Lectures on the modal μ-calculus

Yde Venema*

Abstract

These notes give an introduction to the theory of the modal μ-calculus.

*Institute for Logic, Language and Computation, University of Amsterdam, Science Park 107, NL–1098XG Amsterdam. E-mail: `y.venema@uva.nl`.

Contents

Introduction

The study of the modal μ-calculus can be motivated from various (not necessarily disjoint!) directions.

Process Theory In this area of theoretical computer science, one studies formalisms for describing and reasoning about labelled transition systems — these being mathematical structures that model processes. Such formalisms then have important applications in the specification and verification of software. For such purposes, the modal μ-calculus strikes a very good balance between computational efficiency and expressiveness. On the one hand, the presence of fixpoint operators make it possible to express most, if not all, of the properties that are of interest in the study of (ongoing) behavior. But on the other hand, the formalism is still simple enough to allow an (almost) polynomial model checking complexity and an exponential time satisfiability problem.

Modal Logic From the perspective of modal logic, the modal μ-calculus is a well-behaved extension of the basic formalism, with a great number of attractive logical properties. For instance, it is the bisimulation invariant fragment of second order logic, it enjoys uniform interpolation, and the set of its validities admits a transparent, finitary axiomatization, and has the finite model property. In short, the modal μ-calculus shares (or naturally generalizes) all the nice properties of ordinary modal logic.

Mathematics and Theoretical Computer Science More generally, the modal μ-calculus has a very interesting theory, with lots of connections with neighboring areas in mathematics and theoretical computer science. We mention automata theory (more specifically, the theory of finite automata operating on infinite objects), game theory, universal algebra and lattice theory, and the theory of universal coalgebra.

Open Problems Finally, there are still a number of interesting *open problems* concerning the modal μ-calculus. For instance, it is unknown whether the characterization of the modal μ-calculus as the bisimulation invariant fragment of monadic second order logic still holds if we restrict attention to finite structures, and in fact there are many open problems related to the expressiveness of the formalism. Also, the exact complexity of the model checking problem is not known. And to mention a third example: the completeness theory of modal fixpoint logics is still a largely undeveloped field.

Summarizing, the modal μ-calculus is a formalism with important applications in the field of process theory, with interesting metalogical properties, various nontrivial links with other areas in mathematics and theoretical computer science, and a number of intriguing open problems. Reason enough to study it in more detail.

1 Basic Modal Logic

As mentioned in the preface, we assume familiarity with the basic definitions concerning the syntax and semantics of modal logic. The purpose of this first chapter is to briefly recall notation and terminology. We focus on some aspects of modal logic that feature prominently in its extensions with fixpoint operators.

Convention 1.1 Throughout this text we let Prop be a countably infinite set of *propositional variables*, whose elements are usually denoted as $p, q, r, x, y, z, \ldots$, and we let D be a finite set of (atomic) *actions*, whose elements are usually denoted as $d, e, c, \ldots$. We will usually focus on a finite subset P of Prop, consisting of those propositional variables that occur freely in a particular formula. In practice we will often suppress explicit reference to Prop, P and D.

1.1 Basics

Structures

▶ `Introduce LTSs as process graphs`

Definition 1.2 A *(labelled) transition system*, *LTS*, or *(Kripke) model* of *type* (P, D) is a triple $\mathbb{S} = \langle S, V, R \rangle$ such that S is a set of objects called *states* or *points*, $V : \mathsf{P} \to \wp(S)$ is a *valuation*, and $R = \{R_d \subseteq S \times S \mid d \in \mathsf{D}\}$ is a family of binary *accessibility relations*. In case D is a singleton, we will simply write R for the unique accessibility relation in a model.

Elements of the set $R_d[s] := \{t \in S \mid (s, t) \in R_d\}$ are called *d-successors* of s. A transition system is called *image-finite* or *finitely branching* if $R_d[s]$ is finite, for every $d \in \mathsf{D}$ and $s \in S$.

A *pointed* transition system or Kripke model is a pair $(\mathbb{S}, s)$ consisting of a transition system $\mathbb{S}$ and a designated state s in $\mathbb{S}$. ◁

Remark 1.3 It will occasionally be convenient to work with an alternative, *coalgebraic* presentation of transition systems. Intuitively, it should be clear that instead of having a valuation $V : \mathsf{P} \to \wp(S)$, telling us at which states each proposition letter is true, we could just as well have a *marking* $\sigma_V : S \to \wp(\mathsf{P})$ informing us which proposition letters are true at each state. Also, a binary relation R on a set S can be represented as a map $R[\cdot] : S \to \wp(S)$ mapping a state s to the collection $R[s]$ of its successors. In this line, a family $R = \{R_d \subseteq S \times S \mid d \in \mathsf{D}\}$ of accessibility relations can be seen as a map $\sigma_R : S \to \wp(S)^{\mathsf{D}}$, where $\wp(S)^{\mathsf{D}}$ denotes the set of maps from D to $\wp(S)$.

Combining these two maps into one single function, we see that a transition system $\mathbb{S} = \langle S, V, R \rangle$ of type (P, D) can be seen as a pair $\langle S, \sigma \rangle$, where $\sigma : S \to \wp(\mathsf{P}) \times \wp(S)^{\mathsf{D}}$ is the map given by $\sigma(s) := (\sigma_V(s), \sigma_R(s))$. ◁

For future reference we define the notion of a *Kripke functor*.

Definition 1.4 Fix a set P of proposition letters and a set D of atomic actions. Given a set S, let $\mathsf{K}_{\mathsf{D},\mathsf{P}} S$ denote the set

$$\mathsf{K}_{\mathsf{D},\mathsf{P}} S := \wp(\mathsf{P}) \times \wp(S)^{\mathsf{D}}.$$

This operation will be called the *Kripke functor* associated with D and P.

A typical element of $\mathsf{K}_{\mathsf{D},\mathsf{P}}S$ will be denoted as (π, X), with $\pi \subseteq \mathsf{P}$ and $X = \{X_d \mid d \in \mathsf{D}\}$ with $X_d \subseteq S$ for each $d \in \mathsf{D}$.

When we take this perspective we will sometimes refer to Kripke models as $\mathsf{K}_{\mathsf{D},\mathsf{P}}S$-*coalgebras* or *Kripke coalgebras*. ◁

Given this definition we may summarize Remark 1.3 by saying that any transition system can be presented as a pair $\mathbb{S} = \langle S, \sigma : S \to \mathsf{K}S\rangle$ where K is the Kripke functor associated with $\mathbb{S}$. In practice, we will usually write K rather than $\mathsf{K}_{\mathsf{D},\mathsf{P}}$.

Syntax

Working with fixpoint operators, we may benefit from a set-up in which the use of the negation symbol may only be applied to atomic formulas. The price that one has to pay for this is an enlarged arsenal of primitive symbols. In the context of modal logic we then arrive at the following definition.

Definition 1.5 The language $\mathtt{ML}_{\mathsf{D}}$ of *polymodal logic* in D is defined as follows:

$$\varphi ::= p \mid \overline{p} \mid \bot \mid \top \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond_d \varphi \mid \Box_d \varphi$$

where $p \in \mathsf{Prop}$, and $d \in \mathsf{D}$. Elements of $\mathtt{ML}_{\mathsf{D}}$ are called *(poly-)modal formulas*, or briefly, *formulas*. In case the set D is a singleton, we speak of the language $\mathtt{ML}$ of *basic modal logic* or *monomodal* logic; in this case we will denote the modal operators by $\Diamond$ and $\Box$, respectively.

Given a finite set P of propositional variables, we let $\mathtt{ML}_{\mathsf{D}}(\mathsf{P})$ denote the set of formulas in which only variables from P occur. ◁

Often the sets P and D are implicitly understood, and suppressed in the notation. Generally it will suffice to treat examples, proofs, etc., from monomodal logic.

We will need some definitions and notations concerning atomic formulas.

Definition 1.6 Let P be a set of propositional variables. We define the sets $\mathtt{Lit}(\mathsf{P})$ and $\mathtt{At}(\mathsf{P})$ of, respectively, *literals* and *atomic formulas over* P as follows:

$$\begin{array}{lcl} \mathtt{Lit}(\mathsf{P}) & := & \{p, \overline{p} \mid p \in \mathsf{P}\} \\ \mathtt{At}(\mathsf{P}) & := & \{\bot, \top\} \cup \mathtt{Lit}(\mathsf{P}) \end{array}$$

We will generally use the symbol ℓ to denote an arbitrary literal. ◁

Remark 1.7 The *negation* $\sim\!\varphi$ of a formula φ can inductively be defined as follows:

$$\begin{array}{lcl@{\qquad\qquad}lcl} \sim\bot & := & \top & \sim\top & := & \bot \\ \sim p & := & \overline{p} & \sim\overline{p} & := & p \\ \sim(\varphi \vee \psi) & := & \sim\varphi \wedge \sim\psi & \sim(\varphi \wedge \psi) & := & \sim\varphi \vee \sim\psi \\ \sim\Box_d\varphi & := & \Diamond_d \sim\varphi & \sim\Diamond_d\varphi & := & \Box_d \sim\varphi \end{array}$$

On the basis of this, we can also define the other standard abbreviated connectives, such as $\to$ and $\leftrightarrow$. ◁

We assume that the reader is familiar with standard syntactic notions such as those of a *subformula* or the *construction tree* of a formula, and with standard syntactic operations such as *substitution*. Concerning the latter, we let $\varphi[\psi/p]$ denote the formula that we obtain by substituting all occurrences of p in φ by ψ.

Definition 1.8 We define the collection $Sf(\xi)$ of subformulas of a modal formula ξ by the following induction on the complexity of ξ:

$$\begin{array}{lcll}
Sf(\bot) & := & \{\bot\} & \\
Sf(\top) & := & \{\top\} & \\
Sf(p) & := & \{p\} & \\
Sf(\overline{p}) & := & \{\overline{p}\} & \\
Sf(\varphi \star \psi) & := & \{\varphi \star \psi\} \cup Sf(\varphi) \cup Sf(\psi) & \text{where } \star \in \{\vee, \wedge\} \\
Sf(\heartsuit\varphi) & := & \{\heartsuit\varphi\} \cup Sf(\varphi) & \text{where } \heartsuit \in \{\Diamond_d, \Box_d \mid d \in \mathsf{D}\}
\end{array}$$

We write $\varphi \trianglelefteq \psi$ to denote that φ is a subformula of ψ. The *size* of a formula ξ is defined as the number of its subformulas, $|\xi| := |Sf(\xi)|$. ◁

Semantics

The *relational* semantics of modal logic is well known. The basic idea is that the modal operators $\Diamond_d$ and $\Box_d$ are both interpreted using the *accessibility* relation R_d.

The notion of truth (or satisfaction) is defined as follows.

Definition 1.9 Let $\mathbb{S} = \langle S, \sigma \rangle$ be a transition system of type (P, D). Then the *satisfaction relation* $\Vdash$ between states of $\mathbb{S}$ and formulas of $\mathtt{ML}_{\mathsf{D}}(\mathsf{P})$ is defined by the following formula induction.

$$\begin{array}{lll}
\mathbb{S}, s \Vdash p & \text{if} & s \in V(p), \\
\mathbb{S}, s \Vdash \overline{p} & \text{if} & s \notin V(p), \\
\mathbb{S}, s \Vdash \bot & & \text{never}, \\
\mathbb{S}, s \Vdash \top & & \text{always}, \\
\mathbb{S}, s \Vdash \varphi \vee \psi & \text{if} & \mathbb{S}, s \Vdash \varphi \text{ or } \mathbb{S}, s \Vdash \psi, \\
\mathbb{S}, s \Vdash \varphi \wedge \psi & \text{if} & \mathbb{S}, s \Vdash \varphi \text{ and } \mathbb{S}, s \Vdash \psi, \\
\mathbb{S}, s \Vdash \Diamond_d\varphi & \text{if} & \mathbb{S}, t \Vdash \varphi \text{ for some } t \in R_d[s], \\
\mathbb{S}, s \Vdash \Box_d\varphi & \text{if} & \mathbb{S}, t \Vdash \varphi \text{ for all } t \in R_d[s].
\end{array}$$

We say that φ *is true* or *holds* at s if $\mathbb{S}, s \Vdash \varphi$, and we let the set

$$[\![\varphi]\!]^{\mathbb{S}} := \{s \in S \mid \mathbb{S}, s \Vdash \varphi\}.$$

denote the *meaning* or *extension* of φ in $\mathbb{S}$. ◁

Alternatively (but equivalently), one may define the semantics of modal formulas directly in terms of this meaning function $[\![\varphi]\!]^{\mathbb{S}}$. This approach has some advantages in the context of fixpoint operators, since it brings out the role of the powerset algebra $\wp(S)$ more clearly.

Remark 1.10 Fix an LTS $\mathbb{S}$, then define $[\![\varphi]\!]^{\mathbb{S}}$ by induction on the complexity of φ:

$$\begin{array}{lcllcl}
[\![p]\!]^{\mathbb{S}} & = & V(p) & [\![\overline{p}]\!]^{\mathbb{S}} & = & S \setminus V(p) \\
[\![\bot]\!]^{\mathbb{S}} & = & \varnothing & [\![\top]\!]^{\mathbb{S}} & = & S \\
[\![\varphi \vee \psi]\!]^{\mathbb{S}} & = & [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} & [\![\varphi \wedge \psi]\!]^{\mathbb{S}} & = & [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}} \\
[\![\Diamond_d \varphi]\!]^{\mathbb{S}} & = & \langle R_d \rangle [\![\varphi]\!]^{\mathbb{S}} & [\![\Box_d \varphi]\!]^{\mathbb{S}} & = & [R_d] [\![\varphi]\!]^{\mathbb{S}}
\end{array}$$

Here the operations $\langle R_d \rangle$ and $[R_d]$ on $\wp(S)$ are defined by putting

$$\begin{array}{lcl}
\langle R_d \rangle (X) & := & \{ s \in S \mid R_d[s] \cap X \neq \varnothing \} \\
[R_d](X) & := & \{ s \in S \mid R_d[s] \subseteq X \}.
\end{array}$$

The satisfaction relation $\Vdash$ may be recovered from this by putting $\mathbb{S}, s \Vdash \varphi$ iff $s \in [\![\varphi]\!]^{\mathbb{S}}$. $\lhd$

Definition 1.11 Let s and s' be two states in the transition systems $\mathbb{S}$ and $\mathbb{S}'$ of type (P, D), respectively. Then we say that s and s' are *modally equivalent*, notation: $\mathbb{S}, s \equiv_{(\mathsf{P},\mathsf{D})} \mathbb{S}', s'$, if s and s' satisfy the same modal formulas, that is, $\mathbb{S}, s \Vdash \varphi$ iff $\mathbb{S}', s' \Vdash \varphi$, for all modal formulas $\varphi \in \mathtt{ML}_{\mathsf{D}}(\mathsf{P})$. $\lhd$

Flows, trees and streams

In some parts of these notes *deterministic* transition systems feature prominently.

Definition 1.12 A transition system $\mathbb{S} = \langle S, V, R \rangle$ is called *deterministic* if each $R_d[s]$ is a singleton. $\lhd$

Note that our definition of determinism does not allow $R_d = \varnothing$ for any point s. We first consider the monomodal case.

Definition 1.13 Let P be a set of proposition letters. A deterministic monomodal Kripke model for this language is called a *flow model for* P, or a $\wp(\mathsf{P})$*-flow*. In case such a structure is of the form $\langle \omega, V, Succ \rangle$, where $Succ$ is the standard successor relation on the set ω of natural numbers, we call the structure a *stream model for* P, or a $\wp(\mathsf{P})$*-stream*. $\lhd$

In case the set D of actions is finite, we may just as well identify it with the set $k = \{0, \ldots, k-1\}$, where k is the size of D. We usually restrict to the binary case, that is, $k = 2$. Our main interest will be in Kripke models that are based on the *binary tree*, i.e., a tree in which every node has exactly two successors, a left and a right one.

Definition 1.14 With $2 = \{0, 1\}$, we let 2^* denote the set of finite strings of 0s and 1s. We let ε denote the empty string, while the left- and right successor of a node s are denoted by $s \cdot 0$ and $s \cdot 1$, respectively. Written as a relation, we put

$$Succ_i = \{ (s, s \cdot i) \mid s \in 2^* \}.$$

A *binary tree over* P, or a *binary* $\wp(\mathsf{P})$*-tree* is a Kripke model of the form $\langle 2^*, V, Succ_0, Succ_1 \rangle$. $\lhd$

Remark 1.15 In the general case, the *k-ary tree* is the structure $(k^*, \mathit{Succ}_0, \ldots, \mathit{Succ}_{k-1})$, where k^* is the set of finite sequences of natural numbers smaller than k, and Succ_i is the *i-th successor relation* given by

$$\mathit{Succ}_i = \{(s, s \cdot i) \mid s \in k^*\}.$$

A k-flow model is a Kripke model $\mathbb{S} = \langle S, V, R\rangle$ with k many deterministic accessibility relations, and a *k-ary tree model* is a k-flow model which is based on the k-ary tree. ◁

In deterministic transition systems, the distinction between boxes and diamonds evaporates. It is then convenient to use a single symbol $\bigcirc_i$ to denote either the box or the diamond.

Definition 1.16 The set $\mathrm{MFL}_k(\mathsf{P})$ of formulas of *k-ary Modal Flow Logic* in P is given as follows:

$$\varphi \ ::= \ p \mid \overline{p} \mid \bot \mid \top \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \bigcirc_i \varphi$$

where $p \in \mathsf{P}$, and $i < k$. In case $k = 1$ we will also speak of *modal stream logic*, notation: $\mathrm{MSL}(\mathsf{P})$. ◁

1.2 Game semantics

We will now describe the semantics defined above in game-theoretic terms. That is, we will define the *evaluation game* $\mathcal{E}(\xi, \mathbb{S})$ associated with a (fixed) formula ξ and a (fixed) LTS $\mathbb{S}$. This game is an example of a *board game*. In a nutshell, board games are games in which the players move a token along the edge relation of some graph, so that a *match* of *play* of the game corresponds to a (finite or infinite) path through the graph. Furthermore, the winning conditions of a match are determined by the nature of this path. We will meet many examples of board games in these notes, and in Chapter 5 we will study them in more detail.

The evaluation game $\mathcal{E}(\xi, \mathbb{S})$ is played by two *players*: Éloise ($\exists$ or 0) and Abélard ($\forall$ or 1). Given a player σ, we always denote the *opponent* of σ by $\overline{\sigma}$. As mentioned, a *match* of the game consists of the two players moving a *token* from one position to another. *Positions* are of the form (φ, s), with φ a *subformula* of ξ, and s a state of $\mathbb{S}$.

It is useful to assign *goals* to both players: in an arbitrary position (φ, s), think of $\exists$ trying to show that φ is *true* at s in $\mathbb{S}$, and of $\forall$ of trying to convince her that φ is *false* at s.

Depending on the type of the position (more precisely, on the formula part of the position), one of the two players may move the token to a next position. For instance, in a position of the form $(\Diamond_d \varphi, s)$, it is $\exists$'s turn to move, and she must choose an arbitrary d-successor t of s, thus making (φ, t) the next position. Intuitively, the idea is that in order to show that $\Diamond\varphi$ is true at s, $\exists$ has to come up with a successor of s where φ holds. Formally, we say that the set of *(admissible) next positions* that $\exists$ may choose from is given as the set $\{(\varphi, t) \mid t \in R_d[s]\}$. In the case there is no successor of s to choose, she immediately *loses* the game. This is a convenient way to formulate the rules for winning and losing this game: if a position (φ, s) has no admissible next positions, the player whose turn it is to play at (φ, s) *gets stuck* and immediately loses the game.

This convention gives us a nice handle on positions of the form (p, s) where p is a proposition letter: we always assign to such a position an *empty* set of admissible moves, but we

Position	Player	Admissible moves
$(\varphi_1 \vee \varphi_2, s)$	$\exists$	$\{(\varphi_1, s), (\varphi_2, s)\}$
$(\varphi_1 \wedge \varphi_2, s)$	$\forall$	$\{(\varphi_1, s), (\varphi_2, s)\}$
$(\Diamond_d \varphi, s)$	$\exists$	$\{(\varphi, t) \mid t \in R_d[s]\}$
$(\Box_d \varphi, s)$	$\forall$	$\{(\varphi, t) \mid t \in R_d[s]\}$
$(\bot, s)$	$\exists$	$\varnothing$
$(\top, s)$	$\forall$	$\varnothing$
$(p, s), s \in V(p)$	$\forall$	$\varnothing$
$(p, s), s \notin V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s), s \notin V(p)$	$\forall$	$\varnothing$
$(\overline{p}, s), s \in V(p)$	$\exists$	$\varnothing$

Table 1: Evaluation game for modal logic

make $\exists$ responsible for (p, s) in case p is *false* at s, and $\forall$ in case p is *true* at s. In this way, $\exists$ immediately wins if p is true at s, and $\forall$ if it is otherwise. The rules for the negative literals ($\overline{p}$) and the constants, $\bot$ and $\top$, follow a similar pattern.

The full set of rules of the game is given in Table 1. Observe that all matches of this game are finite, since at each move of the game the active formula is reduced in size. (From the general perspective of board games, this means that we need not worry about winning conditions for matches of infinite length.) We may now summarize the game as follows.

Definition 1.17 Given a modal formula ξ and a transition system $\mathbb{S}$, the *evaluation game* $\mathcal{E}(\xi, \mathbb{S})$ is defined as the board game given by Table 1, with the set $Sf(\xi) \times S$ providing the positions of the game; that is, a position is a pair consisting of a subformula of ξ and a point in $\mathbb{S}$. The instantiation of this game with starting point (ξ, s) is denoted as $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$. $\lhd$

An *instance* of an evaluation game is a pair consisting of an evaluation game and a *starting position* of the game. Such an instance will also be called an *initialized game*, or sometimes, if no confusion is likely, simply a game.

A *strategy* for a player σ in an initialized game is a method that σ uses to select his moves during the play. Such a strategy is *winning for* σ if every match of the game (starting at the given position) is won by σ, provided σ plays according to this strategy. A position (φ, s) is *winning* for σ if σ has a winning strategy for the game initialized in that position. (Note that this definition applies to *all* positions, not only to the ones owned by σ.) The set of winning positions in $\mathcal{E}(\xi, \mathbb{S})$ for σ is denoted as $\mathrm{Win}_\sigma(\mathcal{E}(\xi, \mathbb{S}))$.

The main result concerning these games is that they provide an alternative, but equivalent, semantics for modal logic.

▶ `Example to be added`

Theorem 1.18 (Adequacy) *Let ξ be a modal formula, and let $\mathbb{S}$ be an LTS. Then for any state s in $\mathbb{S}$ it holds that*

$$(\xi, s) \in \mathrm{Win}_\exists(\mathcal{E}(\xi, \mathbb{S})) \iff \mathbb{S}, s \Vdash \xi.$$

The proof of this Theorem is left to the reader.

1.3 Bisimulations and bisimilarity

One of the most fundamental notions in the model theory of modal logic is that of a bisimulation between two transition systems.

▶ `discuss bisimilarity as a notion of behavioral equivalence`

Definition 1.19 Let $\mathbb{S}$ and $\mathbb{S}'$ be two transition systems of the same type (P, D). Then a relation $Z \subseteq S \times S'$ is a *bisimulation* of type (P, D) if the following hold, for every pair $(s, s') \in Z$.
(prop) $s \in V(p)$ iff $s' \in V'(p)$, for all $p \in \mathsf{P}$;
(forth) for all actions d, and for all $t \in R_d[s]$ there is a $t' \in R'_d[s']$ with $(t, t') \in Z$;
(back) for all actions d, and for all $t' \in R'_d[s']$ there is a $t \in R_d[s]$ with $(t, t') \in Z$.

Two states s and s' are called *bisimilar*, notation: $\mathbb{S}, s \leftrightarroweq_{\mathsf{P},\mathsf{D}} \mathbb{S}', s'$ if there is some bisimulation Z of type (P, D) with $(s, s') \in Z$. If no confusion is likely to arise, we generally drop the subscripts, writing '$\leftrightarroweq$' rather than '$\leftrightarroweq_{\mathsf{P},\mathsf{D}}$'. ◁

Bisimilarity and modal equivalence

In order to understand the importance of this notion for modal logic, the starting point should be the observation that the truth of modal formulas is *invariant* under bisimilarity. Recall that $\equiv$ denotes the relation of modal equivalence.

Theorem 1.20 (Bisimulation Invariance) *Let $\mathbb{S}$ and $\mathbb{S}'$ be two transition systems of the same type. Then*

$$\mathbb{S}, s \leftrightarroweq \mathbb{S}', s' \;\Rightarrow\; \mathbb{S}, s \equiv \mathbb{S}', s'$$

for every pair of states s in $\mathbb{S}$ and s' in $\mathbb{S}'$.

Proof. By a straightforward induction on the complexity of modal formulas one proves that bisimilar states satisfy the same formulas. QED

But there is much more to say about the relation between modal logic and bisimilarity than Theorem 1.20. In particular, for some classes of models, one may prove a converse statement, which amounts to saying that the notions of bisimilarity and modal equivalence coincide. Such classes are said to have the *Hennessy-Milner* property. As an example we mention the class of finitely branching transition systems.

Theorem 1.21 (Hennessy-Milner Property) *Let $\mathbb{S}$ and $\mathbb{S}'$ be two finitely branching transition systems of the same type. Then*

$$\mathbb{S}, s \leftrightarroweq \mathbb{S}', s' \iff \mathbb{S}, s \equiv \mathbb{S}', s'$$

for every pair of states s in $\mathbb{S}$ and s' in $\mathbb{S}'$.

Proof. The direction from left to right follows from Theorem 1.20. In order to prove the opposite direction, one may show that the relation $\equiv$ of modal equivalence itself is a bisimulation. Details are left to the reader. QED

This theorem can be read as indication of the expressiveness of modal logic: any difference in behaviour between two states in finitely branching transition systems can in fact be witnessed by a concrete modal formula. As another witness to this expressivity, in section 1.5 we will see that modal logic is sufficiently rich to express all bisimulation-invariant first-order properties. Obviously, this result also adds considerable strength to the link between modal logic and bisimilarity.

As a corollary of the bisimulation invariance theorem, modal logic has the *tree model property*, that is, every satisfiable modal formula is satisfiable on a structure that has the shape of a tree.

Definition 1.22 A transition system $\mathbb{S}$ of type (P, D) is called *tree-like* if the structure $\langle S, \bigcup_{d \in \mathsf{D}} R_d \rangle$ is a tree. $\lhd$

The key step in the proof of the tree model property of modal logic is the observation that every transition system can be 'unravelled' into a bisimilar tree-like model. The basic idea of such an unravelling is the new states encode (part of) the *history* of the old states. Technically, the new states are the *paths* through the old system.

Definition 1.23 Let $\mathbb{S} = \langle S, V, R \rangle$ be a transition system of type (P, D). A *(finite) path* through $\mathbb{S}$ is a nonempty sequence of the form $(s_0, d_1, s_1, d_2, \ldots, s_n)$ such that $R_{d_i} s_{i-1} s_i$ for all i with $0 < i \leq n$. The set of paths through $\mathbb{S}$ is denoted as $\mathit{Paths}(\mathbb{S})$; we use the notation $\mathit{Paths}_s(\mathbb{S})$ for the set of paths starting at s.

The *unravelling* of $\mathbb{S}$ *around* a state s is the transition system $\vec{\mathbb{S}}_s$ which is coalgebraically defined as the structure $\langle \mathit{Paths}_s(\mathbb{S}), \vec{\sigma} \rangle$, where the coalgebra map $\vec{\sigma} = (\vec{\sigma}_V, (\vec{\sigma}_d \mid d \in \mathsf{D}))$ is given by putting

$$\begin{array}{lll} \vec{\sigma}_V(s_0, d_1, s_1, d_2, \ldots, s_n) & := & \sigma_V(s_n), \\ \vec{\sigma}_d(s_0, d_1, s_1, d_2, \ldots, s_n) & := & \{(s_0, d_1, s_1, \ldots, s_n, d, t) \in \mathit{Paths}_s(\mathbb{S}) \mid R_d s_n t\}. \end{array}$$

Finally, the unravelling of a pointed transition system $(\mathbb{S}, s)$ is the pointed structure $(\vec{\mathbb{S}}_s, s)$, where (with some abuse of notation) we let s denote the path of length zero that starts and finishes at s. $\lhd$

Clearly, unravellings are tree-like structures, and any pointed transition system is bisimilar to its unravelling. But then the following theorem is immediate by Theorem 1.20.

Theorem 1.24 (Tree Model Property) *Let φ be a satisfiable modal formula. Then φ is satisfiable at the root of a tree-like model.*

Bisimilarity game

We may also give a game-theoretic characterization of the notion of bisimilarity. We first give an informal description of the game that we will employ. A match of the *bisimilarity game* between two Kripke models $\mathbb{S}$ and $\mathbb{S}'$ is played by two players, $\exists$ and $\forall$. As in the evaluation game, these players move a token around from one *position* of the game to the next one. In the game there are two kinds of positions: pairs of the form $(s, s') \in S \times S'$ are called *basic positions* and belong to $\exists$. The other positions are of the form $Z \subseteq S \times S'$ and belong to $\forall$.

The idea of the game is that at a position (s, s'), $\exists$ claims that s and s' are bisimilar, and to substantiate this claim she proposes a *local bisimulation* $Z \subseteq S \times S'$ (see below) for s and s'. This relation Z can be seen as providing a set of *witnesses* for $\exists$'s claim that s and s' are bisimilar. Implicitly, $\exists$'s claim at a position $Z \subseteq S \times S'$ is that *all* pairs in Z are bisimilar, so $\forall$ can pick an arbitrary pair $(t, t') \in Z$ and challenge $\exists$ to show that these t and t' are bisimilar.

Definition 1.25 Let $\mathbb{S}$ and $\mathbb{S}'$ be two transition systems of the same type (P, D). Then a relation $Z \subseteq S \times S'$ is a *local bisimulation* for two points $s \in S$ and $s' \in S'$, if it satisfies the properties (prop), (back) and (forth) of Definition 1.19 for this specific s and s':
(prop) $s \in V(p)$ iff $s' \in V'(p)$, for all $p \in \mathsf{P}$;
(forth) for all actions d, and for all $t \in R_d[s]$ there is a $t' \in R'_d[s']$ with $(t, t') \in Z$;
(back) for all actions d, and for all $t' \in R'_d[s']$ there is a $t \in R_d[s]$ with $(t, t') \in Z$. $\lhd$

Note that a local bisimulation for s and s' need only relate successors of s to successors of s'. In particular, the pair (s, s') itself will generally not belong to such a relation. It is easy to see that a relation Z between two Kripke models is a bisimulation iff Z is a local bisimulation for every pair $(s, s') \in Z$.

If a player gets stuck in a match of the bisimilarity game, then the opponent wins the match. For instance, if s and s' disagree about some proposition letter, then there is *no* local bisimulation for s and s', and so the corresponding position (s, s') is an immediate loss for $\exists$. Or, if neither s nor s' has successors, and agree on the truth of all proposition letters, then $\exists$ could choose the *empty* relation as a local bisimulation, so that $\forall$ would lose the match at his next move.

A new option arises if neither player gets stuck: this game may also have matches that last *forever*. Nevertheless, we can still declare a winner for such matches, and the agreement is that $\exists$ is the winner of any infinite match. Formally, we put the following.

Definition 1.26 The *bisimilarity game* $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ between two Kripke models $\mathbb{S}$ and $\mathbb{S}'$ is the board game given by Table 2, with the winning condition that finite matches are lost by the player who got stuck, while all infinite matches are won by $\exists$.

A position (s, s') is *winning* for σ if σ has a winning strategy for the game initialized in that position. The set of these positions is denoted as $\mathrm{Win}_\sigma(\mathcal{B}(\mathbb{S}, \mathbb{S}'))$. $\lhd$

Also observe that a bisimulation is a relation which is a local bisimulation for each of its members. The following theorem states that the collection of basic winning positions for $\exists$ forms the *largest bisimulation* between $\mathbb{S}$ and $\mathbb{S}'$.

Position	Player	Admissible moves
$(s, s') \in S \times S'$	$\exists$	$\{Z \in \wp(S \times S') \mid Z$ is a local bisimulation for s and $s'\}$
$Z \in \wp(S \times S')$	$\forall$	$Z = \{(t, t') \mid (t, t') \in Z\}$

Table 2: Bisimilarity game for Kripke models

Theorem 1.27 *Let* $(\mathbb{S}, s)$ *and* $(\mathbb{S}', s')$ *be two pointed Kripke models. Then* $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$ *iff* $(s, s') \in \mathrm{Win}_\exists(\mathcal{B}(\mathbb{S}, \mathbb{S}'))$.

Proof. For the direction from left to right: suppose that Z is a bisimulation between $\mathbb{S}$ and $\mathbb{S}'$ linking s and s'. Suppose that $\exists$, starting from position (s, s'), always chooses the relation Z itself as the local bisimulation. A straightforward verification, by induction on the length of the match, shows that this strategy always provides her with a legitimate move, and that it keeps her alive forever. This proves that it is a winning strategy.

For the converse direction, it suffices to show that the relation $\{(t, t') \in S \times S' \mid (t, t') \in \mathrm{Win}_\exists(\mathcal{B}(\mathbb{S}, \mathbb{S}'))\}$ itself is in fact a bisimulation. We leave the details for the reader. QED

Remark 1.28 ► `The bisimilarity game should not be confused with the` *`bisimulation game`*`.`

◁

Bisimulations via relation lifting

Together, the back- and forth clause of the definition of a bisimulation express that the pair of respective successor sets of two bisimilar states must belong to the so-called *Egli-Milner lifting* $\overline{\wp}Z$ of the bisimulation Z. In fact, the notion of a bisimulation can be completely defined in terms of *relation lifting*.

Definition 1.29 Given a relation $Z \subseteq A \times A'$, define the relation $\overline{\wp}Z \subseteq \wp A \times \wp A'$ as follows:

$$\begin{array}{rcll} \overline{\wp}Z & := & \{(X, X') \mid & \text{for all } x \in X \text{ there is an } x' \in X' \text{ with } (x, x') \in Z \\ & & \& & \text{for all } x' \in X' \text{ there is an } x \in X \text{ with } (x, x') \in Z\}. \end{array}$$

Similarly, define, for a Kripke functor $\mathsf{K} = \mathsf{K}_{\mathsf{D},\mathsf{P}}$, the relation $\overline{\mathsf{K}}Z \subseteq \mathsf{K}A \times \mathsf{K}A'$ as follows:

$$\overline{\mathsf{K}}Z := \{((\pi, X), (\pi', X')) \mid \pi = \pi' \text{ and } (X_d, X'_d) \in \overline{\wp}Z \text{ for each } d \in \mathsf{D}\}.$$

The relations $\overline{\wp}Z$ and $\overline{\mathsf{K}}Z$ are called the *liftings* of Z with respect to $\wp$ and K, respectively. We say that $Z \subseteq A \times A'$ is *full on* $B \in \wp A$ and $B' \in \wp A'$ if $(B, B') \in \overline{\wp}Z$. ◁

It is completely straightforward to check that a nonempty relation Z linking two transition systems $\mathbb{S}$ and $\mathbb{S}'$ is a local bisimulation for two states s and s' iff $(\sigma(s), \sigma'(s')) \in \overline{\mathsf{K}}Z$. In particular, $\exists$'s move in the bisimilarity game at a position (s, s') consists of choosing a binary relation Z such that $(\sigma(s), \sigma'(s')) \in \overline{\mathsf{K}}Z$. The following characterization of bisimulations is also an immediate consequence.

Proposition 1.30 *Let $\mathbb{S}$ and $\mathbb{S}'$ be two Kripke coalgebras for some Kripke functor K, and let $Z \subseteq S \times S'$ be some relation. Then*

$$Z \text{ is a bisimulation iff } (\sigma(s), \sigma'(s')) \in \overline{\mathsf{K}}Z \text{ for all } (s, s') \in Z. \tag{1}$$

1.4 Finite models and computational aspects

- `complexity of model checking`
- `filtration & polysize model property`
- `complexity of satisfiability`
- `complexity of global consequence`

1.5 Modal logic and first-order logic

- `modal logic is the bisimulation invariant fragment of first-order logic`

1.6 Complete derivation systems for modal logic

1.7 The cover modality

As we will see now, there is an interesting alternative for the standard formulation of basic modal logic in terms of boxes and diamonds. This alternative set-up is based on a connective which turns a *set* of formulas into a formula. We first restrict attention to the monomodal case.

Definition 1.31 Let Φ be a finite set of formulas. Then $\nabla\Phi$ is a formula, which holds at a state s in a Kripke model if *every* formula in Φ holds at *some* successor of s, while at the same time, *every* successor of s makes *some* formula in Φ true. The operator ∇ is called the *cover modality.* $\lhd$

It is not so hard to see that the cover modality can be defined in the standard modal language:

$$\nabla\Phi \equiv \Box \bigvee \Phi \wedge \bigwedge \Diamond\Phi, \tag{2}$$

where $\Diamond\Phi$ denotes the set $\{\Diamond\varphi \mid \varphi \in \Phi\}$. Things start to get interesting once we realize that both the ordinary diamond $\Diamond$ *and* the ordinary box $\Box$ can be expressed in terms of the cover modality (and the disjunction):

$$\begin{array}{rcl} \Diamond\varphi & \equiv & \nabla\{\varphi, \top\}, \\ \Box\varphi & \equiv & \nabla\varnothing \vee \nabla\{\varphi\}. \end{array} \tag{3}$$

Here, as always, we use the convention that $\bigvee \varnothing = \bot$ and $\bigwedge \varnothing = \top$.

Remark 1.32 Observe that this definition involves the $\forall\exists\&\forall\exists$ pattern that we know from the definition of a bisimulation. The fundamental concept is the notion of *relation lifting* $\overline{\wp}$ defined in the previous section. In other words, the semantics of the cover modality can be expressed in terms of relation lifting. To be more precise, observe that we may think of the forcing or satisfaction relation $\Vdash$ simply as a binary relation between states and formulas. Then we find that

$$\mathbb{S}, s \Vdash \nabla\Phi \text{ iff } (\sigma_R(s), \Phi) \in \overline{\wp}(\Vdash).$$

for any pointed Kripke model $(\mathbb{S}, s)$ and any finite set Φ of formulas. $\lhd$

Remark 1.33 In the special case where $\Phi = \varnothing$ we find that $\mathbb{S}, s \Vdash \nabla\varnothing$ iff $R[s] = \varnothing$, that is, s has *no* successors. Using this it is easy to see that $\top = \nabla\{\top\} \vee \nabla\varnothing$. $\lhd$

Given that ∇ and $\{\Diamond, \Box\}$ are mutually expressible, we obtain an expressively equivalent language $\mathtt{ML}_\nabla$ if we replace $\Box$ and $\Diamond$ with the cover modality. As we will see further on it will be convenient for us to use a format for this language in which not only the cover modality, but also the disjunction and conjunction connectives take finite sets of formulas as their argument. That is, rather then working with disjunction and conjunction as *binary* connectives, we will work with their *finitary* versions. This perspective also allows us to omit the constants $\bot$ and $\top$ from the basic syntax, since we may consider them as abbreviations: $\bot := \bigvee\varnothing$ and $\top := \bigwedge\varnothing$.

Definition 1.34 The formulas of the language $\mathtt{ML}_\nabla$ are given by the following grammar:

$$\varphi ::= p \mid \overline{p} \mid \bigvee\Phi \mid \bigwedge\Phi \mid \nabla\Phi$$

where p is a propositional variable, and $\Phi \subseteq \mathtt{ML}_\nabla$. $\lhd$

Proposition 1.35 *The languages* $\mathtt{ML}$ *and* $\mathtt{ML}_\nabla$ *are equally expressive.*

Proof. Immediate by (2) and (3). QED

The real importance of the cover modality is that it allows us to almost completely eliminate the Boolean *conjunction*. This remarkable fact is based on the following modal distributive law. Recall from Definition 1.29 that a relation $Z \subseteq A \times A'$ is *full on* A and A' if $(A, A') \in \overline{\wp}Z$, or in other words: $A \subseteq \mathsf{Dom}(Z)$ and $A' \subseteq \mathsf{Ran}(Z)$.

Proposition 1.36 (Binary Modal Distributive Law) *Let* Φ *and* Φ' *be two sets of formulas. Then the following two formulas are equivalent:*

$$\nabla\Phi \wedge \nabla\Phi' \equiv \bigvee\{\nabla\Gamma_Z \mid Z \text{ is full on } \Phi \text{ and } \Phi'\}, \tag{4}$$

where, given a relation $Z \subseteq \Phi \times \Phi'$, *we define*

$$\Gamma_Z := \{\varphi \wedge \varphi' \mid (\varphi, \varphi') \in Z\}.$$

Proof. For the direction from left to right, suppose that $\mathbb{S}, s \Vdash \nabla\Phi \wedge \nabla\Phi'$. Let $Z \subseteq \Phi \times \Phi'$ consist of those pairs (φ, φ') such that the conjunction $\varphi \wedge \varphi'$ is true at some successor t of s. It is then straightforward to verify that Z is full on Φ and Φ', and that $\mathbb{S}, s \Vdash \nabla\Gamma_Z$.

The converse direction follows fairly directly from the definitions. QED

As a corollary of Proposition 1.36 we can restrict the use of conjunction in modal logic to that of a *special conjunction* connective $\bullet$ which may only be applied to a pair consisting of a set of literals and a ∇-formula (or, a certain set of ∇-formulas in the polymodal case). The intended reading of the bullet operator is as follows:

$$\alpha \bullet \Phi \equiv (\bigwedge \alpha) \wedge \nabla \Phi.$$

Definition 1.37 Fix a finite set P of proposition letters. Then the set $\mathtt{DML}(\mathsf{P})$ of *disjunctive monomodal formulas* in P is given by the following grammar:

$$\varphi ::= \top \mid \bigvee \Phi \mid \alpha \bullet \Phi,$$

where α is a finite set of literals over P and Φ is a finite set of formulas in $\mathtt{DML}(\mathsf{P})$. $\lhd$

Note that the proposition letters in P and their negations themselves do not qualify as disjunctive formulas. However, these formulas are easily seen to be equivalent to disjunctive formulas: for instance, we have $\ell \equiv \{\ell\} \bullet \{\top\} \vee \{\ell\} \bullet \varnothing$, for any literal ℓ.

Remark 1.38 In the above definition we do not need to list the formula $\bot$ explicitly as a disjunctive formula, since we can still see it as an abbreviation: $\bot := \bigvee \varnothing$. This is different for the formula $\top$, however. Since we no longer have $\bigwedge$ as a connective, we cannot use it to define $\top$. For this reason we have added $\top$ as a primitive constant. $\lhd$

The following theorem states that every modal formula can be rewritten into an equivalent disjunctive normal form.

Theorem 1.39 *Let P be a set of proposition letters. Then there are effective ways to transform an arbitrary formula in $\mathtt{ML}(\mathsf{P})$ into an equivalent formula in $\mathtt{DML}(\mathsf{P})$, and vice versa. As a corollary, the languages $\mathtt{ML}(\mathsf{P})$ and $\mathtt{DML}(\mathsf{P})$ are expressively equivalent.*

We leave the proof of this result as an exercise to the reader.

Remark 1.40 In the polymodal case we adapt the definition as follows. Let $\Phi = \{\Phi_d \mid d \in \mathsf{D}\}$ be a D-indexed family of formula sets. Then we write $\nabla_{\mathsf{D}} \Phi := \bigwedge_{d \in \mathsf{D}} \nabla_d \Phi_d$, where ∇_d is the cover modality associated with the action d. The following grammar defines the set $\mathtt{DML}_{\mathsf{D}}(\mathsf{P})$ of *disjunctive polymodal formulas* in D and P

$$\varphi ::= \top \mid \bigvee \Phi \mid \alpha \bullet \nabla_{\mathsf{D}} \Phi,$$

where $\alpha \subseteq_\omega \mathtt{Lit}(\mathsf{P})$ and Φ is an D-indexed family of finite sets of $\mathtt{DML}_{\mathsf{D}}(\mathsf{P})$-formulas. One may then formulate and prove a polymodal version of Theorem 1.39, relating $\mathtt{ML}_{\mathsf{D}}$ and $\mathtt{DML}_{\mathsf{D}}$. $\lhd$

Notes

Modal logic has a long history in philosophy and mathematics, for an overview we refer to Blackburn, de Rijke and Venema [4]. The use of modal formalisms as specification languages in process theory goes back at least to the 1970s, with Pratt [25] and Pnueli [24] being two influential early papers.

The notion of bisimulation, which plays an important role in modal logic and process theory alike, was first introduced in a modal logic context by van Benthem [3], who proved that modal logic is the bisimulation invariant fragment of first-order logic. The notion was later, but independently, introduced in a process theory setting by Park [23]. At the time of writing we do not know who first took a game-theoretical perspective on the semantics of modal logic. The cover modality ∇ was introduced independently by Moss [19] and Janin & Walukiewicz [12].

Readers who want to study modal logic in more detail are referred to Blackburn, de Rijke and Venema [4] or Chagrov & Zakharyaschev [7].

Exercises

Exercise 1.1 Prove Theorem 1.18.

Exercise 1.2 Prove that the Hennessy-Milner theorem (Theorem 1.21) also holds if only one of the two structures is finitely branching.

Exercise 1.3 (bisimilarity game) Consider the following version $\mathcal{B}_\omega(\mathbb{S}, \mathbb{S}')$ of the bisimilarity game between two transition systems $\mathbb{S}$ and $\mathbb{S}'$. Positions of this game are of the form either $(s, s', \forall, \alpha)$, $(s, s', \exists, \alpha)$ or (Z, α), with $s \in S$, $s' \in S'$, $Z \subseteq S \times S'$ and α either a natural number or ω. The admissible moves for $\exists$ and $\forall$ are displayed in the following table:

Position	Player	Admissible moves
$(s, s', \forall, \alpha)$	$\forall$	$\{(s, s', \exists, \beta) \mid \beta < \alpha\}$
$(s, s', \exists, \alpha)$	$\exists$	$\{(Z, \alpha) \mid Z$ is a local bisimulation for s and s' $\}$
(Z, α)	$\forall$	$\{(s, s', \forall, \alpha) \mid (s, s') \in Z\}$

Note that all matches of this game have finite length.

We write $\mathbb{S}, s \leftrightarroweq_\alpha \mathbb{S}', s'$ to denote that $\exists$ has a winning strategy in the game $\mathcal{B}_\omega(\mathbb{S}, \mathbb{S}')$ starting at position $(s, s', \forall, \alpha)$. It is not hard to see that $\mathbb{S}, s \leftrightarroweq_\omega \mathbb{S}', s'$ iff $\mathbb{S}, s \leftrightarroweq_k \mathbb{S}', s'$ for all $k < \omega$.

(a) Give concrete examples such that $\mathbb{S}, s \leftrightarroweq_\omega \mathbb{S}', s'$ but not $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$.
(Hint: think of two modally equivalent but not bisimilar states.)

(b) Let $k \geq 0$ be a natural number. Prove that, for all $\mathbb{S}, s$ and $\mathbb{S}', s'$:

$$\mathbb{S}, s \leftrightarroweq_k \mathbb{S}', s' \;\Rightarrow\; \mathbb{S}, s \equiv_k \mathbb{S}', s'.$$

Here $\equiv_k$ denotes the modal equivalence relation with respect to formulas of modal depth at most k. Here we use a slightly nonstandard notion of modal depth, defined as follows: $d(\bot), d(\top) := 0$, $d(p), d(\overline{p}) := 1$ for $p \in \mathsf{P}$, $d(\varphi \wedge \psi), d(\varphi \vee \psi) := \max(d(\varphi), d(\psi))$, and $d(\Diamond\varphi), d(\Box\varphi) := 1 + d(\varphi)$.

(c) Let $\mathbb{S}$ and $\mathbb{S}'$ be finitely branching transition systems. Prove *directly* (i.e., without using part (b)) that (i) $\Rightarrow$ (ii), for all $s \in S$ and $s' \in S'$:

(i) $\mathbb{S}, s \leftrightarroweq_\omega \mathbb{S}', s'$

(ii) $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$.

(d)* Does the implication in (c) hold in the case that only *one* of the two transition systems is finitely branching?

Exercise 1.4 Let Φ and Θ be finite sets of formulas. Prove that

$$\nabla\big(\Phi \cup \{\bigvee \Theta\}\big) \;\equiv\; \bigvee \big\{\nabla\big(\Phi \cup \Theta'\big) \;\mid\; \varnothing \neq \Theta' \subseteq \Theta\big\}.$$

Exercise 1.5 Prove Theorem 1.39.

2 The modal μ-calculus: basics

This chapter is a first introduction to the modal μ-calculus. We define the language, discuss some syntactic issues, and then proceed to its game-theoretic semantics. As a first result, we prove that the modal μ-calculus is bisimulation invariant, and has a strong, 'bounded' version of the tree model property. We then provide some basic information concerning the main complexity measures of μ-calculus formulas: size and alternation depth.

To introduce the formalism, we start with a simple example.

Example 2.1 Consider the formula $\langle d^* \rangle p$ from propositional dynamic logic. By definition, this formula holds at those points in an LTS $\mathbb{S}$ from which there is a finite R_d-path, of unspecified length, leading to a state where p is true.

We leave it for the reader to prove that

$$\mathbb{S}, s \Vdash \langle d^* \rangle p \leftrightarrow (p \vee \langle d \rangle \langle d^* \rangle p)$$

for any pointed transition system $(\mathbb{S}, s)$ (here we write $\langle d \rangle$ rather than $\Diamond_d$). Informally, one might say that $\langle d^* \rangle p$ is a *fixed point* of the formula $p \vee \langle d \rangle x$, or a solution of the 'equation'

$$x \equiv p \vee \langle d \rangle x. \tag{5}$$

One may show, however, that $\langle d^* \rangle p$ is not the only fixpoint of (5). If we let ∞_d denote a formula that is true at those states of a transition system from which an infinite d-path emanates, then the formula $\langle d^* \rangle p \vee \infty_d$ is another fixed point of (5).

In fact, one may prove that the two mentioned fixpoints are the smallest and largest possible solutions of (5), respectively. ◁

As we will see in this chapter, the modal μ-calculus allows one to explicitly refer to such smallest and largest solutions. For instance, as we will see further on, the smallest and largest solution of the 'equation' (5) will be written as $\mu x.p \vee \langle d \rangle x$ and $\nu x.p \vee \langle d \rangle x$, respectively. Generally, the basic idea underlying the modal μ-calculus is to enrich the language of basic modal logic with two explicit fixpoint operators, μ and ν, respectively. Syntacticlly, these operators behave like quantifiers in first-order logic, in the sense that the application of a fixpoint operator μx to a formula φ *binds* all (free) occurrences of the proposition letter x in φ. The word 'fixpoint' indicates that semantically, the formulas $\mu x\, \varphi$ and $\nu x\, \varphi$ are both 'solutions' to the 'equation' $x \equiv \varphi(x)$, in the sense that, writing $\equiv$ for semantic equivalence, we have both

$$\begin{array}{rrcl} & \mu x\, \varphi & \equiv & \varphi[\mu x\, \varphi / x] \\ \text{and} & \nu x\, \varphi & \equiv & \varphi[\nu x\, \varphi / x], \end{array} \tag{6}$$

where $[\mu x.\varphi / x]$ denotes the operation of substituting $\mu x\, \varphi$ for every free occurrence of x. In other words, both $\mu x\, \varphi$ and $\nu x\, \varphi$ are equivalent to their respective *unfoldings*, $\varphi[\mu x\, \varphi / x]$ and $\varphi[\nu x\, \varphi / x]$.

To arrive at this semantics of modal fixpoint formulas one can take two roads. In Chapter 3 we will introduce the algebraic semantics of $\mu x\, \varphi$ and $\nu x\, \varphi$ in an LTS $\mathbb{S}$, in terms of the *least* and *greatest fixpoint*, respectively, of some algebraically defined meaning function. For this purpose, we will consider the formula φ as an *operation* on the power set of (the state space of) $\mathbb{S}$, and we have to prove that this operation indeed has a least and a greatest fixpoint. As we will see, this formal definition of the semantics of the modal μ-calculus may be mathematically transparent, but it is of little help when it comes to unravelling and understanding the actual meaning of individual formulas. In practice, it is much easier to work with the *evaluation games* that we will introduce in this chapter.

This framework builds on the game-theoretical semantics for ordinary modal logic as described in Subsection 1.2, extending it with features for the fixpoint operators and for the bound variables of fixpoint formulas (such as x in the formula $\mu x.p \vee \Diamond x$). The key difference lies in the fact that when a match of an evaluation game reaches a position of the form (x, s), with x a *bound* variable, then an equation such as (5) is used to *unfold* the variable x into its associated formula (in the example, the formula $p \vee \Diamond x$).

As a consequence, the flavour of these games is remarkably different from the evaluation games we met before. Recall that in evaluation matches for *basic* modal formulas, the formula is broken down, step by step, until we can declare a winner of the match. From this it follows that the length of such a match is *bounded* by the length of the formula. Evaluation matches for fixpoint formulas, on the other hand, can last forever, if some fixpoint variables are unfolded infinitely often. Hence, the game-theoretic semantics for fixpoint logics takes us to the area of *infinite games*. In this Chapter we keep our treatment of infinite games informal, in Chapter 5 the reader can find precise definitions of all notions that we introduce here.

2.1 Basic syntax

Formulas

As announced already in the previous chapter, in the case of fixpoint formulas we will usually work with formulas in *positive normal form* in which the only admissible occurrences of the negation symbol is in front of atomic formulas.

Definition 2.2 Given a set D of atomic actions, we define llection $\mu\mathtt{ML}_{\mathsf{D}}$ of *(poly-)modal fixpoint formulas* as follows:

$$\varphi \ ::= \ \top \mid \bot \mid p \mid \overline{p} \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond_d \varphi \mid \Box_d \varphi \mid \mu x\, \varphi \mid \nu x\, \varphi$$

where p and x are propositional variables, and $d \in \mathsf{D}$. There is a restriction on the formation of the formulas $\mu x\, \varphi$ and $\nu x\, \varphi$, namely, that the formula φ is *positive* in x. That is, all occurrences of x in φ are *positive*, or, phrasing it yet differently, no occurrence of x in φ may be in the form of the negative literal $\overline{x}$.

In case the set D of atomic actions is a singleton, we will simply speak of the *modal μ-calculus*, notation: $\mu\mathtt{ML}$.

The syntactic combinations μx and νx are called the *least* and *greatest fixpoint operators*, respectively. We use the symbols η and λ to denote either μ or ν, and we define $\overline{\mu} := \nu$ and $\overline{\nu} := \mu$. ◁

A formula of the form $\eta x\, \varphi$ is called a *fixpoint formula*, and, more specifically, a *μ-formula* if $\eta = \mu$ and a *ν-formula* if $\eta = \nu$. Furthermore, conjunctions and disjunctions will sometimes be called *boolean* μML-formulas, and formulas of the form $\Diamond_d\varphi$ or $\Box_d$ will sometimes be called *modal*.

Convention 2.3 In order to increase readability by reducing the number of brackets, we adopt some standard scope conventions. We let the unary modal connectives, $\Diamond$ and $\Box$, bind stronger than the binary propositional connectives $\wedge$, $\vee$ and $\to$, and use associativity to the left for the connectives $\wedge$ and $\vee$. As an example, we will abbreviate the formula $(\Diamond p \wedge q)$ as $\Diamond p \wedge q$.

Furthermore, we use 'dot notation' to indicate that the fixpoint operators preceding the dot have maximal scope. For instance, $\mu p.\Diamond p \wedge q$ denotes the formula $\mu p\,(\Diamond p \wedge q)$, and not the formula $((\mu p\,\Diamond p) \wedge q)$. As a final example, $\mu x.\overline{p} \vee \Box x \vee y \vee \nu y.q \wedge \Box(x \vee y)$ stands for $\mu x\left(\big((\overline{p} \vee \Box x) \vee y\big) \vee \nu y\,(q \wedge \Box(x \vee y))\right)$.

Remark 2.4 An alternative definition of the language of the modal μ-calculus makes a distinction between propositional *variables* and *proposition letters*. Formulas are now defined as follows:

$$\varphi \ ::= \ \top \mid \bot \mid p \mid \overline{p} \mid x \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond_d\varphi \mid \Box_d\varphi \mid \mu x\,\varphi \mid \nu x\,\varphi$$

where p is a proposition letter, x a propositional variable, and d is an atomic action. In this framework, only propositional variables can be bound. $\lhd$

Length and syntax tree of a formula

There are various ways to *measure* a μ-calculus formula. The most basic measure of a formula is its *length*, which basically corresponds to its number of symbols.

Definition 2.5 Given a μ-calculus formula ξ, we define its *length* $|\xi|^\ell$ inductively as follows:

$$\begin{array}{lcll}
|\varphi|^\ell & := & 1 & \text{if } \varphi \text{ is atomic} \\
|\varphi_0 \circledast \varphi_1|^\ell & := & 1 + |\varphi_0|^\ell + |\varphi_1|^\ell & \text{where } \circledast \in \{\wedge, \vee\} \\
|\heartsuit\varphi|^\ell & := & 1 + |\varphi|^\ell & \text{where } \heartsuit \in \{\Diamond, \Box\} \\
|\eta x.\varphi|^\ell & := & 1 + |\varphi|^\ell & \text{where } \eta \in \{\mu, \nu\}
\end{array}$$

$\lhd$

We assume that the reader is familiar with the concept of the *syntax tree* or *construction tree* $\mathbb{T}_\xi$ of a formula ξ. We will not give a formal definition of this structure, but confine ourselves to an example: in Figure 2.1 we display the syntax tree of the μ-calculus formula $\mu x.(\overline{p} \vee \Diamond x) \vee \nu y.(q \wedge \Box(x \vee y))$. Note that the *length* of a formula corresponds to the number of nodes of its syntax tree, and that an *occurrence* of a certain symbol in a formula may be associated with some node in the formula's syntax tree that is labelled with that symbol; occurrences of literals correspond to *leaves* of the tree.

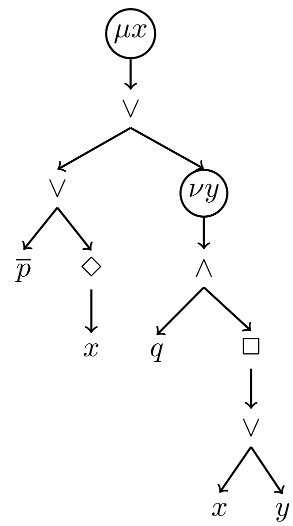


Figure 1: A syntax tree

Subformulas and free/bound variables

The concepts of *subformula* and *proper subformula* are extended from basic modal logic to the modal μ-calculus in the obvious way.

Definition 2.6 We define the set $Sf_0(\xi)$ of *direct subformulas* of a formula $\xi \in \mu\mathtt{ML}$ via the following case distinction:

$$
\begin{array}{llll}
Sf_0(\xi) & := & \varnothing & \text{if } \xi \in \mathtt{At}(\mathsf{P}) \\
Sf_0(\xi_0 \circledast \xi_1) & := & \{\xi_0, \xi_1\} & \text{where } \circledast \in \{\wedge, \vee\} \\
Sf_0(\heartsuit \xi_0) & := & \{\xi_0\} & \text{where } \heartsuit \in \{\Diamond, \Box\} \\
Sf_0(\eta x.\xi_0) & := & \{\xi_0\} & \text{where } \eta \in \{\mu, \nu\},
\end{array}
$$

and we write $\varphi \triangleleft_0 \xi$ if $\varphi \in Sf_0(\xi)$.

For any formula $\xi \in \mu\mathtt{ML}$, $Sf(\xi)$ is the least set of formulas which contains ξ and is closed under taking direct subformulas. Elements of the set $Sf(\xi)$ are called *subformulas* of ξ, and we write $\varphi \trianglelefteq \xi$ ($\varphi \triangleleft \psi$) if φ is a subformula (proper subformula, respectively) of ξ.

The *(subformula) dag* of a formula ξ is defined as the directed acyclic graph $(Sf(\xi), \triangleright_0)$, where $\triangleright_0$ is the converse of the direct subformula relation $\triangleleft_0$. ◁

▶ `Give an example comparing the syntax tree of a formula to its subformula dag.`

Syntactically, the fixpoint operators are very similar to the quantifiers of first-order logic in the way they *bind* variables.

Definition 2.7 Fix a formula φ. The sets $FV(\varphi)$ and $BV(\varphi)$ of *free* and *bound variables* of φ are defined by the following induction on φ:

$$
\begin{array}{lcllcl}
FV(\bot) & := & \varnothing & BV(\bot) & := & \varnothing \\
FV(\top) & := & \varnothing & BV(\top) & := & \varnothing \\
FV(p) & := & \{p\} & BV(p) & := & \varnothing \\
FV(\overline{p}) & := & \{p\} & BV(\overline{p}) & := & \varnothing \\
FV(\varphi \lor \psi) & := & FV(\varphi) \cup FV(\psi) & BV(\varphi \lor \psi) & := & BV(\varphi) \cup BV(\psi) \\
FV(\varphi \land \psi) & := & FV(\varphi) \cup FV(\psi) & BV(\varphi \land \psi) & := & BV(\varphi) \cup BV(\psi) \\
FV(\Diamond_d \varphi) & := & FV(\varphi) & BV(\Diamond_d \varphi) & := & BV(\varphi) \\
FV(\Box_d \varphi) & := & FV(\varphi) & BV(\Box_d \varphi) & := & BV(\varphi) \\
FV(\eta x.\varphi) & := & FV(\varphi) \setminus \{x\} & BV(\eta x.\varphi) & := & BV(\varphi) \cup \{x\}
\end{array}
$$

For a finite set of propositional variables P, we let $\mu\mathtt{ML}_{\mathsf{D}}(\mathsf{P})$ denote the set of $\mu\mathtt{ML}_{\mathsf{D}}$-formulas φ of which all free variables belong to P. ◁

Formulas like $x \lor \mu x.((p \lor x) \land \Box \nu x.\Diamond x)$ may be well formed, but in practice they are very hard to read and to work with. In the sequel we will often work with formulas in which every bound variable uniquely determines a subformula where it is bound, and almost exclusively with formulas in which no variable has both free and bound occurrences in φ.

Definition 2.8 A formula $\varphi \in \mu\mathtt{ML}_{\mathsf{D}}$ is *tidy* if $FV(\varphi) \cap BV(\varphi) = \varnothing$, and *clean* if in addition with every bound variable x of φ we may associate a unique subformula of the form $\eta x.\delta$. In the latter case we let $\varphi_x = \eta_x x.\delta_x$ denote this unique subformula. ◁

Convention 2.9 As a notational convention, we will use the letters $p, q, r, \ldots$ and $x, y, z, \ldots$ to denote, respectively, the free and the bound propositional variables of a $\mu\mathtt{ML}_{\mathsf{D}}$-formula. This convention can be no more than a guideline, since the division between bound and free variables may not be the same for a formula and its subformulas. For instance, the variable x is bound in $\mu x.p \lor \Diamond x$, but free in its subformula $p \lor \Diamond x$.

Remark 2.10 In the alternative definition of the language of the modal μ-calculus as discussed in Remark 2.4, just like in first-order logic one makes a difference between *open formulas* (which may contain free variables) and *sentences* (which may not). Observe that the sentences correspond to the tidy formulas in our framework. For instance, $\mu x\,(p \lor \Diamond x)$ is a sentence, $\mu x\,(y \lor \Diamond x)$ is an open formula, and $\mu p\,(x \lor \Diamond p)$ is not a well-formed formula (assuming that p is a proposition letter, and x is a variable). ◁

Substitution & unfolding

The syntactic operation of substitution is ubiquitous in any account of the modal μ-calculus, first of all because it features in the basic operation of unfolding a fixpoint formula. As usual in the context of a formal language featuring operators that *bind* variables, the precise definition of a substitution operation needs some care. In particular, we need to protect the substitution operation from variable capture.

Example 2.11 To give a concrete example, suppose that we would naively define a substitution operation ψ/x by defining $\varphi[\psi/x]$ to be the formula we obtain from the formula φ by replacing every free occurrences of x with the formula ψ. Now consider the formula $\varphi(q) = \mu p.q \vee \Diamond p$ expressing the reachability of a q-state in finitely many steps. If we substitute p for q in φ, we would expect the resulting formula to express the reachability of a p-state in finitely many steps, but the formula we obtain is $\varphi[p/q] = \mu p.p \vee \Diamond p$, which says something rather different (in fact, it happens to be equivalent to $\bot$). Even worse, the substitution $[\overline{p}/q]$ would produce a syntactic string $\varphi[\overline{p}/q] = \mu p.\overline{p} \vee \Diamond p$ which is not even a well-formed formula. $\lhd$

To avoid such anomalies, for the time being we shall only consider substitutions ξ/x applied to formulas where ξ is free for x.

Definition 2.12 Let ξ and x be respectively a modal μ-calculus formula and a propositional variable. We define what it means for ξ to be *free for x in a formula φ* by the following induction on the complexity of φ:

- if φ is an atomic formula then ξ is free for x in φ, unless $\varphi = \overline{x}$[1];
- ξ is free for x in $\varphi_0 \circledast \varphi_1$ if it is free for x in both φ_0 and φ_1;
- ξ is free for x in $\heartsuit\psi$ if it is free for x in ψ;
- ξ is free for x in $\eta y\,\psi$ if $x \notin FV(\eta y\,\psi)$ or if $y \notin FV(\xi)$ and ξ is free for x in ψ.

$\lhd$

Informally, ξ is *free for x in φ* if φ is positive in x and no free variable in ξ gets bound, after substitution, by a fixpoint operator in φ. A special case of this, that we shall encounter frequently, is the following.

Proposition 2.13 *Let φ, ξ and x be respectively two modal μ-calculus formulas and a propositional variable, such that $FV(\xi) \cap BV(\varphi) \subseteq \{x\}$. Then ξ is free for x in φ.*

Definition 2.14 Let $\{\xi_z \mid z \in Z\}$ be a set of modal μ-calculus formulas, indexed by a set of variables Z, let $\varphi \in \mu\mathtt{ML}$ be positive in each $z \in Z$, and assume that each ξ_z is free for z in φ. We inductively define the *simultaneous substitution* $[\xi_z/z \mid z \in Z]$ as the following operation on $\mu\mathtt{ML}$:

$$\begin{array}{lcl}
\varphi[\xi_z/z \mid z \in Z] & := & \begin{cases} \xi_z & \text{if } \varphi = z \in Z \\ \varphi & \text{if } \varphi \text{ is atomic but } \varphi \notin Z \end{cases} \\
(\heartsuit\psi)[\xi_z/z \mid z \in Z] & := & \heartsuit\psi[\xi_z/z \mid z \in Z] \\
(\varphi_0 \circledast \varphi_1)[\xi_z/z \mid z \in Z] & := & \varphi_0[\xi_z/z \mid z \in Z] \circledast \varphi_1[\xi_z/z \mid z \in Z] \\
(\eta x.\psi)[\xi_z/z \mid z \in Z] & := & \eta x.\psi[\xi_z/z \mid z \in Z \setminus \{x\}]
\end{array}$$

[1] Strictly speaking, this condition is not needed. In particular, as a separate atomic case of our inductive definition, we could define the outcome of the substitution $\overline{p}[\psi/p]$ to be the *negation* of the formula ψ (suitably defined). However, we will only need to look at substitutions $\varphi[\psi/z]$ where we happen to know that φ is positive in z. As a result, our simplified definition does not impose a real restriction.

In case Z is a singleton, say $Z = \{z\}$, we will simply write $\varphi[\xi_z/z]$, or $\varphi(\xi)$ if z is understood. $\lhd$

▶ `Add some examples`

Remark 2.15 In case ψ is not free for some $z \in Z$ in ξ, we take a standard approach using *alphabetical variants*. Roughly, two formulas are alphabetical variants if we can obtain one from the other by renaming bound variables. We then define a correct version of the substitution $\xi[\psi_z/z \mid z \in Z]$ as follows: first we take some canonically chosen alphabetical variant ξ' of ξ such that each ψ_z is free for z in ξ', and then we set

$$\xi[\psi_z/z \mid z \in Z] := \xi'[\psi_z/z \mid z \in Z].$$

However, in almost all situations that we will encounter we will only need perform substitutions that are 'safe' in the sense that the substituted formula is free for the variable it replaces. This means that generally we may avoid taking alphabetical variants. Situations where this is not the case will be explicitly marked. The reason for taking such care is that the operation of taking alphabetical variants is not completely harmless when it comes to size issues. We will come back to this matter in more detail later. $\lhd$

The following proposition is a well known observation in areas where syntax is used that features variable binding. Note however that our version below is a bit subtler than usual since we do not allow the renaming of bound variables.

Proposition 2.16 *Let φ, χ and ρ be μ-calculus formulas, and let x and y be distinct variables such that x is free in φ but not in ρ. Furthermore, assume that χ is free for x in φ and that ρ is free for y in $\varphi[\chi/x]$. Then ρ is free for y in both φ and χ, $\chi[\rho/y]$ is free for x in $\varphi[\rho/y]$, and we have*

$$\varphi[\chi/x][\rho/y] = \varphi[\rho/y][(\chi[\rho/y])/x]. \tag{7}$$

Proof. The proposition can be proved by a straightforward but rather tedious induction on the complexity of φ. We omit details. QED

Unfolding

The reason that the modal μ-calculus, and related formalisms, are called *fixpoint logics* is that, for $\eta = \mu/\nu$, the meaning of the formula $\eta x.\chi$ in a model $\mathbb{S}$ is given as the least/greatest *fixpoint* of the semantic map expressing the dependence of the meaning of χ on (the meaning of) the variable x. As a consequence, the following equivalence lies at the heart of semantics of $\mu\mathtt{ML}$:

$$\eta x.\chi \equiv \chi[\eta x.\chi/x] \tag{8}$$

In words: every formula is equivalent to its *unfolding*.

Definition 2.17 Given a formula $\eta x.\chi \in \mu\mathtt{ML}$, we call the formula $\mathtt{unf}(\xi) := \chi[\eta x.\chi/x]$ its *unfolding*. $\lhd$

Remark 2.18 Unfolding is the central operation in taking the closure of a formula that we are about to define. Unfortunately, the collection of clean formulas is not closed under unfolding (unless we take alphabetical variants). Consider for instance the formula $\varphi(p) = \nu q.\Diamond q \wedge p$, then we see that the formula $\mu p.\varphi$ is clean, but its unfolding $\varphi[\mu p.\varphi/p] = \nu q.\Diamond q \wedge \mu p\, \nu q.\Diamond q \wedge p$ is not. Furthermore, our earlier observation that the naive version of substitution may produce 'formulas' that are not well-formed applies here as well. For instance, with χ denoting the formula $\overline{p} \wedge \nu p.\Box(x \vee p)$, naively unfolding the (untidy) formula $\mu x.\chi$ would produce the ungrammatical $\overline{p} \wedge \nu p.\Box\big((\mu x.\overline{p} \wedge \nu p.\Box(x \vee p)) \vee p\big)$. $\lhd$

Fortunately, the condition of *tidyness* guarantees that we may calculate unfoldings without moving to alphabetical variants, since we can prove that the formula $\eta x.\chi$ is free for x in χ, whenever $\eta x.\chi$ is tidy. In addition, tidyness is preserved under taking unfoldings.

Proposition 2.19 *Let $\eta x.\chi \in \mu\mathtt{ML}$ be a tidy formula. Then*
1) $\eta x.\chi$ is free for x in χ;
2) $\chi[\eta x.\chi/x]$ is tidy as well.

Proof. For part 1), take a variable $y \in FV(\eta x.\chi)$. Then obviously y is distinct from x, while $y \notin BV(\eta x.\chi)$ by tidyness. Clearly then we find $y \notin BV(\chi)$; in other words, χ has *no* subformula of the form $\lambda y.\psi$. Hence it trivially follows that $\eta x.\chi$ is free for x in χ.

Part 2) is immediate by the following identities:

$$\begin{array}{lclcl} FV(\chi[\eta x.\chi/x]) & = & (FV(\chi) \setminus \{x\}) \cup FV(\eta x.\chi) & = & FV(\eta x.\chi) \\ BV(\chi[\eta x.\chi/x]) & = & BV(\chi) \cup BV(\eta x.\chi) & = & BV(\eta x.\chi) \end{array}$$

which can easily be proved. QED

Dependency order

An important role in the theory of the modal μ-calculus is played by a certain order $\leq_\xi$ on the bound variables of a formula ξ, with $x \leq y$ indicating that y is 'more significant' than x, in the sense that the meaning of x/δ_x is (in principle) dependent on the meaning of y/δ_y. The key situation where this happens is when y occurs freely in δ_x. Observe that this can only be the case if $\delta_x \trianglelefteq \delta_y$, so that the relation '$y$ occurs freely in δ_x' does not have any cycles, and thus naturally induces a partial order.

Definition 2.20 Given a clean formula ξ, we define a *dependency* or *subordination order* $\leq_\xi$ on the set $BV(\xi)$, saying that *y ranks higher* than x if $x \leq_\xi y$. The relation $\leq_\xi$ is defined as the least partial order containing all pairs (x, y) such that $y \trianglelefteq \delta_x \trianglelefteq \delta_y$. $\lhd$

2.2 Game semantics

For a definition of the evaluation game of the modal μ-calculus, fix a *clean* formula ξ and an LTS $\mathbb{S}$. Basically, the game $\mathcal{E}(\xi, \mathbb{S})$ for ξ a fixpoint formula is defined in the same way as for plain modal logic formulas.

Definition 2.21 Given a clean modal μ-calculus formula ξ and a transition system $\mathbb{S}$, we define the *evaluation game* or *model checking game* $\mathcal{E}(\xi, \mathbb{S})$ as a board game with players $\exists$ and $\forall$ moving a token around positions of the form $(\varphi, s) \in Sf(\xi) \times S$. The rules, determining the admissible moves from a given position, together with the player who is supposed to make this move, are given in Table 3.

As before, $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ denotes the instantiation of this game where the starting position is fixed as (ξ, s). ◁

One might expect that the main difference with the evaluation game for basic modal formulas would involve the new formula constructors of the μ-calculus: the fixpoint operators. Perhaps surprisingly, we can deal with the fixpoint operators themselves in the most straightforward way possible, viz., by simply *stripping* them. That is, the successor of a position of the form $(\eta x.\delta, s)$ is simply obtained as the pair (δ, s). (In section 2.5 we present an alternative version in which the formula $\eta x\,\delta$ is replaced with its unfolding). Since this next position is thus uniquely determined, the position $(\eta x.\delta, s)$ will not be assigned to either of the players.

The crucial difference lies in the treatment of the *bound variables* of a fixpoint formula ξ. Previously, all positions of the form (p, s) would be *final positions* of the game, immediately determining the winner of the match, and this is still the case here if p is a *free* variable. However, at a position (x, s) with x *bound*, the fixpoint variable x gets *unfolded*; this means that the new position is given as (δ_x, s), where $\eta_x x.\delta_x$ is the unique subformula of ξ where x is bound. Note that for this to be well defined, we need ξ to be clean. The disjointness of $FV(\xi)$ and $BV(\xi)$ ensures that it is always clear whether a variable is to be unfolded or not, and the fact that bound variables are bound by unique occurrences of fixpoint operators guarantees that δ_x is uniquely determined. Finally, since in this case the next position is also completely determined by the current one, positions of the form (x, s) with x *bound* are assigned to neither of the players.

Position	Player	Admissible moves
$(\varphi_1 \vee \varphi_2, s)$	$\exists$	$\{(\varphi_1, s), (\varphi_2, s)\}$
$(\varphi_1 \wedge \varphi_2, s)$	$\forall$	$\{(\varphi_1, s), (\varphi_2, s)\}$
$(\Diamond_d \varphi, s)$	$\exists$	$\{(\varphi, t) \mid t \in \sigma_d(s)\}$
$(\Box_d \varphi, s)$	$\forall$	$\{(\varphi, t) \mid t \in \sigma_d(s)\}$
$(\bot, s)$	$\exists$	$\varnothing$
$(\top, s)$	$\forall$	$\varnothing$
(p, s), with $p \in FV(\xi)$ and $s \in V(p)$	$\forall$	$\varnothing$
(p, s), with $p \in FV(\xi)$ and $s \notin V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$, with $p \in FV(\xi)$ and $s \in V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$, with $p \in FV(\xi)$ and $s \notin V(p)$	$\forall$	$\varnothing$
$(\eta_x x.\delta_x, s)$	–	$\{(\delta_x, s)\}$
(x, s), with $x \in BV(\xi)$	–	$\{(\delta_x, s)\}$

Table 3: Evaluation game for modal fixpoint logic

Example 2.22 Let $\mathbb{S} = \langle S, R, V\rangle$ be the Kripke model based on the set $S = \{0, 1, 2\}$, with $R = \{(0, 1), (1, 1), (1, 2), (2, 2)\}$, and V given by $V(p) = \{2\}$. Now let ξ be the formula $\eta x.p \vee \Box x$, and consider the game $\mathcal{E}(\xi, \mathbb{S})$ initialized at $(\xi, 0)$.

The second position of any match of this game will be $(p \vee \Box x, 0)$ belonging to $\exists$. Assuming that she wants to win, she chooses the disjunct $\Box x$ since otherwise p being false at 0 would mean an immediate loss for her. Now the position $(\Box x, 0)$ belongs to $\forall$ and he will make the only move allowed to him, choosing $(x, 1)$ as the next position. Here an automatic move is made, *unfolding* the variable x, and thus changing the position to $(p \vee \Box x, 1)$. And as before, $\exists$ will choose the right disjunct: $(\Box x, 1)$.

At $(\Box x, 1)$, $\forall$ does have a choice. Choosing $(x, 2)$, however, would mean that $\exists$ wins the match since p being true at 2 enables her to finally choose the first disjunct of the formula $p \vee \Box x$. So $\forall$ chooses $(x, 1)$, a position already visited by the match before.

This means that these strategies force the match to be *infinite*, with the variable x unfolding infinitely often at positions of the form $(x, 1)$, and the match taking the following form:

$$(\xi, 0)(p \vee \Box x, 0)(\Box x, 0)(x, 1)(p \vee \Box x, 1)(\Box x, 1)(x, 1)(p \vee \Box x, 1) \ldots$$

So who is declared to be the winner of this match? This is where the difference between the two fixpoint operators shows up. In case $\eta = \mu$, the above infinite match is *lost* by $\exists$ since the fixpoint variable that is unfolded infinitely often is a μ-variable, and μ-variables are to be unfolded only finitely often. In case $\eta = \nu$, the variable unfolded infinitely often is a ν-variable, and this is unproblematic: $\exists$ wins the match. $\lhd$

The above example shows the principle of unfolding at work. Its effect is that matches may now be of infinite length since formulas are no longer deconstructed at every move of the game. Nevertheless, as we will see, it will still be very useful to declare a *winner* of such an infinite game. Here we arrive at one of the key ideas underlying the semantics of fixpoint formulas, which in a slogan can be formulated as follows:

ν means unfolding, μ means finite unfolding.

Giving a more detailed interpretation to this slogan, in case of a unique variable that is unfolded infinitely often during a match π, we will declare $\exists$ to be the winner of π if this variable is a ν-variable, and $\forall$ in case we are dealing with a μ-variable. But what happens in case that *various* variables are unfolded infinitely often? As we shall see, in these cases there is always a *unique* such variable that ranks higher than any other such variable.

Definition 2.23 Let ξ be a clean $\mu\mathtt{ML}_{\mathsf{D}}$-formula, and $\mathbb{S}$ a labelled transition system. A *match* of the game $\mathcal{E}(\xi, \mathbb{S})$ is a (finite or infinite) sequence of positions

$$\pi = (\varphi_i, s_i)_{i<\kappa}$$

(where κ is either a natural number or ω) which is in accordance with the rules of the evaluation game — that is, π is a path through the game graph given by the admissibility relation of Table 3. A *full match* is either an infinite match, or a finite match in which the player responsible for the last position got stuck. In practice we will always refer to full matches simply as *matches*. A match that is not full is called *partial*.

Given an infinite match π, we let $Unf^{\infty}(\pi) \subseteq BV(\xi)$ denote the set of variables that are unfolded infinitely often during π. $\triangleleft$

Proposition 2.24 *Let ξ be a clean $\mu\mathtt{ML}_{\mathsf{D}}$-formula, and $\mathbb{S}$ a labelled transition system. Then for any infinite match π of the game $\mathcal{E}(\xi,\mathbb{S})$, the set $Unf^{\infty}(\pi)$ has a highest ranking member, in terms of the dependency order of Definition 2.20.*

Proof. Since π is an infinite match, the set $U := Unf^{\infty}(\pi)$ is not empty. Let y be an element of U which is *maximal* (with respect to the ranking order $\leq_{\xi}$) — such an element exists since U is finite. We claim that

$$\text{from some moment on, } \pi \text{ only features subformulas of } \delta_y. \tag{9}$$

To prove this, note that since y is $\leq_{\xi}$-maximal in U, there must be a position in π such that y is unfolded to δ_y, while no variable $z >_{\xi} y$ is unfolded at any later position in π. But then a straightforward induction shows that all formulas featuring at later positions must be subformulas of δ_y: the key observation here is that if $z \trianglelefteq \delta_y$ unfolds to δ_z, and by assumption $z \not>_{\xi} y$, then it must be the case that $\delta_z \trianglelefteq \delta_y$.

As a corollary of (9), we claim that

$$y \text{ is in fact the maximum of } U \text{ (with respect to } \leq_{\xi}). \tag{10}$$

To see this, suppose for contradiction that there is a variable $x \in U$ which is *not* below y. It follows from (9) that $\delta_x \trianglelefteq \delta_y$, and without loss of generality we may assume x to be such that δ_x is a *maximal* subformula of δ_y such that $x \not\leq_{\xi} y$ (in the sense that $z \leq_{\xi} y$ for all $z \in U$ with $\delta_x \triangleleft \delta_z$.) In particular then we have $y \notin FV(\delta_x)$. But since y is unfolded infinitely often, there must be a variable $z \in FV(\delta_x)$ which allows π to 'leave' δ_x infinitely often; this means that $z \in U$, $\delta_x \trianglelefteq \delta_z$ but $\delta_z \not\trianglelefteq \delta_x$. From this it is immediate that $x \leq_{\xi} z$, while from $z \in U$ and (9) we obtain $\delta_z \trianglelefteq \delta_y$. It now follows from our maximality assumption on x that $z \leq_{\xi} y$. But then by transitivity of $\leq_{\xi}$ we find that $x \leq_{\xi} y$ indeed. In other words, we have arrived at the desired contradiction.

This shows that (10) holds indeed, and from this the Proposition is immediate. QED

Given this result, there is now a natural formulation of the winning conditions for infinite matches of evaluation games.

Definition 2.25 Let ξ be a clean $\mu\mathtt{ML}_{\mathsf{D}}$-formula. The winning conditions of the game $\mathcal{E}(\xi,\mathbb{S})$ are given in Table 4. $\triangleleft$

We can now formulate the game-theoretic semantics of the modal μ-calculus as follows.

Definition 2.26 Let ξ be a clean formula of the modal μ-calculus, and let $\mathbb{S}$ be a transition system of the appropriate type. Then we say that ξ is (game-theoretically) *satisfied* at s, notation: $\mathbb{S}, s \Vdash_g \xi$ if $(\xi, s) \in \mathrm{Win}_{\exists}(\mathcal{E}(\xi,\mathbb{S}))$. $\triangleleft$

	$\exists$ wins π	$\forall$ wins π
π is finite	$\forall$ got stuck	$\exists$ got stuck
π is infinite	$\max(\mathit{Unf}^{\infty}(\pi))$ is a ν-variable	$\max(\mathit{Unf}^{\infty}(\pi))$ is a μ-variable

Table 4: Winning conditions of $\mathcal{E}(\xi, \mathbb{S})$

Remark 2.27 As mentioned we have kept this introduction to evaluation games for fixpoint formulas rather informal, referring to Chapter 5 for a more rigorous discussion of infinite games. Nevertheless, we want to mention already here that evaluation games, on the ground of being so-called *parity games*, have two very useful properties that make them attractive to work with. To start with, every evaluation game is *determined* in the sense that every position is winning for exactly one of the two players. And second, one may show that winning strategies for either player of an evaluation game, can always be assumed to be *positional*, that is, do not depend on moves made earlier in the match, but only on the current position. Combining this, evaluation games enjoy *positional determinacy*; that is, every position (φ, s) is winning for exactly one of the two players, and each player $\Pi \in \{\exists, \forall\}$ has a *positional* strategy f_{Π} which is winning for the game $\mathcal{E}(\xi, \mathbb{S})@(\varphi, s)$ for every position (φ, s) that is winning for Π. $\lhd$

Remark 2.28 Observe that we have defined the game-theoretic semantics for *clean* formula only. In the next section we define an alternative version of the evaluation game which works for arbitrary *tidy* formulas.

It is certainly possible to extend this definition to arbitrary fixpoint formulas; a straightforward approach would be to involve the *construction tree* of a non-clean formula ξ, and redefine a *position* of the evaluation game $\mathcal{E}(\xi, \mathbb{S})$ to be a pair, consisting of a node in this construction tree and a point in the Kripke structure. Alternatively, one may work with a clean *alphabetical variant* of the formula ξ; once we have given the algebraic semantics for arbitrary formulas, it is not hard to show that in that semantics, alphabetic variants are equivalent. $\lhd$

2.3 Examples

Example 2.29 As a first example, consider the formulas $\eta x.p \vee x$, and fix a Kripke model $\mathbb{S}$. Observe that any match of the evaluation game $\mathcal{E}(\eta x.p \vee x, \mathbb{S})$ starting at position $(\eta x.p \vee x, s)$ immediately proceeds to position $(p \vee x, s)$, after which $\exists$ can make a choice. In case η is the least fixpoint operator, $\eta = \mu$, we claim that

$$\mathbb{S}, s \Vdash_g \mu x.p \vee x \text{ iff } s \in V(p).$$

For the direction from right to left, assume that $s \in V(p)$. Now, if $\exists$ chooses the disjunct p at the position $(s, p \vee x)$, she wins the match because $\forall$ will get stuck at (s, p). Hence $s \in \mathrm{Win}_{\exists}(\mathcal{E}(\mu x.p \vee x, \mathbb{S}))$.

On the other hand, if $s \notin V(p)$, then $\exists$ will lose if she chooses the disjunct p at position $(s, p \vee x)$. So she must choose the disjunct x which then unfolds to $p \vee x$ so that $\exists$ is back at the position $(s, p \vee x)$. Thus if $\exists$ does not want to get stuck, her only way to survive is to keep playing the position (s, x), thus causing the match to be infinite. But such a match is won by $\forall$ since the only variable that gets unfolded infinitely often is a μ-variable. Hence in this case we see that $s \notin \mathrm{Win}_{\exists}(\mathcal{E}(\nu x.p \vee x, \mathbb{S}))$.

If on the other hand we consider the case where $\eta = \nu$, then $\exists$ can win any match:

$$\mathbb{S}, s \Vdash_g \nu x.p \vee x.$$

It is easy to see that now, the strategy of always choosing the disjunct x at a position of the form $(s, p \vee x)$ is winning. For, it forces all games to be infinite, and since the only fixpoint variable that gets ever unfolded here is a ν-variable, all infinite matches are won by $\exists$.

Concluding, we see that $\mu x.p \vee x$ is equivalent to the formula p, and $\nu x.p \vee x$, to the formula $\top$. $\lhd$

Example 2.30 Now we turn to the formulas $\mu x.\Diamond x$ and $\nu x.\Diamond x$. First consider how a match for any of these formulas proceeds. The first two positions of such a match will be of the form $(\eta x.\Diamond x, s)(\Diamond x, s)$, at which point it is $\exists$'s turn to make a move. Now she either is stuck (in case the state s has no successor) or else the next two positions are $(x, t)(\Diamond x, t)$ for some successor t of s, chosen by $\exists$. Continuing this analysis, we see that there are two possibilities for a match of the game $\mathcal{E}(\eta x.\Diamond x, \mathbb{S})$:

1. the match is an infinite sequence of positions

$$(\eta x.\Diamond x, s_0)(\Diamond x, s_0)(x, s_1)(\Diamond x, s_1)(x, s_2)\ldots$$

corresponding to an infinite path $s_0 R s_1 R s_2 R \ldots$ through $\mathbb{S}$.

2. the match is a finite sequence of positions

$$(\eta x.\Diamond x, s_0)(\Diamond x, s_0)(x, s_1)(\Diamond x, s_1)\ldots(\Diamond x, s_k)$$

corresponding to a finite path $s_0 R s_1 R \ldots s_k$ through $\mathbb{S}$, where s_k has no successors.

Note too that in either case it is only $\exists$ who has turns, and that her strategy corresponds to choosing a *path* through $\mathbb{S}$. From this it is easy to derive that

- $\mu x.\Diamond x$ is equivalent to the formula $\bot$,
- $\mathbb{S}, s \Vdash_g \nu x.\Diamond x$ iff there is an infinite path starting at s. $\lhd$

▶ `Until operator`

The examples that we have considered so far involved only a single fixpoint operator. We now look at an example containing both a least and a greatest fixpoint operator.

Example 2.31 Let ξ be the following formula:

$$\xi = \nu x.\mu y.\underbrace{(p \wedge \Diamond x)}_{\alpha_p} \vee \underbrace{(\overline{p} \wedge \Diamond y)}_{\alpha_{\overline{p}}}$$

Then we claim that for any LTS $\mathbb{S}$, and any state s in $\mathbb{S}$:

$$\mathbb{S}, s \Vdash_g \xi \text{ iff there is some path from } s \text{ on which } p \text{ is true infinitely often.} \quad (11)$$

To see this, first suppose that there is a path $\pi = s_0 s_1 s_2 \ldots$ as described in the right hand side of (11) and suppose that $\exists$ plays according to the following strategy:

(a) at a position $(\alpha_p \vee \alpha_{\overline{p}}, t)$, choose (α_p, t) if $\mathbb{S}, t \Vdash_g p$ and choose $(\alpha_{\overline{p}}, t)$ otherwise;

(b) at a position $(\Diamond\varphi, t)$, distinguish cases:
- if the match so far has followed the path, with $t = s_k$, choose (φ, s_{k+1});
- otherwise, choose an arbitrary successor (if possible).

We claim that this is a winning strategy for $\exists$ in the evaluation game initialized at (ξ, s). Indeed, since $\exists$ always chooses the propositionally safe disjunct of $\alpha_p \vee \alpha_{\overline{p}}$, she forces $\forall$, when faced with a position of the form $(\alpha_{\pm p}, t) = (\pm p \wedge \Diamond z, t)$ to always choose the diamond conjunct $\Diamond z$, or lose immediately. In this way she guarantees to always get to positions of the form $(\Diamond z, s_i)$, and thus she can force the match to last infinitely long, following the infinite path π. But why does she actually *win* this match? The point is that, whenever she chooses α_p, three positions later, x will be unfolded, and likewise with $\alpha_{\overline{p}}$ and y. Thus, p being true infinitely often on π means that the ν-variable x gets unfolded infinitely often. And so, even though the μ-variable y might get unfolded infinitely often as well, she wins the match since x ranks higher than y anyway.

For the other direction, assume that $\mathbb{S}, s \Vdash_g \xi$ so that $\exists$ has a winning strategy in the game $\mathcal{E}(\xi, \mathbb{S})$ initialized at (ξ, s). It should be clear that any winning strategy must follow (a) above. So whenever $\forall$ faces a position $(p \wedge \Diamond z, t)$, p will be true, and likewise with positions $(\overline{p} \wedge \Diamond z, t)$. Now consider a match in which $\forall$ plays propositionally sound, that is, always chooses the diamond conjunct of these positions. This match must be infinite since both players will stay alive forever: $\forall$ because he can always choose a diamond conjunct, and $\exists$ because we assumed her strategy to be winning. But a second consequence of $\exists$ playing a winning strategy, is that it cannot happen that y is unfolded infinitely often, while x is not. So x is unfolded infinitely often, and as before, x only gets unfolded right after the match passed a world where p is true. Thus the path chosen by $\exists$ must contain infinitely many states where p holds. $\triangleleft$

2.4 Bisimulation invariance and the bounded tree model property

Given the game-theoretic characterization of the semantics, it is rather straightforward to prove that formulas of the modal μ-calculus are bisimulation invariant. From this it is immediate that the modal μ–calculus has the tree model property. But in fact, we can use the game semantics to do better than this, proving that every satisfiable modal fixpoint formula is satisfied in a tree of which the branching degree is *bounded* by the size of the formula.

Theorem 2.32 (Bisimulation Invariance) *Let ξ be a modal fixpoint formula with $FV(\xi) \subseteq \mathsf{P}$, and let $\mathbb{S}$ and $\mathbb{S}'$ be two labelled transition systems with points s and s', respectively. If $\mathbb{S}, s \leftrightarroweq_{\mathsf{P}} \mathbb{S}', s'$, then*

$$\mathbb{S}, s \Vdash_g \xi \textit{ iff } \mathbb{S}', s' \Vdash_g \xi.$$

Proof. Assume that $s \leftrightarroweq_{\mathsf{P}} s'$ and that $\mathbb{S}, s \Vdash_g \xi$, with $FV(\xi) \subseteq \mathsf{P}$. We will show that $\mathbb{S}', s' \Vdash_g \xi$. By definition we may assume that $\exists$ has a winning strategy f in the evaluation game $\mathcal{E} := \mathcal{E}(\xi, \mathbb{S})$ initialized at (ξ, s); that is, given an f-guided partial $\mathcal{E}$-match π ending in a position for $\exists$, we let $f(\pi)$ denote the next position as determined by f.

We need to provide her with a winning strategy in the game $\mathcal{E}' := \mathcal{E}(\xi, \mathbb{S}')@(\xi, s')$. She obtains her strategy f' in $\mathcal{E}'$ from playing a *shadow match* of $\mathcal{E}$, using the bisimilarity relation to guide her choices.

To see how this works, let's simply start with comparing the initial position (ξ, s') of $\mathcal{E}'$ with its counterpart (ξ, s) of $\mathcal{E}$. (From now on we will write $s \leftrightarroweq s'$ instead of $s \leftrightarroweq_{\mathsf{P}} s'$).

- In case ξ is a literal, it is easy to see that both (ξ, s) and (ξ, s') are final positions. Also, since f is assumed to be winning, ξ must be true at s, and so it must hold at s' as well. Hence, $\exists$ wins the match.

If ξ is not a literal, we distinguish cases.

- First suppose that $\xi = \xi_1 \vee \xi_2$. If f tells $\exists$ to choose disjunct ξ_i at (ξ, s), then she chooses the same disjunct ξ_i at position (ξ, s'). If $\xi = \xi_1 \wedge \xi_2$, it is $\forall$ who moves. Suppose in $\mathcal{E}'$ he chooses ξ_i, making (ξ_i, s') the next position. We now consider in $\mathcal{E}$ the same move of $\forall$, so that the next position in the shadow match is (ξ_i, s).

- A third possibility is that $\xi = \Diamond\psi$. In order to make her move at (ξ, s'), $\exists$ first looks at (ξ, s). Since f is a winning strategy, it indeed picks a successor t of s. Then because $s \leftrightarroweq s'$, there is a successor t' of s' such that $t \leftrightarroweq t'$. This t' is $\exists$'s move in $\mathcal{E}'$, so that (ψ, t) and (ψ, t') are the next positions in $\mathcal{E}$ and $\mathcal{E}'$, respectively.

- If $\xi = \Box\psi$, we are dealing again with positions for $\forall$. Suppose in $\mathcal{E}'$ he chooses the successor t' of s', so that the next position is (ψ, t'). (In case s' has no successors, $\forall$ immediately loses, so that there is nothing left to prove.) Now again we turn to the shadow match; by bisimilarity of s and s' there is a successor t of s such that $t \leftrightarroweq t'$. So we may assume that $\forall$ moves the game token of $\mathcal{E}$ to position (ψ, t).

- Finally, if $\xi = \eta x\, \delta_x$ then the next positions in $\mathcal{E}$ and $\mathcal{E}'$ are, respectively, (δ_x, s) and (δ_x, s').

The crucial observation is that if $\exists$ does not win immediately, then at least she can guarantee that the next positions in $\mathcal{E}$ and $\mathcal{E}'$ are of the form (φ, u) and (φ, u') respectively, with $u \leftrightarroweq u'$, and such that the move in $\mathcal{E}$ is consistent with f. We may in fact show that she can maintain this condition throughout the match, and it is not hard to see that she can construct a winning strategy based on this.

Making this proof sketch a bit more precise, we introduce some terminology (anticipating the formal treatment of games in Chapter 5). Generally we identify *matches* of a game with certain sequences of positions in that game, and we say that a match $\pi = p_0 p_1 \dots p_n$ is *guided* by a strategy f for player $\Pi \in \{\exists, \forall\}$ if for every $i < n$ such that position p_i belongs to Π, the next position p_{i+1} is indeed the position dictated by the strategy f. In the context of this particular proof we say that an $\mathcal{E}'$-match $\pi' = (\varphi'_0, s'_0)(\varphi'_1, s'_1) \dots (\varphi'_n, s'_n)$ is *linked* to an

$\mathcal{E}$-match $\pi = (\varphi_0, s_0)(\varphi_1, s_1)\dots(\varphi_n, s_n)$ (of the same length), if $\varphi'_i = \varphi_i$ and $\mathbb{S}', s'_i \leftrightarroweq \mathbb{S}, s_i$ for all i with $0 \leq i \leq n$. The key claim in the proof states that, for a $\mathcal{E}'$-match π', if $\exists$ has established such a bisimilarity link with an $\mathcal{E}$-match that is f-guided, then she will either win the $\mathcal{E}'$-game immediately, or else she can maintain the link during one further round of the game.

Claim 1 Let π' be a finite $\mathcal{E}'$-match, and assume that π' is linked to some f-guided $\mathcal{E}$-match π. Then one of the following two cases apply.

1) both *last*(π') and *last*(π) are positions for $\exists$, and $\exists$ can continue π' with a legitimate move (φ, t') such that $\pi' \cdot (\varphi, t')$ is bisimilarity-linked to $\pi \cdot (\varphi, t)$, where (φ, t) is the move dicated by f in π.

2) both *last*(π') and *last*(π) are positions for $\forall$, and for every move (φ', t) for $\forall$ in π' there is a legitimate move (φ, t) for $\forall$ in π such that $\pi' \cdot (\varphi, t')$ is bisimilarity-linked to $\pi \cdot (\varphi, t)$.

The *proof* of this Claim proceeds via an obvious adaptation of the case-by-case argument just given for the initial positions of $\mathcal{E}'$ and $\mathcal{E}$. Omitting the details, we move on to show that based on Claim 1, $\exists$ has a winning strategy in $\mathcal{E}'$.

By a straightforward inductive argument we may provide $\exists$ with a strategy f' in $\mathcal{E}'$, and show how to maintain, simultaneously, for every f'-guided match π, an f-guided $\mathcal{E}$-match which is linked to π'. For the base case of this induction, simply observe that by the assumption that $\mathbb{S}, s \leftrightarroweq \mathbb{S}', s'$, the initial positions of $\mathcal{E}'$ and $\mathcal{E}$ constitute linked (trivial) matches. For the inductive case we consider an f'-guided $\mathcal{E}'$-match π', and inductively assume that there is a bisimilarity-linked f-guided $\mathcal{E}$-match π. Now distinguish cases:

- If *last*(π') is a position for $\exists$, we use item 1) of Claim 1 to define her move (φ, t'); it follows that $\pi' \cdot (\varphi, t')$ and $\pi \cdot (\varphi, t)$ are bisimilarity-linked (where (φ, t) is the move dicated by f in π).
- On the other hand, in case *last*(π') is a position for $\forall$, assume that he makes some move, say, (ψ, t'); now we use item 2) of the claim to define a continuation $\pi \cdot (\psi, t)$ of π that is bisimilarity-linked to $\pi' \cdot (\psi, t')$.

To see why the strategy f' of $\exists$ is *winning* for her, consider a *full* (i.e., finished) f'-guided match π', and distinguish cases. If π' is finite, this means that one of the players must be stuck, and we have to show that this player must be $\forall$. But we just showed that there must be an f-guided match π which is bisimilarity-linked to π'. It follows from the definiton of linked matches that the final positions of π' and π must be, respectively, of the form (φ, t') and (φ, t) for some formula φ and states t', t such that $\mathbb{S}', t' \leftrightarroweq \mathbb{S}, t$. From this it is not hard to derive that the same player who got stuck in π' also got stuck in π; and since π is guided by $\exists$'s supposedly winning strategy f, this player must be $\forall$ indeed.

If π' is infinite, say $\pi' = (\varphi_i, s'_i)_{i<\omega}$, the shadow $\mathcal{E}$-match maintained by $\exists$ is infinite as well. More precisely, the inductive argument given above reveals the existence of an infinite, f-guided $\mathcal{E}$-match $\pi = (\varphi_i, s_i)_{i<\omega}$ such that $\mathbb{S}', s'_i \leftrightarroweq \mathbb{S}, s_i$ for all $i < \omega$. The key observation, however, is that the two sequences of formulas, in the $\mathcal{E}'$-match π' and its $\mathcal{E}$-shadow π, respectively, are exactly the *same*. This means that also in the infinite case the winner of π' is the winner of π, and since π is f-guided, this winner must be $\exists$. QED

As an immediate corollary, we obtain the tree model property for the modal μ-calculus.

Theorem 2.33 (Tree Model Property) *Let ξ be a modal fixpoint formula. If ξ is satisfiable, then it is satisfiable at the root of a tree model.*

Proof. For simplicity, we confine ourselves to the basic modal language. Suppose that ξ is satisfiable at state s of the Kripke model $\mathbb{S}$. Then by bisimulation invariance, ξ is satisfiable at the root of the *unravelling* $\vec{\mathbb{S}}_s$ of $\mathbb{S}$ around s, cf. Definition 1.23. This unravelling clearly is a tree model. QED

For the next theorem, recall that the size of a formula is simply defined as the number of its subformulas.

Theorem 2.34 (Bounded Tree Model Property) *Let ξ be a modal fixpoint formula. If ξ is satisfiable, then it is satisfiable at the root of a tree, of which the branching degree is bounded by the size $|\xi|$ of the formula.*

Proof. Suppose that ξ is satisfiable. By the Bisimulation Invariance Theorem it follows that ξ is satisfiable at the root r of some tree model $\mathbb{T} = \langle T, R, V\rangle$. So $\exists$ has a winning strategy f in the game $\mathcal{E}@(\xi, r)$, where we abbreviate $\mathcal{E} := \mathcal{E}(\xi, \mathbb{T})$. By the Positional Determinacy of the evaluation game, we may assume that this strategy is positional — this will simplify our argument a bit. We may thus represent this strategy as a map f that, among other things, maps positions of the form $(\Diamond\varphi, s)$ to positions of the form (φ, t) with Rst.

We will prune the tree $\mathbb{T}$, keeping only the nodes that $\exists$ needs in order to win the match. Formally, define subsets $(T_n)_{n\in\omega}$ as follows:

$$\begin{array}{lll} T_0 & := & \{r\}, \\ T_{n+1} & := & T_n \cup \{s \mid (\varphi, s) = f(\Diamond\varphi, t) \text{ for some } t \in T_n \text{ and } \Diamond\varphi \trianglelefteq \xi\}, \\ T_\omega & := & \bigcup_{n\in\omega} T_n. \end{array}$$

Let $\mathbb{T}_\omega$ be the subtree of $\mathbb{T}$ based on T_ω. (Note that $\mathbb{T}_\omega$ is in general not a generated submodel of $\mathbb{T}$: not all successors of nodes in $\mathbb{T}_\omega$ need to belong to $\mathbb{T}_\omega$.) From the construction it is obvious that the branching degree of $\mathbb{T}_\omega$ is bounded by the size of ξ, because ξ has at most $|\xi|$ diamond subformulas.

We claim that $\mathbb{T}_\omega, r \Vdash_g \xi$. To see why this is so, let $\mathcal{E}' := \mathcal{E}(\xi, \mathbb{T}_\omega)$ be the evaluation game played on the pruned tree. It suffices to show that the strategy f', defined as the restriction of f to positions of the game $\mathcal{E}'$, is winning for $\exists$ in the game starting at (ξ, r). Consider an arbitrary $\mathcal{E}'$-match $\pi = (\xi, r)(\varphi_1, t_1)\ldots$ which is consistent with f'. The key observation of the proof is that π is also a match of $\mathcal{E}@(\xi, r)$, that is consistent with f. To see this, simply observe that all moves of $\forall$ in π could have been made in the game on $\mathbb{T}$ as well, whereas by construction, all f' moves of $\exists$ in $\mathcal{E}'$ are f moves in $\mathcal{E}$.

Now by assumption, f is a winning strategy for $\exists$ in $\mathcal{E}$, so she wins π in $\mathcal{E}$. But then π is winning as such, i.e., no matter whether we see it as a match in $\mathcal{E}$ or in $\mathcal{E}'$. In other words, π is also winning as an $\mathcal{E}'$-match. And since π was an arbitrary $\mathcal{E}'$-match starting at (ξ, r), this shows that f' is a winning strategy, as required. QED

2.5 Traces, the closure map and the closure game

In this section we define an alternative version of the evaluation game for μ-calculus formulas, in which the equivalence

$$\eta x\, \chi \equiv \chi[\eta x\, \chi/x]$$

is exploited more directly than in the *subformula game* that we defined in section 2.2. The idea in the *closure game* is that, at a position $(\eta x\, \chi, s)$ the fixpoint formula will simply be unfolded, yielding the pair $(\chi[\eta x\, \chi/x], s)$ as the (unique) next position. That is, the admissible moves in the closure game are given in Table 5.

Position	Player	Admissible moves
$(\varphi \vee \psi, s)$	$\exists$	$\{(\varphi, s), (\psi, s)\}$
$(\varphi \wedge \psi, s)$	$\forall$	$\{(\varphi, s), (\psi, s)\}$
$(\Diamond\varphi, s)$	$\exists$	$\{(\varphi, t) \mid sRt\}$
$(\Box\varphi, s)$	$\forall$	$\{(\varphi, t) \mid sRt\}$
(p, s) with $p \in FV(\xi)$ and $s \in V(p)$	$\forall$	$\varnothing$
(p, s) with $p \in FV(\xi)$ and $s \notin V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$ with $p \in FV(\xi)$ and $s \in V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$ with $p \in FV(\xi)$ and $s \notin V(p)$	$\forall$	$\varnothing$
$(\eta x.\varphi, s)$	-	$\{(\varphi[\eta x\, \varphi/x], s)\}$

Table 5: Positions and admissible moves in the closure evaluation game $\mathcal{E}^c(\xi, \mathbb{S})$

In order to turn this table into a proper game, we need to introduce appropriate *winning conditions* for the two players. For this purpose we introduce some terminology and notation, and we make some observations. We start with the notion of a *trace*.

Traces and the closure game

Definition 2.35 Let $\to_C$ be the binary relation between tidy μ-calculus formulas given by the following exhaustive list:

1) $(\varphi_0 \circledast \varphi_1) \to_C \varphi_i$, for any $\varphi_0, \varphi_1 \in \mu\mathtt{ML}$, $\circledast \in \{\wedge, \vee\}$ and $i \in \{0, 1\}$;
2) $\heartsuit\varphi \to_C \varphi$, for any $\varphi \in \mu\mathtt{ML}$ and $\heartsuit \in \{\Diamond, \Box\}$);
3) $\eta x.\varphi \to_C \varphi[\eta x.\varphi/x]$, for any $\eta x.\varphi \in \mu\mathtt{ML}$, with $\eta \in \{\mu, \nu\}$.

We call a $\to_C$-path $\psi_0 \to_C \psi_1 \to_C \cdots \to_C \psi_n$ a *(finite) trace*; similarly, an *infinite trace* is a sequence $(\psi_i)_{i<\omega}$ such that $\psi_i \to_C \psi_{i+1}$ for all $i < \omega$. ◁

Intuitively a trace is a sequence that corresponds to the formula part of a possible match of the closure game. The *closure* of a formula consists of the formulas that can be encountered in such a match.

Definition 2.36 We define the relation $\twoheadrightarrow_C$ as the reflexive and transitive closure of $\to_C$, and define the *closure* of a tidy formula ψ as the set

$$Cl(\psi) := \{\varphi \mid \psi \twoheadrightarrow_C \varphi\}.$$

Given a set of formulas Ψ, we put $Cl(\Psi) := \bigcup_{\psi\in\Psi} Cl(\psi)$, and we call Ψ *closed* if $\Psi = Cl(\Psi)$. Formulas in the set $Cl(\psi)$ are said to be *derived* from ψ. The *closure graph* of ψ is the directed graph $\mathbb{C}_\xi := (Cl(\xi), \to_C)$. $\lhd$

In words, $Cl(\xi)$ is the smallest set which contains ξ and is closed under direct boolean and modal subformulas, and under unfoldings of fixpoint formulas. In terms of traces: a formula χ belongs to the closure of a formula ξ iff there is a trace from ξ to χ. Furthermore, a trace starting at ξ is nothing but a path in the closure graph starting at ξ.

Remark 2.37 The final example of Remark 2.18 shows that the closure of a non-tidy formula may not even be defined — unless we work with alphabetical variants. We will come back to this point later. $\lhd$

The following example will be instructive for understanding the concept of closure, and its relation with subformulas.

Example 2.38 Consider the following formulas:

$$\begin{array}{lcl}
\xi_1 & := & \mu x_1 \nu x_2 \mu x_3.\big(((x_1 \lor x_2) \lor x_3) \land \Box((x_1 \lor x_2) \lor x_3)\big) \\
\xi_2 & := & \nu x_2 \mu x_3.\big(((\xi_1 \lor x_2) \lor x_3) \land \Box((\xi_1 \lor x_2) \lor x_3)\big) \\
\xi_3 & := & \mu x_3.\big(((\xi_1 \lor \xi_2) \lor x_3) \land \Box((\xi_1 \lor \xi_2) \lor x_3)\big) \\
\xi_4 & := & \big(((\xi_1 \lor \xi_2) \lor \xi_3) \land \Box((\xi_1 \lor \xi_2) \lor \xi_3)\big) \\
\alpha & := & (\xi_1 \lor \xi_2) \lor \xi_3 \\
\beta & := & \xi_1 \lor \xi_2,
\end{array}$$

and let Φ be the set $\Phi := \{\xi_1, \xi_2, \xi_3, \xi_4, \Box\alpha, \alpha, \beta\}$.

For $i = 1, 2, 3$, the formula ξ_{i+1} is the unfolding of the formula ξ_i. Thus we find $Cl(\xi_1) = \Phi$; in fact, we have $Cl(\varphi) = \Phi$ for every formula $\varphi \in \Phi$. In Figure 2 we depict the *closure graph* of ξ_1.

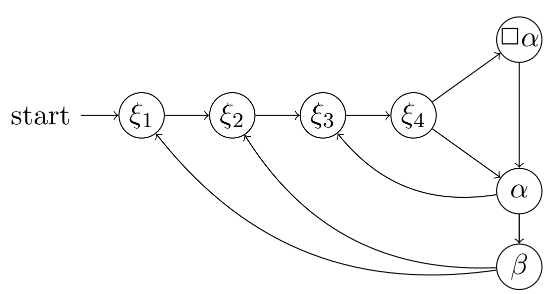


Figure 2: A closure graph

Observe that the formulas ξ_1, ξ_2, ξ_3 and ξ_4 are equivalent to one another, and hence also to α. Note too that the formula ξ_1 is the only clean formula in Φ, and that it is a subformula of *every* formula in $Cl(\xi_1)$. $\lhd$

The closure of ξ consists of the formulas that one may encounter in a match of the closure game $\mathcal{E}^c(\xi, \mathbb{S})$, and, as a consequence of this, we will take $Cl(\xi) \times S$ as the set of positions in this game. As we will see now, the key observation for defining the winning conditions of this game is that every infinite trace can be identified as either a μ-trace or a ν-trace. This is in some sense the analogon of Proposition 2.24.

Proposition 2.39 *1) Let τ be a finite trace. Then there is a unique formula on τ which is a subformula of every formula on τ.*

2) Let τ be a infinite trace. Then there is a unique formula which appears infinitely often on τ, and is a subformula of cofinitely many formulas on τ. This formula is always a fixpoint formula.

Proof.

▶ `Proof to be supplied.`

QED

Definition 2.40 Let τ be an infinite trace. The formula $\eta x\, \varphi$ which appear infinitely often on τ and is a subformula of all formulas on τ is called the *most significant formula* of τ, notation: $\mathtt{msf}(\tau)$. Depending on the nature of η we call τ either a *μ-trace* or a *ν-trace.* $\lhd$

This concept enables us to complete the definition of the closure game.

Definition 2.41 Let $\mathbb{S} = (S, R, V)$ be a Kripke model and let ξ be a tidy formula in $\mu\mathtt{ML}$. We define the *evaluation game* $\mathcal{E}^c(\xi, \mathbb{S})$ as the game (G, E, Ω) of which the board consists of the set $Cl(\xi) \times S$, and the game graph (i.e., the partitioning of $Cl(\xi) \times S$ into positions for the two players, together with the set $E(z)$ of admissible moves at each position), is given in Table 5.

The winner of an infinite match $\pi = (\xi_n, s_n)_{n<\omega}$ is $\exists$ if its left projection $\pi_L := (\xi_n)_{n<\omega}$ is a ν-trace, and $\forall$ if it is a ν-trace. $\lhd$

The closure operation

The closure operation is one of the most fundamental tools in the theory of the modal μ-calculus, and in this subsection we discuss some of its properties, the most important being Proposition 2.45 stating that the closure of a finite set is always finite.

We first gather some basic observations. To start with, while Example 2.38 clearly shows that the unfolding of a clean formula will generally not be clean, tidyness is preserved.

Proposition 2.42 *Let $\xi \in \mu\mathtt{ML}$ be a tidy formula, and let φ be derived from ξ. Then*

1) $BV(\varphi) \subseteq BV(\xi)$ and $FV(\varphi) \subseteq FV(\xi)$;

2) φ is tidy;

3) if ψ is free for x in ξ then it is also free for x in φ.

Proof. The proofs of the first two items proceed by a straightforward induction on the trace $\xi \twoheadrightarrow_C \varphi$. For instance, for the preservation of tidyness it suffices to prove that χ is tidy if $\heartsuit\chi$ is so (where $\heartsuit \in \{\Diamond, \Box\}$), that χ_0 and χ_1 are tidy if $\chi_0 \circledast \chi_1$ is so (where $\circledast \in \{\wedge, \vee\}$), and that the unfolding of a tidy formula is tidy again. The proofs of the first two claims are easy, and the third claim was stated in Proposition 2.19.

▶ `For part` (3)) `...`

QED

Second, the following proposition states that *Cl* is indeed a closure operation. We leave the proof of this proposition as an exercise for the reader.

Proposition 2.43 *Cl is a closure operation on the collection of tidy formulas:*

1) $\Phi \subseteq Cl(\Phi)$*;*
2) Cl is monotone: $\Phi \subseteq \Psi$ *implies* $Cl(\Phi) \subseteq Cl(\Psi)$*;*
3) $Cl(Cl(\Phi)) \subseteq Cl(\Phi)$*.*

The proposition below will prove to be very useful. It details how the closure map interacts with various connectives and formula constructors of the μ-calculus.

Proposition 2.44 *Let ξ be a tidy formula. Then the following hold.*

1) Let $\ell \trianglelefteq \xi$ be a literal occurring in ξ, and assume that $\ell \notin BV(\xi)$. Then $\ell \in Cl(\xi)$.
2) If $\xi = \heartsuit\chi$, then χ is tidy and $Cl(\xi) = \{\heartsuit\chi\} \cup Cl(\chi)$, where $\heartsuit \in \{\Diamond, \Box\}$.
3) If $\xi = \chi_0 \circledast \chi_1$ then both χ_i are tidy and $Cl(\xi) = \{\chi_0 \circledast \chi_1\} \cup Cl(\chi_0) \cup Cl(\chi_1)$, where $\circledast \in \{\wedge, \vee\}$.
4) If $\xi = \chi[\psi/x]$, χ is tidy, $x \in FV(\chi)$ and ψ is free for x in χ, then ψ is tidy and

$$Cl(\xi) = \{\varphi[\psi/x] \mid \varphi \in Cl(\chi)\} \cup Cl(\psi).$$

5) Let $\xi = \eta x.\chi$, where $\eta \in \{\mu, \nu\}$; assume that $x \in FV(\chi)$, and let x^* be some fresh variable. Then $\chi[x^*/x]$ is tidy and

$$Cl(\xi) = \{\varphi[\eta x.\chi/x^*] \mid \varphi \in Cl(\chi[x^*/x])\}. \tag{12}$$

Before we turn to the proof of Proposition 2.44, we briefly comment on the formulation of part 5). Note that if ξ is of the form $\xi = \eta x\, \chi$, then χ is not necessarily tidy, so that $Cl(\chi)$ may not be defined. For this reason we use a fresh propositional variable x^*. However, in case χ is tidy, (12) simplifies to

$$Cl(\xi) = \{\varphi[\eta x.\chi/x] \mid \varphi \in Cl(\chi)\}. \tag{13}$$

Proof. We prove the first and fourth claim of the proposition, leaving the other parts to the reader. The second and third claim are easy to prove, and part 5) is a fairly direct consequence of part 4).

For the first item, define the *height* of ℓ in ξ as the length of the shortest chain of the form $\varphi_0 \lhd_0 \varphi_1 \lhd_0 \cdots \lhd_0 \varphi_n$ such that $\varphi_0 = \ell$, $\varphi_n = \xi$, and, in case ℓ is a propositional variable p, no formula φ_i is of the form $\eta p\,\psi$. It is then straightforward to prove that $\ell \in Cl(\xi)$ by induction on the height of ℓ in ξ. We leave the details for the reader.

For the proof of 4), assume that $x \in FV(\chi)$ and that ψ is free for x in χ. By Proposition 2.42(3), the formula ψ is free for x in every $\varphi \in Cl(\chi)$. To prove the inclusion $\subseteq$ it suffices to show that the set $\{\varphi[\psi/x] \mid \varphi \in Cl(\chi)\} \cup Cl(\psi)\}$ has the required closure properties. This is easily verified, and so we omit the details.

For the opposite inclusion, we first show that

$$\varphi[\psi/x] \in Cl(\chi[\psi/x]), \text{ for all } \varphi \in Cl(\chi), \tag{14}$$

and we prove this by induction on the trace from ξ to φ. It is immediate by the definitions that $\chi[\psi/x] \in Cl(\chi[\psi/x])$, which takes care of the base case of this induction.

In the inductive step we distinguish three cases. First, assume that $\varphi \in Cl(\chi)$ because the formula $\heartsuit\varphi \in Cl(\chi)$, with $\heartsuit \in \{\Diamond, \Box\}$. Then by the inductive hypothesis we find $\heartsuit\varphi[\psi/x] = (\heartsuit\varphi)[\psi/x] \in Cl(\chi[\psi/x])$; but then we may immediately conclude that $\varphi[\psi/x] \in Cl(\chi[\psi/x])$ as well. The second case, where we assume that $\varphi \in Cl(\chi)$ because there is some formula $\varphi \circledast \varphi'$ or $\varphi' \circledast \varphi$ in $Cl(\chi)$ (with $\circledast \in \{\wedge, \vee\}$), is dealt with in a similar way.

In the third case, we assume that $\varphi \in Cl(\chi)$ is of the form $\varphi = \rho[\lambda y.\rho/y]$, with $\lambda \in \{\mu, \nu\}$ and $\lambda y.\rho \in Cl(\chi)$. Then inductively we may assume that $(\lambda y.\rho)[\psi/x] \in Cl(\chi[\psi/x])$. Now we make a case distinction: if $x = y$ we find that $(\lambda y.\rho)[\psi/x] = \lambda y.\rho$, while at the same time we have $\varphi[\psi/x] = \rho[\lambda y.\rho/y][\psi/x] = \rho[\lambda y.\rho/y]$, so that it follows by the closure properties that $\varphi[\psi/x] \in Cl(\chi[\psi/x])$ indeed. If, on the other hand, x and y are distinct variables, then we find $(\lambda y.\rho)[\psi/x] = \lambda y.\rho[\psi/x]$, and so it follows by the closure properties that the formula $(\rho[\psi/x])\big[\lambda y.\rho[\psi/x]/y\big]$ belongs to $Cl(\chi[\psi/x])$. But since ψ is free for x in χ, the variable y is not free in ψ, and so a straightforward calculation shows that $(\rho[\psi/x])\big[\lambda y.\rho[\psi/x]/y\big] = \rho[\lambda y.\rho/y][\psi/x] = \varphi[\psi/x]$, and so we find that $\varphi[\psi/x] \in Cl(\chi[\psi/x])$ in this case as well. This proves (14).

To see why this implies part 4) of the proposition, it remains to show that $Cl(\psi) \subseteq Cl(\xi)$. But from $x \in FV(\chi)$ we infer $x \in Cl(\chi)$ by part 1), and from this we obtain that $\psi = x[\psi/x] \in Cl(\xi)$. This suffices by Proposition 2.43. QED

As an almost immediate corollary of Proposition 2.54 we find that the closure set of a μ-calculus formula is always *finite*.

Proposition 2.45 *Let $\xi \in \mu\mathtt{ML}$ be some formula. Then the set $Cl(\xi)$ is finite.*

Proof. We prove the proposition by induction on the *length* of a formula, as defined in Definition 2.5. More precisely, we claim that

$$|Cl(\xi)| \leq |\xi|^{\ell} \tag{15}$$

for every tidy formula $\xi \in \mu\mathtt{ML}$.

In case ξ is a formula of length 1 it must be atomic, so (15) is obvious. For the inductive case we consider a formula ξ with $|\xi|^{\ell} > 1$; such a formula cannot be atomic, and so it must be a boolean, modal or fixpoint formula. We now make a case distinction, only considering the cases where ξ is a conjunction or a μ-formula.

First let ξ be of the form $\xi = \xi_0 \wedge \xi_1$. By Proposition 2.44(3) we obtain $|Cl(\xi)| \leq |Cl(\xi_0)| + |Cl(\xi_1)|$, and the induction hypothesis yields $|Cl(\xi_i)| \leq |\xi_i|^\ell$. Thus we find $|Cl(\xi)| \leq |\xi_0|^\ell + |\xi_1|^\ell < |\xi|^\ell$.

If ξ is of the form $\xi = \mu x\, \chi$ we further distinguish cases. If x is not free in χ we have $\chi[\xi/x] = \chi$ and so $Cl(\xi) = \{\xi\} \cup Cl(\chi)$. Thus, using the induction hypothesis on χ, we obtain $|Cl(\xi)| \leq 1 + |Cl(\chi)| \leq 1 + |\chi|^\ell = |\xi|^\ell$, as required. On the other hand, if x does occur freely in χ, by Proposition 2.44(5) we find $|Cl(\xi)| \leq |Cl(\chi[x^*/x])|$. But since $\chi[x^*/x]$ has the same length as χ we may use the induction hypothesis for it; this gives $|Cl(\chi[x^*/x])| \leq |\chi[x^*/x]|^\ell = |\chi|^\ell$. Combining these observations we find that $|Cl(\xi)| \leq |\chi|^\ell = |\xi|^\ell - 1$ which obviously suffices to prove (15). QED

Remark 2.46 Note that we can give a much sharper upper bound to the size of a formula's closure set than (15) which bounds this size by the length of the formula. In fact, we will see further on that the number of formulas that can be derived from a formula may be *exponentially smaller* than its number of subformulas, and that the first number is a more suitable size measure than the latter. $\lhd$

2.6 Basic syntax: continued

```
▶ In this section we discuss some further basic syntactic concepts
  o size
  o alternation depth
  o guardedness
  o free subformulas
  o expansion map
```

The size of a formula

Turning to computational aspects of the modal μ-calculus, we will see that two measures of a formulas feature prominently when we are interested in the complexity of algorithms for, e.g., model checking of a formula on a model, or satisfiability checking of a formula: its size and its alternation depth. Both notions are in fact quite subtle in that they admit several non-equivalent definitions.

When it comes to *size*, there are at least three definitions that look reasonable, at first sight: in principle one could define the size of a formula as its length, its *subformula-size*, or its *closure-size*.

```
▶ For reasons that will be discussed later on, we opt for the third option: closure
  size.
```

Definition 2.47 The *size* $|\xi|$ of a tidy formula ξ is given by

$$|\xi| := |Cl(\xi)|,$$

i.e., it is defined as the number of formulas that are derived from ξ.

The *subformula-size* of a clean formula ξ is defined as follows:

$$|\xi|^d := |Sf(\xi)|,$$

i.e., $|\xi|^d$ is given as the number of subformulas of ξ. ◁

- ▶ `Discuss the various options.`
- ▶ `Each definition corresponds to a certain way of` *`representing`* `a formula as a graph-based structure: the length of a formula corresponds to the number of nodes in its syntax tree, its subformula-size to the number of ondes in its subformula dag, and its closure-size to the size of its closure graph.`
- ▶ `Note that while the notion of` *`length`* `applies to all formulas, this is different for the other two measures.`
- ▶ `It is well-known that the subformula-size of a formula can be exponentially smaller than its length, further on we will see that, perhaps counterintuitively, the closure-size of a formula can be exponentially smaller than its subformula size.`

Alternation

- ▶ `For the time being alternation is covered in a separate section.`

Guardedness

We finish our sequence of basic syntactic definitions with the notion of guardedness, which will become important later on.

Definition 2.48 A variable x is *guarded* in a $\mu\mathtt{ML}_\mathsf{D}$-formula φ if every occurrence of x in φ is in the scope of a modal operator. A formula $\xi \in \mu\mathtt{ML}_\mathsf{D}$ is *guarded* if for every subformula of ξ of the form $\eta x.\delta$, x is guarded in δ. ◁

In the next chapter we will prove that every formula can be effectively rewritten into an equivalent, clean and guarded formula.

Free subformulas

We now have a closer look at the relation between the sets $Sf(\xi)$ and $Cl(\xi)$. Our first observation concerns the question, which subformulas of a formula also belong to its closure. This brings us to the notion of a *free* subformula.

Definition 2.49 Let φ and ψ be μ-calculus formulas. We say that φ is a *free* subformula of ψ, notation: $\varphi \trianglelefteq_f \psi$, if $\psi = \psi'[\varphi/x]$ for some formula ψ' such that $x \in FV(\psi')$ and φ is free for x in ψ'. ◁

Note that in particular all literals occurring in ψ are free subformulas of ψ. The following characterisation is useful. Recall that we write $\varphi \twoheadrightarrow_C \psi$ if $\psi \in Cl(\varphi)$, or equivalently, if there is a trace (possibly of length zero) from φ to ψ.

Proposition 2.50 *Let φ and ψ be μ-calculus formulas. If ψ is tidy, then the following are equivalent:*

1) $\varphi \trianglelefteq_f \psi$;
2) $\varphi \trianglelefteq \psi$ and $FV(\varphi) \cap BV(\psi) = \varnothing$;
3) $\varphi \trianglelefteq \psi$ and $\psi \twoheadrightarrow_C \varphi$.

Proof. We will prove the equivalence of the statements 1) - 3) to a fourth statement, viz.:
4) there is a $\triangleleft_0$-chain $\varphi = \chi_0 \triangleleft_0 \chi_1 \triangleleft_0 \cdots \triangleleft_0 \chi_n = \psi$, such that no χ_i has the form $\chi_i = \eta y.\rho_i$ with $y \in FV(\varphi)$.

For the implication 1) $\Rightarrow$ 4), assume that $\varphi \trianglelefteq_f \psi$, then by definition ψ is of the form $\psi'[\varphi/x]$ where $x \in FV(\psi')$ and φ is free for x in ψ'. But if $x \in FV(\psi)$, then it is easy to see that there is a $\triangleleft_0$-chain $x = \chi'_0 \triangleleft_0 \chi'_1 \triangleleft_0 \cdots \triangleleft_0 \chi'_n = \psi'$ such that no χ'_i is of the form $\chi'_i = \langle x.\rho'$. Assume for contradiction that one of the formulas χ'_i is of the form $\chi_i = \eta y.\rho_i$ where $y \in FV(\varphi)$. Since φ is free for x in ψ' this would mean that there is a formula of the form $\langle x.\chi$ with $\eta y.\rho_i \trianglelefteq \langle x.\chi \trianglelefteq \psi'$. However, the only candidates for this would be the formulas χ'_j with $j > i$, and we just saw that these are not of the shape $\langle x.\rho'$. This provides the desired contradiction.

For the opposite implication 4) $\Rightarrow$ 1), assume that there is a $\triangleleft_0$-chain $\varphi = \chi_0 \triangleleft_0 \chi_1 \triangleleft_0 \cdots \triangleleft_0 \chi_n = \psi$ as in the formulation of 4). One may then show by a straightforward induction that $\varphi \trianglelefteq_f \chi_i$, for all $i \geq 0$.

For the implication 2) $\Rightarrow$ 4), assume that $\varphi \trianglelefteq \psi$ and $FV(\varphi) \cap BV(\psi) = \varnothing$. It follows from $\varphi \trianglelefteq \psi$ that there is a $\triangleleft_0$-chain $\varphi = \chi_0 \triangleleft_0 \chi_1 \triangleleft_0 \cdots \triangleleft_0 \chi_n = \psi$. Now suppose for contradiction that one of the formulas χ_i would be of the form $\chi_i = \eta y.\rho_i$ with $y \in FV(\varphi)$. Then we would find $y \in FV(\varphi) \cap BV(\psi)$, contradicting the assumption that $FV(\varphi) \cap BV(\psi) = \varnothing$.

In order to prove the implication 4) $\Rightarrow$ 3), it suffices to show, for any n, that if $(\chi_i)_{0 \leq i \leq n}$ is an $\triangleleft_0$-chain of length $n + 1$ such that no χ_i has the form $\chi_i = \eta y.\rho_i$ with $y \in FV(\chi_0)$, then $\chi_n \twoheadrightarrow_C \chi_0$. We will prove this claim by induction on n. Clearly the case where $n = 0$ is trivial.

For the inductive step we consider a chain

$$\chi_0 \triangleleft_0 \chi_1 \triangleleft_0 \cdots \triangleleft_0 \chi_n \triangleleft_0 \chi_{n+1}$$

such that no χ_i has the form $\chi_i = \eta y.\rho_i$ with $y \in FV(\chi_0)$, and we make a case distinction as to the nature of χ_{n+1}. Clearly χ_{n+1} cannot be an atomic formula.

If χ_{n+1} is of the form $\rho_0 \circledast \rho_1$ with $\circledast \in \{\wedge, \vee\}$, then since $\chi_n \triangleleft_0 \chi_{n+1}$, the first formula must be of the form $\chi_n = \rho_i$ with $i \in \{0, 1\}$. But since it follows by the induction hypothesis that $\chi_n \twoheadrightarrow_C \chi_0$, we obtain from $\chi_{n+1} \rightarrow_C \chi_n$ that $\chi_{n+1} \twoheadrightarrow_C \chi_0$ as required. The case where χ_{n+1} is of the form $\heartsuit\rho$ with $\heartsuit \in \{\Diamond, \Box\}$ is handled similarly.

This leaves the case where $\chi_{n+1} = \lambda y.\rho$ is a fixpoint formula. Then since $\chi_n \triangleleft_0 \chi_{n+1}$ it must be the case that $\chi_n = \rho$. Furthermore, it follows from the assumption in 4) that $y \notin FV(\chi_0)$. From this it is not so hard to see that

$$\chi_0 \triangleleft_0 \chi_1[\chi_{n+1}/y] \triangleleft_0 \cdots \triangleleft_0 \chi_n[\chi_{n+1}/y]$$

is a $\lhd_0$-chain to which the induction hypothesis applies. It follows that $\chi_n[\chi_{n+1}/y] \twoheadrightarrow_C \chi_0$. From this and the observation that $\chi_{n+1} \rightarrow_C \chi_n[\chi_{n+1}/y]$ we find that $\chi_{n+1} \twoheadrightarrow_C \chi_0$ indeed. This finishes the proof of the implication 4) $\Rightarrow$ 3).

Finally, it follows from Proposition 2.42(1) that $\psi \twoheadrightarrow_C \varphi$ implies $FV(\varphi) \cap BV(\psi) \subseteq FV(\psi) \cap FV(\psi) = \varnothing$. From this the implication 3) $\Rightarrow$ 2) is immediate. QED

As a nice application of the notion of a *free subformula*, the following proposition states that under some mild conditions, the substitution operation $[\xi/x]$ is in fact injective. We leave the proof of this proposition as an exercise to the reader.

Proposition 2.51 *Let φ_0, φ_1 and ξ be formulas such that ξ is free for x in both φ_0 and φ_1, and not a free subformula of either φ_i. Then*

$$\varphi_0[\xi/x] = \varphi_1[\xi/x] \textit{ implies } \varphi_0 = \varphi_1. \tag{16}$$

The expansion map

The most important observation here concerns the existence of a surjective map from $Sf(\xi)$ to $Cl(\xi)$, at least for a clean formula ξ. Recall that, given a clean formula ξ, we define the *dependency order* $<_\xi$ on the bound variables of ξ as the least strict partial order such that $x <_\xi y$ if $\delta_x \lhd \delta_y$ and $y \trianglelefteq \delta_x$.

Definition 2.52 Writing $BV(\xi) = \{x_1, \ldots, x_n\}$, where we may assume that $i < j$ if $x_i <_\xi x_j$, we define the *expansion* $\exp_\xi(\varphi)$ of a subformula φ of ξ as:

$$\exp_\xi(\varphi) := \varphi[\eta_{x_1} x_1.\delta_{x_1}/x_1] \ldots [\eta_{x_n} x_n.\delta_{x_n}/x_n].$$

That is, we substitute first x_1 by $\eta_{x_1} x_1.\delta_{x_1}$ in φ; in the resulting formula, we substitute x_2 by $\eta_{x_2} x_2.\delta_{x_2}$, etc. If no confusion is likely we write $\exp(\varphi)$ instead of $\exp_\xi(\varphi)$. A proposition letter p is *active in* φ if p occurs in δ_y for some $y >_\xi x$, or equivalently, if p occurs in $\exp_\xi(\varphi)$. $\lhd$

Without proof we mention the following result.

Proposition 2.53 *Let $\xi \in \mu\mathtt{ML}$ be a clean formula and $\mathbb{S}$ a pointed Kripke structure. Then for all subformulas $\varphi \trianglelefteq \xi$ and all states s in $\mathbb{S}$ we have*

$$(\varphi, s) \in \mathrm{Win}_\exists(\mathcal{E}(\xi, \mathbb{S})) \textit{ iff } \mathbb{S}, s \Vdash_g \exp_\xi(\varphi).$$

The proposition below states that, for a clean formula xi, the expansiom map is a surjection from its set of subformulas of ξ to its closure. As an immediate corollary we obtain that the size of $Cl(\xi)$ is *bounded* by that of $Sf(\xi)$.

Proposition 2.54 *Let ξ be a clean $\mu\mathtt{ML}$-formula. Then*

$$Cl(\xi) = \{\exp_\xi(\varphi) \mid \varphi \trianglelefteq \xi\}. \tag{17}$$

Proof. For the time being we confine ourselves to a brief sketch. For the inclusion $\subseteq$ it suffices to show that the set $\{\exp_\xi(\varphi) \mid \varphi \trianglelefteq \xi\}$ has the relevant closure properties. This is a fairly routine proof. For the opposite inclusion it suffices to prove that $\exp_\xi(\varphi) \in Cl(\xi)$, for every $\varphi \in Sf(\xi)$, which can be done by a straightforward induction. QED

2.7 Alternation depth

After size, the most important complexity measure of modal μ-calculus formulas concerns the degree of nesting of least- and greatest fixpoint operators in the syntax tree (or dag) of the formula. Intuitively, the *alternation depth* of a formula ξ will be defined as the length of a maximal chain of nested, alternating fixpoint operators. As in the case of size, there is more than one reasonable way to make this intuition precise

As a first example, consider the formula

$$\xi_1 = \mu x.(\nu y.p \wedge \Box y) \vee \Diamond x,$$

expressing the reachability of some state from which only p-states will be reachable. Clearly this formula features a ν-operator in the scope of a μ-operator, and in the most straightforward approach one might indeed take this as nesting, and define the (simple) alternation depth of the formula ξ_1 as 2. However, a closer inspection of the formula ξ_1 reveals that, since the variable x does not occur in the subformula $\nu y.p \wedge \Box y$, the latter subformula does not really depend on x. This is different in the following example:

$$\xi_2 = \nu x.\mu y.(p \wedge \Diamond x) \vee \Diamond y,$$

stating the existence of a path on which p is true infinitely often. Here the variable x does occur in the subformula $\mu y.(p \wedge \Diamond x) \vee \Diamond y$; that is, ξ_2 contains a 'real' ν/μ-chain of fixpoint operators. In the definition of alternation depth ad that we shall adopt, we will see that $ad(\xi_2) = 2$ but $ad(\xi_1) = 1$.

The formal definition of alternation depth involves inductively defined formula collections Θ^η_n, where $\eta \in \{\mu, \nu\}$ and n is a natural number. Intuitively, the class Θ^η_n consists of those μ-calculus formulas where n bounds the length of any alternating nesting of fixpoint operators of which the most significant formula is an η-formula. We will make this intuition more precise further on.

For the next definition, recall our notation $\overline{\mu} = \nu$, $\overline{\nu} = \mu$.

Definition 2.55 By natural induction we define classes $\Theta^\mu_n, \Theta^\nu_n$ of μ-calculus formulas. With $\eta, \lambda \in \{\mu, \nu\}$ arbitrary, we set:

1. all atomic formulas belong to Θ^η_0;
2. if $\varphi_0, \varphi_1 \in \Theta^\eta_n$, then $\varphi_0 \vee \varphi_1, \varphi_0 \wedge \varphi_1, \Diamond\varphi_0, \Box\varphi_0 \in \Theta^\eta_n$;
3. if $\varphi \in \Theta^\eta_n$ then $\overline{\eta}x.\varphi \in \Theta^\eta_n$;
4. if $\varphi(x), \psi \in \Theta^\eta_n$, then $\varphi[\psi/x] \in \Theta^\eta_n$, provided that ψ is free for x in φ;
5. all formulas in Θ^λ_n belong to Θ^η_{n+1}.

The *alternation depth* $ad(\xi)$ of a formula ξ is defined as the least n such that $\xi \in \Theta^\mu_n \cap \Theta^\nu_n$.

A formula is *alternation free* if it has alternation depth at most 1. ◁

Roughly, we obtain Θ^μ_0 by closing the set of basic modal formulas under the boolean and modal operators, and the *greatest* fixpoint operator; and similarly for Θ^ν_0. Inductively, we obtain Θ^η_{n+1} by closing the set $\Theta^{\overline{\eta}}_n$ under the boolean and modal operations, substitution, and the $\overline{\eta}$-operator.

Remark 2.56 ▶ `Make connection with Σ/Π- notation (CHECK):`
- $\Sigma_0, \Pi_0 := \Theta^\mu_0 \cap \Theta^\nu_0$
- $\Sigma_{n+1} := \Theta^\nu_n$, $\Pi_{n+1} := \Theta^\mu_n$.

◁

Example 2.57 Observe that the basic modal (i.e., fixpoint-free) formulas are exactly the ones with alternation depth zero. Formulas that use μ-operators or ν-operators, but not both, have alternation depth 1. For example, observe that $\mu x.p \vee x$ belongs to Θ^ν_0 but not to Θ^μ_0: none of the clauses in Definition 2.55 is applicable. On the other hand, using clause (5) it is easy to see that $\mu x.p \vee x \in \Theta^\nu_1 \cap \Theta^\mu_1$, from which it is immediate that $ad(\mu x.p \vee x) = 1$.

Consider the formula $\xi_1 = \mu x.(\nu y.p \wedge \Box y) \wedge \Diamond x$. Taking a fresh variable q, we find $\mu x.q \wedge \Diamond x \in \Theta^\nu_0 \subseteq \Theta^\nu_1$ and $\nu y.p \wedge \Box y \in \Theta^\mu_0 \subseteq \Theta^\nu_1$, so that by the substitution rule we have $\xi_1 = (\mu x.q \wedge \Diamond x)[\nu y.p \wedge \Box y/q] \in \Theta^\nu_1$. Similarly we may show that $\xi_1 \in \Theta^\mu_1$, so that ξ_1 has alternation depth 1.

The formula $\xi_2 = \nu x.\mu y.(p \wedge \Diamond x) \vee \Diamond y$ is of higher complexity. It is clear that the formula $\mu y.(p \wedge \Diamond x) \vee \Diamond y$ belongs to Θ^ν_0 but not to Θ^μ_0. From this it follows that ξ_2 belongs to Θ^μ_1 but there is no way to place it in Θ^ν_1. Hence we find that $ad(\xi_2) = 2$.

As a third example, consider the formula

$$\xi_3 = \mu x.\nu y.(\Box y \wedge \mu z.(\Diamond x \vee z)).$$

This formula looks like a $\mu/\nu/\mu$-formula, in the sense that it contains a nested fixpoint chain $\mu x/\nu y/\mu z$. However, the variable y does not occur in the subformula $\mu z.(\Diamond x \vee z)$, and so the variable z is not dependent on y. Consequently we may in fact consider ξ_3 as a μ/ν-formula. Formally, we observe that $\mu z.\Diamond x \vee z \in \Theta^\nu_0 \subseteq \Theta^\nu_1$ and $\nu z.\Box y \wedge p \in \Theta^\mu_0 \subseteq \Theta^\nu_1$; from this it follows by the substitution rule that the formula $\nu y.(\Box y \wedge \mu z.(\Diamond x \vee z))$ belongs to the set Θ^ν_1 as well; from this it easily follows that $\xi_3 \in \Theta^\nu_1$. It is not hard to show that $\xi_3 \notin \Theta^\mu_1$, so that we find $ad(\xi_3) = 2$. ◁

In the propositions below we make some observations on the sets Θ^η_n and on the notion of alternation depth. First we show that each class Θ^μ_n is closed under subformulas and derived formulas.

Proposition 2.58 *Let Let ξ and φ be μ-calculus formulas.*
1) If $\varphi \trianglelefteq \xi$ and $\xi \in \Theta^\eta_n$ then $\varphi \in \Theta^\eta_n$.
2) If $\xi \twoheadrightarrow_C \varphi$ and $\xi \in \Theta^\eta_n$ then $\varphi \in \Theta^\eta_n$.

Proof. We prove the statement in part 1) by induction on the derivation of $\xi \in \Theta^\eta_n$. In the base case of this induction we have that $n = 0$ and ξ is an atomic formula. But then obviously all subformulas of ξ are atomic as well and thus belong to Θ^η_n.

In the induction step of the proof it holds that $n > 0$; we make a case distinction as to the applicable clause of Definition 2.55.

In case $\xi \in \Theta^\eta_n$ because of clause (2) in Definition 2.55, we make a further case distinction as to the syntactic shape of ξ. First assume that ξ is a conjunction, say, $\xi = \xi_0 \wedge \xi_1$, with $\xi_0, \xi_1 \in \Theta^\eta_n$. Now consider an arbitrary subformula φ of ξ; it is not hard to see that either

$\varphi = \xi$ or $\varphi \trianglelefteqslant \xi_i$ for some $i \in \{0, 1\}$. In the first case we are done, by assumption that $\xi \in \Theta^\eta_n$; in the second case, we find $\varphi \in \Theta^\eta_n$ as an immediate consequence of the induction hypothesis. The cases where ξ is a disjunction, or a formula of the form $\Box\psi$ or $\Diamond\psi$ are treated in a similar way.

If $\xi \in \Theta^\eta_n$ because of clause (3) of the definition, then ξ must be of the form $\xi = \eta x.\chi$, with $\chi \in \Theta^\eta_n$. We proceed in a way similar to the previous case: any subformula $\varphi \trianglelefteqslant \xi$ is either equal to ξ (in which case we are done by assumption), or a subformula of χ, in which we are done by one application of the induction hypothesis.

In the case of clause (4), assume that ξ is of the form $\chi[\psi/x]$, where ψ is free for x in χ, and χ and ψ are in Θ^η_n. Then by the induction hypothesis all subformulas of χ and ψ belong to Θ^η_n as well. Now consider an arbitrary subformula φ of ξ; it is easy to see that either $\varphi \trianglelefteqslant \chi$, $\varphi \trianglelefteqslant \psi$ or else φ is of the form $\varphi = \varphi'[\psi/x]$ where $\varphi' \trianglelefteqslant \chi$. In either case it is straightforward to prove that $\varphi \in \Theta^\eta_n$, as required.

Finally, in case ξ is in Θ^η_n because of clause (5), it belongs to Θ^λ_{n-1} for some $\lambda \in \{\mu, \nu\}$. Then by induction hypothesis all subformulas of ξ belong to Θ^λ_{n-1}. We may then apply the same clause (5) to see that any such φ also belongs to the set Θ^η_n.

To prove part 2), it suffices to show that the class Θ^η_n is closed under unfoldings, since by part 1) we already know it to be closed under subformulas. So assume that $\lambda x.\chi \in \Theta^\eta_n$ for some n and $\lambda \in \{\mu, \nu\}$. Because $\chi \trianglelefteqslant \eta x.\chi$ it follows from part 1) that $\chi \in \Theta^\lambda_n$. But then we may apply clause (4) from Definition 2.55 and conclude that $\chi[\eta.\chi/x] \in \Theta^\lambda_n$. QED

As an immediate corollary of Proposition 2.58 we find the following.

Proposition 2.59 *Let ξ and χ be μ-calculus formulas. Then*
1) if $\chi \in Sf(\xi)$ then $ad(\chi) \leq ad(\xi)$;
2) if $\chi \in Cl(\xi)$ then $ad(\chi) \leq ad(\xi)$.

In the case of a *clean* formula there is a simple characterisation of alternation depth, making precise the intuition about alternating chains, in terms of the formula's dependency order on the bound variables.

Definition 2.60 Let $\xi \in \mu\mathtt{ML}$ be a clean formula. A *dependency chain* in ξ of *length* d is a sequence $\overline{x} = x_1 \cdots x_d$ such that $x_1 <_\xi x_2 \cdots <_\xi x_d$; such a chain is *alternating* if x_i and x_{i+1} have different parity, for every $i < d$. For $\eta \in \{\mu, \nu\}$, we call an alternating dependency chain $x_1 \cdots x_d$ an *η-chain* if x_d is an η-variable, and we let $d_\eta(\xi)$ denote the length of the longest η-chain in ξ; we write $d_\eta(\xi) = 0$ if ξ has no such chains. $\lhd$

Proposition 2.61 *Let ξ be a clean formula. Then for any $k \in \omega$ and $\eta \in \{\mu, \nu\}$ we have*

$$\xi \in \Theta^\eta_k \textit{ iff } d_\eta(\xi) \leq k, \tag{18}$$

As a corollary, the alternation depth of ξ is equal to the length of its longest alternating dependency chain.

One of the key insights in the proof of this Proposition is that, with ψ free for x in φ, any dependency chain in $\varphi[\psi/x]$ originates entirely from either φ or ψ. Recall from Definition 8.1 that we write $\overline{\mu} = \nu$ and $\overline{\nu} = \mu$.

Proof. We prove the implication from left to right in (18) by induction on the derivation that $\xi \in \Theta^{\eta}_{k}$. In the base step of this induction (corresponding to clause (1) in the definition of alternation depth) ξ is atomic, so that we immediately find $d_{\eta}(\xi) = 0$ as required.

In the induction step of the proof, we make a case distinction as to the last applied clause in the derivation of $\xi \in \Theta^{\eta}_{k}$, and we leave the (easy) cases, where this clause was either (2) or (3), for the reader.

Suppose then that $\xi \in \Theta^{\eta}_{k}$ on the basis of clause (4). In this case we find that $\xi = \xi'[\psi/z]$ for some formulas ξ', ψ such that ψ is free for z in ξ' and $\xi', \psi \in \Theta^{\eta}_{k}$. By the 'key insight' mentioned right after the formulation of the Proposition, any η-chain in the formula ξ is a η-chain in either ξ' or ψ. But then by the induction hypothesis it follows that the length of any such chain must be bounded by k.

Finally, consider the case where $\xi \in \Theta^{\eta}_{k}$ on the basis of clause (5). We make a further case distinction. If $\xi \in \Theta^{\eta}_{k-1}$, then by the induction hypothesis we may conclude that $d_{\eta}(\xi) \leq k-1$, and from this it is immediate that $d_{\eta}(\xi) \leq k$. If, on the other hand, $\xi \in \Theta^{\overline{\eta}}_{k-1}$ then the induction hypothesis yields $d_{\overline{\eta}}(\xi) \leq k-1$. But since $d_{\eta}(\xi) \leq d_{\overline{\eta}}(\xi) + 1$ we obtain $d_{\eta}(\xi) \leq k$ indeed.

The opposite, right-to-left, implication in (18) is proved by induction on k. In the base step of this induction we have $d_{\eta}(\xi) = 0$, which means that ξ has no η-variables; from this it is easy to derive that $\xi \in \Theta^{\eta}_{0}$.

For the induction step, we assume as our induction hypothesis that (18) holds for $k \in \omega$, and we set out to prove the same statement for $k+1$ and an arbitrary $\eta \in \{\mu, \nu\}$:

$$\text{if } d_{\eta}(\xi) \leq k+1 \text{ then } \xi \in \Theta^{\eta}_{k+1}. \tag{19}$$

We will prove (19) by an 'inner' induction on the length of ξ. The base step of this inner induction is easy to deal with: if $|\xi| = 1$ then ξ must be atomic so that certainly $\xi \in \Theta^{\eta}_{k+1}$.

In the induction step we are considering a formula ξ with $|\xi| > 1$. Assume that $d_{\eta}(\xi) \leq k+1$. We make a case distinction as to the shape of ξ. The only case of interest is where ξ is a fixpoint formula, say, $\xi = \eta x.\chi$ or $\xi = \overline{\eta}x.\chi$. If $\xi = \overline{\eta}x.\chi$, then obviously we have $d_{\eta}(\xi) = \delta_{\eta}(\chi)$, so by the inner induction hypothesis we find $\chi \in \Theta^{\eta}_{k+1}$. From this we immediately derive that $\xi = \overline{\eta}x.\chi \in \Theta^{\eta}_{k+1}$ as well.

Alternatively, if $\xi = \eta x.\chi$, we split further into cases: If χ has an $\overline{\eta}$-chain $y_1 \cdots y_{k+1}$ of length $k+1$, then obviously we have $x \notin FV(\delta_{k+1})$ (where we write δ_{k+1} instead of $\delta_{y_{k+1}}$), for otherwise we would get $x >_{\xi} y_{k+1}$, so that we could add x to the $\overline{\eta}$-chain $y_1 \cdots y_{k+1}$ and obtain an η-chain $y_1 \cdots y_{k+1}x$ of length $k+2$. But if $x \notin FV(\delta_{k+1})$ we may take some fresh variable z and write $\xi = \xi'[\overline{\eta}y_{k+1}.\delta_{k+1}/z]$ for some formula ξ' where the formula $\overline{\eta}y_{k+1}.\delta_{k+1}$ is free for z. By our inner induction hypothesis we find that both ξ' and $\eta y_{k+1}.\delta_{k+1}$ belong to Θ^{η}_{k+1}. But then by clause (4) of Definition 2.55 the formula ξ also belongs to the set Θ^{η}_{k+1}.

If, on the other hand, χ has *no* $\overline{\eta}$-chain of length $k+1$, then we clearly have $d_{\overline{\eta}}(\chi) \leq k$. Using the outer induction hypothesis we infer $\chi \in \Theta^{\overline{\eta}}_{k}$, and so by clause (3) of Definition 2.55 we also find $\xi = \eta x.\chi \in \Theta^{\overline{\eta}}_{k}$. Finally then, clause (5) gives $\xi \in \Theta^{\eta}_{k+1}$. QED

One may prove a similar (but somewhat more involved) characterisation in the wider setting of tidy formulas, as we will see further on.

Notes

The modal μ-calculus was introduced by D. Kozen [15]. Its game-theoretical semantics goes back to at least Emerson & Jutla [11] (who use alternating automata as an intermediate step). As far as we are aware, the bisimulation invariance theorem, with the associated tree model property, is a folklore result. The bounded tree model property is due to Kozen & Parikh [17].

There are various ways to make the notion of alternation depth precise; we work with the most widely used definition, which originates with Niwiński [22].

▶ `More notes to be supplied.`

Exercises

Exercise 2.1 Express in words the meaning of the following μ-calculus formula:

$$\nu x.\mu y.(p \wedge \Box x) \vee (\overline{p} \wedge \Box y).$$

Exercise 2.2 (defining modal μ-formulas) Give a modal μ-formula $\varphi(p, q)$ such that for all transition systems $\mathbb{S}$, and all states s_0 in $\mathbb{S}$:

$$\mathbb{S}, s_0 \Vdash_g \varphi(p,q) \quad \text{iff} \quad \begin{array}{l}\text{there is a path } s_0Rs_1\ldots Rs_n \ (n \geq 0) \text{ such that } \mathbb{S}, s_n \Vdash_g p \\ \text{and } \mathbb{S}, s_i \Vdash_g q \text{ for all } i \text{ with } 0 \leq i < n.\end{array}$$

Exercise 2.3 (characterizing winning strategies)
A *board* is a structure $\mathbb{B} = \langle B_0, B_1, E\rangle$ such that $B_0 \cap B_1 = \varnothing$ and $E \subseteq B^2$, where $B = B_0 \uplus B_1$ is a set of objects called *positions*. A *match* on $\mathbb{B}$ consists of the *players* 0 and 1 moving a token from one position to another, following the edge relation E. Player i is supposed to move the token when it is situated on a position in B_i. Suppose in addition that B is also partitioned into green and red positions, $B = G \uplus R$.

We will use a modal language to describe this structure, with the modalities being interpreted by the edge relation E, the proposition letter p_0 and r referring to the positions belonging to player 0, and the red positions, respectively. That is, $V(p_0) = B_0$ and $V(r) = R$.

(a) Consider the game where player 0 wins as soon as the token reaches a green position. (That is, all infinite matches are won by player 1. Player 0 wins if player 1 gets stuck, or if the token reaches a green position; player 1 wins a finite match if player 0 gets stuck.) Show that the formula $\varphi_a = \mu x.\overline{r} \vee (p_0 \wedge \Diamond x) \vee (\overline{p}_0 \wedge \Box x)$ characterizes the winning positions for player 0 in this game, in the sense that for any position $b \in B$, we have

$$\mathbb{B}, V, b \Vdash_g \varphi \text{ iff player 0 has a w.s. in the game starting at position } b.$$

(b) Now consider the game where player 0 wins if she manages to reach a green position *infinitely often*. (More precisely, infinite matches are won by 0 iff a green position is reached infinitely often; finite matches are lost by a player is he/she gets stuck.) Give a formula φ_b that characterizes the winning positions in this game.

Exercise 2.4 (characterizing fairness) Let $\mathsf{D} = \{a, b\}$ be the set of atomic actions, and consider the following formula ξ, with subformulas as indicated:

$$\xi = \nu x.\mu y.\nu z.\overbrace{\underbrace{\Box_a x}_{\alpha_1} \wedge \underbrace{(\Box_a \bot \vee \Box_b y)}_{\alpha_2} \wedge \underbrace{\Box_b z}_{\alpha_3}}^{\delta}$$

Fix an LTS $\mathbb{S} = (S, R_a, R_b, V)$. We say that the transition a is *enabled* at state s of $\mathbb{S}$ if $\mathbb{S}, s \Vdash_g \Diamond_a \top$.

Show that ξ expresses some kind of *fairness* condition, i.e., the absence of a path starting at s on which a is enabled infinitely often, but executed only finitely often. More precisely, prove that $\mathbb{S}, s \Vdash_g \xi$ iff there is *no* path of the form $s_0 \stackrel{d_0}{\to} s_1 \stackrel{d_1}{\to} s_2 \cdots$ such that $s = s_0$, $d_i \in \{a, b\}$ for all i, a is enabled at s_i for infinitely many i, but $d_i = a$ for only finitely many i.

Exercise 2.5 (filtration) Recall that, given a finite, closed set of formulas Σ and a model $\mathbb{S} = (S, R, V)$, we say that a model $\mathbb{S}' = (S', R', V')$ is a *filtration* of $\mathbb{S}$ through Σ if there is a surjective map $f : S \to S'$ such that:

a) for all proposition letters $p \in \Sigma$: $u \in V(p)$ iff $f(u) \in V'(p)$.

b) uRv implies $f(u)R'f(v)$

c) if $\Diamond\varphi \in \Sigma$ and $f(u)R'f(v)$, then $\mathbb{S}, v \Vdash_g \varphi$ implies $\mathbb{S}, u \Vdash_g \Diamond\varphi$

d) $f(u) = f(v)$ if and only if u and v satisfy precisely the same formulas in Σ.

Say that a formula ξ of the μ-calculus *admits filtration* if, for every model $\mathbb{S}$, there is a finite set of formulas Σ containing ξ, and a filtration $\mathbb{S}'$ of $\mathbb{S}$ through Σ such that $\mathbb{S}', f(s) \Vdash_g \varphi$ iff $\mathbb{S}, s \Vdash_g \varphi$, for each s in $\mathbb{S}$ and each $\varphi \in \Sigma$.

Prove that the formula $\mu x.\Box x$ does not admit filtration.

Exercise 2.6 We write $\varphi \models \psi$ to denote that ψ is a *local consequence* of φ, that is, if for all pointed Kripke models $(\mathbb{S}, s)$ it holds that $\mathbb{S}, s \Vdash_g \varphi$ implies $\mathbb{S}, s \Vdash_g \psi$.

(a) Show that $\mu x.\nu y.\, \alpha(x, y) \models \nu y.\mu x.\, \alpha(x, y)$, for all formulas α.

(b) Show that $\mu x.\mu y.\, \alpha(x, y) \equiv \mu y.\mu x.\, \alpha(x, y)$, for all formulas α.

(c) Show that $\mu x.(x \vee \gamma(x)) \wedge \delta(x) \models \mu x.\gamma(x) \wedge \delta(x)$, for all formulas γ, δ.

Exercise 2.7 (boolean μ-calculus) Show that the least and greatest fixpoint operators do not add expressive power to classical propositional logic, or, in other words, that the modality-free fragment of the modal μ-calculus is expressively equivalent to classical propositional logic. (Hint: use Exercise 2.6(c).)

Exercise 2.8 (co-induction) Let φ, ψ be any two clean formulas of the modal μ-calculus such that ψ is free for x in φ; it will also be convenient to assume that ψ is not a subformula of φ. Show by a game semantic argument that the following so-called 'co-induction principle' holds for greatest fixpoints: if $\psi \models \varphi[\psi/x]$, then $\psi \models \nu x.\varphi$ also. Here we write '$\models$' for the local consequence relation, as in Exercise 2.6.

Exercise 2.9 (injectivity of substitution) Prove Proposition 2.51.

3 Fixpoints

The game-theoretic semantics of the modal μ-calculus introduced in the previous chapter has some attractive characteristics. It is intuitive, relatively easy to understand, and, as we shall see further on, it can be used to prove some important properties of the formalism. However, it has some drawbacks as well. For instance, the evaluation games of the previous chapter have only been defined for formulas that are either clean or tidy. The game semantics can be extended to arbitrary formulas but this will make the game somewhat more involved, in particular if we want to define evaluation games for formulas that are not in negation normal form.

Furthermore, the game-theoretical semantics is not *compositional*; that is, the meaning of a formula is not defined in terms of the meanings of its subformulas. These shortcomings vanish in the *algebraic semantics* that we are about to introduce. In order to define this term, we first consider an example.

Example 3.1 Recall that in Example 2.1, we informally introduced the formula $\mu x.p \vee \Diamond_d x$ as the smallest fixpoint or solution of the 'equation' $x \equiv p \vee \Diamond_d x$.

To make this intuition more precise, we have to look at the formula $\delta = p \vee \Diamond_d x$ as an operation. The idea is that the value (that is, the extension) of this formula is a function of the value of x, provided that we keep the value of p constant. Varying the value of x boils down to considering 'x-variants' of the valuation V of $\mathbb{S} = \langle S, R, V\rangle$. Let, for $X \subseteq S$, $V[x \mapsto X]$ denote the valuation that is exactly like V apart from mapping x to X, and let $\mathbb{S}[x \mapsto X]$ denote the x-variant $\langle S, R, V[x \mapsto X]\rangle$ of $\mathbb{S}$. Then $[\![\delta]\!]^{\mathbb{S}[x\mapsto X]}$ denotes the extension of δ in this x-variant. It follows from this that the formula δ *induces* the following function $\delta^{\mathbb{S}}_x$ on the power set of S:

$$\delta^{\mathbb{S}}_x(X) := [\![\delta]\!]^{\mathbb{S}[x\mapsto X]}.$$

In our example we have

$$\delta^{\mathbb{S}}_x(X) = V(p) \cup \langle R\rangle(X).$$

Now we can make precise why $\mu x.p \vee \Diamond_d x$ is a fixpoint formula: its extension, the set $[\![\mu x.p \vee \Diamond_d x]\!]$, is a fixpoint of the map $\delta^{\mathbb{S}}_x$:

$$[\![\mu x.p \vee \Diamond_d x]\!] = V(p) \cup \langle R\rangle([\![\mu x.p \vee \Diamond_d x]\!]).$$

In fact, as we shall see in this chapter, the formulas $\mu x.p \vee \Diamond_d x$ and $\nu x.p \vee \Diamond_d x$ are such that their extensions are the *least* and *greatest* fixpoints of the map $\delta^{\mathbb{S}}_x$, respectively. $\lhd$

It is worthwhile to discuss the theory of fixpoint operators at a more general level than that of modal logic. Before we turn to the definition of the algebraic semantics of the modal μ-calculus, we first discuss the general fixpoint theory of monotone operations on complete lattices.

3.1 General fixpoint theory

Basics

In this chapter we assume some familiarity[2] with partial orders and lattices (see Appendix A).

Definition 3.2 Let $\mathbb{P}$ and $\mathbb{P}'$ be two partial orders and let $f : P \to P'$ be some map. Then f is called *monotone* or *order preserving* if $f(x) \leq' f(y)$ whenever $x \leq y$, and *antitone* or *order reversing* if $f(x) \geq' f(y)$ whenever $x \leq y$. ◁

Definition 3.3 Let $\mathbb{P} = \langle P, \leq \rangle$ be a partial order, and let $f : P \to P$ be some map. Then an element $p \in P$ is called a *prefixpoint* of f if $f(p) \leq p$, a *postfixpoint* of f if $p \leq f(p)$, and a *fixpoint* if $f(p) = p$. The sets of prefixpoints, postfixpoints, and fixpoints of f are denoted respectively as $\mathrm{PRE}(f)$, $\mathrm{POS}(f)$ and $\mathrm{FIX}(f)$.

In case the set of fixpoints of f has a least (respectively greatest) member, this element is denoted as $\mathrm{LFP}.f$ ($\mathrm{GFP}.f$, respectively). These least and greatest fixpoints may also be called *extremal fixpoints*. ◁

The following theorem is a celebrated result in fixpoint theory.

Theorem 3.4 (Knaster-Tarski) *Let $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$ be a complete lattice, and let $f : C \to C$ be monotone. Then f has both a least and a greatest fixpoint, and these are given as*

$$\mathrm{LFP}.f = \bigwedge \mathrm{PRE}(f), \tag{20}$$

$$\mathrm{GFP}.f = \bigvee \mathrm{POS}(f). \tag{21}$$

Proof. We will only prove the result for the least fixpoint, the proof for the greatest fixpoint is completely analogous.

Define $q := \bigwedge \mathrm{PRE}(f)$, then we have that $q \leq x$ for all prefixpoints x of f. From this it follows by monotonicity that $f(q) \leq f(x)$ for all $x \in \mathrm{PRE}(f)$, and hence by definition of prefixpoints, $f(q) \leq x$ for all $x \in \mathrm{PRE}(f)$. In other words, $f(q)$ is a lower bound of the set $\mathrm{PRE}(f)$. Hence, by definition of q as the *greatest* such lower bound, we find $f(q) \leq q$, that is, q itself is a prefixpoint of f.

It now suffices to prove that $q \leq f(q)$, and for this we may show that $f(q)$ is a prefixpoint of f as well, since q is by definition a lower bound of the set of prefixpoints. But in fact, we may show that $f(y)$ is a prefixpoint of f for *every* prefixpoint y of f — by monotonicity of f it immediately follows from $f(y) \leq y$ that $f(f(y)) \leq f(y)$. QED

Another way to obtain least and greatest fixpoints is to *approximate* them from below and above, respectively.

Definition 3.5 Let $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$ be a complete lattice, and let $f : C \to C$ be some map. Then by ordinal induction we define the following maps on C:

$$\begin{array}{rcl} f^0_\mu(c) & := & c, \\ f^{\alpha+1}_\mu(c) & := & f(f^\alpha_\mu(c)) \\ f^\lambda_\mu(c) & := & \bigvee_{\alpha<\lambda} f^\alpha_\mu(c) \end{array} \qquad \begin{array}{rcl} f^0_\nu(c) & := & c, \\ f^{\alpha+1}_\nu(c) & := & f(f^\alpha_\nu(c)), \\ f^\lambda_\nu(c) & := & \bigwedge_{\alpha<\lambda} f^\alpha_\nu(c), \end{array}$$

[2]Readers lacking this background may take abstract complete lattices to be concrete power set algebras.

where λ denotes an arbitrary limit ordinal. $\lhd$

Proposition 3.6 *Let $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$ be a complete lattice, and let $f : C \to C$ be monotone. Then f is inductive, that is, $f^\alpha_\mu(\bot) \leq f^\beta_\mu(\bot)$ for all ordinals α and β such that $\alpha < \beta$.*

Proof. We leave this proof as an exercise to the reader. QED

Given a set C, we let $|C|$ denote its cardinality or size.

Corollary 3.7 *Let $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$ be a complete lattice, and let $f : C \to C$ be monotone. Then there is some α of size at most $|C|$ such that* $\mathrm{LFP}.f = f^\alpha_\mu(\bot)$.

Proof. By Proposition 3.6, f is inductive, that is, $f^\alpha_\mu(\bot) \leq f^\beta_\mu(\bot)$ for all ordinals α and β such that $\alpha < \beta$. It follows from elementary set theory that there must be two ordinals α, β of size at most $|C|$ such that $f^\alpha_\mu(\bot) = f^\beta_\mu(\bot)$. From the definition of the approximations it then follows that there must be an ordinal α such that $f^\alpha_\mu(\bot) = f^{\alpha+1}_\mu(\bot)$, or, equivalently, $f^\alpha_\mu(\bot)$ is a fixpoint of f. To show that it is the *smallest* fixpoint, one may prove that $f^\beta_\mu(\bot) \leq \mathrm{LFP}.f$ for every ordinal β. This follows from a straightforward ordinal induction. QED

Definition 3.8 Let $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$ be a complete lattice, and let $f : C \to C$ be monotone. The least ordinal α such that $f^\alpha_\mu(\bot) = f^{\alpha+1}_\mu(\bot)$ is called the *unfolding ordinal* of f. $\lhd$

3.2 Boolean algebras

In the special case that the complete lattice is in fact a (complete) *boolean algebra*, there is more to be said.

Dual maps

In the case of monotone maps on complete boolean algebras, the least and greatest fixed points become interdefinable, using the notion of (boolean) *duals* of maps.

Definition 3.9 A *complete* boolean algebra is a structure $\mathbb{B} = \langle B, \bigvee, \bigwedge, - \rangle$ such that $\langle B, \bigvee, \bigwedge \rangle$ is a complete lattice and $\langle B, \vee, \wedge, -, \bot, \top \rangle$ is a boolean algebra, where $\vee$ and $\wedge$ are the binary versions of $\bigvee$ and $\bigwedge$, respectively, and $\bot := \bigvee \varnothing$, $\top := \bigwedge \varnothing$. $\lhd$

In a boolean algebra $\mathbb{B}$, the complementation operation $- : B \to B$ is an antitone (order-reversing) map such that $x \wedge -x = \bot$ and $x \vee -x = \top$ for all $x \in B$. If $\mathbb{B}$ is complete it holds that $-\bigvee X = \bigwedge\{-x \mid x \in X\}$ and $-\bigwedge X = \bigvee\{-x \mid x \in X\}$.

Definition 3.10 Let $\mathbb{B} = \langle B, \bigvee, \bigwedge, - \rangle$ be a complete boolean algebra. Given a map $f : B \to B$, the function $f^\partial : B \to B$ given by

$$f^\partial(b) := -f(-b).$$

is called the *(boolean) dual* of f. $\lhd$

Proposition 3.11 *Let $\mathbb{B} = \langle B, \bigvee, \bigwedge, - \rangle$ be a complete boolean algebra, and let $g : B \to B$ be monotone. Then g^∂ is monotone as well, $(g^\partial)^\partial = g$, and*

$$\begin{array}{rcl} \mathrm{LFP}.g^\partial & = & -\mathrm{GFP}.g, \\ \mathrm{GFP}.g^\partial & = & -\mathrm{LFP}.g. \end{array}$$

Proof. We only prove that $\mathrm{LFP}.g^\partial = -\mathrm{GFP}.g$, leaving the other parts of the proof as exercises to the reader.

First, note that by monotonicity of g^∂, the Knaster-Tarski theorem gives that

$$\mathrm{LFP}.g^\partial = \bigwedge \mathrm{PRE}(g^\partial).$$

But as a consequence of the definitions, we have that

$$b \in \mathrm{PRE}(g^\partial) \iff -b \in \mathrm{POS}(g).$$

From this it follows that

$$\begin{array}{rcl} \mathrm{LFP}.g^\partial & = & \bigwedge\{b \mid -b \in \mathrm{POS}(g)\} \\ & = & \bigwedge\{-a \mid a \in \mathrm{POS}(g)\} \\ & = & -\bigvee \mathrm{POS}(g) \\ & = & -\mathrm{GFP}.g \end{array}$$

which finishes the proof of the Theorem. QED

Further on we will see that Proposition 3.11 allows us to define negation as an abbreviated operator in the modal μ-calculus.

Games

In case the boolean algebra in question is in fact a *power set algebra*, a nice game-theoretic characterization of least and greatest fixpoint operators can be given.

Definition 3.12 Let S be some set and let $F : \wp(S) \to \wp(S)$ be a monotone operation. Consider the *unfolding games* $\mathcal{U}^\mu(F)$ and $\mathcal{U}^\nu(F)$. The positions and admissible moves of these two graph games are the same, see Table 6.

Position	Player	Admissible moves
$s \in S$	$\exists$	$\{A \in \wp(S) \mid s \in F(A)\}$
$A \in \wp(S)$	$\forall$	A

Table 6: Unfolding games for $F : \wp(S) \to \wp(S)$

The *winning conditions* of finite matches are standard (the player that got stuck loses the match). The difference between $\mathcal{U}^\mu(F)$ and $\mathcal{U}^\nu(F)$ shows up in the winning conditions of infinite matches: $\exists$ wins the infinite matches of $\mathcal{U}^\nu(F)$, but $\forall$ those of $\mathcal{U}^\mu(F)$. $\lhd$

Observe that the positions in a match of the unfolding game alternate between 'state positions' s, where $\exists$ needs to pick a subset $A \subseteq S$ such that s belongs to $F(A)$, and 'subset positions' A, of which $\forall$ has to pick an element.

Example 3.13 In fact, we have already seen an example of the unfolding game $\mathcal{U}^\nu$ in the *bisimilarity game* of Definition 1.26. Given two Kripke models $\mathbb{S}$ and $\mathbb{S}'$, consider the map $F : \wp(S \times S')$ given by

$$F(Z) := \{(s, s') \in S \times S' \mid Z \text{ is a local bisimulation for } s \text{ and } s'\},$$

then it is straightforward to verify that $\mathcal{B}(\mathbb{S}, \mathbb{S}')$ is nothing but the unfolding game $\mathcal{U}^\nu(F)$. $\lhd$

The following proposition substantiates the slogan that 'ν means unfolding, μ means finite unfolding'.

Theorem 3.14 *Let S be some set and let $F : \wp(S) \to \wp(S)$ be a monotone operation. Then*

1. $\mathrm{GFP}.F = \{s \in S \mid s \in \mathrm{Win}_\exists(\mathcal{U}^\nu(F))\}$,
2. $\mathrm{LFP}.F = \{s \in S \mid s \in \mathrm{Win}_\exists(\mathcal{U}^\mu(F))\}$,

Proof. For the inclusion $\supseteq$ of part 1, it suffices to prove that $W := S \cap \mathrm{Win}_\exists(\mathcal{U}^\nu(F))$ is a postfixpoint of F:

$$W \subseteq F(W). \tag{22}$$

Let s be an arbitrary point in W, and suppose that $\exists$'s winning strategy tells her to choose $A \subseteq S$ at position s. Then no matter what element $s_1 \in A$ is picked by $\forall$, $\exists$ can continue the match and win. Hence, all elements of A are winning positions for $\exists$. But from $A \subseteq W$ it follows that $F(A) \subseteq F(W)$, and by the legitimacy of $\exists$'s move A at s it holds that $s \in F(A)$. We conclude that $s \in F(W)$, which proves (22).

For the converse inclusion $\subseteq$ of part 1 of the proposition, take an arbitrary point $s \in \mathrm{GFP}.F$. We need to provide $\exists$ with a winning strategy in the unfolding game $\mathcal{U}^\nu(F)$ starting at s. This strategy is actually as simple as can be: $\exists$ should always play $\mathrm{GFP}.F$. Since $\mathrm{GFP}.F = F(\mathrm{GFP}.F)$, this strategy prescribes legitimate moves for $\exists$ at every point in $\mathrm{GFP}.F$. And, if she sticks to this strategy, $\exists$ will stay alive forever and thus win the match, no matter what $\forall$'s responses are.

For the second part of the theorem, let W denote the set $W := S \cap \mathrm{Win}_\exists(\mathcal{U}^\mu(F))$ of states in S that are winning positions for $\exists$ in $\mathcal{U}^\mu(F)$. We first prove the inclusion $W \subseteq \mathrm{LFP}.F$. Clearly it suffices to show that all points outside the set $\mathrm{LFP}.F$ are winning positions for $\forall$.

Consider a point $s \notin \mathrm{LFP}.F$. If $s \notin F(A)$ for any $A \subseteq S$ then $\exists$ is stuck, hence loses immediately, and we are done. Otherwise, suppose that $\exists$ starts a match of $\mathcal{U}^\mu(F)$ by playing some set $B \subseteq S$ with $s \in F(B)$. We claim that B is not a subset of $\mathrm{LFP}.F$, since otherwise we would have $F(B) \subseteq F(\mathrm{LFP}.F) \subseteq \mathrm{LFP}.F$; which would contradict the fact that $s \notin \mathrm{LFP}.F$. But if $B \not\subseteq \mathrm{LFP}.F$ then $\forall$ may continue the match by choosing a point $s_1 \in B \setminus \mathrm{LFP}.F$. Now $\forall$ can use the same strategy from s_1 as he used from s, and so on. This strategy guarantees that either $\exists$ gets stuck after finitely many rounds (in case $\forall$ manages to pick an s_n for which there is no A such that $s_n \in F(A_n)$), or else the match will last forever. In both cases $\forall$ wins the match.

For the opposite inclusion $\subseteq$ of part 2, it suffices to show that W is a prefixpoint of F, that is, $F(W) \subseteq W$. For that purpose, let $s \in S$ be such that $s \in F(W)$. In order to show that $s \in W$ we need to provide $\exists$ with a winning strategy in $\mathcal{U}^\mu(F)$, starting at s. But this is straightforward: since $s \in F(W)$, the set W itself is a legitimate move for $\exists$ at position s. Then, after $\forall$ picks some element $t \in W$, she can simply continue with her strategy in $\mathcal{U}^\mu(F)$ that is winning when starting at position t. QED

3.3 Vectorial fixpoints

Suppose that we are given a finite family $\{\mathbb{C}_1, \ldots, \mathbb{C}_n\}$ of complete lattices, and put $\mathbb{C} = \prod_{1 \leq i \leq n} \mathbb{C}_i$. Given a finite family of monotone maps $f_1, \ldots, f_n$ with $f_i : C \to C_i$, we may define the map $f : C \to C$ given by $f(c) := (f_1(c), \ldots, f_n(c))$. Monotonicity of f is an easy consequence of the monotonicity of the maps f_i separately, and so by completeness of $\mathbb{C}$, f has a least and a greatest fixpoint. In this context we will also use vector notation, for instance writing

$$\mu \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} . \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ f_2(x_1, \ldots, x_n) \\ \vdots \\ f_n(x_1, \ldots, x_n) \end{pmatrix}$$

for LFP.f. An obvious question is whether one may express these multi-dimensional fixpoints in terms of one-dimensional fixpoints of maps that one may associate with $f_1, \ldots, f_n$.

The answer to this question is positive, and the basic observation facilitating the computation of multi-dimensional fixpoints is the following so-called *Bekič principle.*

Proposition 3.15 *Let $\mathbb{D}_1$ and $\mathbb{D}_2$ be two complete lattices, and let $f_i : D_1 \times D_2 \to D_i$ for $i = 1, 2$ be monotone maps. Then*

$$\eta \begin{pmatrix} x \\ y \end{pmatrix} . \begin{pmatrix} f_1(x, y) \\ f_2(x, y) \end{pmatrix} = \begin{pmatrix} \eta x. f_1(x, \eta y. f_2(x, y)) \\ \eta y. f_2(\eta x. f_1(x, y), y) \end{pmatrix}$$

where η uniformly denotes either μ or ν.

Proof. Define $\mathbb{D} := \mathbb{D}_1 \times \mathbb{D}_2$, and let $f : D \to D$ be given by putting $f(d) := (f_1(d), f_2(d))$. Then f is clearly monotone, and so it has both a least and a greatest fixpoint.

By the order duality principle it suffices to consider the case $\eta = \mu$ of least fixed points only. Suppose that (a_1, a_2) is the least fixpoint of f, and let b_1 and b_2 be given by

$$\begin{cases} b_1 := \mu x. f_1(x, \mu y. f_2(x, y)), \\ b_2 := \mu y. f_2(\mu x. f_1(x, y), y). \end{cases}$$

Then we need to show that $a_1 = b_1$ and $a_2 = b_2$.

By definition of (a_1, a_2) we have

$$\begin{cases} a_1 = f_1(a_1, a_2), \\ a_2 = f_2(a_1, a_2), \end{cases}$$

whence we obtain

$$\begin{cases} \mu x.f_1(x, a_2) & \leq & a_1 & \text{and} \\ \mu y.f_2(a_1, y) & \leq & a_2, \end{cases}$$

From this we obtain by monotonicity that

$$f_2(\mu x.f_1(x, a_2), a_2) \leq f_2(a_1, a_2) = a_2,$$

so that we find $b_2 \leq a_2$ by definition of b_2. Likewise we may show that $b_1 \leq a_1$.

Conversely, by definition of b_1 and b_2 we have

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} f_1(b_1, \mu y.f_2(b_1, y)) \\ f_2(\mu x.f_1(x, b_2), b_2) \end{pmatrix}.$$

Then with $c_2 := \mu y.f_2(b_1, y)$, we have $b_1 = f_1(b_1, c_2)$. Also, by definition of c_2 as a fixpoint, $c_2 = f_2(b_1, c_2)$. Putting these two identities together, we find that

$$\begin{pmatrix} b_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} f_1(b_1, c_2) \\ f_2(b_1, c_2) \end{pmatrix} = f \begin{pmatrix} b_1 \\ c_2 \end{pmatrix}.$$

Hence by definition of (a_1, a_2), we find that $a_1 \leq b_1$ (and that $a_2 \leq c_2$, but that is of less interest now). Analogously, we may show that $a_2 \leq b_2$. QED

Proposition 3.15 allows us to compute the least and greatest fixpoints of any monotone map f on a finite product of complete lattices in terms of the least and greatest fixpoints of operations on the factors of the product, through a *elimination method* that is reminiscent of Gaussian elimination in linear algebra.

To see how it works, suppose that we are dealing with lattices $\mathbb{C}_1, \ldots, \mathbb{C}_{n+1}, \mathbb{C}$ and maps $f_1, \ldots, f_{n+1}, f$, just as described above, and that we want to compute $\eta\vec{x}.f$, that is, find the elements $a_1, \ldots, a_{n+1}$ such that

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n+1} \end{pmatrix} = \eta \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n+1} \end{pmatrix} . \begin{pmatrix} f_1(x_1, \ldots, x_n, x_{n+1}) \\ f_2(x_1, \ldots, x_n, x_{n+1}) \\ \vdots \\ f_{n+1}(x_1, \ldots, x_n, x_{n+1}) \end{pmatrix}$$

We may define

$$g_{n+1}(x_1, \ldots, x_n) := \eta x_{n+1}.f_{n+1}(x_1, \ldots, x_{n+1}),$$

and then use Proposition 3.15, with $\mathbb{D}_1 = \mathbb{C}_1 \times \cdots \times \mathbb{C}_n$, and $\mathbb{D}_2 = \mathbb{C}_{n+1}$, to obtain

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} = \eta \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} . \begin{pmatrix} f_1(x_1, \ldots, x_n, g_{n+1}(x_1, \ldots, x_n)) \\ f_2(x_1, \ldots, x_n, g_{n+1}(x_1, \ldots, x_n)) \\ \vdots \\ f_n(x_1, \ldots, x_n, g_{n+1}(x_1, \ldots, x_n)) \end{pmatrix}$$

We may then inductively assume to have obtained the tuple $(a_1, \ldots, a_n)$. Finally, we may compute $a_{n+1} := g_{n+1}(a_1, \ldots, a_n)$.

Observe that in case $\mathbb{C}_i = \mathbb{C}_j$ for all i, j and the operations f_i are all term definable in some formal fixpoint language, then each of the components a_i of the extremal fixpoints of f can also be expressed in this language.

3.4 Algebraic semantics for the modal μ-calculus

Basic definitions

In order to define the algebraic semantics of the modal μ-calculus, we need to consider formulas as *operations* on the power set of the (state space of a) transitions system, and we have to prove that such operations indeed have least and greatest fixpoints. In order to make this precise, we need some preliminary definitions.

Definition 3.16 Given an LTS $\mathbb{S} = \langle S, V, R \rangle$ and subset $X \subseteq S$, define the valuation $V[x \mapsto X]$ by putting

$$V[x \mapsto X](y) \ := \ \begin{cases} V(y) & \text{if } y \neq x, \\ X & \text{if } y = x. \end{cases}$$

Then, the LTS $\mathbb{S}[x \mapsto X]$ is given as the structure $\langle S, V[x \mapsto X], R \rangle$. $\lhd$

Now inductively assume that $[\![\varphi]\!]^{\mathbb{S}}$ has been defined for all LTSs. Given a labelled transition system $\mathbb{S}$ and a propositional variable $x \in \mathsf{P}$, each formula φ induces a map $\varphi^{\mathbb{S}}_x : \wp(S) \to \wp(S)$ defined by

$$\varphi^{\mathbb{S}}_x(X) := [\![\varphi]\!]^{\mathbb{S}[x \mapsto X]}$$

Example 3.17 a) Where $\varphi_a = p \lor x$ we have $(\varphi_a)^{\mathbb{S}}_x(X) = [\![p \lor x]\!]^{\mathbb{S}[x \mapsto X]} = V(p) \cup X$.

b) Where $\varphi_b = \overline{x}$ we have $(\varphi_b)^{\mathbb{S}}_x(X) = [\![\overline{x}]\!]^{\mathbb{S}[x \mapsto X]} = S \setminus X$.

c) Where $\varphi_c = p \lor \Diamond_d x$ we find $(\varphi_c)^{\mathbb{S}}_x(X) = [\![p \lor \Diamond_d x]\!]^{\mathbb{S}[x \mapsto X]} = V(p) \cup \langle R_d \rangle X$.

d) Where $\varphi_d = \Diamond_d \overline{x}$ we find $(\varphi_d)^{\mathbb{S}}_x(X) = [\![\Diamond_d \overline{x}]\!]^{\mathbb{S}[x \mapsto X]} = \langle R_d \rangle (S \setminus X)$. $\lhd$

Remark 3.18 Clearly, relative to a model $\mathbb{S}$, X is a fixpoint of $\varphi^{\mathbb{S}}_x$ iff $X = \varphi^{\mathbb{S}}_x(X)$; a prefixpoint iff $\varphi^{\mathbb{S}}_x(X) \subseteq X$ and a postfixpoint iff $X \subseteq \varphi^{\mathbb{S}}_x(X)$.

Writing $\mathbb{S} \Vdash \varphi$ for $S = [\![\varphi]\!]^{\mathbb{S}}$, an alternative but equivalent way of formulating this is to say that in $\mathbb{S}$, X is a *prefixpoint of a formula* $\varphi(x)$ iff $\mathbb{S}[x \mapsto X] \Vdash \varphi \to x$, a *postfixpoint* iff $\mathbb{S}[x \mapsto X] \Vdash x \to \varphi$, and a *fixpoint* iff $\mathbb{S}[x \mapsto X] \Vdash x \leftrightarrow \varphi$. $\lhd$

Example 3.19 Consider the formulas of Example 3.17.

a) The sets $V(p)$ and S are fixpoints of φ_a, as is in fact any X with $V(p) \subseteq X \subseteq S$.

b) Since we do not consider structures with empty domain, the formula $\overline{x}$ has no fixpoints at all. (Otherwise X would be identical to its own complement relative to some nonempty set S.)

c) Two fixpoints of φ_c were already given in Example 2.1.

d) Consider any model $\mathbb{Z} = \langle Z, S, V \rangle$ based on the set Z of integers, where $S = \{(z, z+1) \mid z \in Z\}$ is the successor relation. Then the only two fixpoints of φ_d are the sets of even and odd numbers, respectively. $\lhd$

In particular, it is not the case that every formula has a least fixpoint. If we can guarantee that the induced function $\varphi^{\mathbb{S}}_x$ of φ is monotone, however, then the Knaster-Tarski theorem (Theorem 3.4) provides both least and greatest fixpoints of $\varphi^{\mathbb{S}}_x$. Precisely for this reason, in the definition of fixpoint formulas, we imposed the condition in the clauses for $\eta x.\varphi$, that x may only occur positively in φ. As we will see, this condition on x guarantees monotonicity of the function $\varphi^{\mathbb{S}}_x$.

Definition 3.20 Given a $\mu\mathtt{ML}_{\mathsf{D}}$-formula φ and a labelled transition system $\mathbb{S} = \langle S, V, R \rangle$, we define the *meaning* $[\![\varphi]\!]^{\mathbb{S}}$ of φ in $\mathbb{S}$, together with the map $\varphi^{\mathbb{S}}_x : \wp(S) \to \wp(S)$ by the following simultaneous formula induction:

$$
\begin{array}{lcllcl}
[\![\bot]\!]^{\mathbb{S}} & = & \varnothing & [\![\top]\!]^{\mathbb{S}} & = & S \\
[\![p]\!]^{\mathbb{S}} & = & V(p) & [\![\overline{p}]\!]^{\mathbb{S}} & = & S \setminus V(p) \\
[\![\varphi \vee \psi]\!]^{\mathbb{S}} & = & [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} & [\![\varphi \wedge \psi]\!]^{\mathbb{S}} & = & [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}} \\
[\![\Diamond_d \varphi]\!]^{\mathbb{S}} & = & \langle R_d \rangle [\![\varphi]\!]^{\mathbb{S}} & [\![\Box_d \varphi]\!]^{\mathbb{S}} & = & [R_d][\![\varphi]\!]^{\mathbb{S}} \\
[\![\mu x.\varphi]\!]^{\mathbb{S}} & = & \bigcap \mathrm{PRE}(\varphi^{\mathbb{S}}_x) & [\![\nu x.\varphi]\!]^{\mathbb{S}} & = & \bigcup \mathrm{POS}(\varphi^{\mathbb{S}}_x)
\end{array}
$$

The map $\varphi^{\mathbb{S}}_x$, for $x \in \mathsf{Prop}$, is given by $\varphi^{\mathbb{S}}_x(X) = [\![\varphi]\!]^{\mathbb{S}[x \mapsto X]}$. $\lhd$

Theorem 3.21 *Let φ be an $\mu\mathtt{ML}_{\mathsf{D}}$-formula, in which x occurs only positively, and let $\mathbb{S}$ be a labelled transition system. Then $[\![\mu x.\varphi]\!]^{\mathbb{S}} = \mathrm{LFP}.\varphi^{\mathbb{S}}_x$, and $[\![\nu x.\varphi]\!]^{\mathbb{S}} = \mathrm{GFP}.\varphi^{\mathbb{S}}_x$.*

Proof. This is an immediate consequence of the Knaster-Tarski theorem, provided we can prove that $\varphi^{\mathbb{S}}_x$ is monotone in x if all occurrences of x in φ are positive. We leave the details of this proof to the reader (see Exercise 3.2). QED

Negation in the modal μ-calculus

It follows from the definitions that the set $\mu\mathtt{ML}_{\mathsf{D}}$ is closed under taking *negations*. Informally, let $\sim\varphi$ be the result of simultaneously replacing all occurrences of $\top$ with $\bot$, of p with $\overline{p}$ and vice versa (for *free* variables p), of $\wedge$ with $\vee$, of $\Box_d$ with $\Diamond_d$, of μx with νx, and vice versa, while leaving occurrences of bound variables unchanged. As an example, $\sim(\mu x.p \vee \Diamond x) = \nu x.\overline{p} \wedge \Box x$. Formally, it is easiest to define $\sim\varphi$ via the *boolean dual* of φ.

Definition 3.22 Given a modal fixpoint formula φ, we define its *boolean dual* φ^{∂} inductively as follows:

$$
\begin{array}{lcllcl}
\bot^{\partial} & := & \top & \top^{\partial} & := & \bot \\
p^{\partial} & := & p & (\overline{p})^{\partial} & := & \overline{p} \\
(\varphi \vee \psi)^{\partial} & := & \varphi^{\partial} \wedge \psi^{\partial} & (\varphi \wedge \psi)^{\partial} & := & \varphi^{\partial} \vee \psi^{\partial} \\
(\Diamond_d \varphi)^{\partial} & := & \Box_d \varphi^{\partial} & (\Box_d \varphi)^{\partial} & := & \Diamond_d \varphi^{\partial} \\
(\mu x.\varphi)^{\partial} & := & \nu x.\varphi^{\partial} & (\nu x.\varphi)^{\partial} & := & \mu x.\varphi^{\partial}
\end{array}
$$

Based on this definition, we define the formula $\sim\varphi$ as the formula $\varphi^{\partial}[p \rightleftharpoons \overline{p} \mid p \in FV(\varphi)]$ that we obtain from φ^{∂} by replacing all occurrences of p with $\overline{p}$, and vice versa, for all free proposition letters $p \in FV(\varphi)$. $\lhd$

Example 3.23 Here are two examples:

$$
\begin{array}{lcllcl}
\varphi & := & \mu x.p \vee \Diamond(x \wedge \overline{q}) & \psi & := & \nu p\, \mu x.p \vee \Diamond(x \wedge \overline{q}) \\
\varphi^{\partial} & := & \nu x.p \wedge \Box(x \vee \overline{q}) & \psi^{\partial} & := & \mu p\, \nu x.p \wedge \Box(x \vee \overline{q}) \\
\sim\varphi & := & \nu x.\overline{p} \wedge \Box(x \vee q) & \sim\psi & := & \mu p\, \nu x.p \wedge \Box(x \vee q)
\end{array}
$$

Note the difference between $\sim\varphi$ and $\sim\psi$ with respective to the propositional variable p, which is free in φ but bound in ψ. $\lhd$

The following proposition states that the operation $\sim$ functions as a standard boolean negation. We let $\sim_S X := S \setminus X$ denote the complement of X in S.

Proposition 3.24 *Let φ be a modal fixpoint formula. Then $\sim\varphi$ corresponds to the negation of φ, that is,*

$$[\![\sim\varphi]\!]^{\mathbb{S}} = \sim_S [\![\varphi]\!]^{\mathbb{S}} \tag{23}$$

for every labelled transition system $\mathbb{S}$.

Proof. We first show, by induction on φ, that φ^{∂} corresponds to the boolean dual of φ. For this purpose, given a labelled transition system $\mathbb{S} = (S, R, V)$, we let $\mathbb{S}^{\sim}$ denote the *complemented* model, that is, the structure $(S, R, V^{\sim})$, where $V^{\sim}(p) := \sim_S V(p)$. Then we claim that

$$[\![\varphi^{\partial}]\!]^{\mathbb{S}} = \sim_S [\![\varphi]\!]^{\mathbb{S}^{\sim}}, \tag{24}$$

and we prove this statement by induction on the complexity of φ. Leaving all other cases as exercises for the reader, we concentrate on the inductive case where φ is of the form $\mu x.\psi$. In this case the left hand side of (24) evaluates to

$$\begin{aligned} [\![(\mu x.\psi)^{\partial}]\!]^{\mathbb{S}} &= [\![\nu x.\psi^{\partial}]\!]^{\mathbb{S}} && \text{(Definition } (\mu x.\psi)^{\partial}) \\ &= \mathrm{GFP}.(\psi^{\partial})^{\mathbb{S}}_x && \text{(Theorem 3.21)} \end{aligned}$$

while for the right hand side we find

$$\begin{aligned} \sim_S [\![\mu x.\psi]\!]^{\mathbb{S}^{\sim}} &= \sim_S \mathrm{LFP}.\psi^{\mathbb{S}^{\sim}}_x && \text{(Theorem 3.21)} \\ &= \mathrm{GFP}.(\psi^{\mathbb{S}^{\sim}}_x)^{\partial} && \text{(Proposition 3.11)} \end{aligned}$$

In other words, to prove (24) it suffices to show that

$$(\psi^{\partial})^{\mathbb{S}}_x = (\psi^{\mathbb{S}^{\sim}}_x)^{\partial}. \tag{25}$$

To this aim, take an arbitrary subset U of S. Applying the map on the left hand side of (25) to U, we find

$$(\psi^{\partial})^{\mathbb{S}}_x(U) = [\![\psi^{\partial}]\!]^{\mathbb{S}[x \mapsto U]},$$

while the map on the right hand side yields

$$(\psi^{\mathbb{S}^{\sim}}_x)^{\partial}(U) = \sim_S \psi^{\mathbb{S}^{\sim}}_x(\sim_S U) = \sim_S [\![\psi]\!]^{(\mathbb{S}^{\sim}[x \mapsto \sim_S U])} = \sim_S [\![\psi]\!]^{(\mathbb{S}[x \mapsto U])^{\sim}},$$

so that by the inductive hypothesis we find that $(\psi^{\partial})^{\mathbb{S}}_x(U) = (\psi^{\mathbb{S}^{\sim}}_x)^{\partial}(U)$, as required to prove (25), and thus (24).

In other words, we have shown that the formula φ^{∂} indeed behaves as the boolean dual of φ. To see that, likewise, the formula $\sim\varphi$ behaves as the negation of φ, we now show how to derive (23) from (24). First observe that for any formula χ we have

$$[\![\chi[p \rightleftharpoons \overline{p} \mid p \in FV(\chi)]]\!]^{\mathbb{S}} = [\![\chi]\!]^{\mathbb{S}^{\sim}}. \tag{26}$$

But then, taking φ^{∂} for χ, we find that

$$[\![\sim\varphi]\!]^{\mathbb{S}} = [\![\varphi^{\partial}[p \rightleftharpoons \overline{p} \mid p \in FV(\varphi)]]\!]^{\mathbb{S}} = [\![\varphi^{\partial}]\!]^{\mathbb{S}^{\sim}} = \sim_S [\![\varphi]\!]^{(\mathbb{S}^{\sim})^{\sim}} = \sim_S [\![\varphi]\!]^{\mathbb{S}},$$

where the first equality holds by the definition of $\sim\varphi$, the second by (26), the third equality is (24), and the fourth equality follows from the trivial observation that $(\mathbb{S}^{\sim})^{\sim} = \mathbb{S}$. QED

Remark 3.25 It follows from the Proposition above that we could indeed have based the language of the modal μ-calculus on a smaller alphabet of primitive symbols. Given a set D of atomic actions, we could have defined the set of modal fixpoint formulas using the following induction:

$$\varphi ::= \bot \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_d \varphi \mid \mu x.\varphi$$

where p and x are propositional variables, $d \in \mathsf{D}$, and in $\mu x.\varphi$, all free occurrences of x must be positive (that is, under an even number of negation symbols). Here we define $FV(\neg\varphi) = FV(\varphi)$ and $BV(\neg\varphi) = BV(\varphi)$.

In this set-up, the constant $\top$ and the connectives $\wedge$ and $\Box_d$ are defined using the standard abbreviations, while for the greatest fixpoint operator we may put

$$\nu x.\varphi := \neg\mu x.\neg\varphi(\neg x).$$

Note the *triple* use of the negation symbol here, which can be explained by Proposition 3.11 and the observation that we may think of $\neg\varphi(\neg x)$ as the formulas φ^{∂}. ◁

Other immediate consequences

Earlier on we defined the notions of *clean* and *guarded* formulas.

Proposition 3.26 *Every fixpoint formula is equivalent to a clean formula, and hence, to a tidy one.*

Proof. We leave this proof as an exercise for the reader. QED

Proposition 3.27 *Every fixpoint formula is equivalent to a guarded formula.*

Proof.(Sketch) We prove this proposition by formula induction. Clearly the only nontrivial case to consider concerns the fixpoint operators. Consider a formula of the form $\eta x.\delta(x)$, where $\delta(x)$ is guarded and clean, and suppose that x has an unguarded occurrence in δ.

First consider an unguarded occurrence of x in $\delta(x)$ inside a fixpoint subformula, say, of the form $\theta y.\gamma(x, y)$. By induction hypothesis, all occurrences of y in $\gamma(x, y)$ are guarded. Obtain the formula $\overline{\delta}$ from δ by replacing the subformula $\theta y.\gamma(x, y)$ with $\gamma(x, \theta y.\gamma(x, y))$. Then clearly $\overline{\delta}$ is equivalent to δ, and all of the unguarded occurrences of x in $\overline{\delta}$ are outside of the scope of the fixpoint operator θ.

Continuing like this we obtain a formula $\eta x.\overline{\delta}(x)$ which is equivalent to $\eta x.\delta(x)$, and in which none of the unguarded occurrences of x lies inside the scope of a fixpoint operator. That leaves $\wedge$ and $\vee$ as the only operation symbols in the scope of which we may find unguarded occurrences of x.

From now on we only consider the case where $\eta = \mu$, leaving the very similar case where $\eta = \nu$ as an exercise. Clearly, using the laws of classical propositional logic, we may bring the formula $\overline{\delta}$ into conjunctive normal form

$$(x \vee \alpha_1(x)) \wedge \cdots \wedge (x \vee \alpha_n(x)) \wedge \beta(x), \tag{27}$$

where all occurrences of x in $\alpha_1, \ldots, \alpha_n$ and β are guarded. (Note that we may have $\beta = \top$, or $\alpha_i = \bot$ for some i.)

Clearly (27) is equivalent to the formula

$$\delta'(x) := (x \vee \alpha(x)) \wedge \beta(x),$$

where $\alpha = \alpha_1 \wedge \cdots \wedge \alpha_n$. Thus we are done if we can show that

$$\mu x.\delta'(x) \;\equiv\; \mu x.\alpha(x) \wedge \beta(x). \tag{28}$$

Since $\alpha \wedge \beta$ implies δ', it is easy to see (and left for the reader to prove) that $\mu x.\alpha \wedge \beta$ implies $\mu x.\delta'$. For the converse, it suffices to show that $\varphi := \mu x.\alpha(x) \wedge \beta(x)$ is a prefixpoint of $\delta'(x)$. But it is not hard to derive from $\varphi \equiv \alpha(\varphi) \wedge \beta(\varphi)$ that

$$\delta'(\varphi) = (\varphi \vee \alpha(\varphi)) \wedge \beta(\varphi) \equiv ((\alpha(\varphi) \wedge \beta(\varphi)) \vee \alpha(\varphi)) \wedge \beta(\varphi) \equiv \alpha(\varphi) \wedge \beta(\varphi) \equiv \varphi,$$

which shows that φ is in fact a fixpoint, and hence certainly a prefixpoint, of $\delta'(x)$. QED

Combining the proofs of the previous two propositions one easily shows the following.

Proposition 3.28 *Every fixpoint formula is equivalent to a clean, guarded formula, and hence, to a tidy, guarded one.*

Remark 3.29 The equivalences of the above propositions are in fact *effective* in the sense that there are algorithms for computing an equivalent clean and/or guarded equivalent to an arbitrary formula in μML. It is an interesting question what the complexity of these algorithms is, and what the minimum *size* of the equivalent formulas is. We will return to this issue later on, but already mention here that there are formulas that are exponentially smaller than any of their clean equivalents. The analogous question for guarded transformations, i.e., constructions that provide guarded equivalents to an arbitrary formula, is open. $\lhd$

3.5 Adequacy

In this section we prove the *equivalence* of the two semantic approaches towards the modal μ-calculus. Since the algebraic semantics is usually taken to be the more fundamental notion, we refer to this result as the *Adequacy Theorem* stating, informally, that games are an adequate way of working with the algebraic semantics.

▶ `For the time being we only consider the subformula game.`

Theorem 3.30 (Adequacy) *Let ξ be a clean $\mu\mathtt{ML}_{\mathsf{D}}$-formula. Then for all labelled transition systems $\mathbb{S}$ and all states s in $\mathbb{S}$:*

$$s \in [\![\xi]\!]^{\mathbb{S}} \iff (\xi, s) \in \mathrm{Win}_{\exists}(\mathcal{E}(\xi, \mathbb{S})). \tag{29}$$

Proof. The theorem is proved by induction on the complexity of ξ. We only discuss the inductive steps where ξ is of the form $\eta x.\delta$ (with η denoting either μ or ν), leaving the other cases as exercises to the reader.

Preparatory observations Our proof for these inductive cases will involve *three* games: the unfolding game for $\delta_x^{\mathbb{S}}$, and the evaluation games for ξ and δ, respectively. It is based on two key observations: One concerns the nature of the unfolding game for $\delta_x^{\mathbb{S}}$ and its role in the semantics for $\eta x.\delta$; the other observation concerns the similarity between the evaluation games for ξ and for δ.

1. Starting with the first observation, note that by definition of the algebraic semantics of the fixpoint operators, the set $[\![\eta x.\delta]\!]^{\mathbb{S}}$ is the least/greatest fixed point of the map $\delta_x^{\mathbb{S}} : \wp(S) \to \wp(S)$, and that by our earlier Theorem 3.14 on unfolding games, we have

$$[\![\eta x.\delta]\!]^{\mathbb{S}} = \mathrm{Win}_{\exists}(\mathcal{U}^{\eta}(\delta_x^{\mathbb{S}})) \cap S. \tag{30}$$

   Hence, in order to prove (29), it suffices to show that, for any state s_0:

$$s_0 \in \mathrm{Win}_{\exists}(\mathcal{U}^{\eta}(\delta_x^{\mathbb{S}})) \iff (\xi, s_0) \in \mathrm{Win}_{\exists}(\mathcal{E}(\xi, \mathbb{S})). \tag{31}$$

   In other words, the crucial tasks in the proof of this inductive step concern the transformation of a winning strategy for $\exists$ in the unfolding game $\mathcal{U}^{\eta}(\delta_x^{\mathbb{S}})@s_0$ to a winning strategy for her in the evaluation game $\mathcal{E}(\xi, \mathbb{S})@(\xi, s_0)$, and vice versa.

   Given the importance of the unfolding game for $\delta_x^{\mathbb{S}}$ then, let us look at it in a bit more detail. Note that a round of this game, starting at position $s \in S$, consists of $\exists$ picking a subset $A \subseteq S$ that is subject to the constraint that $s \in \delta_x^{\mathbb{S}}(A) = [\![\delta]\!]^{\mathbb{S}[x \mapsto A]}$. But here the inductive hypothesis comes into play: it implies that, for all $A \subseteq S$, we have

$$s \in \delta_x^{\mathbb{S}}(A) \iff (\delta, s) \in \mathrm{Win}_{\exists}(\mathcal{E}(\delta, \mathbb{S}[x \mapsto A])). \tag{32}$$

   In other words, each round of the unfolding game for the map $\delta_x^{\mathbb{S}}$ crucially involves the evaluation game for the formula δ, played on some x-variant $\mathbb{S}[x \mapsto A]$ of $\mathbb{S}$.

2. This leads us to the comparison between the games $\mathcal{G} := \mathcal{E}(\xi, \mathbb{S})$ and $\mathcal{G}_A := \mathcal{E}(\delta, \mathbb{S}[x \mapsto A])$. The second key observation in the inductive step for the fixpoint operators is that these games are *very* similar indeed. For a start, the positions of the two games are essentially the same. Positions of the form (ξ, t), which exist in the first game but not in the second, are the only exception — but in $\mathcal{G}$, any position (ξ, t) is immediately and automatically succeeded by the position (δ, t) which does exist in the second game. What is important is that the positions for $\exists$ are exactly the same in the two games, and thus we may apply her positional strategies for the one game in the other game as well. The only real difference between the games shows up in the rule concerning positions of the form (x, u). In $\mathcal{G}_A$, x is a *free* variable ($x \in FV(\delta)$), so in a position (x, u) the game is over, the winner being determined by u being a member of A or not. In $\mathcal{G}$ however, x is *bound*, so in position (x, u), the variable x will get unfolded to δ.

Combining these two observations, the key insight in the proof of (31) will be to think of $\mathcal{E}(\xi, \mathbb{S})$ as a variant of the unfolding game $\mathcal{U} := \mathcal{U}^{\eta}(\delta_x^{\mathbb{S}})$ where each round of $\mathcal{U}$ corresponds to a version of the game $\mathcal{G}_T$, with T being the subset of S picked by $\exists$ in $\mathcal{U}$. We are now ready for the details of the proof of (31).

For the direction from left to right of (31), suppose that $\exists$ has a winning strategy in the game $\mathcal{U}$ starting at some position s_0. Without loss of generality (see Exercise 3.7) we may assume that this strategy is *positional*. Thus we may represent it as a map $T : S \to \wp(S)$, where we will write T_s rather than $T(s)$. By the legitimacy of this strategy, for every $s \in \mathrm{Win}_\exists(\mathcal{U})$ it holds that $s \in \delta^{\mathbb{S}}_x(T_s)$. So by the inductive hypothesis (32), for each such s we may assume the existence of a winning strategy f_s for $\exists$ in the game $\mathcal{G}_{T_s}@(\delta, s)$. Given the similarities between the games $\mathcal{G}$ and $\mathcal{G}_{T_s}$ (see the discussion above), this strategy is also applicable in the game $\mathcal{G}@(\delta, s)$, at least, until a new position of the form (x, t) is reached. This suggests the following strategy g for $\exists$ in $\mathcal{G}@(\xi, s_0)$:

1. after the initial automatic move, the position of the match is (δ, s_0); $\exists$ first plays her strategy f_{s_0};
2. each time a position (x, s) is reached, the match automatically moves to position (δ, s), where we distinguish cases:
   (a) if $s \in \mathrm{Win}_\exists(\mathcal{U})$ then $\exists$ continues with f_s;
   (b) if $s \notin \mathrm{Win}_\exists(\mathcal{U})$ then $\exists$ continues with a random strategy.

First we show that this strategy guarantees that whenever a position of the form (x, s) is visited, s belongs to $\mathrm{Win}_\exists(\mathcal{U})$, so that case (b) mentioned above never occurs. The proof is by induction on the number of positions (x, s) that have been visited already. For the inductive step, if s is a winning position for $\exists$ in $\mathcal{U}$, then, as we saw, f_s is a winning strategy for $\exists$ in the game $\mathcal{G}_{T_s}@(\delta, s)$. This means that if a position of the form (x, t) is reached, the variable x must be *true* at t in the model $\mathbb{S}[x \mapsto T_s]$, and so t must belong to the set T_s. But by assumption of the map $T : S \to \wp(S)$ being a winning strategy in $\mathcal{U}$, any element of T_s is again a member of $\mathrm{Win}_\exists(\mathcal{U})$.

In fact we have shown that every unfolding of the variable x in $\mathcal{G}$ marks a new round in the unfolding game $\mathcal{U}$. To see why the strategy g guarantees a win for $\exists$ in $\mathcal{G}@(\xi, s_0)$, consider an arbitrary $\mathcal{G}@(\xi, s_0)$-match π in which $\exists$ plays g. Distinguish cases.

First suppose that x is unfolded only finitely often. Let (x, s) be the last basic position in π where this happens. Given the similarities between the games $\mathcal{G}$ and $\mathcal{G}_{T_s}$, the match from this moment on can be seen as both a g-guided $\mathcal{G}$-match and an f_s-guided $\mathcal{G}_{T_s}$-match. As we saw, f_s is a winning strategy for $\exists$ in the game $\mathcal{G}_{T_s}@(\delta, s)$. But since no further position of the form (x, t) is reached, and $\mathcal{G}$ and $\mathcal{G}_{T_s}$ only differ when it comes to x, this means that π is also a win for $\exists$ in $\mathcal{G}$.

If x is unfolded infinitely often during the match π, then by the fact that $\xi = \eta x.\delta$, it is the *highest* variable that is unfolded infinitely often. We have to distinguish the case where $\eta = \nu$ from that where $\eta = \mu$. In the first case, $\exists$ is the winner of the match π, and we are done. If $\eta = \mu$, however, x is a least fixpoint variable, and so $\exists$ would lose the match π. We therefore have to show that this situation cannot occur. Suppose for contradiction that $s_1, s_2, \ldots$ are the positions where x is unfolded. Then it is easy to verify that the sequence $s_0 T_{s_0} s_1 T_{s_1} \ldots$ constitutes a $\mathcal{U}$-match in which $\exists$ plays her strategy T. But this is not possible, since T was assumed to be a *winning* strategy for $\exists$ in the *least* fixpoint game $\mathcal{U} = \mathcal{U}^\mu(\delta^{\mathbb{S}}_x)$.

For the direction from right to left of (31), we will show how each positional winning strategies f for $\exists$ in $\mathcal{G}$ induces a positional strategy for her in $\mathcal{U}$, and that this strategy U_f is winning for her starting at every position $s \in W := \{s \in S \mid (\xi, s) \in \mathrm{Win}_{\exists}(\mathcal{G})\}$.

So fix a positional winning strategy f for $\exists$ in $\mathcal{G}$; that is, $\exists$ is guaranteed to win any f-guided match starting at a position $(\varphi, t) \in \mathrm{Win}_{\exists}(\mathcal{G})$. Observe that, as discussed above, we may and will treat f as a positional strategy in each of the games $\mathcal{G}_A$ as well.

Given a state $s \in W$, we let $\mathbb{T}_f(s)$ be the *strategy tree* induced by f in $\mathcal{G}_A@(\delta, s)$, where A is some arbitrary subset of S. That is, the nodes of $\mathbb{T}_f$ consist of all f-guided finite matches in $\mathcal{G}_A$ that start at (δ, s). In more detail, the root of this tree is the single-position match (δ, s); to define the successor relation of $\mathbb{T}_f$, let Σ be an arbitrary f-guided match starting at position $\mathit{first}(\Sigma) = (\delta, s)$. If $\mathit{last}(\Sigma)$ is a position owned by $\exists$, then Σ will have a single successor in $\mathbb{T}_f$, viz., the unique extension of Σ with the position $f(\Sigma)$ picked by f. On the other hand, if $\mathit{last}(\Sigma)$ is owned by $\forall$, then every possible continuation $\Sigma \cdot b$, where b is an admissible position picked by $\forall$, is a successor of Σ.

We let $U_f(s)$ be the set of states u such that the position (x, u) occurs as the last element $(x, u) = \mathit{last}(\Sigma)$ of some match Σ in $\mathbb{T}_f(s)$. It is easy to see that any $\mathcal{G}_A$-match Σ ending in a position of the form (x, u), is finished immediately, and thus provides a *leaf* of the tree $\mathbb{T}_f$. It is also an easy consequence of the definitions that, whenever $t \in U_f(s)$ for some $s \in W$, then there is an f-guided match $\Sigma_{s,t}$ such that $\mathit{first}(\Sigma_{s,t}) = (\delta, s)$ and $\mathit{last}(\Sigma_{s,t}) = (x, t)$. Note that this match $\Sigma_{s,t}$ can be seen both as a (full) $\mathcal{G}_A$-match and as a (partial) $\mathcal{G}$-match.

Given our definition of a set $U_f(s) \subseteq S$ for every $s \in W$, in effect we have defined a map

$$U_f : W \to \wp(S).$$

Claim 1 Viewing this map U_f as a positional strategy for $\exists$ in $\mathcal{U}$, we claim that in fact it is a *winning* strategy for her in $\mathcal{U}@s_0$.

Proof of Claim We need two auxiliary claims on U_f. First we observe that

$$\text{if } s \in W \text{ then } s \in \delta_x^{\mathbb{S}}(U_f(s)). \tag{33}$$

For a proof of (33), it is obvious from the definition of $U_f(s)$ that f is a positional winning strategy for $\exists$ in $\mathcal{G}_{U_f(s)} = \mathcal{E}(\delta, \mathbb{S}[x \mapsto U_f(s)])$ starting at (δ, s). But then by the inductive hypothesis on δ we obtain that $\mathbb{S}[x \mapsto U_f(s)], s \Vdash \delta$, or, equivalently, $s \in \delta_x^{\mathbb{S}}(U_f(s))$.

Second, we claim that

$$\text{if } s \in W \text{ then } U_f(s) \subseteq W. \tag{34}$$

To see this, first note that if $s \in W$ then by definition $(\xi, s) \in \mathrm{Win}_{\exists}(\mathcal{G})$; but from this it is immediate that $(\delta, s) \in \mathrm{Win}_{\exists}(\mathcal{G})$, and since we assumed f to be a positional winning strategy for $\exists$ in $\mathcal{G}$, it follows by definition of $U_f(s)$ that for every $u \in U_f(s)$ the position (x, u) is winning for $\exists$ in $\mathrm{Win}_{\exists}(\mathcal{G})$. But from this it is easy to derive that both (δ, u) and (ξ, u) are winning position for $\exists$ in $\mathcal{G}$ as well. The latter fact then shows that $u \in W$ and since u was an arbitrary element of $U_f(s)$, (34) follows.

We can now prove that U_f is a winning strategy for $\exists$ in $\mathcal{U}@s_0$. First of all, it follows from (33) that $U_f(s)$ is a legitimate move in $\mathcal{U}$ for every position $s \in W$. From this and (34) we may conclude that $\exists$ never gets stuck in an U_f-guided $\mathcal{U}$-match starting at s_0; that is, she wins every *finite* U_f-guided $\mathcal{U}$-match. In case $\eta = \nu$ this suffices, since in $UG^{\nu}(\delta_x^{\mathbb{S}})$ all infinite matches are won by $\exists$.

Where $\eta = \mu$ we have a bit more work to do, since in this case all infinite matches of $\mathcal{U}^{\mu}(\delta_x^{\mathbb{S}})$ are won by $\forall$. Suppose for contradiction that $\Sigma = s_0 U_f(s_0) s_1 U_f(s_1) \cdots$ would be an infinite U_f-guided match of $\mathcal{U}^{\mu}(\delta_x^{\mathbb{S}})$. Then for every $i \in \omega$ we have that $s_{i+1} \in U_f(s_i)$, so that there is a partial f-guided match $\Sigma_i = \Sigma_{s_i s_{i+1}}$ with $\mathit{first}(\Sigma_i) = (\delta, s_i)$ and $\mathit{last}(\Sigma_i) = (x, s_{i+1})$. But then it is straightforward to verify that the infinite match $\Sigma_{\mathcal{G}} := \Sigma_0 \cdot \Sigma_1 \cdot \Sigma_2 \cdots$ we obtain by concatenating the individual f-guided matches Σ_i, constitutes an infinite f-guided $\mathcal{G}$-match with $\mathit{first}(\Sigma_{\mathcal{G}}) = \mathit{first}(\Sigma_0) = (\xi, s_0)$. Since the highest fixpoint variable unfolded infinitely often during $\Sigma_{\mathcal{G}}$ obviously would be x, this match would be lost by $\exists$. Here we arrive at the desired contradiction, since $(\xi, s_0) \in \mathrm{Win}_{\exists}(\mathcal{G})$, and f was assumed to be a positional winning strategy in $\mathcal{G}$. ◀

QED

Convention 3.31 In the sequel we will use the Adequacy Theorem without further notice. Also, we will write $\mathbb{S}, s \Vdash \varphi$ in case $s \in [\![\varphi]\!]^{\mathbb{S}}$, or, equivalently, $\mathbb{S}, s \Vdash_g \varphi$.

▶ `Adequacy of the closure game to be discussed and proved.`

Notes

What we now call the Knaster-Tarski Theorem (Theorem 3.4) was first proved by Knaster [14] in the context of power set algebras, and subsequently generalized by Tarski [27] to the setting of complete lattices. The Bekič principle (Proposition 3.15) stems from an unpublished technical report.

▶ `more notes and references to be supplied`

As far as we know, the results in section 3.2 on the duality between the least and the greatest fixpoint of a monotone map on a complete boolean algebra, are folklore. The characterization of least and greatest fixpoints in game-theoretic terms is fairly standard in the theory of (co-)inductive definitions, see for instance Aczel [1]. The equivalence of the algebraic and the game-theoretic semantics of the modal μ-calculus (here formulated as the Adequacy Theorem 3.30) was first established by Emerson & Jutla [11].

Exercises

Exercise 3.1 Prove Proposition 3.6: show that monotone maps on complete lattices are inductive.

Exercise 3.2 Prove Theorem 3.21.
(Hint: given complete lattices $\mathbb{C}$ and $\mathbb{D}$, and a monotone map $f : C \times D \to C$, show that the map $g : D \to C$ given by

$$g(d) := \mu x. f(x, d)$$

is monotone. Here $\mu x.f(x,d)$ is the least fixpoint of the map $f_d : C \to C$ given by $f_d(c) = f(c,d)$.)

Exercise 3.3 Let $F : \wp(S) \to \wp(S)$ be some monotone map. A collection $\mathcal{D} \in \wp\wp(S)$ of subsets of S is *directed* if for every two sets $D_0, D_1 \in \mathcal{D}$, there is a set $D \in \mathcal{D}$ with $D_i \subseteq D$ for $i = 0, 1$. Call F *(Scott) continuous* if it preserves directed unions, that is, if $F(\bigcup \mathcal{D}) = \bigcup_{D \in \mathcal{D}} F(D)$ for every directed $\mathcal{D}$.
Prove the following:

(a) F is Scott continuous iff for all $X \subseteq S$: $F(X) = \bigcup \{F(Y) \mid Y \subseteq_\omega X\}$.
(Here $Y \subseteq_\omega X$ means that Y is a finite subset of X.)

(b) If F is Scott continuous then the unfolding ordinal of F is at most ω.

(c) Give an example of a Kripke frame $\mathbb{S} = \langle S, R \rangle$ such that the operation $[R]$ is not continuous.

(d) Give an example of a Kripke frame $\mathbb{S} = \langle S, R \rangle$ such that the operation $[R]$ has closing/unfolding ordinal $\omega + 1$.

Exercise 3.4 By a mutual induction we define, for every finite set P of propositional variables, the fragment $\mu\mathtt{ML}^C_{\mathsf{P}}$ by the following grammar:

$$\varphi ::= p \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond \varphi \mid \mu q.\varphi',$$

where $p \in \mathsf{P}$, $\psi \in \mu\mathtt{ML}$ is a P-free formula, and $\varphi' \in \mu\mathtt{ML}^C_{\mathsf{P}\cup\{q\}}$.

Prove that for every Kripke model $\mathbb{S}$, every formula $\varphi \in \mu\mathtt{ML}^C_{\mathsf{P}}$, and every proposition letter $p \in \mathsf{P}$, the map $\varphi^{\mathbb{S}}_p : \wp(S) \to \wp(S)$ is continuous.

Exercise 3.5 Let $F : \wp(S) \to \wp(S)$ be a monotone operation, and let γ_F be its unfolding ordinal. Sharpen Corollary 3.7 by proving that the cardinality of γ_F is bounded by $|S|$ (rather than by $|\wp(S)|$).

Exercise 3.6 The proof of Theorem 3.14 is based on the characterisation of least fixed points as the intersection of all prefixpoints, and similarly, of greatest fixpoints as the union of all postfixpoints. Can you also prove the theorem using the characterisation of least- and greatest fixpoints via ordinal approximations?

Exercise 3.7 Prove that the unfolding game of Definition 3.12 satisfies *positional determinacy*. That is, let $\mathcal{U}^\mu(F)$ be the least fixpoint unfolding game for some monotone map $F : \wp(S) \to \wp(S)$. Prove the existence of two *positional* strategies $f_\exists : S \to \wp(S)$ and $f_\forall : \wp(S) \to S$ such that for every position p of the game, either $f_\exists$ is a winning strategy for $\exists$ in $\mathcal{U}^\mu(F)@p$, or else $f_\forall$ is a winning strategy for $\forall$ in $\mathcal{U}^\mu(F)@p$.

Exercise 3.8 Let $\mathbb{C}$ be a complete boolean algebra and let $f : C \to C$ be a monotone map. Pick an element $d \in C$ and let $\mu x.f(x)$ be the least fixpoint of f.

(a) Show that $d \wedge \mu x.f(x) = \bot$ iff $d \wedge \mu x.f(x \wedge \neg d) = \bot$, where $\mu x.f(x \wedge \neg d)$ denotes the smallest fixpoint of the map sending any element $x \in C$ to $f(x \wedge \neg d)$.

(b) Conclude that, for any formula of the form $\mu x.\varphi$ and an arbitrary formula γ: the formula $\gamma \wedge \mu x.\varphi$ is satisfiable iff the formula $\gamma \wedge \mu x.\varphi[x \wedge \neg\gamma/x]$ is satisfiable. (A formula φ is called satisfiable if there exists a pointed Kripke model such that $\mathbb{S}, s \Vdash \varphi$.)

- ▶ `add exercise on the closure ordinal of a formula`
- ▶ `add exercise on (complete) additivity`

5 Parity games

A large part of the theory of modal fixpoint logic involves nontrivial concepts and results from the theory of infinite games. In this chapter we discuss some of the highlights of this theory in a fair amount of detail. This allows us to be rather informal about game-theoretic concepts in the rest of the notes.

5.1 Board games

The games that we are dealing with here can be classified as *board* or *graph games*. They are played by two agents, here to be called 0 and 1.

Definition 5.1 If $\sigma \in \{0, 1\}$ is a player, then $\overline{\sigma}$ denotes the *opponent* $1 - \sigma$ of σ. $\lhd$

A board game is played on a *board* or *arena*, which is nothing but a directed graph in which each node is marked with either 0 or 1. A *match* or *play* of the game consists of the two players moving a pebble or token across the board, following the edges of the graph. To regulate this, the collection of graph nodes, usually referred to as *positions* of the game, is partitioned into two sets, one for each player. Thus with each position we may associate a unique player whose turn it is to move when the token lies on position p.

Definition 5.2 A *board* or *arena* is a structure $\mathbb{B} = \langle B_0, B_1, E\rangle$, such that B_0 and B_1 are disjoint sets, and $E \subseteq B^2$, where $B := B_0 \cup B_1$. We will make use of the notation $E[p]$ for the set of *admissible* or *legitimate moves* from a board position $p \in B$, that is, $E[p] := \{q \in B \mid (p, q) \in E\}$. Positions not in $E[p]$ will sometimes be referred to as *illegitimate moves* with respect to p. A position $p \in B$ is a *dead end* if $E[p] = \varnothing$. If $p \in B$, we let σ_p denote the (unique) player such that $p \in B_{\sigma_p}$, and say that p *belongs to* σ_p, or that it is σ_p's *turn* to move at p. $\lhd$

Remark 5.3 Occasionally it will be convenient to represent a board in an alternative yet equivalent manner, viz., as a triple $\mathbb{B} = \langle B, E, \sigma\rangle$ such that (B, E) is a graph and $\sigma : B \to \{0, 1\}$ is a map assigning a player to each position in B. It is obvious how to switch from one presentation to another. $\lhd$

A match of the game may in fact be identified with the sequence of positions visited during play, and thus corresponds to a *path* through the graph. We refer to the Appendix A for some notation concerning paths.

Definition 5.4 A *path* through a board $\mathbb{B} = \langle B_0, B_1, E\rangle$ is a nonempty (finite or infinite) sequence $\pi \in B^\infty$ such that $E\pi_i\pi_{i+1}$ whenever applicable. A *full* or *complete match* or *play* through $\mathbb{B}$ is either an infinite $\mathbb{B}$-path, or a finite $\mathbb{B}$-path π ending with a dead end (i.e. $E[last(\pi)] = \varnothing$).

A *partial match* is a finite path through $\mathbb{B}$ that is not a full match; in other words, the last position of a partial match is not a dead end. We let PM_σ denote the set of partial matches such that σ is the player whose turn it is to move at the last position of the match. In the sequel, we will denote this player as σ_π; that is, $\sigma_\pi := \sigma_{last(\pi)}$. $\lhd$

Each full or completed match is *won* by one of the players, and *lost* by their opponent; that is, there are no draws. A finite match ends if one of the players gets *stuck*, that is, is forced to move the token from a position without successors. Such a finite, completed, match is lost by the player who got stuck.

The importance of this explains the definition of the notion of a *subboard*. Note that any set of positions on a board naturally induces a board of its own, based on the restricted edge relation. We will only call this structure a subboard, however, if there is no disagreement between the two boards when it comes to players being stuck or not.

Definition 5.5 Given a board $\mathbb{B} = \langle B_0, B_1, E\rangle$, a subset $A \subseteq B$ determines the following board $\mathbb{B}_A := \langle A \cap B_0, A \cap B_1, E_{\restriction A}\rangle$, where $E_{\restriction A} := E \cap (A \times A)$ is the *restriction* of E to A. This structure is called a *subboard* of $\mathbb{B}$ if for all $p \in A$ it holds that $E[p] = \varnothing$ iff $E_{\restriction A}[p] = \varnothing$. $\lhd$

If neither player ever gets stuck, an infinite match arises. The flavor of a board game is very much determined by the winning conditions of these infinite matches.

Definition 5.6 Given a board $\mathbb{B}$, a *winning condition* is a map $W : B^\omega \to \{0, 1\}$. An infinite match π is *won* by $W(\pi)$. A *board game* is a structure $\mathcal{G} = \langle B_0, B_1, E, W\rangle$ such that $\langle B_0, B_1, E\rangle$ is a board, and W is a winning condition on B. $\lhd$

Although the winning condition given above applies to all infinite B-sequences, it will only make sense when applied to matches. We have chosen the above definition because it is usually much easier to formulate maps that are defined on all sequences.

Before players can actually start playing a game, they need a starting position. The following definition introduces some terminology and notation.

Definition 5.7 An *initialized board game* is a pair consisting of a board game $\mathcal{G}$ and a position q on the board of the game; such a pair is usually denoted $\mathcal{G}@q$.

Given a (partial) match π, its first element *first*(π) is called the *starting position* of the match. We let $\mathrm{PM}_\sigma(q)$ denote the set of partial matches for σ that start at position q. $\lhd$

Central in the theory of games is the notion of a *strategy*. Roughly, a strategy for a player is a method that the player uses to decide how to continue partial matches when it is their turn to move. More precisely, a strategy is a function mapping partial plays for the player to new positions. It is a matter of definition whether one requires a strategy to always assign moves that are legitimate, or not; here we will not make this requirement.

Definition 5.8 Given a board game $\mathcal{G} = \langle B_0, B_1, E, W\rangle$ and a player σ, a *σ-strategy*, or a *strategy for σ*, is a map $f : \mathrm{PM}_\sigma \to B$. In case we are dealing with an initialized game $\mathcal{G}@q$, then we may take a strategy to be a map $f : \mathrm{PM}_\sigma(q) \to B$. A match π is *consistent with* or *guided by* a σ-strategy f if for any partial match $\pi' \sqsubset \pi$ with *last*$(\pi') \in B_\sigma$, the next position on π (after π') is indeed the element $f(\pi')$.

A σ-strategy f is *surviving* in $\mathcal{G}@q$ if the moves that it prescribes to f-guided partial matches in $\mathrm{PM}_\sigma@q$ are always admissible to σ, and *winning for σ* in $\mathcal{G}@q$ if in addition all

f-guided full matches starting at q are won by σ. A position $q \in B$ is *winning for* σ if σ has a winning strategy for the game $\mathcal{G}@q$; the collection of all winning positions for σ in $\mathcal{G}$ is called the *winning region for* σ *in* $\mathcal{G}$, and denoted as $\mathrm{Win}_\sigma(\mathcal{G})$. $\lhd$

Intuitively, f being a surviving strategy in $\mathcal{G}@q$ means that σ never gets stuck in an f-guided match of $\mathcal{G}@q$, and so guarantees that σ can stay in the game forever.

Convention 5.9 Observe that we allow strategies that prescribe illegitimate moves. In practice, it will often be convenient to extend the definition of a strategy even further to include maps f that are *partial* in the sense that they are only defined on a proper subset of PM_σ. We will only permit ourselves such a sloppiness if we can guarantee that $f(\pi)$ is defined for every $\pi \in \mathrm{PM}_\sigma$ that is consistent with the partial σ-strategy f, so that the situation where the partial strategy actually would fail to suggest a move, will never occur.

It is easy to see that a position in a game $\mathcal{G}$ cannot be winning for *both* players. On the other hand, the question whether a position p is always a winning position for *one* of the players, is a rather subtle one. Observe that in such games the two winning regions *partition* the game board.

Definition 5.10 The game $\mathcal{G}$ on the board $\mathbb{B}$ is *determined* if $\mathrm{Win}_0(\mathcal{G}) \cup \mathrm{Win}_1(\mathcal{G}) = B$; that is, each position is winning for one of the players. $\lhd$

It turns out that the axiom of choice implies the existence of infinite games that admit positions from which neither player has a winning strategy.

► `Add some more detail, including a remark on the axiom of determinacy in set theory.`

In principle, when deciding how to move in a match of a board game, players may use information about the entire history of the match played thus far. However, it will turn out to be advantageous to work with strategies that are simple to compute. Particularly nice are so-called *positional* strategies, which only depend on the current position (i.e., the final position of the partial play). Although their importance is sometimes overrated, positional strategies are convenient to work with, and they will be critically needed in the proofs of some of the most fundamental results in the automata-theoretic approach to fixpoint logic.

Definition 5.11 A strategy f is *positional* or *history-free* if $f(\pi) = f(\pi')$ for any π, π' with $last(\pi) = last(\pi')$. $\lhd$

Convention 5.12 A positional σ-strategy may be represented as a map $f : B_\sigma \to B$.

As a slight generalisation of positional strategies, *finite-memory strategies* can be computed using only a finite amount of information about the history of the match. More details can be found in Exercise 5.2.

5.2 Winning conditions

In case we are dealing with a *finite* board B, then we may nicely formulate winning conditions in terms of the set of positions that occur *infinitely often* in a given match. But in the case of an infinite board, there may be matches in which no position occurs infinitely often (or more than once, for that matter). Nevertheless, we may still define winning conditions in terms of objects that occur infinitely often, if we make use of *finite colorings* of the board. If we assign to each position $b \in B$ a *color*, taken from a finite set C of colors, then we may formulate winning conditions in terms of the *colors* that occur infinitely often in the match.

Definition 5.13 A *coloring* of B is a function $\Gamma : B \to C$ assigning to each position $p \in B$ a *color* $\Gamma(p)$ taken from some finite set C of colors. By putting $\Gamma(p_0 p_1 \cdots) := \Gamma(p_0)\Gamma(p_1)\cdots$ we can naturally extend such a coloring $\Gamma : B \to C$ to a map $\Gamma : B^\omega \to C^\omega$. $\lhd$

Now if $\Gamma : B \to C$ is a coloring, for any infinite sequence $\pi \in B^\omega$, the map $\Gamma \circ \pi \in C^\omega$ forms the associated sequence of colors. But then since C is finite there must be some elements of C that occur infinitely often in this stream.

Definition 5.14 Let $\mathbb{B}$ be a board and $\Gamma : B \to C$ a coloring of B. Given an infinite sequence $\pi \in B^\omega$, we let $\mathit{Inf}_\Gamma(\pi)$ denote the set of colors that occur infinitely often in the sequence $\Gamma \circ \pi$.

A *Muller condition* is a collection $\mathcal{M} \subseteq \wp(C)$ of subsets of C. The corresponding winning condition is defined as the following map $W_\mathcal{M} : B^\omega \to \{0, 1\}$:

$$W_\mathcal{M}(\pi) := \begin{cases} 0 & \text{if } \mathit{Inf}_\Gamma(\pi) \in \mathcal{M} \\ 1 & \text{otherwise.} \end{cases}$$

A *Muller game* is a board game of which the winning conditions are specified by a Muller condition. $\lhd$

In words, player 0 wins an infinite match $\pi = p_0 p_1 \cdots$ if the set of colors one meets infinitely often on this path, belongs to the Muller collection $\mathcal{M}$.

► `Examples to be supplied.`

Muller games have two nice properties. First, they are determined. This follows from a well-known general game-theoretic result, but can also be proved directly. In addition, we may assume that the winning strategies of each player in a Muller game are finite-memory strategies. These results can in fact be generalised to arbitrary *regular games*, that is, board games where the winning condition is given as an ω-regular language over some colouring of the board. We refer to Exercise 5.2) for more details.

These results becomes even nicer if the Muller condition allows a formulation in terms of a *priority map*. In this case, as colors we take natural numbers. Note that by definition of a coloring, the range $\Omega[B]$ of the coloring function Ω is finite. This means that every subset of $\Omega[B]$ has a maximal element. Hence, every match determines a unique natural number, namely, the 'maximal color' that one meets infinitely often during the match. Now a parity winning condition states that the winner of an infinite match is 0 if this number is even, and 1 if it is odd. More succinctly, we formulate the following definition.

Definition 5.15 Let B be some set; a *priority map* on B is a coloring $\Omega : B \to \omega$, that is, a map of finite range. A *parity game* is a board game $\mathcal{G} = \langle B_0, B_1, E, W_\Omega \rangle$ in which the winning condition is given by

$$W_\Omega(\pi) := \max(\mathit{Inf}_\Omega(\pi)) \mod 2.$$

Such a parity game is usually denoted as $\mathcal{G} = \langle B_0, B_1, E, \Omega \rangle$. ◁

The key property that makes parity games so interesting is that they enjoy positional determinacy. We will prove this in section 5.4, but first we turn to a special case, viz., the reachability games.

Before closing this section, however, for future reference we define an important generalisation of *parity games*: the ω-regular ones.

Definition 5.16 An infinite game $\mathcal{G} = \langle B_0, B_1, E, W \rangle$ is called *ω-regular* if there exists an ω-regular language L over some finite alphabet C and a colouring $\Gamma : B \to C$, such that player 0 wins a match $\pi = (p_i)_{i<\omega} \in B^\omega$ precisely if the induced sequence $(\Gamma(p_i))_{i<\omega} \in C^\omega$ belongs to L. ◁

Further on we will see that ω-regular games are closely related to parity games.

5.3 Reachability Games

Reachability games are a special kind of board games. They are played on a board such as described in section 5.1, but now we also choose a subset $A \subseteq B$. The aim of the game is for the one player to move the pebble into A and for the other to avoid this to happen.

Definition 5.17 Fix a board $\mathbb{B}$ and a subset $A \subseteq B$. The reachability game $\mathcal{R}_\sigma(\mathbb{B}, A)$ is then defined as the game over $\mathbb{B}$ in which σ wins as soon as a position in A is reached or if $\overline{\sigma}$ gets stuck. On the other hand, $\overline{\sigma}$ wins if he can manage to keep the token outside of A infinitely long, or if σ gets stuck. ◁

As an example, if $A = \varnothing$, in order to win the game $\mathcal{R}_\sigma(\mathbb{B}, A)$ for player $\overline{\sigma}$ it simply suffices to stay alive forever, while σ can only win by forcing $\overline{\sigma}$ to get stuck.

Remark 5.18 If we want reachability games to fit the format of a board game exactly, we have to modify the board, as follows. Given a reachability game $\mathcal{R}_\sigma(\mathbb{B}, A)$, define the board $\mathbb{B}' := \langle B'_0, B'_1, E' \rangle$ by putting:

$$\begin{array}{lcl} B'_\sigma & := & B_\sigma \setminus A \\ B'_{\overline{\sigma}} & := & B_{\overline{\sigma}} \cup A \\ E' & := & \{(p,q) \in E \mid p \notin A\}. \end{array}$$

In other words, $\mathbb{B}'$ is like $\mathbb{B}$ except that player $\overline{\sigma}$ gets stuck in a position belonging to A. Furthermore, the winning conditions of such a game are very simple: simply define $W : B^\omega \to \{0, 1\}$ as the constant function mapping all infinite matches to $\overline{\sigma}$. This can easily be formulated as a parity condition. ◁

Since reachability games can thus be formulated as very simple parity games, the following theorem, stating that reachability games enjoy positional determinacy, can be seen as a warming up exercise for the general case. We leave the proof of this result as an exercise for the reader.

Theorem 5.19 (Positional determinacy of reachability games) *Let $\mathcal{R}$ be a reachability game. Then there are positional strategies f_0 and f_1 for 0 and 1, respectively, such that for every position q there is a player σ such that f_σ is a winning strategy for σ in $\mathcal{R}@q$.*

Definition 5.20 The winning region for σ in $\mathcal{R}_\sigma(\mathbb{B}, A)$ is called the *attractor set* of σ for A in $\mathbb{B}$, notation: $Attr^{\mathbb{B}}_\sigma(A)$. In the sequel we will fix a positional winning strategy for σ in $\mathcal{R}_\sigma(\mathbb{B}, A)$ and denote it as $attr^{\mathbb{B}}_\sigma(A)$. $\lhd$

Note that σ-attractor sets always contain all points from which σ can make sure that $\overline{\sigma}$ gets stuck. Furthermore, it is easy to see that in $attr_\sigma(A)$-guided matches the pebble never leaves $Attr_\sigma(A)$ (at least if the match starts inside $Attr_\sigma(A)$!).

Proposition 5.21 *$Attr_\sigma$ is a closure operation on $\mathcal{P}(B)$, i.e.*

1. *$A \subseteq A'$ implies $Attr_\sigma(A) \subseteq Attr_\sigma(A')$,*
2. *$A \subseteq Attr_\sigma(A)$,*
3. *$Attr_\sigma(Attr_\sigma(A)) = Attr_\sigma(A)$.*

A kind of counterpart to attractor sets are *traps*. In words, a set A is a σ-trap if σ can't get the pebble out of A, while her opponent has the power to keep it inside A.

Definition 5.22 Given a board $\mathbb{B}$, we call a subset $A \subseteq B$ a *σ-trap* if $E[b] \subseteq A$ for all $b \in A \cap B_\sigma$, while $E[b] \cap A \neq \varnothing$ for all $b \in A \cap B_{\overline{\sigma}}$. $\lhd$

Note that a σ-trap does not contain $\overline{\sigma}$-endpoints and that $\overline{\sigma}$ will therefore never get stuck in a σ-trap. We conclude this section with a useful proposition.

Proposition 5.23 *Let $\mathbb{B}$ be a board and $A \subseteq B$ an arbitrary subset of B. Then the following assertions hold.*

1. *If A is a σ-trap then A is a subboard of B.*
2. *The union $\bigcup\{A_i \mid i \in I\}$ of an arbitrary collection of σ-traps is again a σ-trap.*
3. *If A is a σ-trap then so is $Attr_{\overline{\sigma}}(A)$.*
4. *The complement of $Attr_\sigma(A)$ is a σ-trap.*
5. *If A is a σ-trap in $\mathbb{B}$ then any $C \subseteq A$ is a σ-trap in $\mathbb{B}$ iff C is a σ-trap in $\mathbb{B}_A$.*

Proof. All statements are easily verified and thus the proof is left to the reader. QED

5.4 Positional Determinacy of Parity Games

Theorem 5.24 (Positional Determinacy of Parity Games) *For any parity game $\mathcal{G}$ there are positional strategies f_0 and f_1 for 0 and 1, respectively, such that for every position q there is a player σ such that f_σ is a winning strategy for σ in $\mathcal{G}@q$.*

5.4.1 The finite case

▶ `Details to be supplied`

5.4.2 The general case

To prove positional determinacy for arbitrary parity games, we start with the definition of players' paradises. In words, a subset $A \subseteq B$ is a σ-paradise if σ has a positional strategy f which guarantees her both that she wins the game, and that the token stays in A.

Definition 5.25 Given a parity game $\mathcal{G}(\mathbb{B}, \Omega)$, we call a $\overline{\sigma}$-trap A a *σ-paradise* if there exists a positional winning strategy $f : A \cap B_\sigma \to A$. ◁

The following proposition establishes some basic facts about paradises.

Proposition 5.26 *Let $\mathcal{G}(\mathbb{B}, \Omega)$ be a parity game. Then the following assertions hold:*

1. *The union $\bigcup\{P_i \mid i \in I\}$ of an arbitrary set of σ-paradises is again a σ-paradise.*
2. *There exists a largest σ-paradise.*
3. *If P is a σ-paradise then so is $Attr_\sigma(P)$.*

Proof. The main point of the proof of part (1) is that we somehow have to uniformly choose a strategy on the intersection of paradises, such that we will end up following the strategy of only one paradise. For this purpose, we assume that we have a well-ordering on the index set I (i.e., for the general case we assume the Axiom of Choice).

For the details, assume that $\{P_i \mid i \in I\}$ is a family of paradises, and let f_i be the positional winning strategy for P_i. Note that $P := \bigcup\{P_i \mid i \in I\}$ is a trap for $\overline{\sigma}$ by Proposition 5.23. Assume that $<$ is a well-ordering of I, so that for each $q \in P$ there is a *minimal* index $\min(q)$ such that $q \in P_{\min(q)}$. Define a positional strategy on P by putting

$$f(q) := f_{\min(q)}(q).$$

This strategy ensures at all times that the pebble either stays in the current paradise, or else it moves to a paradise of lower index, and so, any match where σ plays according to f will proceed through a sequence of σ-paradises of decreasing index. Because of the well-ordering, this decreasing sequence of paradises cannot be strictly decreasing, and thus we know that after finitely many steps the pebble will remain in the paradise where it is, say, P_j. From that moment on, the match is continued as an f_j-guided match inside P_j, and since f_j is by assumption a winning strategy when played inside P_j, this match is won by σ.

Part (2) of the proposition should now be obvious: clearly the union of all σ-paradises is the greatest σ-paradise.

In order to prove part (3) we need to show that there exists a winning strategy for σ. The principal idea is to first move to P by $attr_\sigma(P)$ and once there to follow the winning strategy in P. Let f' be the winning strategy for P, we then define the following strategy f on $Attr_\sigma(P)$ by

$$f(p) := \begin{cases} f'(p) & \text{if } p \in P \\ attr_\sigma(P)(p) & \text{otherwise.} \end{cases}$$

A match consistent with this strategy will stay in $Attr_\sigma(P)$ because it is a $\overline{\sigma}$-trap and $f(p) \in Attr_\sigma(P)$ for all $p \in Attr_\sigma(P)$. It is winning because if ever the match arrives at a point $p \in P$ then play continues as if the match were completely in P; and since f' was supposed to be a winning strategy for σ this play is won by σ. However if we start outside P we will at first follow the strategy $attr_\sigma(P)$ which will ensure that σ either wins or that the pebble ends up in P, in which case σ will also win. QED

We are ready to prove the main assertion from which Theorem 5.24 immediately follows.

Proposition 5.27 *The board of a parity game $\mathcal{G}(\mathbb{B}, \Omega)$ can be partitioned into a 0-paradise and a 1-paradise.*

Proof. We will prove this proposition by induction on d, the maximal parity in the game (i.e. $n = \max(\Omega[B])$). If $d = 0$ we are dealing with a reachability game (namely $\mathcal{R}_1(\mathbb{B}, \varnothing)$), and from the results in section 5.3 we may derive that $Attr_1(\varnothing)$ is a 1-paradise and its complement is a 0-paradise. So the proposition holds in case $d = 0$.

Therefore in the remainder we can assume that $d \geq 1$. Let $\sigma := d \mod 2$, that is, σ wins an infinite play π if $\max(Inf(\pi)) = \max(\Omega[B]) = d$. Let $P_{\overline{\sigma}}$ be the maximal $\overline{\sigma}$-paradise, with associated positional strategy f. It now suffices to show that $X := B \setminus P_{\overline{\sigma}}$ is a σ-paradise.

First we shall show that X is a $\overline{\sigma}$-trap. By proposition 5.26(3) it follows that $Attr_{\overline{\sigma}}(P_{\overline{\sigma}})$ is itself also a $\overline{\sigma}$-paradise. By maximality of $P_{\overline{\sigma}}$ and the fact that $Attr_{\overline{\sigma}}$ is a closure operation, it follows that $P_{\overline{\sigma}} = Attr_{\overline{\sigma}}(P_{\overline{\sigma}})$. Thus by Proposition 5.23(4) we see that X, being the complement of a $\overline{\sigma}$-attractor set is a $\overline{\sigma}$-trap.

Consider $\mathcal{G}_X$, the subgame[4] of $\mathcal{G}$ restricted to X. Define $N := \{b \in X \mid \Omega(b) = d\}$ to be the set of all points in X with priority d and let $Z := X \setminus Attr_\sigma^{\mathbb{B}_X}(N)$. Since Z is the complement of a σ-attractor set in $\mathbb{B}_X$ it is a σ-trap in $\mathbb{B}_X$ and hence a σ-trap of $\mathbb{B}$.

By the induction hypothesis we can split the subgame $\mathcal{G}_Z$ into a 0-paradise Z_0 and a 1-paradise Z_1, see the picture. The winning strategies in these paradises we call f_0 and f_1 respectively. (All notions are with regard to the game $\mathcal{G}_Z$.) We want to show that $Z_{\overline{\sigma}} = \varnothing$, so that $Z = Z_\sigma$.

To this aim, we claim that $P_{\overline{\sigma}} \cup Z_{\overline{\sigma}}$ is a $\overline{\sigma}$-paradise in $\mathcal{G}$, and in order to prove this, we consider the following strategy g of $\overline{\sigma}$:

$$g(b) := \begin{cases} f(b) & \text{if } b \in P_{\overline{\sigma}} \\ f_{\overline{\sigma}}(b) & \text{if } b \in Z_{\overline{\sigma}}. \end{cases}$$

[4] For the time being, we take a simple perspective on subgames. Given a parity game $\mathcal{G} = (B_0, B_1, E, \Omega)$, every subset $A \subseteq B$ induces a subgame $\mathcal{G}_A := (B_0 \cap A, B_1 \cap A, E{\restriction_A}, \Omega{\restriction_A})$ where $E{\restriction_A}$ and $\Omega{\restriction_A}$) are simply the restrictions of E and Ω to A.

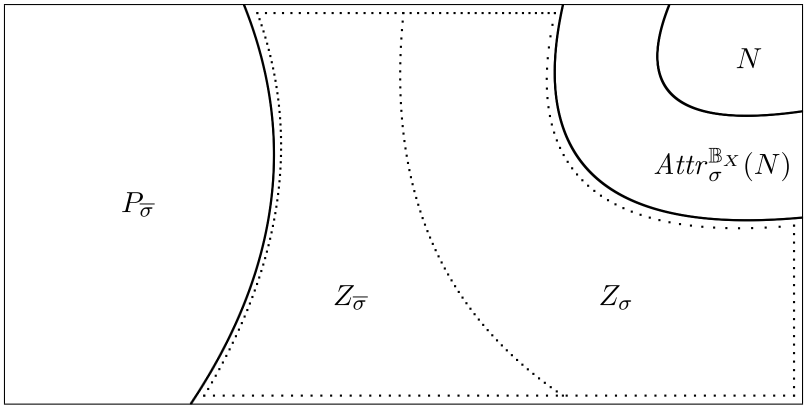


Figure 3: The proof of Proposition 5.27 in a picture

It is left as an exercise for the reader to show that this is indeed a positional winning strategy for $\overline{\sigma}$ in $\mathcal{G}$, and in addition it keeps the pebble inside $P_{\overline{\sigma}} \cup Z_{\overline{\sigma}}$. By the definition of $P_{\overline{\sigma}}$ as the maximal $\overline{\sigma}$-paradise, we see that $P_{\overline{\sigma}} = P_{\overline{\sigma}} \cup Z_{\overline{\sigma}}$ and since $P_{\overline{\sigma}}$ and $Z_{\overline{\sigma}}$ are disjoint we conclude that $Z_{\overline{\sigma}}$ is empty indeed.

This means we can write

$$X = Z_{\sigma} \cup Attr_{\sigma}^{\mathbb{B}_X}(N).$$

We are now almost ready to define the winning strategy for σ which keeps the token inside X. Recall that X is a $\overline{\sigma}$-trap, so that for each $b \in X \cap B_{\sigma}$, we may pick an arbitrary element $k(b) \in E[b] \cap X$. Now define the following strategy h in $\mathcal{G}$ for σ on X.

$$h(b) := \begin{cases} k(b) & \text{if } b \in N \\ attr_{\sigma}(N)(b) & \text{if } b \in Attr_{\sigma}^{\mathbb{B}_X}(N) \setminus N \\ f_{\sigma}(b) & \text{if } b \in Z_{\sigma} = Z. \end{cases}$$

It is left as an exercise for the reader to show that h is indeed a winning strategy for σ in $\mathcal{G}$ and that it keeps the pebble in X. QED

Finally, the assertion made in Theorem 5.24 follows directly from this proposition because by definition of paradises there now exists for every point $b \in B$ a positional winning strategy for the game $\mathcal{G}(\mathbb{B}, \Omega)$.

- ▶ `strategies as 1-player games`
- ▶ `shadow matches?`

5.5 Algorithmic aspects

5.6 Game equivalences and game comparisons

In this section we explore some notions of *equivalence* for board games. In this setting we will frequently represent a board as a triple $\mathbb{B} = \langle B, E, \sigma \rangle$, where $\sigma : B \to \{0, 1\}$ is a map assigning a player to each position in B.

5.6.1 Covers

A very tight link between two games arises if one is a *cover* of the other. Intuitively, $\mathcal{G}$ is a cover of $\mathcal{G}'$ if it is some kind of finitary unravelling of $\mathcal{G}'$.

Definition 5.28 Let $\mathbb{B} = \langle B, E, \sigma \rangle$ and $\mathbb{B}' = \langle B', E', \sigma' \rangle$ be two boards. Then we call a function $f : B \to B'$ a *cover map* for $\mathbb{B}$ and $\mathbb{B}'$ if f is surjective and satisfies the following conditions:

1) f restricts to a bijection between the sets $E[b]$ and $E'[fb]$, for every $b \in B$;

2) f respects ownership: $\sigma'(fb) = \sigma(b)$, for every $b \in B$.

For two board games $\mathcal{G} = \langle B, E, \sigma, W \rangle$ and $\mathcal{G}' = \langle B', E', \sigma', W' \rangle$, we call a function $f : B \to B'$ a *cover map* if f is a cover map for the underlying boards, and in addition satisfies the condition

3) f respects winners: $W(\pi) = W(\pi \circ f)$, for every $\mathcal{G}$-match π. Here we write $\pi \circ f = (fb_n)_{n<\omega}$ for a $\mathcal{G}$-match $\pi = (fb_n)_{n<\omega}$.

If f is a cover map from $\mathcal{G}$ to $\mathcal{G}'$, we write $f : \mathcal{G} \twoheadrightarrow \mathcal{G}'$, say that $\mathcal{G}$ *covers* $\mathcal{G}$ *through* f, and we call $\mathcal{G}$ a *cover* of $\mathcal{G}'$. $\triangleleft$

We gather some basic facts on these concepts, starting with the following observation on boards.

Proposition 5.29 *Let $\mathbb{B} = \langle B, E, \sigma \rangle$ and $\mathbb{B}' = \langle B', E', \sigma' \rangle$ be two boards, and let $f : B \twoheadrightarrow B'$ be a cover map. Then for any path π' in $\mathbb{B}'$ there is a unique path π in $\mathbb{B}$ such that $\pi' = \pi \circ f$.*

- ▶ Turning to games, we may bring the *unravelling* of a game into the picture.
- ▶ Define the unravelling of a game and show that every game is covered by its unravelling.
- ▶ There are in fact good reasons to *identify* a game with any of its covers, since one may show that if $f : \mathcal{G} \twoheadrightarrow \mathcal{G}'$, then $\mathcal{G}$ and $\mathcal{G}'$ have *isomorphic* unravellings.

Proposition 5.30 *Let $f : \mathcal{G} \twoheadrightarrow \mathcal{G}'$ be a cover map. Then for each player $\sigma \in \{0, 1\}$ we have* $\mathrm{Win}_i(\mathcal{G}') = f^{-1}\mathrm{Win}_i(\mathcal{G})]$.

An important example of a cover map is given by the following proposition, which states that every ω-regular game is covered by a parity game. The reader is asked to supply the proof of this Proposition in Exercise 5.2.

Proposition 5.31 (Cover Lemma) *Let $\mathcal{G} = \langle B, E, \sigma, W \rangle$ be an ω-regular game. Then $\mathcal{G}$ is covered by a parity game which is based on the set $B \times M$, where M is the set of states of any deterministic parity automaton recognizing the ω-regular language used to define W.*

Proof. See Exercise 5.2. QED

The following observation is an immediate corollary of the Cover Lemma and Proposition 5.30

Corollary 5.32 *Let $\mathcal{G}$ be an ω-regular game. Then $\mathcal{G}$ is determined.*

For future reference, we need a strengthened version of the Cover Lemma.

Proposition 5.33 (Strengthened Cover Lemma) *Let $\mathcal{G} = \langle B, E, \sigma, W\rangle$ be an ω-regular game, and let $D \subseteq V$ be a set of positions such that every cycle contains at least one position in D, and $E[D] \cap D = \varnothing$. Then $\mathcal{G}$ is covered by a parity game $\mathcal{G}' = \langle B', E', \sigma', \Omega'\rangle$ through a cover map f such that*

1) D 'induces states': $\mathsf{Dom}(\Omega') \subseteq f^{-1}[D]$;

2) E' is 'injective on $f^{-1}[D]$': if $f(u'_i) \in D$ and $(u'_i, v') \in E'$ for $i = 0, 1$, then $u'_0 = u'_1$.

Proof. By the standard cover lemma we may without loss of generality assume that $\mathcal{G}'$ is itself already a parity game, say, with priority map Ω'. We will take care of the conditions 1) and 2) one by one.

For condition 1), define $\mathcal{G}' := \langle V', E', \sigma', \Omega'\rangle$ as follows. First we turn Ω into a total map Ω^* by defining $\Omega^*(u) := \Omega(u)$ if $\Omega(u)$ is defined, and $\Omega^*(u) := -1$ otherwise.

$$
\begin{array}{lcl}
V' & := & V \times \big(\mathsf{Ran}(\Omega) \cup \{-1\}\big) \\
E'[(u,k)] & := & \left\{ \begin{array}{ll} \big\{\big(v, \max(k, \Omega^*(v))\big) \mid v \in E[u]\big\} & \text{if } u \in D \\ \big\{(v, \Omega^*(v)) \mid v \in E[u]\big\} & \text{otherwise} \end{array} \right. \\
\sigma'(u,k) & := & \sigma(u) \\
\Omega(u,k) & := & \left\{ \begin{array}{ll} k & \text{if } u \in D \text{ and } k \geq 0 \\ \text{undefined} & \text{otherwise} \end{array} \right.
\end{array}
$$

Roughly, the intuitions underlying this construction are as follows: a position (u, k) represents a path π in $\mathcal{G}$ with *last*$(\pi) = u$, *first*(π) is the only position on π that belongs to D, and k is a counter that records the highest priority encountered on π (after *first*(π)). The value k is *reset* at a position $u \in D$; that is, if $(v, m) \in E(u, k)$ and $u \in D$ then the value of m is no longer dependent on any value encountered on the path before v, but simply defined as the priority of v.

▶ `Further details to be supplied.`

Now to prove the Proposition it suffices to show that given a parity game $\mathcal{G} = \langle B, E, \sigma, \Omega\rangle$ and a subset D of B such that $\mathsf{Dom}(\Omega) \subseteq D$, we can find a parity game $\mathcal{G}' = \langle B', E', \sigma', \Omega'\rangle$ covering $\mathcal{G}$ through a map f satisfying condition 2) above (i.e., injectivity of E on $f^{-1}[D]$).

For condition 2) (state injectivity) we first define the map $r : V' \to D \uplus \{*\}$ by putting

$$
r(u) := \left\{ \begin{array}{ll} u & \text{if } u \in D \\ * & \text{otherwise.} \end{array} \right.
$$

Now we define

$$
\begin{array}{lcl}
V' & := & ((V \setminus D) \times V) \cup (D \times \{*\}) \\
E'[(u,x)] & := & E[u] \times \{r(u)\}. \\
\sigma'(u,x) & := & \sigma(u) \\
\Omega'(u,x) & := & \begin{cases} \Omega(u) & \text{if } u \in \mathsf{Dom}(\Omega) \\ \text{undefined} & \text{otherwise} \end{cases}
\end{array}
$$

Here the intuition underlying the definition of $E'[(u,v)]$ is the following. If $(v,y) \in E'[(u,x)]$ then we always have $v \in E[u]$; in addition we 'remember' u (in the sense that $y = u$) if $u \in D$, while we 'forget' u (in the sense that $y = *$) if $u \notin D$.

To see that E' is state injective, suppose that we have $(w,x) \in E'[(u_0,v_0)] \cap E'[(u_1,v_1)]$ for $u_0, u_1 \in D$. Then we find $v_0 = v_1 = *$, while $x = u_0 = u_1$, implying that $(u_0,v_0) = (u_1,v_1)$ indeed.

▶ `Further details to be supplied.`

QED

5.6.2 Game bisimulations

Notes

The application of game-theoretic methods in the area of logic and automata theory goes back to work of Büchi. The positional determinacy of parity games was proved independently by Emerson & Jutla [11] and by Mostowski in an unpublished technical report. Our proof of this result is based on Zielonka [29].

Exercises

Exercise 5.1 (positional determinacy of reachability games) Give a direct proof of the positional determinacy of reachability games, that is: prove Theorem 5.19.

Exercise 5.2 (regular games & finite memory strategies) A strategy α for player σ in an infinite game $\mathcal{G} = \langle B_0, B_1, E, W \rangle$ is a *finite memory strategy* if there exists a finite set M, called the *memory set*, an element $m_I \in M$ and a map $(\alpha_1, \alpha_2) : B \times M \to B \times M$ such that for all pairs of sequences $p_0 \cdots p_k \in B^*$ and $m_0 \cdots m_k \in M^*$: if $m_0 = m_I$, $p_0 \cdots p_k \in \mathrm{PM}_\sigma$ and $m_{i+1} = \alpha_2(p_i, m_i)$ (for all $i < k$), then $\alpha(p_0 \cdots p_k) = \alpha_1(p_k, m_k)$.

Now let $\mathcal{G}$ be a regular game.

(a) Define an parity game which covers $\mathcal{G}$, with positions $B \times M$, where M is the carrier of a deterministic parity automaton $\mathbb{M}$ recognizing L.

(b) Show that each player i has a finite memory strategy which is winning for them in $\mathcal{G}@p$ for every $p \in \mathrm{Win}_i$.

Exercise 5.3 (extended cover lemma) Supply the missing details in the proof of the Strengthed Cover Lemma, Proposition 5.33.

7 Tableau games and derivation systems

7.1 The Tableau Game

Introduction

In this section we introduce the *tableau game*: a two-player board game that we will use to investigate whether a given formula (or finite set of formulas), is satisfiable or not. The game will be designed in such a way that a winning strategy for one of the players (Builder, that is) provides a way to construct a model for the formula at stake, while a winning strategy for her opponent (Refuter) can be seen as a *refutation*, that is: a formal proof or derivation showing that the formula is not satisfiable, or equivalently, that its negation is valid.

The tableau game will be closely connected with the *evaluation* game, in that one match of the tableau game corresponds to a *bundle* of matches of the evaluation game. We will be working with the version of the evaluation game that is based on the *closure* of a tidy formula; as a consequence, just as in the closure game, in the tableau game as well a prominent role will be reserved for infinite traces (cf. Definition refd:trace).

▶ `More about this chapter`

7.1.1 The tableau game

As mentioned the tableau game is about the satisfiability of finite sets of formulas, that we shall call *sequents*.

Definition 7.1 A *sequent* is simply a finite set of tidy formulas. We define the following operation on sequents:

$$\begin{array}{lcl} \heartsuit\Sigma & := & \{\heartsuit\varphi \mid \varphi \in \Sigma\} \\ \heartsuit^{-1}\Sigma & := & \{\varphi \mid \heartsuit\varphi \in \Sigma\} \end{array}$$

for every modal operator $\heartsuit$. $\lhd$

Convention 7.2 As is common in proof theory, we will often denote the union operation on sequents simply by a comma instead of using the set-theoretic symbol $\cup$, and we will not write the parentheses in case of a singleton set. For instance, we let Σ, φ denote the sequent $\Sigma \cup \{\varphi\}$.

The *tableau game* $\mathcal{T}$ is a board game with two players: B, or *Builder* (female) and R, or *Refuter* (male). A *position* of this game is either a sequent or a pair consisting of a sequent Σ and a formula φ. Such a pair will be written as $\langle \Sigma, \varphi \rangle$ to distinguish it from the set Σ, φ. For the intuition underlying this game: it is Builder's goal to show that the sequent, which provides the initial position of the game, is *satisfiable* in some pointed Kripke model, while Refuter intends to prove this claim wrong.

The game proceeds in rounds, of two moves each. Both the start and the end of a round consist of some *basic position*, that is, some sequent $\Sigma \subseteq Cl(\Phi)$ (where Φ is the initial position). At a basic position Σ, Refuter is required to pick some formula $\varphi \in \Sigma$; depending on the shape of φ, some rule will be applied to the sequent; the result of which will be a new sequent that provides the next basic position. For instance, if Refuter picks a fixpoint formula then this formula is simply unfolded; if he picks a disjunction then it is up to Builder to pick a disjunct, etc.

The details of the rules and their effects are specified in Table 10, and discussed in Remark 7.7. As usual, we say that a player *gets stuck* if there is no legitimate move available to them; in this case the game is over and the player who got stuck loses the match. For instance, if Φ is *empty* then Refuter will immediately get stuck; this corresponds to our agreement that $\bigwedge \varnothing \equiv \top$: the conjunction of the empty set of formulas is equivalent to the constant $\top$, and hence certainly satisfiable. If none of the players gets stuck the resulting match will be infinite, and we need to check how the *winning conditions* determine a winner of the match.

Position	Player	Admissible moves
Σ	R	$\{\langle \Gamma, \varphi \rangle \mid \Gamma \cup \{\varphi\} = \Sigma\}$
$\langle \Gamma, \bot \rangle$	B	$\varnothing$
$\langle \Gamma, \top \rangle$	$-$	$\{\Gamma\}$
$\langle \Gamma, \ell \rangle$, with $\overline{\ell} \in \Gamma$	B	$\varnothing$
$\langle \Gamma, \ell \rangle$, with $\overline{\ell} \notin \Gamma$	R	$\varnothing$
$\langle \Gamma, \varphi_0 \wedge \varphi_1 \rangle$	$-$	$\{\Gamma \cup \{\varphi_0, \varphi_1\}\}$
$\langle \Gamma, \varphi_0 \vee \varphi_1 \rangle$	B	$\{\Gamma \cup \{\varphi_0\}, \Gamma \cup \{\varphi_1\}\}$
$\langle \Gamma, \eta x\, \psi \rangle$	$-$	$\{\Gamma \cup \{\psi[\eta x\, \psi / x]\}\}$
$\langle \Gamma, \Diamond_d \psi \rangle$	$-$	$\{\{\psi\} \cup \Box_d^{-1} \Gamma\}$
$\langle \Gamma, \Box_d \psi \rangle$	R	$\varnothing$

Table 10: Tableau game

Intuitively, in a game $\mathcal{T}@\Phi$, a winning strategy for Builder should correspond to a model for Φ, whereas a winning strategy for Refuter, which we will refer to as a *refutation*, constitutes a formal *proof* for the unsatisfiability of Φ, and hence, of the validity of the formula $\bigvee \overline{\Phi}$, where $\overline{\Phi} := \{\overline{\varphi} \mid \varphi \in \Phi\}$. In correspondence with this, one may see Table 10 as providing the *proof rules* of some derivation system, with the understanding that it is refuter who determines the order in which these rules are applied.

► `Somewhere: define` $\mathcal{T}(\Phi)$ `as opposed to` $\mathcal{T}@\Phi$`:`
`in` $\mathcal{T}(\Phi)$ `we restrict to the positions that are reachable from` Φ`.`

Definition 7.3 The *tableau game*, denoted as $\mathcal{T}$, is a board game, with players B (or Builder, female) and R (or Refuter, male). Its positions are given by the set

$$\wp_\omega(\mu\mathtt{ML}^t) \cup \{\langle \Gamma, \varphi \rangle \mid \Gamma \in \wp_\omega(\mu\mathtt{ML}^t), \varphi \in \mu\mathtt{ML}^t\},$$

where we recall that $\mu\mathtt{ML}^t$ denotes the set of tidy μ-calculus formulas. Positions of the form $\Sigma \in \wp_\omega(Cl(\mu\mathtt{ML}^t)$ will be called *basic*. The board of the game is given in Table 10, and its winning conditions are given in Definition 7.19 below. ◁

Note that $\mathcal{T}$ is a *global* game, so to say, in the sense that its board consists of the set of *all* sequents of tidy formulas, and *all* sequent/formula pairs of tidy formulas. Nevertheless, our attention will almost exclusively be directed towards initialized games of the form $\mathcal{T}@\Phi$, for some sequent Φ. It is not hard to see that every position that is reachable in $\mathcal{T}$ from Φ will consist of formulas from $Cl(\Phi)$ only.

Before we define the winning conditions we introduce some terminology, and we comment on the dynamics of the game; in particular we discuss the *rules* that we may associate with each position of the form $\langle \Gamma, \varphi \rangle$.

Definition 7.4 Consider a position in $\mathcal{T}$ of the form $\langle \Gamma, \varphi \rangle$. We will refer to φ as the *principal formula* of this position, and to Γ as its *context*; formulas in Γ will be called *side formulas.* We will also call φ an *active* formula of the position $\langle \Gamma, \varphi \rangle$. In case φ is of the form $\Diamond_d \psi$, all formulas of the form $\Box_d \chi$ are called active as well. In all other cases the principal formula φ is the only active formula. $\lhd$

Note that at a basic position Σ, Refuter not only picks a formula $\varphi \in \Sigma$, he also picks a context Γ, and he can either choose $\Gamma = \Sigma$ (so $\varphi \in \Gamma$) or $\Gamma = \Sigma \setminus \{\varphi\}$ (so $\varphi \notin \Gamma$). We need some terminology here.

Definition 7.5 Let Σ be some position in $\mathcal{T}$, and suppose that Refuter picks, as the next position, the pair $\langle \Gamma, \varphi \rangle$. In case $\Gamma = \Sigma$ we call his move *cumulative*; if, on the other hand $\Gamma = \Sigma \setminus \{\varphi\}$, we call it *reductive.* In case Refuter always plays reductively, we say that he follows a *reductive strategy.* $\lhd$

Convention 7.6 For the time being we will always want to restrict Refuter to play reductively. This means that the admissible moves of Refuter at a sequent position Σ are of the form $\langle \Gamma, \varphi \rangle$ with $\Gamma := \Sigma \setminus \{\varphi\}$.

Remark 7.7 In this remark we discuss the various positions of the form $\langle \Gamma, \varphi \rangle$, their owners, and the moves available to these owners.

Case $\varphi = \bot$. In this case the sequent Γ, φ is surely not satisfiable. Accordingly, positions of type $\langle \Gamma, \bot \rangle$ belong to Builder, but since there are no legitimate moves, she will get stuck immediately.

Case $\varphi = \top$. In this case the formula φ has no effect on the satisfiability of the sequent, and so it can be removed. Note that if $\Gamma = \varnothing$, Refuter will get stuck at the next position; this is appropriate, since the singleton $\{\top\}$ is satisfiable.

Case $\varphi = \ell$. Note that this covers both the cases where $\ell = p$ and where $\ell = \overline{p}$, for some proposition letter p; in the first case we have $\overline{\ell} = \overline{p}$ and in the second case, $\overline{\ell} = p$.

Either way we make a further case distinction: if the negation $\overline{\ell}$ of the literal belongs to Γ, the sequent Γ, φ is surely not satisfiable; this position can thus be treated in a similar way as the one where $\varphi = \bot$. On the other hand, if $\overline{\ell}$ does *not* belong to Γ, then there is no way of telling whether Γ, ℓ is satisfiable or not, at least not without further analysis of Γ. Refuter has picked the formula ℓ to soon, as it were, and in order to discourage him from doing so, we designed the game in such a way that he gets stuck in this situation.

Case $\varphi = \varphi_0 \wedge \varphi_1$. Note that the sequent $\Gamma, \varphi_0 \wedge \varphi_1$ is satisfiable iff $\Gamma, \varphi_0, \varphi_1$ is satisfiable; hence if Refuter has picked a conjunction it will immediately be replaced by its conjuncts.

Case $\varphi = \varphi_0 \vee \varphi_1$. In this case we find that $\Gamma, \varphi_0 \vee \varphi_1$ is satisfiable iff at least one of Γ, φ_0 or Γ, φ_1 is satisfiable, and since Builder is the player who aims for showing satisfiability, it is up to her to pick one of these two sequents.

Case $\varphi = \eta x\, \psi$. This case is simple: if a fixpoint formula is principal, then it will simply be unfolded.

Case $\varphi = \Diamond_d \psi$. In this case the match moves to a 'successor state', so to speak, which serves as a witness for the formula ψ. However, at this successor state, then, not only ψ, but also every formula φ such that $\Box_d \varphi \in \Gamma$, must be true.

Case $\varphi = \Box_d \psi$. Picking a box formula constitutes a mistake for Refuter, since box formulas only form a 'real' requirement in tandem with a diamond formula. To discourage Refuter from picking a box formula, we make sure that at a position of the form $\langle \Gamma, \Box_d \psi \rangle$ he loses immediately.

Finally, note that Refuter may pick a diamond formula *at any time*, but that some choices are better than others. In particular, if he picks a diamond formula *too early*, he may loose the possibility of unravelling a least fixpoint formulas that is still 'unpacked' inside a boolean side formula. On the other hand, there are situations in which he cannot wait indefinitely with picking a diamond formula, for instance, if he 'locally' unfolds ν-fixpoints only. $\lhd$

Convention 7.8 In the sequel we will present strategies for Refuter in a proof-theoretic format. That is, a strategy f will be given as a labelled tree, of which the nodes represent the partial f-guided matches. Furthermore, every node t is labelled with the sequent Σ_t representing the final position of the partial match represented by t. Furthermore, at any node t we will underline the formula picked by Refuter's strategy. Assuming that Refuter uses a reductive strategy, the node t thus also reveals the resulting position. In other words, the labelled tree completely determines Refuter's strategy.

Turning to the winning conditions of the tableau game, first we consider the ways in which one of the two players could win or lose a finite match.

To start with, observe that Refuter can force an (almost) immediate win at those sequents that contain $\bot$, or, for some proposition letter p, both p and $\overline{p}$. On the other hand, as we already mentioned, Refuter will get stuck at the empty sequent since there is no principal formula to pick. Another possibility for Refuter to get stuck is at a sequent that consists of atomic and box formulas only, if the propositional part does not contain $\bot$ or a pair $p, \overline{p}$. In such a case Refuter may survive for one more round if the sequent contains the formula $\top$, but after that every principal formula he picks will result in an immediate loss.

Example 7.9 Consider the ML-sequent $\Box\Diamond\overline{p} \vee \Box\bot, \Diamond\Box(\overline{p} \vee q), \Box\Box\overline{q}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\underline{\overline{p}}, p, \overline{q}} \qquad \dfrac{}{\overline{p}, \underline{q}, \overline{q}}}{\overline{p}, \underline{p \vee q)}, \overline{q}}}{\underline{\Diamond\overline{p}}, \Box(p\vee q), \Box\overline{q}}}{\Box\Diamond\overline{p}, \underline{\Diamond\Box(p\vee q)}, \Box\Box\overline{q}} \qquad \dfrac{\dfrac{}{\underline{\bot}, \Box(p\vee q), \Box\overline{q}}}{\Box\bot, \underline{\Diamond\Box(p\vee q)}, \Box\Box\overline{q}}}{\underline{\Box\Diamond\overline{p} \vee \Box\bot}, \Diamond\Box(p\vee q), \Box\Box\overline{q}}$$

The labelled tree above represents a refutation for this sequent, that is, a winning strategy for Refuter. $\lhd$

It is not hard to see that, at least if Refuter plays reductively, all matches of the tableau game for a sequent of *basic* modal formulas will be finite.

7.1.2 Trails and traces

In this subsection we discuss how to assign a winner to an *infinite match* of the tableau game. It will be useful to have a generic notation for an arbitrary match of the tableau game.

Convention 7.10 First of all, note that every match of $\mathcal{T}$ is a path of positions that alternate between sequents and sequent-formula pairs, Observe that the successor of a sequent position of the form Σ is always a sequent formula pair $\langle\Gamma, \chi\rangle$ such that $\Sigma = \Gamma \cup \{\chi\}$, that is, Σ can always be retrieved from Γ and χ. Hence, to denote an infinite match of the form $\pi = \Sigma_0\langle\Gamma_0, \chi_0\rangle\Sigma_1\langle\Gamma_1, \chi_1\rangle\Sigma_2 \cdots$ without loss of information we may write $\pi = (\langle\Gamma_n, \chi_n\rangle)_{n<\omega}$. By a slight abuse of notation we will usually denote this match as $\pi = (\Gamma_n, \chi_n)_{n<\omega}$.

To determine the winner of an infinite match, we need to keep track of the so-called *trails* of formulas. Basically, a trail is a record of the possible development of an individual formula during a match, as determined by the proof rules.

Example 7.11 Consider the sequent

$$\Phi = \{\langle * \rangle(p \vee q), [*]\overline{p}, [*]\overline{q}\},$$

where $\langle * \rangle(p \vee q) = \mu x\,(p \vee q) \vee \Diamond x$, $[*]\overline{p} = \nu y\,\overline{p} \wedge \Box y$, and $[*]\overline{q} = \nu z\,\overline{q} \wedge \Box z$. The labelled tree below represents a strategy for Refuter.

$$\vdots$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p} \wedge \Box[*]\overline{p}, \overline{q} \wedge \Box[*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p} \wedge \Box[*]\overline{p}, [*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), [*]\overline{p}, [*]\overline{q}$$
$$p, \overline{p}, \Box[*]\overline{p}, \overline{q}, \Box[*]\overline{q} \qquad q, \overline{p}, \Box[*]\overline{p}, \overline{q}, \Box[*]\overline{q} \qquad \langle * \rangle(p \vee q), [*]\overline{p}, [*]\overline{q} \quad (\dagger)$$
$$p \vee q, \overline{p}, \Box[*]\overline{p}, \overline{q}, \Box[*]\overline{q} \qquad \Diamond\langle * \rangle(p \vee q), \overline{p}, \Box[*]\overline{p}, \overline{q}, \Box[*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p}, \Box[*]\overline{p}, \overline{q}, \Box[*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p}, \Box[*]\overline{p}, \overline{q} \wedge \Box[*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p} \wedge \Box[*]\overline{p}, \overline{q} \wedge \Box[*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), \overline{p} \wedge \Box[*]\overline{p}, [*]\overline{q}$$
$$(p \vee q) \vee \Diamond\langle * \rangle(p \vee q), [*]\overline{p}, [*]\overline{q}$$
$$\langle * \rangle(p \vee q), [*]\overline{p}, [*]\overline{q}$$

Note that this tree is infinite but regular in the sense that the subtree generated from the node labelled (†) is isomorphic to the tree itself. In the tree we also display the (in this case unique) trail on the infinite branch of the tree which starts at the formula $\langle * \rangle(p \vee q)$ at the root node. ◁

Roughly speaking, we will declare that

> an infinite match of the tableau game is winning for Refuter if it carries a μ-trail.

Clearly then, we need to define this notion of a μ-trail. The following example shows that this definition is somewhat tricky.

Example 7.12 Consider the sequent

$$\Phi := \{\mu x\, x, \nu y\, y\}.$$

First of all, note that since $\mu x\, x \equiv \bot$ and $\nu y\, y \equiv \top$, the sequent Φ is not satisfiable, and so we want Refuter to have a winning strategy in the tableau game. Now consider the following four strategies for Refuter:

$$\begin{array}{cccc} \vdots & \vdots & \vdots & \vdots \\ \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y \\ \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y \\ \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y \\ \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y \\ \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y & \mu x\, x, \nu y\, y \end{array}$$

The difference is that Refuter keeps unfolding $\mu x\, x$ in the leftmost strategy, he keeps unfolding $\nu y\, y$ in the second strategy, he unfolds $\mu x\, x$ twice and then keeps unfolding $\nu y\, y$ in the third strategy, and he alternates between unfolding the two fixpoint formulas in the rightmost strategy. In this example, we shall call the two red sequences μ-trails, since the μ-formula is unfolded infinitely often, whereas the green trails in the middle are *not* μ-trails, since each of them only features finitely many unfoldings of $\mu x\, x$. Hence the first and the fourth strategy are winning for Refuter, the second and the third one are not. $\lhd$

In order to formulate the winning conditions of the tableau game unambiguously, we need a precise definition of the notion of a trail and of its associated trace. For this purpose, consider one round of the tableau game, say, of the form $\Sigma \cdot \langle \Gamma, \chi \rangle \cdot \Sigma'$. With this configuration we associate a *direct trail relation* $\mathsf{T}_{\Gamma,\chi}$ between Σ and Σ'; intuitively, we put a pair (φ, ψ) in this relation if ψ is the 'residu' of φ after the application of the rule associated with the pair (Γ, χ). This situation has two distinct manifestations: one in which φ is an active formula, and one in which φ is a side formula. Roughly, the idea is that the *active* trail relation contains pairs of the form (φ, ψ) where φ is active and ψ is a direct *derivative* of φ, while the *passive* trail relation gathers all pairs (φ, ψ) where φ is a side formula and ψ is *equal* to φ. The trail relation is then simply defined as the union of the active and the passive trail relation.

Example 7.13 To give two simple examples: if $\Sigma = \{\varphi \wedge \psi, \psi, \chi\}$, and Refuter picks the conjunction $\varphi \wedge \psi$ as his principal formula, then the next sequent is $\Sigma' = \{\varphi, \psi, \chi\}$. Now the active trail relation consists of the pairs $(\varphi \wedge \psi, \varphi)$ and $(\varphi \wedge \psi, \psi)$ and the passive trail relation of the pairs (ψ, ψ) and (χ, χ).

If $\Theta = \{\Box\varphi, \Box\psi, \Diamond\varphi, \Diamond\xi, p, \eta x\, \chi\}$ and Refuter picks the formula $\Diamond\varphi$, then the next sequent is $\Theta' = \{\varphi, \psi\}$. The active trail relation consists of the pairs $(\Diamond\varphi, \varphi)$, $(\Box\varphi, \varphi)$ and $(\Box\psi, \psi)$, and the passive trail relation is empty. $\lhd$

Below we spell out the definition in detail; recall that the diagonal relation on a set A is denoted by Id_A.

Definition 7.14 Let Σ, $\langle \Gamma, \varphi \rangle$ and Σ' be positions in the tableau game $\mathcal{T}@\Phi$, and assume that $\langle \Gamma, \varphi \rangle$ is a legitimate move at Σ, and likewise for Σ' at $\langle \Gamma, \varphi \rangle$. We define two relations $\mathsf{A}_{\Gamma,\varphi}, \mathsf{P}_{\Gamma,\varphi} \subseteq \Sigma \times \Sigma'$, by means of the following case distinction:

Case $\varphi = \top$. We define $\mathsf{A}_{\Gamma,\varphi} := \varnothing$ and $\mathsf{P}_{\Gamma,\varphi} := Id_\Gamma$.

Case $\varphi = \varphi_0 \wedge \varphi_1$. We define $\mathsf{A}_{\Gamma,\varphi} := \{(\varphi_0 \wedge \varphi_1, \varphi_0), (\varphi_0 \wedge \varphi_1, \varphi_1)\}$ and $\mathsf{P}_{\Gamma,\varphi} := Id_\Gamma$.

Case $\varphi = \varphi_0 \vee \varphi_1$. We define $\mathsf{A}_{\Gamma,\varphi} := \{(\varphi_0 \vee \varphi_1, \varphi_i)\}$ (depending on Builder's choice) and $\mathsf{P}_{\Gamma,\varphi} := Id_\Gamma$.

Case $\varphi = \eta x\, \psi$. We define $\mathsf{A}_{\Gamma,\varphi} := \{(\eta x\, \psi, \psi[\eta x\, \psi / x])\}$ and $\mathsf{P}_{\Gamma,\varphi} := Id_\Gamma$.

Case $\varphi = \Diamond_d \psi$. We define $\mathsf{A}_{\Gamma,\varphi} := \{(\Diamond_d \psi, \psi)\} \cup \{(\Box_d \chi, \chi) \mid \Box_d \chi \in \Sigma\}$ and $\mathsf{P}_{\Gamma,\varphi} := \varnothing$.

Finally, the *general trail relation* $\mathsf{T}_{\Gamma,\varphi}$ is simply defined as $\mathsf{T}_{\Gamma,\varphi} := \mathsf{A}_{\Gamma,\varphi} \cup \mathsf{P}_{\Gamma,\varphi}$. $\lhd$

Note that in the definition of the trail relation we do not need to consider the cases where the formula φ is a literal, a box formula, or equal to the formula $\bot$, since in these cases the position $\langle \Gamma, \varphi \rangle$ does not have a successor.

Remark 7.15 If Refuter has made a reductive move, resulting in a position $\langle \Gamma, \varphi \rangle$ such that $\varphi \notin \Gamma$, then the relations $\mathsf{A}_{\Gamma,\varphi}$ and $\mathsf{P}_{\Gamma,\varphi}$ are disjoint.

Should we allow cumulative moves, however, then this need not longer be the case. Consider for instance the sequent $\Sigma = \{\mu x\, x, \nu y\, y\}$. If Refuter picks the formula $\mu x\, x$ and cumulatively takes $\Gamma := \Sigma$, then we find that the pair $(\mu x\, x, \mu x\, x)$ belongs to both the active and the passive trail relation. $\lhd$

Definition 7.16 Let $\pi = \Sigma_0(\langle \Gamma_n, \zeta_n \rangle, \Sigma_{n+1})_{n<\kappa}$ be some match of the tableau game. The *trail graph* of π is defined as the pair (V, E) with

$$\begin{array}{lcl} V & := & \{(\varphi, n) \mid 0 \leq n < \kappa,\, \varphi \in \Sigma_n\} \\ E & := & \{\big((\varphi, n), (\psi, n+1)\big) \mid (\varphi, \psi) \in \mathsf{T}_{\Gamma_n,\zeta_n}\}. \end{array}$$

We will often write $\varphi @ n \rightsquigarrow \psi @ n+1$ to denote that $\big((\varphi, n), (\psi, n+1)\big) \in E$. More generally, we will write $\varphi @ n \rightsquigarrow \psi @ m$ if there is a path through the trail graph from (φ, n) to (ψ, m).

A *trail on* π is any sequence $\tau = (\varphi_n)_{n<\omega}$ such that the sequence $(\varphi_n, n)_{n<\omega}$ is a path through the trail graph of π. In case π is infinite, a trail τ on π is called *progressive* if $\tau(n)$ is active infinitely often, that is: $\tau(n) = \zeta_n$ for infinitely many n. $\lhd$

To determine the winner of an infinite match, we are only interested in the active part of its trails. For that purpose we define the notion of a *condensation* of a trail; this is the trace we obtain from the trail by omitting the passive steps.

Definition 7.17 Let $\tau = (\varphi_n)_{n<\kappa}$ be a trail on the match $\pi = \Sigma_0(\langle \Gamma_n, \zeta_n \rangle, \Sigma_{n+1})_{n<\kappa}$ of the tableau game $\mathcal{T}@\Phi$. Then the *condensation* $\widehat{\tau}$ is obtained from τ by omitting all φ_{i+1} from τ for which $(\varphi_i, \varphi_{i+1})$ belongs to the passive trail relation $\mathsf{P}_{v_i,v_{i+1}}$. Any sequence of the form $\widehat{\tau}$ for some trail on π is called a *trace on* π. $\lhd$

Example 7.18 The red trail in the refutation of Example 7.11 is of the form

$$\underline{\varphi}, \psi, \psi, \psi, \psi, \underline{\psi}, \underline{\Diamond\varphi}, \underline{\varphi}, \psi, \psi, \ldots$$

where we abbreviate $\varphi := \langle * \rangle(p \vee q)$ and $\psi := (p \vee q) \vee \Diamond\langle * \rangle(p \vee q)$ (and we indicate whether the formula is active by underlining it). Its condensation is the infinite trace

$$\varphi, \psi, \Diamond\varphi, \varphi, \psi, \Diamond\varphi, \ldots$$

In the four matches of Example 7.12 we get the following four trails of μ-formulas (with active formulas underlined), together with their condensations:

1. $\underline{\mu x\, x}, \underline{\mu x\, x}, \underline{\mu x\, x}, \underline{\mu x\, x}, \underline{\mu x\, x}, \ldots$ $\quad$ $\mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \ldots$
2. $\mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \ldots$ $\quad$ $\mu x\, x$
3. $\underline{\mu x\, x}, \underline{\mu x\, x}, \mu x\, x, \mu x\, x, \mu x\, x, \ldots$ $\quad$ $\mu x\, x, \mu x\, x$
4. $\underline{\mu x\, x}, \mu x\, x, \underline{\mu x\, x}, \mu x\, x, \underline{\mu x\, x}, \ldots$ $\quad$ $\mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \mu x\, x, \ldots$

Clearly only the first and the last trail are progressive and condense into an infinite trace. $\lhd$

It is not difficult to see that condensed trails are *traces*, and that the condensation of a progressive trail is infinite. This observation then provides us with the right tool for defining the winning conditions of the tableau game.

Definition 7.19 Let π be some infinite $\mathcal{T}$-match. For $\eta \in \{\mu, \nu\}$, we define an *η-trail on π* to be a progressive trail on π whose condensation is a η-trace. A infinite $\mathcal{T}$-match π is winning for Refuter if it carries a μ-trail. $\lhd$

Remark 7.20 It will often be convenient to consider a variation $\mathcal{T}^W$ of the tableau game $\mathcal{T}$, the difference lying in the admissible moves for Refuter at sequent positions:

- In $\mathcal{T}^W$, at a sequent position Σ, Refuter may pick any pair $\langle \Gamma, \zeta \rangle$ with $\zeta \not\in \Gamma$ and $\Gamma \cup \{\zeta\} \subseteq \Sigma$. (That is, we relax the condition that $\Gamma \cup \{\zeta\}$ must be equal to Σ.)

In case Refuter, at some position Σ, moves to a position $\langle \Gamma, \zeta \rangle$, we say that he *weakens away* the formulas in $\Sigma \setminus (\Gamma \cup \{\zeta\})$. This adaptation is the game-theoretic analogon of adding a *weakening* rule to a derivation system.

At first sight it may seem that Refuter has more power in $\mathcal{T}^W$ than in $\mathcal{T}$, but in fact the modification does not change the powers of the players. The reader is asked so supply a proof for this statement in Exercise 7.1. $\lhd$

Exercises

Exercise 7.1 (admissibility of weakening)
In this exercise the reader is asked to prove that each player has the same power in the version $\mathcal{T}^W$ of $\mathcal{T}$ in which Refuter is allowed to use some form of *weakening*, cf. Remark 7.20. That is, for every sequent Φ it holds that

$$\Phi \in \mathrm{Win}_R(\mathcal{T}) \text{ iff } \Phi \in \mathrm{Win}_R(\mathcal{T}^W). \tag{49}$$

(a) Prove (49) using the adequacy theorem for tableau games.

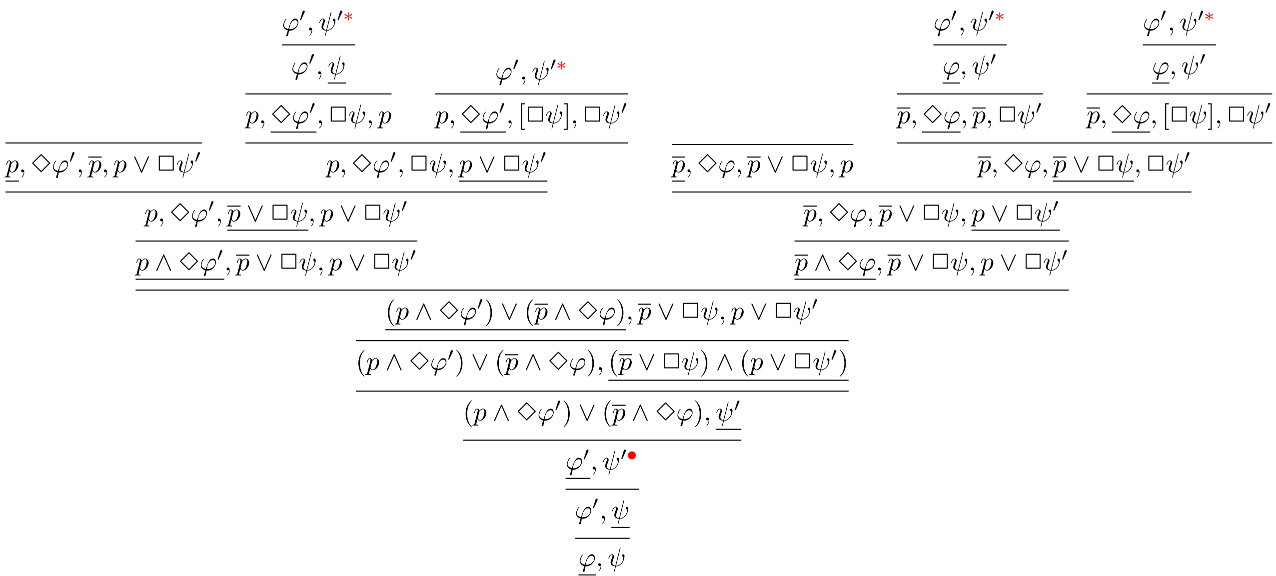
(b) Prove (49) without making reference to the semantics of μ-calculus formulas.

Exercise 7.2 (refutation or not?)
In this exercise we consider the following formulas

$$\begin{array}{lcr}
\varphi & := & \nu y \mu x \big((p \wedge \Diamond x) \vee (\overline{p} \wedge \Diamond y)\big) \\
\varphi' & := & \mu x \big((p \wedge \Diamond x) \vee (\overline{p} \wedge \Diamond \varphi)\big) \\
\psi & := & \nu x \mu y \big((\overline{p} \vee \Box x) \wedge (p \vee \Box y)\big) \\
\psi' & := & \mu y \big((\overline{p} \vee \Box \psi) \wedge (p \vee \Box y)\big)
\end{array}$$

and we note that $\varphi \to_C \varphi' \to_C (p \wedge \Diamond \varphi') \vee (\overline{p} \wedge \Diamond \varphi)$ and $\psi \to_C \psi' \to_C (\overline{p} \vee \Box \psi) \wedge (p \vee \Box \psi')$. Consider the following partial strategy for Refuter in $\mathcal{T}^W @ \{\varphi, \psi\}$.

In this proof tree we indicate the occurrence of weakening by placing the formulas that are weakened away between square brackets. For instance, in the sequent $p, \underline{\Diamond\varphi'}, [\Box\psi], \Box\psi'$ the principle formula is $\Diamond\varphi'$, and the formula $\Box\psi$ is weakened away.



(a) Give, for each of the four leaves labelled $*$, a μ-trace to this leaf starting at the node labelled $\bullet$.

(b) Is the strategy depicted in the proof tree *winning* for Refuter in $\mathcal{T}^W@\{\varphi, \psi\}$?

7.2 Determinacy and adequacy

▶ `introduction to be supplied`

7.2.1 Determinacy

Proposition 7.21 (ω-Regularity) *Let Φ be some μML-sequent. Then the tableau game $\mathcal{T}(\Phi)$ is ω-regular. That is, there is some ω-regular language L such that any match π of $\mathcal{T}(\Phi)$ is won by Builder iff $\pi \in L$.*

▶ `Proof TBS`

Theorem 7.22 (Determinacy) *Let Φ be some μML-sequent. Then the tableau game $\mathcal{T}(\Sigma)$ is determined: either Builder or Refuter has a winning strategy.*

Proof. This is an immediate consequence of Proposition 7.21 and Corollary 5.32. QED

7.2.2 Adequacy

Theorem 7.23 (Adequacy) *Let Φ be some μML-sequent. Then Φ is satisfiable iff Builder has a winning strategy in the tableau game $\mathcal{T}(\Phi)$.*

We will discuss and prove the two directions of the adequacy theorem separately.

Proposition 7.24 (Soundness) *Let Φ be some μML-sequent. If Φ is satisfiable then Builder has a winning strategy in the tableau game $\mathcal{T}@\Phi$.*

Proof. Assume that the μML-sequent Φ is satisfied at the point s_0 of the model $\mathbb{S}$. This means that $(\bigwedge \Phi, s_0)$ is a winning position for $\exists$ in the evaluation game $\mathcal{E} := \mathcal{E}(\bigwedge \Phi, \mathbb{S})@(\bigwedge \Phi, s_0)$. Fix some positional winning strategy f for $\exists$ in this game.

We will use this strategy f to provide Builder with a winning strategy $\widetilde{f}$ in $\mathcal{T}@\Phi$. Next to the definition of $\widetilde{f}$ we will associate, with each $\widetilde{f}$-guided match π, a state $s_\pi \in S$. Intuitively, the role of s_π will be as follows. Let $\pi \in \mathrm{PM}_B$ be some partial match ending at a position of the form $\langle \Gamma, \varphi_0 \vee \varphi_1 \rangle$; that is, Refuter has just picked the disjunction $\varphi_0 \vee \varphi_1$ as his principal formula. Then $\widetilde{f}$ will suggest to Builder to choose the sequent Γ, φ_i, where φ_i is $\exists$'s choice at the position $(\varphi_0 \vee \varphi_1, s_\pi)$. For this to work, and in particular, to make sure that Builder wins all *infinite* $\widetilde{f}$-guided matches, she needs to maintain a rather tight connection between the strategies f and $\widetilde{f}$, to the effect that

(*) every trail on an $\widetilde{f}$-guided $\mathcal{T}$-match corresponds to an f-guided $\mathcal{E}$-match.

More precisely: along with the definition of $\widetilde{f}$ we will inductively define a monotone family ρ_π of functions, where for every $\widetilde{f}$-guided match π that ends at a sequent position, ρ_π maps every trail τ on π to an f-guided $\mathcal{E}$-match $\rho_\pi(\tau)$ such that $\mathit{last}(\rho_\pi(\tau)) = (\mathit{last}(\tau), s_\pi)$ and

$$\widehat{\tau} = (\rho_\pi(\tau))_L,$$

where we recall that, generally, ρ_L denotes the left projection of the match ρ, that is, the trace of formulas determined by ρ, cf. Definition 2.41. Here the monotonicity condition requires that $\pi \sqsubset \pi'$ and $\tau \sqsubset \tau'$ imply $\rho_\pi(\tau) \sqsubset \rho_{\pi'}(\tau')$. Note that it follows from this that $\mathbb{S}, s_\pi \Vdash \varphi$, for every formula φ which belongs to the sequent that constitutes the final position of π.

At the start of the game, the match π consists of the single position Φ; we define $s_\pi := s_0$ and $\rho_\pi(\tau) := (\mathit{last}(\tau), s_0)$, for any trail τ on π. (Note that any such trail consists of a single formula $\mathit{first}(\tau) = \mathit{last}(\tau) \in \Phi$.) With these definitions it is easy to see that the condition (*) is met.

For the inductive step, assume that π is an $\widetilde{f}$-guided match ending at a sequent position, say, $\mathit{last}(\pi) = \Sigma$, and that we have defined the position s_π satisfying (*). We already observed that this implies that $\mathbb{S}, s_\pi \Vdash \Sigma$. Assume that Refuter's move at this position is the pair $\langle \Gamma, \varphi \rangle$, extending π to the match $\pi' = \pi \cdot \langle \Gamma, \varphi \rangle$. We can now simply define $s_{\pi'} := s_\pi$. For the next step we make a case distinction as to the nature of φ.

Case $\varphi = \top$. In this case the next position is Γ, so that the match π' is extended to

$$\pi'' := \pi \cdot \langle \Gamma, \top \rangle \cdot \Gamma.$$

We define $s_{\pi''} := s_\pi$.

In order to check that the condition (*) still holds, consider an arbitrary trail τ on π''. It is easy to see that τ must be of the form $\tau = \sigma \cdot \psi$, where σ is a trail on π,

$last(\sigma) = \psi$ and (ψ, ψ) belongs to the passive trail relation. From this it is immediate that $\widehat{\tau} = \widehat{\sigma}$. Furthermore, by the induction hypothesis we have $\widehat{\sigma} = (\rho_\pi(\sigma))_L$. We define $\rho_{\pi''}(\tau) := \rho_\pi(\sigma)$, and so we obtain $\widehat{\tau} = (\rho_{\pi''}(\tau))_L$ as required.

Case $\varphi = \bot$. Note that, actually, this case cannot occur since $\mathbb{S}, s_\pi \Vdash \Sigma$.

Case $\varphi = \ell$. This case is left as an exercise for the reader.

Case $\varphi = \varphi_0 \vee \varphi_1$. This is the only case where we need to extend the definition of $\widetilde{f}$. Define

$$\widetilde{f}(\pi \cdot \langle \Gamma, \varphi_0 \vee \varphi_1 \rangle) := \Gamma \cup \{\varphi_i\},$$

where φ_i is the formula picked by $\exists$'s winning strategy f at position $(\varphi_0 \vee \varphi_1, s_\pi)$. Note that the latter position is winning for $\exists$ by the inductive hypothesis, so that f provides a legitimate move. In this case the match π' is extended to

$$\pi'' := \pi \cdot \langle \Gamma, \varphi_0 \vee \varphi_1 \rangle \cdot \Gamma \cup \{\varphi_i\}.$$

We define $s_{\pi''} := s_\pi$ and proceed to check the condition (*). For this purpose consider an arbitrary trail τ on π''. There are two subcases to distinguish:

Subcase $\tau = \sigma \cdot \varphi_i$, with $last(\sigma) = \varphi_0 \vee \varphi_1$. By the induction hypothesis we have $\widehat{\sigma} = (\rho_\pi(\sigma))_L$. Now define $\rho_{\pi''}(\tau) := \rho_\pi(\sigma) \cdot (\varphi_i, s_\pi)$, so that we obtain $(\rho_{\pi''}(\tau))_L = (\rho_\pi(\sigma))_L \cdot \varphi_i = \widehat{\sigma} \cdot \varphi_i = \widehat{\sigma \cdot \varphi_i} = \widehat{\tau}$.

Subcase $\tau = \sigma \cdot \psi$, where $last(\sigma) = \psi$ for some idle formula $\psi \neq \varphi_0 \vee \varphi_1$. In this case it is easy to see that $\widehat{\tau} = \widehat{\sigma}$, and we may proceed as in the case where $\varphi = \top$.

Case $\varphi = \varphi_0 \wedge \varphi_1$. In this case the next position in $\mathcal{T}$ is $\Gamma, \varphi_0, \varphi_1$, so that the match π' is extended to

$$\pi'' := \pi \cdot \langle \Gamma, \varphi_0 \wedge \varphi_1 \rangle \cdot \Gamma \cup \{\varphi_0, \varphi_1\}.$$

We define $s_{\pi''} := s_\pi$ and proceed to check the condition (*). For this purpose consider an arbitrary trail τ on π''. As in the previous case, where φ was a disjunction, there are two subcases to consider: an active one where τ is the continuation of a trail σ on π with $last(\sigma) = \varphi$ and $last(\tau) \in \{\varphi_0, \varphi_1\}$, and a passive one where τ is the continuation of a trail σ on φ with the side formula $\psi = last(\sigma)$. In both cases it is straightforward to check that we can update the map ρ in such a way that (*) holds indeed.

Case $\varphi = \eta x\, \psi$. This case is left as an exercise for the reader.

Case $\varphi = \Diamond_d \psi$. In this case the next position in $\mathcal{T}$ is the sequent $\Box_d^{-1}\Sigma \cup \{\psi\}$, and we find

$$\pi'' := \pi \cdot \langle \Gamma, \Diamond_d \psi \rangle \cdot \Box_d^{-1}\Sigma \cup \{\psi\}.$$

Define $s_{\pi''} := t$, where $t \in R_d[s_\pi]$ is $\exists$'s choice at position $(\Diamond_d \psi, s_\pi)$ of the evaluation game as suggested by her positional strategy f — recall that by the induction hypothesis we have $\mathbb{S}, s_\pi \Vdash \Diamond_d \psi$.

To show that the condition (*) holds, we consider an arbitrary trail τ on π''; this trail must be of the form $\sigma \cdot \chi$ for a (unique) π-trail σ, where either $\chi = \psi$ and $last(\sigma) = \Diamond_d \psi$, or $last(\sigma) = \Box_d \chi$. Furthermore, inductively we may assume that $\widehat{\sigma} = (\rho_\pi(\sigma))_L$.

Here we distinguish two subcases:

Subcase $\chi = \psi$ and $last(\sigma) = \Diamond_d \psi$. In this case we have $\tau = \sigma \cdot \psi$. Define $\rho_{\pi''}(\tau) := \rho_\pi(\sigma) \cdot (\psi, t)$, then we find $(\rho_{\pi''}(\tau))_L = (\rho_\pi(\sigma))_L \cdot \psi = \widehat{\sigma} \cdot \psi = \widehat{\sigma \cdot \psi} = \widehat{\tau}$ as required.

Subcase $last(\sigma) = \Box_d \chi$. Now define $\rho_{\pi''}(\tau) := \rho_\pi(\sigma) \cdot (\chi, t)$; that is, the continuation of the $\mathcal{E}$-match $\rho_\pi(\sigma)$ in which $\forall$ at position $(last(\sigma), s_\pi) = (\Box_d \chi, s_\pi)$ picks the successor t of s_π, thus moving the $\mathcal{E}$-match to position $(\chi, t) = (last(\tau), s_{\pi''})$. Here we find $(\rho_{\pi''}(\tau))_L = (\rho_\pi(\sigma))_L \cdot \chi = \widehat{\sigma} \cdot \chi = \widehat{\sigma \cdot \chi} = \widehat{\tau}$, again as required.

Case $\varphi = \Box_d \psi$. In this case Refuter gets stuck and loses immediately.

To see why $\widetilde{f}$ is a *winning* strategy for B in $\mathcal{T}@\Phi$, we consider an arbitrary full $\widetilde{f}$-guided match π. First of all, observe that in all the cases above where the position $\langle \Gamma, \varphi \rangle$ belonged to Builder, we could indeed supply her with some move. Hence, as long as she maintains the condition (*), Builder cannot get stuck. In particular, this means that she wins π in case it is finite.

This leaves the case where π is infinite. Consider an arbitrary progressive trail τ on π, then our goal is to show that $\widehat{\tau}$ is a ν-trace. The point, here, is that τ is the limit of a family of trails, each of which corresponds to an f-guided match of $\mathcal{E}$. But then τ itself also corresponds to an f-guided $\mathcal{E}$-match $\rho_\pi(\tau)$, namely, the limit of the mentioned $\mathcal{E}$-matches — it is here that we need the monotonicity condition. But since f is assumed to be winning strategy for $\exists$ in $\mathcal{E}$, this match $\rho_\pi(\tau)$ is won by $\exists$, which simply means that $\widehat{\tau} = (\rho_\pi(\tau)_L$ is a ν-trace indeed. QED

Turning to the completeness of the tableau game, the intuitions are as follows. Assume that Builder has a winning strategy f in the tableau game $\mathcal{T}@\Phi$, we will use this strategy to construct a model $\mathbb{S}_f$ in which Φ can be satisfied. For the states of this model, we will take f-guided matches — but only the ones in which Refuter plays in a certain, *locally exhaustive* way. To make these intuitions precise we need some definitions.

Definition 7.25 A trace τ is called *local* if it features no transition of the form $\heartsuit\varphi \to_C \varphi$, for any modality $\heartsuit$. A $\mathcal{T}$-match π is called *local* if it features no modal position, that is, no positions of the form $\langle \Gamma, \Diamond_d \psi \rangle$ or $\langle \Gamma, \Box_d \psi \rangle$. $\lhd$

Note that a local trace may *end* at a box- or diamond formula; in fact it may also start with one, but only if it is a one-formula trace. Concerning the relation between local matches and local traces: a match π is local if, and only if, each of its trails condensates to a local trace.

As mentioned, the only strategies of Refuter that we will take into account, when constructing a model $\mathbb{S}_f$ from a winning strategy for Builder, are the locally exhaustive ones. Intuitively, a strategy is locally exhaustive if Refuter makes sure that every boolean or fixpoint formula will become principal, before he is allowed to pick a modal formula.

Definition 7.26 Let Φ be some μML-sequent. A $\mathcal{T}@\Phi$-match π is *locally exhaustive* if it local and satisfies the following conditions:

1. at a sequent position Σ in π, Refuter does not pick an atomic or modal formula if there are still boolean or fixpoint formulas available in Σ;
2. if π is infinite, say, $\pi = (\Gamma_n, \zeta_n)_{n<\omega}$, then for every $k < \omega$ and for every formula $\varphi \in \Sigma_k := \Gamma_n \cup \{\zeta_n\}$ that is either a conjunction, a disjunction or a fixpoint formula, there is some $m \geq k$ such that $\varphi = \zeta_m$.
3. π is maximal with respect to the the above two conditions.

The collection of these matches is denoted as LE.

A strategy g for Refuter is *locally exhaustive* in $\mathcal{T}@\Phi$ if it is reductive, and every local g-guided $\mathcal{T}@\Phi$-match satisfies the first two of the conditions above. $\lhd$

Example 7.27 Of the four strategies in Example 7.12, only the rightmost one is locally exhaustive.

Also note that 'locally exhaustive' does not necessarily mean 'fair', and certainly not 'winning'. For instance, any locally exhaustive strategy operating on the sequent $\Sigma = \{\nu x\, x, \Box p, \Diamond \overline{p}\}$ will keep picking $\nu x\, x$ as the principal formula, and thus cause Refuter to loose the match. He would thus miss the easy win arising from picking the formula $\Diamond \overline{p}$. $\lhd$

Remark 7.28 Locally exhaustive strategy are not hard to find. Refuter can easily arrange one by maintaining, throughout any match of $\mathcal{T}@\Phi$, a priority list of all boolean and fixpoint formulas in $Cl(\Phi)$. At any position during the match π he then picks, as the principal formula, the *first* formula on this list that belongs to Σ; and after this move he updates the list by moving the chosen formula to the end of the list. $\lhd$

Locally exhaustive matches are either infinite or end with a sequent that consists of modal and atomic formulas only. As we will see now, if we can guarantee that these matches are finite, the completeness proof becomes much easier.

Recall that a modal μ-calculus formula ξ is *guarded* if for every subformula of ξ of the form $\eta x.\delta$, x is guarded in δ, that is, every free occurrence of x in δ is in the scope of a modality. The key property of these formulas is that any local trace starting with a guarded formula is *finite*. As a corollary of this, any local $\mathcal{T}$-match starting with a guarded sequent is finite as well, as the following proposition shows.

Proposition 7.29 *Let Φ be some μML-sequent consisting of guarded formulas, and let λ be some local match of $\mathcal{T}@\Phi$ in which Refuter plays reductively. Then λ is finite. In particular, every locally exhaustive match is finite, and ends with a sequent consisting of modal and atomic formulas only.*

Proof. Let $Cl^{loc}(\Phi)$ be the smallest subset of $Cl(\Phi)$ that contains Φ and is closed under taking the direct derivatives of boolean and fixpoint formulas. For any formula $\varphi \in Cl^{loc}(\varphi)$, define $ld(\varphi)$, the *local depth* of φ, as the maximal number of steps from φ to an atomic or modal formula — this is well-defined, precisely by guardedness. Extending this definition to sequents, we put

$$ld(\Phi) := \sum_{\varphi \in \Phi} ld(\varphi).$$

Now let λ be some local match in which Refuter plays reductively. It is then straightforward to see that the local depth of the successive sequents in λ strictly decreases, and from this it immediately follows that λ is finite. QED

Remark 7.30 It also follows from Proposition 7.29 that, in a tableau game that starts from a guarded sequent, every reductive strategy of Refuter that does not pick modal or atomic formulas as principal formulas, is locally exhaustive. $\lhd$

As we will see now, Proposition 7.29 simplifies the construction of a model from a winning strategy for Builder significantly.

Proposition 7.31 (Completeness for guarded formulas) *Let Φ be some μML-sequent consisting of guarded formulas. If Builder has a winning strategy in the tableau game $\mathcal{T}@\Phi$, then Φ is satisfiable.*

Proof. For the time being we restrict the proof to the monomodal case. Assume that B has a winning strategy f in $\mathcal{T}@\Phi$; we will use f to define a Kripke model $\mathbb{S}_f$, and then show that Φ is satisfiable in $\mathbb{S}_f$.

For the definition of $\mathbb{S}_f$ we recall some notation: where $(\pi_i)_{0 \leq i \leq k}$ is a tuple of sequences, we let $\bigodot_{i \leq k} \pi_i$ denote their concatenation, that is: $\bigodot_{i \leq k} \pi_i := \pi_0 \cdot \pi_1 \cdots \pi_k$.

Basically, for the set S_f of states of $\mathbb{S}_f$ we take the collection of f-guided matches where between the modal positions we find locally exhaustive matches. Formally, a *state* of $\mathbb{S}_f$ will be any tuple of the form

$$(\pi_i)_{0 \leq i \leq k},$$

where $k \geq 0$, the sequence $\bigodot_{i \leq k} \pi_i$ is an f-guided match, π_0 is a locally exhaustive match starting at Φ, and for each $i > 0$, π_i is a match of the form $\pi_i = \langle \Gamma_i, \Diamond\varphi_i \rangle \cdot \lambda_i$ with λ_i a locally exhaustive match starting at the position $\Box^{-1}\Gamma_i \cup \{\varphi_i\}$.

Note that by Proposition 7.29 every π_i must be finite (due to guardedness) and end with a sequent position consisting of atomic and modal formula only (due to maximality); we will write $\Sigma_i := last(\pi_i)$ (so $\Sigma_i = last(\lambda_i)$ for $i > 0$). Observe as well that, since $\bigodot_{i \leq k} \pi_i$ must be a well-defined $\mathcal{T}$-match, the latter condition implies that $\Diamond\varphi_i \in \Sigma_{i-1}$ and $\Gamma_i = \Box^{-1}\Sigma_{i-1}$. In a picture we denote the match $\bigodot_{i \leq k} \pi_i$ as follows:

$$\underbrace{\Phi \cdots \cdots \Sigma_0}_{\pi_0} \underbrace{\langle \Gamma_1, \Diamond\varphi_1 \rangle \cdot \underbrace{\Box^{-1}\Gamma_1 \cup \{\varphi_1\} \cdots \cdots \Sigma_1}_{\lambda_1}}_{\pi_1} \quad \cdots \quad \underbrace{\langle \Gamma_k, \Diamond\varphi_k \rangle \cdot \underbrace{\Box^{-1}\Gamma_k \cup \{\varphi_k\} \cdots \cdots \Sigma_k}_{\lambda_k}}_{\pi_k}$$

For the accessibility relation R_f we take

$$R_f := \Big\{ \big((\pi_i)_{i \leq k}, (\pi_i)_{i \leq k+1} \big) \mid (\pi_i)_{i \leq k}, (\pi_i)_{i \leq k+1} \in S_f \Big\},$$

and the valuation V_f is given by

$$V_f(p) := \{(\pi_i)_{i\leq k} \mid p \in \mathit{last}(\pi_k\}.$$

Consider any locally exhaustive match in $\mathcal{T}$ that starts with the position Φ, and let s_0 be the one-item tuple in S_f corresponding with this match. Our goal will be to show that

$$\mathbb{S}_f, s_0 \Vdash \Phi. \tag{50}$$

For the proof of (50), fix some formula $\xi \in \Phi$. We will provide $\exists$ with a winning strategy $\widetilde{f}$ in the evaluation game $\mathcal{E} := \mathcal{E}(\Phi, \mathbb{S}_f)@(\xi, s_0)$. The strategy $\widetilde{f}$ will be defined by induction on the length of a partial $\mathcal{E}$-match, and we will simultaneously prove that $\exists$ can maintain a certain safety condition that we define now.

Consider an arbitrary $\mathcal{E}$-match ρ ending at position (φ, s). Then, focussing on the modal positions in ρ, there is a unique way of writing $\rho = \rho_0 \cdot \ldots \cdot \rho_{k-1} \cdot \rho_k$, such that $\mathit{last}(\rho_0), \ldots, \mathit{last}(\rho_{k-1})$ are the only modal positions on ρ. Similarly, there is a unique way of writing $s = (\pi_i)_{i\leq k'}$ as in the definition of states given above; here for $i > 0$ we will write $\pi_i = \langle \Gamma_i, \Diamond\varphi_i\rangle \cdot \lambda_i$. We call ρ *safe* if $k = k'$ and, for all $i < k$, $(\rho_i)_L$ is a trace on π_i, and $(\rho_k)_L$ is a trace on some initial segment of π_k. (Recall that, given a match ρ of the evaluation game, we write ρ_L to denote the formula part of ρ, that is, where $\rho = (\varphi_n, t_n)_{n<\kappa}$, we have $\rho_L = (\varphi_n)_{n<\kappa}$.)

The key claim in the completeness proof is then the following.

Claim 1 Let ρ be some safe partial match of $\mathcal{E}$. If $\rho \in \mathrm{PM}_\exists$ then $\exists$ has a legitimate move such that the resulting partial match is safe, and if $\rho \notin \mathrm{PM}_\exists$ all possible continuations of ρ are safe.

Proof of Claim Let (φ, s) be the last position of ρ, and let $\rho_0, \ldots, \rho_k$ and $\pi_0, \ldots, \pi_k$ be the respective matches of $\mathcal{E}$ and $\mathcal{T}$ that witness the safety of ρ. By the safety condition, the final part ρ_k of ρ is a trace on some initial segment π'_k of π_k. We make a case distinction as to whether φ is modal or not.

We first consider the case where φ is *not* modal. Since π_k is a locally exhaustive match, we may without loss of generality assume that φ is the principal formula of π'_k, that is, $\mathit{last}(\pi'_k)$ is of the form $\langle \Gamma, \varphi\rangle$ (and π'_k is a *proper* initial segment of π_k). Note that here $\varphi \notin \Gamma$ since we assume that Refuter plays reductively. We make a further case distinction as to the nature of φ.

Case $\varphi = \top$. In this case $\rho \in \mathrm{PM}_\forall$, but since $\forall$ has no legitimate moves, the statement in the claim about all possible continuations of ρ is vacuously true.

Case $\varphi = \bot$. Note that actually this case cannot occur since by assumption the match $\bigodot_{i\leq k} \pi_i$ is f-guided; hence it cannot feature a position of the form $\langle \Gamma, \bot\rangle$ where Builder would get stuck.

Case $\varphi = \varphi_0 \vee \varphi_1$. Builder's strategy f at the partial match $\left(\bigodot_{i<k} \pi_i\right) \odot \pi'_k$ (of which the last position is $\mathit{last}(\pi'_k) = \langle \Gamma, \varphi\rangle$) informs her which one of the disjuncts of φ to pick.

Say that f prescribes to pick the disjunct φ_i, moving in $\mathcal{T}$ to position $\Gamma \cup \{\varphi_i\}$, then in the evaluation game $\exists$ extends the partial match ρ to $\rho^+ := \rho \cdot \varphi_i$.

Now observe that by definition of states, the sequence $\pi'_k \cdot \Gamma, \varphi_i$ is an initial segment of π_k. It is then straightforward to verify that ρ^+ is safe — the main observation is that its final part $(\rho^+_k)_L$ is obviously a trace on $\pi'_k \cdot \Gamma, \varphi_i$.

The remaining non-modal cases are left as exercises for the reader.

We now consider the case where φ *is* modal. Since π_k features no modal positions, any modal formula, once present at some sequent position on π_k, will passively remain present until the final position of π_k is reached. Consequently, we may, without loss of generality, assume that $\pi'_k = \pi_k$. Recall that the final position of π_k must be a sequent position, say, $\Sigma := \mathit{last}(\pi_k)$, and that we have $\varphi \in \Sigma$. We now make a further case distinction as to whether φ is a box- or a diamond formula.

Case $\varphi = \Diamond\psi$. In this case we have to find, in the evaluation game, a successor s^+ of s for $\exists$, and we look for inspiration at the tableau game. That is, suppose that in $\mathcal{T}@\Phi$, at position Σ, Refuter picks $\Diamond\psi$ as the next principal formula, that is, he extends π to the match $\pi \cdot \langle \Gamma, \Diamond\psi\rangle$, where $\Gamma := \Sigma \setminus \{\Diamond\psi\}$. The next position in the tableau game is then fixed as $\Theta := \Box^{-1}\Gamma \cup \{\psi\}$, extending π to $\pi \cdot \langle \Gamma, \Diamond\psi\rangle \cdot \Theta$.

Now let λ_{k+1} some locally exhaustive match starting at position Θ, and such that $\bigodot(\pi_i)_{i \leq k+1}$ is an f-guided match of $\mathcal{T}@\Phi$, where $\pi_{k+1} := \langle \Gamma, \Diamond\psi\rangle \cdot \lambda_{k+1}$. Define $s^+ := (\pi_i)_{i \leq k+1}$, then it is straightforward to verify that $(s, s^+) \in R_f$, and so $\exists$ is allowed to pick s^+ as the required successor of s in $\mathcal{E}$. Furthermore, it is immediate by the definitions that $\rho^+ := \rho \cdot (\psi, s^+)$ is a safe extension of ρ.

Case $\varphi = \Box\psi$. Assume that in $\mathcal{E}$, $\forall$ picks some successor t of s. By definition of R_f, with $s = (\pi_i)_{i \leq k}$, the state t must be of the form $t = (\pi_i)_{i \leq k+1}$, where for some diamond formula $\Diamond\chi \in \Sigma = \mathit{last}(\pi_k)$, we have $\mathit{first}(\pi_{k+1}) = (\Sigma \setminus \{\Diamond\chi\}, \Diamond\chi)$. Write $\pi_{k+1} = (\Sigma \setminus \{\Diamond\chi\}, \Diamond\chi) \cdot \lambda_{k+1}$, then λ_{k+1} is a locally exhaustive match; and writing $\Theta := \mathit{first}(\lambda_{k+1})$, it must be the case that $\Theta = \Box^{-1}(\Sigma \setminus \{\Diamond\chi\}) \cup \{\psi\} = \Box^{-1}\Sigma \cup \{\psi\}$.

But as we already saw, we have $\varphi = \Box\psi \in \Sigma$, so that we find $\psi \in \Theta$. In other words, we have shown that the extension $\rho^+ := \rho \cdot (\psi, t)$ of ρ is safe indeed.

This finishes the proof of the claim. ◀

To see why Claim 1 suffices to prove (50), consider an arbitrary full match ρ of $\mathcal{E}$, where $\exists$ plays the strategy suggested by the claim. If ρ is finite then it is obvious that $\exists$ is the winner, since it is an immediate consequence of the claim that she will not get stuck.

Now consider the case where ρ is infinite. It follows by guardedness that there is a unique way of splitting up ρ as $\rho = \bigodot_{i<\omega} \rho_i$, such that $\mathit{last}(\rho_0), \mathit{last}(\rho_1), \ldots$ are the modal positions on ρ. Furthermore, since $\exists$ maintained the safety conditions throughout the match, with each $k < \omega$ we may associate a position $s_k = (\pi_i)_{i \leq k}$ such that $\bigodot_{i \leq k} \pi_i$ is an f-guided match of $\mathcal{T}@\Phi$, and for all $i < k$, $(\rho_i)_L$ is a trace on π_i. It is then not hard to see that ρ_L is a trace on $\bigodot_{i<\omega} \pi_i$, while the latter match is clearly f-guided. It follows that ρ must be a ν-trace, hence, winning for $\exists$ in $\mathcal{E}$. QED

We now turn to the completeness proof in the general case, that is, where we no longer restrict to guarded formulas. The set-up of the proof is basically the same as in the guarded case: given a sequent Φ for which Builder has a winning strategy in the tableau game, we construct a model $\mathbb{S}_f$ based on Builder's winning strategy f. In fact we would like to define the set S_f of states in exactly the same way as before, but we face the problem that now locally exhaustive matches are no longer necessarily finite. As a consequence, given a tuple $s = (\pi_i)_{i\leq k}$ of such matches, we can no longer concatenate the π_i's, leave alone require that such a concatenation is an f-guided $\mathcal{T}$-match. As a solution to this problem we define, for each infinite match π a finite *representation* $\widetilde{\pi}$. We require this finite representation to be an initial segment of π which is long enough to *cover*, in a sense to be made precise, all finite traces on π. As our states we can then take those tuples $s = (\pi_i)_{i\leq k}$ for which the sequence $\pi_s := (\widetilde{\pi}_i)_{i<k} \cdot \pi_k$ is an f-guided $\mathcal{T}$-match.

To define the operation $\widetilde{\cdot}$ that provides a finite representation of π, we need some preparations. Recall that we write $\tau : \varphi \twoheadrightarrow_C \psi$ if τ is a finite trace such that $\mathit{first}(\tau) = \varphi$ and $\mathit{last}(\tau) = \psi$. In case τ passes a fixpoint formula, we write $\tau : \varphi \twoheadrightarrow^{\xi}_C \psi$, where $\xi = \mathtt{msf}(\tau)$ is the *most significant formula on* τ, and if there is no such fixpoint formula, we write $\tau : \varphi \twoheadrightarrow^{o}_C \psi$.

Definition 7.32 Let τ and τ' be two traces. We say that τ and τ' are *interchangeable*, notation: $\tau \sim \tau'$, if we have both $\tau : \varphi \twoheadrightarrow^{\alpha}_C \psi$ and $\tau' : \varphi \twoheadrightarrow^{\alpha}_C \psi$ for some $\alpha \in \mathit{Cl}(\varphi) \cup \{o\}$. $\lhd$

In words, τ and τ' are interchangeable if $\mathit{first}(\tau) = \mathit{first}(\tau')$ and $\mathit{last}(\tau) = \mathit{last}(\tau')$, and either both τ and τ' pass some fixpoint formula and $\mathtt{msf}(\tau) = \mathtt{msf}(\tau')$, or neither τ nor τ' passes a fixpoint formula. The name given to the interchangeability relation is inspired by the following proposition. We omit its proof, which is a fairly straightforward manipulation of the definitions.

Proposition 7.33 *Let $\tau = \bigodot_{n<\omega} \tau_n$ and $\tau' = \bigodot_{n<\omega} \tau'_n$ be two infinite traces such that $\tau_n \sim \tau'_n$ for all n. Then τ and τ' have the same type, that is, for $\eta \in \{\mu, \nu\}$ it holds that τ is an η-trace iff τ' is an η-trace.*

Obviously, the interchangeability relation, restricted to formulas in the closure of some fixed set Φ, is an equivalence relation of finite index.

Definition 7.34 Let $\lambda = (\Sigma_n \cdot \langle \Gamma_n, \zeta_n \rangle)_{n<\omega}$ be an infinite, locally exhaustive match. Let $\mathtt{MFor}(\lambda)$ and $\mathtt{LFor}(\lambda)$ be, respectively, the sets of modal and literal formulas occurring in some sequent on λ, and let $\mathit{Tr}_{\mathit{lm}}(\lambda)$ be the set of traces on λ that end at a literal or modal formula. For $\tau \in \mathit{Tr}_{\mathit{lm}}(\lambda)$, fix $\widetilde{\tau}$ as some trace of minimal length on λ such that $\widetilde{\tau} \sim \tau$.

Let $\widetilde{\lambda}$ be the shortest initial segment of λ which is long enough to carry every trace in $\{\widetilde{\tau} \mid \tau \in \mathit{Tr}_{\mathit{lm}}(\lambda)\}$ from beginning to end. For any match of the form $\pi = \langle \Gamma, \varphi \rangle \cdot \lambda$, we will write $\widetilde{\pi} := \langle \Gamma, \varphi \rangle \cdot \widetilde{\lambda}$. We shall refer to $\widetilde{\lambda}$ and $\widetilde{\pi}$ as the *finite representation* of λ (respectively, of π). $\lhd$

Arguing for the correctness of this definition, our point is that, due to the relation $\sim$ having finite index, the set $\{\widetilde{\tau} \mid \tau \in \mathit{Tr}_{\mathit{lm}}(\lambda)\}$ is finite. From this it follows that $\widehat{\lambda}$ is well-defined, and in fact, finite.

We are now ready to prove the completeness of the tableau game for arbitrary sequents.

Proposition 7.35 (Completeness (general case)) *Let Φ be some μML-sequent. If Builder has a winning strategy in the tableau game $\mathcal{T}@\Phi$, then Φ is satisfiable.*

Proof. Let Φ be as in the statement of the theorem. As in the guarded case, we will define a model $\mathbb{S}_f = (S_f, R_f, V_f)$ in which we will subsequently show Φ to be satisfiable.

As announced, a *state* of $\mathbb{S}_f$ will be any tuple of the form

$$s = (\pi_i)_{0 \leq i \leq k},$$

provided that $\pi_s := (\bigodot_{i<k} \widetilde{\pi}_i) \cdot \pi_k$ is an f-guided match, π_0 is a locally exhaustive match starting at Φ, and for each $i > 0$, π_i is a match of the form $\pi_i = \langle \Gamma_i, \Diamond\varphi_i \rangle \cdot \lambda_i$ where λ_i is a locally exhaustive match. For the accessibility relation R_f we take

$$R_f := \Big\{ \big((\pi_i)_{i \leq k}, (\pi_i)_{i \leq k+1} \big) \mid (\pi_i)_{i \leq k}, (\pi_i)_{i \leq k+1} \in S_f \Big\},$$

and the valuation V_f is given by

$$V_f(p) := \{ (\pi_i)_{i \leq k} \mid p \in last(\widetilde{\pi}_k \}.$$

Observe that, in the case of guardedness we find that $\widetilde{\lambda} = \lambda$; from this it follows that the two definitions coincide, which justifies our use of the same notation.

As before, let s_0 be the one-item tuple in S_f corresponding with some $\mathcal{T}$-match starting from the position Φ. Our goal will be to show that

$$\mathbb{S}_f, s_0 \Vdash \Phi. \tag{51}$$

As in the guarded case, for the proof of (51), we will provide $\exists$ with a winning strategy $\widetilde{f}$ in the evaluation game $\mathcal{E} := \mathcal{E}(\Phi, \mathbb{S}_f)@(\xi, s_0)$ for some arbitrary but fixed formula $\xi \in \Phi$, and we will show that she can maintain the following notion of safety during any $\widetilde{f}$-guided $\mathcal{E}$-match.

Consider an arbitrary $\mathcal{E}$-match ρ ending at position (φ, s). Then, focussing on the modal positions in ρ, there is a unique way of writing $\rho = \rho_0 \cdot \ldots \cdot \rho_{k-1} \cdot \rho_k$, such that $last(\rho_0), \ldots, last(\rho_{k-1})$ are the only modal positions on ρ. Similarly, there is a unique way of writing $s = (\pi_i)_{i \leq k'}$ as in the definition of states given above; here for $i > 0$ we will write $\pi_i = \langle \Gamma_i, \Diamond\varphi_i \rangle \cdot \lambda_i$. We call ρ *safe* if $k = k'$ and, for all $i < k$, $\widetilde{(\rho_i)_L}$ is a trace on $\widetilde{\pi}_i$, and $(\rho_k)_L$ is a trace on some initial segment of π_k. Note that from this it follows that $\widetilde{(\rho_i)_L} \cdot \ldots \cdot \widetilde{(\rho_{k-1})_L} \cdot (\rho_k)_L$ is a trace on π_s.

The key claim in this version of the completeness proof is the following.

CLAIM 1 Let ρ be some safe partial match of $\mathcal{E}$. If $\rho \in \mathrm{PM}_\exists$ then $\exists$ has an legitimate move such that the resulting partial match is safe, and if $\rho \notin \mathrm{PM}_\exists$ all possible continuations of ρ are safe.

PROOF OF CLAIM The proof of this claim is very similar to that of the corresponding statement in the guarded case. Let (φ, s) be the last position of ρ, and let $\rho_0, \ldots, \rho_k$ and $\pi_0, \ldots, \pi_k$ be the respective matches of $\mathcal{E}$ and $\mathcal{T}$ that witness the safety of ρ.

By the safety condition, the final segment ρ_k of ρ is a trace on some initial segment π'_k of π_k. The key innovative aspect in the current proof is the following. By the definition of a state, the sequence $\pi_s := (\bigodot_{i<k} \widetilde{\pi}_i) \cdot \pi_k$ is a match which has π_k as its tail; moreover, π_s is f-guided, so that we may use it to define $\exists$'s strategy in $\mathcal{E}$.

For a proof of the claim, we make a case distinction as to the nature of φ. We only cover the cases where φ is a disjunction or a diamond formula.

Case $\varphi = \varphi_0 \vee \varphi_1$. Since π_k is a locally exhaustive match, we may without loss of generality assume that φ is the principal formula of π'_k, that is, $last(\pi'_k)$ is of the form $\langle \Gamma, \varphi \rangle$ (and π'_k is a *proper* initial segment of π_k). Builder's strategy f at the partial match π_s (of which the last position is $last(\pi'_k) = \langle \Gamma, \varphi \rangle$) informs her to pick one of the disjuncts of φ, say, φ_i. She now picks the same disjunct in the evaluation game, extending the match to $\rho^+ := \rho \cdot (\varphi_i, s)$. It is completely straightforward to verify that this is indeed a safe continuation of ρ.

In the case where φ is a *modal* formula, the key step is the following. Up to this moment we have been exclusively working with the entire match π_k, and the entire trace ρ_k, being the final part of the $\mathcal{E}$-match ρ. Since we are about to make a modal move, the trace $\tau_k := \widetilde{\rho_k}$, and the finite representative $\widetilde{\pi_k}$ of π_k now come into the picture. Note that by definition of τ_k we have $\tau_k \sim \rho_k$, and that by definition of $\widetilde{\pi_k}$, this τ_k is actually a trace on $\widetilde{\pi_k}$. Furthermore, since $\widetilde{\pi_k}$ is an initial segment of π_k, it is immediate that the sequence $\bigodot_{i\leq k} \widetilde{\pi}_i$, being an initial segment of the f-guided $\mathcal{T}@\Phi$-match $\pi_s = \big(\bigodot_{i<k} \widetilde{\pi}_i\big) \cdot \pi_k$, is itself an f-guided $\mathcal{T}@\Phi$-match as well. Finally, observe that $last(\tau_k) = last(\rho_k) = last(\rho) = \varphi$; so, since φ is modal and τ_k is a trace on $\widetilde{\pi_k}$, this means that we find $\varphi \in \Sigma$, where the sequent Σ is the last position of $\widetilde{\pi_k}$.

Case $\varphi = \Diamond\psi$. In this case we have to find, in the evaluation game, a successor s^+ of s for $\exists$, and we look for inspiration at the tableau game. That is, suppose that in $\mathcal{T}@\Phi$, at position $\Sigma = last(\widetilde{\pi_k})$, Refuter picks $\Diamond\psi$ as the next principal formula, that is, he extends the $\mathcal{T}$-match $\bigodot_{i\leq k} \widetilde{\pi}_i$ to $\big(\bigodot_{i\leq k} \widetilde{\pi}_i\big) \cdot \langle \Gamma, \Diamond\psi \rangle$, where $\Gamma := \Sigma \setminus \{\Diamond\psi\}$. The next position in the tableau game is then fixed as $\Theta := \Box^{-1}\Gamma \cup \{\psi\}$, extending the $\mathcal{T}$-match further to $\big(\bigodot_{i\leq k} \widetilde{\pi}_i\big) \cdot \langle \Gamma, \Diamond\psi \rangle \cdot \Theta$.

Now let λ_{k+1} and $\pi_{k+1} := \langle \Gamma, \Diamond\psi \rangle \cdot \lambda_{k+1}$ be such that λ_{k+1} is some locally exhaustive match starting at position Θ, and $\big(\bigodot_{i\leq k} \widetilde{\pi}_i\big) \cdot \pi_{k+1}$ is an f-guided match of $\mathcal{T}@\Phi$. Define $s^+ := (\pi_i)_{i\leq k+1}$, then it is straightforward to verify that $(s, s^+) \in R_f$, and so $\exists$ is allowed to pick s^+ as the required successor of s in $\mathcal{E}$. Furthermore, it is immediate by the definitions that $\rho^+ := \rho \cdot (\psi, s^+)$ is a safe extension of ρ.

We leave the case where $\varphi = \Box\psi$ as an exercise for the reader. ◀

Let $\widetilde{f}$ be any strategy in $\mathcal{E}$ as suggested by the claim, and let ρ be any $\widetilde{f}$-guided full match of $\mathcal{E}$. In order to show that $\exists$ is the winner of ρ, we distinguish cases. First of all, it is an immediate consequence of the Claim that $\exists$ always has a move available if it is her turn, and so she never gets stuck.

Hence, we may restrict attention to the case where ρ is infinite. For starters, note that every finite initial segment of ρ is safe. We make a further case distinction, as to the number of modal positions in ρ (that is, positions (φ, s) where $\varphi = \heartsuit\psi$ for some modality $\heartsuit$).

► `The case where` ρ `passes only finitely many modal positions is left as an exercise for the reader.`

The interesting case is where ρ passes infinitely many modal positions; in this case there is a unique way of writing $\rho = \bigodot_{k<\omega} \rho_k$, where $(last(\rho_k))_{k<\omega}$ is the sequence of modal positions on ρ. Write $last(\rho_k) = (\varphi_k, s_k)$, then we may define $(\pi_k)_{k<\omega}$ such that $s_k = (\pi_i)_{i\leq k}$ for all k. It easily follows from the definitions of S_f and R_f that the sequence $\bigodot_{k<\omega} \widetilde{\pi_k}$ is in fact an f-guided $\mathcal{T}@\Phi$-match. But since every finite initial segment of ρ is safe, it easily follows that for each k, the trace $\tau_k := \widetilde{(\rho_k)_L}$ is a trace on $\widetilde{\pi_k}$. It then follows that the trace $\tau := \bigodot_{k<\omega} \tau_k$ is a trace on $\bigodot_{k<\omega} \widetilde{\pi_k}$, and hence, a ν-trace. It also follows that $\tau_k \sim (\rho_k)_L$, for each $k < \omega$, and so ρ must also be a ν-trace by Proposition 7.33.

This finishes the proof of (51), and hence, that of the proposition. QED

Exercises

Exercise 7.3 ► `Show that Refuter may always restrict to a reductive strategy. That is, ...`

Exercise 7.4 Show that for guarded formulas, we may fix the order in which Refuter picks formulas. Does this also hold in general?

7.3 Decidability of the satisfiability problem

7.4 A cut-free proof system

7.5 Other derivation systems

10 Model theory of the modal μ-calculus

In this Chapter we will prove some model-theoretic results about the modal μ-calculus.

▶ `overview of chapter to be supplied`

10.1 The cover modality and disjunctive formulas

In the theory of the modal μ-calculus, a fundamental role is played by the so-called *disjunctive formulas*. These are built using the cover modality discussed in Section 1.7, and, as discussed there in the setting of basic modal logic, characterised by a severely restricted use of conjunctions.

▶ `For the time being we confine attention to the monomodal case`

We first introduce the full language of the nabla-based version of the modal μ-calculus. This is simply the extension of the language $\mathtt{ML}_\nabla$ with fixpoint operators. Recall that in this language we work with the *finitary* versions of conjunction and disjunction.

Definition 10.1 The formulas of the language $\mu\mathtt{ML}_\nabla$ are given by the following grammar:

$$\varphi ::= p \mid \overline{p} \mid \bigvee\Phi \mid \bigwedge\Phi \mid \nabla\Phi \mid \mu x\,\varphi \mid \nu x\,\varphi$$

where p and x are propositional variables, $\Phi \subseteq_\omega \mu\mathtt{ML}_\nabla$, and the formation of the formulas $\eta x\,\varphi$ is subject to the proviso that there are no occurrences of the literal $\overline{x}$ in φ. ◁

As in the basic (fixpoint-free) case, the only conjunctions that we allow in a disjunctive formula are of the form $\alpha \bullet \Phi$, which stands for the conjunction $(\bigwedge \alpha) \land \nabla\Phi$. In other words, the basic idea is to define the disjunctive formulas of the modal μ-calculus using the following grammar:

$$\varphi ::= p \mid \overline{p} \mid \bigvee\Phi \mid \alpha \bullet \Phi \mid \mu x\,\varphi \mid \nu x\,\varphi,$$

where, besides the usual positivity constraint on the formation of fixpoint formulas $\eta x\,\varphi$, we also require that no occurrence of x is in the α-part of a subformula $\alpha \bullet \Phi$ of φ.

This is not *exactly* how we define disjunctive formulas, however. It will be convenient to make an a priori distinction between free and bound variables; roughly, the idea is that the free variables can only occur (positively or negatively) among the proposition letters that occur to the left of the bullet conjunctions, while the bound variables can occur anywhere else but not there. It also turns out that we do not need to take literals as disjunctive formulas, see Example 10.3 below.

Definition 10.2 Let P be a finite set of propositional variables. To define the set $\mu\mathtt{DML}(\mathsf{P})$ of *(monomodal) disjunctive formulas in* P we start with the formulas given by the following grammar:

$$\varphi ::= x \mid \bigvee\Phi \mid \alpha \bullet \Phi \mid \mu x\,\varphi \mid \nu x\,\varphi$$

where x is a propositional variable not in P, Φ is a finite set of formulas from this grammar, α is a finite set of literals over P, and the formulas $\mu x\,\varphi$ and $\nu x\,\varphi$ can only be formed if is φ

guarded in x. The set $\mu\mathtt{DML}(\mathsf{P})$ consists of all formulas ξ that meet this pattern and satisfy the condition that $FV(\xi) \subseteq \mathsf{P}$.

We let $\mu\mathtt{DML}$ be the set of formulas that are disjunctive in some set P. $\lhd$

► `Note that disjunctive formulas are tidy and guarded.`

In practice we will often pretend that atomic formulas, and in fact all propositional formulas, are disjunctive. This is justified by the following example.

Example 10.3 As in the basic case, the constant $\bot$ can be seen as an abbreviation of the disjunctive formula $\bigvee \varnothing$. Different from the basic case, however, we can do without the constant $\top$ as a primitive constant either, since the presence of the greatest fixpoint operator enables us to write

$$\top \equiv \nu x\,(\varnothing \bullet \varnothing \vee \varnothing \bullet \{x\}).$$

Literals do not qualify as disjunctive formulas, but any literal ℓ is equivalent to a disjunctive formula as well:

$$\ell \equiv \{\ell\} \bullet \{\top\} \vee \{\ell\} \bullet \varnothing.$$

For this reason we may in practice pretend that atomic formulas, and in fact all propositional formulas, are disjunctive.

Another example of a disjunctive formulas is $\mu x \Big(\{p, \overline{q}\} \bullet \big\{x, \nu y \left(\{p\} \bullet \{x \vee y\}\right)\big\}\Big)$, but not its subformula $\{p, \overline{q}\} \bullet \big\{x, \nu y \left(\{p\} \bullet \{x \vee y\}\right)\big\}$ (since in the latter formula x is free, and hence, it may not occur in the set to the right of either of the bullet conjunctions). Further examples of non-disjunctive formulas are $\mu x\, x$ (unguarded) and $\mu x \Big(\{\overline{p}, \overline{q}\} \bullet \big\{x, \nu y \left(\{p, x\} \bullet \{x, \top\}\right)\big\}\Big)$ (here the subformula $\{p, x\} \bullet \{x, \top\}$ is not admissible since x, being a bound variable, may not occur in the set to the left of the bullet conjunction). $\lhd$

Turning to the semantics of disjunctive formulas, below we introduce the *evaluation game* for this language. For this definition we recall that a relation $Z \subseteq S \times S'$ is *full* on some pair $(U, U') \in \wp(S) \times \wp(S')$ if $U \subseteq \mathsf{Dom}(Z)$ and $U' \subseteq Ran(Z)$, or, in other words, if every $u \in U$ is related by Z to some $u' \in U'$, and vice versa.

Position	Player	Admissible moves
$(\bigvee \Phi, s)$	$\exists$	$\{(\varphi, s) \mid \varphi \in \Phi\}$
$(\alpha \bullet \Phi, s)$	$\forall$	$\{(\bigwedge \alpha, s), (\nabla\Phi, s)\}$
$(\bigwedge \alpha, s)$ with $s \Vdash \bigwedge \alpha$	$\forall$	$\varnothing$
$(\bigwedge \alpha, s)$ with $s \not\Vdash \bigwedge \alpha$	$\exists$	$\varnothing$
$(\nabla\Phi, s)$	$\exists$	$\{Z \subseteq \Phi \times R[s] \mid Z \text{ is full on } \Phi \text{ and } R[s]\}$
$Z \subseteq \mu\mathtt{DML}(\mathsf{P}) \times S$	$\forall$	Z
$(\eta_x x.\delta_x, s)$	–	$\{(\delta_x, s)\}$
(x, s), with $x \in BV(\xi)$	–	$\{(\delta_x, s)\}$

Table 23: Evaluation game for disjunctive formulas (subformula version)

Definition 10.4 The positions and admissible moves of the evaluation game for clean disjunctive formulas are given in Table 23. The winning conditions are as in the evaluation games for arbitrary $\mu\mathtt{ML}$-formulas. $\lhd$

Most of the moves of the evaluation game speak for themselves (given the interpretation of $\alpha \bullet \Phi$ as $(\bigwedge \alpha) \wedge \nabla\Phi$). A minor deviation from the earlier evaluation games is that here we break off a match immedidately if we reach a position of the form $(\bigwedge \alpha, s)$ where α is a set of literals, rather than breaking down the conjunction in subsequent moves.

What makes the evaluation game for disjunctive formulas special is the kind of move that $\exists$ makes at a position of the form $(\nabla\Phi, s)$: here she picks a *relation* $Z \subseteq \mu\mathtt{DML}(\mathsf{P}) \times S$ of witnesses (with the requirement that Z is *full* on Φ and $R[s]$). Such a binary relation Z thus forms a new type of position, which is not a formula-state pair, but rather, a set of such pairs. These relational positions all belong to $\forall$, and his task at a position Z is simply to pick a witness from Z, that is, a pair (ψ, t) in Z. Of course this is in accordance with the semantic meaning of the cover modality.

In the following definition and propositions we isolate the key game-theoretic property of disjunctive formulas. Recall that, for a given strategy f in some evaluation game $\mathcal{E}(\xi, \mathbb{S})$ starting at position (ξ, s), we call a position (φ, t) f-reachable if there is some f-guided match in which the position (φ, t) is reached. We say that the state t is f-reachable if there is some formula φ such that the position (φ, t) is f-reachable.

Definition 10.5 Let ξ be a disjunctive formula, and let $(\mathbb{S}, s)$ be a pointed model.

A strategy f for $\exists$ in the evaluation game $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ is called *separating* if at every f-reachable position of the form $(\nabla\Phi, s)$, f picks a relation $Z \subseteq \Phi \times R[s]$ such that for every $t \in R[s]$ there is exactly one $\varphi \in \Phi$ such that $(\varphi, t) \in Z$.

A strategy f for $\exists$ in $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ is *thin* if for every $t \in S$, if t is f-reachable, then there is at most one formula φ of the form $\alpha \bullet \Phi$ such that (φ, t) is f-reachable.

If f is a separating strategy which is *winning* for $\exists$ in $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ then we say that ξ is *strongly satisfied* in $\mathbb{S}$ at s, notation: $\mathbb{S}, s \Vdash_s \xi$. $\lhd$

The name 'separating' is chosen for obvious reasons: if, at position $(\nabla\Phi, s)$, $\exists$ picks a functional relation Z, she effectively separates the elements of Φ from one another, in the sense that there are no two witnesses $(\varphi, t), (\varphi', t)$ in Z with $\varphi \neq \varphi'$. It is easy to see that separating winning strategies on tree models are thin.

Proposition 10.6 *Let ξ be a disjunctive formula, and let $(\mathbb{S}, s)$ be a tree model. If f is a separating winning strategy for $\exists$ in $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ then f is thin.*

Strong satisfaction is a very strong kind of satisfaction indeed, and in later chapters we will use it as a key model-theoretic tool. The thinness of separating strategies on tree models will turn out to be an extremely useful property. The fundamental model-theoretic property of disjunctive formulas is that without loss of generality we may always assume that winning strategies are separating, provided that we allow ourselves to move to a bisimilar model.

▶ `The proof of this theorem hinges on the semantics of the cover modality.`

Theorem 10.7 *Let ξ be a disjunctive formula, and let $(\mathbb{S}, s)$ be a pointed model. Then the following are equivalent:*

1) $\mathbb{S}, r \Vdash \xi$

2) $\mathbb{S}', r' \Vdash_s \xi$ *for some pointed tree model such that* $\mathbb{S}, r \leftrightarroweq \mathbb{S}', r'$.

Proof. (Sketch) We may focus on the direction from left to right, since the opposite direction is an immediate consequence of bisimulation invariance. So assume that $\mathbb{S}, s \Vdash \xi$, where ξ is disjunctive, and let f be a winning strategy for $\exists$ in the evaluation game $\mathcal{E}$ starting from position (ξ, s). Without loss of generality we may assume that f is positional. Let k be the maximal size of a set Φ such that $\nabla\Phi$ is a subformula of ξ.

▶ `(Define notion of subformula, ensure that` $\bigwedge \alpha$ `and` $\nabla\Phi$ `are direct subformulas of` $\alpha \bullet \Phi$`.)`

We leave it for the reader to construct a tree model $\mathbb{S}'$ with root r', together with a bounded morphism $g : \mathbb{S}' \to \mathbb{S}$ such that every non-root node s' of $\mathbb{S}'$ has at least k many siblings t' such that $g(t') = g(s')$.

Our goal will be to supply $\exists$, in the evaluation game of ξ on $(\mathbb{S}', r')$, with a separating winning strategy f' which is closely linked to f. The key claim in our proof will be the observation that the gain in branching degree enables her to separate the elements of any set Φ, in case the formula $(\nabla\Phi, s)$ is encountered during the play.

Claim 1 Let $s \in S$ and $s' \in S'$ be such that $g(s') = s$. Let $\nabla\Phi$ be a subset of ξ, and let $Z \subseteq \Phi \times R[s]$ be full on Φ and $R[s]$. Then there is a separating relation $Z' \subseteq \Phi \times R'[s']$ such that Z' is full on Φ and $R'[s']$ and $(\varphi, g(t')) \in Z$ whenever $(\varphi, t') \in Z'$.

Proof of Claim Given a successor $t \in R[s]$, we define $\Phi_t := \{\varphi \in \Phi \mid (\varphi, t) \in Z\}$; that is, Φ_t consists of all formulas that $\exists$ connects to t with her choice of the relation Z. Furthermore let $A_t := R[s'] \cap g^{-1}(t)$ consist of all successors of s' that g maps to t; then our assumption on g states that $k \leq |A_t|$.

Clearly then we have $|\Phi_t| \leq |\Phi| \leq k \leq |A_t|$, so that we may assume the existence of a *surjection*

$$\zeta_t : A_t \to \Phi_t,$$

and since the sets A_t partition the set $R'[s']$, we may easily combine the maps ζ_t into a single map

$$\zeta : R'[s'] \to \Phi,$$

simply by putting $\zeta(t') := \zeta_{g(t')}(t')$. It is then straightforward to verify that the relation Z' given by

$$Z' := \{(t', \zeta(t') \mid t' \in R'[s']\}$$

satisfies the requirements of the Claim. ◀

On the basis of this claim we will be able to provide $\exists$ with a winning strategy in the evaluation game in $\mathcal{E}' := \mathcal{E}(\xi, \mathbb{S}')$.

Claim 2 $\exists$ has a strategy f' in $\mathcal{E}'$ which guarantees that every f'-guided play $\pi = (\varphi_1, s'_1) \cdots (\varphi_n, s'_n)$ starting at position $(\varphi_1, s'_1) = (\xi, r')$ is such that the sequence $\pi^g = (\varphi_1, g(s'_1)) \cdots (\varphi_n, g(s'_n))$ is an f-guided match of $\mathcal{E}$ starting at position (ξ, r).

Proof of Claim Since we assumed that $\exists$'s strategy f in $\mathcal{E}$ is positional, we will in fact also be able to provide her with a positional strategy in $\mathcal{E}'$. The definition of f' is straightforward:

- At a position of the form $(\varphi_0 \vee \varphi_1, s')$, check whether the position $(\varphi_0 \vee \varphi_1, g(s'))$ is winning for $\exists$ in $\mathcal{E}$. If so, in $\mathcal{E}'$ at position $(\varphi_0 \vee \varphi_1, g(s'))$, f' picks the same disjunct as f does at the position $(\varphi_0 \vee \varphi_1, g(s'))$. If not, f' picks φ_i randomly.
- At a position of the form $(\nabla\Phi, s')$, check whether the position $(\nabla\Phi, g(s'))$ is winning for $\exists$ in $\mathcal{E}$. If so, suppose that $Z \subseteq \Phi \times R[g(s')]$ is the relation picked by f; then $\exists$ picks some arbitrary but fixed relation $Z' \subseteq \Phi \times R'[s']$ as given by Claim 1. If not, f picks some random legitimate relation Z' (unless she gets stuck).

The statement of the Claim can then be proved by a straightforward induction on the length n of the f'-guided play π, with the note that (for the well-definedness and legitimacy of f') we also need to show that the last position (φ_n, s'_n) of π is such that $(\varphi_n, g(s'_n))$ is winning for f in $\mathcal{E}$. ◀

Now assume that in $\mathcal{E}'$, $\exists$ plays some arbitrary but fixed strategy f' as given by Claim 2. It easily follows from the same claim (and the assumption that f is winning for her in $\mathcal{E}$) that playing f' she will never get stuck. This means that she wins every *finite* f'-guided $\mathcal{E}'$-match.

To see that f' is winning for her in $\mathcal{E}'$, consider an arbitary *infinite* f'-guided match $\pi = (\varphi_n, s'_n)_{n<\omega}$ starting at $(\varphi_0, s'_0) = (\xi, r')$. It follows from Claim 2 that the sequence $\pi^g = (\varphi_n, g(s'_n))_{n<\omega}$ is an f-guided $\mathcal{E}$-match, and thus, won by $\exists$. But then clearly π, which features exactly the same infinite sequence of formulas as π^g, is also winning for her.

Finally it is immediate from its definition and Claim 1 that f' is separating. QED

Since the cover modality can be expressed in terms of the box and diamond operators, it is obvious that μDML can be thought of as a *fragment* of the full language of the μ-calculus. One of the fundamental theorems in the theory of the modal μ-calculus is that μDML has the *same* expressive power as the full language. This equivalence is in fact effective, as stated by the next theorem.

Theorem 10.8 *There are effective procedures transforming an arbitrary formula $\varphi \in \mu$ML into an equivalent disjunctive formula, and vice versa. As a corollary, the languages μML, μML_∇ and μDML all have the same expressive power.*

The *proof* of Theorem 10.7 will be given in a later chapter.

10.2 The small model property

In this section we sketch a proof showing that the modal μ-calculus has the *small model property*. That is, every satisfiable formula $\xi \in \mu$ML is satisfiable in a model of size bounded in the size of ξ.

The key tool in our proof of the small model property will be the following *satisfiability game* that we may associate with a μ-calculus formula. Intuitively the reader may think of this game as the simultaneous projection on $Cl(\xi)$ of all acceptance games of ξ, as should become clear from the proof of Theorem **??** below.

Definition 10.9 Given a formula $\xi \in \mu\mathtt{ML}$, we define the *satisfiability game* $\mathcal{S}(\xi)$ as follows. Its *positions* are provided by the formulas in the closure of ξ, and the ownership and admissible moves of each position can be found in Table 24. The winning condition for infinite matches is defined as the winning condition of the closure game. $\lhd$

Position	Player	Admissible moves
$\bigvee \Phi$	$\exists$	Φ
$\eta x\, \varphi$	$-$	$\{\varphi[\eta x\, \varphi / x\}$
$\alpha \bullet \Phi$, α consistent	$\forall$	Φ
$\alpha \bullet \Phi$, α in consistent	$\exists$	$\varnothing$

Table 24: Satisfiability game for Kripke automata

For this game we have the following result.

Theorem 10.10 *The following are equivalent, for any disjunctive formula ξ:*
1) ξ is satisfiable;
2) $\exists$ has a winning strategy in $\mathcal{S}(\xi)$;
3) ξ is satisfiable in a model based on some subset of $Cl(\xi)$.

Proof. $\boxed{(1) \Rightarrow (2)}$ Assume that ξ is satisfied by the pointed model $(\mathbb{S}, s)$. It is not too hard to transform a winning strategy f for $\exists$ in the evaluation game $\mathcal{E}(\xi, \mathbb{S})@(\xi, s)$ into a winning strategy for her in the game $\mathcal{S}(\xi)@\xi$.

$\boxed{(2) \Rightarrow (3)}$ This is the interesting implication of the theorem. Assume that $\exists$ has some winning strategy f in the satisfiability game $\mathcal{S}(\xi)@\xi$. First of all, sicne $\mathcal{S}$ is a parity game, we may without loss of generality assume that f is *positional.*

Given the nature of the game, it should be clear that if $\exists$ plays this winning strategy starting at an arbitrary winning position $\varphi \in Cl(\xi)$, then after finitely many moves that are either moves of $\exists$ or automatic moves (unfoldings) we reach a unique formula $\widetilde{f}(\varphi)$ of the form $\alpha \bullet \Phi$. More precisely, we define the map $\widetilde{f} : \mathrm{Win}_\exists(\mathcal{S}(\xi)) \to \mathrm{Win}_\exists(\mathcal{S}(\xi))$:

$$\widetilde{f}(\varphi) := \begin{cases} \widetilde{f}(f(\bigvee \Phi)) & \text{if } \varphi = \bigvee \Phi \\ \widetilde{f}(\psi[\eta x\, \psi / x]) & \text{if } \varphi = \eta x\, \psi \\ \alpha \bullet \Phi & \text{if } \varphi = \alpha \bullet \Phi. \end{cases}$$

We leave it for the reader to define that $\widetilde{f}$ is well-defined.

We will now define a model $\mathbb{S}_f$, the states of which are based on the set of winning positions for $\exists$ of the form $\alpha \bullet \Phi$:

$$S_f := \{\alpha \bullet \Phi \in Cl(\xi) \mid \alpha \bullet \Phi \in \mathrm{Win}_\exists(\mathcal{S}(\xi))\}.$$

The definition of the valuation V_f is rather obvious:

$$V_f(p) := \{\alpha \bullet \Phi \in S_f \mid p \in \alpha\},$$

while for the definition of the relation R_f we involve the map $\widetilde{f}$:

$$R_f[\alpha \bullet \Phi] := \{\widetilde{f}(\varphi) \mid \varphi \in \Phi\}.$$

In the sequel it will be convenient to write

$$r_\varphi := \widetilde{f}(\varphi)$$

for any formula $\varphi \in \mathrm{Win}_\exists$. We think of r_φ as the state in $\mathbb{S}_f$ that represents φ.

The implication (2) $\Rightarrow$ (3) follows immediately from the following Claim.

Claim 1 $\mathbb{S}_f, r_\xi \Vdash \xi$.

Proof of Claim The basic idea underlying the proof is that in the evaluation game $\mathcal{E}(\mathbb{S}_f, \xi)$ starting at position (r_ξ, ξ) $\exists$ can easily maintain the following two condition on a partial match $\Sigma = (s_i, \varphi_i)_{0 \leq i \leq n}$ with $(s_0, \varphi_0) = (r_\xi, \xi)$:

(i) every position (s_i, φ_i) is such that $s_i = r_{\varphi_i}$;

(ii) the sequence $(\varphi_i)_{0 \leq i \leq n}$ is an f-guided match of $\mathcal{S}(\xi)$.

It is then easy to see that this provides $\exists$ with a winning strategy in $\mathcal{E}(\mathbb{S}_f, \xi)@(r_\xi, \xi)$, and this suffices to prove the claim. ◀

(3) $\Rightarrow$ (1) Trivial. QED

A Mathematical preliminaries

Sets and functions We use standard notation for set-theoretic operations such as union, intersection, product, etc. The power set of a set S is denoted as $\wp(S)$ or $\wp S$, and we sometimes denote the relative complement operation as $\sim_S X := S \setminus X$. The size or cardinality of a set S is denoted as $|S|$.

Let $f : A \to B$ be a function from A to B. Given a set $X \subseteq A$, we let $f[X] := \{f(a) \in B \mid a \in X\}$ denote the image of X under f, and given $Y \subseteq B$, $f^{-1}[Y] := \{a \in A \mid f(a) \in Y\}$ denotes the preimage of Y. In case f is a bijection, we let f^{-1} denote its inverse. The composition of two functions $f : A \to B$ and $g : B \to A$ is denoted as $g \circ f$ or gf, and the set of functions from A to B will be denoted as either B^A or $A \to B$.

It is well-known that there is a bijective correspondence, often called 'currying':

$$(A \times B) \to C \;\cong\; A \to (B \to C),$$

which associates, with a function $f : A \times B \to C$, the map that, for each $a \in A$, yields the function $f_a : B \to C$ given by $f_a(b) := f(a, b)$.

Relations Given a relation $R \subseteq A \times B$, we introduce the following notation. $\mathsf{Dom}(R)$ and $\mathsf{Ran}(R)$ denote the domain and range of R, respectively. R^{-1} denotes the converse of R. For $R \subseteq S \times S$, R^* denotes the reflexive-transitive closure of R, and R^+ the transitive closure. For $X \subseteq A$, we put $R[X] := \{b \in B \mid (a, b) \in R \text{ for some } a \in X\}$; in case $X = \{s\}$ is a singleton, we write $R[s]$ instead of $R[\{s\}]$. For $Y \subseteq B$, we will write $\langle R\rangle Y$ rather than $R^{-1}[Y]$, while $[R]Y$ denotes the set $\{a \in A \mid b \in Y \text{ whenever } (a, b) \in R\}$. Note that $[R]Y = A \setminus \langle R\rangle(B \setminus Y)$. A relation R on S is *acyclic* if there are no elements s such that R^+ss.

An *equivalence relation* on a set A is a binary relation that is reflexive, symmetric and transitive. The *equivalence class* or *cell* of an element $a \in A$ relative to an equivalence relation is the set of all elements in A that are linked to a by the relation.

A *preorder* is a structure $(P, \sqsubseteq)$ such that $\sqsubseteq$ is a reflexive and transitive relation on P; given such a relation we will write $\sqsubset$ for the asymmetric version of $\sqsubseteq$ (given by $u \sqsubset v$ iff $u \sqsubseteq v$ but not $v \sqsubseteq u$) and $\equiv$ for the equivalence relation induced by $\sqsubseteq$ (given by $u \equiv v$ iff $u \sqsubseteq v$ and $v \sqsubseteq u$). Cells (that is, equivalence classes) of such a relation will often be called *clusters*. A preorder is directed if for any two points u and v there is a w such that $u \sqsubseteq w$ and $v \sqsubseteq w$. A *partial order* is a preorder $\sqsubseteq$ which is antisymmetric, i.e., such that $p \sqsubseteq q$ and $q \sqsubseteq p$ imply $p = q$. Observe that in a poset we have that $p \sqsubset q$ iff $p \sqsubseteq q$ and $p \neq q$.

Sequences, lists and streams Given a set C, we define C^* as the set of finite *lists*, *words* or *sequences* over C. We will write ε for the empty sequence, and define $C^+ := C^* \setminus \{\varepsilon\}$ as the set of nonempty words. An infinite word, or *stream* over C is a map $\gamma : \omega \to C$ mapping natural numbers to elements of C; the set of these maps is denoted by C^ω. We write $\Sigma^\infty := \Sigma^* \cup \Sigma^\omega$ for the set of all sequences over Σ. The concatenation of a (finite) word u and a (finite or infinite) word v is denoted as $u \cdot v$ or uv. Where $\kappa \in \omega \cup \{\omega\}$, and $(\pi_i)_{0 \leq i < \kappa}$ is a sequence of finite sequences, we denote its concatenation as $\bigodot_{0 \leq i < \kappa} \pi_i$ (with the understanding that this denotes the empty sequence ε in case $\kappa = 0$).

We use $\sqsubseteq$ for the initial segment relation between sequences, and $\sqsubset$ for the proper (i.e., irreflexive) version of this relation. For a nonempty sequence π, *first*(π) denotes the first element of π. In the case that π is finite and nonempty we write *last*(π) for the last element of π. Given a stream $\gamma = c_0 c_1 \ldots$ and two natural numbers $i < j$, we let $\gamma[i, j)$ denote the finite word $c_i c_{i+1} \ldots c_{j-1}$.

Graphs and trees A *(directed) graph* is a pair $\mathbb{G} = \langle G, E \rangle$ consisting of a set G of *nodes* or *vertices* and a binary *edge* relation E on G. A *finite path* through such a graph is a nonempty sequence $(s_i)_{0 \leq i \leq n} = s_0 \cdots s_n$ in G^* such that $E s_i s_{i+1}$ for all $i < n$. Similarly, an *infinite path* is a sequence $(s_i)_{0 \leq i < \omega} = s_0 s_1 \cdots$ in G^ω such that $E s_i s_{i+1}$ for all $i < \omega$. A (proper) *cycle* is a path $s_0 \cdots s_n$ such that $n > 0$, $s_0 = s_n$ and $s_0, \ldots, s_{n-1}$ are all distinct. A graph is *acyclic* if it has no cycles.

A *tree* is a graph $\mathbb{T} = (T, R)$ which contains a node r, called a *root* of $\mathbb{T}$, such that every element $t \in T$ is reachable by a *unique* path from r. (In particular, this means that $\mathbb{T}$ is acyclic, and that the root is unique.) Where s and t are nodes in some tree (T, R), if $(s, t) \in R$ we say that t is a *child* of s and that s is the *parent* of t. If $(s, t) \in R^+$ we call s an *ancestor* of t, and t a *descendant* of s. Distinct nodes with the same parent are called *siblings*.

Fact A.1 (König's Lemma) *Let $\mathbb{G}$ be a finitely branching, acyclic tree. If $\mathbb{G}$ is infinite, then it has an infinite path.*

Order and lattices A *partial order* is a structure $\mathbb{P} = \langle P, \leq \rangle$ such that $\leq$ is a reflexive, transitive and antisymmetric relation on P. Given a partial order $\mathbb{P}$, an element $p \in P$ is an *upper bound* (*lower bound, respectively*) of a set $X \subseteq P$ if $p \geq x$ for all $x \in X$ ($p \leq x$ for all $x \in X$, respectively). If the set of upper bounds of X has a minimum, this element is called the *least upper bound*, *supremum*, or *join* of X, notation: $\bigvee X$. Dually, the *greatest lower bound*, *infimum*, or *meet* of X, if existing, is denoted as $\bigwedge X$. Generally, given a statement S about ordered sets, we obtain its *dual statement* by replacing each occurrence of $\leq$ with $\geq$ and vice versa. The following principle often reduces our work load by half;

Order Duality Principle If a statement holds for all ordered sets, then so does its dual statement.

A partial order $\mathbb{P}$ is called a *lattice* if every two-element subset of P has both an infimum and a supremum; in this case, the notation is as follows: $p \wedge q := \bigwedge\{p, q\}$, $p \vee q := \bigvee\{p, q\}$. Such a lattice is *bounded* if it has a minimum $\bot$ and a maximum $\top$. A partial order $\mathbb{P}$ is called a *complete lattice* if every subset of P has both an infimum and a supremum. In this case we abbreviate $\bot := \bigvee \varnothing$ and $\top := \bigwedge \varnothing$; these are the smallest and largest elements of $\mathbb{C}$, respectively. A complete lattice will usually be denoted as a structure $\mathbb{C} = \langle C, \bigvee, \bigwedge \rangle$. Key examples of complete lattices are full power set algebras: given a set S, it is easy to show that the structure $\langle \wp(S), \bigcup, \bigcap \rangle$ is a complete lattice.

Given a family $\{\mathbb{P}_i \mid i \in I\}$ of partial orders, we define the *product* order $\prod_{i \in I} \mathbb{P}_i$ as the structure $\langle \prod_{i \in I} P_i, \leq \rangle$ where $\prod_{i \in I} P_i$ denotes the cartesian product of the family $\{P_i \mid i \in I\}$, and $\leq$ is given by $\pi \leq \pi'$ iff $\pi(i) \leq_i \pi'(i)$ for all $i \in I$. It is not difficult to see that the product of a family of (complete) lattices is again a (complete) lattice, with meets and joins given coordinatewise. For instance, given a family $\{\mathbb{C}_i \mid i \in I\}$ of complete lattices, and a subset $\Gamma \subseteq \prod_{i \in I} C_i$, it is easy to see that Γ has a least upper bound $\bigvee \Gamma$ given by

$$(\bigvee \Gamma)(i) = \bigvee \{\gamma(i) \mid \gamma \in \Gamma\},$$

where the join on the right hand side is taken in $\mathbb{C}_i$.

Ordinals A set S is *transitive* if $S \subseteq \wp(S)$; that is, if every element of S is a subset of S, or, equivalently, if $S'' \in S' \in S$ implies that $S'' \in S$. An *ordinal* is a transitive set of which all elements are also transitive. From this definition it immediately follows that any element of an ordinal is again an ordinal. We let $\mathcal{O}$ denote the class of all ordinals, and use lower case Greek symbols (α, β, γ, ..., λ, ...) to refer to individual ordinals.

The smallest, *finite*, ordinals are

$$\begin{array}{lcll} 0 & := & \varnothing & \\ 1 & := & \{0\} & (= \{\varnothing\}) \\ 2 & := & \{0,1\} & (= \{\varnothing, \{\varnothing\}\}) \\ 3 & := & \{0,1,2\} & (= \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}) \\ \vdots & & & \end{array}$$

In general, the *successor* $\alpha + 1$ of an ordinal α is the set $\alpha \cup \{\alpha\}$; it is easy to check that $\alpha + 1$ is again an ordinal. Ordinals that are not the successor of an ordinal are called *limit* ordinals. Thus the smallest limit ordinal is 0; the next one is the first infinite ordinal

$$\omega \ := \ \{0, 1, 2, 3, \ldots\}.$$

But it does not stop here: the successor of ω is the ordinal $\omega+1$, etc. It is important to realize that there are in fact too many ordinals to form a set: $\mathcal{O}$ is a proper class. As a consequence, whenever we are dealing with a function $f : \mathcal{O} \to A$ from $\mathcal{O}$ into some set A, we can conclude that there exist distinct ordinals $\alpha \neq \beta$ with $f(\alpha) = f(\beta)$. (Such a function f will also be a class, not a set.)

We define an ordering relation $<$ on ordinals by:

$$\alpha < \beta \text{ if } \alpha \in \beta.$$

From this definition it follows that $\alpha = \{\beta \text{ in } \mathcal{O} \mid \beta < \alpha\}$ for every ordinal α. The relation $<$ is obviously transitive (if we permit ourselves to apply such notions to relations that are classes, not sets). It follows from the axioms of ZFC that $<$ is in fact *linear* (that is, for any two ordinals α and β, either $\alpha < \beta$, or $\alpha = \beta$, or $\beta < \alpha$) and *well-founded* (that is, every non-empty set of ordinals has a smallest element).

The fact that $<$ is well-founded allows us to generalize the principle of induction on the natural numbers to the transfinite case.

Transfinite Induction Principle In order to prove that all ordinals have a certain property, it suffices to show that the property is true of an arbitrary ordinal α whenever it is true of all ordinals $\beta < \alpha$.

A proof by transfinite induction typically contains two cases: one for successor ordinals and one for limit ordinals (the base case of the induction is then a special case of a limit ordinal). Analogous to the transfinite inductive proof principle there is a *Transfinite Recursion Principle* according to which we can construct an ordinal-indexed sequence of objects.

References

[1] P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, chapter C.5, pages 739–782. North-Holland Publishing Co., Amsterdam, 1977.

[2] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 2001.

[3] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.

[5] F. Bruse, O. Friedmann, and M. Lange. On guarded transformation in the modal μ-calculus. *Logic Journal of the IGPL*, 23(2):194–216, 2015.

[6] J.R. Büchi. On a decision method in restricted second order arithmetic. In E. Nagel, editor, *Proceedings of the International Congress on Logic, Methodology and the Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[7] A. Chagrov and M. Zakharyaschev. *Modal Logic*, volume 35 of *Oxford Logic Guides*. Oxford University Press, 1997.

[8] G. D'Agostino and M. Hollenberg. Logical questions concerning the μ-calculus. *Journal of Symbolic Logic*, 65:310–332, 2000.

[9] S. Demri, V. Goranko, and M. Lange. *Temporal Logics in Computer Science: Finite-State Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

[10] E.A. Emerson and C.S. Jutla. The complexity of tree automata and logics of programs (extended abstract). In *Proceedings of the 29th Symposium on the Foundations of Computer Science*, pages 328–337. IEEE Computer Society Press, 1988.

[11] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proceedings of the 32nd Symposium on the Foundations of Computer Science*, pages 368–377. IEEE Computer Society Press, 1991.

[12] D. Janin and I. Walukiewicz. Automata for the modal μ-calculus and related results. In *Proceedings of the Twentieth International Symposium on Mathematical Foundations of Computer Science, MFCS'95*, volume 969 of *LNCS*, pages 552–562. Springer, 1995.

[13] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic. In *Proceedings of the Seventh International Conference on Concurrency Theory, CONCUR '96*, volume 1119 of *LNCS*, pages 263–277, 1996.

[14] B. Knaster. Un théorème sur les fonctions des ensembles. *Annales de la Societé Polonaise de Mathematique*, 6:133–134, 1928.

[15] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[16] D. Kozen. A finite model theorem for the propositional μ-calculus. *Studia Logica*, 47:233–241, 1988.

[17] D. Kozen and R. Parikh. A decision procedure for the propositional μ-calculus. In *Proceedings of the Workshop on Logics of Programs 1983*, LNCS, pages 313–325, 1983.

[18] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

[19] L. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999. (Erratum published *Ann.P.Appl.Log.* 99:241–259, 1999).

[20] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In A. Skowron, editor, *Computation Theory*, LNCS, pages 157–168. Springer-Verlag, 1984.

[21] D.E. Muller. Infinite sequences and finite machines. In *Proceedings of the 4th IEEE Symposium on Switching Circuit Theory and Logical Design*, pages 3–16, 1963.

[22] D. Niwiński. On fixed point clones. In L. Kott, editor, *Proceedings of the 13th International Colloquium on Automata, Languages and Programming (ICALP 13)*, volume 226 of *LNCS*, pages 464–473, 1986.

[23] D. Park. Concurrency and automata on infinite sequences. In *Proceedings 5th GI Conference*, pages 167–183. Springer, 1981.

[24] A. Pnueli. The temporal logic of programs. In *Proc. 18th Symp. Foundations of Computer Science*, pages 46–57, 1977.

[25] V.R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proc. 17th IEEE Symposium on Computer Science*, pages 109–121, 1976.

[26] S. Safra. On the complexity of ω-automata. In *Proceedings of the 29th Symposium on the Foundations of Computer Science*, pages 319–327. IEEE Computer Society Press, 1988.

[27] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[28] T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society*, 8:359–391, 2001.

[29] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998.