus the game will end. Neither player can prevent this from happening.

For stopping simple stochastic games, the operator $\mathcal{V}$ has one unique fixpoint which makes them easier to analyze [KRSW20]. This is not a one-to-one corespondance, i.e. the operator $\mathcal{V}$ may have only one fixpoint for a non-stopping simple stochastic games. See the example below where state $\mathbf{1} \in V_{\mathsf{Sink}}$ has payoff 1 and $\boldsymbol{max} \in V_{\mathsf{Max}}$. The game is clearly non-stopping but $\mathcal{V}$ has a unique fixpoint.



This property of stopping is frequently used in the literature and most algorithms have stopping simple stochastic games in mind. Any simple stochastic game can be transformed into an equivalent stopping simple stochastic game [Con92]. The techniques we introduce in Chapters 3 and 5 allow us to analyze non-stopping simple stochastic games - thus we do not care whether a simple stochastic game is stopping or not. We note that recent research is done on general simple stochastic games [KMW23].

Example 2.7.22. *Consider the following simple stochastic game.*



We have $V_{Max} = \{\boldsymbol{max}\}$, $V_{Min} = \{\boldsymbol{min}\}$, $V_{Av} = \{\boldsymbol{av}\}$ *and* $V_{Sink} = \{\mathbf{1}, \boldsymbol{\varepsilon}\}$. *Transitions and their probabilities are to be taken from the picture. The payoff of state* $\mathbf{1}$ *is 1 and of state* $\boldsymbol{\varepsilon}$ *is* ε. *We assume* ε *is some small positive number.*

We directly obtain $v^*_S(\mathbf{1}) = 1$ *and* $v^*_S(\boldsymbol{\varepsilon}) = \varepsilon$. *It is rather imminent that player Max should move to state* ε, *otherwise if he chooses to move to state* $\boldsymbol{av}$, *player Min can just keep the game in the cycle formed by states* $\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}$. *Here, Max would obtain a payoff of 0. Thus* $\sigma^*(\mathbf{max}) = \boldsymbol{\varepsilon}$ *and* $v^*_S(\mathbf{max}) = \varepsilon$. *Now, player Min does not benefit from moving to state* $\mathbf{1}$ *obtaining a payoff of 1. Thus* $\tau^*(\boldsymbol{min}) = \boldsymbol{av}$, *resulting in* $v^*_S(\boldsymbol{min}) = v^*_S(\boldsymbol{av}) = \varepsilon$.

The following table shows how to compute $\mu\mathcal{V}$ *via Kleene iteration (from below):*

	$a^{(0)}$	$a^{(1)}$	$a^{(2)}$	$a^{(3)}$	$a^{(4)}$	$a^{(5)}$	...	$\mu\mathcal{V}$
$\boldsymbol{min}$	0	0	0	0	$\varepsilon/2$	$\varepsilon/2$	...	ε
$\boldsymbol{av}$	0	0	0	$\varepsilon/2$	$\varepsilon/2$	$3\varepsilon/4$	...	ε
$\boldsymbol{max}$	0	0	ε	ε	ε	ε	...	ε
$\mathbf{1}$	0	1	1	1	1	1	...	0
$\boldsymbol{\varepsilon}$	0	ε	ε	ε	ε	ε	...	ε

This simple stochastic game is non-stopping. Thus, $\mathcal{V}$ *may have more than one fixpoint, e.g.* $\nu\mathcal{V}(\mathbf{1}) = \nu\mathcal{V}(\boldsymbol{min}) = \nu\mathcal{V}(\boldsymbol{av}) = \nu\mathcal{V}(\boldsymbol{max}) = 1$ *and* $\nu\mathcal{V}(\boldsymbol{\varepsilon}) = \varepsilon$.

Computation of $\mu\mathcal{V}$. As mentioned, the literature mainly considers stopping simple stochastic games. For them, Kleene iteration from above is applicable since $\mathcal{V}$ omits only one fixpoint. The Hoffmann-Karp algorithm is well known and is in fact an instance of the strategy iterations we describe in Chapter 5. The paper [TVK11] improves on this algorithm by adding randomization.

We can modify any simple stochastic game such that each state has exactly two successors and sink states have a payoff in $\{0, 1\}$. Here, quadratic programming can be used to directly compute $\mu\mathcal{V}$ [KRSW20].

The paper [ABdMS21] proposes a meta-algorithm GSIA such that a number of strategy improvement algorithms for simple stochastic games arise as instances, along with a general complexity bound. Differently from our strategy iterations (Chapter 5), this paper focuses on simple stochastic games and iteration from below. However, it allows for the parametrisation of the algorithm on a subset of edges of interest in the game graph, which is not possible in our approach, and so it can provide interesting suggestions for further generalisations.

Looking Ahead. We compute the approximation of $\mathcal{V}$ in Section 3.6.8.

We can use strategy iteration to compute $\mu\mathcal{V}$ in Section 5.3.2 which is essentially the Hoffmann-Karp algortihm with one extra step (since they consider stopping simple stochastic games).

In Section 5.4.2 we perform a runtime comparison.

2.7.4. Energy Games

Energy games were first proposed in [BCD+11] and further explored in [DKZ19]. These are similar to (discounted) mean-payoff games as they are played on the same graph. Here however, we are interested if one player can keep the energy level positive forever. The other player tries to make the system run out of energy.

Definition 2.7.23 (energy game)**.** *An* ***energy game*** *is a tuple* $\Gamma_E = (G, w)$ *where* $G = (V, E)$ *is a two-player-graph with* $V_{\mathsf{Env}} = \varnothing$*. Additionally, we are given some initial energy level* $c \in \mathbb{N}^\infty$ *and some map* $w\colon E \to \mathbb{Z}$*, assigning a weight* $w(e)$ *to each edge* $e \in E$*. We define* $W = \max\{|w(e)| \mid e \in E\}$*.*

For energy games, the literature refers to player Min as player 0 and to player Max as player 1. We will stick to our naming convention.

Whenever an edge $e \in E$ is traversed, we add $w(e)$ to the current energy level. Thus, if $w(e)$ is positive, the energy level increases and decreases whenever $w(e)$ is negative. An energy game is played as follows:

- A pebble is placed on some initial state $v \in V$. The energy level is c.
- If the pebble is in some state $v \in V_{\mathsf{Min}}$, player Min moves the pebble to some state $v' \in succ(v)$. The energy level increases/reduces by $w(v, v')$. If the energy level is negative, the game ends.
- If the pebble is in some state $v \in V_{\mathsf{Max}}$, player Max moves the pebble to some state $v' \in succ(v)$. The energy level increases/reduces by $w(v, v')$. If the energy level is negative, the game ends.

Player Max wants the energy to run out whereas player Min wants the game to never end.

Given positional strategies σ and τ for players Max and Min, (G_σ, w_σ), (G_τ, w_τ) and $(G_{\sigma,\tau}, w_{\sigma,\tau})$ are energy games as well. Here, $w_\sigma\colon E_\sigma \to \mathbb{Z}$, $w_\tau\colon E_\tau \to \mathbb{Z}$ and $w_{\sigma,\tau}\colon E_{\sigma,\tau} \to \mathbb{Z}$ are defined as $w\colon E \to \mathbb{Z}$ on existing edges.

For energy games, we are interested in finding out how much initial credit player Min requires to keep the game going forever. Player Min might require an inital credit of ∞ to this end. It is rather imminent that whenever player Min requires an energy larger than $n \cdot W$ (there exist lower bounds) that he in fact requires an initial energy of ∞ [BCD+11].

This problem subsumes the problem of finding whether player Max can make the system run out of energy thus ending the game when some initial credit is given. This motivates the following definition.

Definition 2.7.24 (feasible potential, solution of an energy game)**.** *Let* $\Gamma_E = (G, w)$ *be an energy game. A function* $f\colon V \to \mathbb{N}^\infty$ *is a* ***feasible potential*** *iff for every* $v \in V$

- *if* $v \in V_{\mathsf{Min}}$, *then* $f(v) + w(v, v') \geq f(v')$ *for some* $(v, v') \in E$.
- *if* $v \in V_{\mathsf{Max}}$, *then* $f(v) + w(v, v') \geq f(v')$ *for all* $(v, v') \in E$.

We call the feasible potential $g(v) = \min\{f(v) \mid f \text{ is a feasible potential}\}$ *the* ***solution*** *of the energy game* Γ_E.

$g(v)$ equals the minimal required initial credit for player Min to keep the game going forever starting from $v \in V$. To justify calling g the solution as in Definition 2.7.9, we define a matching payoff-function.

Definition 2.7.25 (payoff-function of an energy game)**.** *Given some energy game* $\Gamma_E = (G, w)$. *We define the payoff-function* $P_E\colon V \times \Sigma \times \Pi \to \mathbb{R}^\infty$ *for strategies* $\sigma \in \Sigma$, $\tau \in \Pi$ *and* $v \in V$ *as follows:*

$$P_E(v, \sigma, \tau) = g_{\sigma,\tau}(v)$$

where $g_{\sigma,\tau}$ *is the solution (according to Definition 2.7.24) of* $\Gamma_E = (G_{\sigma,\tau}, w)$.

Now, $v_E^*\colon V \to \mathbb{N}^\infty$, $v_E^*(v) = P_E(v, \sigma^*, \tau^*)$ equals g.

Lemma 2.7.26. *We have* $v_E^* = g$.

Proof. We have $P_E(v, \sigma^*, \tau^*) = g_{\sigma^*,\tau^*}$. Since σ^* and τ^* are optimal strategies, it holds $g = g_{\sigma^*,\tau^*}$. □

Optimal positional strategies exist for both players.

Lemma 2.7.27. *For energy games the following holds: There exist optimal positional strategies* σ^* *and* τ^* *for both players 0 and 1.*

Proof. See [DKZ19]. □

g equals the least fixpoint of the following operator.

Definition 2.7.28 (fixpoint operator of energy games)**.** *Given some energy game $\Gamma_E = (G, w)$, we define the operator $\bar{\mathcal{E}}\colon (\mathbb{N}^\infty)^V \to (\mathbb{N}^\infty)^V$ for $v \in V$ and $a\colon V \to \mathbb{N}^\infty$ as*

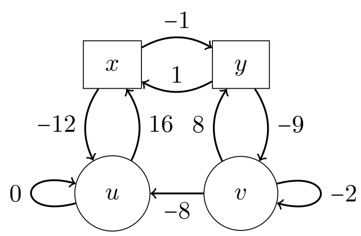
$$\bar{\mathcal{E}}(a)(v) = \begin{cases} \min\limits_{v' \in succ(v)} \max\{a(v') - w(v,v'), 0\} & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{v' \in succ(v)} \max\{a(v') - w(v,v'), 0\} & \text{if } v \in V_{\mathsf{Max}} \end{cases}$$

Lemma 2.7.29. *Given some energy game $\Gamma_E = (G, w)$. We have $\mu\bar{\mathcal{E}} = g$, i.e. the least fixpoint $\mu\bar{\mathcal{E}}$ of $\bar{\mathcal{E}}$ coincides with the solution g of Γ_E.*

Proof. See [DKZ19]. □

Note that $(\mathbb{N}^\infty)^V$ is a complete lattice (see Example 2.3.4) but not a complete MV-chain. Still, Kleene iteration (from below) is applicable and even obtains exact computations as we can set states to ∞ whenever they exceed $n \cdot W$. Note that $\bar{\mathcal{E}}$ may have more than one fixpoint.

Example 2.7.30. *Consider the following energy game, where it is intended that circular and rectangular states belong to player* Min *and player* Max*, respectively.*



The optimal strategy for player Min *is to choose u as the successor to u and v. Thus v requires an initial energy of 8 to keep going forever. For u an initial energy of 0 is sufficient. On the other hand, the optimal strategy for player* Max *is to choose y as successor to x and v as successor to y. This results in a required initial energy of 17 for y and 18 for x.*

Thus, we obtain as least fixpoint $g(x) = 18$, $g(y) = 17$, $g(u) = 0$, $g(v) = 8$ of $\bar{\mathcal{E}}$. Note that, if from u player Min *would choose x, player* Max *could keep the game in a negative cycle.*

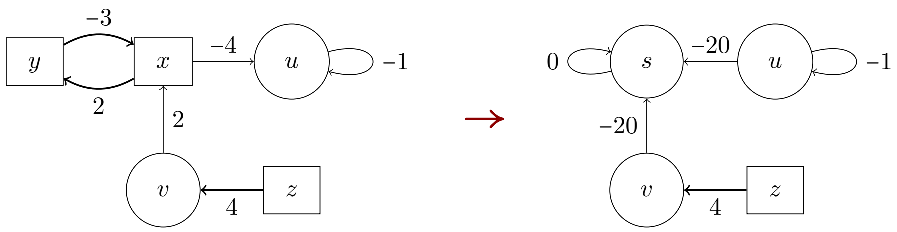
Transformation to Energy Games with Finite Values. Now, since $(\mathbb{N}^\infty)^V$ is not an MV-chain we can not apply the techniques in Chapters 3 and 5. As shown in [DKZ19] any energy game $\Gamma_E = (G, w)$ can be transformed into an energy game $\Gamma'_E = (G', w')$ with finite values.

To this end, an additional state s is introduced. s has only one outgoing edge looping to itself with weight $w(s,s) = 0$. Now, for every state $v \in V_{\mathsf{Min}}$ we add an edge (v,s) with weight $-2nW$. State s serves as an emergency exit for player Min whenever she realizes that the energy she requires tends to infinity. Next, we have to detect all states by player Max which are part of some negative cycle exclusively consisting of player Max states. We remove all these states and all states of player Max which can reach such a negative cycle along a path consisting exclusively of player Max states. All incoming and outgoing edges of these removed states are removed as well. For these removed states it holds that an initial energy of ∞ is required. We thus obtain a new graph $G' = (V', E')$ and according to [DKZ19] it holds for the solutions g of Γ_E and g' of Γ'_E and all $v \in V$ that

$$g(v) = \begin{cases} g'(v) & \text{if } g'(v) < n \cdot W \\ \infty & \text{otherwise} \end{cases}$$

thus we can reconstruct the solution g of Γ_E from the solution g' of $\Gamma_{E'}$.

Example 2.7.31. *The energy game $\Gamma_E = (G, w)$ to the left is transformed to an energy game with finite values $\Gamma'_E = (G', w')$ as follows:*



To clarify, states x and y are removed as they form a negative cycle consisting of states belonging to player Max. State z is not removed - although the required initial energy is ∞ - since its only successor is v which belongs to player Min.

For an energy game with finite values it holds that the solution $g(v)$ is bounded by some natural number $k \in \mathbb{N}$ (e.g. $3 \cdot n \cdot W$) in each element $v \in V$. Let $K = \{0, \ldots, k\}$ which is a complete MV-chain (see Example 2.3.17) and define the operator $\ominus_{\mathbb{Z}}: K \times \mathbb{Z} \to K$ by $x \ominus_{\mathbb{Z}} y = \min\{\max\{x - y, 0\}, k\}$. We define $\mathcal{E}: K^V \to K^V$ for $a: V \to K$ and $v \in V$ as

$$\mathcal{E}(a)(v) = \begin{cases} \min\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v, v') & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v, v') & \text{if } v \in V_{\mathsf{Max}} \end{cases}$$

Lemma 2.7.32. *Let $\Gamma_E = (G, w)$ be an energy game with finite values, bounded by k. Then $\mu\mathcal{E} = g$, i.e. the least fixpoint of $\mathcal{E}$ coincides with the solution of Γ_E.*

Proof. See Appendix: Lemma A.1.5. □

Since K is a complete MV-chain we can derive an approximation for $\mathcal{E}$.

Computation of $\mu\mathcal{E}$. The transformation to an energy game with finite values is not required for the algorithms described in the literature. The paper [BCD+11] proposes an easy value iteration which can be used to compute $\mu\bar{\mathcal{E}}$. In the paper [DKZ19] one can find a more involved algorithm.

Looking Ahead. We will derive the approximation of $\mathcal{E}$ in Section 3.6.9.

We can apply our strategy iterations to compute $\mu\mathcal{E}$ (Section 5.3.3).

Lastly, we will perform a runtime comparison in Section 5.4.3 where we also briefly review the value iteration in [BCD+11].

3 | Fixpoint Theory – Upside Down

In this chapter we will introduce the "Upside-Down" theory which lies at the heart of this thesis as later chapters heavily rely on the theory we will present here. First, we will discuss what we aim to accomplish in the following introduction which also serves as motivation.

3.1. Introduction

Fixpoints are ubiquitous in computer science as they provide a meaning to inductive and coinductive definitions (see, e.g., [San11, NNH10]). A monotone function $f : L \to L$ over a complete lattice $(L, \sqsubseteq)$, by Knaster-Tarski's theorem [Tar55], admits a least fixpoint μf and greatest fixpoint νf which are characterised as the least pre-fixpoint and the greatest post-fixpoint, respectively. This immediately gives well-known proof principles for showing that a lattice element $l \in L$ is *below* νf or *above* μf (cf. Section 2.3)

$$\frac{l \sqsubseteq f(l)}{l \sqsubseteq \nu f} \qquad \frac{f(l) \sqsubseteq l}{\mu f \sqsubseteq l}$$

On the other hand, showing that a given element l is *above* νf or *below* μf is more difficult. One can think of using the characterisation of least and largest fixpoints via Kleene iteration (Theorem 2.3.11). The largest fixpoint is the least element of the (possibly transfinite) descending chain obtained by iterating f from $\top$. Then showing that $f^i(\top) \sqsubseteq l$ for some i, one concludes that $\nu f \sqsubseteq l$. This proof principle is related to the notion of ranking functions. However, this is a less satisfying notion of witness since f has to be applied i times, and this can be inefficient or unfeasible when i is an infinite ordinal.

The aim of this chapter is to present an alternative proof rule for this purpose for functions over lattices of the form $L = \mathbb{M}^Y$ where Y is a finite set and $\mathbb{M}$ is an MV-chain, i.e., a totally ordered complete lattice endowed with suitable operations of sum and complement (cf. Section 2.3.3). This allows us to capture several examples, ranging from ordinary relations for dealing with bisimilarity to behavioural metrics, termination probabilities, simple stochastic games and energy games.

Assume $f : \mathbb{M}^Y \to \mathbb{M}^Y$ monotone and consider the question of proving that some fixpoint $a : Y \to \mathbb{M}$ is the largest fixpoint νf. The idea is to show that there is no “slack” or “wiggle room” in the fixpoint a that would allow us to further increase it. This is done by associating with every $a : Y \to \mathbb{M}$ a function $f_a^\#$ on $\mathcal{P}(Y)$ whose greatest fixpoint gives us the elements of Y where we have a potential for increasing a by adding a constant. If no such potential exists, i.e. $\nu f_a^\#$ is empty, we conclude that a is νf. A similar function

$f^a_\#$ (specifying decrease instead of increase) exists for the case of least fixpoints. Note that the premise is $\nu f^a_\# = \emptyset$, i.e. the witness remains coinductive. The proof rules are:

$$\frac{f(a) = a \qquad \nu f^\#_a = \emptyset}{\nu f = a} \qquad \frac{f(a) = a \qquad \nu f^a_\# = \emptyset}{\mu f = a}$$

For applying the rule we compute a greatest fixpoint on $\mathcal{P}(Y)$, which is finite, instead of working on the potentially infinite $\mathbb{M}^Y$. The rule does not work for all monotone functions $f : \mathbb{M}^Y \to \mathbb{M}^Y$, but we show that whenever f is non-expansive the rule is valid. Actually, it is not only sound, but also reversible, i.e., if $a = \nu f$ then $\nu f^\#_a = \emptyset$, providing an if-and-only-if characterisation of whether a given fixpoint corresponds to the greatest fixpoint.

Quite interestingly, under the same assumptions on f, using a restricted function f^*_a, the rule can be used, more generally, when a is just a *pre-fixpoint* ($f(a) \sqsubseteq a$) and it allows us to conclude that $\nu f \sqsubseteq a$. A dual result holds for *post-fixpoints* in the case of least fixpoints.

$$\frac{f(a) \sqsubseteq a \qquad \nu f^*_a = \emptyset}{\nu f \sqsubseteq a} \qquad \frac{a \sqsubseteq f(a) \qquad \nu f^a_* = \emptyset}{a \sqsubseteq \mu f}$$

The theory above applies to many interesting scenarios: witnesses for non-bisimilarity, algorithms for simple stochastic games [Con92] and energy games [BCD+11], lower bounds for termination probabilities and behavioural metrics in the setting of probabilistic [BBLM17] and metric transition systems [dAFS09] and probabilistic automata [BBL+21]. In particular we were inspired by, and generalise, the self-closed relations of [Fu12], also used in [BBL+21]. See Sections 2.6 and 2.7 for details on these applications.

Motivating Example. Consider a Markov chain (S, T, η) as introduced in Section 2.6.1. S denotes the set of states, $T \subseteq S$ the subset of terminal states and $\eta : S \setminus T \to \mathcal{D}(S)$ the successor function. The termination probability arises as the least fixpoint of the function $\mathcal{T}$ defined in Figure 3.1. The values of $\mu\mathcal{T}$ for the Markov chain on the right are indicated in green (left value).

Now consider the function t assigning to each state the termination probability written in red (right value). It is not difficult to see that t is another fixpoint of $\mathcal{T}$, in which states y and z convince each other incorrectly that they terminate with probability 1, resulting in a vicious cycle that gives "wrong" results. We want to show that $\mu\mathcal{T} \neq t$ without knowing $\mu\mathcal{T}$. Our idea is to compute the set of states that still has some "wiggle room", i.e., those states which could reduce their termination probability by δ if all their successors did the same. This definition has a coinductive flavour and it can be computed as a greatest fixpoint on the finite powerset $\mathcal{P}(S)$ of states, instead of on the infinite lattice $[0,1]^S$.

We hence consider a function $\mathcal{T}^t_\# : \mathcal{P}([S]^t) \to \mathcal{P}([S]^t)$, dependent on t, defined as follows. Let $[S]^t$ be the support of t, i.e., the set of all states s such that $t(s) > 0$, where a reduction in value is in principle possible. Then a state $s \in [S]^t$ is in $\mathcal{T}^t_\#(S')$ iff $s \notin T$ and for all s' for which $\eta(s)(s') > 0$ it holds that $s' \in S'$, i.e. all successors of s are in S'.

$$\mathcal{T} : [0,1]^S \to [0,1]^S$$
$$\mathcal{T}(t)(s) = \begin{cases} 1 & \text{if } s \in T \\ \sum\limits_{s' \in S} \eta(s)(s') \cdot t(s') & \text{otherwise} \end{cases}$$

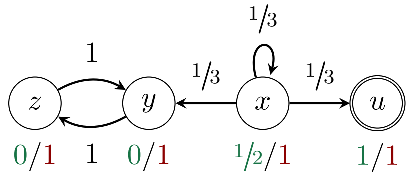


Fig. 3.1.: Function $\mathcal{T}$ (left) and a Markov chain with two fixpoints of $\mathcal{T}$ (right)

The greatest fixpoint of $\mathcal{T}^t_\#$ is $\{y,z\}$ (also called a "vicious cycle"). The fact that it is not empty means that there is some “wiggle room”, i.e., the value of t can be reduced on the elements $\{y,z\}$ and thus t cannot be the least fixpoint of f. Moreover, the intuition that t can be improved on $\{y,z\}$ can be made precise, leading to the possibility of performing the improvement and search for the least fixpoint from there.

Structure of this Chapter. We next aim to formalise the theory outlined above (Section 3.2), showing that the proof rules work for non-expansive monotone functions f on lattices of the form $\mathbb{M}^Y$, where Y is a finite set and $\mathbb{M}$ a (potentially infinite) MV-algebra (Section 3.3). This is done for greatest fixpoints and we describe the dual view for least fixpoints in Section 3.4 which we need for example to handle the motivating example above.

Additionally, as non-expansive functions enjoy good closure properties, given a decomposition of f into smaller sub-functions we show how to obtain the corresponding approximation compositionally (Section 3.5). Then, in order to show that our approach covers a wide range of examples and allows us to derive useful and original algorithms, we discuss various applications in Section 3.6: termination probability, behavioural distances for labeled Markov chains, metric transition systems and probabilistic automata, bisimilarity, simple stochastic games and energy games.

3.2. Approximating the Propagation of Increases

As mentioned in the introduction, our interest is for fixpoints of monotone functions $f : \mathbb{M}^Y \to \mathbb{M}^Y$, where $\mathbb{M}$ is an MV-chain and Y is a finite set. We will see that for non-expansive[1] functions (cf. Definition 2.3.23) we can over-approximate the sets of points in which a given $a \in \mathbb{M}^Y$ can be increased in a way that is preserved by the application of f. This will be the core of the proof rules outlined earlier.

In this Section we will analyze the theory for greatest fixpoints, a dualization of the theory for least fixpoints is done in Section 3.4.

As a reminder (cf. Section 2.3.4), a function $f : \mathbb{M}^Y \to \mathbb{M}^Z$ - where Y, Z are finite sets and $\mathbb{M}$ a complete MV-chain - is non-expansive if for all $a, b \in \mathbb{M}^Y$ it holds that

[1]Any non-expansive functions is monotone by Lemma 2.3.24.

$\|f(b) \ominus f(a)\| \sqsubseteq \|b \ominus a\|$. The norm of $a \in \mathbb{M}^Y$ is defined as $\|a\| = \max\{a(y) \mid y \in Y\}$.

Let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a monotone function and take $a, b \in \mathbb{M}^Y$ with $a \sqsubseteq b$. We are interested in the difference $b(y) \ominus a(y)$ for some $y \in Y$ and on how the application of f "propagates" this difference. The reason is that, understanding that no increase can be propagated will be crucial to establish when a fixpoint of a non-expansive function f is actually the largest one, and, more generally, when a (pre-)fixpoint of f is above the largest fixpoint.

In order to formalise the above intuition, we rely on tools from abstract interpretation. In particular, the following pair of functions, which, under a suitable condition, form a Galois connection (cf. Definition 2.3.13), will play a major role. For this purpose we fix $a \in \mathbb{M}^Y$, $\delta \in \mathbb{M}$. The left adjoint $\alpha_{a,\delta}$ takes as input a set $Y' \subseteq Y$ and, for $y \in Y'$, it increases the values $a(y)$ by δ, while the right adjoint $\gamma_{a,\delta}$ takes as input a function $b \in \mathbb{M}^Y$, $b \in [a, a \oplus \delta]$ and checks for which parameters $y \in Y$ the value $b(y)$ exceeds $a(y)$ by δ.

We also define $[Y]_a$, the subset of elements in Y where $a(y)$ is not 1 and thus there is a potential to increase, and δ_a, which gives us the least of such increases (i.e., the largest increase that can be used on all elements in $[Y]_a$ without "overflowing").

Definition 3.2.1 (functions to sets, and vice versa)**.** *Let $\mathbb{M}$ be an MV-algebra and let Y be a finite set. Define the set $[Y]_a = \{y \in Y \mid a(y) \neq 1\}$ (support of $\overline{a}$) and $\delta_a = \min\{\overline{a(y)} \mid y \in [Y]_a\}$ with $\min \varnothing = 1$. Additionally, for $\delta \in \mathbb{M}$ and $Y' \subseteq Y$, we define $\delta_{Y'}\colon Y \to \mathbb{M}$ for $y \in Y$ as*

$$\delta_{Y'}(y) = \begin{cases} \delta & \textit{if } y \in Y' \\ 0 & \textit{otherwise} \end{cases}.$$

For $0 \sqsubset \delta \in \mathbb{M}$ we consider the functions $\alpha_{a,\delta} : \mathcal{P}([Y]_a) \to [a, a \oplus \delta]$ and $\gamma_{a,\delta} : [a, a \oplus \delta] \to \mathcal{P}([Y]_a)$, defined, for $Y' \in \mathcal{P}([Y]_a)$ and $b \in [a, a \oplus \delta]$, by

$$\alpha_{a,\delta}(Y') = a \oplus \delta_{Y'} \qquad \gamma_{a,\delta}(b) = \{y \in [Y]_a \mid b(y) \ominus a(y) \sqsupseteq \delta\}.$$

Lemma 3.2.2 (well-definedness)**.** *The functions $\alpha_{a,\delta}$, $\gamma_{a,\delta}$ from Definition 3.2.1 are well-defined and monotone.*

Proof. The involved functions $\alpha_{a,\delta}$ and $\gamma_{a,\delta}$ are well-defined. In fact, for $Y' \subseteq [Y]_a$, clearly $\alpha_{a,\delta} = a \oplus \delta_{Y'} \in [a, a \oplus \delta]$. Moreover, for $b \in [a, a \oplus \delta]$ we have $\gamma_{a,\delta}(b) \subseteq [Y]_a$. In fact, if $y \notin [Y]_a$ then $a(y) = 1$, hence $b(y) = 1$ and thus $b(y) \ominus a(y) = 0 \not\sqsupseteq \delta$, and thus $y \notin \gamma_{a,\delta}(b)$. Moreover, they are clearly monotone. □

When δ is sufficiently small, the pair $\langle \alpha_{a,\delta}, \gamma_{a,\delta} \rangle$ is a Galois connection.

Lemma 3.2.3 (Galois connection)**.** *Let $\mathbb{M}$ be an MV-algebra and Y be a finite set. For $0 \neq \delta \sqsubseteq \delta_a$, the pair $\langle \alpha_{a,\delta}, \gamma_{a,\delta} \rangle : \mathcal{P}([Y]_a) \to [a, a \oplus \delta]$ is a Galois connection.*

$$\mathcal{P}([Y]_a) \underset{\gamma_{a,\delta}}{\overset{\alpha_{a,\delta}}{\rightleftarrows}} [a, a \oplus \delta]$$

Proof. For all $Y' \in \mathcal{P}([Y]_a)$ it holds

$$\gamma_{a,\delta}(\alpha_{a,\delta}(Y')) = \gamma_{a,\delta}(a \oplus \delta_{Y'}) = Y'.$$

In fact, for all $y \in Y'$, $(a \oplus \delta_{Y'})(y) = a(y) \oplus \delta$. Moreover, and by the choice of δ and definition of $[Y]_a$, we have $\delta \sqsubseteq \delta_a \sqsubseteq \overline{a(y)}$, by Lemma 2.3.18(9), we have $(a \oplus \delta_{Y'})(y) \ominus a(y) = \delta$ hence $y \in \gamma_{a,\delta}(\alpha_{a,\delta}(Y'))$. Conversely, if $y \notin Y'$, then $(a \oplus \delta_{Y'})(y) = a(y)$, and thus $(a \oplus \delta_{Y'})(y) \ominus a(y) = 0 \not\sqsupseteq \delta$.

Moreover, for all $b \in [a, a \oplus \delta]$ we have

$$\alpha_{a,\delta}(\gamma_{a,\delta}(b)) = a \oplus \delta_{\gamma_{a,\delta}(b)} \sqsubseteq b$$

In fact, for all $y \in Y$, if $y \in \gamma_{a,\delta}(b)$, i.e., $\delta \sqsubseteq b(y) \ominus a(y)$ then $(a \oplus \delta_{\gamma_{a,\delta}(b)})(y) = a(y) \oplus \delta \sqsubseteq a(y) \oplus (b(y) \ominus a(y)) = b(y)$, by Lemma 2.3.18(2). If instead, $y \notin \gamma_{a,\delta}(b)$, then $(a \oplus \delta_{\gamma_{a,\delta}(b)}(b))(y) = a(y) \sqsubseteq b(y)$. □

Observe that differently from what normally happens in abstract interpretation, the component α of the Galois connection, i.e., the left adjoint, transforms abstract values (sets) into concrete ones (functions) and thus it plays the role of a concretisation function.

Example 3.2.4. *We illustrate the definitions with a small example whose sole purpose is to get a better intuition. (See Figure 3.2 for a visual representation.) Consider the MV-chain $\mathbb{M} = [0,1]$, a set $Y = \{y_1, y_2, y_3, y_4\}$ and a function $a\colon Y \to [0,1]$ with $a(y_1) = 0.2$, $a(y_2) = 0.4$, $a(y_3) = 0.9$, $a(y_4) = 1$. In this case $[Y]_a = \{y_1, y_2, y_3\}$ and $\delta_a = 0.1$.*

Choose $\delta = 0.1$ and $Y' = \{y_1, y_3\}$. Then $\alpha_{a,\delta}(Y')$ is a function that maps $y_1 \mapsto 0.3$, $y_2 \mapsto 0.4$, $y_3 \mapsto 1$, $y_4 \mapsto 1$.

We keep $\delta = 0.1$ and consider a function $b\colon Y \to [0,1]$ with $b(y_1) = 0.3$, $b(y_2) = 0.45$, $b(y_3) = b(y_4) = 1$. Then $\gamma_{a,\delta}(b) = \{y_1, y_3\}$.

Whenever f is non-expansive, it is easy to see that it restricts to a function $f : [a, a \oplus \delta] \to [f(a), f(a) \oplus \delta]$ for all $\delta \in \mathbb{M}$.

Lemma 3.2.5 (restricting non-expansive functions to intervals)**.** *Let $\mathbb{M}$ be an MV-chain, let Y, Z be finite sets $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Then f restricts to a function $f_{a,\delta} : [a, a \oplus \delta] \to [f(a), f(a) \oplus \delta]$, defined by $f_{a,\delta}(b) = f(b)$.*

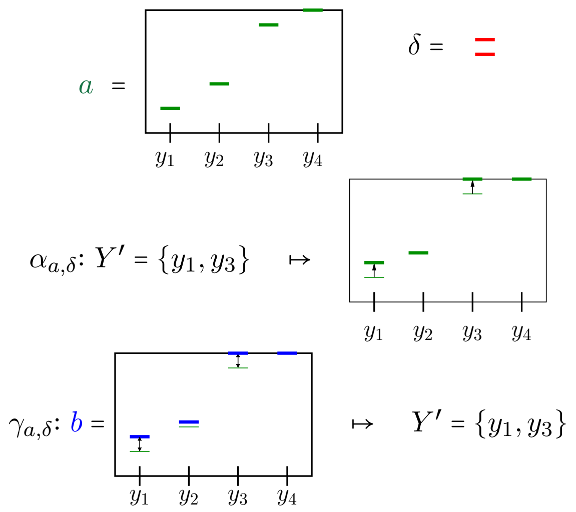


Fig. 3.2.: Visual representation of $\alpha_{a,\delta}$ and $\gamma_{a,\delta}$

Proof. Given $b \in [a, a \oplus \delta]$, by monotonicity of f we have that $f(a) \sqsubseteq f(b)$. Moreover, $f(b) \sqsubseteq f(a \oplus \delta) \sqsubseteq f(a) \oplus \delta$, where the last passage is motivated by Lemma 2.3.25. □

In the following we will simply write f instead of $f_{a,\delta}$.

As mentioned before, a crucial result shows that for all non-expansive functions, under the assumption that Y, Z are finite and the order on $\mathbb{M}$ is total, we can suitably approximate the propagation of increases. In order to state this result, a useful tool is a notion of approximation of a function.

Definition 3.2.6 ((δ, a)-approximation)**.** *Let $\mathbb{M}$ be an MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. For $a \in \mathbb{M}^Y$ and any $\delta \in \mathbb{M}$ we define $f^{\#}_{a,\delta} : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{f(a)})$ as $f^{\#}_{a,\delta} = \gamma_{f(a),\delta} \circ f \circ \alpha_{a,\delta}$.*

Given $Y' \subseteq [Y]_a$, its image $f^{\#}_{a,\delta}(Y') \subseteq [Z]_{f(a)}$ is the set of points $z \in [Z]_{f(a)}$ such that $\delta \sqsubseteq f(a \oplus \delta_{Y'})(z) \ominus f(a)(z)$, i.e., the points to which f propagates an increase of the function a with value δ on the subset Y'.

Example 3.2.7. *We continue with Example 3.2.4 and consider the function $f\colon [0,1]^Y \to [0,1]^Y$ with $f(b) = b \ominus 0.3$ for every $b \in [0,1]^Y$, which is non-expansive (proven in Proposition 3.5.4). We again consider $a\colon Y \to [0,1]$ and $\delta = 0.1$ as in Example 3.2.4, and $Y' = \{y_1, y_2, y_3\}$. The maps a, $\alpha_{a,\delta}(Y')$, $f(a)$ and $f(\alpha_{a,\delta}(Y'))$ are given in the table below and we obtain $f^{\#}_{a,\delta}(Y') = \gamma_{f(a),\delta}(f(\alpha_{a,\delta}(Y'))) = \{y_2, y_3\}$, that is only the increase at y_2 and y_3 can be propagated, while the value of y_1 is too low and y_4 is not even contained in $[Y]_a$ (the domain of $f^{\#}_{a,\delta}$), since its value is already* 1.0 *and there is no slack left. That is, we obtain those elements of Y for which the last two lines in the table below differ by* 0.1*.*

	y_1	y_2	y_3	y_4
a	0.2	0.4	0.9	1.0
$\alpha_{a,\delta}(Y')$	0.3	0.5	1.0	1.0
$f(a)$	0.0	0.1	0.6	0.7
$f(\alpha_{a,\delta}(Y'))$	0.0	0.2	0.7	0.7

In general we have $f^{\#}_{a,\delta}(Y') = Y' \cap \{y_2, y_3\}$ *if* $\delta \leq \delta_a = 0.1$, $f^{\#}_{a,\delta}(Y') = Y' \cap \{y_2\}$ *if* $0.1 < \delta \leq 0.6$ *and* $f^{\#}_{a,\delta}(Y') = \emptyset$ *if* $0.6 < \delta$.

We now show that $f^{\#}_{a,\delta}$ is antitone in the parameter δ, a non-trivial result.

Lemma 3.2.8 (antitonicity)**.** *Let* $\mathbb{M}$ *be an MV-chain, let* Y, Z *be finite sets, let* $f : \mathbb{M}^Y \to \mathbb{M}^Z$ *be a non-expansive function and let* $a \in \mathbb{M}^Y$. *For* $\theta, \delta \in \mathbb{M}$, *if* $\theta \sqsubseteq \delta$ *then* $f^{\#}_{a,\delta} \subseteq f^{\#}_{a,\theta}$.

Proof. Let $Y' \subseteq [Y]_a$ and let us prove that $f^{\#}_{a,\delta}(Y') \subseteq f^{\#}_{a,\theta}(Y')$. Take $z \in f^{\#}_{a,\delta}(Y')$. This means that $\delta \sqsubseteq f(a \oplus \delta_{Y'})(z) \ominus f(a)(z)$.

We have

$$
\begin{aligned}
&\delta \sqsubseteq f(a \oplus \delta_{Y'})(z) \ominus f(a)(z) \\
&\quad \text{[by hypothesis]} \\
&= f(a \oplus \theta_{Y'} \oplus (\delta \ominus \theta)_{Y'})(z) \ominus f(a)(z) \\
&= f(a \oplus \theta_{Y'} \oplus (\delta \ominus \theta)_{Y'})(z) \ominus f(a \oplus \theta_{Y'})(z) \oplus f(a \oplus \theta_{Y'})(z) \ominus f(a)(z) \\
&\sqsubseteq \| f(a \oplus \theta_{Y'} \oplus (\delta \ominus \theta)_{Y'}) \ominus f(a \oplus \theta_{Y'}) \| \oplus f(a \oplus \theta_{Y'})(z) \ominus f(a)(z) \\
&\quad \text{[by definition of norm and monotonicity of } \oplus\text{]} \\
&\sqsubseteq \| a \oplus \theta_{Y'} \oplus (\delta \ominus \theta)_{Y'} \ominus (a \oplus \theta_{Y'}) \| \oplus f(a \oplus \theta_{Y'})(z) \ominus f(a)(z) \\
&\quad \text{[by non-expansiveness of } f \text{ and monotonicity of } \oplus\text{]} \\
&\sqsubseteq \| (\delta \ominus \theta)_{Y'} \| \oplus f(a \oplus \theta_{Y'})(z) \ominus f(a)(z) \\
&\sqsubseteq (\delta \ominus \theta) \oplus f(a \oplus \theta_{Y'})(z) \ominus f(a)(z) \\
&\quad \text{[by definition of norm]}
\end{aligned}
$$

If we subtract $\delta \ominus \theta$ on both sides, we get $\delta \ominus (\delta \ominus \theta) \sqsubseteq f(a \oplus \theta_{Y'})(z) \ominus f(a)(z)$, and, as above, since, by Lemma 2.3.18(10), $\delta \ominus (\delta \ominus \theta) = \theta$ we conclude

$$\theta \sqsubseteq f(a \oplus \theta_{Y'})(z) \ominus f(a)(z)$$

which means $z \in f^{\#}_{a,\theta}(Y')$. □

Since $f^{\#}_{a,\delta}$ increases when δ decreases and there are only finitely many such functions, there must be a value ι^f_a such that all functions $f^{\#}_{a,\delta}$ for $0 \sqsubset \delta \sqsubseteq \iota^f_a$ are equal. The resulting function will be the approximation of interest.

We next show how ι^f_a can be determined. We start by observing that for each $z \in [Z]_{f(a)}$ and $Y' \subseteq [Y]_a$ there is a largest increase θ such that $z \in f^\#_{a,\theta}(Y')$.

Lemma 3.2.9 (largest increase for a point)**.** *Let $\mathbb{M}$ be a complete MV-chain, let Y, Z be finite sets, let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function and fix $a \in \mathbb{M}^Y$. For all $z \in [Z]_{f(a)}$ and $Y' \subseteq [Y]_a$ the set $\{\theta \in \mathbb{M} \mid z \in f^\#_{a,\theta}(Y')\}$ has a maximum, that we denote by $\iota^f_a(Y', z)$.*

Proof. Let $V = \{\theta \in \mathbb{M} \mid z \in f^\#_{a,\theta}(Y')\}$. Expanding the definition we have that

$$V = \{\theta \in \mathbb{M} \mid \theta \sqsubseteq f(a \oplus \theta_{Y'})(z) \ominus f(a)(z)\}.$$

If we let $\eta = \sup V$, for all $\theta \in V$, since $\theta_{Y'} \sqsubseteq \eta_{Y'}$, clearly, by monotonicity

$$\theta \sqsubseteq f(a \oplus \eta_{Y'})(z) \ominus f(a)(z)$$

and therefore, by definition of supremum, $\eta \sqsubseteq f(a \oplus \eta_{Y'})(z) \ominus f(a)(z)$, i.e., $\eta \in V$ is a maximum, as desired. □

We can then provide an explicit definition of ι^f_a and of the approximation of a function.

Lemma 3.2.10 (a-approximation for a function)**.** *Let $\mathbb{M}$ be a complete MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Let*

$$\iota^f_a = \min\{\iota^f_a(Y', z) \mid Y' \subseteq [Y]_a \ \wedge\ z \in [Z]_{f(a)} \ \wedge\ \iota^f_a(Y', z) \neq 0\} \cup \{\delta_a\}.$$

Then for all $0 \neq \delta \sqsubseteq \iota^f_a$ it holds that $f^\#_{a,\delta} = f^\#_{a,\iota^f_a}$.

*The function $f^\#_{a,\iota^f_a}$ is called the **a-approximation** of f and it is denoted by $f^\#_a$.*

*We might refer to ι^f_a as the **ascent constant**.*

Proof. Since $\delta \sqsubseteq \iota^f_a$, by Lemma 3.2.8 we have $f^\#_{a,\delta} \supseteq f^\#_{a,\iota^f_a}$. For the other inclusion let $Y' \subseteq [Y]_a$. We have

$$f^\#_{a,\delta}(Y') = \{z \in [Z]_{f(a)} \mid f(a \oplus \delta_{Y'})(z) \ominus f(a)(z) \sqsupseteq \delta\}$$

by definition. Assume that there exists $z \in f^\#_{a,\delta}(Y')$ where $f(a \oplus (\iota^f_a)_{Y'})(z) \ominus f(a)(z) \not\sqsupseteq \iota^f_a$. But this is a contradiction, since ι^f_a is the minimum of all such non-zero values. □

In the following, we show that indeed, for all non-expansive functions, the a-approximation properly approximates the propagation of increases. Given an MV-chain $\mathbb{M}$ and a finite set Y, we first observe that each function $b \in \mathbb{M}^Y$ can be expressed as a suitable sum of functions of the shape $\delta_{Y'}$.

Lemma 3.2.11 (standard form)**.** *Let $\mathbb{M}$ be an MV-chain and let Y be a finite set. Then for any $b \in \mathbb{M}^Y$ there are $Y_1, \dots, Y_n \subseteq Y$ with $Y_{i+1} \subseteq Y_i$ for $i \in \{1, \dots, n-1\}$ and $\delta^i \in \mathbb{M}$, $0 \neq \delta^i \sqsubseteq \overline{\bigoplus_{j=1}^{i-1} \delta^j}$ for $i \in \{1, \dots, n\}$ such that*

$$b = \bigoplus_{i=1}^{n} \delta^i_{Y_i} \quad \text{and} \quad \|b\| = \bigoplus_{i=1}^{n} \delta^i.$$

where we assume that an empty sum evaluates to 0.

Proof. See Appendix: Lemma A.2.1. ☐

The above characterisation allows us to show a technical property of the functions in the interval $[a, a \oplus \delta]$ of interest.

Lemma 3.2.12. *Let $\mathbb{M}$ be an MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Let $a \in \mathbb{M}^Y$. For $b \in [a, a \oplus \delta]$, let $b \ominus a = \bigoplus_{i=1}^{n} \delta^i_{Y_i}$ be a standard form for $b \ominus a$. If $\gamma_{f(a),\delta}(f(b)) \neq \emptyset$ then $Y_n = \gamma_{a,\delta}(b)$ and $\gamma_{f(a),\delta}(f(b)) \subseteq f^{\#}_{a,\delta^n}(Y_n)$.*

Proof. See Appendix: Lemma A.2.2. ☐

We can finally prove the main result about legitimacy of the approximation.

Theorem 3.2.13 (approximation of non-expansive functions)**.** *Let $\mathbb{M}$ be a complete MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Then for all $0 \sqsubset \delta \in \mathbb{M}$:*

a. $\gamma_{f(a),\delta} \circ f \subseteq f^{\#}_a \circ \gamma_{a,\delta}$

b. for $\delta \sqsubseteq \delta_a$*:* $\delta \sqsubseteq \iota^f_a$ *iff* $\gamma_{f(a),\delta} \circ f = f^{\#}_a \circ \gamma_{a,\delta}$

$$\begin{array}{ccc} [a, a \oplus \delta] & \xrightarrow{\gamma_{a,\delta}} & \mathcal{P}([Y]_a) \\ {\scriptstyle f}\downarrow & \subseteq & \downarrow{\scriptstyle f^{\#}_a} \\ [f(a), f(a) \oplus \delta] & \xrightarrow[\gamma_{f(a),\delta}]{} & \mathcal{P}([Z]_{f(a)}) \end{array}$$

Proof. a. Let $b \in [a, a \oplus \delta]$. First, note that whenever $\gamma_{f(a),\delta}(f(b)) = \emptyset$, the desired inclusion obviously holds.

If instead $\gamma_{f(a),\delta}(f(b)) \neq \emptyset$, let $b \ominus a = \bigoplus_{i=1}^{n} \delta^i_{Y_i}$ be a standard form with $\delta^n \neq 0$. First observe that, by Lemma 3.2.12, we have $Y_n = \gamma_{a,\delta^n}(b)$ and

$$\gamma_{f(a),\delta}(f(b)) \subseteq f^{\#}_{a,\delta^n}(Y_n). \tag{3.1}$$

For all $z \in f^{\#}_{a,\delta^n}(Y_n)$, by definition of $\iota^f_a(Y_n,z)$ we have that $0 \sqsubset \delta_n \sqsubseteq \iota^f_a(Y_n,z)$, therefore $\iota^f_a \sqsubseteq \iota^f_a(Y_n,z)$. Moreover, $z \in f^{\#}_{a,\iota^f_a(Y_n,z)}(Y_n) \subseteq f^{\#}_{a,\iota^f_a}(Y_n) = f^{\#}_a(Y_n)$, where the last inequality is motivated by Lemma 3.2.8 since $\iota^f_a \sqsubseteq \iota^f_a(Y_n,z)$. Therefore, $f^{\#}_{a,\delta^n}(Y_n) \subseteq f^{\#}_a(\gamma_{a,\delta}(b))$, which combined with (3.1) gives the desired result.

b. For (b), we first show the direction from left to right. Assume that $\delta \sqsubseteq \iota^f_a$. By (a) clearly, $\gamma_{f(a),\delta} \circ f(b) \subseteq f^{\#}_a \circ \gamma_{a,\delta}(b)$. For the converse inclusion, note that:

$$\begin{aligned}
&f^{\#}_a(\gamma_{a,\delta}(b)) && [\text{by definition of } f^{\#}_a]\\
&\quad = f^{\#}_{a,\iota^f_a}(\gamma_{a,\delta}(b)) && [\text{by Lemma 3.2.8, since } \delta \sqsubseteq \iota^f_a]\\
&\quad \subseteq f^{\#}_{a,\delta}(\gamma_{a,\delta}(b)) && [\text{by definition of } f^{\#}_{a,\delta}]\\
&\quad = \gamma_{f(a),\delta}(f(\alpha_{a,\delta}(\gamma_{a,\delta}(b)))) && [\text{since } \alpha_{a,\delta} \circ \gamma_{a,\delta}(b) \sqsubseteq b]\\
&\quad \subseteq \gamma_{f(a),\delta}(f(b))
\end{aligned}$$

as desired.

For the other direction, assume $\gamma_{f(a),\delta} \circ f(b) = f^{\#}_a \circ \gamma_{a,\delta}(b)$ holds for all $b \in [a, a \oplus \delta]$. Now, for every $Y' \subseteq [Y]_a$ we have $f^{\#}_{a,\delta}(Y') = \gamma_{f(a),\delta} \circ f \circ \alpha_{a,\delta}(Y') = f^{\#}_a \circ \gamma_{a,\delta} \circ \alpha_{a,\delta}(Y')$. We also have $\gamma_{a,\delta} \circ \alpha_{a,\delta}(Y') = Y'$ (see proof of Lemma 3.2.3), thus $f^{\#}_{a,\delta}(Y') = f^{\#}_a(Y')$. For any δ with $\iota^f_a \sqsubset \delta \sqsubseteq \delta_a$ there exists $Y' \subseteq [Y]_a$ and $z \in [Z]_{f(a)}$ with $z \in f^{\#}_a(Y')$ but $z \notin f^{\#}_{a,\delta}(Y')$, by definition of ι^f_a. Therefore $\delta \sqsubseteq \iota^f_a$ has to hold.

□

Note that if $Y = Z$ and a is a fixpoint of f, i.e., $a = f(a)$, then condition (a) above corresponds exactly to soundness in the sense of abstract interpretation [CC77]. Moreover, when $\delta \sqsubseteq \delta_a$ and thus $\langle \alpha_{a,\delta}, \gamma_{a,\delta} \rangle$ is a Galois connection, $f^{\#}_{a,\delta} = \gamma_{a,\delta} \circ f \circ \alpha_{a,\delta}$ is the best correct approximation of f. In particular, when $\delta \sqsubseteq \iota^f_a$, such a best correct approximation is $f^{\#}_a$, the a-approximation of f, i.e., it becomes independent from δ, and condition (b) corresponds to (γ-)completeness [GRS00] (see also Section 2.3.2).

3.3. Proof Rules

In this section we formalise the proof technique outlined in the introduction for showing that a fixpoint is the largest and, more generally, for checking over-approximations of greatest fixpoints of non-expansive functions.

3.3.1. Proof Rules for Fixpoints

Consider a monotone function $f : \mathbb{M}^Y \to \mathbb{M}^Y$ for some finite set Y. We first focus on the problem of establishing whether some given fixpoint a of f coincides with νf

(without explicitly knowing νf), and, in case it does not, finding an “improvement”, i.e., a post-fixpoint of f, larger than a. To this aim we need a technical lemma.

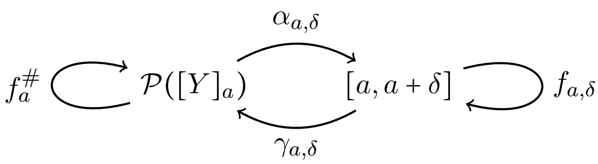
Lemma 3.3.1. *Let $\mathbb{M}$ be a complete MV-chain, Y a finite set and $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a non-expansive function. Let $a \in \mathbb{M}^Y$ be a pre-fixpoint of f (i.e., $f(a) \sqsubseteq a$), let $f_a^\# : \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_{f(a)})$ be the a-approximation of f. Assume $\nu f \not\sqsubseteq a$ and let $Y' = \{y \in [Y]_a \mid \nu f(y) \ominus a(y) = \|\nu f \ominus a\|\}$. Then for all $y \in Y'$ it holds $a(y) = f(a)(y)$ and $Y' \subseteq f_a^\#(Y')$.*

Proof. See Appendix: Lemma A.2.3. □

Observe that, when a is a fixpoint, clearly $[Y]_a = [Y]_{f(a)}$, and thus the a-approximation of f (Lemma 3.2.10) is an endo-function $f_a^\# : [Y]_a \to [Y]_a$ and Y' is its post-fixpoint as in Lemma 3.3.1. Then, we have the following result, which relies on the fact that, due to Theorem 3.2.13 and properties of Galois connections, $\gamma_{a,\delta}$ maps the greatest fixpoint of f to the greatest fixpoint of $f_a^\#$.

Theorem 3.3.2 (soundness and completeness for fixpoints)**.** *Let $\mathbb{M}$ be a complete MV-chain, Y a finite set and $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function. Let $a \in \mathbb{M}^Y$ be a fixpoint of f. Then $\nu f_a^\# = \emptyset$ if and only if $a = \nu f$.*

Proof. Let a be a fixpoint of f and assume that $a = \nu f$. For $\delta = \iota_a^f \sqsubseteq \delta_a$, according to Lemma 3.2.3, we have a Galois connection:



Since a is a fixpoint, then $[Y]_{f(a)} = [Y]_a$ and, by Theorem 3.2.13(b), $\gamma_{a,\delta} \circ f = \gamma_{f(a),\delta} \circ f = f_a^\# \circ \gamma_{a,\delta}$.

Therefore by [CC00, Proposition 14], $\nu f_a^\# = \gamma_{a,\delta}(\nu f)$. Recall that $\gamma_{a,\delta}(\nu f) = \{y \in Y \mid \delta \sqsubseteq \nu f(y) \ominus a(y)\}$. Since $a = \nu f$ and $\delta \sqsupset 0$, we know that $\gamma_{a,\delta}(\nu f) = \emptyset$ and we conclude $\nu f_a^\# = \emptyset$, as desired.

Conversely, in order to prove that if $\nu f_a^\# = \emptyset$ then $a = \nu f$, we prove the contrapositive. Assume that $a \neq \nu f$. Since a is a fixpoint and νf is the largest, this means that $a \sqsubset \nu f$ and thus $\|\nu f \ominus a\| \neq 0$. Consider $Y' = \{y \in [Y]_a \mid \nu f(y) \ominus a(y) = \|\nu f \ominus a\|\} \neq \emptyset$. By Lemma 3.3.1, Y' is a post-fixpoint of $f_a^\#$, i.e., $Y' \subseteq f_a^\#(Y')$, and thus $\nu f_a^\# \supseteq Y'$ which implies $\nu f_a^\# \neq \emptyset$, as desired. □

Whenever a is a fixpoint, but not yet the largest fixpoint of f, from the result above $\nu f_a^\# \neq \emptyset$. Intuitively, $\nu f_a^\#$ is the set of points where a can still be "improved". More precisely, we can show that a can be increased on the points in $\nu f_a^\#$ producing a post-fixpoint of f. In order to determine how much a can be increased we proceed similarly to what we have done for defining ι_a^f (Lemma 3.2.10), but restricting the attention to $\nu f_a^\#$ instead of considering the full $[Y]_a$. While ι_a^f could always be used, by restricting to $\nu f_a^\#$ we are able to find a potentially better, that is larger, value which is still correct.

Definition 3.3.3 (largest increase for a subset)**.** *Let $\mathbb{M}$ be a complete MV-chain and let $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function, where Y is a finite set and let $a \in \mathbb{M}^Y$. For $Y' \subseteq Y$, we define $\delta_a(Y') = \min\{\overline{a(y)} \mid y \in Y'\}$ and $\iota_a^f(Y') = \min\{\iota_a^f(Y', y) \mid y \in Y'\}$.*

Example 3.3.4. *We intuitively explain the computation of the values in the definition above. Let $g\colon [0,1]^Y \to [0,1]^Y$ with $g(b) = b \oplus 0.1$, which is non-expansive (see Proposition 3.5.4). The set Y and the function $a \in [0,1]^Y$ are as in Example 3.2.4 ($Y = \{y_1, y_2, y_3, y_4\}$ and $a(y_1) = 0.2$, $a(y_2) = 0.4$, $a(y_3) = 0.9$, $a(y_4) = 1$).*

Let $Y' = \{y_1, y_2\}$. Then $\delta_a(Y') = 0.6$ and $\iota_a^g(Y') = 0.5$, i.e., since g adds 0.1, we can propagate an increase of at most 0.5.

We next prove that when $a \in \mathbb{M}^Y$ is a fixpoint of f and $Y' = \nu f_a^\#$, the value $\iota_a^f(Y')$ is the largest increase δ below $\delta_a(Y')$ such that $a \oplus \delta_{Y'}$ is a post-fixpoint of f.

Proposition 3.3.5 (from a fixpoint to larger post-fixpoint)**.** *Let $\mathbb{M}$ be a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a non-expansive function, $a \in \mathbb{M}$ a fixpoint of f, and let $Y' = \nu f_a^\#$ be the greatest fixpoint of the corresponding a-approximation. Then $\iota_a^f \sqsubseteq \iota_a^f(Y') \sqsubseteq \delta_a(Y')$. Moreover, for all $\theta \sqsubseteq \iota_a^f(Y')$ the function $a \oplus \theta_{Y'}$ is a post-fixpoint of f, while for $\iota_a^f(Y') \sqsubset \theta \sqsubseteq \delta_a(Y')$ it is not.*

Proof. We first show that $\iota_a^f \sqsubseteq \iota_a^f(Y')$. By Lemma 3.2.10 and since $a = f(a)$, we have that $\iota_a^f = \min\{\iota_a^f(Y'', y) \mid Y'' \subseteq [Y]_a \ \wedge\ y \in [Y]_a \ \wedge\ \iota_a^f(Y'', y) \neq 0\} \cup \{\delta_a\}$. Moreover, we have $Y' = \nu f_a^\# \subseteq [Y]_a$ and $\iota_a^f(Y', y) \neq 0$, for every $y \in Y'$, since $\iota_a^f(Y', y) = \max\{\delta \in \mathbb{M} \mid y \in f_{a,\delta}^\#(Y')\}$ and $y \in Y' = \nu f_a^\# = f_a^\#(\nu f_a^\#) = f_{a,\iota_a^f}^\#(Y')$, hence $\iota_a^f(Y', y) \sqsupseteq \iota_a^f \sqsupset 0$. Therefore, the minimum in $\iota_a^f(Y')$ is computed on a subset of the values on which the one in ι_a^f is, and so the former must be larger or equal to the latter.

Next, we prove that $\iota_a^f(Y') \sqsubseteq \delta_a(Y')$. Observe that for all $y \in Y'$ and $\delta \in \mathbb{M}$, if $y \in f_{a,\delta}^\#(Y')$, by definition of $f_{a,\delta}^\#$, it holds that $\delta \sqsubseteq f(a \oplus \delta_{Y'})(y) \ominus f(a)(y) = f(a \oplus \delta_{Y'})(y) \ominus a(y) \sqsubseteq 1 \ominus a(y) = \overline{a(y)}$, where the second equality is motivated by the fact that a is a fixpoint. Therefore for all $y \in Y'$ we have $\max\{\delta \in \mathbb{M} \mid y \in f_{a,\delta}^\#(Y')\} \sqsubseteq \overline{a(y)}$ and thus $\iota_a^f(Y') = \min_{y \in Y'} \max\{\delta \in \mathbb{M} \mid y \in f_{a,\delta}^\#(Y')\} \sqsubseteq \min_{y \in Y'} \overline{a(y)} = \delta_a(Y')$, as desired.

Given $\theta \sqsubseteq \iota_a^f(Y')$, let us prove that $a \oplus \theta_{Y'}$ is a post-fixpoint of f, i.e., $a \oplus \theta_{Y'} \sqsubseteq f(a \oplus \theta_{Y'})$.

If $y \in Y'$, since $\theta \sqsubseteq \iota_a^f(Y')$, by definition of $\iota_a^f(Y')$, we have $\theta \sqsubseteq \max\{\delta \in \mathbb{M} \mid y \in f^\#_{a,\delta}(Y')\}$ and thus, by antitonicity of $f^\#_{a,\delta}$ with respect to δ, we have $y \in f^\#_{a,\theta}(Y')$. This means that $\theta \sqsubseteq f(a \oplus \theta_{Y'})(y) \ominus f(a)(y) = f(a \oplus \theta_{Y'})(y) \ominus a(y)$, where the last passage uses the fact that a is a fixpoint. Adding $a(y)$ on both sides and using Lemma 2.3.18(2), we obtain $a(y) \oplus \theta \sqsubseteq (f(a \oplus \theta_{Y'})(y) \ominus a(y)) \oplus a(y) = f(a \oplus \theta_{Y'})(y)$. Since $y \in Y'$, $(a \oplus \theta_{Y'})(y) = a(y) \oplus \theta$ and thus $(a \oplus \theta_{Y'})(y) \sqsubseteq f(a \oplus \theta_{Y'})(y)$, as desired.

If instead, $y \notin Y'$, clearly $(a \oplus \theta_{Y'})(y) = a(y) = f(a)(y) \sqsubseteq f(a \oplus \theta_{Y'})(y)$, where we again use the fact that a is a fixpoint and monotonicity of f.

Lastly, we have to show that if $\iota_a^f(Y') \sqsubset \theta \sqsubseteq \delta_a(Y')$, then $a \oplus \theta_{Y'}$ is not a post-fixpoint of f. By definition of $\iota_a^f(Y')$, from the fact that $\iota_a^f(Y') \sqsubset \theta$, we deduce that $\max\{\delta \in \mathbb{M} \mid y \in f^\#_{a,\delta}(Y')\} \sqsubset \theta$ for some $y \in Y'$ and thus $y \notin f^\#_{a,\theta}(Y')$.

By definition of $f^\#_{a,\theta}$ and totality of $\sqsubseteq$, the above means $\theta \sqsupset f(a \oplus \theta_{Y'})(y) \ominus f(a)(y) = f(a \oplus \theta_{Y'})(y) \ominus a(y)$, since a is a fixpoint of f. Since $\theta \sqsubseteq \delta_a(Y')$, we can add $a(y)$ on both sides and, by Lemma 2.3.18(8), we obtain $a(y) \oplus \theta \sqsupset f(a \oplus \theta_{Y'})(y)$. Since $y \in Y'$, the left-hand side is $(a \oplus \theta_{Y'})(y)$. Hence we conclude that indeed $a \oplus \theta_{Y'}$ is not a post fixpoint. □

Using these results one can perform an alternative fixpoint iteration where we iterate to the largest fixpoint from below: start with a post-fixpoint $a_0 \sqsubseteq f(a_0)$ (which is clearly below νf) and obtain, by (possibly transfinite) iteration, an ascending chain that in the order converges[2] to a, the least fixpoint above a_0. Now, letting $Y' = \nu f_a^\#$, check whether $Y' = \emptyset$. If so, by Theorem 3.3.2 we know we have reached $\nu f = a$. If not, $\alpha_{a,\iota_a^f(Y')}(Y') = a \oplus (\iota_a^f(Y'))_{Y'}$ is again a post-fixpoint (cf. Proposition 3.3.5) and we continue this procedure until – for some ordinal – we reach the largest fixpoint νf, for which we have $\nu f^\#_{\nu f} = \emptyset$. We will encounter such a procedure in Chapter 5.

In order to make the above procedure as efficient as possible, one would like to consider, whenever a fixpoint a is reached, the largest possible increase ι which is valid, i.e. such that $a \oplus \iota$ is again a post-fixpoint of f. Thus the question naturally arises asking whether $\iota_a^f(Y')$ is such largest valid increase. From Proposition 3.3.5, it immediately follows that $\iota_a^f(Y')$ is the largest valid increase below $\delta_a(Y')$, but it can be seen that there can be larger valid increases above $\delta_a(Y')$ (an explicit example is provided later in Example 3.6.5, for the dual case of least fixpoints). However, while the set of valid increases below $\delta_a(Y')$ is downward-closed, as proved in Proposition 3.3.5, this is not the case for those above $\delta_a(Y')$. Hence, we believe that the most efficient approach would be to search for $\iota_a^f(Y')$, or some satisfying approximation, via a binary search bounded by $\delta_a(Y')$.

Remark 3.3.6. *We note that for the strategy iterations we will present in Chapter 5 it is less relevant what increase we choose as long as it is sufficiently small. This is the case since any increase will lead to a new strategy. However, even in this instance, a larger increase is to be preferred.*

[2] Convergence in the natural order.

3.3.2. Proof Rules for Pre-fixpoints

Interestingly, the soundness result in Theorem 3.3.2 can be generalised to the case in which a is a pre-fixpoint instead of a fixpoint. In this case, the a-approximation for a function $f : \mathbb{M}^Y \to \mathbb{M}^Y$ is a function $f_a^\# : [Y]_a \to [Y]_{f(a)}$ where domain and codomain are different, hence it would not be meaningful to look for fixpoints. However, as explained below, it can be restricted to an endo-function.

Theorem 3.3.7 (soundness for pre-fixpoints)**.** *Let $\mathbb{M}$ be a complete MV-chain, Y a finite set and $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function. Given a pre-fixpoint $a \in \mathbb{M}^Y$ of f, let $[Y]_{a=f(a)} = \{y \in [Y]_a \mid a(y) = f(a)(y)\}$. Let us define $f_a^* : [Y]_{a=f(a)} \to [Y]_{a=f(a)}$ as $f_a^*(Y') = f_a^\#(Y') \cap [Y]_{a=f(a)}$, where $f_a^\# : \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_{f(a)})$ is the a-approximation of f. If $\nu f_a^* = \emptyset$ then $\nu f \sqsubseteq a$.*

Proof. We prove the contrapositive, i.e., we show that $\nu f \not\sqsubseteq a$ allows us to derive that $\nu f_a^* \neq \emptyset$.

Assume $\nu f \not\sqsubseteq a$, i.e., there exists $y \in Y$ such that $\nu f(y) \not\sqsubseteq a(y)$. Since the order is total, this means that $a(y) \sqsubset \nu f(y)$. Hence, by Lemma 2.3.18(5), $\nu f(y) \ominus a(y) \sqsupset 0$. Then $\delta = \|\nu f \ominus a\| \sqsupset 0$.

Consider $Y' = \{y \in Y_a \mid \nu f(y) \ominus a(y) = \|\nu f \ominus a\|\} \neq \emptyset$. By Lemma 3.3.1, Y' is a post-fixpoint of $f_a^\#$, i.e., $Y' \subseteq f_a^\#(Y')$, and thus $Y' \subseteq \nu f_a^\#$. Moreover, for all $y \in Y'$, $a(y) = f(a)(y)$, i.e., $Y' \subseteq [Y]_{a=f(a)}$. Therefore we conclude $Y' \subseteq f_a^\#(Y') \cap [Y]_{a=f(a)} = f_a^*(Y')$, i.e., Y' is a post-fixpoint also for f_a^*, and thus $\nu f_a^* \supseteq Y' \neq \emptyset$, as desired. □

The reason why we can limit our attention to the set of points where $a(y) = f(a)(y)$ is as follows. Observe that, since a is a pre-fixpoint and $\ominus$ is antitone in the second argument, $\nu f \ominus a \sqsubseteq \nu f \ominus f(a)$. Thus $\|\nu f \ominus a\| \sqsubseteq \|\nu f \ominus f(a)\| = \|f(\nu f) \ominus f(a)\| \sqsubseteq \|\nu f \ominus a\|$, where the last passage is motivated by non-expansiveness of f. Therefore $\|\nu f \ominus a\| = \|\nu f \ominus f(a)\|$. From this we can deduce that, if νf is strictly larger than a on some points, surely some of these points are in $[Y]_{a=f(a)}$. In particular, all points y_0 such that $\nu f(y_0) \ominus a(y_0) = \|\nu f \ominus a\|$ are necessarily in $[Y]_{a=f(a)}$. Otherwise, we would have $f(a)(y_0) \sqsubset a(y_0)$ and thus $\|\nu f \ominus a\| = \nu f(y_0) \ominus a(y_0) \sqsubset \nu f(y_0) \ominus f(a)(y_0) \sqsubseteq \|\nu f \ominus f(a)\|$ (cf. Lemma 3.3.1).

Remark 3.3.8. *Completeness does not generalise to pre-fixpoints, i.e., it is not true that if a is a pre-fixpoint of f and $\nu f \sqsubseteq a$, then $\nu f_a^* = \emptyset$. A pre-fixpoint might contain slack even though it is above the greatest fixpoint. A counterexample is in Example 3.6.28.*

3.4. The Dual View for Least Fixpoints

The theory developed so far can be easily dualised to check under-approximations of least fixpoints. Given a complete MV-algebra $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ and a non-expansive function $f : \mathbb{M}^Y \to \mathbb{M}^Y$, in order to show that a post-fixpoint $a \in \mathbb{M}^Y$ is such that $a \sqsubseteq \mu f$ we can in fact simply work in the dual MV-algebra, $\mathbb{M}^{op} = (M, \otimes, 1, \overline{(\cdot)})$ where the natural order is reversed, i.e. if $\sqsubseteq$ is the natural order on $\mathbb{M}$ then $\sqsupseteq$ is the natural order on $\mathbb{M}^{op}$.

Since $\oplus$ could be the "standard" operation on $\mathbb{M}$, it is convenient to formulate the conditions using $\oplus$ and $\ominus$ and the original order. The notation for the dual case is obtained from that of the original case, referred to as the *primal case* throughout the paper, exchanging subscripts and superscripts.

The pair of functions $\langle \alpha^{a,\theta}, \gamma^{a,\theta} \rangle$ is as follows. Let $a : Y \to \mathbb{M}$ and $0 \sqsubset \theta \in \mathbb{M}$. The set $[Y]^a = \{y \in Y \mid a(y) \neq 0\}$ and $\delta^a = \min\{a(y) \mid y \in [Y]^a\}$

The target of the approximation is $[a, a \otimes \theta]$ in the reverse order, hence $[a \otimes \theta, a]$ in the original order. Recall that $a \otimes \theta = \overline{\overline{a} \oplus \overline{\theta}} = a \ominus \overline{\theta}$. Hence we obtain

$$\mathcal{P}([Y]^a) \underset{\gamma^{a,\theta}}{\overset{\alpha^{a,\theta}}{\rightleftarrows}} [a \ominus \overline{\theta}, a]$$

For $Y' \in \mathcal{P}([Y]^a)$ we define

$$\alpha^{a,\theta}(Y') = a \otimes \theta_{Y'} = a \ominus \overline{\theta}_{Y'}$$

Instead $\gamma^{a,\theta}(b) = \{y \in Y \mid \theta \sqsupseteq b(y) \odiv a(y)\}$ where $\odiv$ is the subtraction in the dual MV-algebra. Observe that $x \odiv y = \overline{\overline{x} \otimes y} = \overline{x \oplus \overline{y}} = \overline{y \ominus x}$. Hence $\theta \sqsupseteq b(y) \odiv a(y)$ iff $a(y) \ominus b(y) \sqsupseteq \overline{\theta}$. Thus for $b \in [a \ominus \overline{\theta}, a]$ we have

$$\gamma^{a,\theta}(b) = \{y \in [Y]^a \mid \theta \sqsupseteq b(y) \odiv a(y)\} = \{y \in [Y]^a \mid a(y) \ominus b(y) \sqsupseteq \overline{\theta}\}.$$

Let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a monotone function. The norm becomes $\|a\| = \min\{a(y) \mid y \in Y\}$. Non-expansiveness in the dual MV-algebra becomes: for all $a, b \in \mathbb{M}^Y$, $\|f(b) \odiv f(a)\| \sqsupseteq \|b \odiv a\|$, which in turn is

$$\min\{\overline{f(a) \ominus f(b)} \mid y \in Y\} \sqsupseteq \min\{\overline{a(y) \ominus b(y)} \mid y \in Y\}$$

i.e., $\|f(a) \ominus f(b)\| \sqsubseteq \|a \ominus b\|$, which coincides with non-expansiveness in the original MV-algebra.

Observe that, instead of taking a generic $\theta \sqsubset 1$ and then working with $\bar{\theta}$, we can directly take $0 \sqsubset \theta$ and replace everywhere $\bar{\theta}$ with θ. Thus we obtain $\alpha^{a,\theta} \colon \mathcal{P}([Y]^a) \to [a \ominus \theta, a]$ and $\gamma^{a,\theta} \colon [a \ominus \theta, a] \to \mathcal{P}([Y]^a)$ as

$$\alpha^{a,\theta}(Y') = a \ominus \theta_{Y'} \text{ and } \gamma^{a,\theta}(b) = \{y \in [Y]^a \mid a(y) \ominus b(y) \sqsupseteq \theta\}.$$

While the approximation of a function in the primal case are denoted $f_a^{\#}$, the approximations in the dual case will be denoted by $f_{\#}^a$ which we obtain by dualizing the theory.

Lemma 3.4.1 (a-approximation (dual) for a function)**.** *Let $\mathbb{M}$ be a complete MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Let $a \in \mathbb{M}^Y$, then for all $z \in [Z]^{f(a)}$ and $Y' \subseteq [Y]^a$ the set $\{\theta \in \mathbb{M} \mid z \in f_\#^{a,\theta}(Y')\}$ has a maximum, that we denote by $\iota_f^a(Y', z)$. Let*

$$\iota_f^a = \min\{\iota_f^a(Y', z) \mid Y' \subseteq [Y]^a \ \wedge \ z \in [Z]^{f(a)} \ \wedge \ \iota_f^a(Y', z) \neq 0\} \cup \{\delta^a\}.$$

Then for all $0 \neq \delta \sqsubseteq \iota_f^a$ it holds that $f_\#^{a,\delta} = f_\#^{a,\iota_f^a}$.
*The function $f_\#^{a,\iota_f^a} = \gamma^{f(a),\iota_f^a} \circ f \circ \alpha^{a,\iota_f^a}$ is called the **a-approximation** of f and it is denoted by $f_\#^a$. We might call ι_f^a the **descent constant**.*

Proof. By dualization of the theory. □

For $Y' \subseteq Y$, we define $\delta^a(Y') = \min\{a(y) \mid y \in Y'\}$ and $\iota_f^a(Y') = \min\{\iota_f^a(Y', y) \mid y \in Y'\}$. We obtain the following proof rules:

Lemma 3.4.2 (proof rules (dual))**.** *Let $\mathbb{M}$ be a complete MV-chain, Y a finite set and $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function. Let $a \in \mathbb{M}^Y$ be a fixpoint of f and let $\nu f_\#^a = Y'$ be the greatest fixpoint of the corresponding approximation. Then*

1. *$\nu f_\#^a = \emptyset$ if and only if $a = \mu f$.*
2. *$\iota_f^a \sqsubseteq \iota_f^a(Y') \sqsubseteq \delta^a(Y')$.*
3. *For all $\theta \sqsubseteq \iota_f^a(Y')$ the function $a \ominus \theta_{Y'}$ is a pre-fixpoint of f while for $\iota_f^a(Y') \sqsubset \theta \sqsubseteq \delta^a(Y')$ it is not.*

Let $a \in \mathbb{M}^Y$ be a post-fixpoint of f and let $[Y]^{a=f(a)} = \{y \in [Y]^a \mid a(y) = f(a)(y)\}$. We define $f_*^a : [Y]^{a=f(a)} \to [Y]^{a=f(a)}$ as $f_*^a(Y') = f_\#^a(Y') \cap [Y]^{a=f(a)}$, where $f_\#^a : \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ is the a-approximation of f. If $\nu f_*^a = \emptyset$ then $\mu f \sqsupseteq a$.

Proof. By dualization of the theory. □

3.5. (De)Composing Functions and Approximations

Given a non-expansive function f and a (pre/post-)fixpoint a, it is often non-trivial to determine the corresponding approximations. However, non-expansive functions enjoy good closure properties (closure under composition and closure under disjoint union) and we will see that the same holds for the corresponding approximations. Furthermore, it turns out that the functions needed in the applications can be obtained from just a few templates. This gives us a toolbox for assembling approximations with relative ease.

We first show that non-expansiveness is preserved by composition.

Lemma 3.5.1 (composing non-expansive functions)**.** *Let $\mathbb{M}$ be an MV-chain and let Y, W, Z be finite sets. If $g : \mathbb{M}^Y \to \mathbb{M}^W$ and $h : \mathbb{M}^W \to \mathbb{M}^Z$ are non-expansive then $h \circ g : \mathbb{M}^Y \to \mathbb{M}^Z$ is non-expansive.*

Proof. Straightforward. We have for any $a, b \in \mathbb{M}^Y$ that

$$\begin{aligned}
&\|h(g(b)) \ominus h(g(a))\| \sqsubseteq \\
&\quad \sqsubseteq \|g(b) \ominus g(a)\| && \text{[by non-expansiveness of } h\text{]} \\
&\quad \sqsubseteq \|b \ominus a\| && \text{[by non-expansiveness of } g\text{]}
\end{aligned}$$

□

Furthermore functions can be combined via disjoint union, preserving non-expansiveness, as follows.

Proposition 3.5.2 (disjoint union of non-expansive functions)**.** *Let $f_i : \mathbb{M}^{Y_i} \to \mathbb{M}^{Z_i}$, for $i \in I$ (I is some index set), be non-expansive and such that the sets Z_i are pairwise disjoint, i.e. $Z_i \cap Z_j = \emptyset$ for all $i, j \in I$ with $i \neq j$. The function $\biguplus_{i \in I} f_i : \mathbb{M}^{\bigcup_{i \in I} Y_i} \to \mathbb{M}^{\biguplus_{i \in I} Z_i}$ defined by*

$$\biguplus_{i \in I} f_i(a)(z) = f_i(a|_{Y_i})(z) \qquad \text{if } z \in Z_i$$

is non-expansive.[3]

Proof. For all $a, b \in \mathbb{M}^{\bigcup_{i \in I} Y_i}$ we have

$$\begin{aligned}
&\|\biguplus_{i \in I} f_i(b) \ominus \biguplus_{i \in I} f_i(a)\| \\
&= \max_{z \in \biguplus_{i \in I} Z_i} \Big(\biguplus_{i \in I} f_i(b)(z) \ominus \biguplus_{i \in I} f_i(a)(z)\Big) \\
&= \max_{i \in I} \max_{z \in Z_i} \big(f_i(b|_{Y_i})(z) \ominus f_i(a|_{Y_i})(z)\big) && \text{[since all } Z_i \text{ are disjoint]} \\
&= \max_{i \in I} \|f_i(b|_{Y_i}) \ominus f_i(a|_{Y_i})\| && \text{[by definition of norm]} \\
&\sqsubseteq \max_{i \in I} \|b|_{Y_i} \ominus a|_{Y_i}\| && \text{[since all } f_i \text{ are non-expansive]} \\
&= \max_{i \in I} \max_{y \in Y_i} (b(y) \ominus a(y)) \\
&= \max_{y \in \bigcup_{i \in I} Y_i} (b(y) \ominus a(y))
\end{aligned}$$

[3] $\uplus$ is used when the involved sets are disjoint.

$$= \|b \ominus a\| \qquad \text{[by definition of norm]}$$

□

We now introduce some basic functions, which will be used as the building blocks for the functions needed in the applications. Note that below we consider distributions on MV-chains of which the standard probability distributions introduced earlier are a special case.

Definition 3.5.3 (basic functions)**.** *Let $\mathbb{M}$ be an MV-chain and let Y, Z be finite sets.*

1. Constant: *For a fixed $k \in \mathbb{M}^Z$, we define $c_k : \mathbb{M}^Y \to \mathbb{M}^Z$ by*

$$c_k(a) = k$$

2. Reindexing: *For $u : Z \to Y$, we define $u^* : \mathbb{M}^Y \to \mathbb{M}^Z$ by*

$$u^*(a) = a \circ u.$$

3. Min/Max: *For $\mathcal{R} \subseteq Y \times Z$, we define* $\min_{\mathcal{R}}, \max_{\mathcal{R}} : \mathbb{M}^Y \to \mathbb{M}^Z$ *by*

$$\min_{\mathcal{R}}(a)(z) = \min_{y\mathcal{R}z} a(y) \qquad \max_{\mathcal{R}}(a)(z) = \max_{y\mathcal{R}z} a(y)$$

4. Average: *For a finite set $D \subseteq \mathcal{D}_{\mathbb{M}}(Y)$, we define* $\mathrm{av}_D : \mathbb{M}^Y \to \mathbb{M}^D$ *by*

$$\mathrm{av}_D(a)(p) = \bigoplus_{y \in Y} p(y) \odot a(y)$$

(cf. Definition 2.3.19 and the surrounding text) where $\odot$ is a binary operator on $\mathbb{M}$ and has the properties described after Definition 2.3.19.

5. Addition/Substraction: *For a fixed $w{:}\, Y \to \mathbb{M}$, we define $add_w, sub_w{:}\, \mathbb{M}^Y \to \mathbb{M}^Y$ by*

$$add_w(a)(y) = a(y) \oplus w(y) \qquad sub_w(a)(y) = a(y) \ominus w(y)$$

A particularly interesting subcase of (3) is when we take as relation the *belongs to* relation $\in \subseteq Y \times \mathcal{P}(Y)$. In this way we obtain functions for selecting the minimum and the maximum, respectively, of an input function over a set $Y' \subseteq Y$, that is, the functions

$\min_\in, \max_\in : \mathbb{M}^Y \to \mathbb{M}^{\mathcal{P}(Y)}$, defined as

$$\min_\in(a)(Y') = \min_{y \in Y'} a(y) \qquad \max_\in(a)(Y') = \max_{y \in Y'} a(y)$$

The basic functions can be shown to be non-expansive.

Proposition 3.5.4. *The basic functions from Definition 3.5.3 are all non-expansive.*

Proof. See Appendix: Proposition A.2.4. □

The next result determines the approximations associated with the basic functions.

Proposition 3.5.5 (approximations of basic functions)**.** *Let $\mathbb{M}$ be an MV-chain, Y, Z be finite sets and let $a \in \mathbb{M}^Y$.*

- Constant: *for $k : \mathbb{M}^Z$, the approximations $(c_k)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{c_k(a)})$, $(c_k)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{c_k(a)})$ are*

$$(c_k)^\#_a(Y') = \emptyset = (c_k)^a_\#(Y')$$

- Reindexing: *for $u : Z \to Y$, the approximations $(u^*)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{u^*(a)})$, $(u^*)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{u^*(a)})$ are*

$$(u^*)^\#_a(Y') = (u^*)^a_\#(Y') = u^{-1}(Y') = \{z \in [Z]_{u^*(a)} \mid u(z) \in Y'\}$$

- Min: *for $\mathcal{R} \subseteq Y \times Z$, the approximations $(\min_\mathcal{R})^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{\min_\mathcal{R}(a)})$, $(\min_\mathcal{R})^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{\min_\mathcal{R}(a)})$ are given below, where $\mathcal{R}^{-1}(z) = \{y \in Y \mid y\mathcal{R}z\}$:*

$$(\min_\mathcal{R})^\#_a(Y') = \{z \in [Z]_{\min_\mathcal{R}(a)} \mid \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y) \subseteq Y'\}$$
$$(\min_\mathcal{R})^a_\#(Y') = \{z \in [Z]^{\min_\mathcal{R}(a)} \mid \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset\}$$

- Max: *for $\mathcal{R} \subseteq Y \times Z$, the approximations $(\max_\mathcal{R})^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{\max_\mathcal{R}(a)})$, $(\max_\mathcal{R})^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{\max_\mathcal{R}(a)})$ are*

$$(\max_\mathcal{R})^\#_a(Y') = \{z \in [Z]_{\max_\mathcal{R}(a)} \mid \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset\}$$
$$(\max_\mathcal{R})^a_\#(Y') = \{z \in [Z]^{\max_\mathcal{R}(a)} \mid \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y) \subseteq Y'\}$$

- Average: *for a finite* $D \subseteq \mathcal{D}_{\mathbb{M}}(Y)$*, the approximations* $(\mathrm{av}_D)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([D]_{\mathrm{av}_D(a)})$, $(\mathrm{av}_D)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([D]^{\mathrm{av}_D(a)})$ *are*

$$(\mathrm{av}_D)^\#_a(Y') = \{p \in [D]_{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\}$$
$$(\mathrm{av}_D)^a_\#(Y') = \{p \in [D]^{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\},$$

where $supp(p) = \{y \in Y \mid p(y) > 0\}$ *for* $p \in \mathcal{D}(Y)$.

- Addition: *for* $w \in \mathbb{M}^Y$*, the approximations* $(add_w)^\#_a : \mathbb{M}^{[Y]_a} \to \mathbb{M}^{[Y]_{add_w(a)}}$, $(add_w)^a_\# : \mathbb{M}^{[Y]^a} \to \mathbb{M}^{[Y]^{add_w(a)}}$ *are*

$$(add_w)^\#_a(Y') = \{y \in Y' \mid a(y) \oplus w(y) \sqsubset 1\}$$
$$(add_w)^a_\#(Y') = \{y \in Y' \mid w(y) \sqsubseteq \overline{a(y)}\}$$

- Substraction: *for* $w \in \mathbb{M}^Y$*, the approximations and* $(sub_w)^\#_a : \mathbb{M}^{[Y]_a} \to \mathbb{M}^{[Y]_{sub_w(a)}}$, $(sub_w)^a_\# : \mathbb{M}^{[Y]^a} \to \mathbb{M}^{[Y]^{sub_w(a)}}$ *are*

$$(sub_w)^\#_a(Y') = \{y \in Y' \mid w(y) \sqsubseteq a(y)\} = Y'$$
$$(sub_w)^a_\#(Y') = \{y \in Y' \mid a(y) \ominus w(y) \sqsupset 0\}$$

Proof. See Appendix: Proposition A.2.5. □

When a non-expansive function arises as the composition of simpler ones (see Lemma 3.5.1) we can obtain the corresponding approximation by just composing the approximations of the simpler functions.

Proposition 3.5.6 (composing approximations)**.** *Let* $g : \mathbb{M}^Y \to \mathbb{M}^W$ *and* $h : \mathbb{M}^W \to \mathbb{M}^Z$ *be non-expansive functions. For all* $a \in \mathbb{M}^Y$ *we have that* $(h \circ g)^\#_a = h^\#_{g(a)} \circ g^\#_a$. *Analogously* $(h \circ g)^a_\# = h^{g(a)}_\# \circ g^a_\#$ *for the dual case.*

Proof. Here we only consider the primal case, the dual case for $(h \circ g)^a_\#$ is analogous.

Let $0 \sqsubset \theta \sqsubseteq \min\{\iota^g_a, \iota^h_{g(a)}\}$. Then, by Theorem 3.2.13(b) we know that

$$g^\#_a = g^\#_{a,\theta} = \gamma_{g(a),\theta} \circ g \circ \alpha_{a,\theta}$$

$$h^\#_{g(a)} = h^\#_{g(a),\theta} = \gamma_{h(g(a)),\theta} \circ h \circ \alpha_{g(a),\theta}$$

Now we will prove that

$$(h \circ g)^\#_{a,\theta} = h^\#_{g(a),\theta} \circ g^\#_{a,\theta}$$

First observe that $g(\alpha_{a,\theta}(Y')) \in [g(a), g(a \oplus \theta)] \subseteq [g(a), g(a) \oplus \theta]$ for all $Y' \subseteq [Y]_a$ by Lemma 2.3.25. Applying Theorem 3.2.13(b) on h we obtain

$$\begin{aligned} & (h \circ g)^{\#}_{a,\theta} = \gamma_{h(g(a)),\theta} \circ h \circ g \circ \alpha_{a,\theta}(Y') = h^{\#}_{g(a),\theta} \circ \gamma_{g(a),\theta} \circ g \circ \alpha_{a,\theta}(Y') \\ = \quad & h^{\#}_{g(a),\theta} \circ g^{\#}_{a,\theta}(Y') = h^{\#}_{g(a)} \circ g^{\#}_{a}(Y') \end{aligned}$$

Hence all functions $(h \circ g)^{\#}_{a,\theta}$ are equal and independent of θ and so it must hold that $(h \circ g)^{\#}_{a,\theta} = (h \circ g)^{\#}_{a}$. Then from Theorem 3.2.13 we can conclude $\min\{\iota^g_a, \iota^h_{g(a)}\} \sqsubseteq \iota^{h \circ g}_a$. □

Also, the corresponding approximation of a disjoint union can be conveniently assembled from the approximations of its components.

Proposition 3.5.7 (disjoint union and approximations)**.** *The approximations for* $\biguplus_{i \in I} f_i$, *where* $f_i : \mathbb{M}^{Y_i} \to \mathbb{M}^{Z_i}$ *are non-expansive and* Z_i *are pairwise disjoint, have the following form. For all* $a : \bigcup_{i \in I} Y_i \to \mathbb{M}$ *and* $Y' \subseteq \bigcup_{i \in I} Y_i$:

$$\Big(\biguplus_{i \in I} f_i\Big)^{\#}_{a}(Y') = \biguplus_{i \in I} (f_i)^{\#}_{a|_{Y_i}}(Y' \cap Y_i) \qquad \Big(\biguplus_{i \in I} f_i\Big)^{a}_{\#}(Y') = \biguplus_{i \in I} (f_i)^{a|_{Y_i}}_{\#}(Y' \cap Y_i)$$

Proof. We just show the statement for the primal case, the dual case is analogous. We abbreviate $Y = \bigcup_{i \in I} Y_i$.

Let $0 \sqsubset \theta \sqsubseteq \delta_a$. According to the definition of a-approximation (Lemma 3.2.10) we have for $Y' \subseteq [Y]_a$:

$$\Big(\biguplus_{i \in I} f_i\Big)^{\#}_{a,\theta}(Y') = \gamma_{\biguplus_{i \in I} f_i(a),\theta} \circ \biguplus_{i \in I} f_i \circ \alpha_{a,\theta}$$

$$(f_i)^{\#}_{a|_{Y_i},\theta} = \gamma_{f_i(a|_{Y_i}),\theta} \circ f_i \circ \alpha_{a|_{Y_i},\theta}$$

for all $i \in I$. Our first step is to prove that

$$\gamma_{\biguplus_{i \in I} f_i(a),\theta} \circ \biguplus_{i \in I} f_i \circ \alpha_{a,\theta}(Y') = \biguplus_{i \in I} \gamma_{f_i(a|_{Y_i}),\theta} \circ f_i \circ \alpha_{a|_{Y_i},\theta}(Y' \cap Y_i)$$

By simply expanding the functions we obtain

$$\gamma_{\biguplus_{i \in I} f_i(a),\theta} \circ \biguplus_{i \in I} f_i \circ \alpha_{a,\theta}(Y') = \{z \in Z_i \mid i \in I \ \wedge \ \theta \sqsubseteq f_i((a \oplus \theta_{Y'})|_{Y_i})(z) \ominus f_i(a|_{Y_i})(z)\}$$

$$\biguplus_{i \in I} \gamma_{f_i(a|_{Y_i}),\theta} \circ f_i \circ \alpha_{a|_{Y_i},\theta}(Y' \cap Y_i) = \biguplus_{i \in I} \{z \in Z_i \mid \theta \sqsubseteq f_i(a|_{Y_i} \oplus \theta_{Y' \cap Y_i})(z) \ominus f_i(a|_{Y_i})(z)\}$$

which are the same set, since for all $i \in I$ clearly $(a \oplus \theta_{Y'})|_{Y_i} = a|_{Y_i} \oplus \theta_{Y' \cap Y_i}$.

This implies

$$\Big(\biguplus_{i \in I} f_i\Big)^{\#}_{a,\theta}(Y') = \biguplus_{i \in I} (f_i)^{\#}_{a|_{Y_i},\theta}(Y' \cap Y_i).$$

Whenever $\theta \sqsubseteq \min_{i \in I} \iota_a^{f_i}$, this can be rewritten to

$$\Big(\biguplus_{i \in I} f_i\Big)^{\#}_{a,\theta}(Y') = \biguplus_{i \in I} (f_i)^{\#}_{a|_{Y_i}}(Y' \cap Y_i).$$

All functions $\big(\biguplus_{i \in I} f_i\big)^{\#}_{a,\theta}$ are equal and independent of θ and so it must hold that $\big(\biguplus_{i \in I} f_i\big)^{\#}_{a,\theta} = \big(\biguplus_{i \in I} f_i\big)^{\#}_{a}$. Then with Theorem 3.2.13 we can also conclude $\min_{i \in I} \iota_a^{f_i} \sqsubseteq \iota_a^{\biguplus_{i \in I} f_i}$. □

Table 3.1.: Basic functions $f\colon \mathbb{M}^Y \to \mathbb{M}^Z$ (constant, reindexing, minimum, maximum, average, addition, substraction), function composition, disjoint union and the corresponding approximations $f_a^{\#}\colon \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{f(a)})$, $f_{\#}^{a}\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{f(a)})$.

Notation: $\mathcal{R}^{-1}(z) = \{y \in Y \mid y\mathcal{R}z\}$, $supp(p) = \{y \in Y \mid p(y) > 0\}$ for $p \in \mathcal{D}(Y)$, $Min_a = \{y \in Y \mid a(y) \text{ minimal}\}$, $Max_a = \{y \in Y \mid a(y) \text{ maximal}\}$, $d(a) = \min\{a(y) \ominus a(y') \mid a(y) \ominus a(y') \sqsupset 0,\ y, y' \in Y\}$, $a\colon Y \to \mathbb{M}$

function f	definition of f	$f_a^{\#}(Y')$ (above), $f_{\#}^{a}(Y')$ (below)	ι_a^f (above), ι_f^a (below)
c_k $(k \in \mathbb{M}^Z)$	$f(a) = k$	$\emptyset$ $\emptyset$	δ_a δ^a
u^* $(u\colon Z \to Y)$	$f(a) = a \circ u$	$u^{-1}(Y')$ $u^{-1}(Y')$	δ_a δ^a
$\min_{\mathcal{R}}$ $(\mathcal{R} \subseteq Y \times Z)$	$f(a)(z) = \min_{y\mathcal{R}z} a(y)$	$\{z \in [Z]_{f(a)} \mid Min_{a\|_{\mathcal{R}^{-1}(z)}} \subseteq Y'\}$ $\{z \in [Z]^{f(a)} \mid Min_{a\|_{\mathcal{R}^{-1}(z)}} \cap Y' \neq \emptyset\}$	$\sqsupseteq \min\{d(a), \delta_a\}$ δ^a
$\max_{\mathcal{R}}$ $(\mathcal{R} \subseteq Y \times Z)$	$f(a)(z) = \max_{y\mathcal{R}z} a(y)$	$\{z \in [Z]_{f(a)} \mid Max_{a\|_{\mathcal{R}^{-1}(z)}} \cap Y' \neq \emptyset\}$ $\{z \in [Z]^{f(a)} \mid Max_{a\|_{\mathcal{R}^{-1}(z)}} \subseteq Y'\}$	δ_a $\sqsupseteq \min\{d(a), \delta^a\}$
av_D $(Z = D \subseteq \mathcal{D}(Y))$	$f(a)(p) = \bigoplus_{y \in Y} p(y) \odot a(y)$	$\{p \in [D]_{f(a)} \mid supp(p) \subseteq Y'\}$ $\{p \in [D]^{f(a)} \mid supp(p) \subseteq Y'\}$	δ_a δ^a
add_w $(Z = Y,\ w\colon Y \to \mathbb{M})$	$f(a)(y) = a(y) \oplus w(y)$	$[Y']_{add_w(a)}$ $\{y \in Y' \mid w(y) \sqsubseteq \overline{a(y)}\}$	$\delta_{add_w(a)}$ δ^a
sub_w $(Z = Y,\ w\colon Y \to \mathbb{M})$	$f(a)(y) = a(y) \ominus w(y)$	$\{y \in Y' \mid w(y) \sqsubseteq a(y)\}$ $[Y']^{sub_w(a)}$	δ_a $\delta^{sub_w(a)}$
$h \circ g$ $(g\colon \mathbb{M}^Y \to \mathbb{M}^W,$ $h\colon \mathbb{M}^W \to \mathbb{M}^Z)$	$f(a) = h(g(a))$	$h^{\#}_{g(a)} \circ g^{\#}_a(Y')$ $h^{g(a)}_{\#} \circ g^{a}_{\#}(Y')$	$\sqsupseteq \min\{\iota_a^g, \iota_{g(a)}^h\}$ $\sqsupseteq \min\{\iota_g^a, \iota_h^{g(a)}\}$
$\biguplus_{i \in I} f_i$ I finite $(f_i\colon \mathbb{M}^{Y_i} \to \mathbb{M}^{Z_i},$ $Y = \bigcup_{i \in I} Y_i,\ Z = \biguplus_{i \in I} Z_i)$	$f(a)(z) = f_i(a\|_{Y_i})(z)$ $(z \in Z_i)$	$\biguplus_{i \in I} (f_i)^{\#}_{a\|_{Y_i}}(Y' \cap Y_i)$ $\biguplus_{i \in I} (f_i)^{a\|_{Y_i}}_{\#}(Y' \cap Y_i)$	 $\min_{i \in I} \iota_{f_i}^a$

We can summarize the desired results (non-expansiveness and approximation) for the basic building blocks and their composition (all schematically reported in Table 3.1).

Corollary 3.5.8. *All basic functions in Definition 3.5.3 are non-expansive. Furthermore non-expansive functions are closed under composition and disjoint union. The approximations are the ones listed in the third column of Table 3.1.*

Proof. Follows directly from Propositions 3.5.4, 3.5.5, 3.5.6, 3.5.2, 3.5.7 and Lemma 3.5.1. □

We can also specify the maximal decrease respectively increase that is propagated (here we are using the notation of Lemma 3.2.10 and Lemma 3.4.1).

Corollary 3.5.9. *We consider the basic functions from Definition 3.5.3, function composition as in Lemma 3.5.1 and disjoint union as in Proposition 3.5.2 and give the corresponding values for ι_a^f and ι_f^a.*
For greatest fixpoints (primal case) we obtain:

- $\iota_a^{c_k} = \iota_a^{u^*} = \iota_a^{\max_\mathcal{R}} = \iota_a^{\mathrm{av}_D} = \iota_a^{sub_w} = \delta_a$
- $\iota_a^{\min_\mathcal{R}} = \min\limits_{z \in [Z]_{\min_\mathcal{R}(a)}} \{a(y) \ominus a(\hat{y}) \mid y\mathcal{R}z, y \notin \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y), \hat{y} \in \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y)\} \cup \{\delta_a\}$
- $\iota_a^{add_w} = \min_{y \in Y}\{\overline{a(y) \oplus w(y)} \mid a(y) \oplus w(y) \sqsubset 1\}$
- $\iota_a^{g \circ f} \sqsupseteq \min\{\iota_a^f, \iota_{f(a)}^g\}$
- $\iota_a^{\biguplus_{i\in I} f_i} = \min_{i \in I} \iota_{a|_{Y_i}}^{f_i}$

For least fixpoints (dual case) we obtain:

- $\iota_{c_k}^a = \iota_{u^*}^a = \iota_{\min_\mathcal{R}}^a = \iota_{\mathrm{av}_D}^a = \iota_{add_w}^a = \delta^a$
- $\iota_{\max_\mathcal{R}}^a = \min\limits_{z \in [Z]^{\min_\mathcal{R}(a)}} \{a(\hat{y}) \ominus a(y) \mid y\mathcal{R}z, \hat{y} \in \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y), y \notin \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y)\} \cup \{\delta^a\}$
- $\iota_{sub_w}^a = \min_{y \in Y}\{a(y) \ominus w(y) \mid a(y) \ominus w(y) \sqsupset 0\}$
- $\iota_{g \circ f}^a \sqsupseteq \min\{\iota_f^a, \iota_g^{f(a)}\}$
- $\iota_{\biguplus_{i\in I} f_i}^a = \min_{i \in I} \iota_{f_i}^{a|_{Y_i}}$

Proof. See Appendix: Corollary A.2.6. □

3.6. Applications

We will now tend to the applications discussed in Chapter 2. As we will see, we can derive (almost) all warranted functions as composition and disjoint union of the basic functions from Definition 3.5.3. We will proceed as follows: First, we show non-expansiveness of each function by disassembling them via these basic functions. Next, we derive the approximation. We also give a lower bound for the ascent constant ι_a^f/descent constant ι_f^a.

3.6.1. Deriving the Approximation for Termination Probability of Markov Chains

We now want to revisit Section 2.6.1, thereby investigating the example in the introduction in more detail. Termination probability of a Markov chain $\mathsf{MC} = (S, T, \eta)$ arises as the least fixpoint of the function $\mathcal{T}\colon [0,1]^S \to [0,1]^S$, defined for $t \in [0,1]^S$ and $s \in S$ as

$$\mathcal{T}(t)(s) = \begin{cases} \sum_{s' \in S} \eta(s)(s') \cdot t(s') & \text{if } s \in S \setminus T \\ 1 & \text{otherwise} \end{cases}$$

The least fixpoint $\mu\mathcal{T}$ assigns to each state its termination probability. Here, we have $\mathbb{M} = [0,1]$. We restrict the codomain of $\eta\colon S \setminus T \to \mathcal{D}(S)$ to $D \subseteq \mathcal{D}(S)$, where D is finite (to ensure that all involved sets are finite), i.e. $D = \{\eta(s) \mid s \in S \setminus T\}$.

Lemma 3.6.1 (decomposing $\mathcal{T}$)**.** *The function $\mathcal{T}$ can be written as $\mathcal{T} = (\eta^* \circ \mathrm{av}_D) \uplus c_k$ where $k\colon T \to [0,1]$ is the constant function 1 defined only on terminal states.*

Proof. Let $t\colon S \to [0,1]$. For $s \in T$ we have

$$\begin{aligned} &((\eta^* \circ \mathrm{av}_D) \uplus c_k)(t)(s) \\ &= c_k(t)(s) && [\text{since } s \in T] \\ &= k(s) = 1 && [\text{by definition of } c_k \text{ and } k] \\ &= \mathcal{T}(t)(s) && [\text{since } s \in T] \end{aligned}$$

For $s \notin T$ we have

$$\begin{aligned} &((\eta^* \circ \mathrm{av}_D) \uplus c_k)(t)(s) \\ &= \eta^* \circ \mathrm{av}_D(t)(s) && [\text{since } s \notin T] \\ &= \mathrm{av}_D(t)(\eta(s)) && [\text{by definition of reindexing}] \\ &= \sum_{s' \in S} \eta(s)(s') \cdot t(s') && [\text{by definition of } \mathrm{av}_D] \\ &= \mathcal{T}(t)(s) && [\text{since } s \notin T] \end{aligned}$$

□

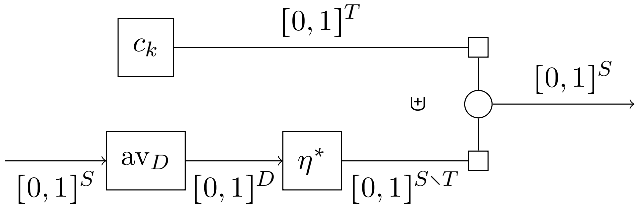


Fig. 3.3.: Decomposition of $\mathcal{T}$

From this representation and Theorem 3.5.8 it is obvious that $\mathcal{T}$ is non-expansive and we can derive the approximation $\mathcal{T}_\#^t$ in the dual sense.

Lemma 3.6.2 (approximating $\mathcal{T}$)**.** *Given a function $t\colon S \to [0,1]$, the t-approximation for $\mathcal{T}$ in the dual sense is $\mathcal{T}_\#^t\colon \mathcal{P}([S]^t) \to \mathcal{P}([S]^{\mathcal{T}(t)})$ with*

$$\mathcal{T}_\#^t(S') = \{s \in [S]^{\mathcal{T}(t)} \mid s \notin T \wedge supp(\eta(s)) \subseteq S'\}.$$

Proof. See Appendix: Lemma A.2.7. □

Intuitively, states which form a cycle where there is no outgoing edge to any terminal state can have their values reduced (as long as their values all are above 0).

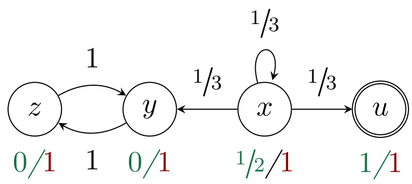
Remark 3.6.3. *The descent constant $\iota_\mathcal{T}^t$ is bounded as follows:*

$$\iota_\mathcal{T}^t \sqsupseteq \delta^t.$$

We can draw a nice string diagram to illustrate the decomposition of $\mathcal{T}$, see Figure 3.3. This can be seen as early motivation for Chapter 4 where we will obtain a categorical view of our theory and show that we are operating in a gs-monoidal setting.

At this point we have all the ingredients needed to formalise the application presented in the introduction (Section 3.1).

Example 3.6.4. *Again, consider the following Markov chain, where we are given $\mu\mathcal{T}$ in green and $\nu\mathcal{T}$ in red.*



We now compute $\nu\mathcal{T}_\#^{\nu\mathcal{T}}$ via Kleene iteration:

1. *$\mathcal{T}_\#^{\nu\mathcal{T}}(S) = \{x, y, z\} = S'$ since u is a terminal state.*

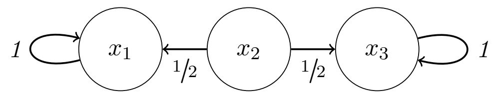
2. $\mathcal{T}_{\#}^{\nu\mathcal{T}}(S') = \{y,z\} = S''$ *since* $u \notin S'$ *and thus* $supp(\eta(x)) \nsubseteq S'$.

3. $\mathcal{T}_{\#}^{\nu\mathcal{T}}(S'') = \{y,z\} = S''$ *since for both* y,z *it holds* $supp(\eta(y)) = z, supp(\eta(z)) \subseteq S''$.

Thus, we know that we can reduce values in $S'' = \nu\mathcal{T}_{\#}^{\nu\mathcal{T}}$ *by* $1 = \delta_{\nu\mathcal{T}}$*, i.e.* $\nu\mathcal{T} \ominus 1_{S''}$ *is a prefixpoint of* $\mathcal{T}$*.*

We present a new example that allows us to illustrate the question of the largest decrease for a fixpoint that still guarantees a pre-fixpoint (the dual problem is treated in Proposition 3.3.5).

Example 3.6.5. *Consider the following Markov chain where* $S = \{x_1, x_2, x_3\}$ *are non-terminal states. The least fixpoint of the underlying fixpoint function* $\mathcal{T}$ *is clearly the constant* 0*, since no state can reach a terminal state.*



Now consider the function $t\colon S \to [0,1]$ *defined by* $t(x_1) = 0.1$*,* $t(x_2) = 0.5$ *and* $t(x_3) = 0.9$*. This is also a fixpoint of* $\mathcal{T}$*.*

Observe that $\mathcal{T}_{\#}^{t}(S) = S$ *and thus, clearly,* $\nu\mathcal{T}_{\#}^{t} = S$*. According to (the dual of) Definition 3.3.3 we have* $\delta_t(S) = 0.1$ *and thus, by (the dual of) Proposition 3.3.5, the function* $t' = t \ominus (0.1)_S$*, with* $t'(x_1) = 0$*,* $t'(x_2) = 0.4$*, and* $t'(x_3) = 0.8$*, is a pre-fixpoint. Indeed,* $\mathcal{T}(t')(x_1) = 0$*,* $\mathcal{T}(t')(x_2) = 0.4$ *and* $\mathcal{T}(t')(x_3) = 0.8$*.*

This is not the largest decrease producing a pre-fixpoint. In fact, we can choose $\theta = 0.9$*, greater than* $\delta^t(S)$ *and we have that* $t \ominus \theta_S$ *is the constant* 0*, i.e., the least fixpoint of* $\mathcal{T}$*. However, if we take* $\theta' = 0.5 \sqsubset \theta$*, then* $t \ominus \theta'_S$ *is not a pre-fixpoint. In fact* $(t \ominus \theta'_S)(x_2) = 0$*, while* $\mathcal{T}(t \ominus \theta'_S)(x_2) = 0.2$*. This means that the set of decreases (beyond* $\delta_t(S)$*) producing a pre-fixpoint is not downward-closed and hence the largest decrease cannot be found by binary search, while, as already mentioned, a binary search will work for decreases below* $\delta^t(S)$*.*

3.6.2. Deriving the Approximation for the Kantorovich Lifting

The Kantorovich lifting converts a metric on X to a metric on probability distributions over X (see Section 2.5.6). In order to ensure finiteness of all the sets involved, we restrict to $D \subseteq \mathcal{D}(X)$, some finite set of probability distributions over X. The Kantorovich lifting was defined as $\mathcal{K} : [0,1]^{X\times X} \to [0,1]^{D\times D}$ where

$$\mathcal{K}(d)(p,q) = \min_{\omega\in\Omega(p,q)} \sum_{(x_1,x_2)\in X\times X} \omega(x_1,x_2)\cdot d(x_1,x_2).$$

for $d \in [0,1]^{X\times X}$ and $p,q \in D \subseteq \mathcal{D}(X)$. As it commonly happens, we define the lifting for general distance functions on $[0,1]$, not restricting to (pseudo-)metrics. We have $\mathbb{M} = [0,1]$.

As a reminder (cf. Section 2.5.6), a *coupling* of $p, q \in D$ is a probability distribution $\omega \in \mathcal{D}(X \times X)$ whose left and right marginals are p, q, i.e., $p(x_1) = m^L_\omega(x_1) \coloneqq \sum_{x_2 \in X} \omega(x_1, x_2)$ and $q(x_2) = m^R_\omega(x_2) \coloneqq \sum_{x_1 \in X} \omega(x_1, x_2)$. The set of all couplings of p, q, denoted by $\Omega(p, q)$, forms a polytope with finitely many vertices: $\Omega_V(p, q)$. The set of all polytope vertices that are obtained by coupling any $p, q \in D$ is also finite and is denoted by $VP_D \subseteq \mathcal{D}(X \times X)$.

Below we provide an alternative characterisation of $\mathcal{K}$, which shows non-expansiveness of $\mathcal{K}$ and allows one to derive its approximations.

Lemma 3.6.6 (decomposing $\mathcal{K}$)**.** *Let* $u : VP_D \to D \times D$, $u(\omega) = (m^L_\omega, m^R_\omega)$. *Then* $\mathcal{K} = \min_u \circ \mathrm{av}_{VP_D}$, *where* $\mathrm{av}_{VP_D} \colon [0,1]^{X \times X} \to [0,1]^{VP_D}$, $\min_u \colon [0,1]^{VP_D} \to [0,1]^{D \times D}$.

Proof. It holds that $u^{-1}(p, q) = \Omega(p, q) \cap VP_D$ for $p, q \in D$. Furthermore note it is sufficient to consider as couplings the vertices, i.e., the elements of VP_D, since the minimum is always attained there [PC20].

Hence we obtain for $d \colon X \times X \to [0,1]$, $p, q \in D$:

$$\begin{aligned}
\min_u(\mathrm{av}_{VP_D}(d))(p, q) &= \min_{\omega \in \Omega(p,q) \cap VP_D} \mathrm{av}_{VP_D}(d)(\omega) \\
&= \min_{\omega \in \Omega(p,q) \cap VP_D} \sum_{x_1, x_2 \in X \times X} \omega(x_1, x_2) \cdot d(x_1, x_2) \\
&= \min_{\omega \in \Omega(p,q)} \sum_{x_1, x_2 \in X \times X} \omega(x_1, x_2) \cdot d(x_1, x_2) \\
&= \mathcal{K}(d)(p, q)
\end{aligned}$$

□

We next present the approximation of the Kantorovich lifting in the dual sense. Intuitively, given a distance function d and a relation M on X, it characterises those pairs (p, q) of distributions whose distance in the Kantorovich metric decreases by a constant when we decrease the distance d for all pairs in M by the same constant.

Lemma 3.6.7 (approximating $\mathcal{K}$)**.** *Let* $d \colon X \times X \to [0,1]$. *The approximation for the Kantorovich lifting* $\mathcal{K}$ *in the dual sense is* $\mathcal{K}^d_\# \colon \mathcal{P}([X \times X]^d) \to \mathcal{P}([D \times D]^{\mathcal{K}(d)})$ *with*

$$\begin{aligned}
\mathcal{K}^d_\#(M) = \{(p, q) \in [D \times D]^{\mathcal{K}(d)} \mid & \exists \omega \in \Omega(p, q), supp(\omega) \subseteq M, \\
& \sum_{u, v \in S} d(u, v) \cdot \omega(u, v) = \mathcal{K}(d)(p, q)\}.
\end{aligned}$$

Proof. See Appendix: Lemma A.2.8. □

Remark 3.6.8. *The descent constant* $\iota^d_{\mathcal{K}}$ *has as lower bound:*

$$\iota^d_{\mathcal{K}} \sqsupseteq \delta^d.$$

3.6.3. Deriving the Approximation for Behavioural Distances of Labeled Markov Chains

The results from the previous section allow us to derive the approximation of the behavioural distance function for labeled Markov chains (cf. Section 2.6.3). $\Delta\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ (in its simplified form) was defined as

$$\Delta(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{K}(d)(\eta(s),\eta(t)) & \text{otherwise} \end{cases}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$ ($\mathbb{M} = [0,1]$). First, we prove non-expansiveness of Δ.

Lemma 3.6.9 (decomposing Δ)**.** *The function* Δ *can be written as*

$$\Delta = \max\nolimits_\rho \circ ((\eta\times\eta)^* \circ \mathcal{K} \uplus c_k)$$

where $\rho\colon (S\times S) \uplus (S\times S) \to (S\times S)$ *with* $\rho((s,t),i) = (s,t)$, $i = 0,1$, *and* $k\colon S\times S \to [0,1]$ *is defined as*

$$k(s,t) = \begin{cases} 1 & \textit{if } \ell(s) \neq \ell(t) \\ 0 & \textit{otherwise} \end{cases}.$$

Proof. Given $d\colon S\times S \to [0,1]$ and $s,t \in S$. We have

$$\begin{aligned} \max\nolimits_\rho \circ ((\eta\times\eta)^* \circ \mathcal{K} \uplus c_k)(d)(s,t) &= \max\{(\eta\times\eta)^* \circ \mathcal{K}(d)(s,t), c_k(d)(s,t)\} \\ &= \max\{\mathcal{K}(d)(\eta(s),\eta(t)), c_k(d)(s,t)\} \\ &= \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{K}(d)(\eta(s),\eta(t)) & \text{otherwise} \end{cases} \end{aligned}$$

□

We quickly remark that both $(\eta\times\eta)^* \circ \mathcal{K}$ and c_k map to $[0,1]^{S\times S}$. We artificially make these sets disjoint by adding an index (0 and 1) which is required by the theory as $\uplus$ only works on disjoint sets, i.e.

$$\begin{aligned} &(\eta\times\eta)^* \circ \mathcal{K}\colon [0,1]^{S\times S} \to [0,1]^{S\times S\times\{0\}},\ c_k\colon [0,1]^{S\times S} \to [0,1]^{S\times S\times\{1\}} \\ &\text{and } \rho\colon [0,1]^{(S\times S\times\{0\})\uplus(S\times S\times\{1\})} \to [0,1]^{S\times S}. \end{aligned}$$

We can draw an illustrating string diagram to illustrate the decomposition of Δ, see Figure 3.4.

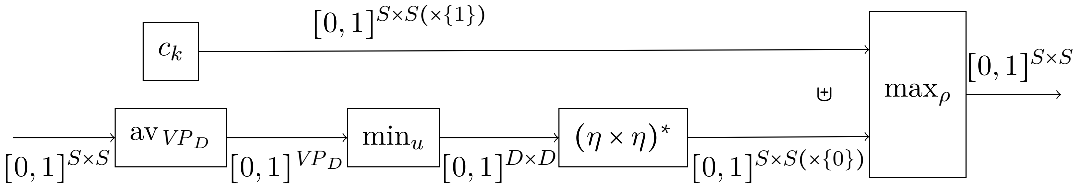


Fig. 3.4.: Decomposition of the fixpoint function for computing behavioural metrics for labeled Markov chains.

Δ is clearly non-expansive and we obtain the following approximation in the dual sense.

Lemma 3.6.10 (approximating Δ)**.** *Let $d\colon S\times S\to[0,1]$. The approximation of Δ in the dual sense is $\Delta^d_\#\colon\mathcal{P}([S\times S]^d)\to\mathcal{P}([S\times S]^{\Delta(d)})$ with*

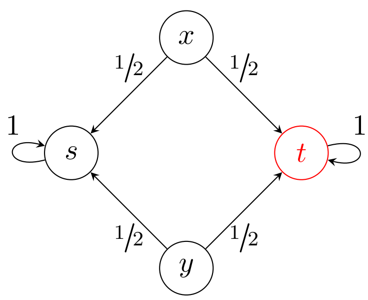
$$\Delta^d_\#(M)=\{(s,t)\in[S\times S]^{\Delta(d)}\mid \ell(s)=\ell(t)\wedge(\eta(s),\eta(t))\in\mathcal{K}^d_\#(M)\}.$$

Proof. See Appendix: Lemma A.2.9. □

Remark 3.6.11. *The descent constant ι^d_Δ has the following lower bound:*

$$\iota^d_\Delta\sqsupseteq\delta^d.$$

Example 3.6.12. *We are given the following labeled Markov chain where all states except for state t have the same label.*



Assume we are given the following fixpoint of Δ: $d(x,x)=d(y,y)=0$, $d(x,v)=d(y,v)=d(s,t)=1$, $d(x,s)=d(y,s)={}^3\!/\!_4$, $d(t,t)=d(s,s)={}^1\!/\!_2$ and $d(x,y)={}^1\!/\!_4$. Symmetrical values are identical.

We now show that $M = \{(s,s),(t,t),(x,y),(y,x)\}$ *is a fixpoint of* $\Delta^d_\#$ *(in fact it is the largest one). Now,* $(u,u) \in \Delta^d_\#(M)$ *since the only viable coupling* $\omega \in \Omega(\eta(s),\eta(s))$ *is given by* $\omega(s,s) = 1$ *and*

$$\mathcal{K}(d)(\eta(s),\eta(s)) = \sum_{u,v\in S} d(u,v)\cdot\omega(u,v) = 1/2.$$

Since $(s,s) \in M$*, we conclude* $(s,s) \in \Delta^d_\#(M)$*. For the same reasoning* $(t,t) \in \Delta^d_\#(M)$*.*

We have two couplings $\omega_1,\omega_2 \in \Omega_V(\eta(x),\eta(y))$ *given by* $\omega_1(s,s) = \omega_1(t,t) = \omega_2(s,t) = \omega_2(t,s) = 1/2$*. We have*

$$\mathcal{K}(d)(\eta(x),\eta(y)) = \sum_{u,v\in S} d(u,v)\cdot\omega_1(u,v) = 1/4$$

and since $(s,s),(t,t) \in \Delta^d_\#(M)$ *we conclude* $(x,y) \in \Delta^d_\#(M)$*. For the same reasoning* $(y,x) \in \Delta^d_\#(M)$*.*

3.6.4. Deriving the Approximation for the Hausdorff Lifting

Given a (pseudo-)metric d on a finite set X, the Hausdorff lifting of $\mathcal{H}(d)$ provides a (pseudo-)metric on the powerset $\mathcal{P}(X)$ (see Section 2.5.5). As for the Kantorovich lifting, we lift distance functions that are not necessarily (pseudo-)metrics. The Hausdorff lifting (primal representation) was defined as $\mathcal{H} : [0,1]^{X\times X} \to [0,1]^{\mathcal{P}(x)\times\mathcal{P}(X)}$ where

$$\mathcal{H}(d)(X_1,X_2) = \max\{\max_{x_1\in X_1}\min_{x_2\in X_2} d(x_1,x_2), \max_{x_2\in X_2}\min_{x_1\in X_1} d(x_1,x_2)\}.$$

for $d \in [0,1]^{X\times X}$ and $X_1,X_2 \in \mathcal{P}(X)$ ($\mathbb{M} = [0,1]$). The dual characterisation (cf. Section 2.5.5) of the Hausdorff lifting due to Mémoli [Mém11], is more convenient for our purposes. Let $u : \mathcal{P}(X\times X) \to \mathcal{P}(X)\times\mathcal{P}(X)$ be defined by $u(C) = (\pi_1[C],\pi_2[C])$, where π_1,π_2 are the projections $\pi_i : X\times X \to X$ and $\pi_i[C] = \{\pi_i(c) \mid c \in C\}$. Then

$$\mathcal{H}(d)(X_1,X_2) = \min\{\max_{(x_1,x_2)\in C} d(x_1,x_2) \mid C \subseteq X\times X \ \wedge\ u(C) = (X_1,X_2)\}.$$

It is easy to see that C is a coupling for X_1,X_2 (as discussed in Section 2.5.5) iff $u(C) = (X_1,X_2)$. Relying on this characterisation, we can obtain the result below, from which we deduce that $\mathcal{H}$ is non-expansive and construct its approximation as the composition of the corresponding functions from Table 3.1.

Lemma 3.6.13 (decomposing $\mathcal{H}$)**.** *It holds that* $\mathcal{H} = \min_u \circ \max_\in$ *where* $\max_\in : \mathbb{M}^{X\times X} \to \mathbb{M}^{\mathcal{P}(X\times X)}$*, with* $\in\, \subseteq (X\times X)\times\mathcal{P}(X\times X)$ *the "is-element-of"-relation on* $X\times X$*, and* $\min_u : \mathbb{M}^{\mathcal{P}(X\times X)} \to \mathbb{M}^{\mathcal{P}(X)\times\mathcal{P}(X)}$.

Proof. Let for $d : X\times X \to \mathbb{M}$, $X_1,X_2 \subseteq X$. Then we have

$$\min_u(\max_\in(d))(X_1,X_2)$$

$$= \min_{u(C)=(X_1,X_2)} (\max_{\in}(d))(C) = \min_{u(C)=(X_1,X_2)} \max_{(x_1,x_2)\in C} d(x_1,x_2)$$

which is exactly the definition of the Hausdorff lifting $\mathcal{H}(d)(X_1,X_2)$ via couplings, due to Mémoli [Mém11]. □

We next determine the approximation of the Hausdorff lifting in the dual sense. Intuitively, given a distance function d and a relation R on X, such function characterises those pairs (X_1,X_2), $X_1,X_2 \subseteq X$, whose distance in the Hausdorff metric decreases by a constant when we decrease the distance d for all pairs in R by the same constant.

Lemma 3.6.14 (approximating $\mathcal{H}$)**.** *The approximation for the Hausdorff lifting $\mathcal{H}$ in the dual sense is as follows. Let $d: X \times X \to \mathbb{M}$, then $\mathcal{H}^d_\#: \mathcal{P}([X \times X]^d) \to \mathcal{P}([\mathcal{P}(X) \times \mathcal{P}(X)]^{\mathcal{H}(d)})$ with*

$$\begin{aligned}
\mathcal{H}^d_\#(R) \quad = \quad & \{(X_1,X_2) \in [\mathcal{P}(X) \times \mathcal{P}(X)]^{\mathcal{H}(d)} \mid \\
& \forall x_1 \in X_1 \big(\min_{x_2' \in X_2} d(x_1,x_2') = \mathcal{H}(d)(X_1,X_2) \Rightarrow \exists x_2 \in X_2: \\
& \qquad (x_1,x_2) \in R \wedge d(x_1,x_2) = \mathcal{H}(d)(X_1,X_2)\big) \wedge \\
& \forall x_2 \in X_2 \big(\min_{x_1' \in X_1} d(x_1',x_2) = \mathcal{H}(d)(X_1,X_2) \Rightarrow \exists x_1 \in X_1: \\
& \qquad (x_1,x_2) \in R \wedge d(x_1,x_2) = \mathcal{H}(d)(X_1,X_2)\big)\}
\end{aligned}$$

Proof. See Appendix: Lemma A.2.10. □

Remark 3.6.15. *The descent constant $\iota^d_{\mathcal{H}}$ has as lower bound:*

$$\iota^d_{\mathcal{H}} \sqsupseteq \min\{\{d(x_1,x_2) - d(y_1,y_2) \mid d(x_1,x_2) - d(y_1,y_2) > 0,\ x_1,x_2,y_1,y_2 \in X\} \cup \{\delta^d\}\}.$$

3.6.5. Deriving the Approximation for Behavioural Distances of Probabilistic Automata

The *probabilistic bisimilarity pseudo-metric* of a probabilistic automaton is the least fixpoint of the function $\mathcal{M}$: $[0,1]^{S\times S} \to [0,1]^{S\times S}$, defined for $d: S \times S \to [0,1]$ and $s,t \in S$ as

$$\mathcal{M}(d)(s,t) = \max\{d_L(\ell(s),\ell(t)),\ \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t))\}$$

where $\mathcal{H}$ is the Hausdorff lifting and $\mathcal{K}$ is the Kantorovich lifting ($\mathbb{M} = [0,1]$), cf. Section 2.6.5.

The fixpoint function $\mathcal{M}$ can be expressed as the composition of basic non-expansive functions and thus, by Theorem 3.5.8, it is non-expansive itself.

Lemma 3.6.16 (decomposing $\mathcal{M}$)**.** *The fixpoint function $\mathcal{M}$ for probabilistic bisimilarity pseudo-metrics can be written as:*

$$\mathcal{M} = \max_\rho \circ (((\eta \times \eta)^* \circ \mathcal{H} \circ \mathcal{K}) \uplus ((\ell \times \ell)^* \circ c_{d_L}))$$

where $\rho\colon (S \times S) \uplus (S \times S) \to (S \times S)$ *with* $\rho((s,t),i) = (s,t)$.

Proof. In fact, given $d\colon S \times S \to [0,1]$ and $s,t \in S$, we have

$$\begin{aligned} & \max_\rho((((\eta \times \eta)^* \circ \mathcal{H} \circ \mathcal{K}) \uplus ((\ell \times \ell)^* \circ c_{d_L}))(d))(s,t) \\ = \ & \max\{(\eta \times \eta)^* \circ \mathcal{H} \circ \mathcal{K})(d)(s,t), ((\ell \times \ell)^* \circ d_L)(s,t)\} \\ = \ & \max\{\mathcal{H}(\mathcal{K}(d)(\eta(s),\eta(t)), d_L(\ell(s),\ell(t))\} \\ = \ & \mathcal{M}(d)(s,t) \end{aligned}$$

□

As discussed in Section 2.6.5, whenever d_L is discrete, this specializes to the probabilistic automata of [BBL+21] and whenever the probability distributions are Dirac distributions we obtain metric transition systems [dAFS09].

The above decomposition also helps in determining the approximation of $\mathcal{M}$.

Lemma 3.6.17 (approximating $\mathcal{M}$)**.** *Let $d\colon S \times S \to [0,1]$. The approximation for $\mathcal{M}$ in the dual sense is $\mathcal{M}^d_\#\colon \mathcal{P}([S \times S]^d) \to \mathcal{P}([S \times S]^{\mathcal{M}(d)})$ with*

$$\begin{aligned} \mathcal{M}^d_\#(X) = \{(s,t) \in [S \times S]^{\mathcal{M}(d)} \mid \ & d_L(\ell(s),\ell(t)) < \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t)) \\ & \wedge (\eta(s),\eta(t)) \in \mathcal{H}^{\mathcal{K}(d)}_\# \circ \mathcal{K}^d_\#(X)\} \end{aligned}$$

Proof. See Appendix: Lemma A.2.11. □

Remark 3.6.18. *We can bound the descent constant $\iota^d_\mathcal{M}$ by:*

$$\iota^d_\mathcal{M} \sqsupseteq \min\{\{d(x_1,x_2) - d(y_1,y_2) \mid d(x_1,x_2) - d(y_1,y_2) > 0,\ x_1,x_2,y_1,y_2 \in X\} \cup \{\delta^d\}\}.$$

Comparison with [BBL+21]. The paper [BBL+21] describes the first method for computing behavioural distances over probabilistic automata. Although the behavioural distance arises as a least fixpoint, it is in fact better, even the only known method, to iterate from above, in order to reach this least fixpoint. This is done by guessing and improving couplings, similarly to what happens for strategy iteration discussed later in Chapter 5. A major complication, faced in [BBL+21], is that the procedure can get stuck at a fixpoint which is not the least and one has to determine that this is the case and decrease the current candidate. This is done by relying on an adaptation of the notion of self-closed relation from [Fu12], and next we argue that this is closely related to the theory we developed in this chapter. In fact this was our inspiration to generalise this technique to a more general setting.

We next establish a formal correspondence with our results. First, note that the probabilistic automata considered in [BBL+21] are a special case of those defined above, where the metric on the set of state labels is required to be discrete. Hence states with different labels are necessarily at distance 1.

Let $\mathsf{PA} = (S, \eta, L, \ell)$ be a fixed probabilistic automata and let us assume that, as in [BBL+21], the metric space of labels (L, d_L) is discrete.

Assume that d is a fixpoint of $\mathcal{M}$, i.e., $d = \mathcal{M}(d)$. In order to check whether $d = \mu\mathcal{M}$, [BBL+21] adapts the notion of a self-closed relation from [Fu12].

Definition 3.6.19 ([BBL+21])**.** *A relation $M \subseteq S \times S$ is* self-closed *with respect to $d = \mathcal{M}(d)$ if, whenever $s\, M\, t$, then*

- *$\ell(s) = \ell(t)$ and $d(s,t) > 0$,*
- *if $p \in \eta(s)$ and $d(s,t) = \min_{q' \in \eta(t)} \mathcal{K}(d)(p, q')$, then there exists $q \in \eta(t)$ and $c \in \Omega(p,q)$ such that $d(s,t) = \sum_{u,v \in S} d(u,v) \cdot c(u,v)$ and $supp(c) \subseteq M$,*
- *if $q \in \eta(t)$ and $d(s,t) = \min_{p' \in \eta(s)} \mathcal{K}(d)(p', q)$, then there exists $p \in \eta(s)$ and $c \in \Omega(p,q)$ such that $d(s,t) = \sum_{u,v \in S} d(u,v) \cdot c(u,v)$ and $supp(c) \subseteq M$.*

The largest self-closed relation, denoted by $\approx_d$, can be shown to be empty if and only if $d = \mu\mathcal{M}$ [BBL+21]. This has an immediate correspondence with our results since we can prove an intimate connection between self-closed relations and post-fixpoints of the approximation of $\mathcal{M}$.

Proposition 3.6.20. *Let $d \colon S \times S \to [0,1]$ where $d = \mathcal{M}(d)$. Then $\mathcal{M}^d_\#\colon \mathcal{P}([S \times S]^d) \to \mathcal{P}([S \times S]^d)$, where $[S \times S]^d = \{(s,t) \in S \times S \mid d(s,t) > 0\}$. Then M is a self-closed relation with respect to d if and only if $M \subseteq [S \times S]^d$ and M is a post-fixpoint of $\mathcal{M}^d_\#$.*

Proof. See Appendix: Proposition A.2.12. □

3.6.6. Deriving the Approximation for Behavioural Distances of Metric Transition Systems

The least fixpoint of the function $\mathcal{J}\colon [0,1]^{S \times S} \to [0,1]^{S \times S}$, defined as

$$\mathcal{J}(d)(s,t) = \max\{d_L(\ell(s), \ell(t)), \mathcal{H}(d)(\eta(s), \eta(t))$$

for $d \in [0,1]^{S \times S}$ and $s, t \in S$ ($\mathbb{M} = [0,1]$), specifies the behavioural distance in a metric transition system. As discussed in Section 2.6.4, any metric transition system $\mathsf{MTS} =$

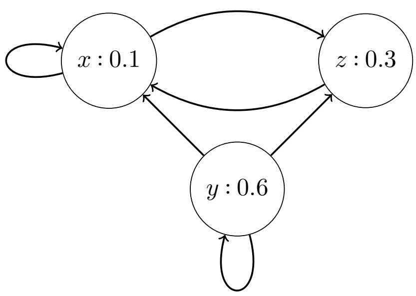
(S, η, ℓ, L) can be seen as a special probabilistic automata, i.e. we can "transform" the transition relation $\eta: S \to \mathcal{P}(S)$ into $\eta': S \to \mathcal{P}(\mathcal{D}(S))$, defining $\eta'(s) = \{\beta_t \mid t \in \eta(s)\}$ (β_t denotes the Dirac distribution). Using this observation, Lemma 3.6.17 and the fact that for a distance $d : S \times S \to [0,1]$ and a pair of states $s,t \in S$, it holds $\mathcal{K}(d)(\beta_s, \beta_t) = d(s,t)$, we obtain the approximation:

$$\mathcal{J}^d_\#(X) = \{(s,t) \in [S \times S]^{\mathcal{J}(d)} \mid \; d_L(\ell(s), \ell(t)) < \mathcal{H}(d)(\eta(s), \eta(t)) \wedge (\eta(s), \eta(t)) \in \mathcal{H}^d_\#(X)\}$$

Remark 3.6.21. *The descent constant $\iota^d_{\mathcal{J}}$ is bounded as follows:*

$$\iota^d_{\mathcal{J}} \sqsupseteq \min\{\{d(x_1,x_2) - d(y_1,y_2) \mid d(x_1,x_2) - d(y_1,y_2) > 0, \; x_1,x_2,y_1,y_2 \in X\} \cup \{\delta^d\}\}.$$

Example 3.6.22. *We consider the metric transition system depicted below, where the metric space of labels is the real interval $[0,1]$ with the Euclidean distance $d_L(x,y) = |x-y|$.*



Here, $\eta(x) = \{x,z\}$, $\eta(y) = \{x,y,z\}$ and $\eta(z) = \{x\}$. Additionally we have $\ell(x) = 0.1$, $\ell(y) = 0.6$ and $\ell(z) = 0.3$ resulting in $d_L(\ell(x), \ell(y)) = 0.5$, $d_L(\ell(x), \ell(z)) = 0.2$ and $d_L(\ell(y), \ell(z)) = 0.3$. The least fixpoint of $\mathcal{J}$ is a pseudo-metric $\mu\mathcal{J}$ given by $\mu\mathcal{J}(x,y) = \mu\mathcal{J}(y,z) = 0.5$ and $\mu\mathcal{J}(x,z) = 0.3$. (Since $\mu\mathcal{J}$ is a pseudo-metric, the remaining entries are fixed: $\mu\mathcal{J}(u,u) = 0$ and $\mu\mathcal{J}(u,v) = \mu\mathcal{J}(v,u)$ for all $u,v \in \{x,y,z\}$.)

Now consider the pseudo-metric d with $d(x,y) = d(x,z) = d(y,z) = 0.5$. This is also a fixpoint of $\mathcal{J}$. Note that $\mathcal{H}(d)(\eta(x),\eta(y)) = \mathcal{H}(d)(\eta(x),\eta(z)) = \mathcal{H}(d)(\eta(y),\eta(z)) = 0.5$. Let us use our technique in order to verify that d is not the least fixpoint of $\mathcal{J}$, by showing that $\nu\mathcal{J}^d_\# \neq \emptyset$.

We start the fixpoint iteration with the approximation $\mathcal{J}^d_\#$ from the top element $[S \times S]^d$, which is given by the symmetric closure[4] of $\{(x,y),(x,z),(y,z)\}$ (since reflexive pairs do not contain slack).

We first observe that the pairs $(x,y),(y,x) \notin \mathcal{J}^d_\#(Sym(\{(x,y),(x,z),(y,z)\}))$ since $d_L(\ell(x),\ell(y)) = 0.5 \not< \mathcal{H}(d)(\eta(x),\eta(y)) = 0.5$.

[4]We denote the symmetric closure of a relation R by $Sym(R)$.

Next, we have $(y,z),(z,y) \notin \mathcal{J}^d_\#(Sym(\{(x,z),(y,z)\}))$ *since it holds* $(\eta(y),\eta(z)) \notin \mathcal{H}^d_\#(Sym(\{(x,z),(y,z)\}))$*. In order to see this, consider the approximation of the Hausdorff lifting in Lemma 3.6.14 and note that for* $y \in \eta(y)$ *we have* $\min_{u\in\eta(z)} d(y,u) = 0.5 = \mathcal{H}(d)(\eta(y),\eta(z))$*, but* $(y,x) \notin Sym(\{(x,z),(y,z)\})$ *(where* x *is the only element in* $\eta(z)$*).*

The pairs $(x,z),(z,x)$ *on the other hand satisfy all conditions and hence*

$$\nu\mathcal{J}^d_\# = Sym(\{(x,z)\}) = \mathcal{J}^d_\#(Sym(\{(x,z)\})) \neq \emptyset$$

Thus we conclude that d *is not the least fixpoint, but, according to Proposition 3.3.5, we can decrease the value of* d *in the positions* $(x,z),(z,x)$ *and obtain a pre-fixpoint from which we can continue the fixpoint iteration.*

3.6.7. Deriving the Approximation for Bisimilarity of Transition Systems

In order to define standard bisimilarity (see Section 2.6.2) we use a variant $\mathcal{G}$ of the Hausdorff lifting $\mathcal{H}$ defined before, where max and min are swapped. More precisely, $\mathcal{G} : \{0,1\}^{X\times X} \to \{0,1\}^{\mathcal{P}(X)\times\mathcal{P}(X)}$ is defined, for $a \in \{0,1\}^{X\times X}$, by

$$\mathcal{G}(a)(X_1,X_2) = \max_u\{\min_{(x_1,x_2)\in C} a(x_1,x_2) \mid C \subseteq X\times X \ \wedge\ u(C) = (X_1,X_2)\}.$$

Lemma 3.6.23 (approximating $\mathcal{G}$)**.** *The approximation for the adapted Hausdorff lifting* $\mathcal{G}$ *is as follows. Let* $a: X\times X \to \{0,1\}$*, then* $\mathcal{G}^\#_a : \mathcal{P}([X\times X]_a) \to \mathcal{P}([\mathcal{P}(X)\times\mathcal{P}(X)]_{\mathcal{G}(a)})$ *with*

$$\begin{aligned}\mathcal{G}^\#_a(R) \quad = \quad \{(X_1,X_2) \in [\mathcal{P}(X)\times\mathcal{P}(X)]_{\mathcal{G}(a)} \mid \\ \forall x_1 \in X_1 \exists x_2 \in X_2 \colon \big((x_1,x_2) \notin [X\times X]_a \vee (x_1,x_2) \in R\big) \\ \wedge \forall x_2 \in X_2 \exists x_1 \in X_1 \colon \big((x_1,x_2) \notin [X\times X]_a \vee (x_1,x_2) \in R\big)\}\end{aligned}$$

Proof. See Appendix: Lemma A.2.13. □

Now we can define the fixpoint function for bisimilarity and its corresponding approximation. The fixpoint function for bisimilarity $\mathcal{B} : \{0,1\}^{X\times X} \to \{0,1\}^{X\times X}$[5] can be expressed by using the Hausdorff lifting $\mathcal{G}$ with $\mathbb{M} = \{0,1\}$.

Lemma 3.6.24 (decomposing $\mathcal{B}$)**.** *Bisimilarity for a transition system* $\mathsf{TS} = (X,\eta)$ *is the greatest fixpoint of* $\mathcal{B} = (\eta\times\eta)^* \circ \mathcal{G}$.

Proof. See Appendix: Lemma A.2.14. □

[5] We change S to X in this section.

Since we are interested in the greatest fixpoint, we are working in the primal sense. Bisimulation relations are represented by their characteristic functions $a: X \times X \to \{0,1\}$, in fact the corresponding relation can be obtained by taking the complement of $[X \times X]_a = \{(x_1, x_2) \in X_1 \times X_2 \mid a(x_1, x_2) = 0\}$.

Lemma 3.6.25 (approximating $\mathcal{B}$). *Let $a: X \times X \to \{0,1\}$. The approximation for the bisimilarity function $\mathcal{B}$ in the primal sense is $\mathcal{B}_a^\#: \mathcal{P}([X \times X]_a) \to \mathcal{P}([X \times X]_{\mathcal{B}(a)})$ with*

$$\begin{aligned} \mathcal{B}_a^\#(R) \quad = \quad & \{(x_1, x_2) \in [X \times X]_{\mathcal{B}(a)} \mid \\ & \quad \forall y_1 \in \eta(x_1) \exists y_2 \in \eta(x_2) \big((y_1, y_2) \notin [X \times X]_a \vee (y_1, y_2) \in R\big) \\ & \wedge \forall y_2 \in \eta(x_2) \exists y_1 \in \eta(x_1) \big((y_1, y_2) \notin [X \times X]_a \vee (y_1, y_2) \in R\big)\} \end{aligned}$$

Proof. See Appendix: Lemma A.2.15. □

Remark 3.6.26. *Clearly, we have the ascent constant*

$$\iota_a^{\mathcal{B}} = 1.$$

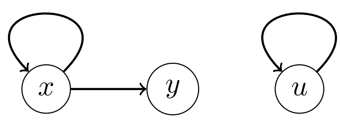
We conclude this section by discussing how this view on bisimilarity can be useful: first, it again opens up the possibility to compute bisimilarity – a greatest fixpoint – by iterating from below, through smaller fixpoints. This could potentially be useful if it is easy to compute the least fixpoint of $\mathcal{B}$ inductively and continue from there.

Furthermore, we obtain a technique for witnessing non-bisimilarity of states. While this can also be done by exhibiting a distinguishing modal formula [HM85, Cle90] or by a winning strategy for the spoiler in the bisimulation game [Sti97], to our knowledge there is no known method that does this directly, based on the definition of bisimilarity.

With our technique we can witness non-bisimilarity of two states $x_1, x_2 \in X$ by presenting a pre-fixpoint a (i.e., $\mathcal{B}(a) \le a$) such that $a(x_1, x_2) = 0$ (equivalent to $(x_1, x_2) \in [X \times X]_a$) and $\nu \mathcal{B}_a^\# = \emptyset$, since this implies $\nu\mathcal{B}(x_1, x_2) \le a(x_1, x_2) = 0$ by our proof rule.

There are two issues to discuss: first, how can we characterise a pre-fixpoint of $\mathcal{B}$ (which is quite unusual, since bisimulations are post-fixpoints)? In fact, the condition $\mathcal{B}(a) \le a$ can be rewritten to: for all $(x_1, x_2) \in [X \times X]_a$ there exists $y_1 \in \eta(x_1)$ such that for all $y_2 \in \eta(x_2)$ we have $(y_1, y_2) \in [X \times X]_a$ (*or* vice versa). Second, at first sight it does not seem as if we gained anything since we still have to do a fixpoint computation on relations. However, the carrier set is $[X \times X]_a$, i.e., a set of non-bisimilarity witnesses and this set can be small even though X might be large, since a might have value 0 only on a small subset of $X \times X$.

Example 3.6.27. *We consider the transition system depicted below.*



Our aim is to construct a witness showing that x, u are not bisimilar. This witness is a function $a: X \times X \to \{0,1\}$ with $a(x,u) = 0 = a(y,u)$ and for all other pairs the value is 1. Hence $[X \times X]_{a=\mathcal{B}(a)} = [X \times X]_a = \{(x,u),(y,u)\}$ and it is easy to check that a is a pre-fixpoint of $\mathcal{B}$ and that $\nu\mathcal{B}_a^* = \emptyset$: we iterate over $\{(x,u),(y,u)\}$ and first remove (y,u) (since y has no successors) and then (x,u). This implies that $\nu\mathcal{B} \le a$ and hence $\nu\mathcal{B}(x,u) = 0$, which means that x,u are not bisimilar.

Example 3.6.28. *We modify Example 3.6.27 and consider a function a where $a(x,u) = 0$ and all other values are 1. Again a is a pre-fixpoint of $\mathcal{B}$ and $\nu\mathcal{B} \le a$ (since only reflexive pairs are in the bisimilarity). However $\nu\mathcal{B}_a^* \neq \emptyset$, since $\{(x,u)\}$ is a post-fixpoint. This is a counterexample to completeness discussed after Theorem 3.3.7.*

Intuitively speaking, the states y,u over-approximate and claim that they are bisimilar, although they are not. (This is permissible for a pre-fixpoint.) This tricks x,u into thinking that there is some wiggle room and that one can increase the value of (x,u). This is true, but only because of the limited, local view, since the "true" value of (y,u) is 0.

3.6.8. Deriving the Approximation for Simple Stochastic Games

In this section we show that $\mathcal{V}: [0,1]^V \to [0,1]^V$ is non-expansive and derive the approximation $\mathcal{V}_\#^a$. $\mathcal{V}$ was defined for $a: V \to [0,1]$ and $v \in V$ as follows:

$$\mathcal{V}(a)(v) = \begin{cases} \max_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

Here, $\mathbb{M} = [0,1]$. The least fixpoint of $\mathcal{V}$ yields the solution of the underlying simple stochastic game (cf. Section 2.7.3). Before we disassemble $\mathcal{V}$ into smaller subfunctions we need to disjoint the set of successors. Thus we rewrite $\mu\mathcal{V}$ as follows

$$\mathcal{V}(a)(v) = \begin{cases} \max_{v' \in \eta_{\max}(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in \eta_{\min}(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} \eta_{av}(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ w(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

where $\eta_{\max}: V_{\mathsf{Max}} \to \mathcal{P}(V)$, $\eta_{\min}: V_{\mathsf{Min}} \to \mathcal{P}(V)$, $\eta: V_{\mathsf{Av}} \to \mathcal{D}(V)$ and $w: V_{\mathsf{Sink}} \to [0,1]$ (renamed c to w in order to avoid confusion). In order to be able to determine the approximation of $\mathcal{V}$ and to apply our techniques, we consider the following equivalent definition.

Lemma 3.6.29 (decomposing $\mathcal{V}$)**.** $\mathcal{V} = (\eta_{\min}^* \circ \min_\in) \uplus (\eta_{\max}^* \circ \max_\in) \uplus (\eta_{\text{av}}^* \circ \text{av}_D) \uplus c_w$, *where $\in \subseteq V \times \mathcal{P}(V)$ is the "is-element-of"-relation on V.*

Proof. Let $a\colon V \to [0,1]$. For $v \in V_{\mathsf{Max}}$ we have

$$\mathcal{V}(a)(v) = (\eta_{\max}^* \circ \max_\epsilon)(a)(v) = \max_\epsilon(a)(\eta_{\max}(v)) = \max_{v' \in \eta_{\max}(v)} a(v').$$

For $v \in V_{\mathsf{Min}}$ we have

$$\mathcal{V}(a)(v) = (\eta_{\min}^* \circ \min_\epsilon)(a)(v) = \min_\epsilon(a)(\eta_{\min}(v)) = \min_{v' \in \eta_{\min}(v)} a(v').$$

For $v \in V_{\mathsf{Av}}$ we have

$$\mathcal{V}(a)(v) = (\eta_{\mathrm{av}}^* \circ \mathrm{av}_D)(a)(v) = \mathrm{av}_D(a)(\eta_{\mathrm{av}}(v)) = \sum_{v' \in V} \eta_{\mathrm{av}}(v)(v') \cdot a(v').$$

For $v \in V_{\mathsf{Sink}}$ we have $\mathcal{V}(a)(v) = c_w(a)(v) = w(v)$. □

As a composition of non-expansive functions, $\mathcal{V}$ is non-expansive as well. Since we are interested in the least fixpoint we work in the dual sense and obtain the following approximation, which intuitively says: we can decrease a value at node v by a constant only if, in the case of a Min node, we decrease the value of one successor where the minimum is reached, in the case of a Max node, we decrease the values of all successors where the maximum is reached, and in the case of an average node, we decrease the values of all successors.

Lemma 3.6.30 (approximating $\mathcal{V}$)**.** *Let $a\colon V \to [0,1]$. The approximation for the value iteration function $\mathcal{V}$ in the dual sense is $\mathcal{V}^a_\#\colon \mathcal{P}([V]^a) \to \mathcal{P}([V]^{\mathcal{V}(a)})$ with*

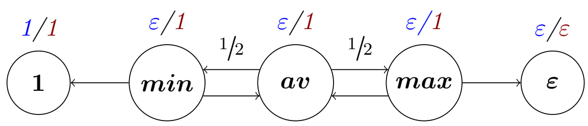
$$\begin{aligned}\mathcal{V}^a_\#(V') \quad = \quad & \{v \in [V]^{\mathcal{V}(a)} \mid \big(v \in V_{\mathit{Min}} \wedge \arg\min_{v' \in \eta_{\min}(v)} a(v') \cap V' \neq \emptyset\big) \vee \\ & \big(v \in V_{\mathit{Max}} \wedge \arg\max_{v' \in \eta_{\max}(v)} a(v') \subseteq V'\big) \vee \big(v \in V_{\mathit{Av}} \wedge \mathit{supp}(\eta_{\mathrm{av}}(v)) \subseteq V'\big)\}\end{aligned}$$

Proof. See Appendix: Lemma A.2.16. □

Remark 3.6.31. *A bound for the descent constant $\iota^a_{\mathcal{V}}$ is given by:*

$$\iota^a_{\mathcal{V}} \sqsupseteq \min\{\{a(v) - a(v') \mid a(v) - a(v') > 0,\ v, v' \in V\} \cup \{\delta^a\}\}.$$

Example 3.6.32. *We consider the game depicted below. Here* **min** *is a Min node with* $\eta_{\min}(\mathrm{min}) = \{\mathbf{1}, \boldsymbol{av}\}$*,* **max** *is a Max node with* $\eta_{\max}(\mathrm{max}) = \{\boldsymbol{\varepsilon}, \boldsymbol{av}\}$*,* **1** *is a sink node with payoff 1,* $\boldsymbol{\varepsilon}$ *is a sink node with some small payoff* $\varepsilon \in (0,1)$ *and* $\boldsymbol{av}$ *is an average node which transitions to both* min *and* max *with probability* $1/2$*. The least fixpoint* $\mu\mathcal{V}$ *is given in blue (left value). This game is not stopping.*



Assume we are given the greatest fixpoint $\nu\mathcal{V}$ (in red, right value). One can easily detect the vicious cycle formed by $V' = \{\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}\}$ as $\boldsymbol{av} \in V'$ ($\boldsymbol{av}$ is a successor of $\boldsymbol{min}$ attaining the minimum and the only successor of $\boldsymbol{max}$ attaining the maximum) and $supp(\eta_{\mathrm{av}}) = \{\boldsymbol{max}, \mathbf{min}\} \subseteq V'$.

3.6.9. Deriving the Approximation for Energy Games

In this section we show that $\mathcal{E} : K^V \to K^V$, defined as

$$\mathcal{E}(a)(v) = \begin{cases} \min\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v,v') & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v,v') & \text{if } v \in V_{\mathsf{Max}} \end{cases}$$

for $a \in K^V$ and $v \in V$, is non-expansive and we derive the approximation $\mathcal{E}^a_\#$. The least fixpoint of $\mathcal{E}$ yields the solution of an energy games with finite values (cf. Section 2.7.4). It holds $K = \{0, 1, \dots, k\}$ for some $k \in \mathbb{N}$ and the operator $\ominus_{\mathbb{Z}}: K \times \mathbb{Z} \to K$ was defined as $x \ominus_{\mathbb{Z}} y = \min\{\max\{x - y, 0\}, k\}$.

First, we disassemble $\mathcal{E}$ into smaller functions. We define

- $E_0 = \{(v,v') \in E \mid v \in V_{\mathsf{Min}}\}$ and $E_1 = \{(v,v') \in E \mid v \in V_{\mathsf{Max}}\}$.
  Immediately we conclude $E_0 \cup E_1 = E$ and $E_0 \cap E_1 = \emptyset$.
- *Projections:* $\pi_i: E \to V$, $i = 1, 2$, where, given $e = (v, v') \in E$ we have $\pi_1(e) = v$ and $\pi_2(e) = v'$.
- *Restricted projections:* We define $\pi_i^j: E_j \to V$ where $\pi_i^j = (\pi_i)_{|E_j}$ for $i \in \{1,2\}$, $j \in \{0,1\}$.
- *Subtraction of edge weights:* Given $w: E \to Z$, define $sub'_w: K^E \to K^E$ via
  $$sub'_w(a)(e) = a(e) \ominus_{\mathbb{Z}} w(e)$$
  for $a: E \to K$ and $e \in E$.
- *Minimum and maximum functions:* We use the functions $\min_u$, $\max_u$ from Table 3.1, where u is one of the projections defined above.

Lemma 3.6.33 (decomposing $\mathcal{E}$)**.** *The function $\mathcal{E}: K^V \to K^V$ can be written as*

$$\mathcal{E} = (\min_{\pi_1^0} \uplus \max_{\pi_1^1}) \circ sub'_w \circ \pi_2^*$$

Proof. Given $a: V \to K$ and $v \in V$, we get

$$\big((\min_{\pi_1^0} \uplus \max_{\pi_1^1}) \circ sub'_w \circ \pi_2^*\big)(a)(v) = \begin{cases} \min\limits_{\pi_1^0(e)=v} (sub'_w \circ \pi_2^*)(a)(e) \\ \max\limits_{\pi_1^1(e)=v} (sub'_w \circ \pi_2^*)(a)(e) \end{cases}$$

$$
\begin{aligned}
&= \begin{cases} \min\limits_{(v,v')\in E} sub'_w \circ \pi_2^*(a)(v,v'), & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{(v,v')\in E} sub'_w \circ \pi_2^*(a)(v,v'), & \text{if } v \in V_{\mathsf{Max}} \end{cases} \\
&= \begin{cases} \min\limits_{(v,v')\in E} \pi_2^*(a)(v,v') \ominus_{\mathbb{Z}} w(v,v'), & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{(v,v')\in E} \pi_2^*(a)(v,v') \ominus_{\mathbb{Z}} w(v,v'), & \text{if } v \in V_{\mathsf{Max}} \end{cases} \\
&= \begin{cases} \min\limits_{(v,v')\in E} a(v') \ominus_{\mathbb{Z}} w(v,v'), & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{(v,v')\in E} a(v') \ominus_{\mathbb{Z}} w(v,v'), & \text{if } v \in V_{\mathsf{Max}} \end{cases} \\
&= \mathcal{E}(a)(v)
\end{aligned}
$$

□

Non-Expansiveness of sub'_w and Approximation $(sub'_w)^a_\#$. We remark that sub'_w differs from the basic function sub_w from Definition 3.5.3 as w maps to $\mathbb{Z}$ and not to $\mathbb{M} = K$. Thus we show non-expansiveness and derive the approximation $(sub'_w)^a_\#$.

Lemma 3.6.34. *The function $sub'_w\colon K^E \to K^E$, defined via $sub'_w(a)(e) = a(e) \ominus_{\mathbb{Z}} w(e)$ for $a\colon E \to K$, $e \in E$ and $w\colon E \to \mathbb{Z}$, is non-expansive.*

Proof. See Appendix: Lemma A.2.17. □

Next we determine the approximation $(sub'_w)^a_\#$.

Lemma 3.6.35. *Given $w\colon E \to \mathbb{Z}$ and $a\colon E \to K$ the approximation $(sub'_w)^a_\#\colon K^{[E]^a} \to K^{[E]^{sub'_w(a)}}$ of $sub'_w\colon K^E \to K^E$, is given by*

$$(sub'_w)^a_\#(E') = \{e \in E' \mid 0 < a(e) - w(e) \le k\}$$

for $E' \subseteq [E]^a$.

Proof. See Appendix: Lemma A.2.18. □

By above considerations, the function $\mathcal{E} : K^V \to K^V$ is clearly non-expansive and we are able to derive the approximation $\mathcal{E}^a_\#$.

Lemma 3.6.36 (approximating $\mathcal{E}$). *Let $V' \subseteq [V]^a$ then $v \in \mathcal{E}^a_\#(V')$ if $v \in [V]^{\mathcal{E}(a)}$ and*

- *whenever $v \in V_{\mathsf{Min}}$ there exists some $(v, v'') \in E$ with $\min_{(v,v')\in E} a(v') \ominus_{\mathbb{Z}} w(v,v') = a(v'') \ominus_{\mathbb{Z}} w(v,v'')$, $0 < a(v'') - w(v,v'') \leq k$ and $v'' \in V'$*
- *whenever $v \in V_{\mathsf{Max}}$: if $(v, v'') \in E$ with $\max_{(v,v')\in E} a(v') \ominus_{\mathbb{Z}} w(v,v') = a(v'') \ominus_{\mathbb{Z}} w(v,v'')$ then $0 < a(v'') - w(v,v'') \leq k$ and $v'' \in V'$*

Proof. See Appendix: Lemma A.2.19. ☐

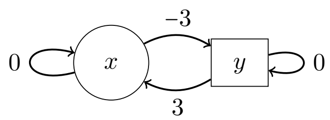
Remark 3.6.37. *We have the following lower bound for the descent constant $\iota^a_{\mathcal{E}}$:*

$$\iota^a_{\mathcal{E}} \sqsupseteq 1.$$

On fixpoints $a = \mathcal{E}(a)$ the above characterization simplifies to $v \in \mathcal{E}^a_\#(V')$ if $v \in [V]^a$ and

- $v \in V_{\mathsf{Min}}$: there exists some $(v, v') \in E$ with $a(v) = a(v') \ominus_{\mathbb{Z}} w(v,v')$, $0 < a(v') - w(v,v') \leq k$ and $v' \in V'$
- $v \in V_{\mathsf{Max}}$: if $(v, v') \in E$ with $a(v) = a(v') \ominus_{\mathbb{Z}} w(v,v')$ then $0 < a(v') - w(v,v') \leq k$ and $v' \in V'$

Example 3.6.38. *Consider the following energy game, where it is intended that circular and rectangular states belong to Player Min and Player Max, respectively. Rather immediate, $\mu\mathcal{E}(x) = \mu\mathcal{E}(y) = 0$.*



We are given the fixpoint $a(x) = 10$ and $a(y) = 7$ of $\mathcal{E}$. We have $\nu\mathcal{E}^a_\# = \{x, y\}$. Additionally, $V' = \{x\}$ is also a fixpoint of $\mathcal{E}^a_\#$ as $a(x) = a(x) \ominus_{\mathbb{Z}} w(x,x)$. On the other hand, $V'' = \{y\}$ is not a fixpoint of $\mathcal{E}^a_\#$ since $a(y) = a(x) \ominus_{\mathbb{Z}} w(y,x)$ but $x \notin V''$.

3.7. Summary and Outlook

In this chapter we developed a theory which allows us to detect whether some fixpoint of an endo-function f is its greatest fixpoint or not. This is done by detecting "vicious cycles" in the system, i.e. states that convince each other that their value is lower than it should be. This works for non-expansive functions $f : \mathbb{M}^Y \to \mathbb{M}^Y$, where $\mathbb{M}$ is a complete MV-chain and Y some finite set. We derived the following proof rule: Let $a \in \mathbb{M}^Y$ be a fixpoint of f then it holds

$$a = \nu f \text{ if and only if } \nu f_a^{\#} = \emptyset.$$

The function $f_a^{\#}: \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_a)$ is given by

$$f_a^{\#}(Y') = \{y \in [Y]_a \mid f(a \oplus \iota_{Y'})(y) \ominus a(y) \sqsupseteq \iota\}$$

for some $Y' \subseteq Y$, some suitable small (and non-zero) constant $\iota \in \mathbb{M}$ and $[Y]_a = \{y \in Y \mid a(y) \neq 1\}$. Whenever $a \neq \nu f$ we are able to give some postfixpoint $b \in \mathbb{M}^Y$ of f which lies above a, i.e. $a \sqsubset b \sqsubseteq \nu f$. We derived a similar proof rule for pre-fixpoint and dualized the whole theory for least fixpoints.

We will see in Chapter 5 that these results are very useful. In particular, the correctness of the strategy iterations we will present can only be proven using the results from this chapter.

Non-expansive functions enjoy good closure properties which allowed us to assemble a handful of complex functions from a few basic functions. These closure properties are extremely useful as we will present a tool UDEfix (see Section 4.7) where a user can create his very own functions - based on these few basic functions - and do the fixpoint checks described above. This allowed us to derive the approximations of rather involved functions.

In Section 4.3 we will also derive a categorical framework for the theory developed in this chapter.

4 | A Monoidal View on Fixpoint Checks

This chapter can be seen as an extension of the previous one. We will embed the approximation framework into a categorical setting and present a tool we developed which can be used to perform fixpoint checks.

4.1. Introduction

We show that gs-monoidal categories (cf. Section 2.4.1) and the composition concepts that come with them can be fruitfully used in a scenario that – at first sight – might seem quite unrelated: methods for fixpoints checks. In particular, we build upon Chapter 3 where a theory is proposed for checking whether a fixpoint of a given function is the least (greatest) fixpoint. As we have seen, the theory applies to a variety of fairly diverse application scenarios. We show that the approximation framework and its compositionality properties can be naturally interpreted in categorical terms. This is done by introducing two gs-monoidal categories in which the concrete functions respectively their approximations live as arrows, together with a gs-monoidal functor, called #, mapping one to the other. Besides shedding further light on the theoretical approximation framework of Chapter 3, this view guided the realisation of a tool, called UDEfix that allows to build functions (and their approximations) like a circuit out of basic building blocks and subsequently perform the fixpoints checks.

We also show that the functor # can be extended to deal with functions $f : \mathbb{M}^Y \to \mathbb{M}^Y$ where Y is not necessarily finite, becoming a lax functor. We prove some properties of this functor that enables us to give a recipe for finding approximations for a special type of functions: predicate liftings that have been introduced for coalgebraic modal logic [Pat03, Sch08b]. This recipe allows us to include a new case study for the machinery for fixpoint checking: coalgebraic behavioural metrics, based on Wasserstein liftings (cf. Section 2.5).

The chapter is organized as follows: In Section 4.2 we define the approximation for functions with an infinite domain. Subsequently in Section 4.3 we introduce two (gs-monoidal) categories $\mathbb{C}$, $\mathbb{A}$ (of concrete and abstract functions), show that the approximation # is a (lax) functor between these categories and prove some of its properties, which are used to handle predicate liftings (Section 4.4) and behavioural metrics (Section 4.5). Next, we show that the categories $\mathbb{C}$, $\mathbb{A}$ and the functor # are indeed gs-monoidal (Section 4.6) and lastly discuss the tool UDEfix in Section 4.7. We end by giving a short conclusion (Section 4.8).

4.2. Defining the Appoximation for an Infinite Domain

We here aim to generalize the theory from Chapter 3 which provides the approximations for functions with an infinite domain, i.e. we consider functions $f\colon \mathbb{M}^Y \to \mathbb{M}^Z$ where Y and Z are possibly infinite. For example, the set of probability distributions (with finite support) $\mathcal{D}_f(Y)$ over some set Y is usually infinite and it might occur that intermediate sets are infinite when disassembling the function of interest.

To this end, we only need to adjust the definition of the norm.

Definition 4.2.1 (norm with infinite domain)**.** *Let* $\mathbb{M}$ *be an MV-chain and let* Y *be a possibly infinite set. Given* $a \in \mathbb{M}^Y$ *we define its* norm *as* $\|a\| = \sup\{a(y) \mid y \in Y\}$.

It is imminent that for a finite set Y, the above definition coincides with Definition 2.3.21. A function $f : \mathbb{M}^Y \to \mathbb{M}^Z$ is again *non-expansive* if for all $a, b \in \mathbb{M}^Y$ it holds $\|f(b) \ominus f(a)\| \sqsubseteq \|b \ominus a\|$.

For Y infinite and $0 \sqsubset \delta \in \mathbb{M}$, we can analogously (cf. Section 3.4 as we will consider the dual view in this chapter) define $[Y]^a = \{y \in Y \mid a(y) \neq 0\}$ and the functions $\alpha^{a,\delta} : \mathcal{P}([Y]^a) \to [a \ominus \delta, a]$ and $\gamma^{a,\delta} : [a \ominus \delta, a] \to \mathcal{P}([Y]^a)$ for $Y' \in \mathcal{P}([Y]^a)$ and $b \in [a \ominus \delta, a]$ as

$$\alpha^{a,\delta}(Y') = a \ominus \delta_{Y'} \text{ and } \gamma^{a,\delta}(b) = \{y \in [Y]^a \mid a(y) \ominus b(y) \sqsupseteq \delta\}$$

For a non-expansive function $f : \mathbb{M}^Y \to \mathbb{M}^Z$ and $\delta \in \mathbb{M}$, we define $f_\#^{a,\delta}\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{f(a)})$ as $f_\#^{a,\delta} = \gamma^{f(a),\delta} \circ f \circ \alpha^{a,\delta}$. The function $f_\#^{a,\delta}$ is antitone in the parameter δ and we define the *a-approximation* of f as

$$f_\#^a = \bigcup_{\delta \sqsupset 0} f_\#^{a,\delta}.$$

As a reminder, for finite sets Y and Z there exists a suitable value $\iota_f^a \sqsupset 0$, such that all functions $f_\#^{a,\delta}$ for $0 \sqsubset \delta \sqsubseteq \iota_f^a$ are equal. In this case, the a-approximation is given by $f_\#^a = f_\#^{a,\iota_f^a}$ and it clearly holds

$$f_\#^{a,\iota_f^a} = \bigcup_{\delta \sqsupset 0} f_\#^{a,\delta}$$

For the primal case, we have $f^\#_{a,\delta}\colon \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{f(a)})$ defined as

$$f_a^\# = \bigcup_{\delta \sqsupset 0} f^\#_{a,\delta}.$$

4.3. A Categorical View of the Approximation Framework

The framework from Chapter 3, is not based on category theory, but – as we shall see – can be naturally reformulated in a categorical setting. In particular, casting the compositionality results into a monoidal structure (see Section 4.6) is a valuable basis for our tool. But first, we will show how the operation # of taking the a-approximation of a function can be seen as a (lax) functor between two categories: a concrete category $\mathbb{C}$ whose arrows are the non-expansive functions for which we seek the least (or greatest) fixpoint and an abstract category $\mathbb{A}$ whose arrows are the corresponding approximations.

More precisely, recall from Section 3.2 that given a non-expansive function $f : \mathbb{M}^Y \to \mathbb{M}^Z$, the approximation of f is relative to a fixed map $a \in \mathbb{M}^Y$. Hence objects in $\mathbb{C}$ are elements $a \in \mathbb{M}^Y$ and an arrow from $a \in \mathbb{M}^Y$ to $b \in \mathbb{M}^Z$ is a non-expansive function $f : \mathbb{M}^Y \to \mathbb{M}^Z$ required to map a into b. The approximations instead live in $\mathbb{A}$. Recall that the approximation is $f^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^b)$. Since their domains and codomains are dependent again on a map a, we still employ elements of $\mathbb{M}^Y$ as objects, but functions between powersets as arrows. We refer to Section 2.4 on the basic definitions of category theory.

Definition 4.3.1 (concrete and abstract categories)**.** *The* ***concrete category*** $\mathbb{C}$ *has as objects maps* $a \in \mathbb{M}^Y$ *where* Y *is a (possibly infinite) set. Given* $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ *an arrow* $f : a \dashrightarrow b$ *is a non-expansive function* $f : \mathbb{M}^Y \to \mathbb{M}^Z$*, such that* $f(a) = b$*.*
The ***abstract category*** $\mathbb{A}$ *has again maps* $a \in \mathbb{M}^Y$ *as objects. Given* $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ *an arrow* $f : a \dashrightarrow b$ *is a monotone (wrt. inclusion) function* $f : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^b)$*.*
Arrow composition and identities are the obvious ones.
The ***lax functor*** $\# : \mathbb{C} \to \mathbb{A}$ *is defined as follows: for an object* $a \in \mathbb{M}^Y$*, we let* $\#(a) = a$ *and, given an arrow* $f : a \dashrightarrow b$*, we let* $\#(f) = f^a_\#$*.*

Note that abstract arrows are dashed ($\dashrightarrow$), while the underlying functions are represented by standard arrows ($\to$).

Lemma 4.3.2 (well-definedness)**.** *The categories* $\mathbb{C}$ *and* $\mathbb{A}$ *are well-defined and* # *is a lax functor, i.e., identities are preserved and* $\#(f \circ g) \subseteq \#(f) \circ \#(g)$ *for composable arrows* f, g *in* $\mathbb{C}$*.*

Proof.

1. $\mathbb{C}$ is a well-defined category: Given arrows $f : a \dashrightarrow b$ and $g : b \dashrightarrow c$ then $g \circ f$ is non-expansive (preserved by composition) and $(g \circ f)(a) = g(b) = c$, thus $g \circ f : a \dashrightarrow c$. Associativity holds and the identities are the units of composition as for standard function composition.

2. $\mathbb{A}$ is a well-defined category: Given arrows $f : a \dashrightarrow b$ and $g : b \dashrightarrow c$ then $g \circ f$ is monotone (preserved by composition) and hence $g \circ f : a \dashrightarrow c$.

   Again associativity and the fact that the identities are units is standard.

3. $\#: \mathbb{C} \to \mathbb{A}$ is a lax functor: we first check that identities are preserved. Let $U \subseteq [Y]^a$, then

$$\begin{aligned}
\#(id_a)(U) &= \bigcup_{\delta \sqsupset 0} (id_a)_{\#}^{a,\delta}(U) \\
&= \bigcup_{\delta \sqsupset 0} \{y \in [Y]^{id_a(a)} \mid id_a(a)(y) \ominus id_a(a \ominus \delta_U)(y) \sqsupseteq \delta\} \\
&= \bigcup_{\delta \sqsupset 0} \{y \in [Y]^a \mid a(y) \ominus (a \ominus \delta_U)(y) \sqsupseteq \delta\} \\
&= U && [U \subseteq [Y]^a] \\
&= id_a(U) = id_{\#(a)}(U).
\end{aligned}$$

Let $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$, $c \in \mathbb{M}^V$, $f: a \dashrightarrow b$, $g: b \dashrightarrow c$ be arrows in $\mathbb{C}$ and $Y' \subseteq [Y]^a$. Then, by definition,

$$\begin{aligned}
\#(g \circ f)(Y') &= \bigcup_{\delta \sqsupset 0} (g \circ f)_{\#}^{a,\delta}(Y') \\
&= \bigcup_{\delta \sqsupset 0} (\gamma^{g(f(a)),\delta} \circ g \circ f \circ \alpha^{a,\delta})(Y') \\
&\subseteq \bigcup_{\delta \sqsupset 0} (\gamma^{c,\delta} \circ g \circ \alpha^{b,\delta} \circ \gamma^{b,\delta} \circ f \circ \alpha^{a,\delta})(Y').
\end{aligned}$$

The latter holds since $(f \circ \alpha^{a,\delta})(Y') = f(a \ominus \delta_{Y'}) \sqsupseteq f(a) \ominus \delta = b \ominus \delta$ by non-expansiveness and is hence contained in $[b \ominus \delta, b]$. Then we can use the fact that $\alpha^{b,\delta}, \gamma^{b,\delta}$ is a Galois connection between $[b \ominus \delta, b]$ and $[Z]^b$ for δ small enough (see Lemma 3.2.3) and obtain $id_{[b \ominus \delta, b]} \sqsubseteq \alpha^{b,\delta} \circ \gamma^{b,\delta}$ and the inequality follows from the monotonicity of the functions involved. Hence

$$\begin{aligned}
\bigcup_{\delta \sqsupset 0} \gamma^{c,\delta} \circ g \circ \alpha^{b,\delta} \circ \gamma^{b,\delta} \circ f \circ \alpha^{a,\delta}(Y') &= \bigcup_{\delta \sqsupset 0} g_{\#}^{b,\delta}(f_{\#}^{a,\delta}(Y')) \\
&\subseteq \bigcup_{\delta \sqsupset 0} g_{\#}^{b,\delta}(\bigcup_{\delta' \sqsupset 0} f_{\#}^{a,\delta'}(Y')) \\
&= \#(g) \circ \#(f)(Y')
\end{aligned}$$

where the inequality stems from the monotonicity of $g_{\#}^{b,\delta}$.

□

It will be convenient to restrict to the subcategory of $\mathbb{C}$ where arrows are reindexings and to subcategories of $\mathbb{C}$, $\mathbb{A}$ with maps on finite sets.

Definition 4.3.3 (reindexing subcategory)**.** *We denote by $\mathbb{C}^*$ the lluf*[1] *sub-category of $\mathbb{C}$ where arrows are reindexing, i.e., given objects $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ we consider only arrows $f : a \dashrightarrow b$ such that $f = g^*$ for some $g: Z \to Y$ (hence, in particular, $b = g^*(a) = a \circ g$). We denote $E: \mathbb{C}^* \hookrightarrow \mathbb{C}$ the embedding functor.*

We prove the following auxiliary lemma:

Lemma 4.3.4. *Given $a \in \mathbb{M}^Y$, $g: Z \to Y$ and $0 \sqsubset \delta \in \mathbb{M}$, then we have*

1. $\alpha^{a \circ g, \delta} \circ g^{-1} = g^* \circ \alpha^{a,\delta}$
2. $\gamma^{a \circ g, \delta} \circ g^* = g^{-1} \circ \gamma^{a,\delta}$

This implies that for two $\mathbb{C}$-arrows $f: a \dashrightarrow b$, $h: b \dashrightarrow c$, it holds that $\#(h \circ f) = \#(h) \circ \#(f)$ whenever f or h is a reindexing, i.e., is contained in $\mathbb{C}^*$.

Proof. See Appendix: Lemma A.3.1. □

We can easily define the finite subcategories.

Definition 4.3.5 (finite subcategories)**.** *We denote by $\mathbb{C}_f$, $\mathbb{A}_f$ the full subcategories of $\mathbb{C}, \mathbb{A}$ where objects are of the kind $a \in \mathbb{M}^Y$ for a* finite *set Y.*

Lemma 4.3.6. *The lax functor $\#: \mathbb{C} \to \mathbb{A}$ restricts to $\#: \mathbb{C}_f \to \mathbb{A}_f$, which is a (proper) functor.*

Proof. Clearly the restriction to categories based on finite sets is well-defined.

We show that $\#$ is a (proper) functor. Let $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$, $c \in \mathbb{M}^V$, $f: a \dashrightarrow b$, $g: b \dashrightarrow c$ and $Y' \subseteq [Y]^a$. Then, by Proposition 3.5.6

$$\#(g \circ f) = (g \circ f)^a_\# = g^{f(a)}_\# \circ f^a_\# = g^b_\# \circ f^a_\# = \#(g) \circ \#(f).$$

The rest follows from Lemma 4.3.2. □

Remark 4.3.7. *For the primal view we can easily adjust the category $\mathbb{A}$ in the sense that an arrow $f : a \dashrightarrow b$ between objects $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ is a monotone (wrt. inclusion) function $f: \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_b)$.*

This allows us to define $\#: \mathbb{C} \to \mathbb{A}$ as follows: for an object $a \in \mathbb{M}^Y$, we let $\#(a) = a$ and, given an arrow $f : a \dashrightarrow b$, we let $\#(f) = f^\#_a$. Here, $\#$ is not a proper functor.

When working with the finite subcategories, $\#$ is again a proper functor.

4.4. Predicate Liftings as Functors

In this section we discuss how predicate liftings can be integrated into our theory (cf. Section 2.5.2). As a reminder, given a functor $F: \mathbf{Set} \to \mathbf{Set}$, a **predicate lifting** is a family of functions $\tilde{F}_Y: \mathbb{M}^Y \to \mathbb{M}^{FY}$ (where Y is a set), such that for $g: Z \to Y$, $a: Y \to \mathbb{M}$

[1] A *lluf sub-category* a sub-category that contains all objects.

it holds that $(Fg)^*(\tilde{F}_Y(a)) = \tilde{F}_Z(g^*(a))$. Thus predicate liftings must commute with reindexings which proves very useful. Predicate liftings are in one-to-one correspondence to evaluation maps $ev\colon F\mathbb{M} \to \mathbb{M}$ and given ev, we define the corresponding lifting to be $\tilde{F}(a) = ev \circ Fa\colon FY \to \mathbb{M}$, where $a\colon Y \to \mathbb{M}$.

We quickly revisit the two main lifitings from Section 2.5.

Example 4.4.1. *We consider the (finite) distribution functor $\mathcal{D}_f$ that maps a set X to all maps $p\colon X \to [0,1]$ that have finite support and satisfy $\sum_{x\in X} p(x) = 1$. (Here $\mathbb{M} = [0,1]$.) One evaluation map is $ev\colon \mathcal{D}_f[0,1] \to [0,1]$ with $ev(p) = \sum_{r\in[0,1]} r \cdot p(r)$, where p is a distribution on $[0,1]$ (expectation). $\tilde{\mathcal{D}}_f$ is well-behaved and non-expansive.*

Example 4.4.2. *Another example is given by the finite powerset functor $\mathcal{P}_f$. We are given the evaluation map $ev\colon \mathcal{P}_f\mathbb{M} \to \mathbb{M}$, defined for $S \subseteq \mathbb{M}$ as $ev(S) = \max S$, where $\max\varnothing = 0$. The lifting $\tilde{\mathcal{P}}_f$ is well-behaved and non-expansive.*

Non-expansive predicate liftings can be seen as functors $\tilde{F}\colon \mathbb{C}^* \to \mathbb{C}^*$. To be more precise, $\tilde{F}$ maps an object $a \in \mathbb{M}^Y$ to $\tilde{F}(a) \in \mathbb{M}^{FY}$ and an arrow $g^* : a \dashrightarrow a \circ g$, , where $g\colon Z \to Y$, to $(Fg)^* : \tilde{F}a \dashrightarrow \tilde{F}(a \circ g)$.

Proposition 4.4.3. *Let $\tilde{F}$ be a (non-expansive) predicate lifting. There is a natural transformation[2] $\beta\colon \#E \Rightarrow \#E\tilde{F}$ between (lax) functors $\#E, \#E\tilde{F}\colon \mathbb{C}^* \to \mathbb{A}$, whose components, for $a \in \mathbb{M}^Y$, are $\beta_a\colon a \dashrightarrow \tilde{F}(a)$ in $\mathbb{A}$, defined by $\beta_a(U) = \tilde{F}^a_\#(U)$ for $U \subseteq [Y]^a$.*
That is, the following diagrams commute for every $g\colon Z \to Y$ (on the left the diagram with formal arrows, omitting the embedding functor E, and on the right the functions with corresponding domains). Note that $\#(g) = g^{-1}$.

$$\begin{array}{ccc} \#(a) & \xrightarrow{\#(g^*)} & \#(a \circ g) \\ \beta_a \downarrow & & \downarrow \beta_{a\circ g} \\ \#(\tilde{F}a) & \xrightarrow{\#(\tilde{F}(g^*))} & \#(\tilde{F}(a \circ g)) \end{array} \qquad \begin{array}{ccc} \mathcal{P}([Y]^a) & \xrightarrow{g^{-1}} & \mathcal{P}([Z]^{a\circ g}) \\ \tilde{F}^a_\# \downarrow & & \downarrow \tilde{F}^{a\circ g}_\# \\ \mathcal{P}([FY]^{\tilde{F}(a)}) & \xrightarrow{(Fg)^{-1}} & \mathcal{P}([FZ]^{\tilde{F}(a\circ g)}) \end{array}$$

Proof. We first define a natural transformation $\eta\colon E \Rightarrow E\tilde{F}$ with components $\eta_a\colon Ea = a \dashrightarrow E\tilde{F}(a) = \tilde{F}(a)$ (for $a \in \mathbb{M}^Y$) by defining $\eta_a(b) = \tilde{F}(b)$ for $b \in \mathbb{M}^Y$. The η_a are non-expansive by assumption. In addition, η is natural due to the definition of a predicate lifting, i.e., $(Fg)^* \circ \tilde{F} = \tilde{F} \circ g^*$ for $g\colon Z \to Y$.

Now we apply # and use the fact that # is functorial even for the full categories $\mathbb{C}$, $\mathbb{A}$ whenever one of the two arrows to which # is applied is a reindexing (see Lemma 4.3.4). Furthermore we observe that $\beta = \#(\eta)$. This immediately gives us the diagram on the left and the diagram on the right just displays the underlying functions. □

[2] cf. Definition 2.4.6

4.5. A Compositional Representation of the Wasserstein Lifting and Behavioural Metrics

In this section we show how the framework for fixpoint checking from Chapter 3 can be used to deal with coalgebraic behavioural metrics.

Given a functor $F\colon \mathbf{Set} \to \mathbf{Set}$ and a coalgebra $\xi\colon X \to FX$ where X is a set. The idea is to endow X with a pseudo-metric $d_\xi\colon X \times X \to \mathbb{M}$ defined as the least fixpoint of the map $d \mapsto d^F \circ (\xi \times \xi)$ where $\_^F$ lifts a metric $d\colon X \times X \to \mathbb{M}$ to a metric $d^F\colon FX \times FX \to \mathbb{M}$ (cf. Section 2.5). Here we again focus on the Wasserstein lifting and show how approximations of the functions involved in the definition of the pseudometric can be determined.

The Wasserstein lifting[3] $\_^F\colon \mathbb{M}^{X\times X} \to \mathbb{M}^{FX\times FX}$ was defined for $d\colon X \times X \to \mathbb{M}$ and $t_1, t_2 \in FX$ as

$$d^F(t_1,t_2) = \inf_{t\in\Gamma(t_1,t_2)} \tilde{F}d(t).$$

We will be working with finitely coupled liftings $\tilde{F}$ (cf. Definition 2.5.8), i.e. we have

$$d^F(t_1,t_2) = \min_{t\in\Gamma(t_1,t_2)} \tilde{F}d(t).$$

For a coalgebra $\xi\colon X \to FX$ the behavioural pseudometric $d : X \times X \to \mathbb{M}$ arises as the least fixpoint of $\mathcal{W} = (\xi \times \xi)^* \circ (\_^F)$ where $(\_^F)$ is the Wasserstein lifting. Note that we do not use a discount factor to ensure contractivity and hence the fixpoint might not be unique. Thus, given some fixpoint d, the d-approximation $\mathcal{W}_\#^d$ can be used for checking whether $d = \mu\mathcal{W}$.

In the rest of the section we show how $\mathcal{W}$ can be decomposed into basic components and study the corresponding approximation.

The Wasserstein lifting can be decomposed as $\_^F = \min_u \circ \tilde{F}$ where $\tilde{F} : \mathbb{M}^{X\times X} \to \mathbb{M}^{F(X\times X)}$ is the predicate lifting – which we require to be non-expansive (cf. Lemma 2.5.4) – and $\min_u$ is the minimum over the coupling function $u\colon F(X\times X) \to FX\times FX$ defined as $u(t) = (F\pi_1(t), F\pi_2(t))$, which means that $\min_u\colon \mathbb{M}^{F(X\times X)} \to \mathbb{M}^{FX\times FX}$ (see Table 3.1).

We can now derive the corresponding d-approximation.

Proposition 4.5.1. *Assume that $\tilde{F}$ is finitely coupled. Let $Y = X \times X$, where X is finite. For $d \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^d$ we have*

$$\begin{aligned}\mathcal{W}_\#^d(Y') = \{(x,y) \in [Y]^d \mid \exists t \in \tilde{F}_\#^d(Y'), u(t) = (\xi(x),\xi(y)),\\ \tilde{F}d(t) = \min\nolimits_{t'\in\Gamma(\xi(x),\xi(y))} \tilde{F}d(t')\}.\end{aligned}$$

Proof. We first remark that since X is finite and $\tilde{F}$ is finitely coupled it is sufficient to restrict to finite subsets of $F(X \times X)$ and $FX \times FX$ (cf. Remark 4.5.7). In other words

[3] We write d^F instead of $d^{\downarrow F}$ here.

$\mathcal{W}$ can be obtained as composition of functions living in $\mathbb{C}_f$, hence $\#$ is a proper functor and approximations can obtained compositionally. We exploit this fact in the following.

For $d \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^d$ we have, using the results of Proposition 3.5.5,

$$\begin{aligned}\mathcal{W}^d_\#(Y') &= \{(x,y) \in [Y]^d \mid (\xi(x),\xi(y)) \in (\min_u)^{\tilde{F}(d)}_\#(\tilde{F}^d_\#(Y'))\} \\ &= \{(x,y) \in [Y]^d \mid Min_{\tilde{F}(d)|u^{-1}(\xi(x),\xi(y))} \cap \tilde{F}^d_\#(Y') \neq \emptyset\} \\ &= \{(x,y) \in [Y]^d \mid \exists t \in \tilde{F}^d_\#(Y'), u(t) = (\xi(x),\xi(y)), \\ &\qquad \tilde{F}d(t) = \min_{t' \in \Gamma(\xi(x),\xi(y))} \tilde{F}d(t')\}\end{aligned}$$

□

Intuitively the statement of Proposition 4.5.1 means that the minimum must be reached in a coupling based on Y'.

For using the above result we next characterize $\tilde{F}^d_\#$. We rely on the fact that d can be decomposed into $d = \pi_1 \circ \bar{d}$, where the projection π_1 is independent of d, and exploit the natural transformation in Proposition 4.4.3.

Proposition 4.5.2. *Let $\pi_1 : \mathbb{M} \times \{0,1\} \to \mathbb{M}$ be the projection to the first component and $\bar{d} : Y \to \mathbb{M} \times \{0,1\}$ with $\bar{d}(y) = (d(y), \chi_{Y'}(y))$ where $\chi_{Y'} : Y \to \{0,1\}$ is the characteristic function of Y'* [4]*. Then $\tilde{F}^d_\#(Y') = (F\bar{d})^{-1}(\tilde{F}^{\pi_1}_\#((\mathbb{M}\backslash\{0\}) \times \{1\}))$.*

Proof. Let $d \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^d$. Note that $\bar{d}^{-1}((\mathbb{M}\backslash\{0\}) \times \{1\}) = Y'$ and $d = \pi_1 \circ \bar{d}$, thus by Proposition 4.4.3:

$$\begin{aligned}\tilde{F}^d_\#(Y') &= \tilde{F}^{\pi_1 \circ \bar{d}}_\#(\bar{d}^{-1}((\mathbb{M}\backslash\{0\}) \times \{1\})) \\ &= (F\bar{d})^{-1}(\tilde{F}^{\pi_1}_\#((\mathbb{M}\backslash\{0\}) \times \{1\}))\end{aligned}$$

□

Here $\tilde{F}^{\pi_1}_\#((\mathbb{M}\backslash\{0\}) \times \{1\}) \subseteq F(\mathbb{M} \times \{0,1\})$ is independent of d and has to be determined only once for every predicate lifting $\tilde{F}$. We will show how this set looks like for our example functors.

Lemma 4.5.3. *Consider the lifting of the distribution functor presented in Example 4.4.1 and let $Z = [0,1] \times \{0,1\}$. Then we have*

$$(\tilde{\mathcal{D}}_f)^{\pi_1}_\#((0,1] \times \{1\}) = \{p \in \mathcal{D}_f Z \mid supp(p) \in (0,1] \times \{1\}\}.$$

Proof. See Appendix: Lemma A.3.2. □

[4] See Section 2.1 on the definition of $\chi_{Y'}$

This means intuitively that a decrease or "slack" can exactly be propagated for elements whose probabilities are strictly larger than 0.

Example 4.5.4. *Let* $X = \{x, y, z\}$ *and the coalgebra* $\xi\colon X \to \mathcal{D}_f X$ *be given by* $\xi(x)(x) = 1$, $\xi(y)(y) = \xi(y)(z) = 1/2$ *and* $\xi(z)(z) = 1$. *The only valid (and thus optimal) coupling* $t' \in \mathcal{D}_f(X \times X)$ *of* $\xi(x), \xi(y)$ *is given by* $t(x,y) = t(x,z) = 1/2$. *Given the pseudometric* $d\colon X \times X \to [0,1]$ *with* $d(x,y) = d(x,z) = 1$ *and* $d(y,z) = 0$, *which is a fixpoint of* $\mathcal{W}$ *and* $Y' = \{(x,y),(x,z)\} \subseteq X \times X$, *it holds that* $(x,y) \in \mathcal{W}^d_\#(Y')$ *as* t' *attains the minimum:*

$$\mathcal{W}(d)(x,y) = d^{\mathcal{D}_f}(\xi(x),\xi(y)) = \min_{t \in \Gamma(\xi(x),\xi(y))} \tilde{\mathcal{D}}_f d(t) = \tilde{\mathcal{D}}_f d(t') = 1/2 \cdot 1 + 1/2 \cdot 1 = 1.$$

and $t' \in (\tilde{\mathcal{D}}_f)^d_\#(Y')$: *We have* $\bar{d}(x,y) = \bar{d}(x,z) = (1,1)$ *and*

$$\mathcal{D}_f \bar{d}(t') = p \in [\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f \pi_1} \text{ with } p(1,1) = 1/2 + 1/2 = 1.$$

Immediately, $p \in (\tilde{\mathcal{D}}_f)^{\pi_1}_\#((0,1] \times \{1\})$ *and* $(\mathcal{D}_f \bar{d})^{-1}(p) = t'$ *Thus* $t' \in (\tilde{\mathcal{D}}_f)^d_\#(Y')$ *by Lemma 4.5.3.*

We now turn to the (finite) powerset functor.

Lemma 4.5.5. *Consider the lifting of the powerset functor from Example 4.4.2 and let* $Z = \mathbb{M} \times \{0,1\}$. *Then we have*

$$(\tilde{\mathcal{P}}_f)^{\pi_1}_\#((\mathbb{M}\backslash\{0\}) \times \{1\}) = \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f \pi_1} \mid \exists (s,1) \in S,\ \forall (s',0) \in S:\ s \sqsupset s'\}.$$

Proof. See Appendix: Lemma A.3.3. □

The idea is that if we decrease the value of an element, then there should be no other element with a value larger or equal which is not decreased.

Example 4.5.6. *Let* $X = \{x,y\}$ *and the coalgebra* $\xi\colon X \to \mathcal{P}_f X$ *be given by* $\xi(x) = \{x,y\}$, $\xi(y) = \{x\}$. *The only valid (and thus optimal) coupling* $t' \in \mathcal{P}_f(X \times X)$ *of* $\xi(x), \xi(y)$ *is given by* $t' = \{(x,x),(y,x)\}$, *i.e.* $u(t') = (\xi(x),\xi(y))$. *Given* $d\colon X \times X \to [0,1]$ *with* $d(x,x) = d(y,y) = 0$ *and* $d(x,y) = d(y,x) = 1$, *which is a fixpoint of* $\mathcal{W}$ *and* $Y' = \{(x,y),(y,x)\} \subseteq X \times X$. *It holds that* $(x,y) \in \mathcal{W}^d_\#(Y')$ *as* t' *attains the minimum:*

$$\mathcal{W}(d)(x,y) = d^{\mathcal{P}_f}(\xi(x),\xi(y)) = \min_{t \in \Gamma(\xi(x),\xi(y))} \tilde{\mathcal{P}}_f d(t) = \tilde{\mathcal{P}}_f d(t') = \max\{0,1\} = 1$$

and $t' \in (\tilde{\mathcal{P}}_f)^d_\#(Y')$: *We have* $\bar{d}(x,x) = \bar{d}(y,y) = (0,0)$, $\bar{d}(x,y) = \bar{d}(y,x) = (1,1)$ *and*

$$\mathcal{P}_f \bar{d}(t') = S = \{(0,0),(1,1)\} \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f \pi_1}$$

Immediately, $S \in (\tilde{\mathcal{P}}_f)^{\pi_1}_\#((\mathbb{M}\backslash\{0\}) \times \{1\})$ *and* $(\mathcal{P}_f \bar{d})^{-1}(S) = t'$. *Thus* $t' \in (\tilde{\mathcal{P}}_f)^d_\#(Y')$ *by Lemma 4.5.5.*

Remark 4.5.7. *Note that # is a functor on the subcategory $\mathbb{C}_f$, while some liftings (e.g., the one for the distribution functor) work with infinite sets. In this case, given a finite set Y, we actually focus on a finite $D \subseteq FY$. (This is possible since we consider coalgebras with finite state space and assume that all liftings are finitely coupled.) Then we consider $\tilde{F}_Y\colon \mathbb{M}^Y \to \mathbb{M}^{FY}$ and $e\colon D \hookrightarrow FY$ (the embedding of D into FY). We set $f = e^* \circ \tilde{F}_Y$. Given $a\colon Y \to \mathbb{M}$, we view f as an arrow $a \dashrightarrow \tilde{F}(a) \circ e$ in $\mathbb{C}$. The approximation in this subsection adapts to the "reduced" lifting, which can be seen as follows (cf. Lemma 4.3.4, which shows that # preserves composition if one of the arrows is a reindexing):*

$$f^a_\# = \#(f) = \#(e^* \circ \tilde{F}_Y) = \#(e^*) \circ \#(\tilde{F}_Y) = e^{-1} \circ \#(\tilde{F}_Y) = \#(\tilde{F}_Y) \cap D.$$

4.6. GS-Monoidality

We will now show that the categories $\mathbb{C}_f$ and $\mathbb{A}_f$ can be turned into gs-monoidal categories. This will give us a way to assemble functions and their approximations compositionally and this method will form the basis for the tool.

A strict gs-monoidal category is a strict symmetric monoidal category $\mathbb{C}$ – where $\otimes$ denotes the tensor product and e the unit – such that for every object a there exist morphisms $\nabla_a\colon a \dashrightarrow a \times a$ (duplicator) and $!_a\colon a \dashrightarrow e$ (discharger) satisfying certain axioms. We denote the symmetry by $\rho_{a,b}\colon a \otimes b \dashrightarrow b \otimes a$ for objects a, b. See Section 2.4.1 for detailed definitions.

In fact, in order to obtain strict gs-monoidal categories with disjoint union, we will work with the skeleton categories where every finite set Y is represented by an isomorphic copy $\{1, \dots, |Y|\}$. This enables us to make disjoint union strict, i.e., associativity holds on the nose and not just up to isomorphism. In particular for finite sets Y, Z, we define disjoint union as $Y + Z = \{1, \dots, |Y|, |Y| + 1, \dots, |Y| + |Z|\}$.

Theorem 4.6.1. *The category $\mathbb{C}_f$ with the following operators is gs-monoidal:*

1. *The tensor $\otimes$ on objects $a \in \mathbb{M}^Y$ and $b \in \mathbb{M}^Z$ is defined as*

   $$a \otimes b = a + b \in \mathbb{M}^{Y+Z}$$

   where for $k \in Y + Z$ we have $(a + b)(k) = a(k)$ if $k \le |Y|$ and $(a + b)(k) = b(k - |Y|)$ if $|Y| < k \le |Y| + |Z|$.

   On arrows $f\colon a \dashrightarrow b$ and $g\colon a' \dashrightarrow b'$ (with $a' \in \mathbb{M}^{Y'}$, $b' \in \mathbb{M}^{Z'}$) tensor is given by

   $$f \otimes g\colon \mathbb{M}^{Y+Y'} \to \mathbb{M}^{Z+Z'}, \quad (f \otimes g)(u) = f(\overleftarrow{u}_Y) + g(\overrightarrow{u}_Y)$$

   for $u \in \mathbb{M}^{Y+Y'}$ where $\overleftarrow{u}_Y \in \mathbb{M}^Y$ and $\overrightarrow{u}_Y \in \mathbb{M}^{Y'}$, defined as $\overleftarrow{u}_Y(k) = u(k)$ $(1 \le k \le |Y|)$ and $\overrightarrow{u}_Y(k) = u(|Y| + k)$ $(1 \le k \le |Y'|)$.

2. *The symmetry* $\rho_{a,b}\colon a \otimes b \dashrightarrow b \otimes a$ *for* $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ *is defined for* $u \in \mathbb{M}^{Y+Z}$ *as*
$$\rho_{a,b}(u) = \vec{u}_Y + \overleftarrow{u}_Y.$$
3. *The unit* e *is the unique mapping* $e\colon \varnothing \to \mathbb{M}$.
4. *The duplicator* $\nabla_a\colon a \dashrightarrow a \otimes a$ *for* $a \in \mathbb{M}^Y$ *is defined for* $u \in \mathbb{M}^Y$ *as*
$$\nabla_a(u) = u + u.$$
5. *The discharger* $!_a\colon a \dashrightarrow e$ *for* $a \in \mathbb{M}^Y$ *is defined for* $u \in \mathbb{M}^Y$ *as* $!_a(u) = e$.

Proof. See Appendix: Theorem A.3.4. □

We now turn to the abstract category $\mathbb{A}_f$. Note that here functions have as parameters sets of the form $U \subseteq [Y]^a \subseteq Y$. Hence, (the cardinality of) Y can not be determined directly from U and we need extra care with the tensor.

Theorem 4.6.2. *The category* $\mathbb{A}_f$ *with the following operators is gs-monoidal:*

1. *The tensor* $\otimes$ *on objects* $a \in \mathbb{M}^Y$ *and* $b \in \mathbb{M}^Z$ *is again defined as* $a \otimes b = a + b$.
   On arrows $f\colon a \dashrightarrow b$ *and* $g\colon a' \dashrightarrow b'$ *(where* $a' \in \mathbb{M}^{Y'}$, $b' \in \mathbb{M}^{Z'}$ *and* $f\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{b'})$, $g\colon \mathcal{P}([Y']^{a'}) \to \mathcal{P}([Z']^{b'})$ *are the underlying functions), the tensor is given by*
$$f \otimes g\colon \mathcal{P}([Y+Y']^{a+a'}) \to \mathcal{P}([Z+Z']^{b+b'}), \quad (f \otimes g)(U) = f(\overleftarrow{U}_Y) \cup_Z g(\vec{U}_Y)$$
   where $\overleftarrow{U}_Y = U \cap \{1, \dots, |Y|\}$ *and* $\vec{U}_Y = \{k \mid |Y| + k \in U\}$. *Furthermore:*
$$U \cup_Y V = U \cup \{|Y| + k \mid k \in V\} \quad (\textit{where } U \subseteq Y)$$
2. *The symmetry* $\rho_{a,b}\colon a \otimes b \dashrightarrow b \otimes a$ *for* $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ *is defined for* $U \subseteq [Y+Z]^{a+b}$ *as*
$$\rho_{a,b}(U) = \vec{U}_Y \cup_Z \overleftarrow{U}_Y \subseteq [Z+Y]^{b+a}$$
3. *The unit* e *is again the unique mapping* $e\colon \varnothing \to \mathbb{M}$.
4. *The duplicator* $\nabla_a\colon a \dashrightarrow a \otimes a$ *for* $a \in \mathbb{M}^Y$ *is defined for* $U \subseteq [Y]^a$ *as*
$$\nabla_a(U) = U \cup_Y U \subseteq [Y+Y]^{a+a}.$$
5. *The discharger* $!_a\colon a \dashrightarrow e$ *for* $a \in \mathbb{M}^Y$ *is defined for* $U \subseteq [Y]^a$ *as* $!_a(U) = \varnothing$.

Proof. See Appendix: Theorem A.3.5. □

Finally, the approximation # is indeed gs-monoidal, i.e., it preserves all the additional structure (tensor, symmetry, unit, duplicator and discharger).

Theorem 4.6.3. *$\#: \mathbb{C}_f \to \mathbb{A}_f$ is a gs-monoidal functor.*

Proof. See Appendix: Theorem A.3.6. □

4.7. UDEfix: A Tool for Fixpoints Checks

We exploit gs-monoidality as discussed before and present a tool, called UDEfix, where the user can compose his or her very own function $f: \mathbb{M}^Y \to \mathbb{M}^Y$ as a sort of circuit. Exploiting the fact that the functor # is gs-monoidal, this circuit is then transformed automatically and in a compositional way into the corresponding abstraction $f_\#^a$, for some given $a \in \mathbb{M}^Y$. By computing the greatest fixpoint of $f_\#^a$ and checking for emptiness, UDEfix can check whether $a = \mu f$.

In fact, UDEfix can handle all basic functions presented in Section 3.5, in particular the functions listed in Table 3.1. In addition to fixpoint checks, it is possible to perform (non-complete) checks whether a given post-fixpoint a is below the least fixpoint μf (cf. Lemma 3.4.2), i.e. if $\nu f_*^a = \varnothing$ then $a \sqsubseteq \mu f$ but whenever $\nu f_*^a \neq \varnothing$ we can not conclude anything. The dual checks (for greatest fixpoint and pre-fixpoints) are implemented as well. Additionally, given some fixpoint $a \in \mathbb{M}^Y$ which is not the least, respectively greatest fixpoint, UDEfix provides a constant ι such that $a \ominus \iota_{\nu f_\#^a} \sqsupseteq \mu f$, respectively $a \oplus \iota_{\nu f_a^\#} \sqsubseteq \nu f$.

Building the desired function $f: \mathbb{M}^Y \to \mathbb{M}^Y$ requires three steps:

- Choosing the MV-algebra $\mathbb{M}$ in File → Settings. Currently the MV-chains $[0, k]$ (algebra 1) and $\{0, \dots, k\}$ (algebra 2) for arbitrary k are supported (cf. Example 2.3.17).
- Creating the required basic functions by specifying their parameters.
- Assembling f from these basic functions.

UDEfix is a Windows-Tool created in Python, which can be obtained from `https://github.com/TimoMatt/UDEfix`. The GUI of UDEfix is separated into three areas: Content area (left), Building area (middle) and Basic-Functions area (right), see Figure 4.1. Under File the user can save/load contents and set the MV-algebra in Settings (under File). Functions built in the Building area can be saved and loaded.

Basic-Functions Area: The Basic-Functions area contains the basic functions, encompassing those listed in Table 3.1, the function sub'_w from Section 3.6.9 and two additional ones. Via drag-and-drop (or right-click) these basic functions can be added to the Building area to create a Function box. Each such box requires three (in the case of av_D two)

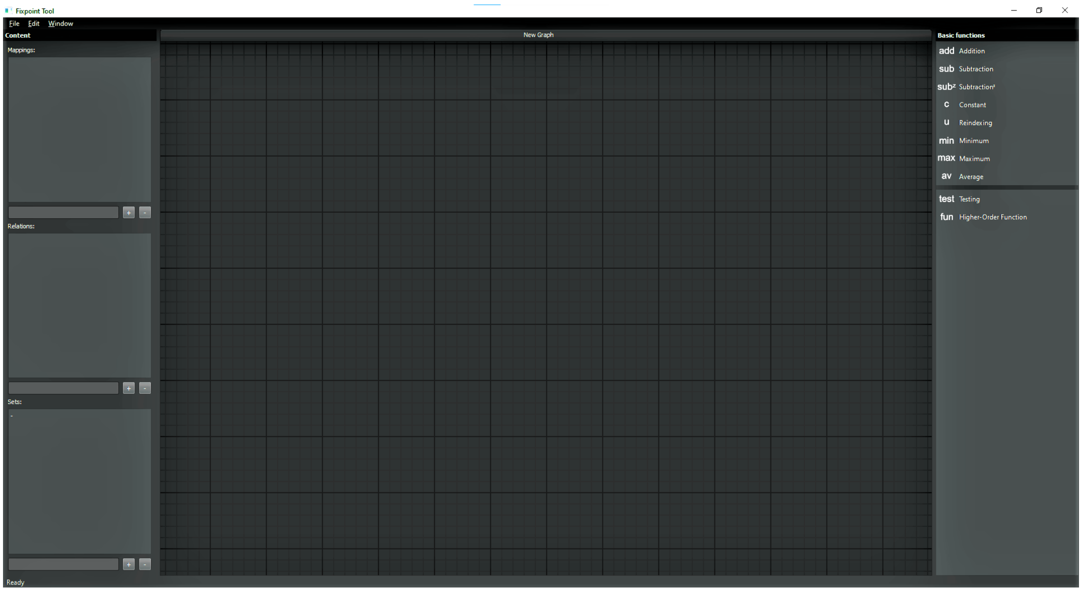


Fig. 4.1.: Appearance of the tool, also see Figure 4.2 for Zoom of the left and right column

Basic Function	c_k	u^*	$\min_{\mathcal{R}} / \max_{\mathcal{R}}$	add_w / sub_w	sub'_w
Req. Parameter	$k \in \mathbb{M}^Z$	$u\colon Z \to Y$	$R \subseteq Y \times Z$	$w \in \mathbb{M}^Y$	$w \in \mathbb{Z}^Y$

Table 4.1.: Additional parameters for the basic functions from Table 3.1

Contents: The Input set, the Output set and an additional parameters, see Table 4.1. These Contents are to be created in the Content area.

Additionally the Basic-Functions area contains the auxiliary function Higher-Order Function which can be used to compose two basic functions (which proves useful for complicated functions) and Testing which we will discuss in the next paragraph.

Building Area: The user can connect the created Function boxes to obtain the function of interest. Composing functions is as simple as connecting two Function boxes in the correct order and disjoint union is achieved by connecting two boxes to the same box. We note that Input and Output sets of connected Function boxes need to match, otherwise the function is not built correctly. In Figure 4.3 we show how the function Δ which computes the behavioural distance for a labeled Markov chain can be assembled (also cf. Figure 3.4). Here, the parameters are instantiated for the labeled Markov chain displayed in Figure 4.4.

The special box Testing is always required at the end. Here, the user can enter some mapping $a\colon Y \to \mathbb{M}$, test if a is a fixpoint/pre-fixpoint/post-fixpoint of the built function f and afterwards compute the greatest fixpoint of the approximation ($\nu f^a_\#$ if we want to check whether $\mu f = a$). If the result is not the empty set ($\nu f^\#_a \neq \emptyset$) one can compute

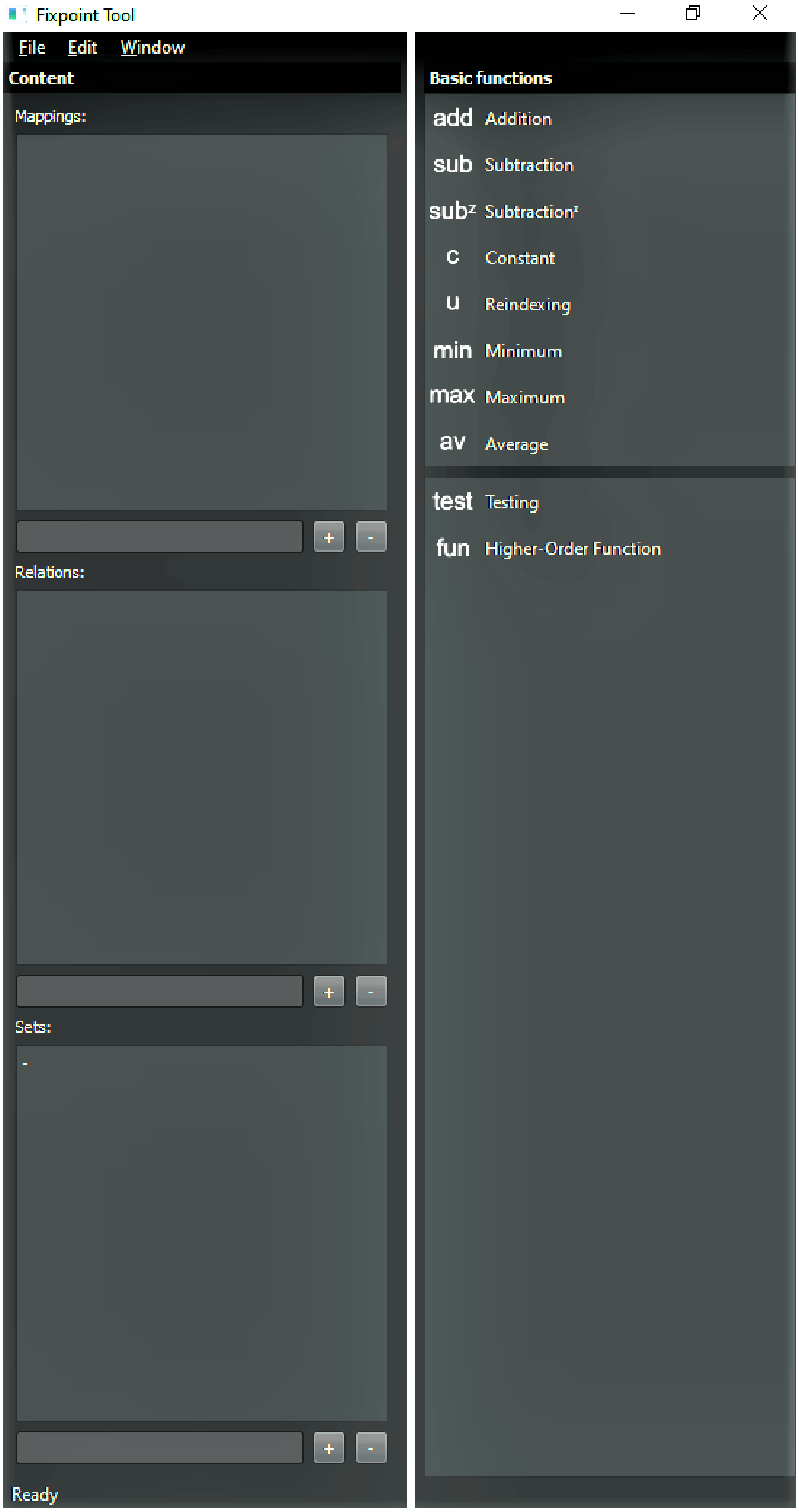


Fig. 4.2.: Zoom of Figure 4.1

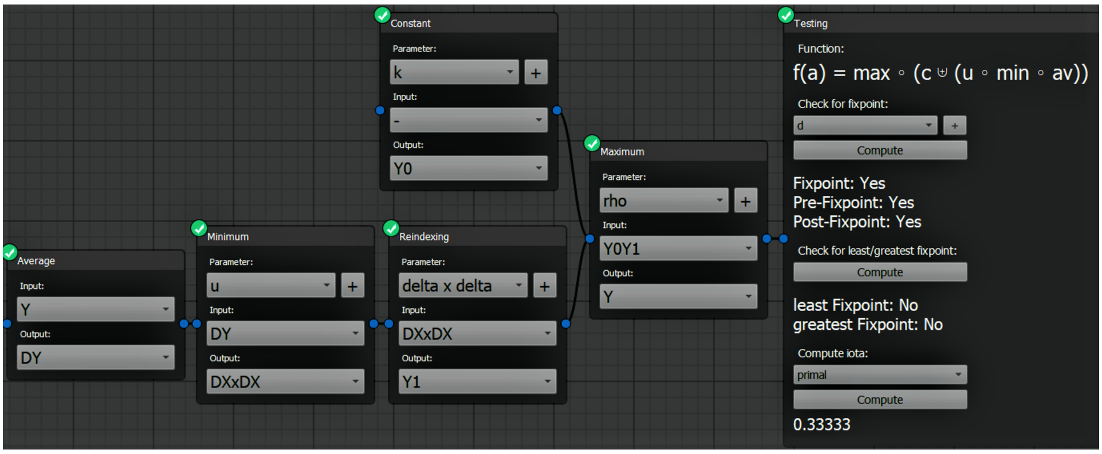


Fig. 4.3.: Assembling the function Δ from Section 2.6.3.

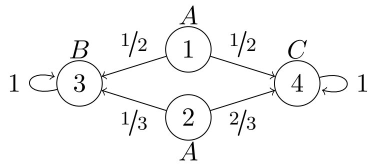


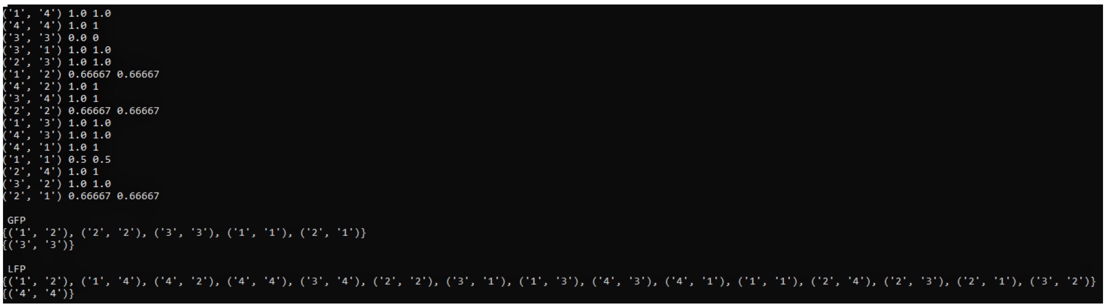
Fig. 4.4.: Example of a labeled Markov chain, labels are denoted by big letters.

a suitable value for decreasing a, needed for iterating to the least fixpoint from above (respectively increasing a for iterating to the greatest fixpoint from below). There is additional support for comparison with pre- and post-fixpoints.

Example 4.7.1. *In the system in Figure 4.4, the function* $d: Y \to [0,1]$ *with* $d(3,3) = 0$, $d(1,1) = 1/2$, $d(1,2) = d(2,1) = d(2,2) = 2/3$ *and* 1 *for all other pairs (*d *is not a pseudometric) is a fixpoint of* Δ *from Section 2.6.3 whose least fixpoint corresponds to the behavioural distances in a labeled Markov chain. By clicking* Compute *in the* Testing*-box,* UDEfix *displays that* d *is a fixpoint and tells us that* d *is in fact not the least and not the greatest fixpoint. It also computes the greatest fixpoints of the approximations step by step (via Kleene iteration) and displays the results to the user, see Figure 4.5 where* $\nu f_a^\# = \{(3,3)\}$ *and* $\nu f_\#^a = \{(4,4)\}$.

Content Area: Here the user can create sets, mappings and relations which are used to specify the basic functions. Creating a set is done by entering a name for the new set and clicking on the plus ("+"). The user can create a variety of different types of sets, for example the basic set $X = \{1,2,3,4\}$ or the set $D = \{p_1, p_2, p_3, p_4\}$ which is a set of mappings resp. probability distributions.

Once, Input and Output sets are created we can define the required parameters (cf. Table 4.1). Here, the created sets can be chosen as domain and co-domain. Relations can

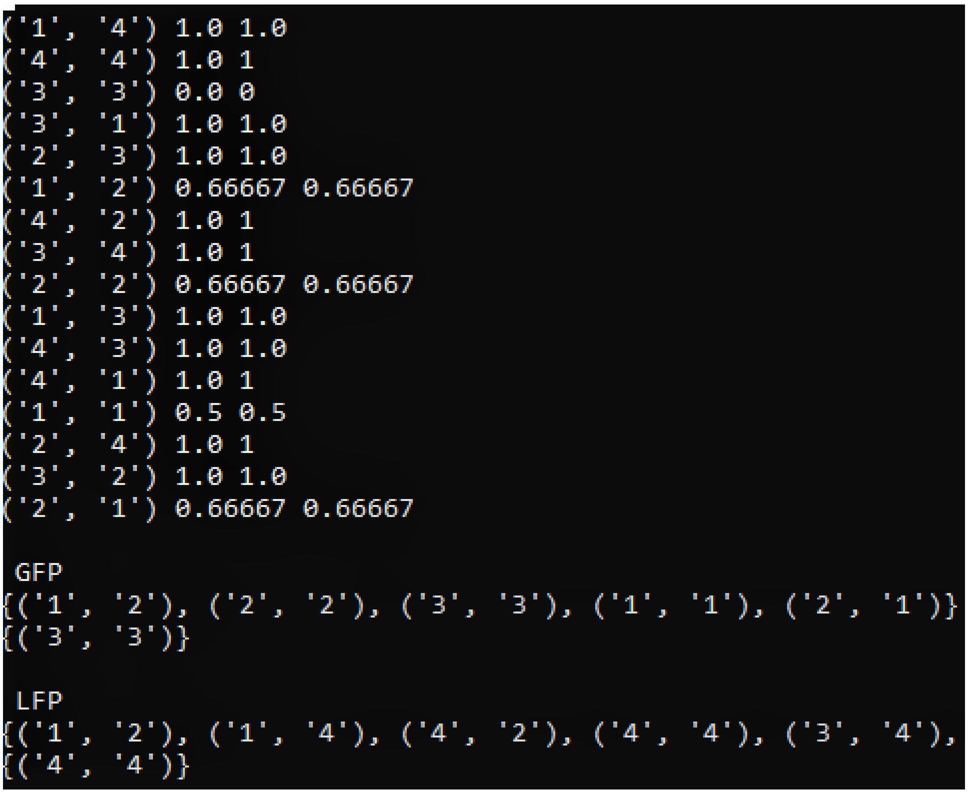


```
('1', '4') 1.0 1.0
('4', '4') 1.0 1
('3', '3') 0.0 0
('3', '1') 1.0 1.0
('2', '3') 1.0 1.0
('1', '2') 0.66667 0.66667
('4', '2') 1.0 1
('3', '4') 1.0 1
('2', '2') 0.66667 0.66667
('1', '3') 1.0 1.0
('4', '3') 1.0 1.0
('4', '1') 1.0 1
('1', '1') 0.5 0.5
('2', '4') 1.0 1
('3', '2') 1.0 1.0
('2', '1') 0.66667 0.66667

 GFP
{('1', '2'), ('2', '2'), ('3', '3'), ('1', '1'), ('2', '1')}
{('3', '3')}

 LFP
{('1', '2'), ('1', '4'), ('4', '2'), ('4', '4'), ('3', '4'), ('2', '2'), ('3', '1'), ('1', '3'), ('4', '3'), ('4', '1'), ('1', '1'), ('2', '4'), ('2', '3'), ('2', '1'), ('3', '2')}
{('4', '4')}
```

Fig. 4.5.: Displaying the computation of the greatest fixpoint of the approximations, see Figure 4.6 for Zoom



```
('1', '4') 1.0 1.0
('4', '4') 1.0 1
('3', '3') 0.0 0
('3', '1') 1.0 1.0
('2', '3') 1.0 1.0
('1', '2') 0.66667 0.66667
('4', '2') 1.0 1
('3', '4') 1.0 1
('2', '2') 0.66667 0.66667
('1', '3') 1.0 1.0
('4', '3') 1.0 1.0
('4', '1') 1.0 1
('1', '1') 0.5 0.5
('2', '4') 1.0 1
('3', '2') 1.0 1.0
('2', '1') 0.66667 0.66667

 GFP
{('1', '2'), ('2', '2'), ('3', '3'), ('1', '1'), ('2', '1')}
{('3', '3')}

 LFP
{('1', '2'), ('1', '4'), ('4', '2'), ('4', '4'), ('3', '4'),
{('4', '4')}
```

Fig. 4.6.: Zoom of Figure 4.5

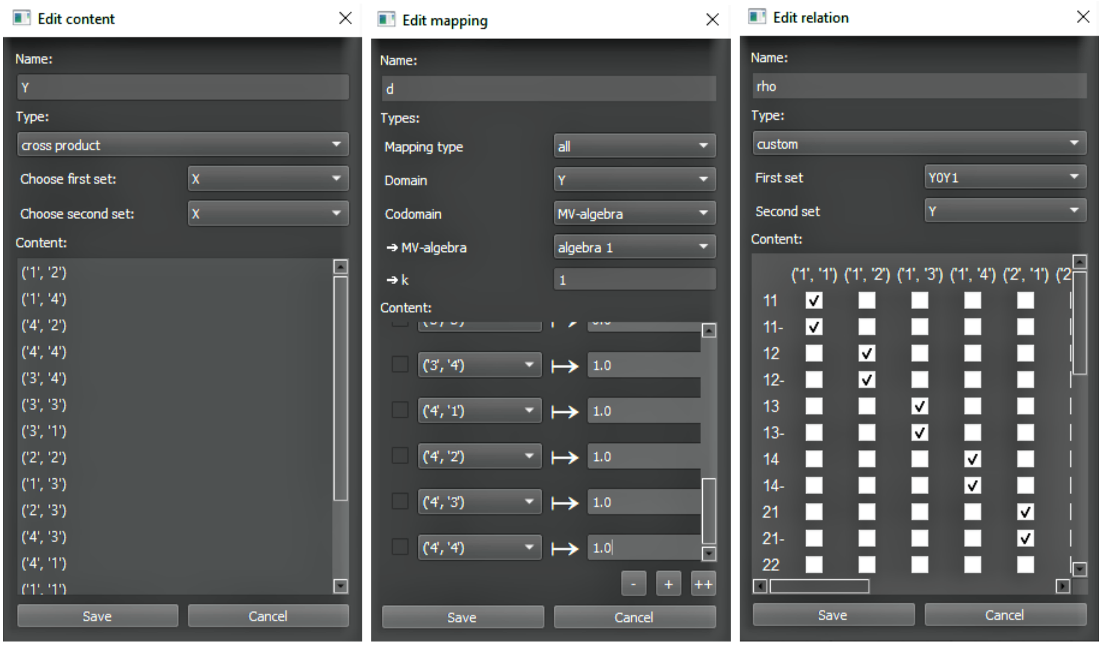


Fig. 4.7.: Contents: Set Y, Mapping d, Relation ρ.

be handled in a similar fashion: Given the two sets one wants to relate, creating a relation can be easily achieved by checking some boxes. Additionally the user has access to some useful in-built relations: "is-element-of"-relation and projections to the i-th component.

To ease the use, by clicking on the "+" in a Function box a new matching content with chosen Input and Output sets is created. The additional parameters (cf. Table 4.1) have domains and co-domains which need to be created or are the chosen MV-algebra. The Testing function d is a mapping as well.

See Figure 4.7 for examples on how to create the contents Y (set), d (distance function) and ρ (relation).

An In-Depth Tutorial on Inserting an Example. To clarify the use of the tool and to give some useful hints on how to optimize the use, we will insert the function $\mathcal{T}\colon [0,1]^S \to [0,1]^S$ whose least fixpoint coincides with the termination probability of a Markov chain (cf. Section 2.6.1). We consider the Markov chain in Figure 4.8. The least fixpoint of $\mathcal{T}$ is given in green and the greatest fixpoint in red.

To start, we choose the correct MV-algebra under Settings:

$$\mathcal{T} : [0,1]^S \to [0,1]^S$$

$$\mathcal{T}(t)(s) = \begin{cases} 1 & \text{if } s \in T \\ \sum\limits_{s' \in S} \eta(s)(s') \cdot t(s') & \text{otherwise} \end{cases}$$

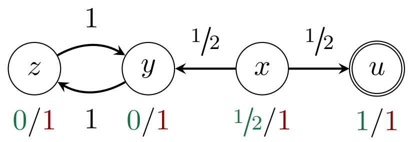
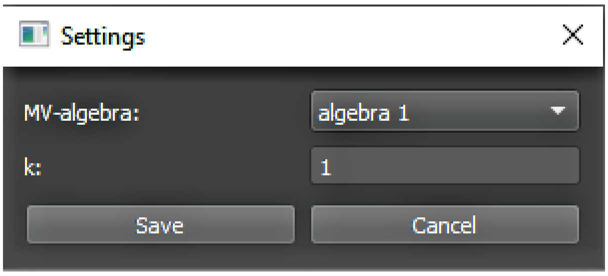


Fig. 4.8.: Function $\mathcal{T}$ (left) and a Markov chain with two fixpoints of $\mathcal{T}$ (right)



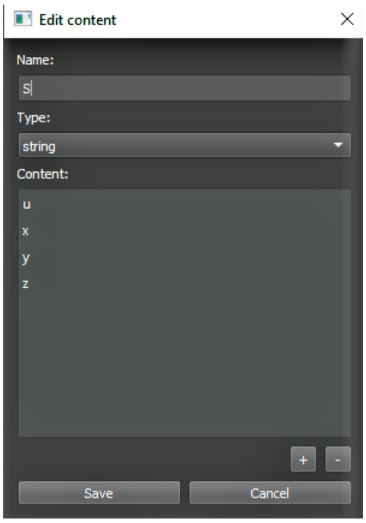
The function $\mathcal{T}$ can be decomposed as follows (cf. Section 3.6.1):

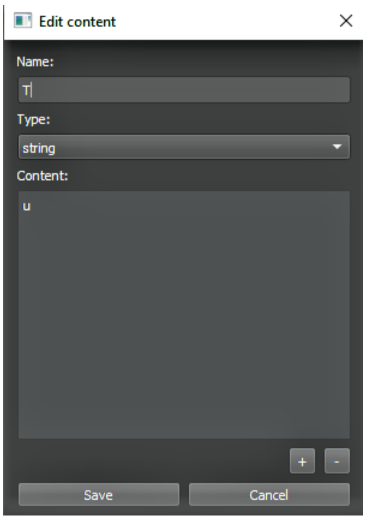
$$\mathcal{T} = (\eta^* \circ \mathrm{av}_D) \uplus c_k$$

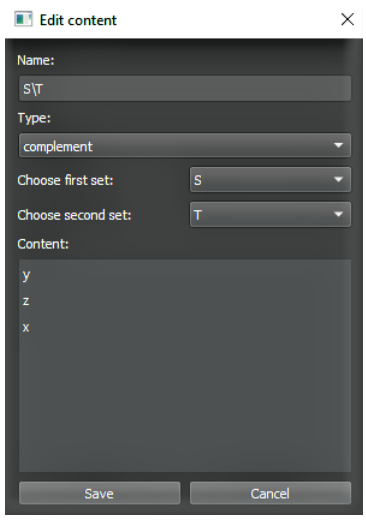
and for the transition system in Figure 4.8 we have

- $c_k\colon [0,1]^\emptyset \to [0,1]^T$, $k\colon T \to [0,1]$ with $k(u) = 1$
- $\mathrm{av}_D\colon [0,1]^S \to [0,1]^D$ with $D = \{p_x, p_y, p_z\} \subseteq \mathcal{D}(S)$ where
  - $p_x(y) = p_x(u) = 1/2$
  - $p_y(z) = p_z(y) = 1$
- $\eta^*\colon [0,1]^D \to [0,1]^{S \setminus T}$ with $\eta(j) = p_j$ for $j \in \{x, y, z\}$

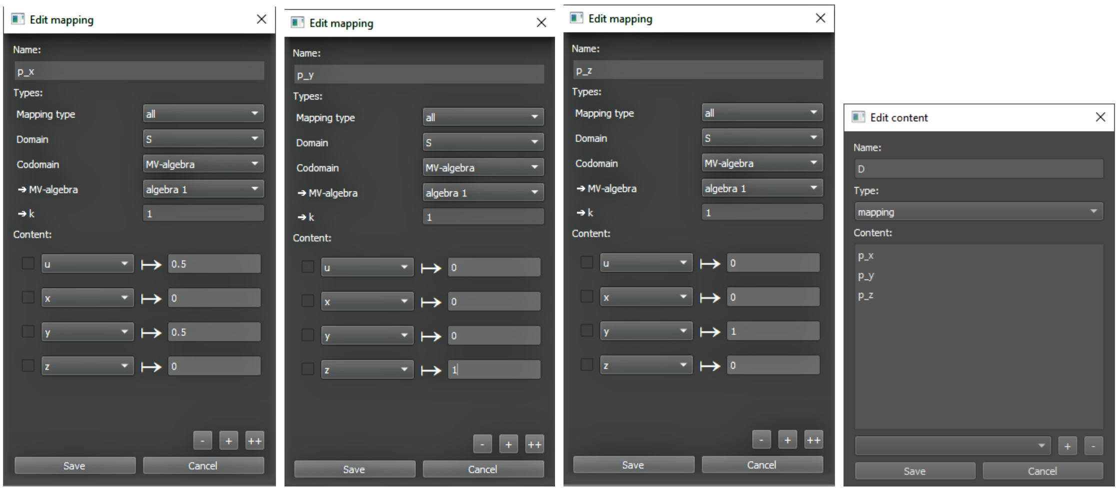
We first insert Input and Output sets. Thus we create the sets $S, T, S \setminus T$:



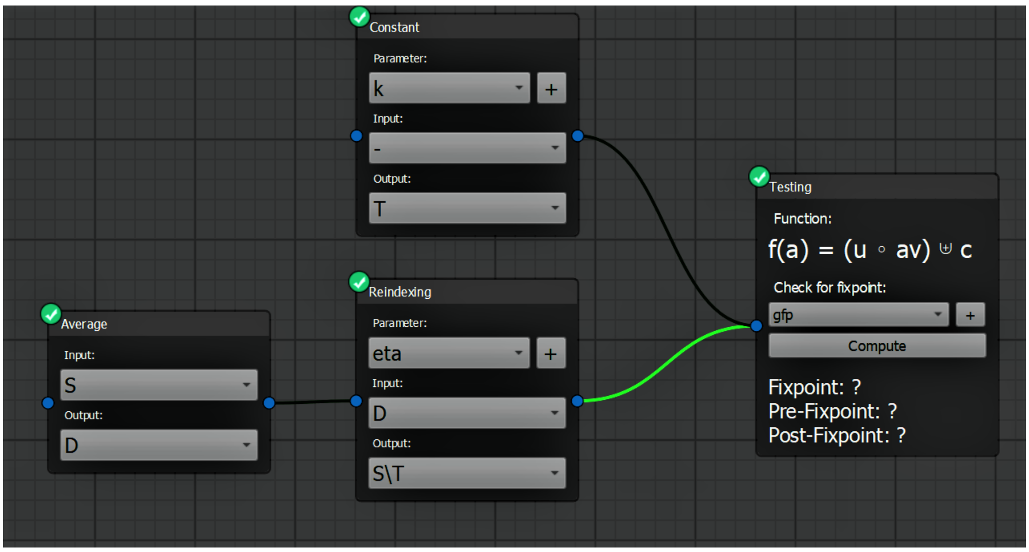




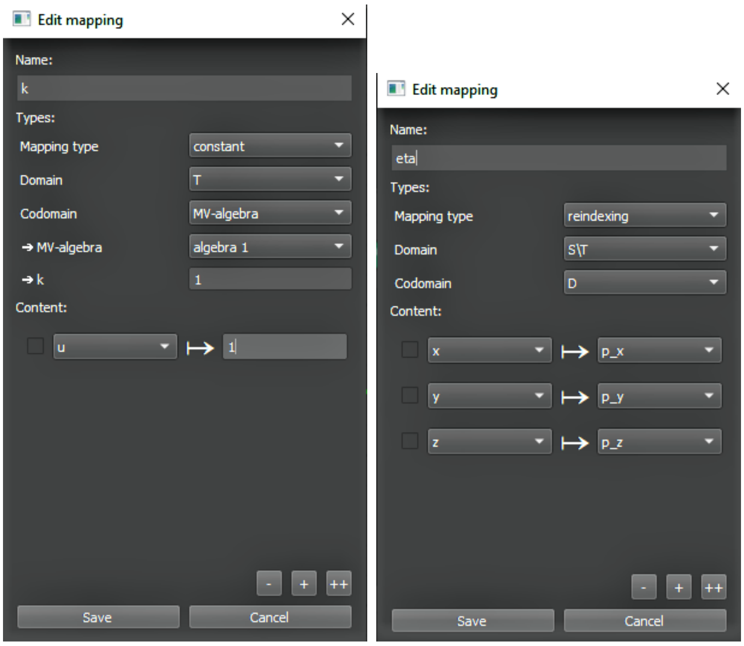
The tool supports a few types of sets and operators on sets (such as complement in the right picture) which eases the use. Next, we create the set D where we first need to create the mappings p_x, p_y, p_z:



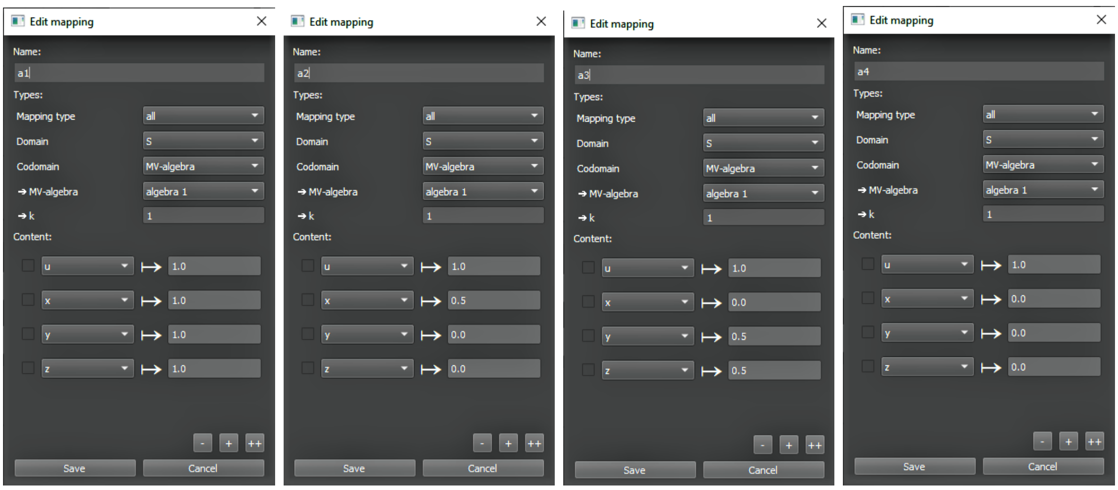
Now we create the basic function boxes and connect them in the correct way:



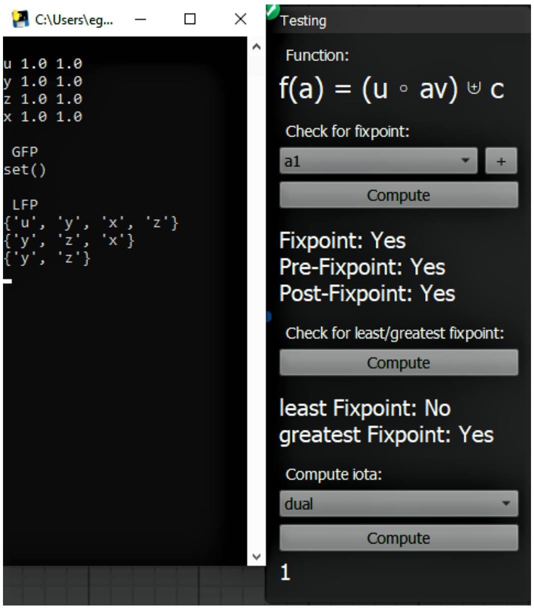
The additional parameters according to Table 4.1 can be easily created by clicking the "+" in the corresponding box:



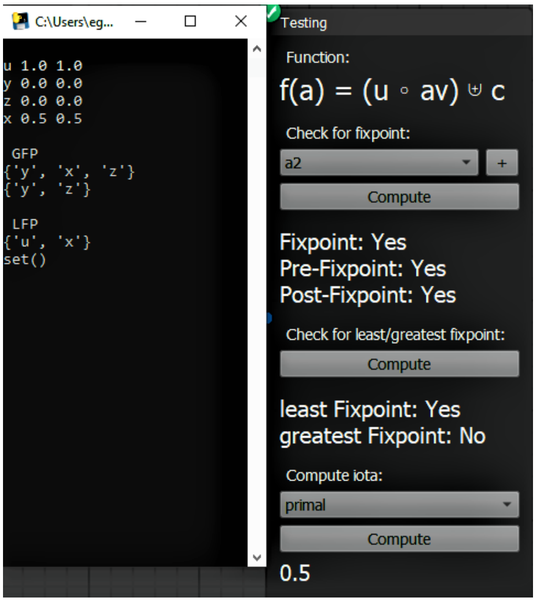
Now the function is successfully created and we only need to insert some test functions. We test $a_1 = \nu\mathcal{T}$, $a_2 = \mu\mathcal{T}$ and the post-fixpoints $a_3(x) = 0, a_3(y) = a_3(z) = 0.5, a_3(u) = 1$ and $a_4(x) = 0, a_4(y) = a_4(z) = 0, a_4(u) = 1$:



When testing a_1 we obtain

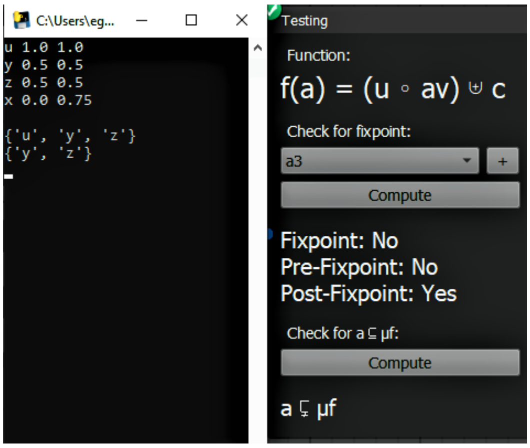


which tells us that a_1 is indeed the greatest fixpoint but not the least fixpoint, i.e. $\nu\mathcal{T}^{a_1}_{\#} = \{y, z\}$ and $a_1 \ominus 1_{\nu\mathcal{T}^{a_1}_{\#}} \sqsupseteq \mu\mathcal{T}$. For a_2 we obtain

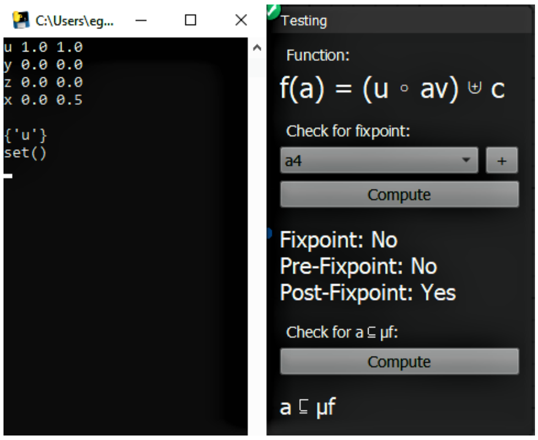


which means us that a_2 is indeed the least fixpoint but not the greatest fixpoint, i.e. $\nu\mathcal{T}^{\#}_{a_2} = \{y, z\}$ and $a_2 \oplus \frac{1}{2}_{\nu\mathcal{T}^{\#}_{a_2}} \sqsubseteq \nu\mathcal{T}$.

For the post-fixpoint a_3 the tool displays



reinforcing that a_3 is indeed a post-fixpoint and that $\mathcal{T}_*^{a_3} = \{y, z\}$, i.e. we can not conclude any relation between a_3 and $\mu\mathcal{T}$[5]. Clearly, $a_3 \sqsubseteq \mu\mathcal{T}$ does not hold but a_3 still contains a vicious cycle. Lastly, for a_4 we obtain



which tells us that a_4 is indeed a post-fixpoint and that $\mathcal{T}_*^{a_4} = \emptyset$, thus $a_4 \sqsubseteq \mu\mathcal{T}$ holds.

Examples: There are pre-defined functions, implementing examples, that are shipped with the tool. These concern case studies on termination probability, bisimilarity, simple stochastic games, energy games, behavioural metrics and Rabin automata.

4.8. Summary

We have shown how our framework from Chapter 3 can be cast into a gs-monoidal setting, justifying the development of the tool UDEfix for a compositional view on fixpoint checks. In addition we studied properties of the gs-monoidal functor $\#$, mapping from the concrete to the abstract universe and giving us a general procedure for approximating

[5] Remember that Lemma 3.4.2 does not provide a complete proof rule for post-fixpoints

predicate liftings and the Wasserstein lifting. Here, we showed and used the fact that our theory allows intermediate sets (of a decomposed function) to be infinite.

5 A Lattice-Theoretical View of Strategy Iteration

As we have seen in Section 3 there are several applications to the theory we developed. The theory enables us to check if some given fixpoint $a \in \mathbb{M}^Y$ of some non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ is in fact the least or the greatest fixpoint. Additionally, if a is not the least/greatest fixpoint, we can decrease/increase to some pre-/postfixpoint $b \in \mathbb{M}^Y$ which therefore lies above/below the least/greatest fixpoint.

The question remains unanswered how one can compute the least/greatest fixpoint via an algorithm. Since $\mathbb{M}$ is a complete MV-chain and thus a complete lattice, we can immediately apply Kleene iteration to iterate towards the least/greatest fixpoint. It is however known that Kleene iteration usually only yields an approximation of the least/greatest fixpoint and not exact computations.

However, the theory in Chapter 3 enables us to exactly compute the least/greatest fxipoint via strategy iteration. We are positioned to detail strategy iteration algorithms in a general framework which encompasses a handful of already existing strategy iterations found in the literature.

5.1. Introduction

Strategy iteration (or policy iteration) is a well known technique in computer science. It has been widely adopted for the solution of two-player games (cf. Section 2.7) where the players, Max and Min, aim at maximising and minimising, respectively, some payoff. In many cases there exists an optimal strategy for each player where no deviation is advisable as long as the other player plays optimally. We here assume a scenario where memoryless (or positional) strategies are sufficient. The general idea of strategy iteration is to iteratively fix a strategy for one player, compute the optimal answering strategy for the other player and then improve the strategy of the first player. As long as there are only finitely many strategies, an optimal strategy is bound to be found at some point. Such strategy iteration methods exist for Markov decision processes [How60] and for a variety of games, such as simple stochastic games [Con92, KH66, ABdMS21], (discounted) mean-payoff games [ZP96, BC10] and parity games [VJ00, Sch08a].

Similar ideas apply also to a wide range of different problems. For instance, the computation of behavioural distances for systems (cf. Section 2.6) embodying quantitative information, e.g., time, probability or cost, is often based on some form of lifting of distances on states [BBLM17, BBKK18, BKP18]. In turn the lifting relies on couplings which play the role of strategies and algorithms based on a progressive improvement of couplings have been devised [BBLM17, BBL+21].

Motivating Example. To help with the intuition, we will detail a strategy iteration for simple stochastic games (cf. Section 2.7.3). When both players Min and Max play optimally, the expected payoff at each state is given by the least fixpoint of the function $\mathcal{V}: [0,1]^V \to [0,1]^V$, defined for $a: V \to [0,1]$ and $v \in V$ by

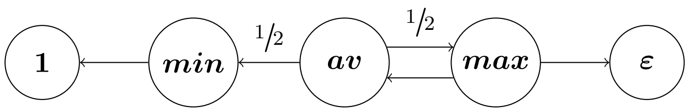
$$\mathcal{V}(a)(v) = \begin{cases} \max_{v \to v'} a(v') & v \in V_{\mathsf{Max}} \\ \min_{v \to v'} a(v') & v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & v \in V_{\mathsf{Av}} \\ c(v) & v \in V_{\mathsf{Sink}} \end{cases}$$

The idea of strategy iteration from below, instantiated to this context, is to compute the least fixpoint $\mu\mathcal{V}$ via an iteration of the following kind:

1. Guess a strategy $\sigma: V_{\mathsf{Max}} \to V$ for player Max, i.e., fix a successor for states in V_{Max}.
2. Compute the least fixpoint of $\mathcal{V}_\sigma : [0,1]^V \to [0,1]^V$, which is defined as $\mathcal{V}$ in all cases apart from $v \in V_{\mathsf{Max}}$, where we set $\mathcal{V}(a)(v) = a(\sigma(v))$. This fixpoint computation is simpler than the original one and it can be done efficiently via linear programming.
3. Based on $\mu\mathcal{V}_\sigma$, try to improve the strategy for Max. If the strategy does not change, we have computed a fixpoint of $\mathcal{V}$ and, since iteration is from below, this is necessarily the least fixpoint. If the strategy changes, continue with step 2.

This procedure is well-known to work for stopping games [Con92], i.e., simple stochastic games where each combination of strategies ensures termination with probability 1, since for these games $\mathcal{V}$ has a unique fixpoint.

Example 5.1.1. *Consider the following simple stochastic game where* $\mathbf{min} \in V_{Min}$, $\mathbf{max} \in V_{Max}$, $\boldsymbol{av} \in V_{Av}$ *and states* $\mathbf{1}, \boldsymbol{\varepsilon} \in V_{Sink}$ *have payoff* 1 *and* $\varepsilon \in (0,1)$, *respectively:*



This simple stochastic game is clearly stopping and we have

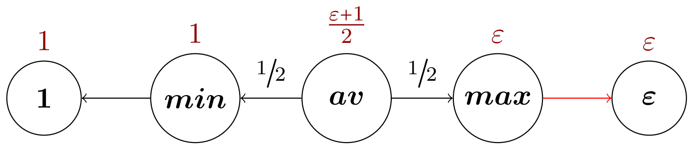
$$\mathcal{V}(a)(v) = \begin{cases} \max\{a(\boldsymbol{\varepsilon}), a(\boldsymbol{av})\} & \textit{if } v = \mathbf{max} \\ \min\{a(\mathbf{1})\} & \textit{if } v = \mathbf{min} \\ 1/2 \cdot a(\mathbf{min}) + 1/2 \cdot a(\mathbf{max}) & \textit{if } v = \boldsymbol{av} \\ 1 & \textit{if } v = \mathbf{1} \\ \varepsilon & \textit{if } v = \boldsymbol{\varepsilon} \end{cases}$$

for $a \in [0,1]^V$. *We demonstrate the sketched algorithm above to illustrate the basic ideas of strategy iteration.*

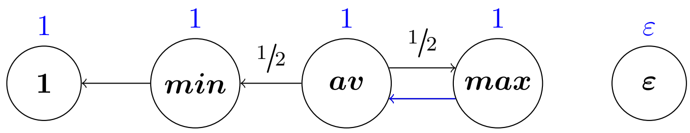
First, Max guesses the strategy $\sigma(\mathbf{max}) = \boldsymbol{\varepsilon}$. We have

$$\mathcal{V}_\sigma(a)(v) = \begin{cases} a(\sigma(\mathbf{max})) = a(\boldsymbol{\varepsilon}) & \text{if } v = \mathbf{max} \\ a(\mathbf{1}) & \text{if } v = \mathbf{min} \\ 1/2 \cdot a(\mathbf{min}) + 1/2 \cdot a(\mathbf{max}) & \text{if } v = \boldsymbol{av} \\ 1 & \text{if } v = \mathbf{1} \\ \varepsilon & \text{if } v = \boldsymbol{\varepsilon} \end{cases}$$

for $a \in [0,1]^V$ and obtain the following expected payoff in red corresponding to $\mu\mathcal{V}_\sigma$ (we will discuss how to compute $\mu\mathcal{V}_\sigma$ in Section 5.3.2):



Now, Max realizes that the expected payoff of his successor $\boldsymbol{av}$ is higher than the payoff of state $\boldsymbol{\varepsilon}$ (i.e. $\mu\mathcal{V}_\sigma(\boldsymbol{av}) > \mu\mathcal{V}_\sigma(\boldsymbol{\varepsilon})$) and thus Max switches his strategy to $\sigma'(\mathbf{max}) = \boldsymbol{av}$. This results in the following expected payoffs in blue corresponding to $\mu\mathcal{V}_{\sigma'}$:



σ' is an optimal positional strategy for Max (cf. Lemma 2.7.18) and it holds $\mu\mathcal{V}_{\sigma'} = \mu\mathcal{V}$.

For general simple stochastic games, when iterating from below we rely on the theory from Chapter 3 as finding an improving strategy for player Max is not as simple as the example above suggests.

A similar approach can be used for converging to the least fixpoint from above. In this case, it is now player Min who fixes a strategy which is progressively improved. This procedure works fine for stopping simple stochastic games. However, for general simple stochastic games, when iterating from above the procedure may get stuck at some fixpoint which is not the least fixpoint of $\mathcal{V}$, a problem which is solved by the theory developed in Chapter 3 which can be used to "skip" this fixpoint and continue the iteration from there.

While, as explained above, the general idea of strategy iteration is used in many different settings, to the best of our knowledge a general definition of strategy iteration is still missing. The goal of this chapter is to provide a general and abstract formulation of an algorithm for strategy iteration, proved correct once and for all, which instantiates to a variety of problems. The key observation is that optimal strategies very often arise from some form of extremal (least or greatest) fixpoint of a suitable non-expansive function f over a complete MV-chain, the paradigmatic example being the real interval $[0,1]$ with the usual order. We propose a framework where the operation of fixing a strategy for one of the players is captured abstractly, in terms of so-called min- or max-decompositions of the function of interest. Then, we devise strategy iteration approaches which converge to the fixpoint of interest by successively improving the strategy for the chosen player. We will assume that the interest is in least fixpoints, but the theory can be dualised. We propose two strategy iteration algorithms that converge to the least fixpoint "from below" and "from above", respectively. As it happens for simple stochastic games, in the latter case the iteration can reach a fixpoint which is not the least. Clearly, whenever the function f of interest has a unique fixpoint this problem disappears. Moreover, in some cases, even though f has multiple fixpoints, it can be "patched" in a way that the modified function has the fixpoint of interest as its only fixpoint. Otherwise, we can rely on the results from Chapter 3 to check whether the reached fixpoint is the least one and whenever it is not, to get closer to the desired fixpoint and continue the iteration.

Strategy iteration approaches can be slow if compared to other algorithms, such as value iteration. However, the benefit of strategy iteration algorithms is that they allow an exact computation of the desired fixpoint, while other algorithms may never reach the sought-after extreme fixpoint but only converge towards it. This is the case, e.g., for simple stochastic games, where strategy iteration algorithms are the standard methods to obtain exact results. Additionally, strategy iteration, besides determining the fixpoint also singles out an optimal strategy which allows one to obtain it, an information which is often of interest.

In summary, we propose the first, to the best of our knowledge, general definition of strategy iteration providing a lattice-theoretic formalisation of this technique. This requires to single out and solve in this general setting the fundamental challenges of these approaches, which already show up in earlier work on simple stochastic games (see, e.g., [BC10]). In the iteration from above, we may converge to a fixpoint that is not the least, while from below it is not straightforward to show that improving the strategy of Max leads to a larger fixpoint.

We will rediscover known algorithms for simple stochastic games, labeled Markov chains [TvB17] and probabilistic automata [BBL$^+$21]. Moreover new ones are obtained for simple stochastic games, discounted mean-payoff games, energy games, bisimilarity and metric transition systems. Given the number of different application domains where strategy iteration is or can be used, we feel that a general framework can unveil unexplored potentials.

The rest of this chapter is structured as follows. In Section 5.2 we devise two generalized strategy iteration algorithms, from above and from below, using simple stochastic games as a running example. In Section 5.3, we show how our technique applies to our applications, while in Section 5.4 we perform a short runtime comparison which aims to justify the use of strategy iteration.

5.2. Generalized Strategy Iteration

In this section we develop two strategy iteration techniques for determining least fixpoints. The first technique requires a so-called min-decomposition and approaches the least fixpoint from above, while the second uses a max-decomposition to ascend to the least fixpoint from below.

Hence fixpoint iteration from above is seen strictly from the point of view of the Min player, while fixpoint iteration from below is from the view of the Max player, who want to minimize respectively maximize the payoff. The player starts by guessing a strategy, which in the case of the Min (Max) player over-approximates (under-approximates) the true payoff. This strategy is then locally improved at each iteration based on the payoff produced by the player following such a strategy. That is, we compute fixpoints for a fixed strategy, which in a two-player game means that the opponent plays optimally. When the set of strategies is finite (or, at least, the search can be restricted to a finite set), an optimal strategy will be found at some point.

5.2.1. Function Decomposition

We next introduce the setting where the generalisations of strategy iteration will be developed. We assume that the game we are interested in is played on a finite set of positions Y and the payoff at each position is an element of a suitable complete MV-chain $\mathbb{M}$. This payoff is given by a function in $\mathbb{M}^Y$ that can be characterised as the least fixpoint of a monotone function $f : \mathbb{M}^Y \to \mathbb{M}^Y$. If we concentrate on the Min player, each position $y \in Y$ is assigned a set of functions $H_{\min}(y) \subseteq (\mathbb{M}^Y \to \mathbb{M})$ where each function $h \in H_{\min}(y)$ is one possible option that can be chosen by Min. Given $a\colon Y \to \mathbb{M}$ as the current estimate of the payoff, $h(a)$ is the resulting payoff at y. If the player does not have a choice, this set is a singleton. Since it is the aim of Min to minimise he will choose an h such that $h(a)$ is minimal.

Definition 5.2.1 (min-decomposition)**.** *Let Y be a finite set and $\mathbb{M}$ be a complete MV-chain. Given a function $f : \mathbb{M}^Y \to \mathbb{M}^Y$, a **min-decomposition** of f is a function $H_{\min} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$[1] such that for all $y \in Y$ the set $H_{\min}(y)$ consists only of monotone functions and for all $a \in \mathbb{M}^Y$ it holds*

$$f(a)(y) = \min_{h \in H_{\min}(y)} h(a).$$

Max-decompositions, with analogous properties, are defined dually:

[1] i.e. we assign a function $h\colon \mathbb{M}^Y \to \mathbb{M}$ to each state $y \in Y$

Definition 5.2.2 (max-decomposition)**.** *Let Y be a finite set and $\mathbb{M}$ be a complete MV-chain. Given a function $f : \mathbb{M}^Y \to \mathbb{M}^Y$, a* ***max-decomposition*** *of f is a function $H_{\max} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ such that for all $y \in Y$ the set $H_{\max}(y)$ consists only of monotone functions and for all $a \in \mathbb{M}^Y$ it holds*

$$f(a)(y) = \max_{h \in H_{\max}(y)} h(a).$$

Observe that any monotone function $f : \mathbb{M}^Y \to \mathbb{M}^Y$ admits a trivial min/max-decomposition I defined by $I(y) = \{h_y\}$ where $h_y(a) = f(a)(y)$ for all $a \in \mathbb{M}^Y$.

Example 5.2.3. *As a running example for illustrating our theory and the resulting algorithms we will use simple stochastic games (cf. Section 2.7.3). We analyze this application in more detail in Section 5.3.2.*

The fixpoint function $\mathcal{V} : [0,1]^V \to [0,1]^V$ which was defined for $a : V \to [0,1]$ and $v \in V$ as follows:

$$\mathcal{V}(a)(v) = \begin{cases} \max_{v' \in succ(v)} a(v') & \text{if } v \in V_{Max} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{Min} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{Av} \\ c(v) & \text{if } v \in V_{Sink} \end{cases}$$

$\mathcal{V}$ admits a min-decomposition $H_{\min} : V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ defined for all $a \in \mathbb{M}^Y$ as follows:

- *for $v \in V_{Min}$, $H_{\min}(v) = \{h_{v'} \mid v' \in succ(v)\}$ with $h_{v'}(a) = a(v')$;*
- *for $v \in V_{Max}$, $H_{\min}(v) = \{h\}$ with $h(a) = \max_{v' \in succ(v)} a(v') = \mathcal{V}(a)(v)$;*
- *for $v \in V_{Av}$, $H_{\min}(v) = \{h\}$ with $h(a) = \sum_{v' \in V} p(v)(v') \cdot a(v') = \mathcal{V}(a)(v)$;*
- *for $v \in V_{Sink}$, $H_{\min}(v) = \{h\}$ with $h(a) = c(v) = \mathcal{V}(a)(v)$.*

A max-decomposition can be defined dually, e.g. $H_{\max} : V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ is defined for all $a \in \mathbb{M}^Y$ as follows:

- *for $v \in V_{Max}$, $H_{\max}(v) = \{h_{v'} \mid v' \in succ(v)\}$ with $h_{v'}(a) = a(v')$;*
- *for $v \in V_{Min}$, $H_{\max}(v) = \{h\}$ with $h(a) = \min_{v' \in succ(v)} a(v') = \mathcal{V}(a)(v)$;*
- *for $v \in V_{Av}$, $H_{\max}(v) = \{h\}$ with $h(a) = \sum_{v' \in V} p(v)(v') \cdot a(v') = \mathcal{V}(a)(v)$;*
- *for $v \in V_{Sink}$, $H_{\max}(v) = \{h\}$ with $h(a) = c(v) = \mathcal{V}(a)(v)$.*

For instance, consider the simple stochastic game in Fig. 5.1 where we have $V = \{\mathbf{1}, \boldsymbol{\varepsilon}, \boldsymbol{av}, \boldsymbol{max}, \boldsymbol{min}\}$ with the obvious partitioning. The fixpoint function is $\mathcal{V} : [0,1]^V \to [0,1]^V$ defined, for $a \in [0,1]^V$, by

$$\mathcal{V}(a)(\mathbf{1}) = 1 \qquad \mathcal{V}(a)(\boldsymbol{\varepsilon}) = \varepsilon \qquad \mathcal{V}(a)(\boldsymbol{av}) = \frac{1}{2}a(\boldsymbol{min}) + \frac{1}{2}a(\boldsymbol{max})$$

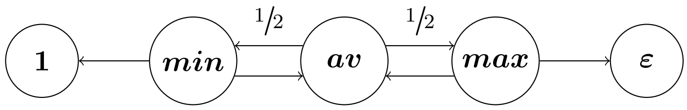


Fig. 5.1.: An example of a simple stochastic game. States $\mathbf{1}$, $\boldsymbol{\varepsilon}$ have payoff 1, $\varepsilon > 0$ respectively.

$$\mathcal{V}(a)(\boldsymbol{max}) = \max\{a(\boldsymbol{\varepsilon}), a(\boldsymbol{av})\} \qquad \mathcal{V}(a)(\boldsymbol{min}) = \min\{a(\mathbf{1}), a(\boldsymbol{av})\}.$$

The min-decomposition defined in general above, in this case is $H_{\min}: V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ *defined for all* $a \in \mathbb{M}^Y$ *as follows:*

- $H_{\min}(v) = \{h\}$ *with* $h(a) = \mathcal{V}(a)(v)$ *for all* $v \in V \setminus \{\boldsymbol{min}\}$
- $H_{\min}(\boldsymbol{min}) = \{h_{\mathbf{1}}, h_{\boldsymbol{av}}\}$ *with* $h_{\mathbf{1}}(a) = a(\mathbf{1})$ *and* $h_{\boldsymbol{av}}(a) = a(\boldsymbol{av})$.

Dually, a max-decomposition $H_{\max}: V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ *is defined for all* $a \in \mathbb{M}^Y$ *as follows:*

- $H_{\max}(v) = \{h\}$ *with* $h(a) = \mathcal{V}(a)(v)$ *for all* $v \in V \setminus \{\boldsymbol{max}\}$
- $H_{\max}(\boldsymbol{max}) = \{h_{\boldsymbol{\varepsilon}}, h_{\boldsymbol{av}}\}$ *with* $h_{\boldsymbol{\varepsilon}}(a) = a(\boldsymbol{\varepsilon})$ *and* $h_{\boldsymbol{av}}(a) = a(\boldsymbol{av})$.

Whenever all $h \in H_{\min}(y)/H_{\max}(y)$ are not only monotone, but also non-expansive, it can be shown easily that f is also non-expansive and we can obtain an approximation as discussed in Chapter 3.

5.2.2. Derivation of the Approximations

In this section we show how a min-decomposition (analogously a max-decomposition) of a mapping $f: \mathbb{M}^Y \to \mathbb{M}^Y$ can be assembled using the basic functions and operators introduced in Section 3.5. This serves two purposes: in this way we show that f is automatically non-expansive if obtained from non-expansive components. Second, this gives us a recipe to obtain the approximation $f^a_{\#}$, required for checking whether a given fixpoint is indeed the least.

Table 3.1 lists the basic non-expansive functions we require for the proof below and operators for composing them. Note that all those functions are non-expansive and the operators preserve non-expansiveness. In addition the table lists the corresponding approximations.

We will now show how to obtain the approximation of a function f given its min-decomposition and approximations for all the functions used in the min-decomposition.

Proposition 5.2.4. *Let Y be a finite set and $\mathbb{M}$ a complete MV-chain. Let $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ be a function and $H_{\min}\colon Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ a given min-decomposition such that, for all $y \in Y$, all functions $h \in H_{\min}(y)$ are non-expansive. Then f is non-expansive and the approximation $f^a_\#(Y')\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ is given by*

$$f^a_\#(Y') = \{y \in [Y]^{f(a)} \mid \exists h \left(h = \arg\min_{h' \in H_{\min}(y)} h'(a) \ \wedge \ h^a_\#(Y') \neq \emptyset\right)\}$$

for $a \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^a$.

Proof. See Appendix: Proposition A.4.1. □

In a similar fashion (same proof until the last step), for a max-decomposition $H_{\max}$ of f, the approximation $f^a_\#\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ is given by

$$f^a_\#(Y') = \{y \in [Y]^{f(a)} \mid \forall h \left(h = \arg\max_{h' \in H_{\max}(y)} h'(a) \ \Rightarrow \ h^a_\#(Y') \neq \emptyset\right)\}$$

for $a \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^a$.

We also obtain $f^\#_a\colon \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_{f(a)})$ defined for $a \in \mathbb{M}^Y$, $Y \subseteq [Y]_a$ and

- a min-decomposition of f (i.e. $f(a)(y) = \min_{h \in H_{\min}(y)} h(a)$):

$$f^\#_a(Y') = \{y \in [Y]_{f(a)} \mid \forall h \left(h = \arg\min_{h' \in H_{\min}(y)} h'(a) \ \Rightarrow \ h^\#_a(Y') \neq \emptyset\right)\}$$

- a min-decomposition of f (i.e. $f(a)(y) = \max_{h \in H_{\max}(y)} h(a)$):

$$f^\#_a(Y') = \{y \in [Y]_{f(a)} \mid \exists h \left(h = \arg\max_{h' \in H_{\max}(y)} h'(a) \ \wedge \ h^\#_a(Y') \neq \emptyset\right)\}.$$

5.2.3. Strategies

Fixing a strategy can be seen as fixing, for all $y \in Y$, some element in $H_{\min}(y)$.

Definition 5.2.5 (strategy)**.** *Let Y be a finite set, $\mathbb{M}$ be a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ and let $H_{\min} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ be a min-decomposition of f. A **strategy** in $H_{\min}$ is a function $C : Y \to (\mathbb{M}^Y \to \mathbb{M})$ such that for all $y \in Y$ it holds that $C(y) \in H_{\min}(y)$. For a fixed C we define $f_C : \mathbb{M}^Y \to \mathbb{M}^Y$ as $f_C(a)(y) = C(y)(a)$ for all $a \in \mathbb{M}^Y$ and $y \in Y$.*
Strategies in a max-decomposition are defined dually, i.e. $C(y) \in H_{\max}(y)$ for all $y \in Y$.

The letter C stands for "choice" and typically μf_C is easier to compute than μf.

Example 5.2.6. *Reconsider the simple stochastic game in Fig. 5.1. As discussed in Example 5.2.3, we obtain the following min-decomposition $H_{\min}\colon V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ for all $a \in \mathbb{M}^Y$: For $v \in V \setminus \{\boldsymbol{min}\}$, we have*

$$H_{\min}(v) = \{h\} \text{ with } h(a) = \mathcal{V}(a)(v),$$

while

$$H_{\min}(\boldsymbol{min}) = \{h_{\mathbf{1}}, h_{\boldsymbol{av}}\} \text{ with } h_{\mathbf{1}}(a) = a(\mathbf{1}) \text{ and } h_{\boldsymbol{av}}(a) = a(\boldsymbol{av}).$$

All strategies in $H_{\min}$ assign to every state $v \in V \setminus \{\boldsymbol{min}\}$ the only element in $H_{\min}(v)$. Hence they are determined by the value on state ***min****: thus there are two strategies $C_1^{\min}, C_2^{\min}$ in $H_{\min}$ with $C_1^{\min}(\boldsymbol{min}) = h_{\mathbf{1}}$ and $C_2^{\min}(\boldsymbol{min}) = h_{\boldsymbol{av}}$.*

The max-decomposition $H_{\max}: V \to \mathcal{P}_f(\mathbb{M}^V \to \mathbb{M})$ was defined for all $a \in \mathbb{M}^Y$ as follows:

$$H_{\max}(v) = \{h\} \text{ with } h(a) = \mathcal{V}(a)(v)$$

for all $v \in V \setminus \{\boldsymbol{max}\}$ and

$$H_{\max}(\boldsymbol{max}) = \{h_{\boldsymbol{\varepsilon}}, h_{\boldsymbol{av}}\} \text{ with } h_{\boldsymbol{\varepsilon}}(a) = a(\boldsymbol{\varepsilon}) \text{ and } h_{\boldsymbol{av}}(a) = a(\boldsymbol{av}).$$

Again, there are two strategies $C_1^{\max}$ and $C_2^{\max}$ in $H_{\max}$ that differ for the value assigned to ***max****: $C_1^{\max}(\boldsymbol{max}) = h_{\boldsymbol{\varepsilon}}$ and $C_2^{\max}(\boldsymbol{max}) = h_{\boldsymbol{av}}$.*

The following lemma reports two easy observations which will be used several times.

Lemma 5.2.7. *Let Y be a finite set, $\mathbb{M}$ be a complete MV-chain and let $H_{\min} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ be a min-decomposition of $f: \mathbb{M}^Y \to \mathbb{M}^Y$.*

1. *For all strategies C in $H_{\min}$ we have that $f \sqsubseteq f_C$, pointwise.*
2. *For all $a \in \mathbb{M}^Y$, there is a strategy C_a such that $C_a(y)(a) = f(a)(y)$ for all $y \in Y$.*

Proof. (1) Just note that for all $a \in \mathbb{M}^Y$ and $y \in Y$, we have

$$\begin{aligned} f_C(a)(y) &= C(y)(a) \\ &\sqsupseteq \min_{h \in H_{\min}(y)} h(a) && [\text{since } C(y) \in H_{\min}(y)] \\ &= f(a)(y) && [\text{by definition of min-decomposition}] \end{aligned}$$

(2) For all $y \in Y$, it holds $f(a)(y) = \min_{h \in H_{\min}(y)} h(a) = h_y(a)$ for some $h_y \in H_{\min}(y)$ since the minimum is realised ($H_{\min}$ is finite). And thus we can define $C_a(y) = h_y$. □

The above result can be easily adapted for a max-decomposition by reversing the order.

5.2.4. Strategy Iteration from Above

In this section we propose a generalized strategy iteration algorithm from above. It is based on a min-decomposition of the function and, intuitively, at each iteration the player Min improves her strategy. An issue here is that this iteration may get stuck at a fixpoint strictly larger than the least one. Recognising and overcoming this problem, thus continuing the iteration until the least fixpoint is reached, requires the theory described in Chapter 3.

The basic result that motivates strategy iteration from above is a characterisation of the least fixpoint of a function in terms of a min-decomposition.

Proposition 5.2.8 (least fixpoint from min-decompositions)**.** *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a monotone function and let $H_{\min} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ be a min-decomposition of f. Then*

$$\mu f = \min\{\mu f_C \mid C \text{ is a strategy in } H_{\min}\}.$$

Proof. By Lemma 5.2.7(1), for all strategies C in $H_{\min}$ we have $f \sqsubseteq f_C$, whence $\mu f \sqsubseteq \mu f_C$. Then $\mu f \sqsubseteq \min\{\mu f_C \mid C \text{ is a strategy in } H_{\min}\}$ follows.

For the converse inequality, note that by Lemma 5.2.7(2) there exists some strategy C in $H_{\min}$, such that $f_C(\mu f)(y) = C(y)(\mu f) = f(\mu f)(y) = \mu f(y)$ for all $y \in Y$. Thus μf is a (pre-)fixpoint of f_C and thus $\mu f_c \sqsubseteq \mu f$. Hence $\mu f \sqsupseteq \min\{\mu f_C \mid C \text{ is a strategy in } H_{\min}\}$. □

Although we do not focus on complexity issues, we observe that – under suitable assumptions – we can show that given a function f as a min-decomposition, the problem of checking whether $\mu f \sqsubseteq b$ for some bound $b \in \mathbb{M}^Y$ is in NP. For each $y \in Y$ we can nondeterministically guess $C(y) \in H_{\min}(y)$ thus defining a strategy. Assuming that the computation of μf_C is polynomial, we can thus determine in non-deterministic polynomial time (in the size of the representation of f) whether $\mu f \sqsubseteq \mu f_C \sqsubseteq b$.

Now in order to compute the least fixpoint μf, the idea is to start from some (arbitrary) strategy, say C_0, in $H_{\min}$. At each iteration, if the current strategy is C_i one tries to construct, on the basis of μf_{C_i}, a new strategy C_{i+1} which improves C_i, in the sense that $\mu f_{C_{i+1}}$ becomes smaller. This motivates the notion of improvement.

Definition 5.2.9 (min-improvement)**.** *Let $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a monotone function, where Y is a finite set and $\mathbb{M}$ a complete MV-chain, and let $H_{\min}$ be a min-decomposition. Given strategies C, C' in $H_{\min}$, we say that C' is a* ***min-improvement*** *of C if $f_{C'}(\mu f_C) \sqsubset \mu f_C$. It is called a* ***stable*** *min-improvement if in addition $C'(y) = C(y)$ for all $y \in Y$ such that $f_{C'}(\mu f_C)(y) = \mu f_C(y)$. We denote by $imp_{\min}(C)$ (respectively $imp^s_{\min}(C)$) the set of (stable) min-improvements of C.*

The notion of stability will turn out to be useful later, for performing strategy iteration from below (as explained in the next section). In a stable min-improvement, the player is only allowed to switch the strategy in a state if this yields a strictly better payoff. Interestingly, instances of this notion are adopted, more or less implicitly, in other strategy improvement algorithms in the literature (cf. [ABdMS21, Definition 13] and the way in which improvements are computed in [BC10]). Clearly $imp^s_{\min}(C) \subseteq imp_{\min}(C)$. In addition, it can be easily seen that there exists a stable min-improvement as long as there is any improvement.

Remark 5.2.10 (obtaining min-improvements). *For a strategy C, if $imp_{\min}(C) \neq \emptyset$, one can obtain a min-improvement of C by taking $C' \neq C$ defined as*

$$C'(y) = \arg \min_{h \in H_{\min}(y)} h(\mu f_C)$$

and a stable min-improvement as:

$$C'(y) = \begin{cases} C(y) & \text{if } f(\mu f_C)(y) = \mu f_C(y) \\ \arg\min_{h \in H_{\min}(y)} h(\mu f_C) & \text{otherwise} \end{cases}$$

There could be several $h \in H_{\min}(y)$ where $h(\mu f_C)$ is minimal. Any such choice is valid.

We next show that, as suggested by the terminology, a min-improvement leads to a smaller least fixpoint.

Lemma 5.2.11 (min-improvements reduce fixpoints). *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a monotone function and $H_{\min}$ a min-decomposition of f. Given a strategy C in $H_{\min}$ and a min-improvement $C' \in imp_{\min}(C)$ it holds $\mu f_{C'} \sqsubset \mu f_C$.*

Proof. By definition of improvement, we have that $f_{C'}(\mu f_C) \sqsubset \mu f_C$, i.e., μf_C is a pre-fixpoint of $f_{C'}$ and it is not a fixpoint. Hence by Knaster-Tarski $\mu f_{C'} \sqsubset \mu f_C$ follows. □

Thus, once the strategy can be improved, we will get closer to the least fixpoint of f. We next show that an improvement of the current strategy exists as long as we have not encountered a fixpoint of f.

Lemma 5.2.12 (min-improvements exist for non-fixpoints). *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a monotone function and $H_{\min}$ a min-decomposition. Given a strategy C in $H_{\min}$, the following are equivalent:*

1. $\mu f_C \notin Fix(f)$
2. $imp_{\min}(C) \neq \emptyset$
3. $f(\mu f_C) \sqsubset \mu f_C$

Proof. (1 ⇒ 2): Assume $\mu f_C \notin Fix(f)$. By Lemma 5.2.7(1) we know $f \sqsubseteq f_C$. Hence $\mu f_C = f_C(\mu f_C) \sqsupseteq f(\mu f_C)$. Since $\mu f_C \notin Fix(f)$, we deduce that the inequality is strict $\mu f_C \sqsupset f(\mu f_C)$.

By Lemma 5.2.7(2) we can take a strategy C', s.t. for all $y \in Y$

$$f(\mu f_C)(y) = C'(y)(\mu f_C) = f_{C'}(\mu f_C)(y)$$

Thus $\mu f_C \sqsupset f_{C'}(\mu f_C)$ and a min-improvement exists by definition.

1. Initialize: guess a strategy C_0 for Player Min, $i := 0$
2. **iterate**
   a) determine μf_{C_i}
   b) **if** $imp_{\min}(C_i) \neq \varnothing$, let $C_{i+1} \in imp_{\min}(C_i)$; $i := i + 1$; **goto** (a)
   c) **else if** $\mu f_{C_i} \neq \mu f$ let $a \sqsubset \mu f_{C_i}$ be a pre-fixpoint of f and determine C_{i+1} via
   $$C_{i+1}(y) = \arg \min_{h \in H_{\min}(y)} h(a)$$
   $i := i + 1$; **goto** 2.(a)
   d) **else stop**: $\mu f_{C_i} = \mu f$

Fig. 5.2.: Computing the least fixpoint μf of $f \colon \mathbb{M}^Y \to \mathbb{M}^Y$, from above

(2 ⇒ 1): Let $C' \in imp_{\min}(C)$. By definition of improvement $f_{C'}(\mu f_C) \sqsubset \mu f_C$. By Lemma 5.2.7(1) we know $f \sqsubseteq f_{C'}$ and thus $f(\mu f_C) \sqsubseteq f_{C'}(\mu f_C)$. Joining the two, we obtain $f(\mu f_C) \sqsubset \mu f_C$ and thus $\mu f_C \notin Fix(f)$, as desired.

(1 ⇔ 3): By Lemma 5.2.7(1) we know that $f(\mu f_C) \sqsubseteq f_C(\mu f_C) = \mu f_C$. Hence $\mu f_C \notin Fix(f)$, i.e., $f(\mu f_C) \neq \mu f_C$ is equivalent to $f(\mu f_C) \sqsubset \mu f_C$, as desired. □

The above result suggests an algorithm for computing a fixpoint of a function $f : \mathbb{M}^Y \to \mathbb{M}^Y$ on the basis of some min-decomposition. The idea is to guess some strategy C, determine μf_C and check $imp_{\min}(C)$. If this set is empty we have reached some fixpoint, otherwise choose $C' \in imp_{\min}(C)$ for the next iteration. Note that for this algorithm it is irrelevant whether we use min-improvements or restrict to stable min-improvements. We also note that this procedure and the developed theory to this point work for monotone functions $f \colon L^Y \to L^Y$ where L is a complete lattice. We will use this fact to apply our strategy iterations to discounted mean-payoff games as the underlying fixpoint function $\mathcal{L}$ has exactly one fixpoint (cf. Section 2.7.2).

When we are interested in the least fixpoint and the function admits many fixpoints, the sketched algorithm determines a fixpoint which might not be the desired one. Exploiting the theory from Chapter 3, we can refine the algorithm to ensure that it computes μf. For this, we have to work with non-expansive functions $f : \mathbb{M}^Y \to \mathbb{M}^Y$ with $\mathbb{M}$ being a complete MV-chain and Y a finite set. In fact, in this setting, given a fixpoint of f, say $a \in \mathbb{M}^Y$, relying on Lemma 3.4.2, we can check whether it is the least fixpoint of f. In case it is not, we can "improve" it obtaining a smaller pre-fixpoint of f in a way that we can continue the iteration from there. The resulting algorithm is reported in Figure 5.2. Observe that in step 2b we clearly do not need to compute all improvements. Rather, a min-improvement, whenever it exists, can be determined, on the basis of Definition 5.2.9, using μf_{C_i} computed in step 2a. Moreover step 2c relies on Lemma 3.4.2.

Theorem 5.2.13 (least fixpoint, from above)**.** *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function and let $H_{\min}$ be a min-decomposition of f. The algorithm in Figure 5.2 terminates and computes μf.*

Proof. We first argue that in the iteration eventually $imp_{\min}(C_i) = \emptyset$, hence we will reach step 2c, and this happens iff we have reached some fixpoint of f.

In fact, for any i, if $imp_{\min}(C_i) \neq \emptyset$, then one considers $C_{i+1} \in imp_{\min}(C_i)$ and by Lemma 5.2.11 $\mu f_{C_{i+1}} \sqsubset \mu f_{C_i}$. Thus the algorithm computes a strictly descending chain μf_{C_i}. This means that we certainly generate a new strategy at each iteration. Since there are only finitely many strategies, the iteration must stop at some point. When this happens we will have $imp_{\min}(C_i) = \emptyset$ and thus, we will reach step 2c. By Lemma 5.2.12, this happens if and only if μf_{C_i} is a fixpoint of f.

Now, by Theorem 3.4.2, we can determine whether $\mu f_{C_i} = \mu f$ and thus the algorithm only terminates when μf is computed.

Otherwise one considers $a \sqsubset \mu f_{C_i}$, a (pre-)fixpoint of f, which is given by Lemma 3.4.2.

Due to Lemma 5.2.7(2) and the way the new strategy C_{i+1} is defined we know that $f_{C_{i+1}}(a)(y) = C_{i+1}(y)(a) = f(a)(y) \sqsubseteq a(y)$ for all $y \in Y$, we have that a is a (pre-)fixpoint also of $f_{C_{i+1}}$, and so $\mu f_{C_{i+1}} \sqsubseteq a \sqsubset \mu f_{C_i}$.

Therefore, again we obtain a descending chain of least fixpoints. The number of strategies is finite, since Y is finite and $H_{\min}(y)$ is finite for all $y \in Y$. Thus we will at some point compute $\mu f_{C_j} = \mu f$ for some strategy C_j and terminate since by Lemma 3.4.2 we can determine when this is the case. □

Termination easily follows from the fact that the number of strategies is finite (since Y is finite and $H_{\min}(y)$ is finite for all $y \in Y$). Given that at any iteration the fixpoint decreases, no strategy can be considered twice, and thus the number of iterations is bounded by the number of strategies.

Example 5.2.14. *Let us revisit Example 5.2.3 and the fixpoint function $\mathcal{V}$ defined there. Its least fixpoint satisfies $\mu\mathcal{V}(\mathbf{1}) = 1$ and $\mu\mathcal{V}(v) = \varepsilon$ for any $v \in V \setminus \{\mathbf{1}\}$.*

The optimal strategy for Min *is to choose* $\boldsymbol{av}$ *as its successor since this forces* Max *to exit the cycle formed by* $\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}$ *to* $\boldsymbol{\varepsilon}$*, yielding a payoff of* ε *for these states. If* Max *would behave in a way that the play keeps cycling he would obtain a payoff of* 0*, which is suboptimal.*

We now apply our algorithm. We start by guessing a strategy for Min*, so we assume* $C_0(\boldsymbol{min}) = h_{\mathbf{1}}$*, i.e.* $C_0 = C_1^{\min}$ *(for the naming of the strategies we refer to Example 5.2.6). The least fixpoint* $\mu\mathcal{V}_{C_0}$ *can be found by solving the following linear program (details will be discussed in Section 5.3.2):*

$$\min \sum_{v \in V} a(v) \qquad a(\mathbf{1}) = 1 \qquad a(\boldsymbol{\varepsilon}) = \varepsilon \qquad a(\boldsymbol{av}) = \frac{1}{2}a(\boldsymbol{min}) + \frac{1}{2}a(\boldsymbol{max})$$
$$a(\boldsymbol{max}) \geq a(\boldsymbol{\varepsilon}) \quad a(\boldsymbol{max}) \geq a(\boldsymbol{av}) \quad a(\boldsymbol{min}) = a(\mathbf{1})$$

with $0 \leq a(v) \leq 1$ *for* $v \in V$*, which yields* $\mu\mathcal{V}_{C_0}(\boldsymbol{\varepsilon}) = \varepsilon$ *and* $\mu\mathcal{V}_{C_0}(v) = 1$ *for all* $v \in V \setminus \{\boldsymbol{\varepsilon}\}$*. Now* $\mu\mathcal{V}_{C_0}$ *is a fixpoint of* $\mathcal{V}$ *– but not the least – and thus we find the vicious cycle formed* *by* $\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}$*, i.e.* $\nu\mathcal{V}_{\#}^{\mu\mathcal{V}_{C_0}} = \{\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}\}$ *and decrease the values of those states in* a *by* δ*, i.e. we obtain* $a = \mu\mathcal{V}_{C_0} \ominus \delta_{\{\boldsymbol{min},\boldsymbol{av},\boldsymbol{max}\}}$*. This results in* $a(\mathbf{1}) = 1$*,* $a(\boldsymbol{\varepsilon}) = \varepsilon$ *and* $a(v) = 1 - \delta$ *for all* $v \in V \setminus \{\mathbf{1}, \boldsymbol{\varepsilon}\}$*. Any* $\delta \in (0, 1-\varepsilon]$ *is a valid choice.*

Computing $C_1(y) = \arg\min_{h \in H_{\min}(y)} h(a)$ *yields the strategy* $C_1 = C_2^{\min}$*, i.e.* $C_1(\boldsymbol{min}) = h_{\boldsymbol{av}}$*. By linear programming (replace* $a(\boldsymbol{min}) = a(\mathbf{1})$ *by* $a(\boldsymbol{min}) = a(\boldsymbol{av})$*) we obtain* $\nu\mathcal{V}_{\#}^{\mu f_{C_1}} = \emptyset$*, thus* $\mu\mathcal{V}_{C_1} = \mu\mathcal{V}$ *and the algorithm terminates.*

5.2.5. Strategy Iteration from Below

Here we present a different generalized strategy iteration algorithm approaching the least fixpoint from below. Intuitively, now it is player Max who improves his strategy step by step, creating an ascending chain of least fixpoints which reaches the least fixpoint of the underlying function f. Despite the fact that in this case we cannot get stuck at a fixpoint which is not the least, the correctness argument is more involved.

We will deal with max-decompositions of a function and we will need a notion of (stable) max-improvement which is naturally defined as a dualisation of the notion of (stable) min-improvement (Definition 5.2.9).

Definition 5.2.15 (max-improvement)**.** *Let* Y *be a finite set,* $\mathbb{M}$ *a complete MV-chain,* $f : \mathbb{M}^Y \to \mathbb{M}^Y$ *a monotone function and* $H_{\max}$ *a max-decomposition. Given* C, C' *strategies in* $H_{\max}$*, we say that* C' *is a* ***max-improvement*** *of* C *if* $\mu f_C \sqsubset f_{C'}(\mu f_C)$*. It is called a* ***stable*** *max-improvement if in addition* $C'(y) = C(y)$ *for all* $y \in Y$ *such that* $f_{C'}(\mu f_C)(y) = \mu f_C(y)$*. We denote by* $imp_{\max}(C)$ *(respectively* $imp^s_{\max}(C)$*) the set of (stable) max-improvements of* C*.*

Analogously to Remark 5.2.10 we can easily obtain (stable) max-improvements.

Remark 5.2.16 (obtaining max-improvements)**.** *For a strategy* C*, if* $imp_{\max}(C) \neq \emptyset$*, one can obtain a max-improvement of* C *by taking* $C' \neq C$ *defined as*

$$C'(y) = \arg\max_{h \in H_{\min}(y)} h(\mu f_C)$$

and a stable max-improvement as:

$$C'(y) = \begin{cases} C(y) & \text{if } f(\mu f_C)(y) = \mu f_C(y) \\ \arg\max_{h \in H_{\min}(y)} h(\mu f_C) & \text{otherwise} \end{cases}$$

There could be several $h \in H_{\max}(y)$ *where* $h(\mu f_C)$ *is maximal. Any such choice is valid.*

When iterating from above it was rather easy to show that given a strategy C and a min-improvement C', the latter yields a smaller least fixpoint $\mu f_{C'} \sqsubset \mu f_C$ (Lemma 5.2.11). Observing that μf_C is a pre-fixpoint of $f_{C'}$ was enough to prove this.

Here, however, we cannot simply dualise the argument. If C' is a max-improvement of C, we obtain that μf_C is a post-fixpoint of $f_{C'}$ which, in general, does not guarantee $\mu f_{C'} \sqsupset \mu f_C$. We have to resort to stable max-improvements and, in order to show that such improvements in fact yield greater least fixpoints, we need, again, to use the theory developed in Chapter 3. Hence, we have to work with non-expansive functions $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ where $\mathbb{M}$ is a complete MV-chain.

In order to prove that a stable max-improvement leads to greater least fixpoint, we require the soundness result in Lemma 3.4.2 to the case in which a is a post-fixpoint instead of a fixpoint.

Lemma 5.2.17 (max-improvements increase fixpoints)**.** *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a non-expansive function and $H_{\max}$ a max-decomposition. Given a strategy C in $H_{\max}$ and a stable max-improvement $C' \in imp^s_{\max}(C)$, then $\mu f_C \sqsubset \mu f_{C'}$.*

Proof. We show that $\nu(f_{C'})_*^{\mu f_C} = \emptyset$ and thus conclude that $\mu f_C \sqsubseteq \mu f_{C'}$ by Lemma 3.4.2. Recall that $(f_{C'})_*^{\mu f_C}\colon [Y]^{\mu f_C = f_{C'}(\mu f_C)} \to [Y]^{\mu f_C = f_{C'}(\mu f_C)}$, i.e., it restricts to those elements of μf_C where μf_C and $f_{C'}(\mu f_C)$ coincide.

We proceed by showing that $(f_C)_*^{\mu f_C}$ and $(f_{C'})_*^{\mu f_C}$ agree on $[Y]^{\mu f_C = f_{C'}(\mu f_C)}$, which, by definition, is a subset of $[Y]^{\mu f_C} = [Y]^{\mu f_C = f_C(\mu f_C)}$ (remember that μf_C is a fixpoint of f_C). It holds that

$$
\begin{aligned}
(f_C)_*^{\mu f_C}(Y') &= \gamma^{f_C(\mu f_C),\delta}(f_C(\alpha^{\mu f_C,\delta}(Y'))) \\
(f_{C'})_*^{\mu f_C}(Y') &= \gamma^{f_{C'}(\mu f_C),\delta}(f_{C'}(\alpha^{\mu f_C,\delta}(Y'))) \cap [Y]^{\mu f_C = f_{C'}(\mu f_C)}
\end{aligned}
$$

for a suitable constant δ and if we choose δ small enough we can use the same constant in both cases.

Now let $y \in [Y]^{\mu f_C = f_{C'}(\mu f_C)}$: by definition it holds that

$$
\begin{aligned}
& y \in (f_C)_*^{\mu f_C}(Y') = \gamma^{f_C(\mu f_C),\delta}(f_C(\alpha^{\mu f_C,\delta}(Y'))) \\
\iff\ & f_C(\mu f_C)(y) \ominus f_C(\alpha^{\mu f_C,\delta}(Y'))(y) \sqsupseteq \delta \\
\iff\ & C(y)(\mu f_C) \ominus C(y)(\alpha^{\mu f_C,\delta}(Y')) \sqsupseteq \delta
\end{aligned}
$$

Now, since $\mu f_C(y) = f_{C'}(\mu f_C)(y)$, by definition of stable max-improvement (remember that we require $C' \in imp^s_{\max}(C)$) we have $C(y) = C'(y)$, and thus

$$
\begin{aligned}
& C(y)(\mu f_C) \ominus C(y)(\alpha^{\mu f_C,\delta}(Y')) \sqsupseteq \delta \\
\iff\ & C'(y)(\mu f_C) \ominus C'(y)(\alpha^{\mu f_C,\delta}(Y')) \sqsupseteq \delta \\
\iff\ & y \in (f_{C'})_*^{\mu f_C}(Y')
\end{aligned}
$$

Thus $\nu(f_{C'})_*^{\mu f_C} \subseteq \nu(f_C)_*^{\mu f_C} = \emptyset$.

In conclusion, we have that $\mu f_{C'} \sqsupseteq \mu f_C$ and since μf_C is not a fixpoint of $f_{C'}$ (because C' is a max-improvement of C) we conclude $\mu f_{C'} \sqsupset \mu f_C$. □

Example 5.2.18. *We note that working with max-improvements which are stable is essential for the validity of Lemma 5.2.17 above. In fact, consider the simple stochastic*

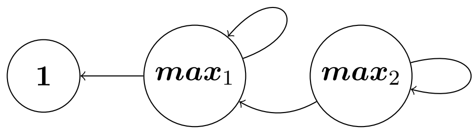


Fig. 5.3.: An example of a simple stochastic game where state **1** has payoff 1.

game in Figure 5.3 where $\boldsymbol{max}_1, \boldsymbol{max}_2 \in V_{\mathsf{Max}}$ *and* $\mathbf{1} \in V_{\mathsf{Sink}}$*, with reward* 1*. Let* C *be the strategy for* Max *where* $\boldsymbol{max}_1$ *and* $\boldsymbol{max}_2$ *have as successors* $\mathbf{1}$ *and* $\boldsymbol{max}_2$*, respectively. It is easy to see that* $\mu\mathcal{V}_C(\mathbf{1}) = \mu\mathcal{V}_C(\boldsymbol{max}_1) = 1$ *and* $\mu\mathcal{V}_C(\boldsymbol{max}_2) = 0$*. Now, an improvement in* $imp_{\max}(C)$ *can be the strategy* C' *which chooses* $\boldsymbol{max}_1$ *as a successor for both* $\boldsymbol{max}_1$ *and* $\boldsymbol{max}_2$*. Then we have* $\mu\mathcal{V}_{C'}(\boldsymbol{max}_1) = \mu\mathcal{V}_{C'}(\boldsymbol{max}_2) = 0$*, hence* $\mu\mathcal{V}_C \sqsupset \mu\mathcal{V}_{C'}$*. The reason why this happens is that* C' *is not a stable improvement of* C *since it uselessly changes the successor of* $\boldsymbol{max}_1$ *from* $\mathbf{1}$ *to* $\boldsymbol{max}_1$*, both mapped to* 1 *by* $\mu\mathcal{V}_C$*. A stable improvement of* C *is* C'' *where* $\boldsymbol{max}_1$ *and* $\boldsymbol{max}_2$ *have as successors* $\mathbf{1}$ *and* $\boldsymbol{max}_1$*, respectively. Then it can be seen that* $\mu\mathcal{V}_{C''}(v) = 1$ *for all states* $v \in V$*.*

Relying on Lemma 5.2.17, we can easily prove the dual of Lemma 5.2.12, showing that a strategy admits a stable max-improvement as long as we have not reached a fixpoint of f.

Lemma 5.2.19 (max-improvements exist for non-fixpoints)**.** *Let* Y *be a finite set,* $\mathbb{M}$ *a complete MV-chain,* $f : \mathbb{M}^Y \to \mathbb{M}^Y$ *a monotone function and* $H_{\max}$ *a max-decomposition. Given a strategy* C *in* $H_{\max}$*, the following are equivalent:*

1. $\mu f_C \notin \mathit{Fix}(f)$
2. $imp^s_{\max}(C) \neq \emptyset$
3. $\mu f_C \sqsubset f(\mu f_C)$

Proof. Adapt Lemma 5.2.12, observing that whenever there is a max-improvement, i.e. $imp_{\max}(C) \neq \emptyset$, then there exists one which is stable, i.e. $imp^s_{\max}(C) \neq \emptyset$, and, clearly, vice versa. This holds since, given a strategy $C' \in imp_{\max}(C)$, for all $y \in Y$, we can define $C''(y) = C'(y)$ when $\mu f_C(y) \sqsubset f_{C'}(\mu f_C)(y)$, and $C''(y) = C(y)$ otherwise. Then $\mu f_C \sqsubset f_{C''}(\mu f_C)$ and $C'' \in imp^s_{\max}(C)$. □

To summarise, given a strategy C with $\mu f_C \notin \mathit{Fix}(f)$ we can construct a strategy C' with $\mu f_C \sqsubset \mu f_{C'}$. This creates an ascending chain of least fixpoints and since there are only finitely many strategies we will at some point find an optimal strategy C^* with $\mu f_{C^*} = \mu f$.

1. Initialize: guess a strategy C_0 for Player Max, $i := 0$
2. **iterate**
   a) determine μf_{C_i}
   b) **if** $imp^s_{\max}(C_i) \neq \emptyset$, let $C_{i+1} \in imp^s_{\max}(C_i)$; $i := i+1$; **goto** (a)
   c) else **stop**: $\mu f_{C_i} = \mu f$

Fig. 5.4.: Computing the least fixpoint μf of $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$, from below

Proposition 5.2.20 (least fixpoint from max-decomposition)**.** *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ a non-expansive function and let $H_{\max} : Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ be a max-decomposition of f. Then*

$$\mu f = \max\{\mu f_C \mid C \textit{ is a strategy in } H_{\max}\}.$$

Proof. By Lemma 5.2.7(1) (adapted to the max case) we have $f_C \sqsubseteq f$ for any strategy C in $H_{\max}$. Therefore, $\mu f_C \sqsubseteq \mu f$ and thus

$$\mu f \sqsupseteq \max\{\mu f_C \mid C \text{ is a strategy in } H_{\max}\}.$$

Assume, by contradiction, that the inequality above is strict. Then for all strategies C in $H_{\max}$ we have $\mu f_C \sqsubset \mu f$. Then, by Lemma 5.2.19, each strategy admits a stable max-improvement. Starting from any strategy C_0 one could thus generates a sequence of stable max-improvements C_1, C_2, ... Since by Lemma 5.2.17, $\mu f_{C_i} \sqsubset \mu f_{C_{i+1}}$, all these improvements would be different, thus contradicting the finiteness of the max-decomposition and hence the fact that there are finitely many strategies. □

The above results lead us to a generalised strategy iteration algorithm which approaches the least fixpoint from below.

Theorem 5.2.21 (least fixpoint, from below)**.** *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function and let $H_{\max}$ be a max-decomposition of f. The algorithm in Figure 5.4 terminates and computes μf.*

Proof. For any i, if μf_{C_i} is not a fixpoint of f, then, by Lemma 5.2.19, $imp^s_{\max}(C_i) \neq \emptyset$. By Lemma 5.2.17 $\mu f_{C_i} \sqsubset \mu f_{C_{i+1}}$ for any $C_{i+1} \in imp^s_{\max}(C_i)$. Thus the algorithm computes a strictly ascending chain μf_{C_i}. This means that we certainly generate a new strategy at each iteration.

Since there are only finitely many strategies (as argued in the proof of Theorem 5.2.13), the iteration must stop at some point. When this happens we will have $imp^s_{\max}(C_i) = \emptyset$

and thus, by Lemma 5.2.19, μf_{C_i} is a fixpoint of f. By Proposition 5.2.20 we conclude that $\mu f_{C_i} = \mu f$. □

The iteration from below may seem more appealing since it cannot get stuck at any fixpoint of f. However, it has to be noted that the computation of μf_C – for a chosen strategy C - may be more difficult than before, which is illustrated by the following example.

Example 5.2.22. *Let us apply the algorithm from Figure 5.4 to the simple stochastic game in Example 5.2.3. Recall that the least fixpoint is given by $\mu\mathcal{V}(\mathbf{1}) = 1$ and $\mu\mathcal{V}(v) = \varepsilon$ for all $v \in V \setminus \{\mathbf{1}\}$.*

We start by guessing a strategy for Max, so we assume $C_0(\boldsymbol{max}) = h_{\boldsymbol{av}}$, i.e. $C_0 = C_2^{\boldsymbol{max}}$. With this choice of strategy, Min is able to keep the game going infinitely in the cycle formed by $\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}$ and thus payoff 0 is obtained. Now $\mu\mathcal{V}_{C_0}$ is given by $\mu\mathcal{V}_{C_0}(\boldsymbol{\varepsilon}) = \varepsilon$, $\mu\mathcal{V}_{C_0}(\mathbf{1}) = 1$ and $\mu\mathcal{V}_{C_0}(v) = 0$ for all $v \in V \setminus \{\boldsymbol{\varepsilon}, \mathbf{1}\}$.

We note that $\mu\mathcal{V}_{C_0}$ cannot immediately be computed via linear programming, but there is a way to modify the fixpoint equation to have a unique fixpoint and hence linear programming can be used again. This is done by precomputing states from which Min can force a non-terminating play and assigning payoff value 0 to them. See Section 5.3.2 for a detailed explanation.

Next, Max updates his strategy and we obtain $C_1 = C_1^{\max}$. As above we can compute $\mu\mathcal{V}_{C_1}$ – which, this time, equals $\mu\mathcal{V}$ – via linear programming.

Remark 5.2.23. *Given μf (without the corresponding strategy) an interesting question is how one can derive optimal strategies for Min or Max. Note that each presented strategy iteration algorithm only produces an optimal strategy for one player, but not for the other.*

It is rather easy to find an optimal strategy with respect to $H_{\min}$. We can simply compute $C^*(y) = \arg\min_{h \in H_{\min}(y)} h(\mu f)$ which yields some optimal strategy C^*, i.e. $\mu f_{C^*} = \mu f$. It is enough to choose some minimum, even if this is ambiguous and there are several choices, each of which produces an optimal strategy. The strategy C^* is optimal since μf is a pre-fixpoint of f_{C^*} and $\mu f = \mu f_{C^*}$ follows from Proposition 5.2.8.

On the other hand, given μf, we cannot easily obtain an optimal strategy in $H_{\max}$. We will see in Example 5.3.4 that defining $C^*(y) = \arg\max_{h \in H_{\max}(y)} h(\mu f)$ for an arbitrary h where the value is maximal does not work in general.

One way to obtain an optimal strategy in $H_{\max}$ given μf is to compute all strategies C with $f_C(\mu f) = \mu f$ and check whether $\nu(f_C)_{\#}^{\mu f} = \emptyset$. If this holds for a strategy C^* - and it has to hold for at least one strategy C with $f_C(\mu f) = \mu f$ - then it is rather imminent that C^* is an optimal strategy. In cases where one has to check many strategies this procedure might be inefficient.

5.2.6. Strategy Iterations to Compute Greatest Fixpoints

We can easily dualize the generalized strategy iteration algorithms presented in Figure 5.2 and Figure 5.4 to compute the greatest fixpoint νf of $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$. These procedures are depicted in Figure 5.5 and Figure 5.6.

1. Initialize: guess a strategy C_0 for Player Max, $i := 0$
2. **iterate**
   a) determine νf_{C_i}
   b) **if** $imp_{\max}(C_i) \neq \varnothing$, let $C_{i+1} \in imp_{\max}(C_i)$; $i := i + 1$; **goto** (a)
   c) **else if** $\nu f_{C_i} \neq \nu f$ let $a \sqsupset \nu f_{C_i}$ be a post-fixpoint of f and determine C_{i+1} via
   $$C_{i+1}(y) = \arg \max_{h \in H_{\max}(y)} h(a)$$
   $i := i + 1$; **goto** 2.(a)
   d) **else stop**: $\nu f_{C_i} = \nu f$

Fig. 5.5.: Computing the greatest fixpoint νf of $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$, from below

1. Initialize: guess a strategy C_0 for Player Min, $i := 0$
2. **iterate**
   a) determine νf_{C_i}
   b) **if** $imp^s_{\min}(C_i) \neq \varnothing$, let $C_{i+1} \in imp^s_{\min}(C_i)$; $i := i + 1$; **goto** (a)
   c) else **stop**: $\nu f_{C_i} = \nu f$

Fig. 5.6.: Computing the greatest fixpoint νf of $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$, from above

Dual min-/max-improvements are defined as one would think:

Definition 5.2.24 (min/max-improvement (dual))**.** *Let $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a monotone function, where Y is a finite set and $\mathbb{M}$ a complete MV-chain, and let $H_{\min}/H_{\max}$ be a min/max-decomposition. Given strategies C, C' in $H_{\min}/H_{\max}$, we say that C' is a (dual)* ***min/max-improvement*** *of C if $f_{C'}(\nu f_C) \sqsupset \nu f_C$ ($f_{C'}(\nu f_C) \sqsubset \nu f_C$). It is called a* ***stable*** *min/max-improvement if in addition $C'(y) = C(y)$ for all $y \in Y$ such that $f_{C'}(\nu f_C)(y) = \nu f_C(y)$. We denote by $imp_{\min}(C)/imp_{\max}(C)$ (respectively $imp^s_{\min}(C)/imp^s_{\max}(C)$) the set of (stable) min/max-improvements of C.*

Dual min/max-improvements can be obtained as in Remarks 5.2.10 and 5.2.16, i.e. for a strategy C, if $imp_{\min}(C) \neq \emptyset$, one can obtain a min-improvement of C by taking $C' \neq C$ defined as $C'(y) = \arg\min_{h \in H_{\min}(y)} h(\nu f_C)$ and a stable min-improvement as:

$$C'(y) = \begin{cases} C(y) & \text{if } f(\nu f_C)(y) = \nu f_C(y) \\ \arg\min_{h \in H_{\min}(y)} h(\nu f_C) & \text{otherwise} \end{cases}$$

And for a strategy C, if $imp_{\max}(C) \neq \emptyset$, one can obtain a max-improvement of C by taking $C' \neq C$ defined as $C'(y) = \arg\max_{h \in H_{\max}(y)} h(\nu f_C)$ and a stable min-improvement as:

$$C'(y) = \begin{cases} C(y) & \text{if } f(\nu f_C)(y) = \nu f_C(y) \\ \arg\max_{h \in H_{\max}(y)} h(\nu f_C) & \text{otherwise} \end{cases}$$

There could be several $h \in H_{\min}(y)/H_{\max}(y)$ where $h(\nu f_C)$ is minimal/maximal. Any such choice is valid.

Theorem 5.2.25. *Let Y be a finite set, $\mathbb{M}$ a complete MV-chain, $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function and let $H_{\max}/H_{\min}$ be a max/min-decomposition of f. The algorithms in Figure 5.5/Figure 5.6 terminate and compute νf.*

Proof. By dualization of the theory from the previous sections. □

Note that the algorithm in Figure 5.5 can be seen as the dual algorithm to the one in Figure 5.4 and the one in Figure 5.6 as the dual algorithm to the one in Figure 5.2. I.e. when doing strategy iteration for player Max (from below) we may encounter a fixpoint which is not the greatest fixpoint whereas when doing strategy iteration for Player Min (from above) we need to work with stable min-improvements. This also means, that whenever $f : L^Y \to L^Y$ (where L is a complete lattice) is a monotone function and has a unique fixpoint we can use strategy iteration for player Max to compute νf.

5.3. Applications

We will now discuss how our strategy iterations can be utilized to solve the applications we studied previously. For these applications we either already derived the approximations in Section 3.6 or the fixpoint is unique, i.e. no approximation is needed. Thus, we only need to derive the min-/max-decompositions of the problems at hand. Additionally, since this is problem specific, we will discuss, how - given a function f and a fixed strategy C for one player - one can compute the least/greatest fixpoint of the resulting function f_C.

In some instances it is possible to compute $\mu f_C/\nu f_C$ via strategy iteration (of the other player). Such a procedure will be called strategy-in-strategy iteration.

To ease the reading, we put min-/max-decompositions of functions into blue boxes, strategy induced functions (f_C) into green boxes and linear programs/linear systems of equations which compute least/greatest fixpoints of strategy induced functions into red boxes.

We will start by discussing the two-player-games as strategy iterations are very intuitive in these instances.

5.3.1. Strategy Iterations for Discounted Mean-Payoff Games

In this section we show how strategy iteration can be applied to compute the solution of a discounted mean-payoff game $\Gamma_M = (G, w, \lambda)$ (cf. Section 2.7.2). The function $\mathcal{L}\colon [-W, W]^V \to [-W, W]^V$ is defined for $v \in V$ and $a\colon V \to [-W, W]$ as

$$\mathcal{L}(a)(v) = \begin{cases} \max_{(v,u)\in E}(1-\lambda)\cdot w(v,u) + \lambda \cdot a(u) & v \in V_{\mathsf{Max}} \\ \min_{(v,u)\in E}(1-\lambda)\cdot w(v,u) + \lambda \cdot a(u) & v \in V_{\mathsf{Min}} \end{cases}$$

The unique fixpoint of $\mathcal{L}$ yields the solution of the underlying discounted mean-payoff game.

Strategy iterations are applicable since $[-W, W]$ is a complete lattice and V a finite set. This suffices since $\mathcal{L}$ admits exactly one fixpoint. For $\mathcal{L}$ we can give both a non-trivial min- and max-decomposition. We are also able to perform strategy-in-strategy iteration which appears promising in this instance.

Min-Decomposition of $\mathcal{L}$. We have

$$H_{\min}(v) = \begin{cases} \{h^v_u \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{L}$) where

$$h^v(a) = \mathcal{L}(a)(v) \text{ and } h^v_u(a) = (1-\lambda)\cdot w(v,u) + \lambda \cdot a(u).$$

A strategy for player Min fixes the successor of each state $v \in V_{\mathsf{Min}}$, i.e. it corresponds to a positional strategy $\tau\colon V_{\mathsf{Min}} \to V$.

Assume, we are given a strategy C_{Min} for Player Min, i.e. $C_{\mathsf{Min}}(v) \in H_{\min}(v)$ for all $v \in V$, which corresponds to a positional strategy $\tau_{C_{\mathsf{Min}}} \colon V_{\mathsf{Min}} \to V$. We obtain the function $\mathcal{L}_{C_{\mathsf{Min}}} \colon [-W, W]^V \to [-W, W]^V$, defined as

$$\mathcal{L}_{C_{\mathsf{Min}}}(a)(v) = \begin{cases} \max_{(v,u) \in E} (1-\lambda) \cdot w(v,u) + \lambda \cdot a(u) & v \in V_{\mathsf{Max}} \\ (1-\lambda) \cdot w(v, \tau_{C_{\mathsf{Min}}}(v)) + \lambda \cdot a(\tau_{C_{\mathsf{Min}}}(v)) & v \in V_{\mathsf{Min}} \end{cases}$$

for $a \in [-W, W]^V$ and $v \in V$. This function has one unique fixpoint (as $(G_\tau, w_\tau, \lambda)$ is a discounted mean payoff game itself for any positional strategy $\tau \colon V_{\mathsf{Min}} \to V$).

The following linear program computes $\mu \mathcal{L}_{C_{\mathsf{Min}}}$ directly:

$$\min \sum_{v \in V} a(v)$$
$$a(v) \geq (1-\lambda) \cdot w(v,u) + \lambda \cdot a(u) \qquad \forall v \in V_{\mathsf{Max}}, \forall (v,u) \in E$$
$$a(v) = (1-\lambda) \cdot w(v, \tau_{C_{\mathsf{Min}}}(v)) + \lambda \cdot a(\tau_{C_{\mathsf{Min}}}(v)) \qquad \forall v \in V_{\mathsf{Min}}$$

Max-Decomposition of $\mathcal{L}$ In a similar vein, we have

$$H_{\max}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{L}$) where

$$h^v(a) = \mathcal{L}(a)(v) \text{ and } h_u^v(a) = (1-\lambda) \cdot w(v,u) + \lambda \cdot a(u).$$

A strategy for player Max again fixes the successor of each state $v \in V_{\mathsf{Max}}$, i.e. it corresponds to a positional strategy $\sigma \colon V_{\mathsf{Max}} \to V$.

We are given a strategy C_{Max} for Player Max, i.e. $C_{\mathsf{Max}}(v) \in H_{\max}(v)$ for all $v \in V$, which corresponds to a positional strategy $\sigma_{C_{\mathsf{Max}}} \colon V_{\mathsf{Max}} \to V$. We obtain the function $\mathcal{L}_{C_{\mathsf{Max}}} \colon [-W, W]^V \to [-W, W]^V$, defined as

$$\mathcal{L}_{C_{\mathsf{Max}}}(a)(v) = \begin{cases} \min_{(v,u) \in E} (1-\lambda) \cdot w(v,u) + \lambda \cdot a(u) & v \in V_{\mathsf{Min}} \\ (1-\lambda) \cdot w(v, \sigma_{C_{\mathsf{Max}}}(v)) + \lambda \cdot a(\sigma_{C_{\mathsf{Max}}}(v)) & v \in V_{\mathsf{Max}} \end{cases}$$

for $a \in [-W, W]^V$ and $v \in V$. This function has one unique fixpoint (as $(G_\sigma, w_\sigma, \lambda)$ is a discounted mean payoff game itself for any positional strategy $\sigma \colon V_{\mathsf{Max}} \to V$).

The following linear program computes $\mu \mathcal{L}_{C_{\mathsf{Max}}}$ directly:

$$\max \sum_{v \in V} a(v)$$
$$a(v) \le (1-\lambda)\cdot w(v,u) + \lambda \cdot a(u) \qquad \forall v \in V_{\mathsf{Min}}, \forall (v,u) \in E$$
$$a(v) = (1-\lambda)\cdot w(v,\sigma(v)) + \lambda \cdot a(\sigma(v)) \qquad \forall v \in V_{\mathsf{Max}}$$

Strategy-in-Strategy Iteration. As one can see, $\mathcal{L}_{C_{\mathsf{Min}}}$ admits a non-trivial max-decomposition and $\mathcal{L}_{C_{\mathsf{Max}}}$ a non-trivial min-decomposition. This allows for a computation of $\mu\mathcal{L}_{C_{\mathsf{Min}}}$ and $\mu\mathcal{L}_{C_{\mathsf{Max}}}$ via strategy iteration - i.e. we can perform what we call strategy-in-strategy iteration.

Given a fixed strategy C_{Min} for Player Min, we have

$$H_{\max}^{C_{\mathsf{Min}}}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{L}_{C_{\mathsf{Min}}}$) where

$$h^v(a) = \mathcal{L}_{C_{\mathsf{Min}}}(a)(v) \text{ and } h_u^v(a) = (1-\lambda)\cdot w(v,u) + \lambda \cdot a(u)$$

and given a fixed strategy C_{Max} for Player Max, we have

$$H_{\min}^{C_{\mathsf{Max}}}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{L}_{C_{\mathsf{Max}}}$) where

$$h^v(a) = \mathcal{L}_{C_{\mathsf{Max}}}(a)(v) \text{ and } h_u^v(a) = (1-\lambda)\cdot w(v,u) + \lambda \cdot a(u).$$

Now, given a strategy C_{Min} for Player Min and a strategy C_{Max} for Player Max corresponding to positional strategies $\tau_{C_{\mathsf{Min}}}$ and $\sigma_{C_{\mathsf{Max}}}$, respectively, we obtain the functions $(\mathcal{L}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}, (\mathcal{L}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}} : [-W,W]^V \to [-W,W]^V$, defined as

$$\begin{aligned}(\mathcal{L}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}(a)(v) &= (\mathcal{L}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}(a)(v) \\ &= \begin{cases} (1-\lambda)\cdot w(v,\tau_{C_{\mathsf{Min}}}(v)) + \lambda \cdot a(\tau_{C_{\mathsf{Min}}}(v)) & v \in V_{\mathsf{Min}} \\ (1-\lambda)\cdot w(v,\sigma_{C_{\mathsf{Max}}}(v)) + \lambda \cdot a(\sigma_{C_{\mathsf{Max}}}(v)) & v \in V_{\mathsf{Max}} \end{cases}\end{aligned}$$

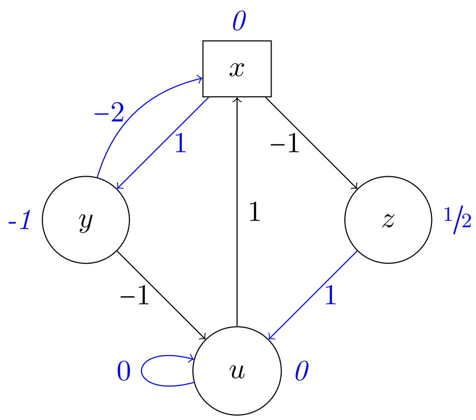
for $a \in [-W,W]^V$ and $v \in V$. Both functions have one unique fixpoint (as $(G_{\sigma,\tau}, w_{\sigma,\tau}, \lambda)$ is a discounted mean payoff game itself for any positional strategies $\sigma : V_{\mathsf{Max}} \to V$ and

$\tau\colon V_{\mathsf{Min}} \to V$). The solution to the following linear system of equations yields $\mu(\mathcal{L}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}} = \mu(\mathcal{L}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}$ directly:

$$
\begin{aligned}
a(v) &= (1-\lambda)\cdot w(v,\tau(v)) + \lambda\cdot a(\tau(v)) && \forall v \in V_{\mathsf{Min}}\\
a(v) &= (1-\lambda)\cdot w(v,\sigma(v)) + \lambda\cdot a(\sigma(v)) && \forall v \in V_{\mathsf{Max}}.
\end{aligned}
$$

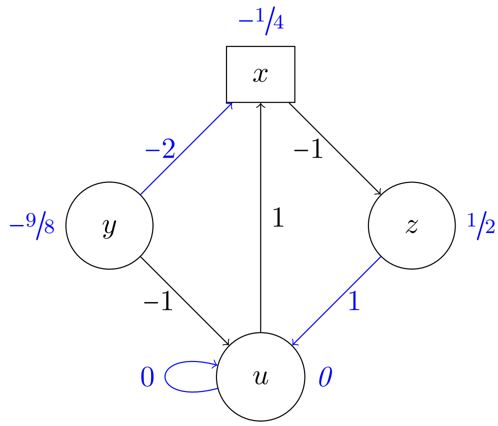
This appears promising as solving a linear system of equation is rather simple and fast compared to solving a linear program.

Example 5.3.1. *We revisit the discounted mean payoff game from Example 2.7.15 with $\lambda = 0.5$ (circles belong to Player Min, rectangles to Player Max). We have deduced $\mu\mathcal{L}$ (values and optimal strategies in blue):*



We note up front that optimal strategies for both players can directly be deducted from $\mu\mathcal{L}$, i.e. $\sigma^*(v) = \arg\max_{u\in succ(v)}(1-\lambda)\cdot w(v,u) + \lambda\cdot\mu\mathcal{L}(u)$ ($v \in V_{\mathsf{Max}}$) and $\tau^*(v) = \arg\min_{u\in succ(v)}(1-\lambda)\cdot w(v,u) + \lambda\cdot\mu\mathcal{L}(u)$ ($v \in V_{\mathsf{Min}}$).

We begin with performing strategy iteration for player Max. Let $\sigma_{C^{(0)}_{\mathsf{Max}}}(x) = z$, then via linear programming or strategy iteration for player Min we obtain $\mu\mathcal{L}_{C^{0}_{\mathsf{Max}}}$ (values and optimal answering strategies in blue):

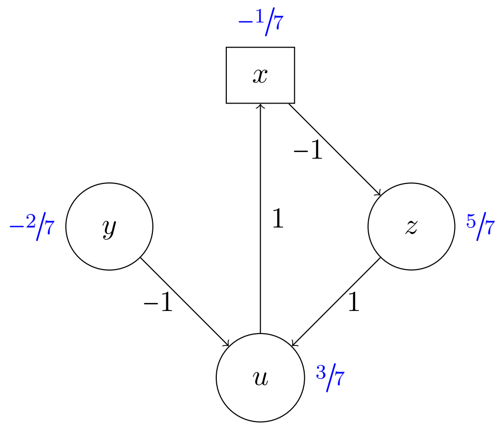


Since

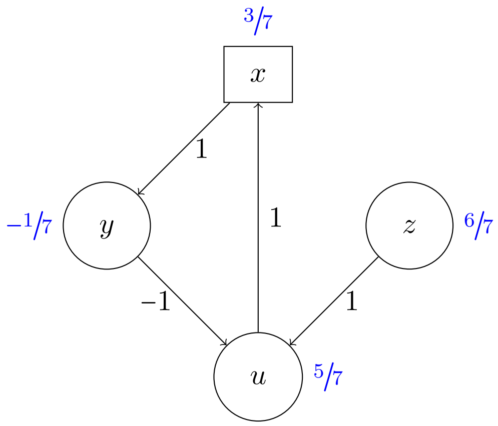
$$(1-\lambda)\cdot w(x,y)+\lambda\cdot\mu\mathcal{L}_{C^{(0)}_{\mathsf{Max}}}(y) = -1/16 > -1/4 = (1-\lambda)\cdot w(x,z)+\lambda\cdot\mu\mathcal{L}_{C^{(0)}_{\mathsf{Max}}}(z)$$

we obtain $\sigma_{C^{(1)}_{\mathsf{Max}}}(x)=y$. *Via linear programming or strategy iteration for player* Min*, we obtain* $\mu\mathcal{L}_{C^{(1)}_{\mathsf{Max}}}=\mu\mathcal{L}$.

Next, we perform strategy iteration for player Min*. Let* $\tau_{C^{(0)}_{\mathsf{Min}}}(y)=u$, $\tau_{C^{(0)}_{\mathsf{Min}}}(z)=u$, $\tau_{C^{(0)}_{\mathsf{Min}}}(u)=x$. *To compute* $\mu\mathcal{L}_{C^{(0)}_{\mathsf{Min}}}$ *we perform strategy iteration for player* Max*, i.e.* $\sigma_{C^{(0)}_{\mathsf{Max}}}(x)=z$. *The solution of the resulting linear system of equation results in* $\mu(\mathcal{L}_{C^{(0)}_{\mathsf{Min}}})_{C^{(0)}_{\mathsf{Max}}}$ *(values in blue):*



Now, Max *can improve his strategy w.r.t.* $\mu(\mathcal{L}_{C^{(0)}_{\mathsf{Min}}})_{C^{(0)}_{\mathsf{Max}}}$ *and we have* $\sigma_{C^{(1)}_{\mathsf{Max}}}(x)=y$. *We attain* $\mu(\mathcal{L}_{C^{(0)}_{\mathsf{Min}}})_{C^{(1)}_{\mathsf{Max}}}=\mu\mathcal{L}_{C^{(0)}_{\mathsf{Min}}}$ *(values in blue):*



Now it is player Min who improves her strategy. We obtain $\tau_{C^{(1)}_{Min}}(y) = x$, $\tau_{C^{(1)}_{Min}}(z) = u$ *and* $\tau_{C^{(1)}_{Min}}(u) = u$. *This is an optimal strategy and we obtain* $\mu\mathcal{L}_{C^{(1)}_{Min}} = \mu\mathcal{L}$ *(via linear programming or strategy iteration for player Max).*

Runtime Comparison. In Section 5.4.1 one can find a runtime comparison between the displayed algorithms which compute the solution of a discounted mean-payoff game.

5.3.2. Strategy Iterations for Simple Stochastic Games

In this section we show how strategy iteration can be applied to compute the solution of a simple stochastic game $\Gamma_S = (G, p, c)$ (cf. Section 2.7.3). The function $\mathcal{V}: [0,1]^V \to [0,1]^V$ was defined for $a: V \to [0,1]$ and $v \in V$ as follows:

$$\mathcal{V}(a)(v) = \begin{cases} \max_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

The least fixpoint of $\mathcal{V}$ yields the solution of the underlying simple stochastic game. $\mathcal{V}$ can admit more than one fixpoint.

Strategy iterations are applicable since V is a finite set and $[0,1]$ a complete MV-chain. Strategy-in-strategy iteration is also applicable.

For $\mathcal{V}$ we can derive both a non-trivial min- and max-decomposition.

Min-Decomposition of $\mathcal{V}$. We have

$$H_{\min}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{V}$) where

$$h^v(a) = \mathcal{V}(a)(v) \text{ and } h_u^v(a) = a(u).$$

A strategy for player Min fixes the successor of each state $v \in V_{\mathsf{Min}}$, i.e. it corresponds to a positional strategy $\tau\colon V_{\mathsf{Min}} \to V$.

Assume we are given a strategy C_{Min} for player Min, i.e. $C_{\mathsf{Min}}(v) \in H_{\min}(v)$ for all $v \in V$, which corresponds to a positional strategy $\tau_{C_{\mathsf{Min}}}\colon V_{\mathsf{Min}} \to V$. We obtain the function $\mathcal{V}_{C_{\mathsf{Min}}}\colon [0,1]^V \to [0,1]^V$, defined as

$$\mathcal{V}_{C_{\mathsf{Min}}}(a)(v) = \begin{cases} \max_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \\ a(\tau_{C_{\mathsf{Min}}}(v)) & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

for $a \in [0,1]^V$ and $v \in V$. $\mathcal{V}_{C_{\mathsf{Min}}}$ might again admit multiple fixpoints.

$\mu\mathcal{V}_{C_{\mathsf{Min}}}$ can be computed via the following linear program

$$\begin{aligned} &\min \sum_{v \in V} a(v) \\ &a(v) \geq a(u) && \forall v \in V_{\mathsf{Max}}, \forall (v,u) \in E \\ &a(v) = a(\tau_{C_{\mathsf{Min}}}(v)) && \forall v \in V_{\mathsf{Min}} \\ &a(v) = \sum_{v' \in V} p(v)(v') \cdot a(v') && \forall v \in V_{\mathsf{Av}} \\ &a(v) = c(v) && \forall v \in V_{\mathsf{Sink}} \end{aligned}$$

which yields an exact computation of $\mu\mathcal{V}_{C_{\mathsf{Min}}}$.

Max-Decomposition of $\mathcal{V}$. In a similar vein, we have

$$H_{\max}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{V}$) where

$$h^v(a) = \mathcal{V}(a)(v) \text{ and } h_u^v(a) = a(u).$$

A strategy for player Max again fixes the successor of each state $v \in V_{\mathsf{Max}}$, i.e. it corresponds to a positional strategy $\sigma\colon V_{\mathsf{Max}} \to V$.

Given a strategy C_{Max} for player Max, i.e. $C_{\mathsf{Max}}(v) \in H_{\max}(v)$ for all $v \in V$, which corresponds to a positional strategy $\sigma_{C_{\mathsf{Max}}}\colon V_{\mathsf{Max}} \to V$. We obtain the function $\mathcal{V}_{C_{\mathsf{Max}}}\colon [0,1]^V \to [0,1]^V$, defined as

$$V_{C_{\mathsf{Max}}}(a)(v) = \begin{cases} a(\sigma_{C_{\mathsf{Max}}}(v)) & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

for $a \in [0,1]^V$ and $v \in V$. This function may again have multiple fixpoints.

$\mu\mathcal{V}_{C_{\mathsf{Max}}}$ can again be computed via linear programming

$$\begin{aligned}
&\max \sum_{v \in V} a(v) && \\
&a(v) = 0 && \forall v \in Q_{\sigma_{C_{\mathsf{Max}}}} \\
&a(v) = a(\sigma_{C_{\mathsf{Max}}}) && \forall v \in V_{\mathsf{Max}} \setminus Q_{\sigma_{C_{\mathsf{Max}}}} \\
&a(v) \le a(u) && \forall v \in V_{\mathsf{Min}} \setminus Q_{\sigma_{C_{\mathsf{Max}}}}, \forall (v,u) \in E \\
&a(v) = \sum_{v' \in V} p(v)(v') \cdot a(v') && \forall v \in V_{\mathsf{Av}} \setminus Q_{\sigma_{C_{\mathsf{Max}}}} \\
&a(v) = c(v) && \forall v \in V_{\mathsf{Sink}}
\end{aligned}$$

where the set $Q_{\sigma_{C_{\mathsf{Max}}}}$ contains those nodes which will guarantee a non-terminating play if Min plays optimally, given the fixed strategy $\sigma_{C_{\mathsf{Max}}}$ (corresponding to C_{Max}) of Max. This yields an exact computation of $\mu\mathcal{L}_{C_{\mathsf{Max}}}$.

Lemma 5.3.2. *The linear program above computes $\mu\mathcal{V}_{C_{\mathsf{Max}}}$.*

Proof. See Appendix: Lemma A.4.2. □

Comparison to the Hoffmann-Karp Algorithm. The Hoffmann-Karp algorithm [TVK11] can be seen as a direct instance of our strategy iteration. Since they consider stopping simple stochastic games, $\mathcal{V}$ and $\mathcal{V}_C$ have unique fixpoints. Thus, there are no tricks required, i.e. skipping fixpoints when approaching from above or modifying the linear program (by adding Q_σ) when approaching from below.

Figure 5.7 displays both strategy iterations instantiated for simple stochastic games in standard notation. Here, given a strategy τ for player Min, we define $\mathcal{V}_\tau\colon [0,1]^V \to [0,1]^V$ as

$$\mathcal{V}_\tau(a)(v) = \begin{cases} a(\tau(v)) & \text{if } v \in V_{\mathsf{Min}} \\ \mathcal{V}(a)(v) & \text{otherwise} \end{cases}$$

and given a strategy σ for player Max, we define $\mathcal{V}_\sigma\colon [0,1]^V \to [0,1]^V$ as

$$\mathcal{V}_\sigma(a)(v) = \begin{cases} a(\sigma(v)) & \text{if } v \in V_{\mathsf{Max}} \\ \mathcal{V}(a)(v) & \text{otherwise} \end{cases}$$

for $a \in [0,1]^V$. Given $a \in [0,1]^V$ and a strategy $\tau\colon V_{\mathsf{Min}} \to V$, a node $v \in V_{\mathsf{Min}}$ is a switch node if

$$\min_{v' \in succ(v)} a(v') < a(\tau(v))$$

and $\tau' = sw_{\min}(\tau, a)$ gives a new strategy

$$\tau'(v) = \begin{cases} arg\min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \text{ is a switch node} \\ \tau(v) & \text{otherwise} \end{cases}.$$

Dually, given $a \in [0,1]^V$ and a strategy $\sigma\colon V_{\mathsf{Max}} \to V$, a node $v \in V_{\mathsf{Max}}$ is a switch node if

$$\max_{v' \in succ(v)} a(v') > a(\sigma(v))$$

and $\sigma' = sw_{\max}(\tau, a)$ gives a new strategy

$$\sigma'(v) = \begin{cases} arg\max_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Max}} \text{ is a switch node} \\ \sigma(v) & \text{otherwise} \end{cases}.$$

Strategy-in-Strategy Iteration. As for discounted mean-payoff games we can compute $\mu\mathcal{V}_{C_{\mathsf{Min}}}$ and $\mathcal{V}_{C_{\mathsf{Min}}}$ via strategy iteration - i.e. we can perform strategy-in-strategy iteration.

Given a fixed strategy C_{Min} for player Min, we have

Determine $\mu\mathcal{V}$ (from above)

1. Guess a Min-strategy $\tau^{(0)}$, $i := 0$
2. $a^{(i)} := \mu\mathcal{V}_{\tau^{(i)}}$
3. $\tau^{(i+1)} := sw_{\min}(\tau^{(i)}, a^{(i)})$
4. If $\tau^{(i+1)} \neq \tau^{(i)}$ then $i := i+1$ and goto 2
5. Compute $V' = \nu\mathcal{V}^a_{\#}$, where $a = a^{(i)}$.
6. If $V' = \varnothing$ then stop and return $a^{(i)}$. Otherwise $a^{(i+1)} := a - (\iota^a_{\mathcal{V}}(V'))_{V'}$, $\tau^{(i+2)} := sw_{\min}(\tau^{(i)}, a^{(i+1)})$, $i := i+2$, goto 2

(a) Strategy iteration from above

Determine $\mu\mathcal{V}$ (from below)

1. Guess a Max-strategy $\sigma^{(0)}$, $i := 0$
2. $a^{(i)} := \mu\mathcal{V}_{\sigma^{(i)}}$
3. $\sigma^{(i+1)} := sw_{\max}(\sigma^{(i)}, a^{(i)})$
4. If $\sigma^{(i+1)} \neq \sigma^{(i)}$ then $i := i+1$ and goto 2 Otherwise stop and return $a^{(i)}$.

(b) Strategy iteration from below

Fig. 5.7.: Strategy iteration for simple stochastic games from above and below

$$H^{C_{\mathsf{Min}}}_{\max}(v) = \begin{cases} \{h^v_u \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{V}_{C_{\mathsf{Min}}}$) where

$$h^v(a) = \mathcal{V}_{C_{\mathsf{Min}}}(a)(v) \text{ and } h^v_u(a) = a(u)$$

and given a fixed strategy C_{Max} for player Max, we have

$$H^{C_{\mathsf{Max}}}_{\min}(v) = \begin{cases} \{h^v_u \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{V}_{C_{\mathsf{Max}}}$) where

$$h^v(a) = \mathcal{V}_{C_{\mathsf{Max}}}(a)(v) \text{ and } h^v_u(a) = a(u).$$

Now, given a strategy C_{Min} for player Min and a strategy C_{Max} for player Max corresponding to positional strategies $\tau_{C_{\mathsf{Min}}}$ and $\sigma_{C_{\mathsf{Max}}}$, respectively. Then we obtain the functions $(\mathcal{V}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}, (\mathcal{V}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}} \colon [0,1]^V \to [0,1]^V$, defined as

$$(\mathcal{V}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}(a)(v) = (\mathcal{V}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}(a)(v)$$
$$= \begin{cases} a(\sigma_{C_{\mathsf{Max}}}(v)) & \text{if } v \in V_{\mathsf{Max}} \\ a(\tau_{C_{\mathsf{Min}}}(v)) & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}$$

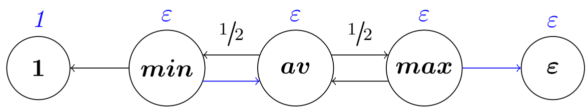
for $a \in [0,1]^V$ and $v \in V$. Both functions may admit multiple fixpoints.

The solution to the following linear system of equations directly yields $\mu(\mathcal{V}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}} = \mu(\mathcal{V}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}$:

$$\begin{aligned} a(v) &= 0 && v \in Q^{\tau_{C_{\mathsf{Min}}}}_{\sigma_{C_{\mathsf{Max}}}} \\ a(v) &= a(\tau_{C_{\mathsf{Min}}}) && v \in V_{\mathsf{Min}}, v \notin Q^{\tau_{C_{\mathsf{Min}}}}_{\sigma_{C_{\mathsf{Max}}}} \\ a(v) &= a(\sigma_{C_{\mathsf{Max}}}(v)) && v \in V_{\mathsf{Max}}, v \notin Q^{\tau_{C_{\mathsf{Min}}}}_{\sigma_{C_{\mathsf{Max}}}} \\ a(v) &= \sum_{v' \in V} a(v') \cdot p(v)(v') && v \in V_{\mathsf{Av}}, v \notin Q^{\tau_{C_{\mathsf{Min}}}}_{\sigma_{C_{\mathsf{Max}}}} \\ a(v) &= c(v) && v \in V_{\mathsf{Sink}} \end{aligned}$$

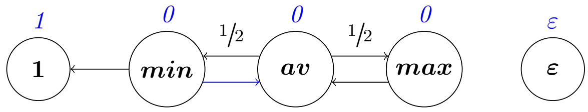
Here, $Q^{\tau_{C_{\mathsf{Min}}}}_{\sigma_{C_{\mathsf{Max}}}}$ is the set of states that reach any sink state with probability 0 in $G_{\sigma_{C_{\mathsf{Max}}},\tau_{C_{\mathsf{Min}}}}$ which very easy to compute. This appears promising as solving a linear system of equation is rather simple and fast compared to solving a linear program, see the runtime comparison in Section 5.4.2.

Example 5.3.3. *We revisit Example 2.7.22 ($\mu\mathcal{V}$ and optimal strategies in blue):*



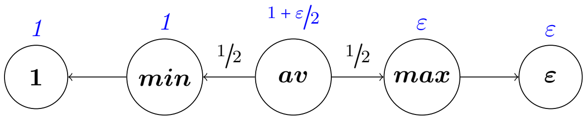
We note that an optimal strategies for player Min can directly be deduced from $\mu\mathcal{V}$, i.e. $\tau^*(v) = \arg\min_{u \in succ(v)} \mu\mathcal{V}(u)$. This is not the case for player Max as both successors of **max** *have the same value w.r.t. $\mu\mathcal{V}$ but only one successor (ε) is optimal.*

We start with strategy iteration for player Max. Let $\sigma_{C^{(0)}_{Max}}(\boldsymbol{max}) = \boldsymbol{av}$, then via linear programming, we attain $\mu\mathcal{V}_{C^{(0)}_{Max}}$ (values and optimal answering strategy in blue):

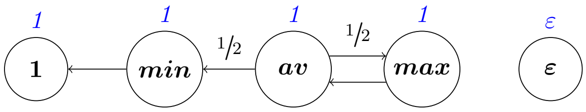


Note that $Q_{\sigma_{C^{(0)}_{\mathsf{Max}}}} = \{\boldsymbol{min}, \boldsymbol{av}, \mathbf{max}\}$ *is the set of states Min can force to never reach any sink. Next, Max switches his strategy since* $\mu\mathcal{V}_{C^{(0)}_{\mathsf{Max}}}(\boldsymbol{av}) > \mu\mathcal{V}_{C^{(0)}_{\mathsf{Max}}}(\boldsymbol{\varepsilon})$ *and for* $\sigma_{C^{(1)}_{\mathsf{Max}}}(\boldsymbol{max}) = \boldsymbol{\varepsilon}$ *we attain* $\mu\mathcal{V}_{C^{(1)}_{\mathsf{Max}}} = \mu\mathcal{V}$ *(e.g. via linear programming).*

Next, we perform strategy iteration for player Min. Let $\tau_{C^{(0)}_{\mathsf{Min}}}(\boldsymbol{min}) = \mathbf{1}$*. We compute* $\mu\mathcal{V}_{C^{(0)}_{\mathsf{Min}}}$ *via strategy iteration for player Max. Max guesses* $\sigma_{C^{(0)}_{\mathsf{Max}}}(\boldsymbol{max}) = \boldsymbol{\varepsilon}$ *which results in* $\mu(\mathcal{V}_{C^{(0)}_{\mathsf{Min}}})_{C^{(0)}_{\mathsf{Max}}}$ *(values in blue):*



$\mu(\mathcal{V}_{C^{(0)}_{\mathsf{Min}}})_{C^{(0)}_{\mathsf{Max}}}$ *can be computed by solving a linear system of equations* ($Q^{\tau_{C^{(0)}_{\mathsf{Min}}}}_{\sigma_{C^{(0)}_{\mathsf{Max}}}} = \varnothing$)*. Max now switches his strategy, i.e.* $\sigma_{C^{(1)}_{\mathsf{Max}}}(\boldsymbol{max}) = \boldsymbol{av}$*. This is rather interesting as moving to state* $\boldsymbol{\varepsilon}$ *is an optimal strategy for player Max but moving to state* $\boldsymbol{av}$ *is an optimal answering strategy to this fixed strategy for player Min. We attain* $\mu(\mathcal{V}_{C^{(0)}_{\mathsf{Min}}})_{C^{(1)}_{\mathsf{Max}}} = \mu\mathcal{V}_{C^{(0)}_{\mathsf{Min}}}$ *(values in blue):*



$\mu\mathcal{V}_{C^{(0)}_{\mathsf{Min}}}$ *is a fixpoint of* $\mu\mathcal{V}$ *but our approximation detects the vicious cycles formed by states* $\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}$*, i.e.* $\nu\mathcal{V}_{\#}^{\mu\mathcal{V}_{C^{(0)}_{\mathsf{Min}}}} = \{\boldsymbol{min}, \boldsymbol{av}, \boldsymbol{max}\}$*. We reduce these values which leads Min to switch his strategy, i.e.* $\tau_{C^{(1)}_{\mathsf{Min}}}(\boldsymbol{min}) = \boldsymbol{av}$*. Now,* $\mu\mathcal{V}_{C^{(0)}_{\mathsf{Min}}} = \mu\mathcal{V}$ *which can be computed via linear programming or strategy iteration for player Max.*

Runtime Comparison. In Section 5.4.2 one can find a runtime comparison between the displayed algorithms which compute the solution of a simple stochastic game.

5.3.3. Strategy Iterations for Energy Games

In this section we show how strategy iteration can be applied to compute the solution of an Energy Game $\Gamma_E = (G, w)$ (cf. Section 2.7.4). We can only apply our strategy iterations to energy games with finite values. Note that any energy game can easily be transformed into an energy game with finite values (see Section 2.7.4). The function $\mathcal{E}: K^V \to K^V$ is defined for $v \in V$ and $a: V \to K$ as follows[2]

$$\mathcal{E}(a)(v) = \begin{cases} \min\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v, v') & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v, v') & \text{if } v \in V_{\mathsf{Max}} \end{cases}.$$

[2] $\ominus_{\mathbb{Z}}: K \times \mathbb{Z} \to K,\ x \ominus_{\mathbb{Z}} y = \min\{\max\{x - y, 0\}, k\}$

The least fixpoint of $\mathcal{E}$ yields the solution of the underlying energy game. Note that $\mathcal{E}$ may have multiple fixpoints.

Strategy iterations are applicable since V is a a finite set and $K = \{0, \dots k\}$ a complete MV-chain. Stategy-in-strategy iteration is also applicable.

For $\mathcal{E}$ we can give a non-trivial min- and max-decomposition.

Min-Decomposition of $\mathcal{E}$. We have

$$H_{\min}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{E}$) where

$$h^v(a) = \mathcal{E}(a)(v) \text{ and } h_u^v(a) = a(u) \ominus_{\mathbb{Z}} w(v,u).$$

A strategy for player Min fixes the successor of each state $v \in V_{\mathsf{Min}}$, i.e. it corresponds to a positional strategy $\tau \colon V_{\mathsf{Min}} \to V$.

Given a strategy C_{Min} for player Min, i.e. $C_{\mathsf{Min}}(v) \in H_{\min}(v)$ for all $v \in V$, which corresponds to a positional strategy $\tau_{C_{\mathsf{Min}}} \colon V_{\mathsf{Min}} \to V$. We obtain the function $\mathcal{E}_{C_{\mathsf{Min}}} \colon K^V \to K^V$ defined as

$$\mathcal{E}_{C_{\mathsf{Min}}}(a)(v) = \begin{cases} a(\tau_{C_{\mathsf{Min}}}(v)) \ominus_{\mathbb{Z}} w(v, \tau_{C_{\mathsf{Min}}}(v)) & \text{if } v \in V_{\mathsf{Min}} \\ \max\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v,v') & \text{if } v \in V_{\mathsf{Max}} \end{cases}$$

for $a \in K^V$ and $v \in V$. This function may admit multiple fixpoints and can **under certain circumstances** be computed via the following linear program

$$\begin{aligned} & \min \sum_{v \in V} a(v) \\ & a(v) \geq a(u) - w(v,u) && \forall v \in V_{\mathsf{Max}}, \forall (v,u) \in E \\ & a(v) \geq a(\tau_{C_{\mathsf{Min}}}(v)) - w(v, \tau_{C_{\mathsf{Min}}}(v)) && \forall v \in V_{\mathsf{Min}} \\ & a(v) \geq 0 && \forall v \in V \end{aligned}$$

which yields an exact computation of $\mu\mathcal{E}_C$. We note that the above linear program indeed yields an exact computation as long as the strategy by player Min does not create a negative cycle[3] which is possible in general. One can start the iteration by setting

[3] i.e. there exists no negative cycle in G_τ

$\tau_{C^{(0)}_{\mathsf{Min}}}(v) = s$ for all states $v \in V_{\mathsf{Min}}$, i.e. we always choose the emergency exit (we can always add an emergency exit similar to what is done when reducing to an energy game with finite values in Section 2.7.4). When improving the strategy now, no negative cycle can be formed as $\mu\mathcal{E}_{C^{(0)}} \sqsupset \mu\mathcal{E}_{C^{(1)}}$.

However, other iterations like Kleene iteration which also obtain exact results seem more appealing when it comes to computing $\mu\mathcal{E}_C$. Additionally, always choosing the emergency exit does not appeal as a good starting strategy in general.

Max-Decomposition of $\mathcal{E}$. In a similar vein, we have

$$H_{\max}(v) = \begin{cases} \{h^v_u \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{E}$) where

$$h^v(a) = \mathcal{E}(a)(v) \text{ and } h^v_u(a) = a(u) \ominus_{\mathbb{Z}} w(v,u).$$

A strategy for player Max again fixes the successor of each state $v \in V_{\mathsf{Max}}$, i.e. it corresponds to a positional strategy $\sigma : V_{\mathsf{Max}} \to V$.

Given a strategy C_{Max} for player Max, i.e. $C_{\mathsf{Max}}(v) \in H_{\max}(v)$ for all $v \in V$, which corresponds to a positional strategy $\sigma_{C_{\mathsf{Max}}} : V_{\mathsf{Max}} \to V$. We obtain the function $\mathcal{E}_{C_{\mathsf{Max}}} : K^V \to K^V$ defined as

$$\mathcal{E}_{C_{\mathsf{Max}}}(a)(v) = \begin{cases} a(\sigma_{C_{\mathsf{Max}}}(v)) \ominus_{\mathbb{Z}} w(v, \sigma_{C_{\mathsf{Max}}}(v)) & \text{if } v \in V_{\mathsf{Max}} \\ \min\limits_{v' \in succ(v)} a(v') \ominus_{\mathbb{Z}} w(v,v') & \text{if } v \in V_{\mathsf{Min}} \end{cases}$$

for $a \in K^V$ and $v \in V$. This function may have multiple fixpoints. Here, linear programming is not directly applicable and we spare the work to formulate a linear program (which may not even be feasible) since other algorithms seem more appealing when it comes to computing $\mu\mathcal{E}_{C_{\mathsf{Max}}}$.

Strategy-in-Strategy Iteration. We can compute $\mu\mathcal{E}_{C_{\mathsf{Min}}}$ and $\mathcal{E}_{C_{\mathsf{Min}}}$ via strategy iteration - i.e. we can perform strategy-in-strategy iteration.

Given a fixed strategy C_{Min} for player Min, we have

$$H_{\max}^{C_{\mathsf{Min}}}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Max}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{E}_{C_{\mathsf{Min}}}$) where

$$h^v(a) = \mathcal{E}_{C_{\mathsf{Min}}}(a)(v) \text{ and } h_u^v(a) = a(u) \ominus_{\mathbb{Z}} w(v,u)$$

and given a fixed strategy C_{Max} for player Max, we have

$$H_{\min}^{C_{\mathsf{Max}}}(v) = \begin{cases} \{h_u^v \mid (v,u) \in E\} & \text{if } v \in V_{\mathsf{Min}} \\ \{h^v\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{E}_{C_{\mathsf{Max}}}$) where

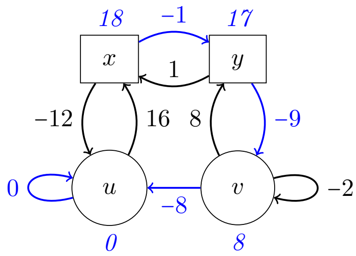
$$h^v(a) = \mathcal{E}_{C_{\mathsf{Max}}}(a)(v) \text{ and } h_u^v(a) = a(u) \ominus_{\mathbb{Z}} w(v,u).$$

Now, given a strategy C_{Min} for player Min and a strategy C_{Max} for player Max corresponding to positional strategies $\tau_{C_{\mathsf{Min}}}$ and $\sigma_{C_{\mathsf{Max}}}$, respectively. Then we obtain the functions $(\mathcal{E}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}, (\mathcal{E}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}} : K^V \to K^V$, defined as

$$\begin{aligned}(\mathcal{E}_{C_{\mathsf{Min}}})_{C_{\mathsf{Max}}}(a)(v) &= (\mathcal{E}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}(a)(v) \\ &= \begin{cases} a(\tau_{C_{\mathsf{Min}}}(v)) \ominus_{\mathbb{Z}} w(v, \tau_{C_{\mathsf{Min}}}(v)) & \text{if } v \in V_{\mathsf{Min}} \\ a(\sigma_{C_{\mathsf{Max}}}(v)) \ominus_{\mathbb{Z}} w(v, \sigma_{C_{\mathsf{Max}}}(v)) & \text{if } v \in V_{\mathsf{Max}} \end{cases}\end{aligned}$$

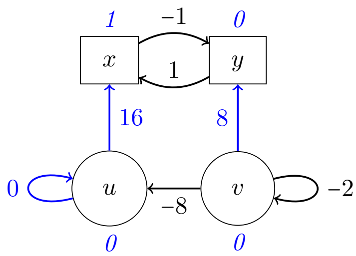
for $a \in K^V$ and $v \in V$. Both functions may admit multiple fixpoints. The least fixpoints can be computed via the usual methods for energy games. Since every state in $G_{\sigma,\tau}$ has exactly one successor one might think of checking the end components of $G_{\sigma,\tau}$ (i.e. cycles in the graph) and determine the sum of values of these end components. From this it is easy to derive the value of each state. This did not prove practical in our runtime analysis (Section 5.4.3).

Example 5.3.4. *We revisit the energy game from Example 5.3.4, where it is intended that circular and rectangular states belong to player Min and player Max, respectively. Values of $g = \mu\mathcal{E}$ and optimal strategies are given in blue:*

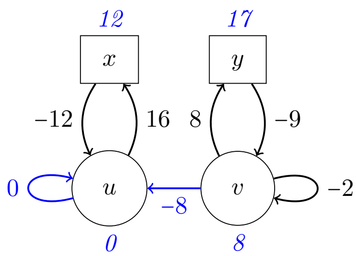


We note that optimal strategies for player Min can directly be deducted from g, i.e. $\tau^*(v) = \arg\min_{u \in succ(v)} g(u) \ominus_{\mathbb{Z}} w(v,u)$. This is not the case for player Max (in general). Given only the least fixpoint g, the strategy of player Max is not deducible with a local reasoning, since from y the choices x, v are indistinguishable (in fact $g(y) = 17 = g(x) - 1 = g(v) - (-9)$). However, if x is chosen as successor to y (and still y as successor to x), we end up in a value vector where Min needs 0 initial energy in state y to keep going forever.

We perform strategy iteration for player Max, i.e. $\sigma_{C^{(0)}_{Max}}(x) = y$ and $\sigma_{C^{(0)}_{Max}}(y) = x$. As hinted at before, we obtain $\mu\mathcal{E}_{C^{(0)}_{Max}}$ (values and optimal answering strategies in blue; state u has two optimal successors):



Now, player Max switches his strategy to $\sigma_{C^{(1)}_{Max}}(x) = u$ and $\sigma_{C^{(1)}_{Max}}(y) = v$, resulting in $\mu\mathcal{E}_{C^{(1)}_{Max}}$ (values and optimal answering strategies in blue):



Again, player Max switches his strategy to $\sigma_{C^{(2)}_{Max}}(x) = y$ and $\sigma_{C^{(1)}_{Max}}(y) = v$, resulting in $\mu\mathcal{E}_{C^{(2)}_{Max}} = g$.

KASI Algorithm. Another interesting setting of application is the lower-weak-upper-bound problem in mean-payoff games [BFL+08], reminiscent of energy games. For this problem, differently from the usual definition, the aim for one player is to maximise, never going negative, some resource which cannot exceed a given bound, while the other player has to minimise it. Also in this case, the solution can be computed as a least fixpoint. Due to the upper bound imposed to the resource, the function is not non-expansive, thus it is not captured by our theory. Still, the algorithm KASI proposed in [BC10], which computes the solution via strategy iteration, shares many similarities with our approach from below: at each iteration the algorithm computes a stable max-improvement of the current strategy. Indeed, when applying KASI to the special case where there is no upper bound to the accumulated resource, called lower-bound problem in [BFL+08] (also studied under different names in [CdAHS03, LP06]), the algorithm comes out as an exact instantiation of our general strategy iteration from below.

Runtime Comparison. In Section 5.4.3 one can find a runtime comparison between the displayed algorithms which compute the solution of an energy game.

5.3.4. Strategy Iterations for Behavioural Distances of Labeled Markov Chains

We consider the function $\Delta\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ whose least fixpoint coincides with the behavioural distance in a labeled Markov chain. Δ can be written as

$$\Delta(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{K}(\eta(s),\eta(t)) & \text{otherwise} \end{cases}$$
$$= \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \min_{c\in\Omega_V(\eta(s),\eta(t))} \sum_{(u,v)\in S\times S} c(u,v)\cdot d(u,v) & \text{otherwise} \end{cases}$$

for $d\colon S\times S \to [0,1]$ and $s,t \in S$ (discrete metric on L), cf. Section 2.6.3.

Strategy iterations are applicable since S is a finite set and $[0,1]$ a complete MV-chain.

Min-Decomposition of Δ. For Δ we can give a non-trivial min-decomposition. We have

$$H_{\min}(s,t) = \begin{cases} h_1 & \text{if } \ell(s) \neq \ell(t) \\ \{h^{s,t}_{c_{s,t}} \mid c_{s,t} \in \Omega_V(\eta(s),\eta(t))\} & \text{otherwise} \end{cases}$$

(a min-decomposition of Δ) where

$$h_1(d) = 1 \text{ and } h^{s,t}_{c_{s,t}}(d) = \sum_{(u,v)\in S\times S} c_{s,t}(u,v)\cdot d(u,v)$$

A strategy C_{Min} fixes a vertex of $c_{s,t}^{C_{\mathsf{Min}}} \in \Omega_V(\eta(s),\eta(t))$ for each pair $s,t \in S$ with the same label.

Given a fixed strategy C_{Min} for player Min, we attain the function $\Delta_{C_{\mathsf{Min}}}\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$, defined as

$$\Delta_{C_{\mathsf{Min}}}(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \sum_{(u,v)\in S\times S} c_{s,t}^{C_{\mathsf{Min}}}(u,v)\cdot d(u,v) & \text{otherwise} \end{cases}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$.

$\mu\Delta_{C_{\mathsf{Min}}}$ can be computed via the following linear program

$$\begin{aligned} &\min \sum_{s,t\in S} d(s,t) \\ &d(s,t) = 1 && \forall s,t\in S,\ \ell(s)\neq\ell(t) \\ &d(s,t) = \sum_{(u,v)\in S\times S} c_{s,t}(u,v)\cdot d(u,v) && \forall s,t\in S,\ \ell(s)=\ell(t) \\ &d(s,t) \geq 0 && \forall s,t\in S \end{aligned}$$

or the following linear system of equations

$$\begin{aligned} &d(s,t) = 1 && \forall s,t\in S,\ \ell(s)\neq\ell(t) \\ &d(s,t) = 0 && \forall s,t\in S,\ s\sim t \\ &d(s,t) = \sum_{(u,v)\in S\times S} c_{s,t}(u,v)\cdot d(u,v) && \forall s,t\in S,\ \ell(s)=\ell(t),\ s\not\sim t \end{aligned}$$

Comparison to the Policy Iterations in [TvB17]. We refer to [TvB17] on details regarding the above strategy iteration since their policy iteration (first described in [BBL+21]) is in fact an instance of our strategy iteration. They analyze the function Λ where the distance of bisimilar states is set to 0 (which makes the fixpoint unique).

They also present a partial policy algorithm to compute $\mu\Lambda$ which can be seen as an instance of a strategy-in-strategy iteration. Λ has the trivial max-decomposition

$$H_{\max}(s,t) = \{h_0, h^{s,t}\}$$

(a (trivial) max-decomposition of Λ) where

$$h_0(a) = 0 \text{ and } h^{s,t}(a) = \Lambda(a)(s,t).$$

Now given a strategy C_{Max} for player Max, i.e. $C_{\mathsf{Max}}(s,t) \in H_{\max}(s,t)$, they perform strategy iteration for player Min for the resulting function $\Lambda_{C_{\mathsf{Max}}}$ to compute $\mu\Lambda_{C_{\mathsf{Max}}}$. The min-decomposition of $\Lambda_{C_{\mathsf{Max}}}$ is analogous to the one described before (for Δ) where some states fix the strategy h_0 ($h_0(a) = 0$). They do this so they do not have to consider the whole state space but only a subset of states. They also improve the strategy for player Max only in some states which makes sense for their purposes (as they want to build up the used state space as needed)[4]. Still, in each outer iteration, when improving the strategy for player Max they inch closer to $\mu\Lambda$. The paper [TvB17] also gives a short runtime comparison.

This direction might be fruitful for new research as our theory is able to capture this partial policy algorithm.

5.3.5. Strategy Iterations for Behavioural Distances of Metric Transition Systems

We aim to analyze the function $\mathcal{J}\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ which was defined as

$$\begin{aligned}\mathcal{J}(d)(s,t) &= \max\{d_L(\ell(s),\ell(t)), \mathcal{H}(d)(\eta(s),\eta(t))\}\\ &= \max\{|\ell(s)-\ell(t)|, \max_{u\in\eta(s)}\min_{v\in\eta(t)} d(u,v), \max_{v\in\eta(t)}\min_{u\in\eta(s)} d(v,u)\}\end{aligned}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$ ($L = [0,1]$, d_L is the Euclidian metric), cf. Section 2.6.4. $\mathcal{J}$ usually has multiple fixpoints.

Strategy iterations are applicable since S is a finite set and $[0,1]$ a complete MV-chain.

Max-Decomposition for $\mathcal{J}$. $\mathcal{J}$ has a non-trivial max-decomposition. We have

$$H_{\max}(s,t) = \{h_L^{s,t}\} \cup \{h_u^{s,t} \mid u \in \eta(s)\} \cup \{h_v^{s,t} \mid v \in \eta(t)\}$$

(a max-decomposition of $\mathcal{J}$) where

$$h_L^{s,t}(d) = |\ell(s)-\ell(t)|,\ h_{u_s}^{s,t}(d) = \min_{v\in\eta(t)} d(u,v) \text{ and } h_{u_t}^{s,t}(d) = \min_{v\in\eta(s)} d(v,u).$$

A strategy for player Max fixes either the Euclidian distance between the labels or an element in $\eta(s)$ or $\eta(t)$ for all states $s,t \in S$.

[4]This still comes out to be a stable max-improvement.

Given a fixed strategy C for player Max, i.e. $C(s,t) \in H_{\max}(s,t)$ for all $s,t \in S$, we obtain the function $\mathcal{J}_C\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ defined as

$$\mathcal{J}_C(d)(s,t) = \begin{cases} |\ell(s) - \ell(t)| & \text{if } C(s,t) = h_L^{s,t} \\ \min_{v\in\eta(t)} d(u,v) & \text{if } C(s,t) = h_{u_s}^{s,t} \\ \min_{v\in\eta(s)} d(v,u) & \text{if } C(s,t) = h_{u_t}^{s,t} \end{cases}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$. We can compute $\mathcal{J}_C$ via Kleene iteration (from below). $\mathcal{J}_C$ also admits a non-trivial min-decomposition which we will not detail here.

There exists another (rather trivial) max-decomposition of $\mathcal{J}$. We have

$$H_{\max}(s,t) = \{h_L^{s,t}\} \cup \{h_{\mathcal{H}}^{s,t}\}$$

(a second max-decomposition of $\mathcal{J}$) where

$$h_L^{s,t}(d) = |\ell(s) - \ell(t)|,\ h_{\mathcal{H}}^{s,t} = \mathcal{H}(d)(\eta(s),\eta(t).$$

Here, a strategy for player Max fixes either the Euclidian distance between the labels or the Hausdorff distance of the successors.

Given a fixed strategy C for player Max, i.e. $C(s,t) \in H_{\max}(s,t)$ for all $s,t \in S$, we obtain the function $\mathcal{J}_C\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ defined as

$$\mathcal{J}_C(d)(s,t) = \begin{cases} |\ell(s) - \ell(t)| & C(s,t) = h_L^{s,t} \\ \min\limits_{W}\{ \max\limits_{(x,y)\in W} d(x,y) \mid W \subseteq S\times S \wedge u(W) = (\eta(s),\eta(t))\} & C(s,t) = h_{\mathcal{H}}^{s,t} \end{cases}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$ (cf. Section 3.6.4 for the dual characterization of $\mathcal{H}$, renaming couplings to W to avoid confusion). We can derive a min-decomposition for the function $\mathcal{J}_C$ (almost) dual to the max-decomposition of the function $\mathcal{B}$ we will analyze in the next section (fixing the strategy $h_L^{s,t}$ for states with $C(s,t) = h_L^{s,t}$). We can even go further: Fixing a strategy in this min-decomposition results in a function for which we can also derive a min-decomposition (almost) dual to the max-decomposition we will derive in the next section (strategy-in-strategy-in-strategy). Also, complete distributivity can be used to obtain another function which omits a max-decomposition (almost dual to what we are doing in the next section).

5.3.6. Strategy Iterations for Bisimilarity of Transition Systems

We consider the function $\mathcal{B}\colon \{0,1\}^{S\times S} \to \{0,1\}^{S\times S}$ whose greatest fixpoint coincides with the bisimilarity of a transition system. $\mathcal{B}$ can be written as

$$\mathcal{B}(a)(s,t) = \max_{W}\{ \min_{(s',t')\in W} a(s',t') \mid W \subseteq S\times S \ \wedge\ u(W) = (\eta(s),\eta(t))\}$$

for $a \in \{0,1\}^{S\times S}$ and $s,t \in S$ (cf. Section 3.6.7). We write W for C to avoid confusion.

Strategy iterations are applicable since S is a a finite set and $\{0,1\}$ a complete MV-chain. Stategy-in-strategy iteration is also applicable.

This instance is somewhat interesting as it is the only one where we aim to compute a greatest fixpoint.

We note up front that we can derive a min-decomposition (almost) dual to the max-decomposition described for $\mathcal{J}$ in the previous section when taking the primal representation of $\mathcal{B}$, see Definition 2.6.5.

Max-Decomposition of $\mathcal{B}$. For $\mathcal{B}$ we can give a non-trivial max-decomposition. We have

$$H_{\max}(s,t) = \begin{cases} \{h^{s,t}_{W_{s,t}} \mid W_{s,t} \subseteq S\times S \ \wedge\ u(W_{s,t}) = (\eta(s),\eta(t))\} & \eta(s) = \emptyset \Leftrightarrow \eta(t) = \emptyset \\ \{h_0\} & \text{otherwise} \end{cases}$$

(a max-decomposition of $\mathcal{B}$) where

$$h^{s,t}_{W_{s,t}}(a) = \min_{(s',t')\in W} a(s',t') \text{ and } h_0(a) = 0.$$

A strategy for player Max fixes a coupling $W_{s,t}$ of $\eta(s),\eta(t)$ for each pair $s,t \in S$ (as long as one exists).

Given a fixed strategy C_{Max} for player Max we attain the function $\mathcal{B}_{C_{\mathsf{Max}}}\colon \{0,1\}^S \to \{0,1\}^S$, defined as

$$\mathcal{B}_{C_{\mathsf{Max}}}(a)(s,t) = \begin{cases} \min_{(s',t')\in W_{s,t}} a(s',t') & \text{if } \eta(s) = \emptyset \Leftrightarrow \eta(t) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

for $a \in \{0,1\}^S$ and $s,t \in S$.

The greatest fixpoint of $\mathcal{B}_{C_{\mathsf{Max}}}$ can be computed via Kleene iteration (from above) or by solving the following linear program

$$\max \sum_{s,t \in S} a(s,t)$$

$$\begin{array}{ll} a(s,t) \le a(s',t') & \forall s,t \in S, \eta(s) = \varnothing \Leftrightarrow \eta(t) = \varnothing, \forall (s',t') \in W_{s,t} \\ a(s,t) = 0 & \forall s,t \in S, \eta(s) = \varnothing \Leftrightarrow \eta(t) \neq \varnothing \\ a(s,t) \le 1 & \forall s,t \in S \end{array}$$

Strategy-in-Strategy Iteration. Now, given a fixed strategy C_{Max} for player Max we can define a min-decomposition

$$H^{C_{\mathsf{Max}}}_{\min}(s,t) = \begin{cases} \{h_{s',t'} \mid (s',t') \in W_{s,t}\} & \text{if } \eta(s) = \varnothing \Leftrightarrow \eta(t) = \varnothing \\ \{h_0\} & \text{otherwise} \end{cases}$$

(a min-decomposition of $\mathcal{B}_{C_{\mathsf{Max}}}$) where

$$h_{s',t'}(a) = a(s',t')$$

A min-decomposition fixes an element of the coupling $W_{s,t}$ of states $s,t \in S$. Given a strategy C_{Min} for player Min we have $\mathcal{B}_{C_{\mathsf{Max}}} \colon \{0,1\}^S \to \{0,1\}^S$, defined as

$$(\mathcal{B}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}(a)(s,t) = \begin{cases} a(s',t') & \text{if } \eta(s) = \varnothing \Leftrightarrow \eta(t) = \varnothing \\ h_0 & \text{otherwise} \end{cases}$$

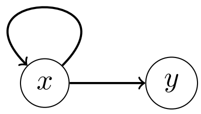
for $a \in \{0,1\}^S$ and $s,t \in S$. Here, $C_{\mathsf{Max}}(s,t) = h^{s,t}_{W_{s,t}}$ and $C_{\mathsf{Min}}(s,t) = h^{s,t}_{s',t'}$. It holds $(s',t') \in W$.

We can compute $\nu(\mathcal{B}_{C_{\mathsf{Max}}}))_{C_{\mathsf{Min}}}$ via Kleene iteration (from above) or by solving the following linear program:

$$\max \sum_{s,t \in S} a(s,t)$$

$$\begin{array}{ll} a(s,t) = a(s',t') & \forall s,t \in S, \eta(s) = \varnothing \Leftrightarrow \eta(t) = \varnothing, C_{\mathsf{Max}}(s,t) = h^{s,t}_{W_{s,t}}, C_{\mathsf{Min}}(s,t) = h_{s',t'} \\ a(s,t) = 0 & \forall s,t \in S, \eta(s) = \varnothing \Leftrightarrow \eta(t) \neq \varnothing \\ a(s,t) \le 1 & \forall s,t \in S \end{array}$$

Example 5.3.5. *Consider the following transition system.*



The max-decomposition is given as follows:

- $H_{\max}(x,x) = \{h_1, h_2\}$ *with* $h_1(a) = \min\{a(x,x), a(y,y)\}$ *and* $h_2(a) = \min\{a(x,y), a(y,x)\}$ *corresponding to the two possible (minimal) couplings of* $\eta(x)$ *and* $\eta(x)$
- $H_{\max}(y,y) = \{h^{y,y}\}$ *with* $h^{y,y}(a) = \min_{(x',y')\in\varnothing} a(x',y') = 1$
- $H_{\max}(x,y) = H_{\max}(y,x) = \{h_0\}$ *with* $h_0(a) = 0$ *(the set of couplings is empty)*

Assume, Max *guesses the strategy* $C^{(0)}(x,x) = h_2$ *(all other strategies are unique). Now* $\nu\mathcal{B}_{C^{(0)}}(x,x) = 0 = \nu\mathcal{B}_{C^{(0)}}(x,y) = \nu\mathcal{B}_{C^{(0)}}(y,x)$ *and* $\nu\mathcal{B}_{C^{(0)}}(y,y) = 1$*. One can see that* $\nu\mathcal{B}_{C^{(0)}} = \mu\mathcal{B}$*. Now,* $\nu\mathcal{B}^{\#}_{\nu\mathcal{B}_{C^{(0)}}} = \{(x,x)\}$ *thus we continue the iteration with the post-fixpoint* $a(x,x) = a(y,y) = 1$ *and* $a(x,y) = a(y,x) = 0$*. Now,* Max *switches his strategy, i.e.* $C^{(1)}(x,x) = h_1$ *and obtains* $\nu\mathcal{B}_{C^{(1)}}(x,x) = \nu\mathcal{B}_{C^{(1)}}(y,y) = 1$ *and* $\nu\mathcal{B}_{C^{(1)}}(x,y) = \nu\mathcal{B}_{C^{(1)}}(y,x) = 0$*. One can see that* $\nu\mathcal{B}_{C^{(1)}} = \nu\mathcal{B}(= a)$*.*

Rewriting $\mathcal{B}$. Taking advantage of complete distributivity (Definition 2.3.6), we can reformulate

$$\begin{aligned}\mathcal{B}(a)(s,t) &= \max_W \{ \min_{(s',t')\in W} a(s',t') \mid W \subseteq S\times S \ \wedge\ u(W) = (\eta(s),\eta(t))\} \\ &= \min_{f\in F} \max_W \{a(f(W)) \mid W \subseteq S\times S \ \wedge\ u(W) = (\eta(s),\eta(t))\}\end{aligned}$$

Now $F = \{f\colon \mathcal{W} \to S\times S\}$ where $\mathcal{W}$ is the set of all couplings of successors of all pairs of states $(s,t)\in S^2$.

A min-decomposition for this rewritten function is given by

$$H_{\min}(s,t) = \begin{cases} h_f^{s,t}(a) & \text{if } \eta(s) = \varnothing \Leftrightarrow \eta(t) = \varnothing \\ \{h_0\} & \text{otherwise}\end{cases}$$

(a min-decomposition of $\mathcal{B}$) where

$$h_f^{s,t}(a) = \max_W \{a(f(W)) \mid W \subseteq S\times S \ \wedge\ u(W) = (\eta(s),\eta(t))\} \text{ and } h_0(a) = 0.$$

Here, a strategy fixes a pair of states in each coupling for all states $s,t\in S$.

One can also easily derive a max-decomposition for the resulting function $\mathcal{B}_{C_{\mathsf{Min}}}$ where C_{Min} is a strategy for player Min. A strategy C_{Max} for player Max would specify the function $(\mathcal{B}_{C_{\mathsf{Max}}})_{C_{\mathsf{Min}}}$ from before.

Example 5.3.6. *Revisit the previous example. We can now perform strategy iteration from above to compute $\nu\mathcal{B}$. There are two ways to couple $\eta(x)$ with $\eta(x)$: $W_1 = \{(x,x),(y,y)\}$ and $W_2 = \{(x,y),(y,x)\}$. A strategy in $H_{\min}(x,x)$ fixes an element in each coupling. Thus the min-decomposition is given as follows:*

- $H_{\min}(x,x) = \{h_1,h_2,h_3,h_4\}$ *with*
  - $h_1(a) = \max\{a(x,x),a(x,y)\}$
  - $h_2(a) = \max\{a(x,x),a(y,x)\}$
  - $h_3(a) = \max\{a(y,y),a(x,y)\}$
  - $h_4(a) = \max\{a(y,y),a(y,x)\}$
- $H_{\min}(y,y) = \{h^{y,y}\}$ *with* $h^{y,y}(a) = 1$ *(we only have the coupling $C = \emptyset$)*
- $H_{\min}(x,y) = H_{\min}(y,x) = \{h_0\}$ *with* $h_0(a) = 0$ *(the set of couplings is empty)*

Here, in fact, any strategy is optimal since player Max would always choose coupling W_1 for (x,x). I.e. let the four strategies be given by $C^1_{Min}, C^2_{Min}, C^3_{Min}, C^4_{Min}$ (corresponding to h_1,h_2,h_3,h_4) we obtain $\nu\mathcal{B} = \nu\mathcal{B}_{C^1_{Min}} = \nu\mathcal{B}_{C^2_{Min}} = \nu\mathcal{B}_{C^3_{Min}} = \nu\mathcal{B}_{C^4_{Min}}$.

We discussed in Section 2.6.2 some algorithms from the literature which compute the bisimilarity. It seems that our strategy iterations have not been considered before, thus a comparison between known algorithms and our strategy iterations might be interesting.

5.3.7. Strategy Iterations for Behavioural Distances of Probabilistic Automata

We aim to analyze the function $\mathcal{M}\colon [0,1]^{S\times S} \to [0,1]^{S\times S}$ which was defined in Section 2.6.5 as

$$\mathcal{M}(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t)) & \text{otherwise} \end{cases}$$

for $d \in [0,1]^{S\times S}$ and $s,t \in S$ ($L = [0,1]$, d_L is the discrete metric). $\mathcal{M}$ usually has multiple fixpoints.

In order to cast this problem in our framework, we identify a suitable min-decomposition of $\mathcal{M}$. Observe that, for $d \in [0,1]^{S\times S}$ and $s,t \in S$ such that $\ell(s) = \ell(t)$, expanding the definitions of the liftings and taking advantage of complete distributivity (Definition 2.3.6), we have

$$\begin{aligned} \mathcal{M}(d)(s,t) &= \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t)) \\ &= \min_{R\in\mathcal{R}(\eta(s),\eta(t))} \max_{(\beta,\beta')\in R} \min_{\omega\in\Omega_V(\beta,\beta')} \sum_{u,v\in S} d(u,v)\cdot\omega(u,v) \\ &= \min_{R\in\mathcal{R}(\eta(s),\eta(t))} \min_{f\in F_R} \max_{(\beta,\beta')\in R} \sum_{u,v\in S} d(u,v)\cdot f(\beta,\beta')(u,v) \end{aligned}$$

where $F_R = \{f\colon R \to \mathcal{D}(S\times S) \mid f(\beta,\beta') \in \Omega_V(\beta,\beta') \text{ for } (\beta,\beta')\in R\}$, which is a finite set and $\mathcal{R}(\eta(s),\eta(t)) = \{R \in \mathcal{P}(\mathcal{D}(S)\times\mathcal{D}(S)) \mid \pi_1(R) = \eta(s) \wedge \pi_2(R) = \eta(t)\}$ is the set of set-couplings of $\eta(s),\eta(t)$.

Min-Decomposition of $\mathcal{M}$. We denote by $H_{\min}$ the min-decomposition of $\mathcal{M}$ defined as follows.

For $s, t \in S$ such that $\ell(s) = \ell(t)$, we let

$$H_{\min}(s,t) = \{h_{R,f} \mid R \in \mathcal{R}(\delta(s), \delta(t)), f \in F_R\},$$

with $h_{R,f} : [0,1]^{S\times S} \to [0,1]$ defined as

$$h_{R,f}(d) = \max_{(\beta,\beta')\in R} \sum_{u,v\in S} d(u,v) \cdot f(\beta,\beta')(u,v).$$

If instead $\ell(s) \neq \ell(t)$, we let $H_{\min}(s,t) = \{h_1\}$ where $h_1(d) = 1$ for all d.

A strategy C in $H_{\min}$ maps each pair of states $s, t \in S$ to a function in $H_{\min}(s,t)$, that is

- if $\ell(s) \neq \ell(t)$, to the unique element $h_1 \in H_{\min}(s,t)$;
- if $\ell(s) = \ell(t)$ to some $h_{R,f} \in H_{\min}(s,t)$, with $R \in \mathcal{R}(\delta(s), \delta(t))$ set-coupling and $f \in F_R$.

We can instantiate the algorithm in Figure 5.2 to compute the least fixpoint from above. The following lemma is helpful in the instantiation of the algorithm, when we need to construct a new strategy.

Lemma 5.3.7. *Let PA $= (S, \eta, L, \ell)$ be a probabilistic automaton and let $H_{\min}$ be the min-decomposition of $\mathcal{M}$. Given a strategy C in H_{min} and $d : Y \times Y \to [0,1]$, a strategy $C'(y) = \arg\min_{h\in H_{\min}(y)} h(d)$ can be defined as follows: for $(s,t) \in S \times S$*

- *if $\ell(s) \neq \ell(t)$ then $C'(s,t) = C(s,t)$*
- *if $\ell(s) = \ell(t)$ then $C'(s,t) = h_{R',f'}$ where*

$$R' = \arg\min_{R\in\mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta')\in R} K(d)(\beta,\beta')$$

and for $(\beta, \beta') \in R'$:

$$f'(\beta,\beta') = \arg \min_{\omega\in\Omega_V(\beta,\beta')} \sum_{u,v\in S} d(u,v) \cdot \omega(u,v).$$

Proof. See Appendix: Lemma A.4.3. □

The algorithm starts by fixing a strategy C_0 (item (1)). Then, at each iteration, if $imp_{\min}(C_i) \neq \varnothing$ (item (2b) which by Lemma 5.2.12, can be checked by verifying if $\mathcal{M}(\mu\mathcal{M}_{C_i}) \sqsubset \mu\mathcal{M}_{C_i})$, we consider a new strategy $C_{i+1} \in imp_{\min}(C_i)$. According to Remark 5.2.10, this can be defined as follows: for $(s,t) \in S \times S$

- if $\ell(s) \neq \ell(t)$ then $C_{i+1}(s,t) = C_i(s,t)$
- if $\ell(s) = \ell(t)$ then $C_{i+1}(s,t) = h_{R',f'}$ chosen in a way that minimises $h_{R',f'}(\mu\mathcal{M}_{C_i})$. Concretely (see Lemma 5.3.7), one can define

$$R' = \arg\min_{R \in \mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta') \in R} K(\mu\mathcal{M}_{C_i})(\beta,\beta')$$

and for $(\beta,\beta') \in R'$:

$$f'(\beta,\beta') = \arg\min_{\omega \in \Omega_V(\beta,\beta')} \sum_{u,v \in S} \mu\mathcal{M}_{C_i}(u,v) \cdot \omega(u,v).$$

If instead, $imp_{\min}(C_i) = \varnothing$ (item (2c)) and thus, by Lemma 5.2.12, $\mu\mathcal{M}_{C_i}$ is a fixpoint of $\mathcal{M}$, we check whether it is the least fixpoint by verifying if $\nu\mathcal{M}_{\#}^{\mu\mathcal{M}_{C_i}} = \varnothing$ (Lemma 3.4.2), and in case it is not, we use Lemma 3.4.2 to determine a pre-fixpoint $a \sqsubset \mu\mathcal{M}_{C_i}$, which is then used to obtain C_{i+1}. The approximation $\mathcal{M}_{\#}^{d}$ is spelled out in Lemma 3.6.17. Furthermore $\mu\mathcal{M}_{C_i}$ is again obtained by linear programming, similar to the case of simple stochastic games (also see [BBL+21]).

Comparison to [BBL+21]. The resulting algorithm is quite similar to the one specifically developed for probabilistic automata in [BBL+21]. In particular, it can be seen that, apart from the different presentation, a strategy C corresponds to what [BBL+21] refers to as a *coupling structure*. In addition, the step in item (2c) of the algorithm (see Figure 5.2) is analogous to that in [BBL+21]. In fact, in order to check whether the fixpoint obtained with the current strategy C_i, i.e. $\mu\mathcal{M}_{C_i}$, is the least fixpoint of $\mathcal{M}$, one considers the approximation $\mathcal{M}_{\#}^{\mu\mathcal{M}_{C_i}}$ and checks whether its greatest fixpoint is empty. Recalling that the post-fixpoints of $\mathcal{M}_{\#}^{\mu\mathcal{M}_{C_i}}$ have been shown in Proposition 3.6.20 to be the self-closed relations of [BBL+21], one derives that verifying the emptiness of the greatest fixpoint of $\mathcal{M}_{\#}^{\mu\mathcal{M}_{C_i}}$ corresponds exactly to checking whether the largest self-closed relation is empty.

More in detail, in our case, a strategy C in $H_{\min}$ maps each pair of states $s,t \in S$ with $\ell(s) = \ell(t)$ to some $h_{R,f} \in H_{\min}(s,t)$, where $R \in \mathcal{R}(\delta(s),\delta(t))$ is a set-coupling and $f \in F_R$ maps each $(\beta,\beta') \in R$ to a probabilistic coupling. Note that the choice of the probabilistic couplings is "local", i.e., we could have different pairs of states $(s,t),(u,v) \in S \times S$ and $C(s,t) = (R,f)$, $C(u,v) = (R',f')$, with $(\beta,\beta') \in R \cap R'$ and $f(\beta,\beta') \neq f'(\beta,\beta')$. However, it is easy to see that we can assume (and it is computationally convenient to do so) that the choice of the probabilistic coupling is actually "global", i.e., that for a strategy C, there is a (partial) function $F : \mathcal{D}(S) \times \mathcal{D}(S) \to \mathcal{D}(S \times S)$ such that for each $(s,t) \in S \times S$ we have $C(s,t) = (R, F_{|R})$. In this view, a strategy C can be identified with a pair (ρ, F), where ρ gives the set-couplings, i.e., $\rho(s,t) \in \mathcal{R}(\delta(s),\delta(t))$, and $F : \mathcal{D}(S) \times \mathcal{D}(S) \to \mathcal{D}(S \times S)$ the probabilistic couplings. This exactly corresponds to the notion of *coupling structure* in [BBL+21].

A difference concerns how strategy updates are performed. While in the algorithm derived above all set-couplings are updated at the same time (ρ-component), in [BBL+21] the set-coupling is updated only for a single pair of states. Since the "local" update produces a min-improvement, also the algorithm in [BBL+21] can be seen as an instance of the algorithm in Figure 5.2. Updating all components can be more expensive, but it might accelerate convergence. A more precise comparison should be carried out via an experimental approach.

5.4. Runtime Comparison

From a theoretical point of view, the number of iterations of both strategy algorithms is bounded by the number of strategies of the corresponding player $p \in \{\min, \max\}$, which is exponential in the input size (the number of strategies is $\prod_{y \in Y} |H_p(y)|$). This suggests that, depending on the setting, the fastest algorithm is the one using the smaller decomposition $H_{\min}$ respectively $H_{\max}$. However, a deeper analysis is still needed, as a smaller decomposition usually leads to a higher cost for computing μf_C.

We will now perform a short runtime comparison for all three presented two player games. These results aim to justify the need for strategy iteration and to answer the question above.

Implementations were done in MATLAB.

5.4.1. Runtime Comparison for Discounted Mean-Payoff Games

We now compare runtimes of the algorithms for randomly generated discounted mean-payoff games. We compare Kleene iteration (KI) with strategy iteration (SI) and strategy-in-strategy iteration (SiSI) (ch. Section 5.3.1). It has to be noted, that Kleene iteration from above and below produces extremely similar runtimes. This is not surprising since discounted mean-payoff games are symmetric with regards to the players' objectives and the games were randomly created. The same holds for strategy iteration and strategy-in-strategy iteration (we chose to always approach from below).

The analyzed games have n states which randomly belong to player Max or Min, we chose $W = n$, $\lambda = 1/2$ and the probability of an edge existing is $1/2$, i.e. for each state an edge to some other state exists with probability $1/2$ with some random weight in $[-W, W]$. Cumulative runtimes (in seconds) are given for 1000 random runs in Figure 5.8. We also give the number of iterations (inner iteration for strategy-in-strategy iteration as the number of outer iterations is the same as for strategy iteration). The tolerance for Kleene iteration is given by 10^{-13} in 2-norm.

It is rather obvious, that strategy-in-strategy iteration is more efficient than Kleene iteration which also just yields an approximation and no exact result. We also note, that Kleene iteration really struggles when λ is close to 1 (which frequently occurs when reducing a mean-payoff game to a disounted mean-payoff game). The other algorithms seemed unaffected.

We also did a second comparison with the same parameters except that we tweaked

	runtime (seconds)			number of iterations		
n	KI	SI	SiSI	KI	SI	SiSI
10	0.04	4.94	0.01	38452	2929	8426
20	0.08	5.88	0.12	40479	3718	13988
30	0.16	6.14	0.03	41375	4169	17502
40	0.27	6.62	0.05	41992	4476	19987
50	0.42	7.16	0.08	42727	4694	22240
60	0.75	8.20	0.10	42998	4906	24001
70	1.27	9.31	0.14	43048	5022	25345
80	2.04	10.88	0.18	43892	5204	26848
90	2.83	12.07	0.22	44021	5280	28007
100	3.92	15.88	0.27	44213	5373	28884

Fig. 5.8.: Runtime comparison for discounted mean-payoff games

the probability of a state belonging to player Max to be $1/4$ and it belonging to player Min with probability $3/4$. Table 5.9 shows this comparison for strategy iteration from above (SIA) and below (SIB) as we wanted to show that - although the number of iterations are smaller in SIB, as there are less strategies to consider - the total runtime balances out since the computation of $\mu\mathcal{L}_C$ is more expansive in SIB.

As one can see, the number of iterations is clearly smaller for SIB but the runtime is actually longer. This enforces the assumption that the total number of strategies may not be in one-to-one correspondance to a lower the runtime.

Varying the other parameters did not yield interesting results.

5.4.2. Runtime Comparison for Simple Stochastic Games

We implemented strategy iteration from above and from below – in the following abbreviated by SIA and SIB – and classical Kleene iteration (KI). In Kleene iteration we terminate with a tolerance of 10^{-14}, i.e., we stop if the change from one iteration to the next is below this value (in 2-norm). Additionally, we implemented strategy-in-strategy iteration from above (SiSIA) and below (SiSIB).

In order to test the algorithms we created random stochastic games with n nodes, where each Max, Min respectively average node has a maximal number of m successors. For each node we choose randomly one of the four types of nodes. Sink nodes are given a random weight uniformly in $[0, 1]$. Successors are randomly assigned to Max and Min nodes and for an average node we assign a random number to each of its successors, followed by normalisation to obtain a probability distribution.

We performed 1000 runs with different randomly created systems for each value of n and $m = n/2$. In Figure 5.10 we display the cumulative runtimes (in seconds) for all algorithms and the number of nodes with a value of 0 (to clarify what kind of systems are created). In Figure 5.11 we display the number of iterations (inner iterations for the

	runtime (seconds)		number of iterations	
n	SIA	SIB	SIA	SIB
10	5.19	5.24	3228	2280
20	5.57	5.90	4103	3064
30	6.48	7.28	4573	3538
40	6.05	7.41	4836	3906
50	6.40	8.69	5073	4071
60	7.71	11.61	5285	4225
70	7.39	12.79	5394	4380
80	7.49	14.52	5562	4499
90	7.88	17.13	5645	4633
100	8.24	20.11	5761	4700

Fig. 5.9.: Second Runtime comparison for discounted mean-payoff games

strategy-in-strategy iteration) and the number of other fixpoints encountered on the way (outer: strategy iteration from above, inner: strategy-in-strategy iteration from below).

Note that SIB usually performs slightly better than SIA. Moreover KI neatly beats both of them. Here we need to remember that KI only converges to the solution and it is known that the rate of convergence can be exponentially slow [Con92]. Also convergence in 2-norm is rather fast when many states have a value of 0.

Note that the linear optimisation problems are quite costly to solve, especially for large systems. Thus additional iterations are substantially more costly compared to KI.

The number of nodes with a payoff of 0 seems to grow linearly with the number of nodes in the system. The number of times SIA/SiSIB gets stuck at a fixpoint different from $\mu\mathcal{V}$ however seems comparatively small.

We note that we also varied the number of nodes which belong to player Max, respectively player Min, i.e. there were substantially more nodes in V_{Max} than in V_{Min} or vice versa. However, although the number of iterations varied, the total runtime of both SIA and SIB did not change that much. This indicated that the extra cost in computing $\mu\mathcal{V}_C$ balances out with the number of iterations.

Strategy-in-strategy iteration performed rather well - in fact better than the usual strategy iterations.

We also did a runtime comparison for randomly generated stopping simple stochastic games in the same manner (see Figure 5.12). Here, each state has exactly two successors and there exist exactly two sinks with payoff 0, respectively 1.

Here, it is rather obvious that strategy iteration is the way to go as Kleene iteration struggles since it may take many iterations to reach a sink. The results are rather unclear which strategy iteration is to be preferred. For example, SIA performed worst when it came to the first comparison, but actually performed the best for 100 nodes in the second comparison. There probably is some way to create systems where each of these four strategy iterations is to be preferred.

	runtime (seconds)					number of payoff 0
n	KI	SIA	SiSIA	SIB	SiSIB	
10	0.27	9.96	1.50	10.19	1.58	2439
20	0.56	13.75	1.84	14.09	1.94	4714
30	1.24	15.70	2.46	15.66	2.65	7268
40	2.61	16.04	3.08	14.82	3.32	9806
50	2.11	16.54	3.94	14.41	4.30	12573
60	1.77	17.93	4.88	14.07	5.43	15151
70	1.46	20.87	6.01	14.37	6.75	17473
80	1.85	26.47	7.44	14.72	8.56	20031
90	2.22	35.80	9.05	15.26	10.65	22390
100	2.69	48.03	10.93	15.86	13.17	25163

Fig. 5.10.: Runtime comparion for simple stochastic games

	number of iterations					number of other fp	
n	KI	SIA	SiSIA	SIB	SiSIB	outer	inner
10	19336	1523	2534	1626	2476	203	300
20	20118	2234	5280	2395	5100	206	386
30	21970	2317	5649	2506	5400	86	166
40	22375	2219	5083	2354	4839	60	97
50	18884	2067	4422	2208	4138	72	103
60	6479	1991	4110	2117	3830	39	50
70	4261	1977	4039	2094	3707	36	47
80	4171	1980	4002	2053	3755	26	34
90	4186	1983	3976	2042	3695	14	17
100	4230	1988	3963	2028	3696	9	13

Fig. 5.11.: Number of iterations

	runtime (seconds)				
n	KI	SIA	SiSIA	SIB	SiSIB
10	0.55	9.72	1.59	11.08	1.62
20	2.08	12.77	2.10	14.04	2.19
30	5.06	14.94	3.20	16.88	3.52
40	14.74	16.75	4.89	18.79	5.69
50	26.64	20.61	8.14	22.73	10.00
60	64.18	23.54	12.55	26.89	16.02
70	73.79	26.31	17.68	29.83	23.55
80	75.36	28.47	23.43	32.85	32.53
90	110.35	32.63	31.73	37.96	45.34
100	150.75	38.22	43.56	45.13	64.31
	number of iteration				
n	KI	SIA	SiSIA	SIB	SiSIB
10	51845	1386	2293	1657	2207
20	60539	1842	3757	2047	3626
30	71433	2100	4983	2379	4813
40	122011	2354	6103	2602	5831
50	137387	2607	7211	2788	6956
60	225775	2715	7945	2957	7766
70	184569	2886	8865	3086	8508
80	156179	3027	9682	3227	9401
90	179107	3137	10351	3365	10082
100	192922	3252	11132	3490	10837

Fig. 5.12.: Runtime comparison for stopping simple stochastic games

Lastly, we note that SIB, SiSIA and SiSIB provide an optimal strategy for player Max which is otherwise rather hard to obtain (see Remark 5.2.23.

5.4.3. Runtime Comparison for Energy Games

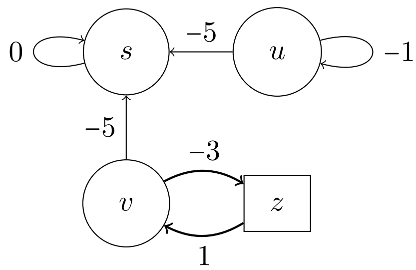
In this section we compare runtimes of our strategy algorithms to known algorithms for energy games.

Value Iteration. We briefly discuss a value iteration technique developed by [BCD⁺11] that resembles a worklist algorithm used in the context of dataflow analysis. The algorithm starts with a value function $g = 0$ and computes a list L of invalid states w.r.t. g, i.e.

- $v \in V_0$ invalid iff $w(v, v') + g(v) - g(v') < 0$ for all $(v, v') \in E$
- $v \in V_1$ invalid iff $w(v, v') + g(v) - g(v') < 0$ for some $(v, v') \in E$

Now, we iteratively pick any state $v \in L$ and increase $g(v)$ until v is valid. This may produce new invalid states which need to be added to L. The algorithm terminates when there are no more invalid states, i.e. $L = \varnothing$, and we obtain $g = \mu\bar{\mathcal{E}}$. We note that this algorithm works for any energy game.

Example 5.4.1. *Consider the follwing energy game (circles belong to player Min, rectangles to player Max):*



Initially $L = \{u, v\}$. Now, we choose $v \in L$ and make it valid, thus $g(v) = 3$. Now, $L = \{u, z\}$ since state z is now invalid. Next, we increase $g(u)$ to 5 to make u valid and $L = \{z\}$ since no other state is added. We increase $g(z) = 2$ which makes v invalid again. Next, we increase $g(v) = 5$ and finally $g(z) = 4$ (since z was made invalid again). The iteration terminates since $L = \varnothing$. Thus $\mu\bar{\mathcal{E}}(s) = 0$, $\mu\bar{\mathcal{E}}(u) = 5$, $\mu\bar{\mathcal{E}}(v) = 5$, $\mu\bar{\mathcal{E}}(z) = 4$.

Runtime Results. We now compare runtimes for each algorithm by generating random games. The number of states in our system is denoted by n, while p represents the probability that any edge (u, v) exists, i.e. given states u and v, the probability that the edge (u, v) exists is given by p. We guarantee at least one outgoing edge for each state. If an edge exists, its weight is some uniformly random integer in $[-W, W]$ where W is the maximum edge weight.

Our runtime-tables use the following abbreviations:

- **TF**: transformation of Γ to Γ'
- **KLE**: Kleene iteration
- **VI**: Value iteration from [BCD+11]
- **SIA**: Strategy iteration for player Min (iteration from above)
- **SIB**: Strategy iteration for player Max (iteration from below)

We use value iteration to compute the $\mu\mathcal{E}_{C_i}$ in both strategy iterations given any strategy C_i. This is more efficient than using linear programming in **SI0**. Additionally, since the value iteration is rather efficient, strategy-in-strategy iteration did not yield better results, thus we omit these runtimes.

Cumulative runtimes (in seconds) are given for 1000 random runs for $W = n$ and $p = {}^2\!/n$, where each state randomly belongs to player Min or player Max. A deviation with respect to W and p barely impacts the runtime comparison.

First, we examine random systems of exclusively finite values:

	KLE	**VI**	**SIB**	**SIA**
$n = 20$	0.04	0.02	0.07	0.27
$n = 40$	0.13	0.05	0.19	1.88
$n = 80$	0.59	0.17	0.74	14.79

It is rather clear that approaching from above (**SIA**) does not seem fruitful since values are usually rather small. Note that **VI** performs the best.

Next, we examine random systems where infinite values are allowed. Hence, we need to transform these systems to systems with finite values (**TF**) beforehand and then apply our algorithms to these reduced system. We note that around every second state requires an infinite initial energy in the original systems:

	TF	**KLE**	**VI**	**SIB**	**SIA**
$n = 20$	0.05	0.33	0.19	0.48	0.1
$n = 40$	0.19	2.6	0.79	2.5	0.3
$n = 80$	1.07	26.7	4.96	19.41	0.98

It is not rare that a handful of states attain a value of around $2 \cdot n \cdot W$. Thus, **SIA** is rather efficient in this instance when choosing the sink state s as the initial successor for each state in V_{Min} (this strategy also guarantees finite values). Here, **SIA** is very competitive, even compared to **VI**. We however note that **SIB** is the only algorithm which produces also an optimal strategy for player Max.

5.5. Summary

We derived abstract algorithms for strategy iterations which compute least/greatest fixpoints. Here, we rely on the function of interest having a suitable min-/max-decomposition. A strategy corresponds to an element of this min-/max-decomposition and our algorithms successively improve the chosen strategy until an optimal one is found. This optimal strategy C corresponds to μf in the sense that $\mu f_C = \mu f$ (or dually to νf).

We were successful in applying these strategy iterations to a handful of applications and showed their usefulness in runtime comparisons.

6 | Conclusion and Future Work

In this thesis we tackled the problem how - given a non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ and a fixpoint a of f - we could verify that a is the least/greatest fixpoint of f. Additionally, we were able to derive methods which yield an exact computation of $\mu f/\nu f$ - generalized strategy iterations. Both theories were used on a handful of applications, i.e. we derived the corresponding approximations and instantiated our generalized strategy iterations to the problems at hand.

We applied these theories to many applications: termination probability of Markov chains, bisimilarity for transition systems, behavioural distances for labeled Markov chains, behavioural distances for metric transition system, behavioural distances for probabilistic automata, discounted mean-payoff games, simple stochastic games and energy games. There exist a handful of other applications that were not discussed in this thesis but can easily be handled by our theories, for example bisimilarity for labeled transition systems and behavioural metrics for Rabin automata. Other applications like probabilistic programs and incorrectness calculus might be interesting to investigate in our framework [MMKK18].

We will briefly review the main results and suggest future work.

6.1. Fixpoint Theory - Upside Down

We devised a method that - given some non-expansive function $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ and some fixpoint a of f - can be used to verify if a is the least/greatest fixpoint of f. To this end, we derived the approximations $f^a_\#\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ and $f_a^\#\colon \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_{f(a)})$. Any fixpoint of these approximations corresponds to a "vicious cycle" in which states convince each other that their value is lower/higher than what it should be. Once such a vicious cycle is detected, the theory provides a constant which can be used to scale up/down the values in the vicious cycle to stay below/above the greatest/least fixpoint. We were also able to derive an (incomplete) proof rules which provides upper/lower bounds for greatest/least fixpoints.

We were able to cast a handful of applications into the desired framework and derived the corresponding approximations. To this end, the (de)composition of non-expansive functions into smaller sub-functions - as well as the corresponding approximations - proved immensely useful.

Future Work Suggestions. In the future we suggest to lift some of the restrictions of our approach. First, an extension to an infinite domain Y would of course be desirable, but since several of our results currently depend on finiteness, such a generalisation does not seem to be easy. We partially tackle this problem in Chapter 4.

The restriction to total orders, instead, seems easier to lift: in particular, if the partially ordered MV-algebra $\bar{\mathbb{M}}$ is of the form $\mathbb{M}^I$ where I is a finite index set and $\mathbb{M}$ an MV-chain. (E.g., finite Boolean algebras are of this type.) In this case, our function space is $\bar{\mathbb{M}}^Y = \left(\mathbb{M}^I\right)^Y \cong \mathbb{M}^{Y\times I}$ and we have reduced to the setting presented in this paper. This will allow us to handle featured transition systems [CCP+12] for compactly specifying software product lines in a single transition system. There, transitions are equipped with boolean formulas that specify for which products (or features) a transition can be taken.

Another suggestion is to investigate whether some examples can be handled with other types of Galois connections: here we used an additive variant, but looking at multiplicative variants (multiplication by a constant factor) might also be fruitful.

Lastly, as the proof rules for pre-/post-fixpoints is incomplete, it would be very desirable to have a way of constructing lower/upper bounds for least/greatest fixpoints - via some sort of algorithm.

6.2. A Monoidal View on Fixpoint Checks

We were able to embed the approximation framework into a categorical setting. The resulting functor # was shown to be gs-monoidal which allowed us to construct the tool UDEfix where the user can compose his/her very own functions and perform fixpoint checks.

Additionally, we defined our approximation framework with infinite domains which we used to handle general predicate liftings and the Wasserstein lifting.

Future Work Suggestions. One important question is still open: we defined a lax functor #, relating the concrete category $\mathbb{C}$ of functions of type $\mathbb{M}^Y \to \mathbb{M}^Z$ – where Y, Z might be infinite – to their approximations, living in $\mathbb{A}$. It is unclear whether # is a proper functor, i.e., preserves composition. For finite sets functoriality derives from a non-trivial result in Section 3.5 and it is unclear whether it can be extended to the infinite case. If so, this would be a valuable step to extend the theory.

We illustrated the approximation for predicate liftings via the powerset and the distribution functor. It would be interesting to study more functors and hence broaden the applicability to other types of transition systems.

Concerning UDEfix, we suggest to extend the tool to compute fixpoints, either via Kleene iteration or strategy iteration (strategy iteration from above and below), as detailed in the next chapter. Furthermore for convenience it would be useful to have support for generating fixpoint functions directly from a given coalgebra, respectively transition system.

6.3. A Lattice-Theoretical View of Strategy Iteration

We developed abstract algorithms for strategy iterations which allow to compute least fixpoints (or, dually, greatest fixpoints) of non-expansive functions over MV-algebras. The idea consists in expressing the function of interest as a minimum (or a maximum), and view the process of computing the function as a game between players Min and Max trying to minimise and maximise, respectively, the outcome. Then the algorithms proceed via a sequence of steps which converge to the least fixpoint from above, progressively improving the strategy of player Min, or from below, progressively improving the strategy of the player Max, until an optimal strategy is found.

We instantiated these iterations to a handful of problems. Here, the main task was to detect a min-/max-decomposition and explain how the computation of least/greatest fixpoints stemming from a given strategy can be archived.

Future Work Suggestions. Our abstract strategy iteration algorithms rely on the assumption that, once a strategy for one of the players is fixed, the optimal "answering" strategy for the opponent can be computed efficiently. Identifying abstract settings where a min- or max-decompositions of a function ensures that the answering strategy can be indeed computed efficiently (e.g., via linear programming as it happens for simple stochastic games), is an interesting direction of future research.

It might be interesting to consider functions where the min (or max) is nested, i.e. functions of the kind $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ defined as

$$f(a)(y) = \iota_y\Big(\min_{h \in H_{\min}(y)} h(a)\Big)$$

for $a \in \mathbb{M}^Y$, functions $\iota_y\colon \mathbb{M} \to \mathbb{M}$ and $H_{\min} \subseteq \mathbb{M}^Y \to \mathbb{M}$. Can we perform strategy iteration for these kind of functions similar to what we are doing here?

6.4. Concluding Remarks

We were able to derive a framework and method for fixpoint checks of non-expansive functions $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ and a generalization of strategy iteration to compute least/greatest fixpoints. The generality of these methods and the various applications we found shows the value of the contribution made in this thesis.

A | Appendix

In this chapter one can find proofs which were deemed too extensive and/or less central and were thus removed from the main part of this thesis.

A.1. Proofs of Chapter 2

Lemma A.1.1 (Lemma 2.3.18)**.** *Let* $\mathbb{M} = (M, \oplus, 0, \overline{(\cdot)})$ *be an MV-algebra. For all* $x, y, z \in M$ *it holds*

1. $x \oplus \overline{x} = 1$
2. $x \sqsubseteq y$ *iff* $\overline{x} \oplus y = 1$ *iff* $x \otimes \overline{y} = 0$ *iff* $y = x \oplus (y \ominus x)$
3. $x \sqsubseteq y$ *iff* $\overline{y} \sqsubseteq \overline{x}$
4. $\oplus$, $\otimes$ *are monotone in both arguments,* $\ominus$ *monotone in the first and antitone in the second argument.*
5. *if* $x \sqsubset y$ *then* $0 \sqsubset y \ominus x$*;*
6. $(x \oplus y) \ominus y \sqsubseteq x$
7. $z \sqsubseteq x \oplus y$ *if and only if* $z \ominus x \sqsubseteq y$*.*
8. *if* $x \sqsubset y$ *and* $z \sqsubseteq \overline{y}$ *then* $x \oplus z \sqsubset y \oplus z$*;*
9. $y \sqsubseteq \overline{x}$ *if and only if* $(x \oplus y) \ominus y = x$*;*
10. $x \ominus (x \ominus y) \sqsubseteq y$ *and if* $y \sqsubseteq x$ *then* $x \ominus (x \ominus y) = y$*.*
11. *Whenever* $\mathbb{M}$ *is an MV-chain,* $x \sqsubset y$ *and* $0 \sqsubset z$ *imply* $(x \oplus z) \ominus y \sqsubset z$

Proof. The proof of properties (1), (2), (3), (4) can be found directly in [Mun07]. For the rest:

5. Immediate consequence of (2). In fact, given $x \sqsubset y$, if we had $y \ominus x = 0$ then by (2), $y = x \oplus (y \ominus x) = x \oplus 0 = x$, contradicting the hypothesis.
6. Observe that $(x \oplus y) \ominus y = \overline{\overline{(x \oplus y)} \oplus y} = \overline{(\overline{x} \ominus y) \oplus y} = \overline{(y \ominus \overline{x}) \oplus \overline{x}} \sqsubseteq \overline{\overline{x}} = x$, where the last inequality is motivated by the fact that $\overline{x} \sqsubseteq (y \ominus \overline{x}) \oplus \overline{x}$ and point (3).

7. The direction from left to right is an immediate consequence of (6). In fact, if $z \sqsubseteq x \oplus y$ then $z \ominus x \sqsubseteq (x \oplus y) \ominus x \sqsubseteq y$.

   The other direction goes as follows: if $z \ominus x \sqsubseteq y$, then – by monotonicity (4) – $(z \ominus x) \oplus x \sqsubseteq y \oplus x = x \oplus y$. The left hand side can be rewritten to $(x \ominus z) \oplus z \sqsupseteq z$.

8. Assume that $x \sqsubset y$ and $z \sqsubseteq \overline{y}$. We know, by property (4) that $x \oplus z \sqsubseteq y \oplus z$. Assume by contradiction that $x \oplus z = y \oplus z$. Then we have

$$\begin{aligned}
\overline{x} &\sqsubseteq \overline{(x \oplus z) \ominus z} && \text{[by properties (3) and (6)]}\\
&\sqsubseteq \overline{(y \oplus z) \ominus z} && \text{[since } x \oplus z = y \oplus z\text{]}\\
&= (\overline{y} \ominus z) \oplus z && \text{[definition of } \ominus\text{]}\\
&= \overline{y} && \text{[since } z \sqsubseteq \overline{y} \text{ and property (2)]}
\end{aligned}$$

   And with point (3) this is a contradiction.

9. Assume $y \sqsubseteq \overline{x}$. We know $(x \oplus y) \ominus y \sqsubseteq x$. If it were $(x \oplus y) \ominus y \sqsubset x$, then $((x \oplus y) \ominus y) \oplus y \sqsubset x \oplus y$, with (8). Since the left-hand side is equal to $(y \ominus (x \oplus y)) \oplus (x \oplus y) \sqsupseteq x \oplus y$, this is a contradiction.

   For the other direction assume that $(x \oplus y) \ominus y = x$. Hence we have $x = (x \oplus y) \ominus y = \overline{\overline{(x \oplus y)} \oplus y}$. By complementing on both sides we obtain $\overline{x} = \overline{(x \oplus y)} \oplus y$ which implies that $y \sqsubseteq \overline{x}$.

10. Observe that, by (7), we have $\overline{y} \sqsubseteq \overline{x} \oplus (\overline{y} \ominus \overline{x}) = \overline{x} \oplus (x \ominus y) = \overline{x \ominus (x \ominus y)}$. Therefore, by (3), $x \ominus (x \ominus y) \sqsubseteq y$, as desired.

    For the second part, assume $y \sqsubseteq x$ and thus, by (3), $\overline{x} \sqsubseteq \overline{y}$. Using (2), we obtain $\overline{y} = \overline{x} \oplus (\overline{y} \ominus \overline{x}) = \overline{x} \oplus \overline{y \oplus \overline{\overline{x}}} = \overline{x} \oplus (x \ominus y)$. Hence $y = \overline{\overline{x} \oplus (x \ominus y)} = x \ominus (x \ominus y)$.

11. We first observe that, given $u, v \in \mathbb{M}$, $u \sqsubseteq v \oplus (u \ominus v)$. This is a direct consequence of axiom (3) of MV-algebras and the definition of natural order.

    Second, in an MV-chain if $u, v \sqsupset 0$, then $u \ominus v \sqsubset u$. In fact, if $u \sqsubseteq v$ and thus $u \ominus v = 0 \sqsubset u$. If instead, $v \sqsubset u$ we have $0 \sqsubset v$ and $u \ominus v \sqsubseteq 1 \ominus v = \overline{v}$, hence by (8) it holds that $0 \oplus (u \ominus v) \sqsubset v \oplus (u \ominus v)$. Recalling that $v \sqsubset u$ and thus by (2), $(u \ominus v) \oplus v = u$, we conclude $u \ominus v \sqsubset u$.

    Now

$$\begin{aligned}
&(x \oplus z) \ominus y\\
&\sqsubseteq (x \oplus (y \ominus x) \oplus (z \ominus (y \ominus x))) \ominus y && \text{[by first obs. above]}\\
&= (y \oplus (z \ominus (y \ominus x))) \ominus y && \text{[since } x \sqsubseteq y \text{, by (2)]}\\
&\sqsubseteq z \ominus (y \ominus x) && \text{[by (6)]}\\
&\sqsubset z && \text{[by second obs. above, since } z \sqsupset 0\\
& && \text{and } y \ominus x \sqsupset 0 \text{ by (5)]}
\end{aligned}$$

□

Lemma A.1.2 (Lemma 2.3.20)**.** *Let* $\mathbb{M}$ *be a complete MV-algebra. Then* $(\mathbb{M}, \oplus, \sqsupseteq)$ *is a unital and commutative quantale.*

Proof. We know $\mathbb{M}$ is a complete lattice. Binary meets are given by

$$x \sqcap y = \overline{\overline{x \oplus \overline{y}} \oplus \overline{y}}. \tag{A.1}$$

Moreover $\oplus$ is associative and commutative, with 0 as neutral element.

It remains to be shown that $\oplus$ distributes with respect to $\sqcap$ (note that $\sqcap$ is the join for the reverse order), i.e., that for all $X \subseteq \mathbb{M}$ and $a \in \mathbb{M}$, it holds

$$a \oplus \bigsqcap X = \bigsqcap \{a \oplus x \mid x \in X\}$$

Clearly, since $\bigsqcap X \leq x$ for all $x \in X$ and $\oplus$ is monotone, we have $a \oplus \bigsqcap X \sqsubseteq \bigsqcap \{a \oplus x \mid x \in X\}$. In order to show that $a \oplus \bigsqcap X$ is the greatest lower bound, let z be another lower bound for $\{a \oplus x \mid x \in X\}$, i.e., $z \sqsubseteq a \oplus x$ for all $x \in X$. Then observe that for $x \in X$, using (A.1), we get

$$x \sqsupseteq x \sqcap \overline{a} = \overline{\overline{(x \oplus a)} \oplus a} \sqsupseteq \overline{\overline{z} \oplus a} = z \ominus a$$

Therefore $\bigsqcap X \sqsupseteq z \ominus a$ and thus

$$a \oplus \bigsqcap X \sqsupseteq a \oplus (z \ominus a) \sqsupseteq z$$

as desired. □

Lemma A.1.3 (Lemma 2.3.25)**.** *Let* $f : \mathbb{M}^Y \to \mathbb{M}^Z$ *be a monotone function, where* $\mathbb{M}$ *is an MV-chain and* Y, Z *are finite sets. Then* f *is non-expansive iff for all* $a \in \mathbb{M}^Y$, $\theta \in \mathbb{M}$ *and* $z \in Z$ *it holds* $f(a \oplus \theta)(z) \ominus f(a)(z) \sqsubseteq \theta$.

Proof. Let f be non-expansive and let $a \in \mathbb{M}^Y$ and $\theta \in \mathbb{M}$. We have that for all $z \in Z$

$$\begin{aligned}
&f(a \oplus \theta)(z) \ominus f(a)(z) \sqsubseteq \\
&\quad \sqsubseteq \|f(a \oplus \theta) \ominus f(a)\| && \text{[by definition of norm]} \\
&\quad \sqsubseteq \|(a \oplus \theta) \ominus a\| && \text{[by hypothesis]} \\
&\quad \sqsubseteq \|\lambda y.\theta\| && \text{[by Lemma 2.3.18(6) and monotonicity of norm]} \\
&\quad = \theta && \text{[by definition of norm]}
\end{aligned}$$

Conversely, assume that for all $a \in \mathbb{M}^Y$, $\theta \in \mathbb{M}$ and $z \in Z$ it holds $f(a \oplus \theta)(z) \ominus f(a)(z) \sqsubseteq \theta$. For $a, b \in \mathbb{M}^Y$, first observe that for all $y \in Y$ it holds $b(y) \ominus a(y) \sqsubseteq \|b \ominus a\|$, hence, if we let $\theta = \|b \ominus a\|$, we have $b \sqsubseteq a \oplus \theta$ and thus, by monotonicity, $f(b) \ominus f(a) \sqsubseteq f(a \oplus \theta) \ominus f(a)$. Thus

$$\begin{aligned}
&\|f(b) \ominus f(a)\| \sqsubseteq \\
&\quad \sqsubseteq \|f(a+\theta) \ominus f(a)\| = \\
&\quad \text{[by the observation above and monotonicity of norm]} \\
&\quad = \max\{f(a+\theta)(z) \ominus f(a)(z) | z \in Z\} && \text{[by definition of norm]} \\
&\quad \sqsubseteq \theta && \text{[by hypothesis]} \\
&\quad = \|b \ominus a\| && \text{[by the choice of } \theta\text{]}
\end{aligned}$$

□

Lemma A.1.4 (Lemma 2.5.4). *Let $ev: F\mathbb{M} \to \mathbb{M}$ be an evaluation map and assume that its corresponding lifting $\tilde{F}: \mathbb{M}^Y \to \mathbb{M}^{FY}$ is well-behaved. Then $\tilde{F}$ is non-expansive iff for all $\delta \in \mathbb{M}$ it holds that $\tilde{F}\delta_Y \sqsubseteq \delta_{FY}$, where δ is seen as the constant δ-predicate on Y, respectively FY.*

Proof. The proof is inspired by [WS22, Lemma 3.9] and uses the fact that a monotone function $f: \mathbb{M}^Y \to \mathbb{M}^Z$ is non-expansive iff $f(a \oplus \delta) \sqsubseteq f(a) \oplus \delta$ for all a, δ.

"⇒" Fix a set Y and assume that $\tilde{F}: \mathbb{M}^Y \to \mathbb{M}^{FY}$ is non-expansive. Then

$$\tilde{F}(\delta) = \tilde{F}(0 \oplus \delta) \sqsubseteq \tilde{F}(0) \oplus \delta \sqsubseteq 0 \oplus \delta = \delta$$

"⇐" Now assume that $\tilde{F}(\delta) \sqsubseteq \delta$. Then, using the lemma referenced above,

$$\tilde{F}(a \oplus \delta) \sqsubseteq \tilde{F}(a) \oplus \tilde{F}(\delta) \sqsubseteq \tilde{F}(a) \oplus \delta$$

In both cases we write δ for both δ_Y, δ_{FY} and both deductions rely on the fact that $\tilde{F}$ is well-behaved. □

Lemma A.1.5 (Lemma 2.7.32). *Let $\Gamma_E = (G, w)$ be an energy game with finite values, bounded by k. Then $\mu\mathcal{E} = g$, i.e. the least fixpoint of $\mathcal{E}$ coincides with the solution of Γ_E.*

Proof. From Lemma 2.7.29 we have $g = \mu\bar{\mathcal{E}}$.

The claim follows straightforwardly from the fact that, by distributivity, $\mathcal{E}(a) = \min\{\bar{\mathcal{E}}(a), k\}$. Hence $\mathcal{E} \leq \bar{\mathcal{E}}$ and so $\mu\mathcal{E} \leq \mu\bar{\mathcal{E}}$. For the other direction observe that by assumption the solution g is bounded by k ($g \leq k$). Hence $\bar{\mathcal{E}}(\mu\mathcal{E}) \leq \bar{\mathcal{E}}(\mu\bar{\mathcal{E}}) = \mu\bar{\mathcal{E}} = g \leq k$. Hence $\bar{\mathcal{E}}(\mu\mathcal{E}) = \min\{\bar{\mathcal{E}}(\mu\mathcal{E}), k\} = \mathcal{E}(\mu\mathcal{E}) = \mu\mathcal{E}$, which means that $\mu\mathcal{E}$ is some fixpoint of $\bar{\mathcal{E}}$, implying that $\mu\bar{\mathcal{E}} \leq \mu\mathcal{E}$. □

A.2. Proofs of Chapter 3

Lemma A.2.1 (Lemma 3.2.11)**.** *Let $\mathbb{M}$ be an MV-chain and let Y be a finite set. Then for any $b \in \mathbb{M}^Y$ there are $Y_1, \dots, Y_n \subseteq Y$ with $Y_{i+1} \subseteq Y_i$ for $i \in \{1, \dots, n-1\}$ and $\delta^i \in \mathbb{M}$, $0 \neq \delta^i \sqsubseteq \overline{\bigoplus_{j=1}^{i-1} \delta^j}$ for $i \in \{1, \dots, n\}$ such that*

$$b = \bigoplus_{i=1}^{n} \delta^i_{Y_i} \qquad \textit{and} \qquad \|b\| = \bigoplus_{i=1}^{n} \delta^i.$$

where we assume that an empty sum evaluates to 0.

Proof. Given $b \in \mathbb{M}^Y$, consider $V = \{b(y) \mid y \in Y\}$. If V is empty, then Y is empty and thus $b = 1_Y$, i.e., we can take $n = 1$, $\delta^1 = 1$ and $Y_1 = Y$. Otherwise, if $Y \neq \emptyset$, then V is a finite non-empty set. Let $V = \{v_1, \dots, v_n\}$, with $v_i \sqsubseteq v_{i+1}$ for $i \in \{1, \dots, n-1\}$. For $i \in \{1, \dots, n\}$ define $Y_i = \{y \in Y \mid v_i \sqsubseteq b(y)\}$. Clearly, $Y_1 \supseteq Y_2 \supseteq \dots \supseteq Y_n$. Moreover let $\delta^1 = v_1$ and $\delta^{i+1} = v_{i+1} \ominus v_i$ for $i \in \{1, \dots, n-1\}$.

Observe that for each i, we have $v_i = \bigoplus_{j=1}^{i} \delta^i$, as it can easily shown by induction. Hence $\delta^{i+1} = v_{i+1} \ominus v_i = v_{i+1} \ominus \bigoplus_{j=1}^{i} \delta^i \sqsubseteq 1 \ominus \bigoplus_{j=1}^{i} \delta^i = \overline{\bigoplus_{j=1}^{i} \delta^i}$.

We now show that $b = \bigoplus_{i=1}^{n} \delta^i_{Y_i}$ by induction on n.

- If $n = 1$ then $V = \{v_1\}$ and thus b is a constant function $b(y) = v_1$ for all $y \in Y$. Hence $Y_1 = Y$ and thus $b = \delta^1_Y = \delta^1_{Y_1}$, as desired.
- If $n > 1$, let $b' \in \mathbb{M}^Y$ defined by $b'(y) = b(y)$ for $y \in Y \backslash Y_n$ and $b'(y) = v_{n-1}$ for $y \in Y_n$. Note that $\{b'(y) \mid y \in Y\} = \{v_1, \dots, v_{n-1}\}$. Hence, by inductive hypothesis, $b' = \bigoplus_{i=1}^{n-1} \delta^i_{Y_i}$. Moreover, $b'(y) = b \oplus \delta^n_{Y_n}$, and thus we conclude.

Finally observe that the statement requires $\delta^i \neq 0$ for all i. We can enjoy this property by just omitting the first summand when $v_1 = 0$. □

Lemma A.2.2 (Lemma 3.2.12)**.** *Let $\mathbb{M}$ be an MV-chain, let Y, Z be finite sets and let $f : \mathbb{M}^Y \to \mathbb{M}^Z$ be a non-expansive function. Let $a \in \mathbb{M}^Y$. For $b \in [a, a \oplus \delta]$, let $b \ominus a = \bigoplus_{i=1}^{n} \delta^i_{Y_i}$ be a standard form for $b \ominus a$. If $\gamma_{f(a),\delta}(f(b)) \neq \emptyset$ then $Y_n = \gamma_{a,\delta}(b)$ and $\gamma_{f(a),\delta}(f(b)) \subseteq f^{\#}_{a,\delta^n}(Y_n)$.*

Proof. By hypothesis $\gamma_{f(a),\delta}(f(b)) \neq \emptyset$. Let $z \in \gamma_{f(a),\delta}(f(b))$. This means that $\delta \sqsubseteq f(b)(z) \ominus f(a)(z)$. First observe that

$$\begin{aligned}
\delta &\sqsubseteq f(b)(z) \ominus f(a)(z) && \text{[by hypothesis]}\\
&\sqsubseteq \|f(b) \ominus f(a)\| && \text{[by definition of norm]}\\
&\sqsubseteq \|b \ominus a\| && \text{[by non-expansiveness of } f\text{]}\\
&\sqsubseteq \delta && \text{[since } b \in [a, a \oplus \delta]\text{]}
\end{aligned}$$

Hence

$$\|f(b) \ominus f(a)\| = \delta = \|b \ominus a\| = \bigoplus_{i=1}^{n} \delta^i.$$

Also observe that, since $\delta^n \neq 0$, we have $(b \ominus a)(z) = \delta$ iff $z \in Y_n$. In fact, if $z \in Y_n$ then $z \in Y_i$ for all $i \in \{1, \dots, n\}$ and thus $(b \ominus a)(z) = \bigoplus_{i=1}^{n} \delta^i_{Y_i}(z) = \bigoplus_{i=1}^{n} \delta^i = \delta$. Conversely, if $z \notin Y_n$, then $(b \ominus a)(z) \sqsubseteq \bigoplus_{i=1}^{n-1} \delta^i \sqsubset \delta$. In fact, $0 \sqsubset \delta^n$ and $\bigoplus_{i=1}^{n-1} \delta^i \sqsubseteq \overline{\delta^n}$. Thus by Lemma 2.3.18(8), $\bigoplus_{i=1}^{n-1} \delta^i \sqsubset \delta^n \oplus \bigoplus_{i=1}^{n-1} \delta^i = \bigoplus_{i=1}^{n} \delta^i = \delta$. Hence $Y_n = \gamma_{a,\delta}(b)$.

Let us now show that $\gamma_{f(a),\delta}(f(b)) \subseteq f^{\#}_{a,\delta^n}(Y_n)$. Given $z \in \gamma_{f(a),\delta}(f(b))$, we show that $z \in f^{\#}_{a,\delta^n}(Y_n)$. Observe that

$$\delta \sqsubseteq f(b)(z) \ominus f(a)(z) =$$

[by hypothesis]

$$= f(a \oplus (b \ominus a))(z) \ominus f(a)(z) =$$

[by Lemma 2.3.18(2), since $a \sqsubseteq b$]

$$= f(a \oplus \bigoplus_{i=1}^{n} \delta^i_{Y_i})(z) \ominus f(a)(z) =$$

[by construction]

$$= f(a \oplus \bigoplus_{i=1}^{n} \delta^i_{Y_i}))(z) \ominus f(a \oplus \delta^n_{Y_n})(z) \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by Lemma 2.3.18(2), since $f(a \oplus \delta^n_{Y_n})(z) \sqsubseteq f(a \oplus \bigoplus_{i=1}^{n} \delta^i_{Y_i})(z)$]

$$\sqsubseteq \|f(a \oplus \bigoplus_{i=1}^{n} \delta^i_{Y_i}) \ominus f(a \oplus \delta^n_{Y_n})\| \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by definition of norm and monotonicity of $\oplus$]

$$\sqsubseteq \|a \oplus \bigoplus_{i=1}^{n} \delta^i_{Y_i} \ominus (a \oplus \delta^n_{Y_n})\| \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by non-expansiveness of f and monotonicity of $\oplus$]

$$= \|a \oplus \delta^n_{Y_n} \oplus \bigoplus_{i=1}^{n-1} \delta^i_{Y_i} \ominus (a \oplus \delta^n_{Y_n})\| \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by algebraic manipulation]

$$\sqsubseteq \|\bigoplus_{i=1}^{n-1} \delta^i_{Y_i}\| \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by Lemma 2.3.18(6) and monotonicity of norm]

$$\sqsubseteq \bigoplus_{i=1}^{n-1} \delta^i \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by Lemma 2.3.22(1) and the fact that $\|\delta^i_{Y_i}\| = \delta^i$]

$$= (\delta \ominus \delta^n) \oplus f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$$

[by construction, since $\delta^n = \overline{\bigoplus_{i=1}^{n-1} \delta^i}$]

If we subtract $\delta \ominus \delta^n$ on both sides, we get $\delta \ominus (\delta \ominus \delta^n) \sqsubseteq f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z)$, i.e., since, by Lemma 2.3.18(10), $\delta \ominus (\delta \ominus \delta^n) = \delta^n$ we conclude

$$\delta^n \sqsubseteq f(a \oplus \delta^n_{Y_n})(z) \ominus f(a)(z).$$

Hence $z \in \gamma_{f(a),\delta^n}(f(\alpha_{a,\delta^n}(Y_n)) = f^{\#}_{a,\delta^n}(Y_n)$, which is the desired result. □

Lemma A.2.3 (Lemma 3.3.1). *Let $\mathbb{M}$ be a complete MV-chain, Y a finite set and $f : \mathbb{M}^Y \to \mathbb{M}^Y$ be a non-expansive function. Let $a \in \mathbb{M}^Y$ be a pre-fixpoint of f (i.e., $f(a) \sqsubseteq a$), let $f^{\#}_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Y]_{f(a)})$ be the a-approximation of f. Assume $\nu f \not\sqsubseteq a$ and let $Y' = \{y \in [Y]_a \mid \nu f(y) \ominus a(y) = \|\nu f \ominus a\|\}$. Then for all $y \in Y'$ it holds $a(y) = f(a)(y)$ and $Y' \subseteq f^{\#}_a(Y')$.*

Proof. Let $\delta = \|\nu f \ominus a\|$. Assume $\nu f \not\sqsubseteq a$, i.e., there exists $y \in Y$ such that $\nu f(y) \not\sqsubseteq a(y)$. Since the order is total, this means that $a(y) \sqsubset \nu f(y)$. Hence, by Lemma 2.3.18(5), $\nu f(y) \ominus a(y) \sqsupset 0$. Then $\delta = \|\nu f \ominus a\| \sqsupset 0$. Moreover, for all $y \in Y'$, $\overline{a(y)} = 1 \ominus a(y) \sqsupseteq \nu f(y) \ominus a(y) = \delta$.

First, observe that

$$\nu f \sqsubseteq a \oplus \delta, \tag{A.2}$$

since for all $y \in Y$ $\nu f(y) \ominus a(y) \sqsubseteq \delta$ by definition of δ and then (A.2) follows from Lemma 2.3.18(7).

Concerning the first part, let $y \in Y'$. Since a is a pre-fixpoint, $f(a)(y) \sqsubseteq a(y)$. Assume by contradiction that $f(a)(y) \sqsubset a(y)$. Then we have

$f(a \oplus \delta)(y) =$
[by Lemma 2.3.18(2), since f is monotone and thus $f(a) \sqsubseteq f(a \oplus \delta)$]
$= f(a)(y) \oplus (f(a \oplus \delta)(y) \ominus f(a)(y))$
[since f is non-expansive, by Lemma 2.3.25, hence $f(a \oplus \delta)(y) \ominus f(a)(y) \sqsubseteq \delta$]
$\sqsubseteq f(a)(y) \oplus \delta$
[by $f(a)(y) \sqsubset a(y)$, $\delta \sqsubseteq \overline{a(y)}$ and Lemma 2.3.18(6)]
$\sqsubset a(y) \oplus \delta$
[by Lemma 2.3.18(2) since $a(y) \sqsubseteq \nu f(y)$ and $\delta = \nu f(y) \ominus a(y)$]
$= \nu f(y)$
$= f(\nu f)(y)$
[since $\nu f \sqsubseteq a \oplus \delta$ (A.2) and f monotone]
$\sqsubseteq f(a \oplus \delta)(y)$

i.e., a contradiction. Hence it must be $a(y) = f(a)(y)$.

For the second part, in order to show $Y' \subseteq f^{\#}_a(Y')$, we let $b = \nu f \sqcup a$. By using (A.2) we immediately have that $b \in [a, a \oplus \delta]$.

We next prove that

$$Y' = \gamma_{a,\delta}(b).$$

We show separately the two inclusions. If $y \in Y'$ then $a(y) \sqsubset \nu f(y)$ and thus $b(y) = a(y) \sqcup \nu f(y) = \nu f(y)$ and thus $b(y) \ominus a(y) = \nu f(y) \ominus a(y) = \delta$. Hence $y \in \gamma_{a,\delta}(b)$. Conversely, if $y \in \gamma_{a,\delta}(b)$, then $a(y) \sqsubset \nu f(y)$. In fact, if it were $a(y) \sqsupseteq \nu f(y)$, then, by definition of b we would have $b(y) = a(y)$ and $b(y) \ominus a(y) = 0 \not\sqsupseteq \delta$. Therefore, $b(y) = \nu f(y)$ and thus $\nu f(y) \ominus a(y) = b(y) \ominus a(y) \sqsupseteq \delta$, whence $y \in Y'$.

We can now conclude. In fact, since f is non-expansive, by Theorem 3.2.13(a), we have

$$\gamma_{f(a),\delta}(f(b)) \subseteq f_a^{\#}(Y').$$

Moreover $Y' \subseteq \gamma_{f(a),\delta}(f(b))$. In fact, let $y \in Y'$, i.e., $y \in [Y]_a$ and $\delta \sqsubseteq b(y) \ominus a(y)$. Since $a(y) = f(a)(y)$, we have that $y \in [Y]_{f(a)}$. In order to conclude that $y \in \gamma_{f(a),\delta}(f(b))$ it is left to show that $\delta \sqsubseteq f(b)(y) \ominus f(a)(y)$. We have

$$\begin{aligned}
f(b)(y) \ominus f(a)(y) &= f(b)(y) \ominus a(y) && [\text{since } y \in Y'] \\
&= f(\nu f \sqcup a)(y) \ominus a(y) && [\text{definition of } b] \\
&\sqsupseteq (f(\nu f)(y) \sqcup f(a)(y)) \ominus a(y) && [\text{properties of } \sqcup] \\
&= (\nu f(y) \sqcup a(y)) \ominus a(y) && [\text{since } \nu f \text{ fixpoint and } y \in Y'] \\
&= b(y) \ominus a(y) && [\text{definition of } b] \\
&\sqsupseteq \delta && [\text{since } y \in Y']
\end{aligned}$$

Combining the two inclusions, we have $Y' \subseteq f_a^{\#}(Y')$, as desired. □

Proposition A.2.4 (Proposition 3.5.4)**.** *The basic functions from Definition 3.5.3 are all non-expansive.*

Proof.

- *Constant functions:* immediate.
- *Reindexing:* Let $u : Z \to Y$. For all $a, b \in \mathbb{M}^Y$, we have

$$\begin{aligned}
&\|u^*(b) \ominus u^*(a)\| \\
&= \max_{z \in Z}(b(u(z)) \ominus a(u(z))) \\
&\sqsubseteq \max_{y \in Y}(b(y) \ominus a(y)) && [\text{since } u(Z) \subseteq Y] \\
&= \|b \ominus a\| && [\text{by definition of norm}]
\end{aligned}$$

- *Minimum:* Let $\mathcal{R} \subseteq Y \times Z$ be a relation. For all $a, b \in \mathbb{M}^Y$, we have

$$\|\min_{\mathcal{R}}(b) \ominus \min_{\mathcal{R}}(a)\| = \max_{z \in Z}(\min_{y \mathcal{R} z} b(y) \ominus \min_{y \mathcal{R} z} a(y))$$

Observe that

$$\max_{z\in Z}(\min_{y\mathcal{R}z} b(y) \ominus \min_{y\mathcal{R}z} a(y)) = \max_{z\in Z'}(\min_{y\mathcal{R}z} b(y) \ominus \min_{y\mathcal{R}z} a(y))$$

where $Z' = \{z \in Z \mid \exists\, y \in Y.\, y\mathcal{R}z\}$, since on every other $z \in Z\backslash Z'$ the difference would be 0. Now, for every $z \in Z'$, take $y_z \in Y$ such that $y_z\mathcal{R}z$ and $a(y_z) = \min_{y\mathcal{R}z} a(y)$. Such a y_z is guaranteed to exist whenever Y is finite. Then, we have

$$\begin{aligned}
&\max_{z\in Z'}(\min_{y\mathcal{R}z} b(y) \ominus \min_{y\mathcal{R}z} a(y)) \\
&\sqsubseteq \max_{z\in Z'}(b(y_z) \ominus a(y_z)) && [\ominus \text{ monotone in first arg.}] \\
&\sqsubseteq \max_{z\in Z'} \|b \ominus a\| && [\text{by definition of norm}] \\
&= \|b \ominus a\| && [\|b \ominus a\| \text{ is independent from } z]
\end{aligned}$$

- *Maximum:* Let $\mathcal{R} \subseteq Y \times Z$ be a relation. For all $a, b \in \mathbb{M}^Y$ we have

$$\begin{aligned}
&\|\max_{\mathcal{R}}(b) \ominus \max_{\mathcal{R}}(a)\| \\
&= \max_{z\in Z}(\max_{y\mathcal{R}z} b(y) \ominus \max_{y\mathcal{R}z} a(y)) \\
&\sqsubseteq \max_{z\in Z}(\max_{y\mathcal{R}z}((b(y) \ominus a(y)) \oplus a(y)) \ominus \max_{y\mathcal{R}z} a(y)) \\
&\qquad [\text{since } (b(y) \ominus a(y)) \oplus a(y) = a(y) \sqcup b(y) \text{ and } \ominus \text{ monotone in first arg.}] \\
&\sqsubseteq \max_{z\in Z}((\max_{y\mathcal{R}z}(b(y) \ominus a(y)) \oplus \max_{y\mathcal{R}z} a(y)) \ominus \max_{y\mathcal{R}z} a(y)) \\
&\qquad [\text{by definition of max and monotonicity of } \oplus] \\
&\sqsubseteq \max_{z\in Z} \max_{y\mathcal{R}z}(b(y) \ominus a(y)) \qquad [\text{by Lemma 2.3.18(6)}] \\
&\sqsubseteq \max_{z\in Z} \max_{y\mathcal{R}z} \|b \ominus a\| \qquad [\text{by definition of norm}] \\
&= \|b \ominus a\| \qquad [\text{since } \|b \ominus a\| \text{ is independent}]
\end{aligned}$$

- *Average:* We first note that, when $p : Y \to \mathbb{M}$, with Y finite, is a distribution, then an inductive argument based on weak distributivity, allows one to show that for all $x \in \mathbb{M}$, $Y' \subseteq Y$, $x \odot \bigoplus_{y\in Y'} p(y) = \bigoplus_{y\in Y'} x \odot p(y)$.

  For all $a, b \in \mathbb{M}^Y$ we have

$$\begin{aligned}
&\|\mathrm{av}_D(b) \ominus \mathrm{av}_D(a)\| \\
&= \max_{p\in D}(\bigoplus_{y\in Y} p(y) \odot b(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)) \\
&\sqsubseteq \max_{p\in D}(\bigoplus_{y\in Y} p(y) \odot ((b(y) \ominus a(y)) \oplus a(y)) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)) \\
&\qquad [\text{by monotonicity of } \odot, \oplus, \ominus \text{ and } (b(y) \ominus a(y)) \oplus a(y) = a(y) \sqcup b(y)]
\end{aligned}$$

$$= \max_{p \in D} (\bigoplus_{y \in Y} (p(y) \odot (b(y) \ominus a(y))) \oplus (p(y) \odot a(y)) \ominus \bigoplus_{y \in Y} p(y) \odot a(y))$$

[since $b(y) \ominus a(y) \sqsubseteq 1 \ominus a(y) = \overline{a(y)}$, and $\odot$ weakly distributes over $\oplus$]

$$= \max_{p \in D} ((\bigoplus_{y \in Y} p(y) \odot (b(y) \ominus a(y)) \oplus \bigoplus_{y \in Y} p(y) \odot a(y)) \ominus \bigoplus_{y \in Y} p(y) \odot a(y))$$

$$\sqsubseteq \max_{p \in D} \bigoplus_{y \in Y} p(y) \odot (b(y) \ominus a(y)) \qquad \text{[by Lemma 2.3.18(6)]}$$

$$\sqsubseteq \max_{p \in D} \bigoplus_{y \in Y} p(y) \odot \|b \ominus a\| \qquad \text{[by definition of norm and monotonicity of } \odot\text{]}$$

$$= \max_{p \in D} \|b \ominus a\| \odot \bigoplus_{y \in Y} p(y) \qquad \text{[since } p \text{ is a distr. and } \odot \text{ weakly distributes over } \oplus\text{]}$$

$$= \max_{p \in D} (\|b \ominus a\| \odot 1) \qquad \text{[since } p \text{ is a distribution over } Y\text{]}$$

$$= \|b \ominus a\| \qquad \text{[since } \|b \ominus a\| \text{ is independent from } p\text{]}$$

- *Addition:* Let $w, a, b \in \mathbb{M}^Y$. Without loss of generality, we have

$$\|add_w(b) \ominus add_w(a)\| = add_w(b) \ominus add_w(a) = add_w(b)(y) \ominus add_w(a)(y)$$

for some $y \in Y$ (otherwise swap a and b, the maximum is obtained at some $y \in Y$). Now,

$$\begin{aligned}
\|add_w(b) \ominus add_w(a)\| &= add_w(b)(y) \ominus add_w(a)(y) \\
&= (b(y) \oplus w(y)) \ominus (a(y) \oplus w(y)) \\
&= \overline{\overline{(b(y) \oplus w(y))} \oplus (a(y) \oplus w(y))} && \text{[Definition of } \ominus\text{]} \\
&= \overline{\overline{(b(y) \oplus w(y))} \oplus (w(y) \oplus a(y))} && \text{[Commutativity]} \\
&= \overline{(\overline{(b(y) \oplus w(y))} \oplus w(y)) \oplus a(y)} && \text{[Associativity]} \\
&\sqsubseteq \overline{\overline{b(y)} \oplus a(y)} && \text{[see below]} \\
&= b(y) \ominus a(y) && \text{[Definition of } \ominus\text{]} \\
&\sqsubseteq \|b \ominus a\|.
\end{aligned}$$

One inequality remains to be shown. For $x, y \in \mathbb{M}$, it holds

$$\begin{aligned}
x &\sqsupseteq (x \oplus y) \ominus y && \text{[Lemma 2.3.18(6)]} \\
\Leftrightarrow \overline{x} &\sqsubseteq \overline{(x \oplus y) \ominus y} && \text{[Lemma 2.3.18(3)]} \\
&= \overline{(x \oplus y)} \oplus y && \text{[Definition of } \ominus\text{]}
\end{aligned}$$

Thus we have ($x = b(y)$, $y = w(y)$)

$$\overline{b(y)} \sqsubseteq \overline{(b(y) \oplus w(y))} \oplus w(y)$$

which implies

$$\overline{b(y)} \oplus a(y) \sqsubseteq (\overline{(b(y) \oplus w(y))} \oplus w(y)) \oplus a(y)$$

by Lemma 2.3.18(4). Thus, by 2.3.18(3),

$$\overline{((\overline{(b(y) \oplus w(y))} \oplus w(y)) \oplus a(y)} \sqsubseteq \overline{\overline{b(y)} \oplus a(y)}$$

holds.

- *Substraction:* Let $w, a, b \in \mathbb{M}^Y$. Without loss of generality, we have

$$\|sub_w(b) \ominus sub_w(a)\| = sub_w(b) \ominus sub_w(a) = sub_w(b)(y) \ominus sub_w(a)(y)$$

for some $y \in Y$ (otherwise swap a and b, the maximum is obtained at some $y \in Y$). Now,

$$\begin{aligned}
\|sub_w(b) \ominus sub_w(a)\| &= sub_w(b)(y) \ominus sub_w(a)(y) \\
&= (b(y) \ominus w(y)) \ominus (a(y) \ominus w(y)) \\
&= \overline{\overline{(b(y) \ominus w(y))} \oplus (a(y) \ominus w(y))} && \text{[Definition of } \ominus] \\
&= \overline{(\overline{b(y)} \oplus w(y)) \oplus \overline{(\overline{a(y)} \oplus w(y))}} && \text{[Definition of } \ominus] \\
&= \overline{\overline{(\overline{a(y)} \oplus w(y))} \oplus (w(y) \oplus \overline{b(y)})} && \text{[Commutativity]} \\
&= \overline{((\overline{\overline{a(y)} \oplus w(y))} \oplus w(y)) \oplus \overline{b(y)}} && \text{[Associativity]} \\
&\sqsubseteq \overline{a(y) \oplus \overline{b(y)}} && \text{[see below]} \\
&= \overline{\overline{b(y)} \oplus a(y)} && \text{[Commutativity]} \\
&= b(y) \ominus a(y) && \text{[Definition of } \ominus] \\
&\sqsubseteq \|b \ominus a\|.
\end{aligned}$$

By considerations in the proof for addition it holds

$$a(y) \sqsubseteq \overline{(\overline{a(y)} \oplus w(y))} \oplus w(y)$$

$(x = \overline{a(y)},\ y = w(y))$ and therefore by Lemma 2.3.18(4)

$$a(y) \oplus \overline{b(y)} \sqsubseteq ((\overline{\overline{a(y)} \oplus w(y))} \oplus w(y)) \oplus \overline{b(y)}$$

which implies

$$\overline{((\overline{\overline{a(y)} \oplus w(y))} \oplus w(y)) \oplus \overline{b(y)}} \sqsubseteq \overline{a(y) \oplus \overline{b(y)}}$$

by Lemma 2.3.18(3).

□

Proposition A.2.5 (Proposition 3.5.5)**.** *Let* $\mathbb{M}$ *be an MV-chain,* Y, Z *be finite sets and let* $a \in \mathbb{M}^Y$*.*

- Constant: *for* $k : \mathbb{M}^Z$*, the approximations* $(c_k)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{c_k(a)})$*,* $(c_k)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{c_k(a)})$ *are*

$$(c_k)^\#_a(Y') = \emptyset = (c_k)^a_\#(Y')$$

- Reindexing: *for* $u : Z \to Y$*, the approximations* $(u^*)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{u^*(a)})$*,* $(u^*)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{u^*(a)})$ *are*

$$(u^*)^\#_a(Y') = (u^*)^a_\#(Y') = u^{-1}(Y') = \{z \in [Z]_{u^*(a)} \mid u(z) \in Y'\}$$

- Min: *for* $\mathcal{R} \subseteq Y \times Z$*, the approximations* $(\min_\mathcal{R})^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{\min_\mathcal{R}(a)})$*,* $(\min_\mathcal{R})^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{\min_\mathcal{R}(a)})$ *are given below, where* $\mathcal{R}^{-1}(z) = \{y \in Y \mid y\mathcal{R}z\}$*:*

$$(\min_\mathcal{R})^\#_a(Y') = \{z \in [Z]_{\min_\mathcal{R}(a)} \mid \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y) \subseteq Y'\}$$
$$(\min_\mathcal{R})^a_\#(Y') = \{z \in [Z]^{\min_\mathcal{R}(a)} \mid \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset\}$$

- Max: *for* $\mathcal{R} \subseteq Y \times Z$*, the approximations* $(\max_\mathcal{R})^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([Z]_{\max_\mathcal{R}(a)})$*,* $(\max_\mathcal{R})^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{\max_\mathcal{R}(a)})$ *are*

$$(\max_\mathcal{R})^\#_a(Y') = \{z \in [Z]_{\max_\mathcal{R}(a)} \mid \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset\}$$
$$(\max_\mathcal{R})^a_\#(Y') = \{z \in [Z]^{\max_\mathcal{R}(a)} \mid \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y) \subseteq Y'\}$$

- Average: *for a finite* $D \subseteq \mathcal{D}_\mathbb{M}(Y)$*, the approximations* $(\mathrm{av}_D)^\#_a : \mathcal{P}([Y]_a) \to \mathcal{P}([D]_{\mathrm{av}_D(a)})$*,* $(\mathrm{av}_D)^a_\# : \mathcal{P}([Y]^a) \to \mathcal{P}([D]^{\mathrm{av}_D(a)})$ *are*

$$(\mathrm{av}_D)^\#_a(Y') = \{p \in [D]_{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\}$$
$$(\mathrm{av}_D)^a_\#(Y') = \{p \in [D]^{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\},$$

where $supp(p) = \{y \in Y \mid p(y) > 0\}$ *for* $p \in \mathcal{D}(Y)$*.*

- Addition: *for* $w \in \mathbb{M}^Y$*, the approximations* $(add_w)^\#_a : \mathbb{M}^{[Y]_a} \to \mathbb{M}^{[Y]_{add_w(a)}}$*,* $(add_w)^a_\# : \mathbb{M}^{[Y]^a} \to \mathbb{M}^{[Y]^{add_w(a)}}$ *are*

$$(add_w)^\#_a(Y') = \{y \in Y' \mid a(y) \oplus w(y) \sqsubset 1\}$$
$$(add_w)^a_\#(Y') = \{y \in Y' \mid w(y) \sqsubseteq \overline{a(y)}\}$$

- Substraction: *for* $w \in \mathbb{M}^Y$*, the approximations and* $(sub_w)^\#_a : \mathbb{M}^{[Y]_a} \to \mathbb{M}^{[Y]_{sub_w(a)}}$*,* $(sub_w)^a_\# : \mathbb{M}^{[Y]^a} \to \mathbb{M}^{[Y]^{sub_w(a)}}$ *are*

$$(sub_w)^\#_a(Y') = \{y \in Y' \mid w(y) \sqsubseteq a(y)\} = Y'$$
$$(sub_w)^a_\#(Y') = \{y \in Y' \mid a(y) \ominus w(y) \sqsupset 0\}$$

Proof. We only consider the primal cases, the dual ones are analogous (maximum is dual to minimum, addition is dual to substraction).

Let $a \in \mathbb{M}^Y$.

- *Constant:* for all $0 \sqsubset \theta \sqsubseteq \delta_a$ and $Y' \subseteq [Y]_a$ we have

$$\begin{aligned}(c_k)^{\#}_{a,\theta}(Y') &= \gamma_{c_k(a),\theta} \circ c_k \circ \alpha_{a,\theta}(Y') \\ &= \{z \in [Z]_{c_k(a)} \mid \theta \sqsubseteq c_k(a \oplus \theta_{Y'})(z) \ominus c_k(a)(z)\} \\ &= \{z \in [Z]_{c_k(a)} \mid \theta \sqsubseteq k \ominus k\} = \{z \in Z \mid \theta \sqsubseteq 0\} = \varnothing\end{aligned}$$

  Hence all values $\iota^f_a(Y', z)$ are equal to 0 and we have $\iota^f_a = \delta_a$. Replacing θ by ι^f_a we obtain $(c_k)^{\#}_a(Y') = \varnothing$.

- *Reindexing:* for all $0 \sqsubset \theta \sqsubseteq \delta_a$ and $Y' \subseteq [Y]_a$ we have

$$\begin{aligned}(u^*)^{\#}_{a,\theta}(Y') &= \gamma_{u^*(a),\theta} \circ u^* \circ \alpha_{a,\theta}(Y') \\ &= \{z \in [Z]_{u^*(a)} \mid \theta \sqsubseteq (a \oplus \theta_{Y'})(u(z)) \ominus a(u(z))\}.\end{aligned}$$

  We show that this corresponds to $u^{-1}(Y') = \{z \in Z \mid u(z) \in Y'\}$. It is easy to see that for all $z \in u^{-1}(Y')$, we have

$$(a \oplus \theta_{Y'})(u(z)) \ominus a(u(z)) = \theta = a(u(z)) \ominus (a \ominus \theta_{Y'})(u(z))$$

  since $u(z) \in Y'$ and $\theta \sqsubseteq \delta_a$. Since $u(z) \in Y' \subseteq [Y]_a$, we have $u^*(a)(z) = a(u(z)) \neq 1$ and hence $z \in [Z]_{u^*(a)}$. On the other hand, for all $z \notin u^{-1}(Y')$, we have

$$(a \oplus \theta_{Y'})(u(z)) = a(u(z)) = (a \ominus \theta_{Y'})(u(z))$$

  since $u(z) \notin Y'$, and so

$$(a \oplus \theta_{Y'})(u(z)) \ominus a(u(z)) = a(u(z)) \ominus (a \ominus \theta_{Y'})(u(z)) = 0 \sqsubset \theta.$$

  Therefore $(u^*)^{\#}_{a,\theta}(Y') = u^{-1}(Y')$.

  We observe that for $Y' \subseteq [Y]_a$, $z \in [Z]_{u^*(a)}$ either $u^*(a \oplus \theta_{Y'})(z) \ominus u^*(a)(z) \sqsubset \theta$ for all $0 \sqsubset \theta \sqsubseteq \delta_a$ – and in this case $\iota^{u^*}_a(Y', z) = 0$ – or $u^*(a \oplus \theta_{Y'})(z) \ominus u^*(a)(z) = \theta$ for all $0 \sqsubset \theta \sqsubseteq \delta_a$ – and in this case $\iota^{u^*}_a(Y', z) = \delta_a$. By taking the minimum over all non-zero values, we get $\iota^{u^*}_a = \delta_a$.

  And finally we observe that $(u^*)^{\#}_a(Y') = (u^*)^{\#}_{a,\iota^{u^*}_a}(Y') = u^{-1}(Y')$.

- *Minimum:* let $0 \sqsubset \theta \sqsubseteq \delta_a$. For all $Y' \subseteq [Y]_a$ we have

$$\begin{aligned}(\min\nolimits_{\mathcal{R}})^{\#}_{a,\theta}(Y') &= \gamma_{\min_{\mathcal{R}}(a),\theta} \circ \min\nolimits_{\mathcal{R}} \circ \alpha_{a,\theta}(Y') \\ &= \{z \in [Z]_{\min_{\mathcal{R}}(a)} \mid \theta \sqsubseteq \min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \ominus \min_{y\mathcal{R}z} a(y)\}\end{aligned}$$

  We compute the value $V = \min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \ominus \min_{y\mathcal{R}z} a(y)$ and consider the following cases:

– Assume that there exists $\hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$ where $\hat{y} \notin Y'$.

  Then $(a \oplus \theta_{Y'})(\hat{y}) = a(\hat{y}) \sqsubseteq a(y) \sqsubseteq (a \oplus \theta_{Y'})(y)$ for all $y \in \mathcal{R}^{-1}(z)$, which implies that $\min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = a(\hat{y})$. We also have $\min_{y\mathcal{R}z} a(y) = a(\hat{y})$ and hence $V = 0$.

– Assume that $\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'$ and $\theta \sqsubseteq a(y) \ominus a(\hat{y})$ in all cases where $\hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$, $y \notin Y'$ and $y\mathcal{R}z$.

  Since $\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'$ we observe that

$$\min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \quad = \quad \min\{\min_{y\in\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)}(a(y) \oplus \theta), \min_{y\mathcal{R}z, y\notin Y'} a(y)\}$$

  We can omit the values of all y with $y\mathcal{R}z$, $y \notin \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$, $y \in Y'$, since we will never attain the minimum there.

  Now let $\hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$ and y with $y\mathcal{R}z$ and $y \notin Y'$. Then $\theta \sqsubseteq a(y) \ominus a(\hat{y})$ by assumption, which implies $a(\hat{y}) \oplus \theta \sqsubseteq a(y)$, since $a(\hat{y}) \sqsubseteq a(y)$ and Lemma 2.3.18(2) holds.

  From this we can deduce $\min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = a(\hat{y}) \oplus \theta$. We also have $\min_{y\mathcal{R}z} a(y) = a(\hat{y})$ and hence – since $a(\hat{y}) \sqsubseteq \overline{\theta}$ (due to $\theta \sqsubseteq \delta_a \sqsubseteq \overline{a(\hat{y})}$) and Lemma 2.3.18(9) holds – $V = (a(\hat{y}) \oplus \theta) \ominus a(\hat{y}) = \theta$.

– In the remaining case we have $\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'$ and there exist $\hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$, $y \notin Y'$, $y\mathcal{R}z$ such that $a(y) \ominus a(\hat{y}) \sqsubset \theta$.

  This implies $a(y) \sqsubseteq (a(y) \ominus a(\hat{y})) \oplus a(\hat{y}) \sqsubset \theta \oplus a(\hat{y})$ since again $a(\hat{y}) \sqsubseteq \overline{\theta}$ and Lemma 2.3.18(8) holds. Hence $\min_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \sqsubseteq a(y)$, which means that $V \sqsubseteq a(y) \ominus a(\hat{y}) \sqsubset \theta$.

Summarizing, for $\theta \sqsubseteq \delta_a$ we observe that $V = \theta$ if and only if $\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'$ and $\theta \sqsubseteq a(y) \ominus a(\hat{y})$ whenever $\hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$, $y \notin Y'$ and $y\mathcal{R}z$.

Hence if $\arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'$ we have

$$\iota_a^{\min_\mathcal{R}}(Y', z) = \min\{a(y) \ominus a(\hat{y}) \mid \hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y), y \notin Y', y\mathcal{R}z\} \cup \{\delta_a\}$$

otherwise $\iota_a^{\min_\mathcal{R}}(Y', z) = 0$.

The values above are minimal whenever $Y' = \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)$ and thus we have:

$$\iota_a^{\min_\mathcal{R}} = \min_{z\in[Z]_{\min_\mathcal{R}(a)}} \{a(y) \ominus a(\hat{y}) \mid y\mathcal{R}z, \hat{y} \in \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y), y \notin \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y)\} \cup \{\delta_a\}.$$

Finally we deduce that

$$(\min_\mathcal{R})_a^\#(Y') = (\min_\mathcal{R})_{a,\iota_a^{\min_\mathcal{R}}}^\#(Y') = \{z \in [Z]_{\min_\mathcal{R}(a)} \mid \arg\min_{y\in\mathcal{R}^{-1}(z)} a(y) \subseteq Y'\}.$$

- *Maximum:* let $0 \sqsubset \theta \sqsubseteq \delta_a$. For all $Y' \subseteq [Y]_a$ we have

$$\begin{aligned}(\max_{\mathcal{R}})^{\#}_{a,\theta}(Y') &= \gamma_{\max_{\mathcal{R}}(a),\theta} \circ \max_{\mathcal{R}} \circ \alpha_{a,\theta}(Y') \\ &= \{z \in [Z]_{\max_{\mathcal{R}}(a)} \mid \theta \sqsubseteq \max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \ominus \max_{y\mathcal{R}z} a(y)\}\end{aligned}$$

We observe that

$$\begin{aligned}\max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = \max\{ &\max_{y \in \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)} (a \oplus \theta_{Y'})(y), \\ &\max_{\substack{y\mathcal{R}z, y\in Y' \\ y \notin \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)}} (a(y) \oplus \theta)\}\end{aligned}$$

We can omit the values of all y with $y\mathcal{R}z$, $y \notin \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)$, $y \notin Y'$, since we will never attain the maximum there.

We now compute the value $V = \max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) \ominus \max_{y\mathcal{R}z} a(y)$ and consider the following cases:

  - Assume that there exists $\hat{y} \in \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)$ where $\hat{y} \in Y'$.

    Then $(a \oplus \theta_{Y'})(\hat{y}) = a(\hat{y}) \oplus \theta \sqsupseteq (a \oplus \theta_{Y'})(y) \sqsupseteq a(y)$ for all $y \in \mathcal{R}^{-1}(z)$, which implies that $\max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = a(\hat{y}) \oplus \theta$. We also have $\max_{y\mathcal{R}z} a(y) = a(\hat{y})$ and hence – since $a(\hat{y}) \sqsubseteq \overline{\theta}$ and Lemma 2.3.18(9) holds – $V = (a(\hat{y}) \oplus \theta) \ominus a(\hat{y}) = \theta$.

  - Assume that $\arg\max_{y\in\mathcal{R}^{-1}(z)} a(y) \cap Y' = \emptyset$. Now let $\hat{y} \in \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)$ and $y \notin \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)$ with $y\mathcal{R}z$ and $y \in Y'$. Then

$$\begin{aligned}\max_{y \in \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)} (a \oplus \theta_{Y'})(y) &= a(\hat{y}) \\ \max_{\substack{y\mathcal{R}z, y\in Y' \\ y \notin \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)}} (a(y) \oplus \theta) &= a(y') \oplus \theta\end{aligned}$$

    for some value y' with $y'\mathcal{R}z$, $y' \in Y'$, $y' \notin \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y)$, that is $a(y') \sqsubset a(\hat{y})$.

    So then either $\max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = a(\hat{y})$ and $V = a(\hat{y}) \ominus a(\hat{y}) = 0$. Or $\max_{y\mathcal{R}z}(a \oplus \theta_{Y'})(y) = a(y') \oplus \theta$ and by Lemma 2.3.18(11) $V = (a(y') \oplus \theta) \ominus a(\hat{y}) \sqsubset \theta$.

Summarizing, for $\theta \sqsubseteq \delta_a$ we observe that $V = \theta$ if and only if $\arg\max_{y\in\mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset$, where the latter condition is independent of θ.

Hence, as in the case of reindexing, we have $\iota_a^{\max_{\mathcal{R}}} = \delta_a$. Finally we have

$$(\max_{\mathcal{R}})^{\#}_a(Y') = (\max_{\mathcal{R}})^{\#}_{a,\iota_a^{\max_{\mathcal{R}}}}(Y') = \{z \in [Z]_{\max_{\mathcal{R}}(a)} \mid \arg\max_{y\in\mathcal{R}^{-1}(z)} a(y) \cap Y' \neq \emptyset\}.$$

- *Average:* for all $0 \sqsubset \theta \sqsubseteq \delta_a$ and $Y' \subseteq [Y]_a$ by definition

$$
\begin{aligned}
(\mathrm{av}_D)^{\#}_{a,\theta}(Y') &= \gamma_{\mathrm{av}_D(a),\theta} \circ \mathrm{av}_D \circ \alpha_{a,\theta}(Y') \\
&= \{p \in [D]_{\mathrm{av}_D(a)} \mid \theta \sqsubseteq \bigoplus_{y\in Y} p(y) \odot (a \oplus \theta_{Y'})(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)\}
\end{aligned}
$$

We show that this set corresponds to $\{p \in [D]_{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\}$.

Consider $p \in [D]_{\mathrm{av}_D(a)}$ such that $supp(p) \subseteq Y'$. Note that clearly $\bigoplus_{y\in Y'} p(y) = 1$. Now we have

$$
\begin{aligned}
&\bigoplus_{y\in Y} p(y) \odot (a \oplus \theta_{Y'})(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y) \\
&= \bigoplus_{y\in Y'} p(y) \odot (a(y) \oplus \theta) \oplus \bigoplus_{y\in Y\setminus Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y) \\
&= \bigoplus_{y\in Y'} (p(y) \odot a(y) \oplus p(y) \odot \theta) \oplus \bigoplus_{y\in Y\setminus Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y) \\
&\qquad \text{[by weak distributivity, since for } y \in Y' \subseteq [Y]_a,\ a(y) \sqsubseteq \overline{\delta_a}] \\
&= \bigoplus_{y\in Y'} p(y) \odot \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \oplus \bigoplus_{y\in Y\setminus Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y) \\
&= \bigoplus_{y\in Y'} p(y) \odot \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y'} p(y) \odot a(y) \\
&\qquad \text{[since, for } y \notin Y' \supseteq supp(p),\ p(y) = 0 \text{ and thus } p(y) \odot a(y) = 0] \\
&= (\bigoplus_{y\in Y'} p(y)) \odot \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y'} p(y) \odot a(y) \\
&\qquad \text{[by weak distributivity, since } p \text{ is a distribution]} \\
&= 1 \odot \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y'} p(y) \odot a(y) \\
&\qquad \text{[since } p \text{ is a distribution]} \\
&= \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y'} p(y) \odot a(y) \\
&= \theta
\end{aligned}
$$

In order to motivate the last passage, observe that for all $y \in Y' \subseteq [Y]_a$, we have $a(y) \sqsubseteq \overline{\delta_a}$, and thus $\bigoplus_{y\in Y'} p(y) \odot a(y) \sqsubseteq \bigoplus_{y\in Y'} p(y) \odot \overline{\delta_a} = (\bigoplus_{y\in Y'} p(y)) \odot \overline{\delta_a} = 1 \odot \overline{\delta_a} = \overline{\delta_a}$, where the third last passage is motivated by weak distributivity. Since $\theta \sqsubseteq \delta_a$, by Lemma 2.3.18(3), we have $\overline{\delta_a} \sqsubseteq \overline{\theta}$ and thus $\bigoplus_{y\in Y'} p(y) \odot a(y) \sqsubseteq \overline{\theta}$. In turn, using this fact, Lemma 2.3.18(9) motivates the last equality in the chain above, i.e., $\theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y'} p(y) \odot a(y) = \theta$.

On the other hand, for all $p \in [D]_{\mathrm{av}_D(a)}$ such that $supp(p) \nsubseteq Y'$, there exists $y' \in Y\setminus Y'$ such that $p(y') \neq 0$. Then, we have

$$
\bigoplus_{y\in Y} p(y) \odot (a \oplus \theta_{Y'})(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)
$$

$$= \bigoplus_{y\in Y'} p(y) \odot (a(y) \oplus \theta) \oplus \bigoplus_{y\in Y\setminus Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)$$

$$= \bigoplus_{y\in Y'} p(y) \odot \theta \oplus \bigoplus_{y\in Y'} p(y) \odot a(y) \oplus \bigoplus_{y\in Y\setminus Y'} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)$$

[by weak distributivity, since for $y \in Y' \subseteq [Y]_a$, $a(y) \sqsubseteq \overline{\delta_a}$]

$$= \bigoplus_{y\in Y'} p(y) \odot \theta \oplus \bigoplus_{y\in Y} p(y) \odot a(y) \ominus \bigoplus_{y\in Y} p(y) \odot a(y)$$

$$\sqsubseteq \bigoplus_{y\in Y'} p(y) \odot \theta$$

[by Lemma 2.3.18(6)]

$$= \theta \odot \bigoplus_{y\in Y'} p(y)$$

[by weak distributivity, since p is a distribution]

$$\sqsubset \theta$$

In order to motivate the last inequality, we proceed as follows. We have that $supp(p) \nsubseteq Y'$. Let $y_0 \in supp(p)\setminus Y'$. We know that $\overline{p(y_0)} \sqsubseteq \bigoplus_{y\in Y\setminus\{y\}} p(y) \sqsubseteq \bigoplus_{y\in Y'} p(y)$. Therefore $\overline{\bigoplus_{y\in Y'} p(y)} \sqsubseteq p(y_0) \neq 0$. Hence $\bigoplus_{y\in Y'} p(y) \sqsubset 1$.

The strict inequality above now follows, if we further show that given an $x \in \mathbb{M}$, $x \neq 1$ then $\theta \odot x \sqsubset \theta$. Note that $\overline{x} \neq 0$. Therefore $\theta = \theta \odot 1 = \theta \odot (x \oplus \overline{x}) = \theta \odot x \oplus \theta \odot \overline{x}$, where the last equality follows by weak distributivity. Now $\theta \odot \overline{x} \sqsubseteq \overline{x} \sqsubseteq \overline{\theta \odot x}$, and thus, by Lemma 2.3.18(9), we obtain $\theta \odot x = \theta \odot x \oplus \theta \odot \overline{x} \ominus \theta \odot \overline{x} = \theta \ominus \theta \odot \overline{x} \sqsubset \theta$, as desired. The last passage follows by the fact that $\theta, \overline{x} \neq 0$ and thus $\theta \odot \overline{x} \neq 0$.

Since these results hold for all $\theta \sqsubseteq \delta_a$, we have $\iota_a^{\mathrm{av}_D} = \delta^a$.

And finally $(\mathrm{av}_D)^{\#}_{a,\theta}(Y') = (\mathrm{av}_D)_{\#}^{a,\theta}(Y') = \{p \in [D]_{\mathrm{av}_D(a)} \mid supp(p) \subseteq Y'\}$.

- *Addition:* let $0 \sqsubset \theta \sqsubseteq \delta_a$. For all $Y' \subseteq [Y]_a$ we have

$$(add_w)^{\#}_{a,\theta}(Y') = \{y \in [Y]_{add_w(a)} \mid add_w(a \oplus \theta_{Y'})(y) \ominus add_w(a)(y) \sqsupseteq \theta\}.$$

For $y \notin Y'$, we have

$$add_w(a \oplus \theta_{Y'})(y) \ominus add_w(a)(y) = 0 \sqsubset \theta$$

For $y \in Y'$, we have

$$add_w(a \oplus \theta_{Y'})(y) \ominus add_w(a)(y) = (\theta \oplus (a(y) \oplus w(y))) \ominus (a(y) \oplus w(y)) = \theta$$

if and only if (Lemma 2.3.18(9))

$$(a(y) \oplus w(y)) \sqsubseteq \overline{\theta} \ (\Leftrightarrow \overline{(a(y) \oplus w(y))} \sqsupseteq \theta).$$

Now $y \in [Y]_{add_w(a)}$ if and only if $a(y) \oplus w(y) \sqsubset 1$. Therefore, for sufficiently small $\theta \sqsupset 0$,

$$(a(y) \oplus w(y)) \sqsubseteq \overline{\theta}$$

holds for any $y \in [Y']_{add_w(a)}$. To conclude

$$(add_w)^\#_a(Y') = [Y']_{add_w(a)} = \{y \in Y' \mid a(y) \oplus w(y) \sqsubset 1\}$$

and

$$\iota^{add_w}_a = \min_{y \in Y} \{\overline{a(y) \oplus w(y)} \mid a(y) \oplus w(y) \sqsubset 1\}.$$

- *Substraction:* let $0 \sqsubset \theta \sqsubseteq \delta_a$. For all $Y' \subseteq [Y]_a$ we have

$$(sub_w)^\#_{a,\theta}(Y') = \{y \in [Y]_{sub_w(a)} \mid sub_w(a \oplus \theta_{Y'})(y) \ominus sub_w(a)(y) \sqsupseteq \theta\}.$$

For $y \notin Y'$, we have

$$sub_w(a \oplus \theta_{Y'})(y) \ominus sub_w(a)(y) = 0 \sqsubset \theta$$

For $y \in Y'$, we have

$$\begin{aligned} sub_w(a \oplus \theta_{Y'})(y) \ominus sub_w(a)(y) &= ((a(y) \oplus \theta) \ominus w(y)) \ominus (a(y) \ominus w(y)) \\ &= \overline{\overline{(a(y) \oplus \theta)} \oplus (w(y) \oplus (a(y) \ominus w(y)))} \end{aligned}$$

Now,

$$\overline{\overline{(a(y) \oplus \theta)} \oplus (w(y) \oplus (a(y) \ominus w(y)))} = \overline{\overline{(a(y) \oplus \theta)} \oplus a(y)}$$

if and only if $w(y) \sqsubseteq a(y)$ (Lemma 2.3.18(2)). We continue

$$\overline{\overline{(a(y) \oplus \theta)} \oplus a(y)} = (a(y) \oplus \theta) \ominus a(y) = \theta$$

if and only if $a(y) \sqsubseteq \overline{\theta}$ (Lemma 2.3.18(9)). This immediately holds since $\theta \sqsubseteq \delta_a$.

To summarize (note that $[Y]_{sub_w(a)} \subseteq [Y]_a$ since $w(y) \in \mathbb{M}$ for all $y \in Y$)

$$(sub_w)^\#_a(Y') = \{y \in Y' \mid w(y) \sqsubseteq a(y)\}$$

and $\iota^{sub_w}_a = \delta_a$.

□

Corollary A.2.6 (Corollary 3.5.9). *We consider the basic functions from Definition 3.5.3, function composition as in Lemma 3.5.1 and disjoint union as in Proposition 3.5.2 and give the corresponding values for* ι_a^f *and* ι_f^a.
For greatest fixpoints (primal case) we obtain:

- $\iota_a^{c_k} = \iota_a^{u^*} = \iota_a^{\max_\mathcal{R}} = \iota_a^{\mathrm{av}_D} = \iota_a^{sub_w} = \delta_a$
- $\iota_a^{\min_\mathcal{R}} = \min\limits_{z \in [Z]_{\min_\mathcal{R}(a)}} \{a(y) \ominus a(\hat{y}) \mid y\mathcal{R}z, y \notin \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y), \hat{y} \in \arg\min_{y \in \mathcal{R}^{-1}(z)} a(y)\} \cup \{\delta_a\}$
- $\iota_a^{add_w} = \min_{y \in Y} \{\overline{a(y) \oplus w(y)} \mid a(y) \oplus w(y) \sqsubset 1\}$
- $\iota_a^{g \circ f} \sqsupseteq \min\{\iota_a^f, \iota_{f(a)}^g\}$
- $\iota_a^{\biguplus_{i \in I} f_i} = \min_{i \in I} \iota_{a|_{Y_i}}^{f_i}$

For least fixpoints (dual case) we obtain:

- $\iota_{c_k}^a = \iota_{u^*}^a = \iota_{\min_\mathcal{R}}^a = \iota_{\mathrm{av}_D}^a = \iota_{add_w}^a = \delta^a$
- $\iota_{\max_\mathcal{R}}^a = \min\limits_{z \in [Z]^{\min_\mathcal{R}(a)}} \{a(\hat{y}) \ominus a(y) \mid y\mathcal{R}z, \hat{y} \in \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y), y \notin \arg\max_{y \in \mathcal{R}^{-1}(z)} a(y)\} \cup \{\delta^a\}$
- $\iota_{sub_w}^a = \min_{y \in Y} \{a(y) \ominus w(y) \mid a(y) \ominus w(y) \sqsupset 0\}$
- $\iota_{g \circ f}^a \sqsupseteq \min\{\iota_f^a, \iota_g^{f(a)}\}$
- $\iota_{\biguplus_{i \in I} f_i}^a = \min_{i \in I} \iota_{f_i}^{a|_{Y_i}}$

Proof. The values ι_a^f can be obtained by inspecting the proofs of Propositions 3.5.5, 3.5.6 and 3.5.2.

It only remains to show that $\iota := \iota_a^{\biguplus_{i \in I} f_i} \sqsubseteq \min_{i \in I} \iota_{a|_{Y_i}}^{f_i}$ (cf. Proposition 3.5.2), which means showing $\iota \sqsubseteq \iota_{a|_{Y_i}}^{f_i}$ for every $i \in I$. We abbreviate $\iota_i := \iota_{a|_{Y_i}}^{f_i}$.

If $\iota \sqsupset \iota_i$ for some $i \in I$, we will find a $z \in [Z_i]_{f_i(a)}$ and $Y' \subseteq [Y]_a$, such that $z \in (f_i)^\#_{a|_{Y_i}, \iota_i}(Y' \cap Y_i) = (f_i)^\#_{a|_{Y_i}}(Y' \cap Y_i)$ but $z \notin (f_i)^\#_{a|_{Y_i}, \iota}(Y' \cap Y_i)$ by definition (cf. Lemma 3.2.9). This is a contradiction since

$$z \in \biguplus_{i \in I} (f_i)^\#_{a|_{Y_i}}(Y' \cap Y_i) = \Big(\biguplus_{i \in I} f_i\Big)^\#_a(Y') = \Big(\biguplus_{i \in I} f_i\Big)^\#_{a,\iota}(Y') = \biguplus_{i \in I} (f_i)^\#_{a|_{Y_i}, \iota}(Y' \cap Y_i)$$

and since $z \in Z_i$, $z \notin (f_i)^\#_{a|_{Y_i}, \iota}(Y' \cap Y_i)$ and cannot be contained in the union.

The arguments for the values ι_f^a in the dual case are analogous. □

Lemma A.2.7 (Lemma 3.6.2)**.** *Given a function $t: S \to [0,1]$, the t-approximation for $\mathcal{T}$ in the dual sense is $\mathcal{T}_{\#}^{t}: \mathcal{P}([S]^{t}) \to \mathcal{P}([S]^{\mathcal{T}(t)})$ with*

$$\mathcal{T}_{\#}^{t}(S') = \{s \in [S]^{\mathcal{T}(t)} \mid s \notin T \wedge supp(\eta(s)) \subseteq S'\}.$$

Proof. In the following let $t: S \to [0,1]$ and $S' \subseteq [S]^{t}$. By Lemma 3.6.1 we know that $\mathcal{T} = (\eta^{*} \circ \mathrm{av}_{D}) \uplus c_{k}$, then by Propositions 3.5.7, 3.5.6, and 3.5.5 we have

$$\begin{aligned}
\mathcal{T}_{\#}^{t}(S') &= ((\eta^{*} \circ \mathrm{av}_{D}) \uplus c_{k})_{\#}^{t}(S') \\
&= (\eta^{*} \circ \mathrm{av}_{D})_{\#}^{t}(S') \cup (c_{k})_{\#}^{t}(S') \\
&= (\eta^{*})_{\#}^{\mathrm{av}_{D}(t)} \circ (\mathrm{av}_{D})_{\#}^{t}(S') \cup (c_{k})_{\#}^{t}(S') \\
&= \{s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))} \mid \eta(s) \in \{q \in [D]^{\mathrm{av}_{D}(t)} \mid supp(q) \subseteq S'\}\} \cup \varnothing \\
&= \{s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))} \mid \eta(s) \in [D]^{\mathrm{av}_{D}(t)} \wedge supp(\eta(s)) \subseteq S'\}
\end{aligned}$$

Observe that actually for all $s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))}$ it always holds that $\eta(s) \in [D]^{\mathrm{av}_{D}(t)}$. In fact, since $s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))}$ we must have that $\eta^{*}(\mathrm{av}_{D}(t))(s) = \mathrm{av}_{D}(t)(\eta(s)) \neq 0$, and thus $\eta(s) \in \{q \in D \mid \mathrm{av}_{D}(t)(q) \neq 0\} = [D]^{\mathrm{av}_{D}(t)}$. Therefore, we have that

$$\begin{aligned}
&\{s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))} \mid \eta(s) \in [D]^{\mathrm{av}_{D}(t)} \wedge supp(\eta(s)) \subseteq S'\} \\
&= \{s \in [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))} \mid supp(\eta(s)) \subseteq S'\}
\end{aligned}$$

Finally, the set above is the same as

$$\{s \in [S]^{\mathcal{T}(t)} \mid s \notin T \wedge supp(\eta(s)) \subseteq S'\} = \{s \in [S\backslash T]^{\mathcal{T}(t)} \mid supp(\eta(s)) \subseteq S'\}$$

because, for all $s \in S\backslash T$, hence $s \notin T$, we have that $\mathcal{T}(t)(s) = \sum_{s' \in S} \eta(s)(s') \cdot t(s') = \eta^{*}(\mathrm{av}_{D}(t))(s)$, and so $[S\backslash T]^{\mathcal{T}(t)} = [S\backslash T]^{\eta^{*}(\mathrm{av}_{D}(t))}$. □

Lemma A.2.8 (Lemma 3.6.7)**.** *Let $d: X \times X \to [0,1]$. The approximation for the Kantorovich lifting $\mathcal{K}$ in the dual sense is $\mathcal{K}_{\#}^{d}: \mathcal{P}([X \times X]^{d}) \to \mathcal{P}([D \times D]^{\mathcal{K}(d)})$ with*

$$\begin{aligned}
\mathcal{K}_{\#}^{d}(M) \quad = \quad &\{(p,q) \in [D \times D]^{\mathcal{K}(d)} \mid \exists \omega \in \Omega(p,q), supp(\omega) \subseteq M, \\
&\sum_{u,v \in S} d(u,v) \cdot \omega(u,v) = \mathcal{K}(d)(p,q)\}.
\end{aligned}$$

Proof. Let $d: X \times X \to [0,1]$ and $M \subseteq [X \times X]^{d}$. Then we have:

$$\mathcal{K}_{\#}^{d}(M) \quad = \quad (\min_{u})_{\#}^{\mathrm{av}_{VP_{D}}(d)}((\mathrm{av}_{VP_{D}})_{\#}^{d}(M))$$

where

$$(\mathrm{av}_{VP_{D}})_{\#}^{d}: \mathcal{P}([X \times X]^{d}) \to \mathcal{P}([VP_{D}]^{\mathrm{av}_{VP_{D}}(d)})$$

$$(\min_u)_{\#}^{\mathrm{av}_{VP_D}(d)}: \mathcal{P}([VP_D]^{\mathrm{av}_{VP_D}(d)}) \to \mathcal{P}([D \times D]^{\mathcal{K}(d)})$$

We are using the approximations associated to non-expansive functions, given in Proposition 3.5.5, and obtain:

$$\begin{aligned}
\mathcal{K}_{\#}^{d}(M) &= \{(p,q) \in [D \times D]^{\mathcal{K}(d)} \mid \arg\min_{\omega \in u^{-1}(p,q)} \mathrm{av}_{VP_D}(d)(\omega) \cap (\mathrm{av}_{VP_D})_{\#}^{d}(M) \neq \emptyset\} \\
&= \{(p,q) \in [D \times D]^{\mathcal{K}(d)} \mid \exists \omega \in \Omega(p,q), \omega \in (\mathrm{av}_{VP_D})_{\#}^{d}(M), \\
&\qquad \mathrm{av}_{VP_D}(d)(\omega) = \min_{\omega' \in \Omega(p,q)} \mathrm{av}_{VP_D}(d)(\omega')\} \\
&= \{(p,q) \in [D \times D]^{\mathcal{K}(d)} \mid \exists \omega \in \Omega(p,q), \omega \in (\mathrm{av}_{VP_D})_{\#}^{d}(M), \\
&\qquad \mathrm{av}_{VP_D}(d)(\omega) = \mathcal{K}(d)(p,q)\} \\
&= \{(p,q) \in [D \times D]^{\mathcal{K}(d)} \mid \exists \omega \in \Omega(p,q), \mathit{supp}(\omega) \subseteq M, \\
&\qquad \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) = \mathcal{K}(d)(p,q)\}
\end{aligned}$$

□

Lemma A.2.9 (Lemma 3.6.10)**.** *Let $d: S \times S \to [0,1]$. The approximation of Δ in the dual sense is $\Delta_{\#}^{d}: \mathcal{P}([S \times S]^{d}) \to \mathcal{P}([S \times S]^{\Delta(d)})$ with*

$$\Delta_{\#}^{d}(M) = \{(s,t) \in [S \times S]^{\Delta(d)} \mid \ell(s) = \ell(t) \wedge (\eta(s), \eta(t)) \in \mathcal{K}_{\#}^{d}(M)\}.$$

Proof. First, $(c_k)_{\#}^{d}(M) = \emptyset$. Now, since $(\eta \times \eta)_{\#}^{\mathcal{K}(d)} = (\eta \times \eta)^{-1}$, we have

$$(s,t) \in (\eta \times \eta)_{\#}^{\mathcal{K}(d)} \circ \mathcal{K}_{\#}^{d}(M) \Leftrightarrow (\eta(s), \eta(t)) \in \mathcal{K}_{\#}^{d}(M).$$

Lastly,

$$\begin{aligned}
\Delta_{\#}^{d}(M) &= (\max_\rho)_{\#}^{((\eta\times\eta)^* \circ \mathcal{K} \uplus c_k)(d)} \circ ((\eta \times \eta)^* \circ \mathcal{K} \uplus c_k)_{\#}^{d}(M) \\
&= (\max_\rho)_{((\eta\times\eta)^* \circ \mathcal{K} \uplus c_k)(d)}^{\#} \circ (((\eta \times \eta)^* \circ \mathcal{K})_{\#}^{d}(M) \uplus (c_k)_{\#}^{d}(M)) \\
&= \{(s,t) \in [S \times S]^{\Delta(d)} \mid \arg\max_{y \in \rho^{-1}(s,t)} ((\eta \times \eta)^* \circ \mathcal{K} \uplus c_k)(d)(y) \\
&\qquad \subseteq ((\eta \times \eta)^* \circ \mathcal{K})_{\#}^{d}(M) \times \{0\}\}
\end{aligned}$$

Recalling that $\rho^{-1}(s,t) = \{((s,t),0), ((s,t),1)\}$, the inclusion

$$\arg\max_{y \in \rho^{-1}(s,t)} ((\eta \times \eta)^* \circ \mathcal{K} \uplus c_k)(d)(y) \subseteq ((\eta \times \eta)^* \circ \mathcal{K})_{\#}^{d}(M) \times \{0\}$$

can only hold if $(\eta \times \eta)^* \circ \mathcal{K}(d)(s,t) > c_k(d)(s,t)$ (and hence the maximum is achieved by $(\eta \times \eta)^* \circ \mathcal{K}(d)$ instead of $c_k(d)$) which holds if and only if $\ell(s) \neq \ell(t)$ (for $(s,t) \in [S \times S]^{\Delta(d)}$) and additionally $((s,t),0) \in ((\eta \times \eta)^* \circ \mathcal{K})_{\#}^{d}(M) \times \{0\}$. Thus

$$\Delta_{\#}^{d}(M) = \{(s,t) \in [S \times S]^{\Delta(d)} \mid c_k(d)(s,t) < (\eta \times \eta)^* \circ \mathcal{K}(d)(s,t)$$

$$\begin{aligned}&\wedge\ (s,t) \in ((\eta\times\eta)^* \circ \mathcal{K})^d_\#(M)\}\\ &= \{(s,t)\in [S\times S]^{\Delta(d)} \mid \ell(s)=\ell(t) \wedge (\eta(s),\eta(t)) \in \mathcal{K}^d_\#(M)\}\end{aligned}$$

□

Lemma A.2.10 (Lemma 3.6.14)**.** *The approximation for the Hausdorff lifting $\mathcal{H}$ in the dual sense is as follows. Let $d\colon X\times X\to\mathbb{M}$, then $\mathcal{H}^d_\#\colon\mathcal{P}([X\times X]^d)\to\mathcal{P}([\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)})$ with*

$$\begin{aligned}\mathcal{H}^d_\#(R) \quad = \quad & \{(X_1,X_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)} \mid \\ & \forall x_1\in X_1\big(\min_{x_2'\in X_2} d(x_1,x_2') = \mathcal{H}(d)(X_1,X_2) \Rightarrow \exists x_2\in X_2\colon \\ & \qquad (x_1,x_2)\in R\wedge d(x_1,x_2)=\mathcal{H}(d)(X_1,X_2)\big)\wedge \\ & \forall x_2\in X_2\big(\min_{x_1'\in X_1} d(x_1',x_2) = \mathcal{H}(d)(X_1,X_2) \Rightarrow \exists x_1\in X_1\colon \\ & \qquad (x_1,x_2)\in R\wedge d(x_1,x_2)=\mathcal{H}(d)(X_1,X_2)\big)\}\end{aligned}$$

Proof. Let $d\colon X\times X\to\mathbb{M}$ and $R\subseteq[X\times X]^d$. Then we have:

$$\mathcal{H}^d_\#(R) \quad = \quad (\min_u)^{\max_\in(d)}_\#((\max_\in)^d_\#(R))$$

where

$$\begin{aligned}(\max_\in)^d_\#\colon&\mathcal{P}([X\times X]^d)\to\mathcal{P}([\mathcal{P}(X\times X)]^{\max_\in(d)})\\ (\min_u)^{\max_\in(d)}_\#\colon&\mathcal{P}([\mathcal{P}(X\times X)]^{\max_\in(d)})\to\mathcal{P}([\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)})\end{aligned}$$

We are using the approximations associated to non-expansive functions, given in Proposition 3.5.5, and obtain:

$$\begin{aligned}\mathcal{H}^d_\#(R) \quad = \quad & \{(X_1,X_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)} \mid \arg\min_{C\in u^{-1}(X_1,X_2)} \max_\in(d)(C) \\ & \qquad \cap(\max_\in)^d_\#(R)\neq\emptyset\} \\ = \quad & \{(X_1,X_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)} \mid \exists C\subseteq X\times X, u(C)=(X_1,X_2), \\ & \quad C\in(\max_\in)^d_\#(R), \max_\in(d)(C) = \min_{u(C')=(X_1,X_2)} \max_\in(d)(C')\} \\ = \quad & \{(X_1,X_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)} \mid \exists C\subseteq X\times X, u(C)=(X_1,X_2), \\ & \quad C\in(\max_\in)^d_\#(R), \max d[C] = \min_{u(C')=(X_1,X_2)} \max d[C']\} \\ = \quad & \{(X_1,X_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]^{\mathcal{H}(d)} \mid \exists C\subseteq X\times X, u(C)=(X_1,X_2), \\ & \quad \arg\max_{(y_1,y_2)\in C} d(y_1,y_2)\subseteq R, \max d[C]=\mathcal{H}(d)(X_1,X_2)\}\end{aligned}$$

We show that this is equivalent to the characterisation in the statement of the lemma.

- Assume that for all $x_1 \in X_1$ such that $\min_{x_2' \in X_2} d(x_1, x_2') = \mathcal{H}(d)(X_1, X_2)$, there exists $x_2 \in X_2$ such that $(x_1, x_2) \in R$ and $d(x_1, x_2) = \mathcal{H}(d)(X_1, X_2)$ (and vice versa).

  We define a set C_m that contains all such pairs (x_1, x_2), obtained from this guarantee. Now let $x_1 \notin \pi_1[C_m]$. Then necessarily $\min_{x_2' \in X_2} d(x_1, x_2') < \mathcal{H}(d)(X_1, X_2)$ (because the minimal distance to an element of X_2 cannot exceed the Hausdorff distance of the two sets). Construct another set C' that contains all such (x_1, x_2) where x_2 is an argument where the minimum is obtained. Also add elements $x_2 \notin \pi_2[C_m]$ and their corresponding partners to C'.

  The $C = C_m \cup C'$ is a coupling for X_1, X_2, i.e., $u(C) = (X_1, X_2)$. Furthermore $\arg\max_{(y_1,y_2) \in C} d(y_1, y_2) = C_m \subseteq R$ and $\max d[C] = \max d[C_m] = \mathcal{H}(d)(X_1, X_2)$.

- Assume that there exists $C \subseteq X \times X$, $u(C) = (X_1, X_2)$, $\arg\max_{(y_1,y_2) \in C} d(y_1, y_2) \subseteq R$, $\max d[C] = \mathcal{H}(d)(X_1, X_2)$.

  Now let $x_1 \in X_1$ such that $\min_{x_2' \in X_2} d(x_1, x_2') = \mathcal{H}(d)(X_1, X_2)$. Since C is a coupling of X_1, X_2, there exists $x_2 \in X_2$ such that $(x_1, x_2) \in C \subseteq R$. It is left to show that $d(x_1, x_2) = \mathcal{H}(d)(X_1, X_2)$, which can be done as follows:

  $$\mathcal{H}(d)(X_1, X_2) = \min_{x_2' \in X_2} d(x_1, x_2') \leq d(x_1, x_2) \leq \max d[C] = \mathcal{H}(d)(X_1, X_2).$$

  For an $x_2 \in X_2$ such that $\min_{x_1' \in X_1} d(x_1', x_2) = \mathcal{H}(d)(X_1, X_2)$ the proof is analogous.

□

Lemma A.2.11 (Lemma 3.6.17)**.** *Let $d: S \times S \to [0,1]$. The approximation for $\mathcal{M}$ in the dual sense is $\mathcal{M}_\#^d : \mathcal{P}([S \times S]^d) \to \mathcal{P}([S \times S]^{\mathcal{M}(d)})$ with*

$$\mathcal{M}_\#^d(X) = \{(s,t) \in [S \times S]^{\mathcal{M}(d)} \mid \; d_L(\ell(s), \ell(t)) < \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t)) \\ \wedge\ (\eta(s), \eta(t)) \in \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d(X)\}$$

Proof. Let $d: S \times S \to [0,1]$ and $X \subseteq [S \times S]^d$. We abbreviate $g = (\eta \times \eta)^* \circ \mathcal{H} \circ \mathcal{K}: [0,1]^{S \times S} \to [0,1]^{S \times S}$ and $j = (\ell \times \ell)^* \circ c_{d_L}: [0,1]^{S \times S} \to [0,1]^{S \times S}$, so that $\mathcal{M} = \max_\rho \circ (g \uplus j)$. Thus we obtain

$$\mathcal{M}_\#^d(X) = (\max_\rho)_\#^{(g \uplus j)(d)} \circ (g \uplus j)_\#^d(X)$$

Since $c_{d_L}: [0,1]^{S \times S} \to [0,1]^{L \times L}$ is a constant function and $((\ell \times \ell)^*)_\#^{c_{d_L}(d)} = (\ell \times \ell)^{-1}$, we deduce that

$$j_\#^d(X) = ((\ell \times \ell)^*)_\#^{c_{d_L}(d)} \circ (c_{d_L})_\#^d(X) = (\ell \times \ell)^{-1}(\emptyset) = \emptyset$$

On the other hand

$$g_\#^d = ((\eta \times \eta)^*)_\#^{\mathcal{H}(\mathcal{K}(d))} \circ \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d$$

where

$$\mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d : \mathcal{P}([S \times S]^d) \to \mathcal{P}([\mathcal{P}(S) \times \mathcal{P}(S)]^{\mathcal{H}(\mathcal{K}(d))})$$

$$((\eta \times \eta)^*)_\#^{\mathcal{H}(\mathcal{K}(d))} : \mathcal{P}([\mathcal{P}(S) \times \mathcal{P}(S)]^{\mathcal{H}(\mathcal{K}(d))}) \to \mathcal{P}([S \times S]^{g(d)})$$

We recall that $((\eta \times \eta)^*)_\#^{\mathcal{H}(\mathcal{K}(d))} = (\eta \times \eta)^{-1}$, and hence

$$(s,t) \in g_\#^d(X) \Leftrightarrow (\eta(s), \eta(t)) \in \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d(X)$$

Lastly, we obtain

$$\begin{aligned}\mathcal{M}_\#^d(X) &= (\max_\rho)_\#^{(g \uplus j)(d)} \circ (g \uplus j)_\#^d(X) \\ &= (\max_\rho)_\#^{(g \uplus j)(d)}(g_\#^d(X) \uplus j_\#^d(X)) \\ &= \{(s,t) \in [S \times S]^{\mathcal{M}(d)} \mid \arg\max_{y \in \rho^{-1}(s,t)} (g \uplus j)(d)(y) \subseteq g_\#^d(X) \times \{0\}\}\end{aligned}$$

Recalling that $\rho^{-1}(s,t) = \{((s,t),0),((s,t),1)\}$, the inclusion

$$\arg\max_{y \in \rho^{-1}(s,t)} (g \uplus j)(d)(y) \subseteq g_\#^d(X) \times \{0\}$$

can only hold if $g(d)(s,t) > j(d)(s,t)$ (and hence the maximum is achieved by $g(d)$ instead of $j(d)$) and additionally $((s,t),0) \in g_\#^d(X) \times \{0\}$. Thus

$$\begin{aligned}\mathcal{M}_\#^d(X) &= \{(s,t) \in [S \times S]^{\mathcal{M}(d)} \mid j(d)(s,t) < g(d)(s,t) \ \wedge \ (s,t) \in g_\#^d(X)\} \\ &= \{(s,t) \in [S \times S]^{\mathcal{M}(d)} \mid d_L(\ell(s), \ell(t)) < \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t)) \\ &\qquad \wedge \ (\eta(s), \eta(t)) \in \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d(X)\}\end{aligned}$$

□

Proposition A.2.12 (Proposition 3.6.20). *Let $d: S \times S \to [0,1]$ where $d = \mathcal{M}(d)$. Then $\mathcal{M}_\#^d : \mathcal{P}([S \times S]^d) \to \mathcal{P}([S \times S]^d)$, where $[S \times S]^d = \{(s,t) \in S \times S \mid d(s,t) > 0\}$.*
Then M is a self-closed relation with respect to d if and only if $M \subseteq [S \times S]^d$ and M is a post-fixpoint of $\mathcal{M}_\#^d$.

Proof. First note that whenever M is self-closed, it holds that $d(s,t) > 0$ for all $(s,t) \in M$ and hence $M \subseteq [S \times S]^d$.

Observe that we would have $d_L(\ell(s), \ell(t)) = 1 \geq \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t))$ whenever $\ell(s) \neq \ell(t)$. On the other hand, when $\ell(s) = \ell(t)$, instead, we have $d_L(\ell(s), \ell(t)) = 0 < \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t))$, since $\mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t)) > 0$ for all $(s,t) \in [S \times S]^d$. So, by Lemma 3.6.17, we obtain that

$$\begin{aligned}\mathcal{M}_\#^d(M) &= \{(s,t) \in [S \times S]^d \mid d_L(\ell(s), \ell(t)) < \mathcal{H}(\mathcal{K}(d))(\eta(s), \eta(t)) \\ &\qquad \wedge \ (\eta(s), \eta(t)) \in \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d(M)\} \\ &= \{(s,t) \in [S \times S]^d \mid \ell(s) = \ell(t) \ \wedge \ (\eta(s), \eta(t)) \in \mathcal{H}_\#^{\mathcal{K}(d)} \circ \mathcal{K}_\#^d(M)\}\end{aligned}$$

$$= \{(s,t) \in S \times S \mid d(s,t) > 0 \ \wedge \ \ell(s) = \ell(t) \ \wedge \ (\eta(s),\eta(t)) \in \mathcal{H}_{\#}^{\mathcal{K}(d)} \circ \mathcal{K}_{\#}^{d}(M)\}$$

Using the characterisation of the associated approximation of the Hausdorff lifting in Lemma 3.6.14, we obtain that this is equivalent to

for all $p \in \eta(s)$, whenever $\min_{q' \in \eta(t)} \mathcal{K}(d)(p,q') = \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t))$, then there exists $q \in \eta(t)$ such that $(p,q) \in \mathcal{K}_{\#}^{d}(M)$ and $\mathcal{K}(d)(p,q) = \mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t))$ (and vice versa),

assuming that $\ell(s) = \ell(t)$ (this is a requirement in the definition of $\mathcal{M}_{\#}^{d}(M)$), since then we have $\mathcal{H}(\mathcal{K}(d))(\eta(s),\eta(t)) = d(s,t) > 0$ and hence $(\eta(s),\eta(t)) \in [\mathcal{P}(D) \times \mathcal{P}(D)]^{\mathcal{H}(\mathcal{K}(d))}$.

Since also $d = \mathcal{M}(d)$, the condition above can be rewritten to

for all $p \in \eta(s)$, whenever $\min_{q' \in \eta(t)} \mathcal{K}(d)(p,q') = d(s,t)$, then there exists $q \in \eta(t)$ such that $(p,q) \in \mathcal{K}_{\#}^{d}(M)$ and $\mathcal{K}(d)(p,q) = d(s,t)$ (and vice versa).

From Lemma 3.6.7 we know that $(p,q) \in \mathcal{K}_{\#}^{d}(M)$ iff $\mathcal{K}(d)(p,q) > 0$ and there exists $c \in \Omega(p,q)$ such that $supp(c) \subseteq M$ and $\sum_{u,v \in S} c(u,v) \cdot d(u,v) = \mathcal{K}(d)(p,q)$. We instantiate the condition above accordingly and obtain

for all $p \in \eta(s)$, whenever $d(s,t) = \min_{q' \in \eta(t)} \mathcal{K}(d)(p,q')$, then there exists $q \in \eta(t)$ such that there exists $c \in \Omega(p,q)$ with $supp(c) \subseteq M$, $\mathcal{K}(d)(p,q) = \sum_{u,v \in S} c(u,v) \cdot d(u,v)$ and $\mathcal{K}(d)(p,q) = d(s,t)$ (and vice versa).

The two last equalities can be simplified to $d(s,t) = \sum_{u,v \in S} c(u,v) \cdot d(u,v)$, since

$$\mathcal{K}(d)(p,q) \leq \sum_{u,v \in S} c(u,v) \cdot d(u,v) = d(s,t) = \min_{q' \in \eta(t)} \mathcal{K}(d)(p,q') \leq \mathcal{K}(d)(p,q)$$

and hence $\mathcal{K}(d)(p,q) = d(s,t)$ can be inferred from the remaining conditions.

We finally obtain the following equivalent characterisation:

for all $p \in \eta(s)$, whenever $d(s,t) = \min_{q' \in \eta(t)} \mathcal{K}(d)(p,q')$, then there exists $q \in \eta(t)$ such that there exists $c \in \Omega(p,q)$ with $supp(c) \subseteq M$, $d(s,t) = \sum_{u,v \in S} c(u,v) \cdot d(u,v)$ (and vice versa).

Hence we obtain that $(\eta(s),\eta(t)) \in \mathcal{H}_{\#}^{\mathcal{K}(d)} \circ \mathcal{K}_{\#}^{d}(M)$ is equivalent to the the second and third item of Definition 3.6.19 (under the assumption that $\ell(s) = \ell(t)$), while the first item is covered by the other conditions ($d(s,t) > 0$ and $\ell(s) = \ell(t)$) in the characterisation of $\mathcal{M}_{\#}^{d}(M)$. □

Lemma A.2.13 (Lemma 3.6.23). *The approximation for the adapted Hausdorff lifting $\mathcal{G}$ is as follows. Let $a: X \times X \to \{0,1\}$, then $\mathcal{G}_a^{\#}: \mathcal{P}([X \times X]_a) \to \mathcal{P}([\mathcal{P}(X) \times \mathcal{P}(X)]_a)$ with*

$$\begin{aligned}\mathcal{G}_a^{\#}(R) \quad = \quad & \{(X_1, X_2) \in [\mathcal{P}(X) \times \mathcal{P}(X)]_{\mathcal{G}(a)} \mid \\ & \forall x_1 \in X_1 \exists x_2 \in X_2 \colon \big((x_1, x_2) \notin [X \times X]_a \vee (x_1, x_2) \in R\big) \\ & \wedge \forall x_2 \in X_2 \exists x_1 \in X_1 \colon \big((x_1, x_2) \notin [X \times X]_a \vee (x_1, x_2) \in R\big)\}\end{aligned}$$

Proof. We rely on the characterisation of $\mathcal{H}_{\#}^a$ (dual case) of Lemma 3.6.14 and we examine the case where $\mathbb{M} = \{0,1\}$. In this case, whenever we have $(X_1, X_2) \in [\mathcal{P}(X) \times \mathcal{P}(X)]^{\mathcal{H}(a)}$ it must necessarily hold that $\mathcal{H}(a)(X_1, X_2) = 1$. Hence, the first part of the conjunction simplifies to:

$$\forall x_1 \in X_1 \big(\min_{x_2' \in X_2} a(x_1, x_2') = 1 \Rightarrow \exists x_2 \in X_2 \colon (x_1, x_2) \in R \wedge a(x_1, x_2) = 1 \big),$$

from which we can omit $a(x_1, x_2) = 1$ from the conclusion, since this holds automatically. Furthermore $\min_{x_2' \in X_2} a(x_1, x_2') = 1$ can be rewritten to $\forall x_2 \in X_2 \colon a(x_1, x_2) = 1$. This gives us:

$$\begin{aligned} & \forall x_1 \in X_1 \big(\neg \forall x_2 \in X_2 \colon a(x_1, x_2) = 1 \vee \exists x_2 \in X_2 \colon (x_1, x_2) \in R \big) \\ \equiv \quad & \forall x_1 \in X_1 \big(\exists x_2 \in X_2 \colon a(x_1, x_2) = 0 \vee \exists x_2 \in X_2 \colon (x_1, x_2) \in R \big) \\ \equiv \quad & \forall x_1 \in X_1 \exists x_2 \in X_2 \big((x_1, x_2) \notin [X \times X]^a \vee (x_1, x_2) \in R \big). \end{aligned}$$

Since this characterisation is independent of the order, we can replace $[X \times X]^a$ by $[X \times X]_a$ and obtain a characterizing condition for $\mathcal{G}_a^{\#}$ (primal case). ☐

Lemma A.2.14 (Lemma 3.6.24). *Bisimilarity for a transition system $\mathsf{TS} = (X, \eta)$ is the greatest fixpoint of $\mathcal{B} = (\eta \times \eta)^* \circ \mathcal{G}$.*

Proof. Let for $a : X \times X \to \{0,1\}$, $x, y \in X$. Then we have

$$\begin{aligned}(\eta \times \eta)^* \circ \mathcal{G}(a)(x, y) \quad &= \quad \mathcal{G}(a)(\eta(x), \eta(y)) \\ &= \quad \max_u(\min_\in(a))(\eta(x), \eta(y)) \\ &= \quad \max_{u(C) = (\eta(x), \eta(y))} (\min_\in^{X \times X}(a))(C) \\ &= \quad \max_{u(C) = (\eta(x), \eta(y))} \min_{(x', y') \in C} a(x', y')\end{aligned}$$

Now we prove that this, indeed, corresponds with the standard bisimulation function, i.e. $\max_{u(C) = (\eta(x), \eta(y))} \min_{(x', y') \in C} a(x', y') = 1$ if and only if for all $x' \in \eta(x)$ there exists $y' \in \eta(y)$ such that $a(x', y') = 1$ and vice versa. For the first implication, assume that $\max_{u(C) = (\eta(x), \eta(y))} \min_{(x', y') \in C} a(x', y') = 1$. This means that there exists $C \subseteq X \times X$ such that $u(C) = (\pi_1(C), \pi_2(C)) = (\eta(x), \eta(y))$ and $\min_{(x',y')\in C} a(x', y') = 1$. Then we have two cases. Either $C = \emptyset$, which means that $\eta(x) = \eta(y) = \emptyset$, that is, x and y have no successors, and so the bisimulation property vacuously holds. Otherwise, $C \neq \emptyset$, and we must have $a(x', y') = 1$ for all $(x', y') \in C$. Then, since $(\pi_1(C), \pi_2(C)) = (\eta(x), \eta(y))$, for all $x' \in \eta(x)$ there must exists $y' \in \eta(y)$ such that $(x', y') \in C$, and thus $a(x', y') = 1$. Vice versa, for all $y' \in \eta(y)$ there must exists $x' \in \eta(x)$ such that $(x', y') \in C$, and thus $a(x', y') = 1$. So the bisimulation property holds.

For the other implication, assume that for all $x' \in \eta(x)$ there exists $y' \in \eta(y)$ such that $a(x', y') = 1$ and call $c_1(x')$ such a y'. Vice versa, assume also that for all $y' \in \eta(y)$ there exists $x' \in \eta(x)$ such that $a(x', y') = 1$ and call $c_2(y')$ such a x'. This means that for all $x' \in \eta(x)$ and $y' \in \eta(y)$, we have $a(x', c_1(x')) = a(c_2(y'), y') = 1$. Now let $C' = \{(x', y') \in \eta(x) \times \eta(y) \mid c_1(x') = y' \vee x' = c_2(y')\}$. Since we assumed that for all $x' \in \eta(x)$ there exists $y' \in \eta(y)$ such that $c_1(x') = y'$, we must have that $\pi_1(C') = \eta(x)$. The same holds for all $y' \in \eta(y)$, thus $\pi_2(C') = \eta(y)$. Therefore, we know that $u(C') = (\eta(x), \eta(y))$, and we can conclude by showing that $a(x', y') = 1$ for all $(x', y') \in C'$, in which case also $\max_{u(C)=(\eta(x),\eta(y))} \min_{(x',y')\in C} a(x', y') = 1$. By definition of C' either $c_1(x') = y'$ or $x' = c_2(y')$, or both, must hold. Assume the first one holds, the other case is similar. Then, we can immediately conclude since by hypothesis we know that $a(x', c_1(x')) = 1$.

Since we proved that the function $\mathcal{B}$ is the same of the standard bisimulation function, then its greatest fixpoint $\nu\mathcal{B}$ is the bisimilarity on η. □

Lemma A.2.15 (Lemma 3.6.25)**.** *Let $a\colon X \times X \to \{0,1\}$. The approximation for the bisimilarity function $\mathcal{B}$ in the primal sense is $\mathcal{B}_a^\#\colon \mathcal{P}([X \times X]_a) \to \mathcal{P}([X \times X]_{\mathcal{B}(a)})$ with*

$$\begin{aligned} \mathcal{B}_a^\#(R) \quad = \quad & \{(x_1, x_2) \in [X \times X]_{\mathcal{B}(a)} \mid \\ & \quad \forall y_1 \in \eta(x_1) \exists y_2 \in \eta(x_2)\big((y_1, y_2) \notin [X \times X]_a \vee (y_1, y_2) \in R\big) \\ & \quad \wedge \forall y_2 \in \eta(x_2) \exists y_1 \in \eta(x_1)\big((y_1, y_2) \notin [X \times X]_a \vee (y_1, y_2) \in R\big)\} \end{aligned}$$

Proof. From Lemma 3.6.14 we know that

$$\begin{aligned} \mathcal{G}_a^\#\colon [X \times X]_a \quad & \to \quad [\mathcal{P}(X) \times \mathcal{P}(X)]_{\mathcal{G}(a)} \\ \mathcal{G}_a^\#(R) \quad & = \quad \{(X_1, X_2) \in [\mathcal{P}(X) \times \mathcal{P}(X)]_{\mathcal{G}(a)} \mid \\ & \qquad \forall x_1 \in X_1 \exists x_2 \in X_2\colon \big((x_1, x_2) \notin [X \times X]_a \vee (x_1, x_2) \in R\big) \\ & \qquad \wedge \forall x_2 \in X_2 \exists x_1 \in X_1\colon \big((x_1, x_2) \notin [X \times X]_a \vee (x_1, x_2) \in R\big)\}. \end{aligned}$$

Furthermore

$$\begin{aligned} ((\eta \times \eta)^*)^\#_{\mathcal{G}(a)}\colon [\mathcal{P}(X) \times \mathcal{P}(X)]_{\mathcal{G}(a)} \quad & \to \quad [X \times X]_{\mathcal{B}(a)} \\ ((\eta \times \eta)^*)^\#_{\mathcal{G}(a)}(Q) \quad & = \quad (\eta \times \eta)^{-1}(Q) \end{aligned}$$

Composing these functions we obtain:

$$
\begin{aligned}
\mathcal{B}_a^{\#}\colon [X\times X]_a &\to [X\times X]_{\mathcal{B}(a)}\\
\mathcal{B}_a^{\#}(R) &= (\eta\times\eta)^{-1}(\{(Y_1,Y_2)\in[\mathcal{P}(X)\times\mathcal{P}(X)]_{\mathcal{G}(a)}\mid\\
&\qquad \forall y_1\in Y_1\exists y_2\in Y_2\colon\big((y_1,y_2)\notin[X\times X]_a\vee(y_1,y_2)\in R\big)\\
&\qquad \wedge\forall y_2\in Y_2\exists y_1\in Y_1\colon\big((y_1,y_2)\notin[X\times X]_a\vee(y_1,y_2)\in R\big)\})\\
&= \{(x_1,x_2)\in[X\times X]_{\mathcal{B}(a)}\mid\\
&\qquad \forall y_1\in\eta(x_1)\exists y_2\in\eta(x_2)\colon\big((y_1,y_2)\notin[X\times X]_a\vee(y_1,y_2)\in R\big)\\
&\qquad \wedge\forall y_2\in\eta(x_2)\exists y_1\in\eta(x_1)\colon\big((y_1,y_2)\notin[X\times X]_a\vee(y_1,y_2)\in R\big)\}.
\end{aligned}
$$

□

Lemma A.2.16 (Lemma 3.6.30). *Let $a\colon V\to[0,1]$. The approximation for the value iteration function $\mathcal{V}$ in the dual sense is $\mathcal{V}_{\#}^a\colon\mathcal{P}([V]^a)\to\mathcal{P}([V]^{\mathcal{V}(a)})$ with*

$$
\begin{aligned}
\mathcal{V}_{\#}^a(V') = \{v\in[V]^{\mathcal{V}(a)}\mid&\big(v\in V_{\mathsf{Min}}\wedge\arg\min_{v'\in\eta_{\min}(v)}a(v')\cap V'\neq\varnothing\big)\vee\\
&\big(v\in V_{\mathsf{Max}}\wedge\arg\max_{v'\in\eta_{\max}(v)}a(v')\subseteq V'\big)\vee\big(v\in V_{\mathsf{Av}}\wedge supp(\eta_{\mathrm{av}}(v))\subseteq V'\big)\}
\end{aligned}
$$

Proof. Let $a\colon V\to[0,1]$ and $V'\subseteq[V]^a$. By Proposition 3.5.7 we have:

$$
\begin{aligned}
\mathcal{V}_{\#}^a(V') =&\big(V_{\mathsf{Min}}\cap(\eta_{\min}^*\circ\min\nolimits_{\in})_{\#}^a(V')\big)\cup\big(V_{\mathsf{Max}}\cap(\eta_{\max}^*\circ\max\nolimits_{\in})_{\#}^a(V')\big)\cup\\
&\big(V_{\mathsf{Av}}\cap(\eta_{\mathrm{av}}^*\circ\mathrm{av}_D)_{\#}^a(V')\big)\cup\big(V_{\mathsf{Sink}}\cap(c_w)_{\#}^a(V')\big)
\end{aligned}
$$

It holds that $(\eta_{\min}^*)_{\#}^{\min_{\in}(v)}=\eta_{\min}^{-1}$, $(\eta_{\max}^*)_{\#}^{\max_{\in}(v)}=\eta_{\max}^{-1}$ and $(\eta_{\mathrm{av}}^*)_{\#}^{\mathrm{av}_D(v)}=\eta_{\mathrm{av}}^{-1}$. Using previous results (Proposition 3.5.5) we deduce

$$
\begin{aligned}
&v\in(\eta_{\min}^*\circ\min\nolimits_{\in})_{\#}^a(V')\Leftrightarrow\eta_{\min}(v)\in(\min\nolimits_{\in})_{\#}^a(V')\Leftrightarrow\arg\min_{v'\in\eta_{\min}(v)}a(v')\cap V'\neq\varnothing\\
&v\in(\eta_{\max}^*\circ\max\nolimits_{\in})_{\#}^a(V')\Leftrightarrow\eta_{\max}(v)\in(\max\nolimits_{\in})_{\#}^a(V')\Leftrightarrow\arg\max_{v'\in\eta_{\max}(v)}a(v')\subseteq V'\\
&v\in(\eta_{\mathrm{av}}^*\circ\mathrm{av}_D)_{\#}^a(V')\Leftrightarrow\eta_{\mathrm{av}}(v)\in(\mathrm{av}_D)_{\#}^a(V')\Leftrightarrow supp(\eta_{\mathrm{av}}(v))\subseteq V'
\end{aligned}
$$

Lastly $(c_w)_{\#}^a(V')=\varnothing$ for any $V'\subseteq V$ since c_w is a constant function which concludes the proof. □

Lemma A.2.17 (Lemma 3.6.34). *The function $sub_w'\colon K^E\to K^E$, defined via $sub_w'(a)(e)=a(e)\ominus_{\mathbb{Z}}w(e)$ for $a\colon E\to K$, $e\in E$ and $w\colon E\to\mathbb{Z}$, is non-expansive.*

Proof. Let $w\colon E\to\mathbb{Z}$, $a,b\colon E\to K$. Without loss of generality, assume

$$
\|\, sub_w'(b)\ominus sub_w'(a)\,\| = sub_w'(b)(e)\ominus sub_w'(a)(e) = (b(e)\ominus_{\mathbb{Z}}w(e))\ominus(a(e)\ominus_{\mathbb{Z}}w(e))
$$

for some $e \in E$. We omit the trivial case, i.e. we assume $\| sub'_w(b) \ominus sub'_w(a) \| > 0$. Thus $b(e) > a(e)$ has to hold by monotonicity.

We make the following distinction of cases.

1. $w(e) \geq 0 \wedge a(e) \geq w(e)$:

$$\begin{aligned} &(b(e) \ominus_{\mathbb{Z}} w(e)) \ominus (a(e) \ominus_{\mathbb{Z}} w(e)) \\ &= (b(e) - w(e)) - (a(e) - w(e)) && [b(e) > a(e) \geq w(e)] \\ &= b(e) - a(e) \\ &\leq \| b \ominus a \| \end{aligned}$$

2. $w(e) \geq 0 \wedge a(e) < w(e)$:

$$\begin{aligned} &(b(e) \ominus_{\mathbb{Z}} w(e)) \ominus (a(e) \ominus_{\mathbb{Z}} w(e)) \\ &= (b(e) \ominus w(e)) \ominus 0 && [a(e) < w(e)] \\ &\leq b(e) - a(e) && [a(e) < w(e),\ a(e) < b(e)] \\ &\leq \| b \ominus a \| \end{aligned}$$

3. $w(e) < 0 \wedge b(e) - w(e) \leq k$:

$$\begin{aligned} &(b(e) \ominus_{\mathbb{Z}} w(e)) \ominus (a(e) \ominus_{\mathbb{Z}} w(e)) \\ &= (b(e) - w(e)) - (a(e) - w(e)) && [b(e) > a(e)] \\ &= b(e) - a(e) \\ &\leq \| b \ominus a \| \end{aligned}$$

4. $w(e) < 0 \wedge b(e) - w(e) > k$:

$$\begin{aligned} &(b(e) \ominus_{\mathbb{Z}} w(e)) \ominus (a(e) \ominus_{\mathbb{Z}} w(e)) \\ &= k - (a(e) \ominus_{\mathbb{Z}} w(e)) && [a(e) \ominus_{\mathbb{Z}} w(e) \in K] \\ &\leq k - (a(e) \ominus_{\mathbb{Z}} (b(e) - k)) && [w(e) < b(e) - k] \\ &= k - \min\{\max\{a(e) - b(e) + k, 0\}, k\} \\ &= k - \min\{a(e) - b(e) + k, k\} && [k \geq b(e)] \\ &= k - (a(e) - b(e) + k) && [b(e) > a(e)] \\ &= b(e) - a(e) \\ &\leq \| b \ominus a \| \end{aligned}$$

□

Lemma A.2.18 (Lemma 3.6.35). *Given* $w\colon E \to \mathbb{Z}$ *and* $a\colon E \to K$ *the approximation* $(sub'_w)^a_\#\colon K^{[E]^a} \to K^{[E]^{sub'_w(a)}}$ *of* $sub'_w\colon K^E \to K^E$*, is given by*

$$(sub'_w)^a_\#(E') = \{e \in E' \mid 0 < a(e) - w(e) \le k\}$$

for $E' \subseteq [E]^a$*.*

Proof. Note that here the minimal possible decrease is $\delta = 1 \le \delta^a$, which we will use in the following

Now:

$$\begin{aligned}(sub'_w)^{a,\delta}_\#(E') &= \{e \in [E]^{sub'_w(a)} \mid sub'_w(a)(e) \ominus sub'_w(a \ominus \delta_{E'})(e) \ge \delta\}\\ &= \{e \in [E]^{sub'_w(a)} \mid (a(e) \ominus_\mathbb{Z} w(e)) \ominus ((a \ominus \delta_{E'})(e) \ominus_\mathbb{Z} w(e)) \ge \delta\}\end{aligned}$$

First note, that whenever $e \notin E'$ then

$$(a(e) \ominus_\mathbb{Z} w(e)) \ominus ((a \ominus \delta_{E'})(e) \ominus_\mathbb{Z} w(e)) = (a(e) \ominus_\mathbb{Z} w(e)) \ominus (a(e) \ominus_\mathbb{Z} w(e)) = 0 < \delta.$$

Now let $e \in [E']^{sub'_w(a)}$, i.e. $a(e) - w(e) > 0$. Whenever $a(e) - w(e) > k$ we obtain

$$(a(e) \ominus_\mathbb{Z} w(e)) \ominus ((a \ominus \delta_{E'})(e) \ominus_\mathbb{Z} w(e)) = k - \min\{a(e) - \delta - w(e), k\} < k - \min\{k - \delta, k\} \le \delta.$$

If on the other hand $e \in E'$ with $0 < a(e) - w(e) \le k$, we have:

$$(a(e) \ominus_\mathbb{Z} w(e)) \ominus ((a \ominus \delta_{E'})(e) \ominus_\mathbb{Z} w(e)) = (a(e) - w(e)) - (a(e) - \delta - w(e)) = \delta,$$

since $E' \subseteq [E]^a$ and by choice of δ, $a(e) - \delta - w(e) \ge 0$ holds.

To summarize we obtain

$$(sub'_w)^a_\#(E') = \{e \in [E']^{sub'_w(a)} \mid a(e) - w(e) \le k\} = \{e \in E' \mid 0 < a(e) - w(e) \le k\}.$$

□

Lemma A.2.19 (Lemma 3.6.36). *Let* $V' \subseteq [V]^a$ *then* $v \in \mathcal{E}^a_\#(V')$ *if* $v \in [V]^{\mathcal{E}(a)}$ *and*

- *whenever* $v \in V_{Min}$ *there exists some* $(v, v'') \in E$ *with* $\min_{(v,v') \in E} a(v') \ominus_\mathbb{Z} w(v, v') = a(v'') \ominus_\mathbb{Z} w(v, v'')$, $0 < a(v'') - w(v, v'') \le k$ *and* $v'' \in V'$
- *whenever* $v \in V_{Max}$*: if* $(v, v'') \in E$ *with* $\max_{(v,v') \in E} a(v') \ominus_\mathbb{Z} w(v, v') = a(v'') \ominus_\mathbb{Z} w(v, v'')$ *then* $0 < a(v'') - w(v, v'') \le k$ *and* $v'' \in V'$

Proof. We have

$$\begin{aligned}
\mathcal{E}^a_{\#}(V') = \{v \in [V]^{\mathcal{E}(a)} \mid &(v \in V_{\mathsf{Min}} \wedge Min_{sub'_w \circ \pi_2^*(a)|_{E_0}} \cap (sub'_w)_{\#}^{\pi_2^*(a)}(\pi_2^{-1}(V')) \neq \varnothing) \\
&\vee (v \in V_{\mathsf{Max}} \wedge Max_{sub'_w \circ \pi_2^*(a)|_{E_1}} \subseteq (sub'_w)_{\#}^{\pi_2^*(a)}(\pi_2^{-1}(V')))\} \\
= \{v \in [V]^{\mathcal{E}(a)} \mid &(v \in V_{\mathsf{Min}} \wedge Min_{sub'_w \circ \pi_2^*(a)|_{E_0}} \\
&\cap \{(v,v') \in E \mid 0 < a(v') - w(v,v') \leq k,\ v' \in V'\} \neq \varnothing) \\
&\vee (v \in V_{\mathsf{Max}} \wedge Max_{sub'_w \circ \pi_2^*(a)|_{E_1}} \\
&\subseteq \{(v,v') \in E \mid 0 < a(v') - w(v,v') \leq k,\ v' \in V'\}\} \\
= \{v \in [V]^{\mathcal{E}(a)} \mid &(v \in V_{\mathsf{Min}} \wedge \text{ there exists some } (v,v'') \in E \text{ with} \\
&\min_{(v,v') \in E} a(v') \ominus_{\mathbb{Z}} w(v,v') = a(v'') \ominus_{\mathbb{Z}} w(v,v''), \\
&0 < a(v'') - w(v,v'') \leq k \text{ and } v'' \in V') \\
&\vee (v \in V_{\mathsf{Max}} \wedge \text{ if } (v,v'') \in E \text{ with} \\
&\max_{(v,v') \in E} a(v') \ominus_{\mathbb{Z}} w(v,v') = a(v'') \ominus_{\mathbb{Z}} w(v,v'') \\
&\text{then } 0 < a(v'') - w(v,v'') \leq k \text{ and } v'' \in V')\}
\end{aligned}$$

□

A.3. Proofs of Chapter 4

Lemma A.3.1 (Lemma 4.3.4). *Given $a \in \mathbb{M}^Y$, $g\colon Z \to Y$ and $0 \sqsubset \delta \in \mathbb{M}$, then we have*

1. $\alpha^{a \circ g, \delta} \circ g^{-1} = g^* \circ \alpha^{a,\delta}$
2. $\gamma^{a \circ g, \delta} \circ g^* = g^{-1} \circ \gamma^{a,\delta}$

This implies that for two $\mathbb{C}$-arrows $f\colon a \dashrightarrow b$, $h\colon b \dashrightarrow c$, it holds that $\#(h \circ f) = \#(h) \circ \#(f)$ whenever f or h is a reindexing, i.e., is contained in $\mathbb{C}^*$.

Proof.

1. Let $Y' \subseteq [Y]^a$. Then

$$\begin{aligned}
g^*(\alpha^{a,\delta}(Y')) = g^*(a \ominus \delta_{Y'}) = (a \ominus \delta_{Y'}) \circ g = a \circ g \ominus \delta_{Y'} \circ g \\
= a \circ g \ominus \delta_{g^{-1}(Y')} = \alpha^{a \circ g, \delta}(g^{-1}(Y'))
\end{aligned}$$

where we use that $(\delta_{Y'} \circ g)(z) = \delta$ if $g(z) \in Y'$, equivalent to $z \in g^{-1}(Y')$, and 0 otherwise. Hence $\delta_{Y'} \circ g = \delta_{g^{-1}(Y')}$.

2. Let $b \in \mathbb{M}^Y$ with $a \ominus \delta \sqsubseteq b \sqsubseteq a$. Then

$$\begin{aligned}\gamma^{a\circ g,\delta} \circ g^*(b) &= \{z \in Z \mid a(g(z)) \ominus b(g(z)) \sqsupseteq \delta\} = \{z \in Z \mid g(z) \in \gamma^{a,\delta}(b)\} \\ &= g^{-1}(\gamma^{a,\delta}(b))\end{aligned}$$

It is left to show that $\#(h \circ f) = \#(h) \circ \#(f)$ whenever f or h is a reindexing. Note that on reindexings it holds that $\#(g^*) = g^{-1}$.

Let $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$, $c \in \mathbb{M}^W$ and assume first that f is a reindexing, i.e., $f = g^*$ for some $g\colon Z \to Y$. Let $Y' \subseteq [Y]^a$, then

$$\begin{aligned}\#(h \circ f) = (h \circ f)^a_\# &= \bigcup_{\delta \sqsupset 0} (\gamma^{h(f(a)),\delta} \circ h \circ f \circ \alpha^{a,\delta})(Y') \\ &= \bigcup_{\delta \sqsupset 0} (\gamma^{h(f(a)),\delta} \circ h \circ g^* \circ \alpha^{a,\delta})(Y') \\ &= \bigcup_{\delta \sqsupset 0} (\gamma^{h(f(a)),\delta} \circ h \circ \alpha^{a\circ g,\delta})(g^{-1}(Y')) && [1.] \\ &= \bigcup_{\delta \sqsupset 0} (\gamma^{h(f(a)),\delta} \circ h \circ \alpha^{f(a),\delta})(\#(g^*)(Y')) \\ &= \#(h)((\#(f)(Y')))\end{aligned}$$

Now we assume that h is a reindexing, i.e., $h = g^*$ for some $g\colon W \to Z$. Let again $Y' \subseteq [Y]^a$, then:

$$\begin{aligned}\#(h \circ f) = (h \circ f)^a_\# &= \bigcup_{\delta \sqsupset 0} (\gamma^{h(f(a)),\delta} \circ h \circ f \circ \alpha^{a,\delta})(Y') \\ &= \bigcup_{\delta \sqsupset 0} (\gamma^{f(a)\circ g,\delta} \circ g^* \circ f \circ \alpha^{a,\delta})(Y') \\ &= \bigcup_{\delta \sqsupset 0} g^{-1}((\gamma^{f(a),\delta} \circ f \circ \alpha^{a,\delta})(Y')) && [2.] \\ &= g^{-1}(\bigcup_{\delta \sqsupset 0} (\gamma^{f(a),\delta} \circ f \circ \alpha^{a,\delta})(Y')) && [\text{preimage preserves union}] \\ &= \#(g^*)(\bigcup_{\delta \sqsupset 0} (\gamma^{f(a),\delta} \circ f \circ \alpha^{a,\delta})(Y')) \\ &= \#(h)((\#(f)(Y')))\end{aligned}$$

□

Lemma A.3.2 (Lemma 4.5.3)**.** *Consider the lifting of the distribution functor presented in Example 4.4.1 and let $Z = [0,1] \times \{0,1\}$. Then we have*

$$(\tilde{\mathcal{D}}_f)^{\pi_1}_\#((0,1] \times \{1\}) = \{p \in \mathcal{D}_f Z \mid supp(p) \in (0,1] \times \{1\}\}.$$

Proof. Let $\delta > 0$. We define

$$\tilde{\pi}_1^\delta := \alpha^{\pi_1,\delta}((0,1] \times \{1\})$$

where $\tilde{\pi}_1^\delta(x,0) = x$, $\tilde{\pi}_1^\delta(x,1) = x \ominus \delta$ for $x \in [0,1]$. Note that $[\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f \pi_1} = \{p \in \mathcal{D}_f Z \mid \exists(x,b) \in \mathit{supp}(p) \text{ with } x \geq 0\}$. Now

$$\begin{aligned}(\tilde{\mathcal{D}}_f)_\#^{\pi_1,\delta}((0,1]\times\{1\}) &= \{p \in [\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f\pi_1} \mid \tilde{\mathcal{D}}_f\pi_1(p) \ominus \tilde{\mathcal{D}}_f(\tilde{\pi}_1^\delta)(p) \geq \delta\} \\ &= \{p \in [\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f\pi_1} \mid \Big(\sum_{x\in[0,1]} x\cdot p(x,0) \oplus \sum_{x\in[0,1]} x \cdot p(x,1)\Big) \\ &\qquad \ominus \Big(\sum_{x\in[0,1]} x\cdot p(x,0) \oplus \sum_{x\in[0,1]} (x\ominus\delta)\cdot p(x,1)\Big) \geq \delta\} \\ &= \{p \in [\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f\pi_1} \mid \sum_{x\in[0,\delta)} x\cdot p(x,1) + \sum_{x\in[\delta,1]} \delta\cdot p(x,1) \geq \delta\} \\ &= \{p \in [\mathcal{D}_f Z]^{\tilde{\mathcal{D}}_f\pi_1} \mid \mathit{supp}(p) \in [\delta,1]\times\{1\}\}.\end{aligned}$$

Where the second last equality uses the fact that $x \ominus (x \ominus \delta) = \delta$ if $x \geq \delta$ and x otherwise.
Now, we obtain

$$(\tilde{\mathcal{D}}_f)_\#^{\pi_1}((0,1]\times\{1\}) = \bigcup_{\delta \sqsupset 0} (\tilde{\mathcal{D}}_f)_\#^{\pi_1,\delta}((0,1]\times\{1\}) = \{p \in \mathcal{D}_f Z \mid \mathit{supp}(p) \in (0,1]\times\{1\}\}.$$

□

Lemma A.3.3 (Lemma 4.5.5)**.** *Consider the lifting of the powerset functor from Example 4.4.2 and let $Z = \mathbb{M}\times\{0,1\}$. Then we have*

$$(\tilde{\mathcal{P}}_f)_\#^{\pi_1}((\mathbb{M}\backslash\{0\})\times\{1\}) = \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f\pi_1} \mid \exists(s,1)\in S,\ \forall(s',0)\in S:\ s \sqsupset s'\}.$$

Proof. Let $\delta \sqsupset 0$ and define $\tilde{\pi}_1^\delta$ as in the proof of Lemma 4.5.3. Then

$$\begin{aligned}(\tilde{\mathcal{P}}_f)_\#^{\pi_1,\delta}((\mathbb{M}\backslash\{0\})\times\{1\}) &= \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f\pi_1} \mid \tilde{\mathcal{P}}_f\pi_1(S) \ominus \tilde{\mathcal{P}}_f(\tilde{\pi}_1^\delta)(S) \sqsupseteq \delta\} \\ &= \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f\pi_1} \mid \max_{(s,s')\in S} s \ominus (\max_{(s,s')\in S} s \ominus s'\cdot\delta) \sqsupseteq \delta\} \\ &= \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f\pi_1} \mid \exists(s,1)\in S,\ \forall(s',0)\in S:\ s\ominus\delta \sqsupseteq s'\}\end{aligned}$$

For the last step we note that this condition ensures that the second maximum equates to $\max_{(s,s')\in S} s \ominus \delta$ which is required for the inequality to hold. Now, we obtain

$$\begin{aligned}(\tilde{\mathcal{P}}_f)_\#^{\pi_1}((\mathbb{M}\backslash\{0\})\times\{1\}) &= \bigcup_{\delta\sqsupset 0} (\tilde{\mathcal{P}}_f)_\#^{\pi_1,\delta}((\mathbb{M}\backslash\{0\})\times\{1\}) \\ &= \{S \in [\mathcal{P}_f Z]^{\tilde{\mathcal{P}}_f\pi_1} \mid \exists(s,1)\in S,\ \forall(s',0)\in S:\ s \sqsupset s'\}.\end{aligned}$$

□

Theorem A.3.4 (Theorem 4.6.1)**.** *The category $\mathbb{C}_f$ with the following operators is gs-monoidal:*

1. *The tensor $\otimes$ on objects $a \in \mathbb{M}^Y$ and $b \in \mathbb{M}^Z$ is defined as*

$$a \otimes b = a + b \in \mathbb{M}^{Y+Z}$$

where for $k \in Y + Z$ we have $(a+b)(k) = a(k)$ if $k \le |Y|$ and $(a+b)(k) = b(k-|Y|)$ if $|Y| < k \le |Y| + |Z|$.

On arrows $f: a \dashrightarrow b$ and $g: a' \dashrightarrow b'$ (with $a' \in \mathbb{M}^{Y'}$, $b' \in \mathbb{M}^{Z'}$) tensor is given by

$$f \otimes g : \mathbb{M}^{Y+Y'} \to \mathbb{M}^{Z+Z'}, \quad (f \otimes g)(u) = f(\overleftarrow{u}_Y) + g(\overrightarrow{u}_Y)$$

for $u \in \mathbb{M}^{Y+Y'}$ where $\overleftarrow{u}_Y \in \mathbb{M}^Y$ and $\overrightarrow{u}_Y \in \mathbb{M}^{Y'}$, defined as $\overleftarrow{u}_Y(k) = u(k)$ $(1 \le k \le |Y|)$ and $\overrightarrow{u}_Y(k) = u(|Y| + k)$ $(1 \le k \le |Y'|)$.

2. *The symmetry $\rho_{a,b}: a \otimes b \dashrightarrow b \otimes a$ for $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ is defined for $u \in \mathbb{M}^{Y+Z}$ as*

$$\rho_{a,b}(u) = \overrightarrow{u}_Y + \overleftarrow{u}_Y.$$

3. *The unit e is the unique mapping $e: \varnothing \to \mathbb{M}$.*

4. *The duplicator $\nabla_a: a \dashrightarrow a \otimes a$ for $a \in \mathbb{M}^Y$ is defined for $u \in \mathbb{M}^Y$ as*

$$\nabla_a(u) = u + u.$$

5. *The discharger $!_a: a \dashrightarrow e$ for $a \in \mathbb{M}^Y$ is defined for $u \in \mathbb{M}^Y$ as $!_a(u) = e$.*

Proof. In the following let $a \in \mathbb{M}^Y$, $a' \in \mathbb{M}^{Y'}$, $b \in \mathbb{M}^Z$, $b' \in \mathbb{M}^{Z'}$, $c \in \mathbb{M}^W$, $c' \in \mathbb{M}^{W'}$ be objects in $\mathbb{C}_f$.

We know that $\mathbb{C}_f$ is a well-defined category from Lemma 4.3.2. We also note that disjoint unions of non-expansive functions are non-expansive and given $f: a \dashrightarrow b$ and $g: a' \dashrightarrow b'$, that

$$\begin{aligned} & (f \otimes g)(a \otimes a') = (f \otimes g)(a + a') \\ = \quad & f(\overleftarrow{(a+a')}_Y) + g(\overrightarrow{(a+a')}_Y) = f(a) + g(a') = b + b' = b \otimes b'. \end{aligned}$$

Thus, we have a well-defined arrow $f \otimes g : a \otimes a' \dashrightarrow b \otimes b'$.

We now verify all the axioms of gs-monoidal categories given in Definition 2.4.9. In general the calculations are straightforward, but we give them here for completeness.

We will in the following often use the fact that $\overleftarrow{u}_Y + \overrightarrow{u}_Y = u$ whenever Y is a subset of the domain of u.

1. functoriality of tensor:

- $id_{a\otimes b} = id_a \otimes id_b$:

  Let $u \in \mathbb{M}^{Y+Z}$. Then

  $$(id_a \otimes id_b)(u) = id_a(\overleftarrow{u}_Y) + id_b(\vec{u}_Y) = \overleftarrow{u}_Y + \vec{u}_Y = u = id_{a\otimes b}(u)$$

- $(g \otimes g') \circ (f \otimes f') = (g \circ f) \otimes (g' \circ f')$:

  This equation is required to hold if both sides are defined. Hence let $f{:}\, a \dashrightarrow b$, $g{:}\, b \dashrightarrow c$, $f'{:}\, a' \dashrightarrow b'$, $g'{:}\, b' \dashrightarrow c'$ and $u \in \mathbb{M}^{Y+Y'}$. We obtain:

  $$\begin{aligned} & (g \otimes g') \circ (f \otimes f')(u) = (g \otimes g')(f(\overleftarrow{u}_Y) + f'(\vec{u}_Y)) \\ = \; & g(f(\overleftarrow{u}_Y)) + g'(f'(\vec{u}_Y)) = ((g \circ f) \otimes (g' \circ f'))(\overleftarrow{u}_Y + \vec{u}_Y) \\ = \; & ((g \circ f) \otimes (g' \circ f'))(u) \end{aligned}$$

2. monoidality:

- $f \otimes id_e = f = id_e \otimes f$:

  Let $f{:}\, a \dashrightarrow b$ and $u \in \mathbb{M}^Y$. It holds that:

  $$\begin{aligned} & (f \otimes id_e)(u) = (f \otimes id_e)(u + e) = f(u) + id_e(e) \\ = \; & f(u) + e = f(u) = e + f(u) \\ = \; & id_e(e) + f(u) = (id_e \otimes f)(e + u) = (id_e \otimes f)(u) \end{aligned}$$

- $(f \otimes g) \otimes h = f \otimes (g \otimes h)$:

  Let $f{:}\, a \dashrightarrow a'$, $g{:}\, b \dashrightarrow b'$ and $h{:}\, c \dashrightarrow c'$ and $u \in \mathbb{M}^{Y+Z+W}$, then

  $$\begin{aligned} & ((f \otimes g) \otimes h)(u) = (f \otimes g)(\overleftarrow{u}_{Y+Z}) + h(\vec{u}_{Y+Z}) \\ = \; & (f(\overleftarrow{(\overleftarrow{u}_{Y+Z})}_Y) + g(\overrightarrow{(\overleftarrow{u}_{Y+Z})}_Y)) + h(\vec{u}_{Y+Z}) \\ = \; & f(\overleftarrow{u}_Y) + (g(\overleftarrow{(\vec{u}_Y)}_Z) + h(\overrightarrow{(\vec{u}_Y)}_Z)) \\ = \; & f(\overleftarrow{u}_Y) + (g \otimes h)(\vec{u}_Y) = (f \otimes (g \otimes h))(u) \end{aligned}$$

  where we use the fact that $\overrightarrow{(\overleftarrow{u}_{Y+Z})}_Y = \overleftarrow{(\vec{u}_Y)}_Z$.

3. naturality:

- $(f' \otimes f) \circ \rho_{a,a'} = \rho_{b,b'} \circ (f \otimes f')$:

  Let $f{:}\, a \dashrightarrow b$ and $f'{:}\, a' \dashrightarrow b'$. Then for $u \in \mathbb{M}^{Y+Y'}$:

  $$\begin{aligned} & (\rho_{b,b'} \circ (f \otimes f'))(u) = \rho_{b,b'}(f(\overleftarrow{u}_Y) + f'(\vec{u}_Y)) \\ = \; & f'(\vec{u}_Y) + f(\overleftarrow{u}_Y) = (f' \otimes f)(\vec{u}_Y + \overleftarrow{u}_Y) = ((f' \otimes f) \circ \rho_{a,a'})(u) \end{aligned}$$

4. symmetry:

- $\rho_{e,e} = id_e$:

  We note that e is the unique function from $\varnothing$ to $\mathbb{M}$ and furthermore $e \otimes e = e + e = e$. Then

$$\rho_{e,e}(e) = \rho_{e,e}(e + e) = e + e = e = id_e(e)$$

- $\rho_{b,a} \circ \rho_{a,b} = id_{a \otimes b}$:

  Let $u \in \mathbb{M}^{Y+Z}$, then:

$$(\rho_{b,a} \circ \rho_{a,b})(u) = \rho_{b,a}(\vec{u}_Y + \vec{u}_Y) = \vec{u}_Y + \vec{u}_Y = u = id_{a \otimes b}(u)$$

- $\rho_{a \otimes b,c} = (\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c})$:

  Let $u \in \mathbb{M}^{Y+Z+W}$, then:

$$\begin{array}{rl} & ((\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c}))(u) \\ = & (\rho_{a,c} \otimes id_b)(id_a(\vec{u}_Y) + \rho_{b,c}(\vec{u}_Y)) \\ = & (\rho_{a,c} \otimes id_b)(\vec{u}_Y + \overrightarrow{(\vec{u}_Y)}_Z + \overleftarrow{(\vec{u}_Y)}_Z) \\ = & \rho_{a,c}(\vec{u}_Y + \vec{u}_{Y+Z}) + id_b(\overleftarrow{(\vec{u}_Y)}_Z) = \vec{u}_{Y+Z} + \vec{u}_Y + \overrightarrow{(\vec{u}_{Y+Z})}_Y \\ = & \vec{u}_{Y+Z} + \vec{u}_{Y+Z} = \rho_{a \otimes b,c}(u) \end{array}$$

  where we use the fact that $\overrightarrow{(\vec{u}_Y)}_Z = \vec{u}_{Y+Z}$ and $\overleftarrow{(\vec{u}_Y)}_Z = \overrightarrow{(\vec{u}_{Y+Z})}_Y$.

5. gs-monoidality:

- $!_e = \nabla_e = id_e$:

  Since e is the unique function of type $\varnothing \to \mathbb{M}$ and $e + e = e$, we obtain:

$$!_e(e) = e = id_e(e) = e = e + e = \nabla_e(e)$$

- coherence axioms:

  For $u \in \mathbb{M}^Y$, we note that $\overleftarrow{(u + u)}_Y = \overrightarrow{(u + u)}_Y = u$.

  - $(id_a \otimes \nabla_a) \circ \nabla_a = (\nabla_a \otimes id_a) \circ \nabla_a$:

    Let $u \in \mathbb{M}^Y$, then:

$$\begin{array}{rl} & ((id_a \otimes \nabla_a) \circ \nabla_a)(u) = (id_a \otimes \nabla_a)(u + u) \\ = & id_a(u) + \nabla_a(u) = u + u + u = \nabla_a(u) + id_a(u) \\ = & (\nabla_a \otimes id_a)(u + u) = (\nabla_a \otimes id_a)(\nabla_a(u)) \end{array}$$

– $id_a = (id_a \otimes !_a) \circ \nabla_a$:

Let $u \in \mathbb{M}^Y$, then:

$$\begin{aligned} & ((id_a \otimes !_a) \circ \nabla_a)(u) = (id_a \otimes !_a)(u + u) \\ = \quad & id_a(u) + !_a(u) = id_a(u) + e = id_a(u) \end{aligned}$$

– $\rho_{a,a} \circ \nabla_a = \nabla_a$:

Let $u \in \mathbb{M}^Y$, then:

$$(\rho_{a,a} \circ \nabla_a)(u) = \rho_{a,a}(u + u) = u + u = \nabla_a(u)$$

- monoidality axioms:

  – $!_{a \otimes b} = !_a \otimes !_b$:

  Let $u \in \mathbb{M}^{Y+Z}$, then:

$$!_{a \otimes b}(u) = e = e + e = !_a(\overleftarrow{u}_Y) + !_b(\vec{u}_Y) = (!_a \otimes !_b)(u)$$

  – $\nabla_a \otimes \nabla_b = (id_a \otimes \rho_{b,a} \otimes id_b) \circ \nabla_{a \otimes b}$:

  Let $u \in \mathbb{M}^{Y+Z}$, then:

$$\begin{aligned} & (id_a \otimes \rho_{b,a} \otimes id_b)(\nabla_{a \otimes b}(u)) = (id_a \otimes \rho_{b,a} \otimes id_b)(u + u) \\ = \quad & (id_a \otimes \rho_{b,a} \otimes id_b)(\overleftarrow{u}_Y + \vec{u}_Y + \overleftarrow{u}_Y + \vec{u}_Y) \\ = \quad & \overleftarrow{u}_Y + \overleftarrow{u}_Y + \vec{u}_Y + \vec{u}_Y = \nabla_a(\overleftarrow{u}_Y) + \nabla_b(\vec{u}_Y) \\ = \quad & (\nabla_a \otimes \nabla_b)(\overleftarrow{u}_Y + \vec{u}_Y) = (\nabla_a \otimes \nabla_b)(u) \end{aligned}$$

□

Theorem A.3.5 (Theorem 4.6.2)**.** *The category $\mathbb{A}_f$ with the following operators is gs-monoidal:*

1. *The tensor $\otimes$ on objects $a \in \mathbb{M}^Y$ and $b \in \mathbb{M}^Z$ is again defined as $a \otimes b = a + b$. On arrows $f: a \dashrightarrow b$ and $g: a' \dashrightarrow b'$ (where $a' \in \mathbb{M}^{Y'}$, $b' \in \mathbb{M}^{Z'}$ and $f: \mathcal{P}([Y]^a) \to \mathcal{P}([Z]^{b'})$, $g: \mathcal{P}([Y']^{a'}) \to \mathcal{P}([Z']^{b'})$ are the underlying functions), the tensor is given by*

$$f \otimes g: \mathcal{P}([Y+Y']^{a+a'}) \to \mathcal{P}([Z+Z']^{b+b'}), \quad (f \otimes g)(U) = f(\overleftarrow{U}_Y) \cup_Z g(\overrightarrow{U}_Y)$$

where $\overleftarrow{U}_Y = U \cap \{1, \dots, |Y|\}$ and $\overrightarrow{U}_Y = \{k \mid |Y| + k \in U\}$. Furthermore:

$$U \cup_Y V = U \cup \{|Y| + k \mid k \in V\} \quad \textit{(where } U \subseteq Y\textit{)}$$

2. *The symmetry $\rho_{a,b}: a \otimes b \dashrightarrow b \otimes a$ for $a \in \mathbb{M}^Y$, $b \in \mathbb{M}^Z$ is defined for $U \subseteq [Y+Z]^{a+b}$ as*

$$\rho_{a,b}(U) = \overrightarrow{U}_Y \cup_Z \overleftarrow{U}_Y \subseteq [Z+Y]^{b+a}$$

3. *The unit e is again the unique mapping $e: \varnothing \to \mathbb{M}$.*

4. *The duplicator $\nabla_a: a \dashrightarrow a \otimes a$ for $a \in \mathbb{M}^Y$ is defined for $U \subseteq [Y]^a$ as*

$$\nabla_a(U) = U \cup_Y U \subseteq [Y+Y]^{a+a}.$$

5. *The discharger $!_a: a \dashrightarrow e$ for $a \in \mathbb{M}^Y$ is defined for $U \subseteq [Y]^a$ as $!_a(U) = \varnothing$.*

Proof. Let $a \in \mathbb{M}^Y$, $a' \in \mathbb{M}^{Y'}$, $b \in \mathbb{M}^Z$, $b' \in \mathbb{M}^{Z'}$, $c \in \mathbb{M}^W$, $c' \in \mathbb{M}^{W'}$ be objects in $\mathbb{A}_f$.

We know that $\mathbb{A}_f$ is a well-defined category from Lemma 4.3.2. We note that, disjoint unions of monotone functions are monotone, making the tensor well-defined.

We now verify the axioms of gs-monoidal categories (see Definition 2.4.9), the calculations are mostly straightforward.

We will in the following often use the fact that $\overleftarrow{U}_Y \cup_Y \overrightarrow{U}_Y = U$ whenever $U \in \mathcal{P}([Z]^b)$ and $Y \subseteq Z$.

1. functoriality of tensor:
   - $id_{a\otimes b} = id_a \otimes id_b$:

     Let $U \subseteq [Y+Z]^{a+b}$, then:

$$\begin{aligned} & (id_a \otimes id_b)(U) = (id_a \otimes id_b)(\overleftarrow{U}_Y \cup_Y \overrightarrow{U}_Y) \\ = \;& id_a(\overleftarrow{U}_Y) \cup_Y id_b(\overrightarrow{U}_Y) = \overleftarrow{U}_Y \cup_Y \overrightarrow{U}_Y = U = id_{a\otimes b}(U) \end{aligned}$$

   - $(g \otimes g') \circ (f \otimes f') = (g \circ f) \otimes (g' \circ f')$:

Let $f\colon a \dashrightarrow b$, $g\colon b \dashrightarrow c$, $f'\colon a' \dashrightarrow b'$, $g'\colon b' \dashrightarrow c'$ and $u \in \mathbb{M}^{Y+Y'}$. We obtain:

$$\begin{aligned}
& ((g \otimes g') \circ (f \otimes f'))(U) = (g \otimes g')(f(\overleftarrow{U}_Y) \cup_Z f'(\vec{U}_Y)) \\
= \;& g(f(\overleftarrow{U}_Y)) \cup_W g'(f'(\vec{U}_Y)) = ((g \circ f) \otimes (g' \circ f'))(\overleftarrow{U}_Y \cup_Y \vec{U}_Y) \\
= \;& ((g \circ f) \otimes (g' \circ f'))(U)
\end{aligned}$$

2. monoidality:
   - $f \otimes id_e = f = id_e \otimes f$:

     Let $f\colon a \dashrightarrow b$ and $U \subseteq [Y]^a$. It holds that:

$$\begin{aligned}
& (f \otimes id_e)(U) = f(\overleftarrow{U}_Y) \cup_Z id_e(\vec{U}_Y) = f(U) \cup_Z id_e(\varnothing) = f(U) \cup_Z \varnothing \\
= \;& f(U) = \varnothing \cup_\varnothing f(U) = id_e(\varnothing) \cup_\varnothing f(U) = id_e(\overleftarrow{U}_\varnothing) \cup_\varnothing f(\vec{U}_\varnothing) \\
= \;& (id_e \otimes f)(U)
\end{aligned}$$

     where we use the fact that $\overleftarrow{U}_Y = U$ and $\vec{U}_Y = \varnothing$, since $U \subseteq Y$, as well as $\overleftarrow{U}_\varnothing = \varnothing$ and $\vec{U}_\varnothing = U$.
   - $(f \otimes g) \otimes h = f \otimes (g \otimes h)$:

     Let $f\colon a \dashrightarrow a'$, $g \in\colon b \dashrightarrow b'$ and $h\colon c \dashrightarrow c'$ and $U \subseteq [Y + Z + W]^{a+b+c}$. Then:

$$\begin{aligned}
& ((f \otimes g) \otimes h)(U) = (f \otimes g)(\overleftarrow{U}_{Y+Z}) \cup_{Y'+Z'} h(\vec{U}_{Y+Z}) \\
= \;& \big(f(\overleftarrow{(\overleftarrow{U}_{Y+Z})_Y}) \cup_{Y'} g(\overrightarrow{(\overleftarrow{U}_{Y+Z})_Y})\big) \cup_{Y'+Z'} h(\vec{U}_{Y+Z}) \\
= \;& \big(f(\overleftarrow{U}_Y) \cup_{Y'} g(\overleftarrow{(\vec{U}_Y)_Z})\big) \cup_{Y'+Z'} h(\vec{U}_{Y+Z}) \\
= \;& f(\overleftarrow{U}_Y) \cup_{Y'} \big(g(\overleftarrow{(\vec{U}_Y)_Z}) \cup_{Z'} h(\overrightarrow{(\vec{U}_Y)_Z})\big) \\
= \;& f(\overleftarrow{U}_Y) \cup_{Y'} (g \otimes h)(\vec{U}_Y) = (f \otimes (g \otimes h))(U)
\end{aligned}$$

     where we use the fact that $\overleftarrow{(\overleftarrow{U}_{Y+Z})_Y} = \overleftarrow{U}_Y$, $\overrightarrow{(\overleftarrow{U}_{Y+Z})_Y} = \overleftarrow{(\vec{U}_Y)_Z}$ and $\vec{U}_{Y+Z} = \overrightarrow{(\vec{U}_Y)_Z}$.

3. naturality:
   - $(f' \otimes f) \circ \rho_{a,a'} = \rho_{b,b'} \circ (f \otimes f')$:

     Let $f\colon a \dashrightarrow b$ and $f'\colon a' \dashrightarrow b'$. Then for $U \subseteq [Y + Y']^{a+a'}$ it holds that:

$$\begin{aligned}
& (\rho_{b,b'} \circ (f \otimes f'))(U) \\
= \;& \rho_{b,b'}(f(\overleftarrow{U}_Y) \cup_Z f'(\vec{U}_Y)) \\
= \;& \overrightarrow{(f(\overleftarrow{U}_Y) \cup_Z f'(\vec{U}_Y))_Z} \cup_{Z'} \overleftarrow{(f(\overleftarrow{U}_Y) \cup_Z f'(\vec{U}_Y))_Z} \\
= \;& f'(\vec{U}_Y) \cup_{Z'} f(\overleftarrow{U}_Y) \\
= \;& f'(\overleftarrow{(\vec{U}_Y \cup_{Y'} \overleftarrow{U}_Y)_{Y'}}) \cup_{Z'} f(\overrightarrow{(\vec{U}_Y \cup_{Y'} \overleftarrow{U}_Y)_{Y'}})
\end{aligned}$$

$$= \quad (f' \otimes f)(\vec{U}_Y \cup_{Y'} \overleftarrow{U}_Y) = (f' \otimes f)(\rho_{a,a'}(U))$$

where we use the fact that $\overleftarrow{(U \cup_Y V)}_Y = U$ and $\overrightarrow{(U \cup_Y V)}_Y = V$.

4. symmetry:

- $\rho_{e,e} = id_e$:

  Note that the only possibly argument is $\varnothing$ and hence:

  $$\rho_{e,e}(\varnothing) = \vec{\varnothing}_\varnothing \cup_\varnothing \overleftarrow{\varnothing}_\varnothing = \varnothing \cup_\varnothing \varnothing = \varnothing = id_e(\varnothing)$$

- $\rho_{b,a} \circ \rho_{a,b} = id_{a \otimes b}$:

  Let $U \subseteq [Y + Z]^{a+b}$, then:

  $$\begin{aligned} & (\rho_{b,a} \circ \rho_{a,b})(U) = \rho_{b,a}(\vec{U}_Y \cup_Z \overleftarrow{U}_Y) \\ = \quad & \overrightarrow{(\vec{U}_Y \cup_Z \overleftarrow{U}_Y)}_Z \cup_Y \overleftarrow{(\vec{U}_Y \cup_Z \overleftarrow{U}_Y)}_Z = \overleftarrow{U}_Y \cup_Y \vec{U}_Y = U \\ = \quad & id_{a \otimes b}(U) \end{aligned}$$

- $\rho_{a \otimes b,c} = (\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c})$:

  Let $U \subseteq [Y + Z + W]^{a+b+c}$, then:

  $$\begin{aligned} & ((\rho_{a,c} \otimes id_b) \circ (id_a \otimes \rho_{b,c}))(U) \\ = \quad & (\rho_{a,c} \otimes id_b)(id_a(\overleftarrow{U}_Y) \cup_Y \rho_{b,c}(\vec{U}_Y)) \\ = \quad & (\rho_{a,c} \otimes id_b)(\overleftarrow{U}_Y \cup_Y (\overrightarrow{(\vec{U}_Y)}_Z \cup_W \overleftarrow{(\vec{U}_Y)}_Z)) \\ = \quad & (\rho_{a,c} \otimes id_b)(\overleftarrow{U}_Y \cup_Y (\vec{U}_{Y+Z} \cup_W \overleftarrow{(\vec{U}_Y)}_Z)) \\ = \quad & (\rho_{a,c} \otimes id_b)((\overleftarrow{U}_Y \cup_Y \vec{U}_{Y+Z}) \cup_{Y+W} \overleftarrow{(\vec{U}_Y)}_Z)) \\ = \quad & \rho_{a,c}(\overleftarrow{U}_Y \cup_Y \vec{U}_{Y+Z}) \cup_{W+Y} id_b(\overleftarrow{(\vec{U}_Y)}_Z) \\ = \quad & (\vec{U}_{Y+Z} \cup_W \overleftarrow{U}_Y) \cup_{W+Y} \overleftarrow{(\vec{U}_Y)}_Z \\ = \quad & \vec{U}_{Y+Z} \cup_W (\overleftarrow{U}_Y \cup_Y \overleftarrow{(\vec{U}_Y)}_Z) \\ = \quad & \vec{U}_{Y+Z} \cup_W (\overleftarrow{(\overleftarrow{U}_{Y+Z})}_Y \cup_Y \overrightarrow{(\overleftarrow{U}_{Y+Z})}_Y) \\ = \quad & \vec{U}_{Y+Z} \cup_W \overleftarrow{U}_{Y+Z} = \rho_{a \otimes b,c}(U) \end{aligned}$$

  where we use the fact that $\overrightarrow{(\vec{U}_Y)}_Z = \vec{U}_{Y+Z}$, $\overleftarrow{U}_Y = \overleftarrow{(\overleftarrow{U}_{Y+Z})}_Y$ and $\overleftarrow{(\vec{U}_Y)}_Z = \overrightarrow{(\overleftarrow{U}_{Y+Z})}_Y$.

5. gs-monoidality:

- $!_e = \nabla_e = id_e$:

  In this case $\varnothing$ is the only possible argument and we have:

$$!_e(\varnothing) = \varnothing = id_e(\varnothing) = \varnothing = \varnothing \cup_{\varnothing} \varnothing = \nabla_e(\varnothing)$$

- coherence axioms:

  For $U \subseteq [Y]^a$, we note that $\overleftarrow{(U \cup_Y U)}_Y = \overrightarrow{(U \cup_Y U)}_Y = U$.

  - $(id_a \otimes \nabla_a) \circ \nabla_a = (\nabla_a \otimes id_a) \circ \nabla_a$:

    Let $U \subseteq [Y]^a$, then:

$$\begin{aligned} & ((id_a \otimes \nabla_a) \circ \nabla_a)(U) = (id_a \otimes \nabla_a)(U \cup_Y U) = id_a(U) \cup_Y \nabla_a(U) \\ = \;& U \cup_Y (U \cup_Y U) = (U \cup_Y U) \cup_{Y+Y} U = \nabla_a(U) \cup_{Y+Y} id_a(U) \\ = \;& (\nabla_a \otimes id_a)(U \cup_Y U) = (\nabla_a \otimes id_a)(\nabla_a(U)) \end{aligned}$$

  - $id_a = (id_a \otimes !_a) \circ \nabla_a$:

    Let $U \subseteq [Y]^a$, then:

$$\begin{aligned} & ((id_a \otimes !_a) \circ \nabla_a)(U) = (id_a \otimes !_a)(U \cup_Y U) = id_a(U) \cup_Y !_a(U) \\ = \;& id_a(U) \cup_Y \varnothing = id_a(U) \end{aligned}$$

  - $\rho_{a,a} \circ \nabla_a = \nabla_a$:

    Let $U \subseteq [Y]^a$, then:

$$(\rho_{a,a} \circ \nabla_a)(U) = \rho_{a,a}(U \cup_Y U) = U \cup_Y U = \nabla_a(U)$$

- monoidality axioms:

  - $!_{a \otimes b} = !_a \otimes !_b$:

    Let $U \subseteq [Y+Z]^{a+b}$, then:

$$\begin{aligned} !_{a\otimes b}(U) &= \varnothing = \varnothing \cup_{\varnothing} \varnothing = !_a(\overleftarrow{U}_Y) \cup_{\varnothing} !_b(\overrightarrow{U}_Y) \\ &= (!_a \otimes !_b)(\overleftarrow{U}_Y \cup_Y \overrightarrow{U}_Y) = (!_a \otimes !_b)(U) \end{aligned}$$

  - $\nabla_a \otimes \nabla_b = (id_a \otimes \rho_{b,a} \otimes id_b) \circ \nabla_{a \otimes b}$:

    Let $U \subseteq [Y+Z]^{a+b}$, then:

$$\begin{aligned} & (id_a \otimes \rho_{b,a} \otimes id_b)(\nabla_{a\otimes b}(U)) = (id_a \otimes (\rho_{b,a} \otimes id_b))(U \cup_{Y+Z} U) \\ = \;& id_a(\overleftarrow{(U \cup_{Y+Z} U)}_Y) \cup_Y (\rho_{b,a} \otimes id_b)(\overrightarrow{(U \cup_{Y+Z} U)}_Y) \\ = \;& id_a(\overleftarrow{U}_Y) \cup_Y (\rho_{b,a} \otimes id_b)(\overrightarrow{(U \cup_{Y+Z} U)}_Y) \end{aligned}$$

$$
\begin{aligned}
&= \overleftarrow{U}_Y \cup_Y (\rho_{b,a} \otimes id_b)(\overrightarrow{(U \cup_{Y+Z} U)}_Y)\\
&= \overleftarrow{U}_Y \cup_Y (\rho_{b,a} \otimes id_b)((\overleftarrow{U}_Y \cup_Z \overleftarrow{U}_Y) \cup_{Z+Y} \overleftarrow{U}_Y)\\
&= \overleftarrow{U}_Y \cup_Y ((\overleftarrow{U}_Y \cup_Y \overleftarrow{U}_Y) \cup_{Y+Z} \overleftarrow{U}_Y) = (\overleftarrow{U}_Y \cup_Y \overleftarrow{U}_Y) \cup_{Y+Y} (\overleftarrow{U}_Y \cup_Z \overleftarrow{U}_Y)\\
&= \nabla_a(\overleftarrow{U}_Y) \cup_{Y+Y} \nabla_b(\overleftarrow{U}_Y) = (\nabla_a \otimes \nabla_b)(\overleftarrow{U}_Y \cup_Y \overleftarrow{U}_Y) = (\nabla_a \otimes \nabla_b)(U)
\end{aligned}
$$

where we use the fact that $\overleftarrow{(U \cup_{Y+Z} U)}_Y = \overleftarrow{U}_Y$ and $\overrightarrow{(U \cup_{Y+Z} U)}_Y = (\overleftarrow{U}_Y \cup_Z \overleftarrow{U}_Y) \cup_{Z+Y} \overleftarrow{U}_Y$.

□

Theorem A.3.6 (Theorem 4.6.3). *$\#\colon \mathbb{C}_f \to \mathbb{A}_f$ is a gs-monoidal functor.*

Proof. We write $e', \otimes', !', \nabla', \rho'$ for the corresponding operators in category $\mathbb{A}_f$. Note that by definition $e = e'$ and $\otimes$, $\otimes'$ agree on objects.

First, categories $\mathbb{C}_f$ and $\mathbb{A}_f$ are gs-monoidal categories by Theorem 4.6.1 and 4.6.2. Furthermore we have to verify that (cf. Definition 2.4.10):

1. monoidality:

   - $\#(e) = e'$:

     We have $\#(e) = e = e'$

   - $\#(a \otimes b) = \#(a) \otimes' \#(b)$:

     We have:

     $$\#(a \otimes b) = a \otimes b = \#(a) \otimes' \#(b)$$

2. symmetry:

   - $\#(\rho_{a,b}) = \rho'_{\#(a),\#(b)}$:

     Let $U \subseteq [Y+Z]^{a+b}$, then for sufficiently small $\delta \sqsupset 0$ (note that such δ exists due to finiteness):

     $$
     \begin{aligned}
     &\#(\rho_{a,b})(U)\\
     &= (\rho_{a,b})_{\#}^{a+b,\delta}(U)\\
     &= \{w \in [Z+Y]^{b+a} \mid \rho_{a,b}(a+b)(w) \ominus \rho_{a,b}((a+b) \ominus \delta_U)(w) \sqsupseteq \delta\}\\
     &= \{w \in [Z+Y]^{b+a} \mid (b+a)(w) \ominus ((b+a) \ominus \delta_{\rho'_{a,b}(U)})(w) \sqsupseteq \delta\}\\
     &= \rho'_{a,b}(U) = \rho'_{\#(a),\#(b)}(U)
     \end{aligned}
     $$

     since $\rho_{a,b}$ distributes over componentwise subtraction and $\rho_{a,b}(\delta_U) = \delta_{\rho'_{a,b}(U)}$. The second-last equality holds since for all w in the set we have $(b+a)(w) \sqsupset 0$.

3. gs-monoidality:

   - $\#(!_a) = !'_{\#(a)}$:

     Let $U \subseteq [Y]^a$, then for some δ:

     $$\#(!_a)(U) = (!_a)_{\#}^{a,\delta}(U) = \varnothing = !'_a(U) = !'_{\#(a)}(U)$$

     since the codomain of $(!_a)_{\#}^{a,\delta}(U)$ is $\mathcal{P}(\varnothing)$ and hence the only possible value for $(!_a)_{\#}^{a,\delta}(U)$ is $\varnothing$.

   - $\#(\nabla_a) = \nabla'_{\#(a)}$:

     Let $U \subseteq [Y]^a$, then for sufficiently small $\delta \sqsupset 0$:

     $$\begin{aligned}\#(\nabla_a)(U) &= (\nabla_a)_{\#}^{a,\delta}(U)\\ &= \{w \in [Y+Y]^{a+a} \mid \nabla_a(a)(w) \ominus \nabla_a(a \ominus \delta_U)(w) \sqsupseteq \delta\}\\ &= \{w \in [Y+Y]^{a+a} \mid (a+a)(w) \ominus ((a+a) \ominus \delta_{\nabla'_a(U)})(w) \sqsupseteq \delta\}\\ &= \nabla'_a(U) = \nabla'_{\#(a)}(U)\end{aligned}$$

     since ∇_a distributes over componentwise subtraction and $\nabla_a(\delta_U) = \delta_{\nabla'_a(U)}$. The second-last equality holds since for all w in the set we have $(a+a)(w) \sqsupset 0$.

□

A.4. Proofs of Chapter 5

Proposition A.4.1 (Proposition 5.2.4)**.** *Let Y be a finite set and $\mathbb{M}$ a complete MV-chain. Let $f\colon \mathbb{M}^Y \to \mathbb{M}^Y$ be a function and $H_{\min}\colon Y \to \mathcal{P}_f(\mathbb{M}^Y \to \mathbb{M})$ a given min-decomposition such that, for all $y \in Y$, all functions $h \in H_{\min}(y)$ are non-expansive. Then f is non-expansive and the approximation $f_{\#}^a(Y')\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ is given by*

$$f_{\#}^a(Y') = \{y \in [Y]^{f(a)} \mid \exists h\left(h = \arg\min_{h' \in H_{\min}(y)} h'(a) \ \wedge \ h_{\#}^a(Y') \neq \varnothing\right)\}$$

for $a \in \mathbb{M}^Y$ and $Y' \subseteq [Y]^a$.

Proof. We first show that f is non-expansive by proving that it can be expressed as a composition of non-expansive functions. (Recall that function composition and disjoint union preserve non-expansiveness.)

For all $y \in Y$, let $H_{\min}(y) = \{h_y^1, \ldots, h_y^{k_y}\}$ and let $I_y = \{1, \ldots, k_y\}$ be the corresponding index set. We have $h_y^i\colon \mathbb{M}^Y \to \mathbb{M}$ for each $i \in I_y$, where – for convenience – we view each function h_y^i as being of type $\mathbb{M}^Y \to \mathbb{M}^{\{i\}}$, where $\{i\}$ is the singleton set containing i.

We introduce auxiliary functions $g_y\colon \mathbb{M}^Y \to \mathbb{M}^{I_y}$ and $g\colon \mathbb{M}^Y \to \mathbb{M}^I$ (with $I = \biguplus_{y\in Y} I_y$) defined as below, where $a \in \mathbb{M}^Y$ and $i \in I_y$:

$$g_y = \biguplus_{j\in I_y} h_y^j \quad g_y(a)(i) = h_y^i(a) \qquad g = \biguplus_{y\in Y} g_y \quad g(a)(i) = g_y(a)(i)$$

Next we define $u\colon I \to Y$ where $u(i) = y$ for all $i \in I_y$. This allows decomposition of f as

$$f = \min_u \circ g$$

with $\min_u\colon \mathbb{M}^I \to \mathbb{M}^Y$. More intuitively, given $a \in \mathbb{M}^Y$ and $y \in Y$, we have

$$f(a)(y) = \min_{i, u(i)=y} g_y(a)(i) = \min_{i\in I_y} h_y^i(a)$$

and

$$f(a) = \min_u(g(a)) = \min_u(\biguplus_{y\in Y} g_y(a)) = \min_u(\biguplus_{y\in Y} \biguplus_{u(i)=y} h_y^i(a)).$$

Next, we want to express the approximation $f_\#^a\colon \mathcal{P}([Y]^a) \to \mathcal{P}([Y]^{f(a)})$ for $a \in \mathbb{M}^Y$ in terms of the approximations of the components $(h_y^i)_\#^a\colon \mathcal{P}([Y]^a) \to \mathcal{P}([\{i\}]^{h_y^i(a)})$.

In the following $Y' \subseteq [Y]^a$ is some subset of $[Y]^a$. We note that $(h_y^i)_\#^a(Y')$ is either $\{i\}$ or the empty set $\emptyset$.

First observe, that $(g_y)_\#^a\colon \mathcal{P}([Y]^a) \to \mathcal{P}([I_y]^{g_y(a)})$, as recalled in Table 3.1, is given by

$$(g_y)_\#^a(Y') = \bigcup_{i\in I_y} \Big((h_y^i)_\#^a(Y') \Big).$$

Moreover $g_\#^a\colon \mathcal{P}([Y]^a) \to \mathcal{P}([I]^{g(a)})$ is given by

$$g_\#^a(Y') = \bigcup_{y\in Y} (g_y)_\#^a = \bigcup_{y\in Y} \bigcup_{i\in I_y} \Big((h_y^i)_\#^a(Y') \Big)$$

Hence we obtain for $i \in I_y$:

$$i \in g_\#^a(Y') \text{ iff } i \in (g_y)_\#^a(Y') \text{ iff } i \in (h_y^i)_\#^a(Y') \text{ iff } (h_y^i)_\#^a(Y') \neq \emptyset.$$

Finally, we can conclude

$$\begin{aligned}
f_\#^a(Y') &= \{y \in [Y]^{f(a)} \mid Min_{g(a)|u^{-1}(y)} \cap g_\#^a(Y') \neq \emptyset\} \\
&= \{y \in [Y]^{f(a)} \mid \exists i \big(i = \arg\min_{j\in I_y} g(a)(j) \ \wedge \ i \in g_\#^a(Y')\big)\} \\
&= \{y \in [Y]^{f(a)} \mid \exists i \big(i = \arg\min_{j\in I_y} g_y(a)(j) \ \wedge \ i \in (g_y)_\#^a(Y')\big)\} \\
&= \{y \in [Y]^{f(a)} \mid \exists i \big(i = \arg\min_{j\in I_y} h_y^j(a) \ \wedge \ i \in (h_y^i)_\#^a(Y')\big)\} \\
&= \{y \in [Y]^{f(a)} \mid \exists h \big(h = \arg\min_{h'\in H_{\min}(y)} h'(a) \ \wedge \ h_\#^a(Y') \neq \emptyset\big)\}
\end{aligned}$$

□

Lemma A.4.2 (Lemma 5.3.2). *The linear program above computes $\mu\mathcal{V}_{C_{\mathsf{Max}}}$.*

Proof. Given a strategy σ for Max (corresponding to C_{Max}), we can determine $a = \mu\mathcal{V}_\sigma$ by solving the following linear program:

$$
\begin{aligned}
&\max \sum_{v \in V} a(v) && \\
&a(v) = 0 && \forall v \in Q_\sigma \\
&a(v) = a(\sigma_{C_{\mathsf{Max}}}) && \forall v \in V_{\mathsf{Max}} \setminus Q_\sigma \\
&a(v) \leq a(u) && \forall v \in V_{\mathsf{Min}} \setminus Q_\sigma, \forall (v,u) \in E \\
&a(v) = \sum_{v' \in V} p(v)(v') \cdot a(v') && \forall v \in V_{\mathsf{Av}} \setminus Q_\sigma \\
&a(v) = c(v) && \forall v \in V_{\mathsf{Sink}}
\end{aligned}
$$

The set Q_σ contains those nodes which will guarantee a non-terminating play if Min plays optimally, given the fixed max-strategy σ.

The set Q_σ can again be computed via fixpoint-iteration by computing the greatest fixpoint of q_σ via Kleene iteration on $\mathcal{P}(V)$ from above:

$$
\begin{aligned}
q_\sigma : \mathcal{P}(V) &\to \mathcal{P}(V) \\
q_\sigma(V') &= \{v \in V \mid (v \in V_{\mathsf{Min}} \wedge succ(v) \cap V' \neq \emptyset) \vee (v \in V_{\mathsf{Max}} \wedge \sigma(v) \in V') \\
&\qquad \vee (v \in V_{\mathsf{Av}} \wedge supp(p(v)) \subseteq V')\}
\end{aligned}
$$

It is easy to see that $Q_\sigma = \nu q_\sigma$ contains all those nodes from which Min can force a non-terminating play and hence achieve payoff 0. (Note that there are further nodes that guarantee payoff 0 – namely sinks with that payoff and nodes which can reach such sinks – but those will obtain value 0 in any case.)

We now show that this linear program computes $\mu\mathcal{V}_\sigma$ (= $\mu\mathcal{V}_{C_{\mathsf{Max}}}$ where σ corresponds to C_{Max}), which is given by

$$
\mathcal{V}_\sigma(a)(v) = \begin{cases} a(\sigma(v)) & \text{if } v \in V_{\mathsf{Max}} \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}
$$

for $a \in [0,1]^V$ and $v \in V$. First, by requiring $a(v) \leq a(u)$ for all $v \in V_{\mathsf{Min}}$, $u \in succ(v)$, we guarantee $a(v) = \min_{u \in succ(v)} a(u)$ since we maximise. Hence we obtain the greatest fixpoint of the following function $\mathcal{V}'_\sigma : [0,1]^V \to [0,1]^V$:

$$
V'_\sigma(a)(v) = \begin{cases} 0 & \text{if } v \in Q_\sigma \\ a(\sigma(v)) & \text{if } v \in V_{\mathsf{Max}} \setminus Q_\sigma \\ \min_{v' \in succ(v)} a(v') & \text{if } v \in V_{\mathsf{Min}} \setminus Q_\sigma \\ \sum_{v' \in V} p(v)(v') \cdot a(v') & \text{if } v \in V_{\mathsf{Av}} \setminus Q_\sigma \\ c(v) & \text{if } v \in V_{\mathsf{Sink}} \end{cases}
$$

It is easy to show that the least fixpoints of $\mathcal{V}'_\sigma$ and $\mathcal{V}_\sigma$ agree, i.e., $\mu\mathcal{V}'_\sigma$ and $\mu\mathcal{V}_\sigma$:

- $\mu\mathcal{V}'_\sigma \le \mu\mathcal{V}_\sigma$ can be shown by observing that $\mathcal{V}'_\sigma \le \mathcal{V}_\sigma$.

- $\mu\mathcal{V}_\sigma \le \mu\mathcal{V}'_\sigma$ can be shown by proving that $\mu\mathcal{V}'_\sigma$ is a pre-fixpoint of $\mathcal{V}_\sigma$, which can be done via a straightforward case analysis.

  We have to show $\mathcal{V}_\sigma(\mu\mathcal{V}'_\sigma)(v) \le \mu\mathcal{V}'_\sigma(v)$ for all $v \in V$. We only spell out the case where $v \in V_{\mathsf{Av}}$, the other cases are similar. In this case either $v \notin Q_\sigma$, which means that

  $$\mathcal{V}_\sigma(\mu\mathcal{V}'_\sigma)(v) = \mathcal{V}'_\sigma(\mu\mathcal{V}'_\sigma)(v) = \mu\mathcal{V}'_\sigma(v).$$

  If instead $v \in Q_\sigma$, we have that $\mathit{supp}(p(v)) \subseteq Q_\sigma$ and so $\mu\mathcal{V}'_\sigma(v') = 0$ for all $v' \in \mathit{supp}(p(v))$. Hence

  $$\mathcal{V}_\sigma(\mu\mathcal{V}'_\sigma)(v) = \sum_{v' \in V} p(v)(v') \cdot \mu\mathcal{V}'_\sigma(v') = 0 = \mu\mathcal{V}'_\sigma(v)$$

If we can now show that $\mathcal{V}'_\sigma$ has a unique fixpoint, we are done. The argument for this goes as follows: assume that this function has another fixpoint a' different from $\mu\mathcal{V}'_\sigma$. Clearly $[V]^{a'} \cap Q_\sigma = \varnothing$, where $[V]^{a'} = \{v \in V \mid a'(v) \ne 0\}$. Hence, if we compare $(\mathcal{V}'_\sigma)^a_\#\colon \mathcal{P}([V]^a) \to \mathcal{P}([V]^{\mathcal{V}'_\sigma(a)})$ (defined analogously to Lemma 3.6.30) and q_σ above, we observe that $(\mathcal{V}'_\sigma)^{a'}_\# \subseteq q_\sigma|_{\mathcal{P}([V]^{a'})}$. (Both functions coincide, apart from their treatment of nodes $v \in V_{\mathsf{Min}}$, where $q_\sigma(V')$ contains v whenever one of its successors is contained in V', whereas $(\mathcal{V}'_\sigma)^{a'}_\#(V')$ additionally requires that the value of this successor is minimal.) Since a' is not the least fixpoint we have by Lemma 3.4.2 that

$$\varnothing \ne \nu(\mathcal{V}'_\sigma)^{a'}_\# \subseteq \nu(q_\sigma|_{\mathcal{P}([V]^{a'})}) \subseteq \nu q_\sigma = Q_\sigma.$$

This is a contradiction, since $[V]^{a'} \cap Q_\sigma = \varnothing$ as observed above.

This shows that $\mathcal{V}'_\sigma$ has a unique fixpoint and completes the proof. Note that if we do not explicitly require that the values of all nodes in Q_σ are 0, $\mathcal{V}'_\sigma$ will potentially have several fixpoints and the linear program would not characterise the least fixpoint. □

Lemma A.4.3 (Lemma 5.3.7)**.** *Let* $PA = (S, \eta, L, \ell)$ *be a probabilistic automaton and let* $H_{\min}$ *be the min-decomposition of* $\mathcal{M}$*. Given a strategy* C *in* H_{min} *and* $d : Y \times Y \to [0,1]$*, a strategy* $C'(y) = \arg\min_{h \in H_{\min}(y)} h(d)$ *can be defined as follows: for* $(s,t) \in S \times S$

- *if* $\ell(s) \neq \ell(t)$ *then* $C'(s,t) = C(s,t)$
- *if* $\ell(s) = \ell(t)$ *then* $C'(s,t) = h_{R',f'}$ *where*

$$R' = \arg\min_{R \in \mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta') \in R} K(d)(\beta,\beta')$$

and for $(\beta,\beta') \in R'$*:*

$$f'(\beta,\beta') = \arg \min_{\omega \in \Omega_V(\beta,\beta')} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v).$$

Proof. Let $(s,t) \in S \times S$. If $\ell(s) \neq \ell(t)$ then $H_{\min}(s,t) = \{h_1\}$, hence the only possible choice is $C'(s,t) = h_1 = C(s,t)$.

If instead $\ell(s) = \ell(t)$ then $C'(s,t) = h_{R',f'}$ is chosen in a way that minimises

$$h_{R',f'}(d) = \max_{(\beta,\beta') \in R'} \sum_{u,v \in S} d(u,v) \cdot f'(\beta,\beta')(u,v) \tag{A.3}$$

In order to minimise the value above, whatever R' will be, for all $(\beta,\beta') \in R'$ the choice of $f'(\beta,\beta')$ should minimise $\sum_{u,v \in S} d(u,v) \cdot \omega(u,v)$. Formally, we can define $F : \mathcal{D}(S) \times \mathcal{D}(S) \to \mathcal{D}(S \times S)$ as

$$F(\beta,\beta') = \arg \min_{\omega \in \Omega_V(\beta,\beta')} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v).$$

Then the set-coupling R' can be

$$\begin{aligned} R' &= \arg\min_{R \in \mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta') \in R} \sum_{u,v \in S} d(u,v) \cdot F'(\beta,\beta')(u,v) \\ &= \arg\min_{R \in \mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta') \in R} \min_{\omega \in \Omega_V(\beta,\beta')} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \\ &= \arg\min_{R \in \mathcal{R}(\delta(s),\delta(t))} \max_{(\beta,\beta') \in R} K(d)(\beta,\beta') \end{aligned}$$

and finally we can define $f' = F_{|R'}$, i.e., explicitly, for all $(\beta,\beta') \in R'$

$$f'(\beta,\beta') = \arg \min_{\omega \in \Omega_V(\beta,\beta')} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v).$$

as desired. □

Bibliography

[ABdMS21] David Auger, Xavier Badin de Montjoye, and Yann Strozecki. A Generic Strategy Improvement Method for Simple Stochastic Games. In *Proc. of MFCS 2021*, volume 202 of *LIPIcs*, pages 12:1–12:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[BBKK18] Paolo Baldan, Filippo Bonchi, Henning Kerstan, and Barbara König. Coalgebraic Behavioral Metrics. *Logical Methods in Computer Science*, 14(3), 2018. Selected Papers of the 6th Conference on Algebra and Coalgebra in Computer Science (CALCO 2015).

[BBL+21] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Radu Mardare, Qiyi Tang, and Franck van Breugel. Computing Probabilistic Bisimilarity Distances for Probabilistic Automata. *Log. Methods Comput. Sci.*, 17(1), 2021.

[BBLM17] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. On-the-Fly Exact Computation of Bisimilarity Distances. *Logical Methods in Computer Science*, 13(2:13):1–25, 2017.

[BC10] Luboš Brim and Jakub Chaloupka. Using Strategy Improvement to Stay Alive. *Electronic Proceedings in Theoretical Computer Science*, 25:40–54, 2010.

[BCD+11] Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal Methods Syst. Des.*, 38(2):97–118, 2011.

[Beh14] E. Behrends. *Introduction to Markov Chains: With Special Emphasis on Rapid Mixing*. Advanced Lectures in Mathematics. Vieweg+Teubner Verlag, 2014.

[BEK+23a] Paolo Baldan, Richard Eggert, Barbara König, Timo Matt, and Tommaso Padoan. A Monoidal View on Fixpoint Checks, 2023. arXiv:2305.02957.

[BEK+23b] Paolo Baldan, Richard Eggert, Barbara König, Timo Matt, and Tommaso Padoan. A monoidal view on fixpoint checks. In *Proc. of ICGT '23*, pages 3–21. Springer, 2023. LNCS 13961.

[BEKP21a] Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint Theory – Upside Down. In *Proc. of FOSSACS '21*, pages 62–81. Springer, 2021. LNCS/ARCoSS 12650.

[BEKP21b] Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint Theory – Upside Down, 2021. arXiv:2101.08184.

[BEKP22] Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. A Lattice-Theoretical View of Strategy Iteration, 2022. arXiv:2207.09872.

[BEKP23a] Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. A Lattice-Theoretical View of Strategy Iteration. In *Proc. of CSL '23*, volume 252 of *LIPIcs*, pages 7:1–7:19. Schloss Dagstuhl – Leibniz Center for Informatics, 2023.

[BEKP23b] Paolo Baldan, Richard Eggert, Barbara König, and Tommaso Padoan. Fixpoint Theory – Upside Down. *Logical Methods in Computer Science*, 19(2:15), June 2023. Selected Papers of the 24th International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2022).

[BFL+08] Patricia Bouyer, Uli Fahrenberg, Kim G. Larsen, Nicolas Markey, and Jiří Srba. Infinite Runs in Weighted Timed Automata with Energy Constraints. In Franck Cassez and Claude Jard, editors, *Formal Modeling and Analysis of Timed Systems*, pages 33–47. Springer, 2008.

[Bir40] Garrett Birkhoff. *Lattice Theory.* New York, 1940.

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.

[BKP18] Filippo Bonchi, Barbara König, and Daniela Petrişan. Up-To Techniques for Behavioural Metrics via Fibrations. In *Proc. of CONCUR 2018*, volume 118 of *LIPIcs*, pages 17:1–17:17. Schloss Dagstuhl – Leibniz Center for Informatics, 2018.

[BKP20] Paolo Baldan, Barbara König, and Tommaso Padoan. Abstraction, Up-to Techniques and Games for Systems of Fixpoint Equations. In *Proc. of CONCUR '20*, volume 171 of *LIPIcs*, pages 25:1–25:20. Schloss Dagstuhl – Leibniz Center for Informatics, 2020.

[Bos92] Uwe Bosse. An "Ehrenfeucht-Fraïssé Game" for Fixpoint Logic and Stratified Fixpoint Logic. In Egon Börger, Gerhard Jäger, Hans Kleine Büning, Simone Martini, and Michael M. Richter, editors, *Computer Science Logic, 6th Workshop, CSL '92, San Miniato, Italy, September 28 - October 2, 1992, Selected Papers*, volume 702 of *Lecture Notes in Computer Science*, pages 100–114. Springer, 1992.

[CC77] Patrick Cousot and Radhia Cousot. Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi, editors, *Conference Record of the Fourth ACM Symposium on Principles of*

Programming Languages, Los Angeles, California, USA, January 1977, pages 238–252. ACM, 1977.

[CC00] Patrick Cousot and Radhia Cousot. Temporal Abstract Interpretation. In Mark N. Wegman and Thomas W. Reps, editors, *Proc. of POPL '00*, pages 12–25. ACM, 2000.

[CCP+12] Maxime Cordy, Andreas Classen, Gilles Perrouin, Pierre-Yves Schobbens, Patrick Heymans, and Axel Legay. Simulation-based abstractions for software product-line model checking. In *Proc. of ICSE '12 (International Conference on Software Engineering)*, pages 672–682. IEEE, 2012.

[CdAHS03] Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Mariëlle Stoelinga. Resource Interfaces. In Rajeev Alur and Insup Lee, editors, *Prof. of EMSOFT '03 (International Workshop on Embedded Software)*, volume 2855 of *Lecture Notes in Computer Science*, pages 117–133. Springer, 2003.

[CF11] Krishnendu Chatterjee and Nathanaël Fijalkow. A reduction from parity games to simple stochastic games. In Giovanna D'Agostino and Salvatore La Torre, editors, *Proceedings of Second International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2011, Minori, Italy, 15-17th June 2011*, volume 54 of *EPTCS*, pages 74–86, 2011.

[CG99] Andrea Corradini and Fabio Gadducci. An Algebraic Presentation of Term Graphs, via GS-Monoidal Categories. *Appl. Categorical Struct.*, 7(4):299–331, 1999.

[CJK+17] Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *Proc. of STOC 2017*, pages 252–263, 2017.

[Cle90] Rance Cleaveland. On Automatically Explaining Bisimulation Inequivalence. In *Proc. of CAV '90*, pages 364–372. Springer, 1990. LNCS 531.

[Con90] Anne Condon. On Algorithms for Simple Stochastic Games. In *Advances In Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–71, 1990.

[Con92] Anne Condon. The Complexity of Stochastic Games. *Inf. Comput.*, 96(2):203–224, 1992.

[CvBW12] Di Chen, Franck van Breugel, and James Worrell. On the Complexity of Computing Probabilistic Bisimilarity. In *Proc. of FOSSACS 2012*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2012.

[dAFS09] Luca de Alfaro, Marco Faella, and Mariëlle Stoelinga. Linear and Branching System Metrics. *IEEE Trans. Software Eng.*, 35(2):258–273, 2009.

[DGJP04] Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318:323–354, 2004.

[DKZ19] Dani Dorfman, Haim Kaplan, and Uri Zwick. A Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPIcs*, pages 114:1–114:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[DP02] B. A. Davey and H. A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2 edition, 2002.

[EK19] Richard Eggert and Barbara König. Computing Coalgebraic Behavioural Metrics On-The-Fly. In *CALCO Early Ideas '19*, 2019.

[Fea10] John Fearnley. Exponential Lower Bounds for Policy Iteration. In *Proc. of ICALP 2010*, volume 6199 of *Lecture Notes in Computer Science*, pages 551–562. Springer, 2010.

[Fri11] Oliver Friedmann. An Exponential Lower Bound for the Latest Deterministic Strategy Iteration Algorithms. *Log. Methods Comput. Sci.*, 7(3), 2011.

[Fu12] Hongfei Fu. Computing Game Metrics on Markov Decision Processes. In *Proc. of ICALP '12, Part II*, pages 227–238. Springer, 2012. LNCS 7392.

[Gad96] Fabio Gadducci. *On The Algebraic Approach To Concurrent Term Rewriting*. PhD thesis, University of Pisa, 1996.

[GH97] Fabio Gadducci and Reiko Heckel. An inductive view of graph transformation. In *Recent Trends in Algebraic Development Techniques, 12th International Workshop, WADT '97*, pages 223–237. Springer-Verlag, 1997. LNCS 1376.

[GO07] Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, J. F. A. K. van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*, pages 249–329. North-Holland, 2007.

[GRS00] Roberto Giacobazzi, Francesco Ranzato, and Francesca Scozzari. Making abstract interpretations complete. *Journal of the ACM*, 47(2):361–416, 2000.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[GVdV18] Jan Friso Groote, Jao Rivera Verduzco, and Erik P. de Vink. An Efficient Algorithm to Determine Probabilistic Bisimulation. *Algorithms*, 11(9):131, 2018.

[Hal93] Paul R. Halmos. *Finite-Dimensional Vector Spaces.* Undergraduate Texts in Mathematics. Springer, 1993.

[HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency, journal = Journal of the ACM. 32:137–161, 1985.

[How60] Ronald A. Howard. *Dynamic Programming and Markov Processes.* MIT Press, 1960.

[HS21] Daniel Hausmann and Lutz Schröder. Quasipolynomial Computation of Nested Fixpoints. In Jan Friso Groote and Kim Guldstrand Larsen, editors, *Proc. of TACAS 2021*, volume 12651 of *Lecture Notes in Computer Science*, pages 38–56. Springer, 2021.

[JL17] Marcin Jurdzinski and Ranko Lazic. Succinct progress measures for solving parity games. In *Proc. of LICS 2017*, pages 1–9. ACM/IEEE, 2017.

[KH66] Richard M. Karp and Alan J. Hoffman. On nonterminating stochastic games. *Management Science*, 12(5):359–370, 1966.

[KKKW18] Edon Kelmendi, Julia Krämer, Jan Křetínský, and Maximilian Weininger. Value Iteration for Simple Stochastic Games: Stopping Criterion and Learning Algorithm. In *Proc. of CAV '18*, pages 623–642. Springer, 2018. LNCS 10981.

[KMW23] Jan Kretínský, Tobias Meggendorfer, and Maximilian Weininger. Stopping Criteria for Value Iteration on Stochastic Games with Quantitative Objectives. *CoRR*, abs/2304.09930, 2023.

[KRSW20] Jan Kretínský, Emanuel Ramneantu, Alexander Slivinskiy, and Maximilian Weininger. Comparison of Algorithms for Simple Stochastic Games (Full Version). *CoRR*, abs/2008.09465, 2020.

[KT] Bronislaw Knaster and Alfred Tarski. Un théorème sur les fonctions d'ensembles. *Ann. Soc. Polon. Math.*, vol. 6, pages 133-134, 1928.

[KUK+22] Mayuko Kori, Natsuki Urabe, Shin-ya Katsumata, Kohei Suenaga, and Ichiro Hasuo. The Lattice-Theoretic Essence of Property Directed Reachability Analysis. In Sharon Shoham and Yakir Vizel, editors, *Computer Aided Verification - 34th International Conference, CAV 2022, Haifa, Israel, August 7-10, 2022, Proceedings, Part I*, volume 13371 of *Lecture Notes in Computer Science*, pages 235–256. Springer, 2022.

[KV12] B. H. Korte and Jens Vygen. *Combinatorial Optimization: Theory and Algorithms.* Springer-Verlag, New York, NY, 2012.

[Leh18] Karoliina Lehtinen. A modal μ perspective on solving parity games in quasi-polynomial time. In *Proc. of LICS 2018*, pages 639–648. ACM/IEEE, 2018.

[LP06] Yu. M. Lifshits and D. S. Pavlov. Potential theory for mean payoff games. *Zapiski Nauchnykh Seminarov POMI*, 340:61–75, 2006.

[LPSS12] Tadeusz Litak, Dirk Pattinson, Katsuhiko Sano, and Lutz Schröder. Coalgebraic Predicate Logic. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, volume 7392 of *Lecture Notes in Computer Science*, pages 299–311. Springer, 2012.

[Mac71] Saunders Mac Lane. *Categories for the Working Mathematician.* Springer-Verlag, 1971.

[Mém11] Facundo Mémoli. Gromov-Wasserstein Distances and the Metric Approach to Object Matching. *Found. Comput. Math.*, 11(4):417–487, 2011.

[MMKK18] Annabelle McIver, Carroll Morgan, Benjamin Lucien Kaminski, and Joost-Pieter Katoen. A new proof rule for almost-sure termination. *Proc. ACM Program. Lang.*, 2(POPL):33:1–33:28, 2018.

[Mor94] Peter. Morris. *Introduction to Game Theory by Peter Morris.* Universitext. Springer New York, New York, NY, 1st ed. 1994. edition, 1994.

[Mun07] Daniele Mundici. MV-algebras. A short tutorial. Available at `http://www.matematica.uns.edu.ar/IXCongresoMonteiro/Comunicaciones/Mundici_tutorial.pdf`, 2007.

[NNH10] Flemming Nielson, Hanne R. Nielson, and Chris Hankin. *Principles of Program Analysis.* Springer Publishing Company, Incorporated, 2010.

[Pat03] Dirk Pattinson. Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309(1):177–193, 2003.

[PC20] Gabriel Peyré and Marco Cuturi. Computational Optimal Transport, 2020. arXiv:1803.00567.

[Pou07] Damien Pous. Complete Lattices and Up-To Techniques. In *Proc. of APLAS '07*, pages 351–366. Springer, 2007. LNCS 4807.

[PTB85] Robert Paige, Robert Endre Tarjan, and Robert Bonic. A Linear Time Solution to the Single Function Coarsest Partition Problem. *Theor. Comput. Sci.*, 40:67–84, 1985.

[Rab63] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[Rut00] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249:3–80, 2000.

[San11] Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.

[Sch08a] Sven Schewe. An Optimal Strategy Improvement Algorithm for Solving Parity and Payoff Games. In *Proc. of CSL 2008*, volume 5213 of *Lecture Notes in Computer Science*, pages 369–384. Springer, 2008.

[Sch08b] Lutz Schröder. Expressivity of Coalgebraic Modal Logic: The Limits and Beyond. *Theoretical Computer Science*, 390:230–247, 2008.

[SR11] Davide Sangiorgi and J. Rutten. Advanced Topics in Bisimulation and Coinduction. *Journal of Computer and System Sciences - JCSS*, 01 2011.

[Sti97] Colin Stirling. Bisimulation, model checking and other games. Notes for Mathfit instructional meeting on games and computation, Edinburgh, June 1997.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285 – 309, 1955.

[TvB17] Qiyi Tang and Franck van Breugel. Algorithms to Compute Probabilistic Bisimilarity Distances for Labelled Markov Chains. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, volume 85 of *LIPIcs*, pages 27:1–27:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[TVK11] Rahul Tripathi, Elena Valkanova, and V. S. Anil Kumar. On strategy improvement algorithms for simple stochastic games. *J. Discrete Algorithms*, 9(3):263–278, 2011.

[vB17] Franck van Breugel. Probabilistic Bisimilarity Distances. *ACM SIGLOG News*, 4(4):33–51, 2017.

[Vil09] Cédric Villani. *Optimal Transport – Old and New*, volume 338 of *A Series of Comprehensive Studies in Mathematics*. Springer, 2009.

[VJ00] Jens Vöge and Marcin Jurdzinski. A Discrete Strategy Improvement Algorithm for Solving Parity Games. In *Proc. of CAV 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2000.

[WS22] Paul Wild and Lutz Schröder. Characteristic Logics for Behavioural Hemimetrics via Fuzzy Lax Extensions. *Logical Methods in Computer Science*, 18(2), 2022.

[ZP95] Uri Zwick and Mike Paterson. The Complexity of Mean Payoff Games. In Ding-Zhu Du and Ming Li, editors, *Computing and Combinatorics, First Annual International Conference, COCOON '95, Xi'an, China, August 24-26, 1995, Proceedings*, volume 959 of *Lecture Notes in Computer Science*, pages 1–10. Springer, 1995.

[ZP96] Uri Zwick and Mike Paterson. The Complexity of Mean Payoff Games on Graphs. *Theoretical Computer Science*, 158(1&2):343–359, 1996.

Index

Nomenclature

$(\alpha^{a,\delta}, \gamma^{a,\delta})$ Galois connection of the approximation framework (dual sense), page 88

$(\alpha_{a,\delta}, \gamma_{a,\delta})$ Galois connection of the approximation framework (primal sense), page 77

$(L, \sqsubseteq)$ (Partially) ordered set, page 19

(X, d) (Pseudo)metric Space, page 14

$\bar{\mathcal{E}}$ Fixpoint operator of energy games, page 70

$\bigsqcap Y$ Meet of $Y \subseteq L$, page 19

$\bigsqcup Y$ Join of $Y \subseteq L$, page 19

$\bot$ Bottom Element, page 20

χ Characteristic function, page 12

Δ Fixpoint operator for behavioural distances for labeled Markov chains, page 53

δ^a $\min\{a(y) \mid y \in [Y]^a\}$, page 88

δ_a $\min\{\overline{a(y)} \mid y \in [Y]_a\}$, page 77

Γ_E Instance of an energy game, page 68

$\Gamma_F(t_1, t_2)$ Set of coupling of $t_1, t_2 \in FX$, page 40

Γ_M Instance of a discounted mean-payoff game, page 62

Γ_S Instance of a simple stochastic game, page 64

ι_a^f Sufficiently small constant, such that $f_a^{\#} = f_{a,\iota_a^f}^{\#}$, page 81

ι_f^a Sufficiently small constant, such that $f_{\#}^a = f_{\#}^{a,\iota_f^a}$, page 89

LMC Instance of a labeled Markov chain, page 51

$\mathbb{A}(A, B)$ Collection of arrows between $A, B \in ob(\mathbb{A})$, page 31

$\mathbb{A} \times \mathbb{B}$ Product category of categories $\mathbb{A}$ and $\mathbb{B}$, page 32

Set Category of sets, page 31

$\mathcal{B}$ Fixpoint operator for bisimilarity for transition systems, page 50

$\mathcal{D}(X)$ Set of probability distributions over X, page 13

$\mathcal{D}_f(X)$ Set of probability distributions with finite support over X, page 13

$\mathcal{D}_{\mathbb{M}}(Y)$ Set of distributions $p\colon Y \to \mathbb{M}$, page 28

$\mathcal{E}$ Fixpoint operator of energy games with finite values, page 71

$\mathcal{H}$ Hausdorff distance, page 43

$\mathcal{J}$ Fixpoint operator for behavioural distances for metric transition systems, page 55

$\mathcal{K}$ Kantorovich distance, page 45

$\mathcal{L}$ Fixpoint operator for discounted mean-payoff games, page 63

$\mathcal{M}$ Fixpoint operator for behavioural distances for probabilistic automata, page 57

$\mathcal{P}(X)$ Powerset of X, page 13

$\mathcal{T}$ Fixpoint operator for termination probability of Markov chains, page 48

$\mathcal{V}$ Fixpoint operator for simple stochastic games, page 66

add_w Basic function: addition, page 91

ev_F Evaluation map $ev_F\colon F\mathbb{M} \to \mathbb{M}$, page 39

$Fix(f)$ Set of fixpoints of f, page 22

id_A Identity of $A \in ob(\mathbb{A})$, page 31

$ob(\mathbb{A})$ Collection of objects of category $\mathbb{A}$, page 31

$Post(f)$ Set of post-fixpoints of f, page 22

$Pre(f)$ Set of pre-fixpoints of f, page 22

sub_w Basic function: substraction, page 91

sub'_w Basic function: substraction (whole numbers), page 113

$supp(f)$ Support of the function f, page 13

av_D Basic function: average, page 91

$imp^s_{\max}(C)$ Set of stable max-improvements of C, page 153

$imp_{\max}(C)$ Set of max-improvements of C, page 153

MC Instance of a Markov chain, page 47

$imp^{s}_{\min}(C)$ Set of stable min-improvements of C, page 149

$imp_{\min}(C)$ Set of min-improvements of C, page 149

$\mathbb{M}$ MV-algebra, page 26

MTS Instance of a metric transition system, page 54

PA Instance of a probabilistic automaton, page 56

Π Set of all positional strategies in a two-player game for Min, page 60

$\mathcal{P}_f(X)$ Finite powerset of X, page 13

Σ Set of all positional strategies in a two-player game for Max, page 60

σ A positional strategy for Max in a two-player-game, page 60

σ^* Optimal strategy for Max, page 61

τ A positional strategy for Min in a two-player-game, page 60

τ^* Optimal strategy for Min, page 61

$\tilde{F}$ Predicate lifting of functor $F\colon \mathbf{Set} \to \mathbf{Set}$, page 38

$\top$ Top Element, page 20

TS Instance of a transition system, page 49

$[Y]_a$ Support of $\overline{a}$, page 77

$[Y]^a$ Support of a, page 88

c_k Basic function: constant, page 91

d^F Lifting of a pseudometric d, page 39

$d^{\downarrow F}$ Wasserstein distance, page 41

f^a_* Approximation of f corresponding to any a in the dual sense, page 89

$f^a_\#$ Approximation of f corresponding to fixpoint a in the dual sense, page 89

f^*_a Approximation of f corresponding to any a in the primal sense, page 87

$f^\#_a$ Approximation of f corresponding to fixpoint a in the primal sense, page 79

$FA/F(A)$ Functor F applied to object A, page 32

$G = (V, E)$ Graph G, set of vertices V, set of edges E, page 59

$G_\sigma/G_\tau/G_{\sigma_\tau}$ Strategy-induced graphs, page 60

DOI: 10.17185/duepublico/81347
URN: urn:nbn:de:hbz:465-20240111-150001-9