10 Modal automata

10.1 Introduction

In this chapter we introduce and discuss the automata that we shall use to study the modal μ-calculus. These automata come in various shapes and types, but they all operate on the same type of structures, namely pointed Kripke structures, or transition systems.

The basic idea is that automata can be seen as alternatives to formulas. In particular, an automaton $\mathbb{A}$ will either accept of reject a given pointed Kripke model, and thus it can be compared to a formula ξ, which will either be true or false at a point in a Kripke model. This inspires the following definition.

Definition 10.1 Let $\mathbb{A}$ be an automaton, and assume that we have defined the notions of acceptance and rejection of a pointed Kripke model by such an automaton. In case $\mathbb{A}$ *accepts* the pointed Kripke structure $(\mathbb{S}, s)$ we write $\mathbb{S}, s \Vdash \mathbb{A}$, while *rejection* of $(\mathbb{S}, s)$ by $\mathbb{A}$ is denoted as $\mathbb{S}, s \nVdash \mathbb{A}$. The class of pointed Kripke models that are accepted by a given automaton $\mathbb{A}$ is denoted as $\mathcal{Q}(\mathbb{A})$, and we will sometimes refer to $\mathcal{Q}(\mathbb{A})$ as the class or *query* that is *recognized* by $\mathbb{A}$. Two automata $\mathbb{A}$ and $\mathbb{A}'$ are *equivalent*, notation: $\mathbb{A} \equiv \mathbb{A}'$, if $\mathcal{Q}(\mathbb{A}) = \mathcal{Q}(\mathbb{A}')$.

We say that a formula ξ is *equivalent* to $\mathbb{A}$, notation: $\xi \equiv \mathbb{A}$, if $\mathbb{S}, s \Vdash \xi$ iff $\mathbb{A}$ accepts $(\mathbb{S}, s)$, for every pointed Kripke model $(\mathbb{S}, s)$. $\lhd$

All our automata will be of the form $\mathbb{A} = \langle A, \Theta, Acc, a_I \rangle$ where A is a finite set of states, $Acc \subseteq A^\omega$ is the acceptance condition (usually given by a parity map Ω), $a_I \in A$ is the starting state of the automaton, and the transition map Θ has as its domain the set $A \times C$, where $C = \wp(\mathsf{P})$ is the set of *colors* over some set P of proposition letters. We will almost exclusively work with automata that are themselves *logic-based*, in the sense that the co-domain of Θ is some logical language consisting of relatively simple *one-step* formulas over the carrier set A of the automata. In other words, the states in A will play a double role as propositional *variables.*

For each type of automaton that we will encounter, the question whether such a device accepts or rejects a given pointed Kripke model $(\mathbb{S}, s)$ is determined by playing some kind of infinite board game that we call the *acceptance game* associated with the automaton and the Kripke structure. This game will always proceed in *rounds*, each of which starts and ends at a so-called *basic position* $(a, s) \in A \times S$, and consists of the two players, $\exists$ and $\forall$, moving a token via some intermediate position(s) to a new basic position. For a rough, intuitive understanding of the acceptance game, the reader may think of $\exists$ claiming, at a basic position (a, s), that the automaton $\mathbb{A}$, taken from the perspective a, is a good 'description' of the pointed structure $(\mathbb{S}, s)$.

The rules of the game are determined by the precise shape of the transition function Θ, and in each case will be given explicitly. The winning conditions of the acceptance game are fixed. Finite matches, as always, are lost by the player who got stuck. The winner of an infinite match Σ is always determined by applying the acceptance condition Acc to the infinite A-stream $a_I a_1 a_2 \cdots$ which is induced by the sequence $(a_I, s)(a_1, s_1)(a_2, s_2) \cdots$ of basic positions occurring in Σ. The definition of acceptance is also fixed: the automaton $\mathbb{A}$ *accepts* the pointed Kripke model $(\mathbb{S}, s)$ precisely if the pair (a_I, s) is a winning position for $\exists$ in the acceptance game.

To understand the connections between the various kinds of automata, it is good to understand *how* one round of the game takes a match from one basic position (a_i, s_i) to the next (a_{i+1}, s_{i+1}). In principle, it is $\exists$'s task to propose a set $Z_i \subseteq A \times S$ of *witnesses* that substantiate her claim that the automaton $\mathbb{A}$, taken from the perspective a_i, is a good description of the pointed model $(\mathbb{S}, s_i)$. Then it is $\forall$ who picks the new basic position (a_{i+1}, s_{i+1}) as an element of the set Z_i. In fact, all acceptance games featuring in this chapter could be formulated in such a way that these are exactly the moves that players can make. However, we will usually take a slightly different perspective on the witness relation. In particular, since we are often thinking of A as a set of propositional variables, it will make sense to represent a relation $Z \subseteq A \times S$ as either a *valuation* $V_Z : A \to \wp S$ or as a *marking* or *coloring* $m_Z : S \to \wp A$, defined by putting, respectively,

$$\begin{array}{rcl} V_Z(a) & := & \{s \in S \mid (a, s) \in Z\} \\ m_Z(s) & := & \{a \in A \mid (a, s) \in Z\}. \end{array}$$

As already mentioned, the automata that we shall meet here come in various shapes, and they can be classified in many ways. One crucial distinction to make is that between *alternating* and *non-deterministic* automata. Where the generic modal automaton that we will introduce here is of the alternating type, many results on the modal μ-calculus are proved using the subclass of non-deterministic automata, where the transition map is of a conceptually simpler kind. What makes an automaton *nondeterministic* is the interaction pattern between the two players in the acceptance game: when the automaton is non-deterministic, a winning strategy for $\exists$ should in principle (but depending on the branching structure of the transition system) reduce the role of $\forall$ to that of a *path finder* in the model.

▶ `For the time being we restrict attention to the mono-modal case.`

10.2 Modal automata

Modal automata are based on the *modal one-step language*. This language consists of very simple modal formulas, built up from a collection A of propositional *variables*, corresponding to the bound variables of a formula.

Definition 10.2 Given a set X, we define the set $\texttt{Latt}(X)$ of *lattice terms* over X through the following grammar:

$$\pi \ ::= \ \bot \mid \top \mid x \mid \pi \wedge \pi \mid \pi \vee \pi,$$

where $x \in X$. Given a set A, we define the set $\texttt{1ML}(A)$ of *modal one-step formulas* over A by the following grammar:

$$\alpha \ ::= \ \bot \mid \top \mid \Diamond\pi \mid \Box\pi \mid \alpha \wedge \alpha \mid \alpha \vee \alpha,$$

with $\pi \in \texttt{Latt}(A)$. ◁

Examples of one-step formula are $\Diamond(a \wedge b)$ or $\Box\bot \vee (\Diamond a \wedge \Box b)$. Observe that the set of modal one-step formulas over A corresponds to the set of lattice terms over the set $\{\Diamond\pi, \Box\pi \mid \pi \in \mathtt{Latt}(A)\}$. Observe too that every occurrence of an element of A must be positive, and in the scope of exactly one modality.

Definition 10.3 A *modal* P*-automaton* $\mathbb{A}$ is a quadruple (A, Θ, Ω, a_I) where A is a non-empty finite set of *states*, of which $a_I \in A$ is the *initial* state, $\Omega : A \to \omega$ is the *priority map*, and the *transition map*

$$\Theta : A \times \wp\mathsf{P} \to \mathtt{1ML}(A)$$

maps states to one-step formulas. The class of modal automata over the set P is denoted as $Aut_{\mathsf{P}}(\mathtt{1ML})$. $\lhd$

The operational semantics of modal automata is defined in terms of a so-called *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ associated with a modal automaton $\mathbb{A}$ and a Kripke structure $\mathbb{S}$. $\exists$'s moves in this game will consist of 'local' valuations for the propositional variables in A, or rather, *markings* $m : S \to \wp A$. Such a marking turns a Kripke model over P into a Kripke model over the set $\mathsf{P} \cup A$.

Throughout this chapter we will represent a Kripke model (S, R, V) *coalgebraically* as a triple (S, R, σ_V) where we think of the binary relation R as a map $R : S \to \wp(S)$, and represent the valuation $V : \mathsf{P} \to \wp(S)$ as its transpose colouring $\sigma_V : S \to \wp(\mathsf{P})$.

Definition 10.4 Let P and A be disjoint sets of proposition letters and propositional variables, respectively. Given a Kripke model $\mathbb{S} = (S, R, \sigma_V)$ over the set P, and an A-marking $m : S \to \wp A$, we let $\mathbb{S} \oplus m$ denote the Kripke model $(S, R, \sigma_V \cup m)$, where $\sigma_V \oplus m$ is the marking given by $\sigma_V \oplus m(s) := \sigma_V(s) \cup m(s)$. $\lhd$

Definition 10.5 The *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ associated with such an automaton $\mathbb{A}$ and a pointed Kripke model $(\mathbb{S}, s)$ is the parity game that is determined by the rules given in Table 19. Positions of the form $(a, s) \in A \times S$ are called *basic.* $\lhd$

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$\exists$	$\{m : S \to \wp A \mid \mathbb{S} \oplus m, s \Vdash \Theta(a, \sigma_V(s))\}$	$\Omega(a)$
$m : S \to \wp A$	$\forall$	$\{(b, t) \mid b \in m(t)\}$	0

Table 19: Acceptance game for modal automata

As explained in the introduction to this chapter, matches of the acceptance game proceed in *rounds*, moving from one basic position to the next. During a round of the game, the players are inspecting a local 'window' into the Kripke model, by means of a one-step formula. Concretely, at the start of a round, $\exists$'s task at a basic position (a, s) is to *satisfy* the one-step formula $\Theta(a, \sigma_V(s))$ at the state s in $\mathbb{S}$. For this purpose, she has to come up with a interpretation for the variables in A, since this is not provided by the valuation V of $\mathbb{S}$. More specifically, $\exists$ has to select a marking $m : S \to \wp A$, in such a way that the formula

$\Theta(a, \sigma_V(s))$ becomes true at s in the model $\mathbb{S} \oplus m$ (as given in Definition 10.4). Once $\exists$ has made her choice, it is $\forall$'s turn; he needs to pick a new basic position from the witness set $\{(b,t) \mid b \in m(t)\}$.

Observe that both players could get stuck in such a match. For instance, it might be impossible for $\exists$ to satisfy the formula $\Theta(a, \sigma_V(s))$ at the state s, because the formula requires s to have successors where it has none. Alternatively, if $\exists$ could pick the *empty* marking m at a position (a, s), then she would immediately win the match since $\forall$ would get stuck.

▶ `examples of modal automata`

Remark 10.6 Note that it is in $\exists$'s interest to keep, at any basic position (s, a) of the acceptance game, the set of witnesses as small as possible. More precisely, if at some position (a, s) of the game, $\exists$ has two admissible markings, say, m and m', at her disposal, and these are such that $Z_m := \{(b,t) \in S \times A \mid b \in m(t)\} \subseteq Z_{m'} := \{(b,t) \in S \times A \mid b \in m'(t)\}$, then it will always be to her advantage to choose the marking m rather than m'. In particular, since all occurrences of propositional variables from A in one-step formulas must be in the scope of exactly one modality, to satisfy such a formula at a given point s of the model, the only points that matter are the successors of s. For these reasons, we may without loss of generality restrict the admissible moves of $\exists$ at a position (a, s) of the acceptance game to those markings m of which the domain is the collection of successors of current point s. In section 10.4 we will work out this perspective. $\lhd$

Convention 10.7 We will usually identify a match $\Sigma = (a_0, s_0)m_0(a_1, s_1)m_1(a_2, s_2)m_2 \ldots$ of the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ with the sequence $(a_0, s_0)(a_1, s_1)(a_2, s_2)\ldots$ of its *basic positions.*

Some basic concepts concerning modal automata are introduced in the following definition.

Definition 10.8 Fix a modal P-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$.

Given a state a of $\mathbb{A}$, we write $\eta_a = \mu$ if $\Omega(a)$ is odd, and $\eta_a = \nu$ if $\Omega(a)$ is even; we call η_a the *(fixpoint) type* of a and say that a is an η_a-state. The sets of μ- and ν-states are denoted with A^μ and A^ν, respectively.

The *occurrence graph* of $\mathbb{A}$ is the directed graph $(G, E_{\mathbb{A}})$, where $E_{\mathbb{A}}ab$ if b occurs in $\Theta(a, c)$ for some $c \in \wp(\mathsf{P})$. We let $\lhd_{\mathbb{A}}$ denote the transitive closure of the converse relation $E_{\mathbb{A}}^{-1}$ of $E_{\mathbb{A}}$ and say that b is *active* in a if $b \lhd_{\mathbb{A}} a$. We write $a \bowtie_{\mathbb{A}} b$ if $a \lhd_{\mathbb{A}} b$ and $b \lhd_{\mathbb{A}} a$. A *cluster* of $\mathbb{A}$ is a cell of the equivalence relation generated by $\bowtie_{\mathbb{A}}$ (i.e., the smallest equivalence relation on A containing $\bowtie_{\mathbb{A}}$); a cluster C is *degenerate* if it is of the form $C = \{a\}$ with $a \not\lhd_{\mathbb{A}} a$. The unique cluster to which a state $a \in A$ belongs is denoted as C_a. We write $a \sqsubset_{\mathbb{A}} b$ if $\Omega(a) < \Omega(b)$, and $a \sqsubseteq_{\mathbb{A}} b$ if $\Omega(a) \leq \Omega(b)$.

An *alternating* Ω*-chain* of *lenth* k in $\mathbb{A}$ is a sequence $a_0 a_1 \cdots a_k$ of states that all belong to the same cluster and satisfy, for all $i < k$, that $\Omega(a_i) < \Omega(a_{i+1})$ while a_i and a_{i+1} have different parity. $\lhd$

The following proposition is immediate by the definitions.

Proposition 10.9 *Let* $\mathbb{A} = \langle A, \Theta, \Omega, a_I\rangle$ *and* $\mathbb{A}' = \langle A, \Theta', \Omega, a_I\rangle$ *be two modal automata such that* $\Theta(a, c) \equiv \Theta'(a, c)$ *for each* $a \in A$ *and* $c \in \wp(\mathsf{P})$*. Then* $\mathbb{A} \equiv \mathbb{A}'$*.*

Remark 10.10 Another way of defining the semantics of modal automata is via the 'slow' acceptance game of Table 20, which is perhaps closer to the evaluation games of the modal μ-calculus. In this set-up, at a basic position (a, s) $\exists$ does not have to come up with a marking m, but rather, the state a is 'unfolded' into the formula $\Theta(a, \sigma_V(s))$, and the two players engage in a little sub-game in order to determine whether $\Theta(a, \sigma_V(s))$ is true at s or not. At the end of this sub-game, unless one of the players got stuck, the match arrives at another basic position. We leave it as an exercise for the reader to check that the two games are in fact equivalent. $\lhd$

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$-$	$\{(\Theta(a, \sigma_V(s)), s)\}$	$\Omega(a)$
$(\top, s)$	$\forall$	$\varnothing$	0
$(\bot, s)$	$\exists$	$\varnothing$	0
$(\Diamond\pi, s)$	$\exists$	$\{(\pi, t) \mid t \in R(s)\}$	0
$(\Box\pi, s)$	$\forall$	$\{(\pi, t) \mid t \in R(s)\}$	0
$(\varphi_0 \vee \varphi_1, s)$	$\exists$	$\{(\varphi_0, s), (\varphi_1, s)\}$	0
$(\varphi_0 \wedge \varphi_1, s)$	$\forall$	$\{(\varphi_0, s), (\varphi_1, s)\}$	0

Table 20: Slow acceptance game for modal automata

Regarding complexity matters, we define the *size* of a modal automaton to get a nice fit with the (slow) acceptance game defined in Remark 10.10. In particular, this means that we cannot simply define the size of an automaton as its number of states, we have to take the *transition map* of the device into account as well. Note that the size $|\alpha|$ of a modal one-step α is simply defined as its number of subformulas, or, equivalently, as the size of its closure. The *index* of modal automata is defined in the same way as for parity formulas.

Definition 10.11 Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal automaton. The *size* $|\mathbb{A}|$ of $\mathbb{A}$ is defined as follows:

$$|\mathbb{A}| := \sum_{(a,c) \in A \times C} |\Theta(a, c)|.$$

Its *index* $ind(\mathbb{A})$ is given as the maximal length of an alternating Ω-chain in $\mathbb{A}$. $\lhd$

Later on this chapter we will provide effective translations transforming a μ-calculus formula into an equivalent modal automaton, and vice versa. As a corollary of this result we obtain that modal automata are bisimulation invariant — in Exercise 10.2 the reader is asked to give a direct proof.

Theorem 10.12 *Let $\mathbb{A}$ be a modal automaton.. Then for any bisimilar pair $(\mathbb{S}, s)$ and $(\mathbb{S}', s')$ of pointed Kripke models it holds that*

$$\mathbb{S}, s \Vdash \mathbb{A} \iff \mathbb{S}', s' \Vdash \mathbb{A}.$$

10.3 Disjunctive modal automata

A key tool in the study of the model μ-calculus is provided by the automata that we are about to introduce now, viz., the nondeterministic variants of the modal automata that we just met in section 10.2. The *disjunctive* automata, as we shall call them, are obtained by restricting the co-domain of the transition map of a modal automaton to the set of so-called disjunctive one-step formulas, which are based on the cover modality discussed in section 1.7.

Definition 10.13 Given a finite set A, we define the set $\mathtt{1DML}(A)$ of disjunctive modal one-step formulas in A as follows

$$\alpha ::= \bot \mid \top \mid \nabla B \mid \alpha \vee \alpha,$$

where $B \subseteq A$.

A modal P-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is called *disjunctive* or *non-deterministic* if $\Theta(a, c) \in \mathtt{1DML}(A)$, for every $a \in A$ and $c \in \wp(\mathsf{P})$. $\lhd$

▶ `example(s) to be supplied`

Remark 10.14 As a variant of Definition 10.13, we will sometimes require that the range of the transition map Θ of a disjunctive automaton is given by the formulas of the slightly more restricted one-step language $\mathtt{1DML}_r$ given by the following grammar:

$$\alpha ::= \bot \mid \nabla B \mid \alpha \vee \alpha,$$

where $B \subseteq A$. In other words, in this set-up every formula $\Theta(a, c)$ is a finite disjunction of nabla formulas; the difference with the language of Definition 10.13 is that here, the formula $\top$ is not allowed.

We leave it as an exercise to the reader to prove that the two versions of the definition are equivalent, in the sense that there are transformations from one type of automaton into the other. $\lhd$

As already mentioned, the key property making an automaton *non-deterministic* is that, on Kripke structure with a sufficiently nice branching structure, a winning strategy for $\exists$ in the acceptance game should always be able to find markings that are *functional.* We will now make this statement more precise.

Definition 10.15 Let $\mathbb{A}$ and $\mathbb{S}$ be a modal automaton and a Kripke structure, respectively. A strategy f for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is called *separating* if for all partial matches Σ ending in a basic position (a, s), the marking $m_\Sigma : S \to \wp A$ picked by f satisfies $|m_\Sigma(t)| \leq 1$ for all $t \in S$, and $|m_\Sigma(t)| = 0$ for all $t \notin \sigma_R(s)$. $\lhd$

In words, a strategy is separating if it picks markings that assign to each point in S at most one state in A, and assign the empty set to any point that is not a successor of the currently inspected point of S. For a (non-)example, consider the one-step formula $\Diamond a_0 \wedge \Box a_1$; it should be clear that to satisfy this formula at a point s, one needs at least one successor of s where *both* a_0 and a_1 hold. This means that no separating strategy will prescribe a legitimate move for a position of the form (a, s) if the formula that $\exists$ needs to satisfy is $\Theta(a, \sigma_V(s)) = \Diamond a_0 \land \Box a_1$.

Separating winning strategies have the following property, which we will put to good use in the sequel.

Definition 10.16 Let $\mathbb{A}$ and $(\mathbb{S}, r)$ be a modal automaton and a pointed Kripke structure, respectively. A strategy f for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$ is called *functional* if for every $s \in S$ there is at most one $a \in A$ such that the position (a, s) is reachable in an f-guided match of $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$.

In case $\exists$ has a *functional* winning strategy in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, r)$, we say that $\mathbb{A}$ *strongly* accepts $(\mathbb{S}, r)$, and write $\mathbb{S}, r \Vdash_s \mathbb{A}$. $\lhd$

Proposition 10.17 *Let $\mathbb{A}$ be a modal automaton, and let $(\mathbb{S}, s)$ be a pointed tree model. Then every separating winning strategy in the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s)$ is functional.*

We have now arrived at the key result about disjunctive automata.

Theorem 10.18 *Let $\mathbb{A}$ and $(\mathbb{S}, r)$ be a disjunctive modal automaton and a pointed Kripke model, respectively. Then $\mathbb{S}, r \Vdash \mathbb{A}$ iff there is a rooted tree model $(\mathbb{S}', r')$ such that $\mathbb{S}, r \leftrightarroweq (\mathbb{S}', r')$ and $\mathbb{S}', r' \Vdash_s \mathbb{A}$.*

Proof of Theorem 10.18. With $\mathbb{A} = (A, \Theta, \Omega, a_I)$, let $\kappa := |A|$ be the state-size of $\mathbb{A}$. We leave it for the reader to construct a tree model $\mathbb{S}'$ with root r', and a bounded morphism $g : \mathbb{S}' \to \mathbb{S}$ such that $g(r') = r$ and such that every $s' \neq r'$ in $\mathbb{S}'$ has at least $\kappa - 1$ many siblings t' such that $g(t') = g(s')$.

By positional determinacy we may assume that $\exists$ has a positional strategy f in $\mathcal{A}(\mathbb{A}, \mathbb{S})$ which is winning when played from any winning position for $\exists$. We will use this strategy to define a separating positional winning strategy for $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S}')$.

The key claim is the following.

Claim 1 Let $s \in S$ and $s' \in S'$ be such that $g(s') = s$, let $\alpha \in \mathtt{1DML}(A)$ be a one-step formula and let $m : R(s) \to \wp(A)$ be a marking such that $\mathbb{S} \oplus m, g(s') \Vdash \alpha$. Then there is a separating marking $m' : R'(s') \to \wp(A)$ such that $\mathbb{S}' \oplus m', s' \Vdash \alpha$ and $m'(t') \subseteq m(g(t'))$, for all $t' \in R'(s')$.

Proof of Claim In case α contains $\top$ as one of its disjuncts, we simply take the empty marking for m', that is, we define $m'(t') := \varnothing$ for every $t' \in S'$.

In the sequel we focus on the case where α does not contain $\top$ as one of its disjuncts (in fact this is without loss of generality, cf. Remark 10.14). It follows from the legitimacy of m, as a move for $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$, that $\mathbb{S}, m, s \Vdash \alpha$; this means that $\mathbb{S} \oplus m, s \Vdash \nabla B$ for some disjunct ∇B of α, where $B \subseteq A$. We now consider two subcases.

If $B = \varnothing$, it follows from $\mathbb{S} \oplus m, s \Vdash \nabla B$ that $\sigma_R = \varnothing$; but then we also have $\sigma_{R'}(s') = \varnothing$, since g is a bounded morphism. In this case we also define m' as the empty marking.

Finally, assume that $B \neq \varnothing$; since $\mathbb{S} \oplus m, s \Vdash \nabla B$ we may without loss of generality assume that $\varnothing \neq m(t) \subseteq B$, for all $t \in \sigma_R(s)$. Now consider an arbitrary successor t of s. By the assumption on g there are at least κ many successors t' of s' such that $g(t') = t$, and since $\kappa = |A|$ this implies that there is a surjection $h : g^{-1}(t) \to m(t)$. Define $m' : \sigma_{R'}(s') \to \wp A$ by putting

$$m'(t') := \{h(t')\}.$$

We leave it as an exercise for the reader to check that $\mathbb{S}', m', s' \Vdash \nabla B$. This means that $\mathbb{S}', m', s' \Vdash \alpha$, thus establishing that m' is a legitimate move for $\exists$ at position (a, s') in $\mathcal{A}(\mathbb{A}, \mathbb{S}')$ indeed. Finally, it is immediate from the definition of m' that $m'(t') \subseteq m(g(t'))$, for all $t' \in \sigma_{R'}(s')$. ◀

Based on Claim 1, we may provide $\exists$ with the following positional strategy f' in $\mathcal{A}(\mathbb{A}, \mathbb{S}')$. Given a position (a, s'), in case $(a, g(s'))$ is a winning position for $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$, we let f' pick a fixed marking m' as given by the claim, while f' picks an random move in case $(a, g(s')) \notin \mathrm{Win}_\exists(\mathcal{A}(\mathbb{A}, \mathbb{S}))$.

It is not hard to prove that for any f'-guided (partial) match $\Sigma = (a_n, s'_n)_{n<\lambda}$ of $\mathcal{A}(\mathbb{A}, \mathbb{S}')$, its g-projection $\Sigma^g := (a_n, g(s'_n))_{n<\lambda}$ is a f-guided (partial) match of $\mathcal{A}(\mathbb{A}, \mathbb{S}')$. From this it is immediate that f' is a winning strategy when played from a winning position, while it is obvious from its definition that f is separating. QED

Further on in this chapter we will prove a *Simulation Theorem*, providing a construction which effectively transforms a given modal automaton into an equivalent disjunctive modal automaton.

10.4 One-step logics and their automata

Modal one-step logic

As we saw in section 10.2, modal one-step formulas provide the co-domain of the transition map of a modal automaton. The operational semantics of modal automata is given by a two-player acceptance game, and a match of this game proceeds in rounds, during which the players investigate a local window into the Kripke structure, by means of the semantics of one of these one-step formulas. It will be rewarding to introduce some terminology for this 'local window' and study the semantics of one-step formulas in some more detail. This will allow us to introduce the notion of a *one-step logic* and use it to generalise the notion of a modal automaton.

The crucial observation is the following. Consider a modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and a Kripke model $\mathbb{S}$. At a basic position (a, s) of the acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$, $\exists$ has to come up with a marking m which makes the one-step formula $\Theta(a, \sigma_V)$ true at s in the expanded model $\mathbb{S} \oplus m$. The point is that, because of the special shape of modal one-step formulas, we do not use all information on the model $\mathbb{S} \oplus m$: in fact all we need access to is the set $R[s]$ of successors of s, and the marking m. In the sequel it will convenient to present this information in the format of a *one-step model*, which is nothing but a set, together with a marking for the set of variables.

Definition 10.19 Fix a set A. A *one-step A-model over a set Y* is a pair (Y, m) such that $m : Y \to \wp(A)$ is an *A-marking* of the elements of Y with A-colors. ◁

Remark 10.20 In order to deal with blind worlds (points in a Kripke model that have no successors), we need to allow one-step models with an *empty domain.* Observe that there is in fact exactly one such structure: the pair $(\varnothing, \varnothing)$. Apart from this exception, a one-step model is nothing but a *structure* in the sense of first-order model theory, for the signature consisting of a monadic predicate for each element of A. That is, we may consider the A-model (Y, m) as the structure (Y, V_m), simply by representing the marking m by its associated valuation V_m interpreting the variables as subsets of the domain Y. $\lhd$

Definition 10.21 The *one-step satisfaction relation* $\Vdash^1$ between one-step models and modal one-step formulas is defined as follows. Fix a one-step model (Y, m).

First, we define the value $[\![\pi]\!]^0$ of a formula $\pi \in \mathtt{Latt}(A)$ by the following induction:

$$\begin{array}{lcllcl}
[\![a]\!]^0 & := & V_m(a) \quad (= \{t \in Y \mid a \in m(t)\}) & & & \\
[\![\top]\!]^0 & := & Y & [\![\bot]\!]^0 & := & \varnothing \\
[\![\pi_0 \vee \pi_1]\!]^0 & := & [\![\pi_0]\!]^0 \cup [\![\pi_1]\!]^0 & [\![\pi_0 \wedge \pi_1]\!]^0 & := & [\![\pi_0]\!]^0 \cap [\![\pi_1]\!]^0.
\end{array}$$

Sometimes we write $(Y, m), t \Vdash^0 \pi$ in case $t \in [\![\pi]\!]^0$.

Second, we inductively define the one-step satisfaction relation as follows:

$$\begin{array}{lll}
(Y, m) \Vdash^1 \top & & \\
(Y, m) \nVdash^1 \bot & & \\
(Y, m) \Vdash^1 \Box\pi & \text{if} & [\![\pi]\!]^0 = Y \\
(Y, m) \Vdash^1 \Diamond\pi & \text{if} & [\![\pi]\!]^0 \cap Y \neq \varnothing \\
(Y, m) \Vdash^1 \alpha_0 \wedge \alpha_1 & \text{if} & (Y, m) \Vdash^1 \alpha_0 \text{ and } (Y, m) \Vdash^1 \alpha_1 \\
(Y, m) \Vdash^1 \alpha_0 \vee \alpha_1 & \text{if} & (Y, m) \Vdash^1 \alpha_0 \text{ or } (Y, m) \Vdash^1 \alpha_1
\end{array}$$

In case $(Y, m) \Vdash^1 \alpha$ we say that α is *true* in the one-step model (Y, m). $\lhd$

Example 10.22 In this format, the semantics of disjunctive formulas boils down to the following, as can easily be verified, for a subset $B \subseteq A$:

$$(Y, m) \Vdash^1 \nabla B \text{ iff } B \subseteq \bigcup\{m(y) \mid y \in Y\} \text{ and } m(y) \cap B \neq \varnothing, \text{ for all } y \in Y.$$

That is, ∇B holds in a one-step model (Y, m) iff every $b \in B$ is satisfied at some $y \in Y$, and every $y \in Y$ satisfies some $b \in B$.

Furthermore, observe that the empty model will satisfy every formula of the form $\Box\pi$, and no formula of the form $\Diamond\pi$. We have $(Y, m) \Vdash^1 \nabla\varnothing$ iff $Y = \varnothing$. $\lhd$

The following proposition, which can be proved by a straightforward induction on the complexity of one-step formulas, shows that the one-step semantics developed above is just an alternative perspective on the standard semantics of one-step formulas.

Proposition 10.23 *Let $\mathbb{S} = (S, R, V)$ be a Kripke model, let s be a point in S, let $m : R[s] \to \wp(A)$ be an A-marking, and let $\alpha \in \mathtt{1ML}(A)$ be a modal one-step formula. Then*

$$\mathbb{S} \oplus m, s \Vdash \alpha \text{ iff } (R[s], m) \Vdash^1 \alpha.$$

Given Proposition 10.23, the acceptance game of modal automata can now be naturally defined in terms of this one-step semantics, as in Table 21.

Position	Player	Admissible moves	Priority
$(a,s) \in A \times S$	$\exists$	$\{m : R(s) \to \wp A \mid (R(s), m) \Vdash^1 \Theta(a, \sigma_V(s))\}$	$\Omega(a)$
m	$\forall$	$\{(b,t) \mid b \in m(t)\}$	0

Table 21: Acceptance game for one-step automata

General one-step logic

As we will see below, the notion of a one-step logic provides a way to generalise the concept of a modal automaton to a much wider setting.

Definition 10.24 A *one-step language* is a map $\mathtt{L}$ which assigns to any finite set A a collection $\mathtt{L}(A)$ of *one-step formulas over* A. This map is subject to the constraint that every map $\tau : A \to A'$ induces a *substitution* or *renaming* $[\tau] : \mathtt{L}(A) \to \mathtt{L}(A')$ such that
1) $[id_A] = id_{\mathtt{L}(A)}$;
2) $[\tau' \circ \tau] = [\tau'] \circ [\tau]$, for any pair $\tau : A \to A'$ and $\tau' : A' \to A''$;
3) $\alpha[\tau] = \alpha$ for any $\alpha \in \mathtt{L}(A)$, if $\tau : A \to A'$ is such that $\tau(a) = a$ for all $a \in A$. $\triangleleft$

We will use postfix notation for this renaming, writing $\alpha[\tau]$ for the formula we obtain from α by renaming every variable $a \in A$ by $\tau(a) \in A'$. For instance, where $\alpha \in \mathtt{1ML}(A)$ is the formula $\Diamond a \wedge \Box(b \vee c)$ and $\tau : A \to A'$ satisfies $\tau(a) = \tau(c) = a'$ and $\tau(b) = b'$, we find $\alpha[\tau] = \Diamond a' \wedge \Box(b' \vee a')$. Note that it follows from the above definition that $A \subseteq A'$ implies $\mathtt{L}(A) \subseteq \mathtt{L}(A')$, for any one-step language $\mathtt{L}$.

Definition 10.25 A *one-step logic* is a pair $(\mathtt{L}, \Vdash^1)$ consisting of a one-step language $\mathtt{L}$ and an *interpretation* $\Vdash^1$ which indicates, for every one-step A-model (Y,m) and every one-step formula $\alpha \in \mathtt{L}(A)$, whether α is *true* or *false* in (Y,m), denoted as, respectively, $(Y,m) \Vdash^1 \alpha$ and $(Y,m) \nVdash^1 \alpha$.

The interpretation $\Vdash^1$ is subject to the condition of *monotonicity*: if $m(t) \subseteq m'(t)$, for all $t \in Y$, then $(Y,m) \Vdash^1 \alpha$ implies $(Y,m') \Vdash^1 \alpha$, for all $\alpha \in \mathtt{L}(A)$. Furthermore, the interpretation is supposed to be well-behaved with respect to renamings, in the following sense. Observe that a map $\tau : A' \to A$ transforms any A-valuation $V : A \to \wp(Y)$ to an A'-valuation $V \circ \tau : A' \to \wp(Y)$; we will require that $(Y, m_V) \Vdash^1 \alpha[\tau]$ iff $(Y, m_{V \circ \tau}) \Vdash^1 \alpha$, for any formula $\alpha \in \mathtt{L}(A)$. $\triangleleft$

We will generally be sloppy and blur the distinction between a one-step language and a one-step logic, in the understanding that the interpretation of one-step languages is generally fixed (and always clear from context).

In Definition 10.21 we introduced the one-step perspective on modal logic. As a different, particularly interesting example of a one-step logic, we may consider two versions of *monadic first-order logic*, where we see the variables in A as monadic predicate symbols.

Definition 10.26 The set $\mathtt{MFOE}(A)$ of *monadic first-order formulas over* A is given by the following grammar:

$$\alpha ::= \top \mid \bot \mid a(x) \mid \neg a(x) \mid x \doteq y \mid x \not\doteq y \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

where $a \in A$ and x, y are first-order (individual) variables. The language $\mathtt{MFO}(A)$ of *monadic first-order logic* is the equality-free fragment of $\mathtt{MFOE}(A)$; that is, atomic formulas of the form $x \doteq y$ and $x \not\doteq y$ are not permitted:

$$\alpha \ ::= \ \top \mid \bot \mid a(x) \mid \neg a(x) \mid \alpha \vee \alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha \mid \forall x.\alpha$$

In both languages we use the standard definition of free and bound variables, and we call a formula a *sentence* if it has no free variables. For each of the languages $\mathtt{L} \in \{\mathtt{1FO}, \mathtt{1FOE}\}$, we define the *positive fragment* $\mathtt{L}^+$ of $\mathtt{L}$ as the language obtained by almost the same grammar as for $\mathtt{L}$, but with the difference that we do not allow negative formulas of the form $\neg a(x)$ (but do allow formulas $x \not\doteq y$). $\lhd$

To define the *semantics* of these formulas, we make a distinction between the empty one-step model and non-empty models, cf. Remark 10.20. In the latter case we view a one-step model (Y, m) as the first-order structure (Y, V_m). If we add to such a model an *assignment* g, interpreting individual variables of the language as elements of the domain, we may inductively define, in a completely straightforward way, the notion of a monadic formulas being *true* in a model-with-assignment:

$$(Y, m), g \models \alpha.$$

Note the truth of a *sentence* of the language does not depend on the assignment, so that may simply write

$$(Y, m) \models \alpha$$

in case $(Y, m), g \models \alpha$ for some/each assignment.

The *empty* model must be dealt with differently. Since we cannot define assignments on the empty model in a meaningful way, we cannot interpret arbitrary formulas in the empty model. Fortunately, however, we can give an interpretation for every *sentence* of the language, simply by making every formula of the form $\forall x.\alpha$ *true*, and every formula of the form $\exists x.\alpha$ *false* in the empty model. Using this as a basis for an inductive definition, we easily define a truth relation

$$(\varnothing, \varnothing) \models \alpha$$

for any monadic first-order sentence α.

In the light of the above discussion, we will take the (positive) sentences of the languages $\mathtt{MFOE}(A)$ and $\mathtt{MFOE}(A)$ as two respective one-step languages.

Definition 10.27 We define the *one-step languages* $\mathtt{1FOE}(A)$ and $\mathtt{1FO}(A)$ as the collection of *positive sentences* in $\mathtt{MFOE}(A)$ and $\mathtt{MFOE}(A)$, respectively. The semantics $\Vdash^1$ of these languages is defined by putting

$$(Y, m) \Vdash^1 \alpha \text{ iff } (Y, m) \models \alpha,$$

for any one-step model (Y, m). $\lhd$

One-step logic

Continuing our general discussion, we introduce some natural notions pertaining to one-step logics.

Definition 10.28 Two one-step formulas α and α' are *(one-step) equivalent*, denoted $\alpha \equiv_1 \alpha'$, if they are satisfied by exactly the same one-step models. $\lhd$

Example 10.29 Examples of one-step equivalent pairs of formulas include instance of the standard propositional distributive laws, such as the modal distributive law:

$$(\Diamond a_1 \vee \Diamond a_2) \wedge \Box b \equiv_1 (\Diamond a_1 \wedge \Box b) \vee (\Diamond a_2 \wedge \Box b),$$

the familiar axioms of modal logic, such as

$$\Box(a \wedge b) \equiv_1 \Box a \wedge \Box b,$$

but also formulas involving the nabla modality, such as

$$\nabla B \wedge \nabla B' \equiv_1 \bigvee \Big\{ \nabla \{ b \wedge b' \mid bRb' \} \mid R \subseteq B \times B' \text{ and } (B, B') \in \overline{\wp} R \Big\}$$

(cf. Proposition 1.36(1)).

Examples such as

$$\Diamond(a_1 \wedge a_2) \wedge \Box b \equiv_1 \exists x \, (a_1(x) \wedge a_2(x)) \wedge \forall y \, b(y).$$

show that Definition 10.28 also covers the notion of one-step equivalence across languages. $\lhd$

We may lift the notion of equivalence to the level of one-step logics.

Definition 10.30 We say that two one-step $(\mathtt{L}, \Vdash^1)$ and $(\mathtt{L}', \Vdash^{1'})$ languages are *(effectively) equivalent* if for every formula in $\mathtt{L}$ there is an (effectively obtainable) equivalent formula in $\mathtt{L}'$, and vice versa. $\lhd$

A particular interesting example of such an equivalence is the following.

Proposition 10.31 *The one-step languages* $\mathtt{1ML}$ *and* $\mathtt{1FO}$ *are effectively equivalent.*

Proof. It is easy to rewrite a modal one-step formula into an equivalent first-order formula. For the opposite direction, the key observation is that in equality-free *monadic* first-order logic, every formula can be rewritten into a normal form where every monadic predicate is in the scope of exactly one quantifier. QED

Among the results about the modal one-step language that we shall need later is the following one-step version of the usual bisimulation invariance result for modal logic, i.e. all one-step formulas are invariant for bisimulations between one-step models in a precise sense.

Definition 10.32 We say that two one-step A-models (Y, m) and (Y', m') are *one-step bisimilar*, notation: $(Y, m) \leftrightarroweq^1 (Y', m')$, if they satisfy the following conditions:
(forth) for all $s \in S$, there is $s' \in S'$ with $m(s) = m'(s')$;
(back) for all $s' \in S'$, there is $s \in S$ with $m(s) = m'(s')$. $\lhd$

Proposition 10.33 (One-step Bisimulation Invariance) *Let (Y, m) and (Y', m') be two one-step A-models. If $(Y, m) \leftrightarroweq^1 (Y', m')$, then both one-step models satisfy the same formulas in $\mathtt{1ML}(A)$.*

Automata for one-step logics

We now see how the concept of one-step logic naturally give rise to the following generalisation of modal automata.

Definition 10.34 Let $(\mathtt{L}, \Vdash^1)$ be a one-step logic. An $\mathtt{L}$-*automaton* over a set P of proposition letters is a quadruple $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$, where A is a finite state set with initial state a_I, $\Theta : A \times \wp(\mathsf{P}) \to \mathtt{L}(A)$ is a transition function, and $\Omega : A \to \omega$ is a priority map.

The *semantics* of $\mathtt{L}$-automata is given by a two-player acceptance game, of which the rules are given in exactly the same way as those for modal automata, cf. Table 21. $\lhd$

As we will see later on, the automata for $\mathtt{1FO}$ and $\mathtt{1FOE}$ are of particular interest since they correspond to, respectively, the modal μ-calculus and (on tree models) monadic second-order logic. The first observation is immediate by our earlier observations on the equivalence of $\mu\mathtt{ML}$ and modal automata, and Proposition 10.31.

An important theme in the study of these automata is how their properties are already determined at the one-step level. Here are some first examples, regarding the closure properties of $\mathtt{L}$-automata. Recall that a *query* is simply a class of pointed Kripke models.

Definition 10.35 Given be a one-step logic $(\mathtt{L}, \Vdash^1)$, we call a query K $\mathtt{L}$-*recognisable* if there is some $\mathtt{L}$-automaton $\mathbb{A}$ that *recognises* K, i.e., such that $\mathbb{S}, s \Vdash \mathbb{A}$ iff $\mathbb{S}, s$ belongs to K. $\lhd$

We will generally be interested in *closure properties* of the class of recognisable queries. It is rather easy to see that if a one-step language is closed under taking conjunctions/disjunctions, then the associated class of recognisable languages is closed under taking intersections/unions. The question of closure under complementation is more interesting; note that since our one-step languages consist of *monotone* formulas only, closure under negation at the one-step level is not possible.

Definition 10.36 Let $(\mathtt{L}, \Vdash^1)$ be a one-step logic. We say that $\mathtt{L}$ is *closed under taking conjunctions*, if, given a pair of one-step formulas α and β, there is a one-step formula γ such that any one-step model satisfies γ iff it satisfies both α and β. The notion of *closure under disjunctions* is defined analogously.

Given two one-step formulas α and β in $\mathtt{L}(A)$, we call β a *boolean dual* of α if for every one-step model (Y, m) we have that

$$(Y, m) \Vdash^1 \beta \text{ iff } (Y, \overline{m}) \not\Vdash^1 \alpha,$$

where $\overline{m}$ is the *complement marking* of m, given by $\overline{m}(t) := A \setminus m(t)$, for all $t \in Y$. We say that L is *closed under taking boolean duals* if every formula in L has a boolean dual in L. $\triangleleft$

Example 10.37 The one-step modal language is closed under taking conjunctions, disjunctions and boolean duals. We let α^{∂} be the formula we obtain from a formula $\alpha \in \mathtt{1ML}$ by simultaneously replacing all occurrences of $\bot$ by $\top$, all conjunctions by disjunctions, all diamonds by boxes, and vice versa. For example: $\big(\Diamond\top \wedge \Box(a \vee b)\big)^{\partial} = \Box\bot \vee \Diamond(a \wedge b)$. It is easy to verify that for every $\alpha \in \mathtt{1ML}$, the formulas α and α^{∂} are boolean duals of one another.

The one-step language of *disjunctive* modal logic is closed under taking disjunctions, but not conjunctions or boolean duals. $\triangleleft$

Proposition 10.38 *Let* $(\mathtt{L}, \Vdash^1)$ *be a one-step logic.*

1) *If* L *is closed under taking conjunctions, then the* L*-recognisable queries are closed under taking intersections.*
2) *If* L *is closed under taking disjunctions, then the* L*-recognisable queries are closed under taking unions.*
3) *If* L *is closed under taking boolean duals, then the* L*-recognisable queries are closed under complementation.*

Proof. We leave the proof of the first two statements as an exercise to the reader. For the proof of the third part we need to show that with any L-automaton $\mathbb{A}$ we can associate an L-automaton $\overline{\mathbb{A}}$ which accepts exactly those pointed Kripke models that are rejected by $\mathbb{A}$.

Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be an L-automaton, and define $\overline{\mathbb{A}}$ to be the structure $\overline{\mathbb{A}} := (A, \Theta^{\partial}, \Omega', a_I)$ given by putting $\Theta^{\partial}(a, c) := \Theta(a, c)^{\partial}$ and $\Omega'(a) := 1 + \Omega(a)$.

Now take an arbitrary pointed Kripke model $(\mathbb{S}, s)$. Comparing the acceptance games $\mathcal{A}(\mathbb{A}, \mathbb{S})$ and $\mathcal{A}(\overline{\mathbb{A}}, \mathbb{S})$ we observe that the role of $\exists$ in the latter game is basically the same as that of $\forall$ in the first. From this it follows that any position (a, s) is winning for $\exists$ in $\mathcal{A}(\overline{\mathbb{A}}, \mathbb{S})$ iff it is winning for $\forall$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$. Using determinacy we derive that $\mathbb{S}, s \Vdash \overline{\mathbb{A}}$ iff $\mathbb{S}, s \not\Vdash \mathbb{A}$, as required. QED

10.5 From formulas to automata and back

In this section we will substantiate our earlier claim that modal automata are indeed an alternative way to look at the modal μ-calculus. That is, we will provide effective constructions that transform a (parity) formula into an equivalent modal automaton, and vice versa. In both directions we will let these transformations pass via the intermediate structures of *transparent modal automata*; these are variations of modal automata in which the proposition letters, instead of featuring as part of the domain of the transition map, may occur on the co-domain side. That is, we have to extend the definition of one-step formulas, allowing (unguarded) occurrences of proposition letters.

Definition 10.39 Given a set P of proposition letters and a set A of propositional variables, we define the set $\mathtt{1EML}(\mathsf{P}, A)$ of *extended* one-step modal formulas over P and A using the following grammar:

$$\alpha ::= \bot \mid \top \mid p \mid \overline{p} \mid \Diamond\pi \mid \Box\pi \mid \alpha \wedge \alpha \mid \alpha \vee \alpha,$$

with $P \in \mathsf{P}$ and $\pi \in \mathtt{Latt}(A)$. $\lhd$

Observe that in an extended modal one-step formula, the proposition letters from P may only occur 'at the surface', that is, *not* in the scope of a modality; as in $\mathtt{1ML}(A)$-formulas, every occurrence of a variable from A must be in the scope of exactly one modality.

Definition 10.40 A *transparent modal automaton* over a set P of proposition letters is a quadruple of the form $\mathbb{A} = (A, \Theta, \Omega, a_I)$, where A is a finite set of states, of which a_I is the *initial* state, $\Omega : A \to \omega$ is a priority map, and

$$\Theta : A \to \mathtt{1EML}(\mathsf{P}, A)$$

is the transition map.

Given a Kripke model $\mathbb{S} = (S, R, V)$, we define the *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ as the parity game of which the admissible moves and the priority map are given in Table 22. $\lhd$

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$-$	$\{(\Theta(a), s)\}$	$\Omega(a)$
(p, s), with $p \in \mathsf{P}$ and $s \in V(p)$	$\forall$	$\varnothing$	0
(p, s), with $p \in \mathsf{P}$ and $s \notin V(p)$	$\exists$	$\varnothing$	0
$(\overline{p}, s)$, with $p \in \mathsf{P}$ and $s \in V(p)$	$\exists$	$\varnothing$	0
$(\overline{p}, s)$, with $p \in \mathsf{P}$ and $s \notin V(p)$	$\forall$	$\varnothing$	0
$(\top, s)$	$\forall$	$\varnothing$	0
$(\bot, s)$	$\exists$	$\varnothing$	0
$(\varphi_0 \vee \varphi_1, s)$	$\exists$	$\{(\varphi_0, s), (\varphi_1, s)\}$	0
$(\varphi_0 \wedge \varphi_1, s)$	$\forall$	$\{(\varphi_0, s), (\varphi_1, s)\}$	0
$(\Diamond\pi, s)$	$\exists$	$\{(\pi, t) \mid t \in R(s)\}$	0
$(\Box\pi, s)$	$\forall$	$\{(\pi, t) \mid t \in R(s)\}$	0

Table 22: Acceptance game for transparent modal automata

The key feature of this acceptance game is that at a basic position of the form $(a, s) \in A \times S$, the one-step formula $\Theta(a)$ that $\exists$ needs to satisfy at s does not depend on the colour of s. On the other hand, this formula may now contain literals over P, and in this way the colour of s does play a role when the players evaluate the truth of $\Theta(a)$.

In the sequel we will refer to standard modal automata (i.e., as given in Definition 10.3) as *chromatic* to distinguish them from the transparent ones introduced here.

The main part of this section consists of constructions that transform chromatic modal automata into transparent ones and vice versa, and transform parity formulas into transparent modal automata and vice versa. In all cases we will compare the *size* and *index* of the input and the output structure (these notions are defined for transparent automata as for chromatic ones). Throughout the remainder we fix a set P of proposition letters, and we think of the sizes of P and $\wp(\mathsf{P})$ as being constant.

Proposition 10.41 *There is an effective construction that transforms a transparent modal* P*-automaton* $\mathbb{A}$ *into a chromatic modal* P*-automaton* $\mathbb{A}^c$*, such that*
1) $\mathbb{A}^c \equiv \mathbb{A}$*;*
2) $|\mathbb{A}^c| = \mathcal{O}(|\mathbb{A}|)$*;*
3) $ind(\mathbb{A}^c) = ind(\mathbb{A})$*.*

Proof. The intuition behind the transformation is that in the acceptance game for a transparent automaton we may encounter literals over P, which are to be evaluated at the current state. Depending on the colour of the current state, every such literal will be evaluated to be either true or false. This means, that if we fix this colour, as we do in the acceptance game of a chromatic automaton, we can simply *replace* every literal with the appropriate boolean constant ($\top$ or $\bot$), thus obtaining at a one-step formula in the 'not-extended' language $\mathtt{1ML}(A)$. Performing this substitution systematically, we arrive at the following definitions.

Given a colour $c \in \wp(\mathsf{P})$, we define the *substitution* $\tau_c : \mathtt{1EML}(\mathsf{P}, A) \to \mathtt{1ML}(A)$ given by

$$\tau_c(p) := \begin{cases} \top & \text{if } p \in c \\ \bot & \text{if } p \notin c. \end{cases}$$

Based on this we go from a transparent modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ to its chromatic counterpart $\mathbb{A}^c := (A, \Theta', \Omega, a_I)$ by putting

$$\Theta'(a, c) := \Theta(a)[\tau_c].$$

The key observation about these substitutions is that for any Kripke model $\mathbb{S} = (S, R, V)$ over P, any s in $\mathbb{S}$, any A-marking m on s, and any extended one-step formula α we have

$$\mathbb{S} \oplus m, s \Vdash \alpha \text{ iff } \mathbb{S} \oplus m, s \Vdash \alpha[\tau_{c_s}],$$

where c_s is the colour of s under V.

It is this equivalence that enables us to move smoothly between the acceptance games $\mathcal{A}(\mathbb{A}, \mathbb{S})$ and $\mathcal{A}(\mathbb{A}^c, \mathbb{S})$: it shows that at any basic position (a, s), any marking $m : S \to \wp(A)$ is legitimate in $\mathcal{A}(\mathbb{A}, \mathbb{S})$ iff it is legitimate in $\mathcal{A}(\mathbb{A}^c, \mathbb{S})$. From this we easily infer that the winning positions for $\exists$ in the two games coincide, which clearly suffices to prove the equivalence of $\mathbb{A}$ and $\mathbb{A}^c$ (1). The statements (2) and (3) are trivial consequences of the definitions. QED

In the opposite direction there is an equally simple transformation.

Proposition 10.42 *There is an effective construction that transforms a chromatic modal* P*-automaton* $\mathbb{A}$ *into a transparent modal* P*-automaton* $\mathbb{A}^t$*, such that*
1) $\mathbb{A}^t \equiv \mathbb{A}$*;*
2) $|\mathbb{A}^t| = \mathcal{O}(|\mathbb{A}|)$*;*
3) $ind(\mathbb{A}^t) = ind(\mathbb{A})$*.*

Proof. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a chromatic automaton over some set P of proposition letters. We will define $\mathbb{A}^t := (A, \Theta^t, \Omega, a_I)$, where $\Theta^t : A \to \mathtt{1EML}(\mathsf{P}, A)$ is given by

$$\Theta^t(a) := \bigvee_{c \in \wp(\mathsf{P})} \Big(\odot c \wedge \Theta(a, c) \Big).$$

Here $\odot c$ is the formula 'exactly c':

$$\odot c := \bigwedge_{p \in c} p \wedge \bigwedge_{p \in \mathsf{P} \setminus c} \overline{p},$$

which holds in a state s in a Kripke model over P if c is exactly the colour of s. It is easily verified that $\mathbb{A}^t$ satisfies the conditions listed in the statement of the theorem. QED

We now turn to the equivalence of parity formulas and transparent modal automata. The transformation of the first into the latter type of structure is the most complex construction in this section — but the hardest part of the work has already been done in section 6.6 where we discussed *guarded transformations* of parity formulas.

Proposition 10.43 *There is an effective construction that transforms a parity P-formula $\mathbb{G}$ into a transparent modal P-automaton $\mathbb{A}_{\mathbb{G}}$, such that*
1) $\mathbb{A}_{\mathbb{G}} \equiv \mathbb{G}$;
2) $|\mathbb{A}_{\mathbb{G}}| \leq 2^{\mathcal{O}(|\mathbb{G}|)}$
3) $ind(\mathbb{A}_{\mathbb{G}}) = ind(\mathbb{G})$.

Proof. Recall that by Theorem 6.63 there is an algorithm that transforms $\mathbb{G}$ into an equivalent strongly guarded parity formula $\mathbb{H}$ of size (roughly) exponential in $|\mathbb{G}|$, and index $ind(\mathbb{H}) = ind(\mathbb{G})$. Without loss of generality we may assume that every state of $\mathbb{H}$ is the successor of some modal node, cf. Remark 6.66.

The transparent modal automaton $\mathbb{A}$ will be directly based on $\mathbb{H}$. First of all, we let the carrier A of $\mathbb{A}$ be the set of successors of modal nodes, together with the initital vertex v_I, that is:

$$A := \{v_I\} \cup E[V_m].$$

Clearly then all states of $\mathbb{H}$ belong to A, and with every modal node u we may associate an element $a_u \in A$: its unique successor. We define $a_I := v_I$, and as the priority map of $\mathbb{A}$ we take the map $\Omega' : A \to \omega$ given by

$$\Omega'(a) := \begin{cases} \Omega(a) & \text{if } a \in \mathsf{Dom}(\Omega) \\ 0. & \text{otherwise} \end{cases}$$

It is left to define the transition map $\Theta : A \to \mathtt{1EML}(\mathsf{P}, A)$. Basically, for any $a \in A$ we will read off $\Theta(a)$ from a directed acyclic graph $\mathbb{D}_a := (D_a, E_a)$ that we will cut out from the underlying graph (V, E) of $\mathbb{H}$. We define D_a as the smallest subset D of V that contains a and is closed under taking E-successors of non-modal nodes (that is, if $v \in D \setminus V_n$, then $E[v] \subseteq D$). Clearly, any node $u \in D_a$ must be either modal or atomic if $E[u]$ is empty, and either boolean or silent if it is not. The relation E_a can now be defined as follows:

$$E_a := \{(u, v) \in E \cap (D_a \times D_a) \mid v \neq a\}.$$

It follows from the strong guardedness of $\mathbb{H}$ that $\mathbb{D}$ is *acyclic*, so that we may use the relation E_a for recursive definitions. (It is for this reason that we did not define E_a as the *restriction* of E to the set D_a; this would create cycles in case D_a would contain a modal node u such that Eua.) In particular, we will define a formula $\theta_a(u) \in \mathtt{1EML}$ for every $u \in D_a$:

$$\theta_a(u) := \begin{cases} L(u) & \text{if } u \text{ is atomic} \\ \heartsuit a_u & \text{if } u \text{ is modal and } L(u) = \heartsuit \\ \bigodot\{L(v) \mid Euv\} & \text{if } u \text{ is boolean and } L(u) = \odot \\ \theta_a(v) & \text{if } L(u) = \varepsilon \text{ and } Euv. \end{cases}$$

Finally, then, we define

$$\Theta(a) := \theta_a(a).$$

It is easy to verify that every formula of the form $\theta_a(u)$ is an extended modal one-step formula over P and A. This implies that $\Theta : A \to \mathtt{1EML}(\mathsf{P}, A)$ is of the required type.

It is an immediate consequence of the definitions that $|\mathbb{A}| \leq \mathbb{H}$ and $ind(\mathbb{A}) \leq ind(\mathbb{H})$; from this we obtain the items (2) and (3) of the theorem. It thus remains to prove the equivalence of $\mathbb{A}$ and $\mathbb{H}$. But a moment of reflection will show that, for any Kripke model $\mathbb{S}$, the evaluation game $\mathcal{E} := \mathcal{E}(\mathbb{H}, \mathbb{S})$ and the acceptance game $\mathcal{A} := (\mathbb{A}, \mathbb{S})$ are *isomorphic*, apart from the automatic moves of type $(a, s) \to (\Theta(a), s)$ in $\mathcal{A}$, which have no counterpart in $\mathcal{E}$. QED

Proposition 10.44 *There is an effective construction that transforms a transparent modal* P*-automaton* $\mathbb{A}$ *into a parity* P*-formula* $\mathbb{G}_{\mathbb{A}}$*, such that*

1) $\mathbb{G}_{\mathbb{A}} \equiv \mathbb{A}$*;*

2) $|\mathbb{G}_{\mathbb{A}}| = |\mathbb{A}|$*;*

3) $ind(\mathbb{G}_{\mathbb{A}}) = ind(\mathbb{A})$*.*

Proof. Given $\mathbb{A} = (A, \Theta, \Omega, a_I)$, define $\mathbb{G}_{\mathbb{A}} = (V, E, L, \Omega, v_I)$ by putting

$$\begin{array}{lll} V & := & A \cup \bigcup_{a \in A} Sf(\Theta(a)) \\ E & := & \big\{(a, \Theta(a)) \mid a \in A\big\} \cup \big(\rhd_0 \cap (V \times V)\big) \\ \Omega(v) & := & \begin{cases} \Omega(v) & \text{if } v \in A \\ \uparrow & \text{otherwise} \end{cases} \\ v_I & := & a_I, \end{array}$$

where we recall that $\rhd_0$ is the converse of the direct subformula relation $\lhd_0$. We leave it for the reader to verify that $\mathbb{G}_{\mathbb{A}}$ satisfies the conditions (1), (2) and (3). QED

10.6 Simulation Theorem

In this section we will prove the most important result of this chapter, viz., the *Simulation Theorem* stating that every modal automaton can be replaced with an equivalent disjunctive modal automaton.

Theorem 10.45 *There is a construction* sim *transforming a modal automaton* $\mathbb{A}$ *into an equivalent disjunctive modal automaton* $\mathsf{sim}(\mathbb{A})$*.*

The definition of the simulating automaton proceeds in two stages. We first come up with an automaton $\mathbb{A}^\sharp$ of which the transition map already has the right shape, but the acceptance condition is not a parity condition but a so-called *ω-regular* set over the carrier $A^\sharp$ of $\mathbb{A}^\sharp$ (i.e., a subset of $(A^\sharp)^\omega$ that itself can be recognized by some finite stream automaton with a parity acceptance condition). As we shall see, the move from $\mathbb{A}$ to $\mathbb{A}^\sharp$ involves a 'change of basis': the states of $\mathbb{A}^\sharp$ will be taken from the set $A^\sharp := \wp(A \times A)$ of binary relations over A, and the definition of the transition map $\Theta^\sharp$ of $\mathbb{A}^\sharp$ is based on various links between the one-step languages we obtain by taking A and $A^\sharp$ as sets of (formal) variables. In the second step of the construction we then show how $\mathbb{A}^\sharp$, like any automaton with an ω-regular acceptance condition, can be transformed into a standard modal automaton with a parity condition.

In fact, we shall prove a slightly more general version of Theorem 10.45, by abstracting from the precise shape of the one-step languages $\mathtt{1ML}$ and $\mathtt{1DML}$ that form the codomain of the transition function of modal and disjunctive modal automata, respectively. Our proof will only use a certain distributive law that holds between $\mathtt{1ML}(A)$ and $\mathtt{1DML}(A)$, and for future reference it will make sense to formulate our definitions and results for two arbitrary one-step languages satisfying such a distributive law.

Convention 10.46 Throughout this section we we shall be dealing with two one-step languages $\mathtt{L}_1$ and $\mathtt{L}_2$, providing sets $\mathtt{L}_i(A)$ of formulas for each set A of propositional variables.

Recall that, in line with the context of fixpoint logics that we are working in, we will assume that, for any one-step logic $\mathtt{L}$, the formulas in $\mathtt{L}(A)$ are all *monotone*. Recall as well that in Definition 10.34 we introduced the notion of an $\mathtt{L}$-automaton, and that in Table 21 we summarize the rules of the acceptance game of such automata.

Our purpose will be to prove that, under some natural constraints on the relation between two one-step languages $\mathtt{L}_1$ and $\mathtt{L}_2$, every $\mathtt{L}_1$-automaton can be *simulated by* an $\mathtt{L}_2$-automaton, that is, transformed into an equivalent $\mathtt{L}_2$-automaton. In the case where $\mathtt{L}_1 = \mathtt{1ML}$ and $\mathtt{L}_2 = \mathtt{1DML}$, the simulating language $\mathtt{1DML}$ corresponds to some *fragment* of $\mathtt{1ML}$, in which the use of conjunctions is severely restricted. Here the construction of the simulating automaton corresponds to finding a *disjunctive normal form* for the modal automata.

In order to formulate the condition on $\mathtt{L}_1$ and $\mathtt{L}_2$ under which we can prove a simulation theorem, we need some preparatory work. Informally, let $\mathtt{L}^\wedge(A)$ denote the version of the language $\mathtt{L}$ that allows conjunctions of proposition letters from A to occur at positions where $\mathtt{L}$ only allows the proposition letters from A themselves. As an example, recall that the language $\mathtt{1DML}(A)$ is built up from basic formulas ∇B, where $B \subseteq A$. Examples of formulas in $\mathtt{1DML}^\wedge(A)$ are $\nabla\{a \wedge b, b\}$ and $\bot \vee \nabla\{a_1 \wedge a_2 \wedge a_3, \top\}$. Observe that these two formulas do not belong to $\mathtt{1DML}(A)$, and thus bear witness to the fact that the latter language forms a *proper* subset of $\mathtt{1DML}^\wedge(A)$. On the other hand, it is easy to see that $\mathtt{1ML}(A) = \mathtt{1ML}^\wedge(A)$.

A convenient way of thinking about the formulas in $\mathtt{L}^\wedge(A)$ is that they are *substitution instances* of formulas in $\mathtt{L}(\wp A)$ under a special substitution θ_A. Formally we define the languare as follows.

Definition 10.47 For any set A and any language $\mathtt{L}$, we define the language

$$\mathtt{L}^\wedge(A) := \{\varphi[\theta_A] \mid \varphi \in \mathtt{L}(\wp A)\},$$

where we let θ_A denote the substitution that replaces, for any subset $B \subseteq A$, the (formal) variable B with the conjunction $\bigwedge B$. $\lhd$

As an example, we obtain the formula $\Box a \wedge \Box(a \wedge b) \in \mathtt{1ML}(A)$ from the formula $\Box\{a\} \wedge \Box\{a,b\} \in \mathtt{1ML}(\mathsf{P}, \wp A)$ by substituting $a = \bigwedge\{a\}$ for $\{a\}$, and $a \wedge b = \bigwedge\{a,b\}$ for $\{a,b\}$.

Now we can define the key condition on two languages $\mathtt{L}_1$ and $\mathtt{L}_2$, making that $\mathtt{L}_2$-automata can simulate $\mathtt{L}_1$-automata, as follows.

Definition 10.48 $\mathtt{L}_2$ is $\bigwedge$-*distributive* over $\mathtt{L}_1$ if, for each set A, and for every finite set Φ of $\mathtt{L}_1(A)$-formulas we have

$$\bigwedge \Phi \ \equiv \ \psi[\theta_A],$$

for some formula $\psi \in \mathtt{L}_2(\wp A)$. $\lhd$

Informally, $\mathtt{L}_2$ is $\bigwedge$-distributive over $\mathtt{L}_1$ if every finite conjunction of $\mathtt{L}_1(A)$-formulas is equivalent to some $\mathtt{L}_2^{\wedge}(A)$-formula. The terminology can be motivated as follows: $\mathtt{L}_2$ is $\bigwedge$-distributive over $\mathtt{L}_1$ if every conjunction of $\mathtt{L}_1$-formulas is equivalent to an $\mathtt{L}_2$-formula of conjunctions; that is, if conjunctions in $\mathtt{L}_1$ 'distribute over $\mathtt{L}_2$-formulas'. As a key example of $\bigwedge$-distributivity we have the following result, which can be proved along the same lines as Proposition 1.36.

Proposition 10.49 *$\mathtt{1DML}(A)$ is $\bigwedge$-distributive over $\mathtt{1ML}(A)$.*

The importance of the notion of $\bigwedge$-distributivity lies in the following Theorem, which obviously generalises the simulation theorem for modal automata.

Theorem 10.50 (Simulation Theorem) *Let $\mathtt{L}_1$ and $\mathtt{L}_2$ be two one-step languages such that $\mathtt{L}_2$ is $\bigwedge$-*distributive *over $\mathtt{L}_1$. Then there is an effective construction sim transforming an $\mathtt{L}_1$-automaton $\mathbb{A}$ into an equivalent $\mathtt{L}_2$-automaton $\mathsf{sim}(\mathbb{A})$.*

We now turn to the definition of the $\mathtt{L}_2$-automaton $\mathbb{A}^\sharp$ that simulates an arbitrary but fixed $\mathtt{L}_1$-automaton $\mathbb{A}$. Note that our prime example concerns a simulation theorem where the transition structure of the simulating automaton is of a significantly simpler nature than that of the simulated one. The intuition underlying the definition of $\mathbb{A}^\sharp$ is that one $\mathbb{A}^\sharp$-match will correspond to a *bundle of several $\mathbb{A}$-matches in parallel*, and that to win an $\mathbb{A}^\sharp$-match, $\exists$ has to win *each* of these parallel $\mathbb{A}$-matches. It is thus to be expected that we will obtain $\mathbb{A}^\sharp$ via some kind of power construction on $\mathbb{A}$.

For some more detail, suppose that $\exists$ is faced with a set $\{(a,s) \mid a \in B_s\}$ of positions in some $\mathbb{A}$-acceptance game, for some subset $B_s \subseteq A$ (and one single state s). She could try to respond to all challenges posed by these positions *in one go* by coming up with a single marking $m : R[s] \to \wp A$ such that $(R[s], m) \Vdash^1 \bigwedge\{\Theta(a, c_s) \mid a \in B\}$. Then for each such successor t of s, we can see $B_t = m(t)$ as the set of new challenges that she should take care of at t in parallel. In this way, we may think of a match of the simulating automaton moving in rounds, from one 'macro-position' (B_i, s_i) (corresponding to the set $\{(b, s_i) \mid b \in B_i\}$) to another 'macro-position' (B_{i+1}, s_{i+1}) (corresponding to the set $\{(b, s_{i+1}) \mid b \in B_{i+1}\}$).

This approach would suggest to take $\wp A$ as the carrier set of $\mathbb{A}^\sharp$. However, if we would simply take the states of $\mathbb{A}^\sharp$ to be *macro-states* of $\mathbb{A}$, i.e., subsets of $\mathbb{A}$, we would get into trouble when defining the acceptance condition of $\mathbb{A}$, similar to the problems one encounters when determinizing stream automata. The problem is that from a sequence $B_1B_2B_3\ldots$ of subsets of A, representing an $\mathbb{A}^\sharp$-match, we cannot recognize the set of parallel $\mathbb{A}$-matches that this sequence corresponds to. We can take an elegant way out of this problem by defining the carrier set $A^\sharp$ of $\mathbb{A}^\sharp$ to be the set of *binary relations* over A, and to link $A^\sharp$-sequences and A-sequences via the notion of a *trace* through a sequence of binary relations.

Definition 10.51 Fix a set A. We let $A^\sharp$ denote the set of binary relations over A, that is,

$$A^\sharp := \wp(A \times A).$$

Given an infinite word $\rho = R_1R_2R_3\ldots$ over the set $A^\sharp$, a *trace* through ρ is either a finite A-word $\alpha = a_0a_1a_2\ldots a_k$, or an A-stream $\alpha = a_0a_1a_2\ldots$, such that $a_iR_{i+1}a_{i+1}$ for all $i < k$ (respectively, for all $i < \omega$). Finite traces through finite $A^\sharp$-sequences are defined similarly. $\lhd$

The key idea behind the definition of $\mathbb{A}^\sharp$ and the proof of its equivalence to $\mathbb{A}$, is that with each $\mathcal{A}(\mathbb{A}^\sharp, \mathbb{S})$-match with basic positions

$$(R_1, s_1)(R_2, s_2)(R_3, s_3)\ldots$$

and each trace $a_0a_1a_2$ through $R_1R_2R_3\ldots$ we may associate an $\mathcal{A}(\mathbb{A}, \mathbb{S})$-match with basic positions

$$(a_1, s_1)(a_2, s_2)(a_3, s_3)\ldots$$

This explains the winning condition of the automaton $\mathbb{A}^\sharp$: an $A^\sharp$-stream should be winning for $\exists$ if *all* traces through it are winning according to the acceptance condition of $\mathbb{A}$.

Definition 10.52 Relative to a parity condition Ω on A, call an infinite trace $\alpha \in A^\omega$ *bad* if the maximum priority occurring infinitely often on α is an odd number. Let NBT_Ω denote the set of infinite $A^\sharp$-words that contain no bad traces relative to Ω. $\lhd$

Note that the automaton $\mathbb{A}^\sharp$ will be equipped with this set NBT_Ω as its acceptance condition, and while we will be able to establish that $\mathbb{A}^\sharp$ is equivalent to $\mathbb{A}$, NBT_Ω clearly is not a parity condition. This we will take care of in the second part of the construction.

Before giving the formal details, let us first provide some further intuitions behind the definition of $\mathbb{A}^\sharp$. Our starting point is that a state R of $\mathbb{A}^\sharp$ encodes the macro-state $\mathsf{Ran}(R) := \{b \in A \mid (a, b) \in R \text{ for some } a \in A\}$, that is, the range of R. This already suffices to motivate the definition of the initial state of $\mathbb{A}^\sharp$:

$$R_I := \{(a_I, a_I)\}.$$

In order to introduce the definition of $\Theta^\sharp : (A^\sharp \times \wp\mathsf{P}) \to \mathtt{L}_2(A^\sharp)$, consider a model $\mathbb{S}$ and a position of the form (R, s) in the acceptance game $\mathcal{G}^\sharp = \mathcal{A}(\mathbb{A}^\sharp, \mathbb{S})$. Take a state $a \in \mathsf{Ran}(R)$, then at the position (a, s) in the game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{S})$, $\exists$ has to come up with a marking $m_{a,s} : R[s] \to \wp(A)$ such that $(R[s], m_{a,s}) \Vdash^1 \Theta(a, c_s)$. Since the position (R, s) encodes the 'macro-position' $\{(a,s) \mid a \in \mathsf{Ran}(R)\}$, we need to consider all of the formulas $\Theta(a,c_s)$ (with $a \in \mathsf{Ran}(R)$) in parallel; this would suggest to consider the conjunction $\bigwedge\{\Theta(a,c_s) \mid a \in \mathsf{Ran}(R)\}$. However, in this conjunction we are no longer able to retrieve the 'origin' of a propositional variable $b \in A$. For this reason we use the following trick. We consider any pair $(a,b) \in A \times A$ as a new propositional variable, representing the variable b tagged with the 'origin' a.

Definition 10.53 Given a language $\mathtt{L}$ and a variable a, let τ_a be the substitution replacing any variable $b \in A$ with the variable $(a,b) \in A \times A$. In words, we say that τ_a *tags* each variable b with a. Given a state a of $\mathbb{A}$ and a color $c \in \wp\mathsf{P}$, let $\Theta^\star(a,c) \in \mathtt{L}_1(A \times A)$ be the formula

$$\Theta^\star(a,c) := \Theta(a,c)[\tau_a],$$

that is, each $b \in A$ occurring in $\Theta(a)$ is replaced with (a,b). $\lhd$

As an example, if $\Theta(a,c) = \Diamond a \wedge \Box b$, then $\Theta^\star(a,c) = \Diamond(a,a) \wedge \Box(a,b)$.

Using this trick we can think of a state $R \in A^\sharp$ unfolding into the formula $\bigwedge\{\Theta^\star(a,c_s) \mid a \in \mathsf{Ran}(R)\} \in \mathtt{L}_1(A \times A)$. Observe that any variable in this formula that is in the scope of a modality, must be of the form $(a,b) \in A \times A$, thus encoding a 'direct meaning' b together with its 'origin' a. Also note that any binary relation $Q \in A^\sharp$ now represents a set of (formal) variables, and so it makes sense to consider for instance the conjunction $\bigwedge Q$.

The following proposition is immediate by the definitions.

Proposition 10.54 *Let $\mathtt{L}_1$ and $\mathtt{L}_2$ be two languages such that $\mathtt{L}_2$ is $\bigwedge$-distributive over $\mathtt{L}_1$, and let A be some set. Then for every finite set Φ of formulas in $\mathtt{L}_1(A \times A)$ there is a formula $\psi \in \mathtt{L}_2(A^\sharp)$ such that*

$$\bigwedge \Phi \;\equiv\; \psi[\theta_{A\times A}], \tag{118}$$

where $\theta_{A\times A}$ is the substitution replacing every relation $Q \subseteq A \times A$ with the conjunction $\bigwedge Q$.

We are now ready for the formal definition of the automaton $\mathbb{A}^\sharp$.

Definition 10.55 Let $\mathtt{L}_1$ and $\mathtt{L}_2$ be two languages such that $\mathtt{L}_2$ is $\bigwedge$-distributive over $\mathtt{L}_1$, and let $\mathbb{A} = \langle A, \Theta, \Omega, a_I\rangle$ be an $\mathtt{L}_1$-automaton. $\mathbb{A}^\sharp$ is given as the $\mathtt{L}_2$-automaton

$$\mathbb{A}^\sharp := \langle A^\sharp, \Theta^\sharp, \mathrm{NBT}_\Omega, R_I\rangle.$$

Here $A^\sharp = \wp(A \times A)$ is the set of binary relations on A, the initial state R_I is the relation $R_I := \{(a_I, a_I)\}$. The transition function $\Theta^\sharp$ is given by fixing, for $\Theta^\sharp(R,c)$, a formula $\psi \in \mathtt{L}_2(A^\sharp)$ satisfying

$$\bigwedge\{\Theta^\star(a,c) \mid a \in \mathsf{Ran}(R)\} \;\equiv\; \psi[\theta_{A\times A}], \tag{119}$$

Finally, the acceptance condition $\mathrm{NBT}_\Omega \subseteq (A^\sharp)^\omega$ is as given in Definition 10.52. $\lhd$

The main technical result of this section concerns the following equivalence.

Proposition 10.56 *Let L_1 and L_2 be two languages such that L_2 is $\bigwedge$-distributive over L_1, and let $\mathbb{A}$ be an L_1-automaton. Then $\mathbb{A}$ is equivalent to $\mathbb{A}^\sharp$.*

A key proposition, relating the various formulas, languages and substitutions that feature in the simulation construction, is the following.

Proposition 10.57 *Let $\mathbb{A}$ be an L_1-automaton and let D be some set. Suppose that for each $a \in A$ a marking $m_a : D \to \wp A$ is given. For $R \in A^\sharp$, let $m_R : D \to \wp(A \times A)$ and $m^\sharp_R : D \to \wp(A^\sharp)$ be the markings given by*

$$\begin{array}{rcl} m_R(d) & := & \{(a,b) \mid a \in \mathsf{Ran}(R) \;\&\; b \in m_a(d)\} \\ m^\sharp_R(d) & := & \{m_R(d)\}. \end{array}$$

Then the following are equivalent, for any $c \in \wp\mathsf{P}$:

1. *$(D, m_a) \Vdash^1 \Theta(a,c)$ for each $a \in \mathsf{Ran}(R)$;*
2. *$(D, m_R) \Vdash^1 \bigwedge\{\Theta^\star(a,c) \mid a \in \mathsf{Ran}(R)\}$;*
3. *$(D, m^\sharp_R) \Vdash^1 \Theta^\sharp(R,c)$.*

We leave the (straightforward) proof of this Proposition as an exercise to the reader.

Proof of Proposition 10.56. Fix an arbitrary pointed model $(\mathbb{S}, s_0)$, then it suffices to prove that

$$\mathbb{A} \text{ accepts } (\mathbb{S}, s_0) \text{ iff } \mathbb{A}^\sharp \text{ accepts } (\mathbb{S}, s_0). \tag{120}$$

For the direction from left to right, define a position (R,s) to be *safe* if for all $a \in \mathsf{Ran}(R)$, (a,s) is winning for $\exists$ in the acceptance game $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s_0)$. Now define the following strategy for $\exists$ in $\mathcal{G}^\sharp = \mathcal{A}(\mathbb{A}^\sharp, \mathbb{S})@(R_I, s_0)$:

- If (R,s) is safe, then $\exists$ uses Proposition 10.57 to transform the set of moves $\{m_{a,s} \mid a \in \mathsf{Ran}(R)\}$, given by her winning strategy in $\mathcal{G}$, into a marking $m^\sharp_{R,s} : R[s] \to \wp A^\sharp$.
- If (R,s) is not safe, then $\exists$ plays in a random way.

It is not very hard to prove the following three claims on this strategy.

CLAIM 1 If (R,s) is safe then the moves suggested by the above strategy are legitimate.

CLAIM 2 If (R,s) is safe then all pairs (Q,t) such that $Q \in m^\sharp_{R,s}(t)$ are safe.

CLAIM 3 Consider an infinite $\mathcal{G}^\sharp$-match, guided by the above strategy for $\exists$, with basic positions $(R_I, s_0)(R_1, s_1)(R_2, s_2)\ldots$, and let $a_I a_I a_1 a_2 \ldots$ be a trace through $R_I R_1 R_2 \ldots$ Then there is an infinite $\mathcal{G}$-match, guided by $\exists$'s winning strategy, of which the basic positions are $(a_I, s_0)(a_1, s_1)(a_2, s_2)\ldots$

On the basis of these three claims, it easily follows that the given strategy is winning for $\exists$ from any safe position. In particular, it follows from the assumption that $(a_I, s_0) \in \mathrm{Win}_\exists(\mathcal{G})$ that (R_I, s_0) is safe, and hence winning for $\exists$ in $\mathcal{G}^\sharp$. This shows that $\mathbb{A}^\sharp$ accepts $(\mathbb{S}, s_0)$, as required.

The proof of the opposite direction ('$\Leftarrow$') of (120) is somewhat similar, and left as an exercise. QED

Regular automata

In the previous subsection we defined a nondeterministic automaton $\mathbb{A}^\sharp$ and proved it to be equivalent to the given automaton $\mathbb{A} = \langle A, \Theta, \Omega, a_I \rangle$. The only shortcoming of the automaton $\mathbb{A}^\sharp$ is that its acceptance condition $\mathrm{NBT}_\Omega \subseteq (A^\sharp)^\omega$ is not given by a parity function. We will now see that this problem can easily be overcome since NBT_Ω has the form of an ω-regular language over the alphabet $A^\sharp$, that is, it is recognized by some stream automaton.

Definition 10.58 An automaton $\mathbb{A} = \langle A, \Theta, Acc, a_I \rangle$ is called *ω-regular* if $Acc \subseteq A^\omega$ is an ω-regular language, i.e., if Acc is the stream language recognized by some deterministic stream automaton with a parity (or Muller) acceptance condition. $\lhd$

Here we shall prove that, given an regular automaton $\mathbb{A}$ of which the acceptance condition is given by some deterministic parity stream automaton $\mathbb{Z}$, we can effectively construct a parity automaton $\mathbb{A} \odot \mathbb{Z}$ that is equivalent to $\mathbb{A}$. First, however, we show that, indeed, $\mathbb{A}^\sharp$ is a regular automaton, by constructing a stream automaton recognizing the ω-language NBT_Ω.

Proposition 10.59 *Let A be some finite set, and let $\Omega : A \to \omega$ be a parity function on A. Then the set* NBT_Ω *is an ω-regular language over the alphabet $A^\sharp$.*

Proof. First we define a nondeterministic $A^\sharp$-stream parity automaton $\mathbb{B}$ which accepts exactly those infinite $A^\sharp$-streams that *do* contain a bad trace. Given the properties of parity stream automata it is fairly straightforward to continue from here. First, take a deterministic equivalent $\mathbb{B}'$ of $\mathbb{B}$; such an automaton exists by Theorem 4.27. And second, since $\mathbb{B}'$ is deterministic, it is easy to perform complementation on it, that is, define an automaton $\mathbb{C}$ that accepts exactly those $A^\sharp$-streams that are rejected by $\mathbb{B}'$. In short: $L_\omega(\mathbb{C}) = (A^\sharp)^\omega \setminus L_\omega(\mathbb{B}') = (A^\sharp)^\omega \setminus L_\omega(\mathbb{B})$. Clearly then $L_\omega(\mathbb{C}) = \mathrm{NBT}_\Omega$.

For the definition of $\mathbb{B}$, take an object $b_I \notin A$, and define $B := A \cup \{b_I\}$. Let $\Delta : B \times A^\sharp \to \wp(B)$ be given by putting

$$\Delta(b, R) := \begin{cases} \mathsf{Ran}(R) & \text{if } b = b_I, \\ R[b] & \text{if } b \in A, \end{cases}$$

and define Ω^{+1} by putting $\Omega^{+1}(a) := \Omega(a) + 1$ for $a \in A$, and $\Omega^{+1}(b_I) := 0$. Then $\mathbb{B}$ is the automaton $\langle B, \Delta, \Omega^{+1}, b_I \rangle$.

It is immediate from the definitions that $b_I \xrightarrow{R} a$ iff $a \in \mathsf{Ran}(R)$, that is, if there is some $a' \in A$ such that $a'Ra$. From this and the definition of Δ it follows that

$$b_I \xrightarrow{R_1} a_1 \xrightarrow{R_2} a_2 \xrightarrow{R_3} \dots$$

is a run of $\mathbb{B}$ iff there is some $a_0 \in A$ such that $a_0 a_1 a_2 \dots$ is a trace through $R_1 R_2 \dots$ Then the definition of Ω^{+1} ensures that $\mathbb{B}$ indeed accepts those $A^\sharp$-streams that contain a bad trace.
QED

It follows from Proposition 10.59 that the automaton $\mathbb{A}^\sharp$ defined in the previous section is a regular automaton. Hence we have proved the main result of this section if we can show that every disjunctive regular automaton can be replaced by a disjunctive modal automaton with a parity acceptance condition. This is what we will focus on now. In fact, we will effectively transform a nondeterministic, regular automaton $\mathbb{A}$ (of which the acceptance condition is given as the stream language recognized by some stream automaton $\mathbb{Z}$) into an equivalent parity automaton $\mathbb{A} \odot \mathbb{Z}$.

Definition 10.60 Let $\mathbb{Z} = \langle Z, \zeta, \Omega, a_I \rangle$ be a deterministic parity A-stream automaton, and let $\mathbb{A} = \langle A, \Theta, Acc, a_I \rangle$ be a disjunctive modal automaton. Then $\mathbb{A} \odot \mathbb{Z}$ is the disjunctive modal automaton given as

$$\mathbb{A} \odot \mathbb{Z} = \langle A \times Z, \Theta^{\zeta}, \Psi, (a_I, z_I) \rangle,$$

where $\Theta^{\zeta} : \big((A \times Z) \times \wp \mathsf{P}\big) \to \mathtt{1DML}(A \times Z)$ is given by

$$\Theta^{\zeta}\big((a,z),c\big) := \Theta(a,c)[(b,\zeta(z,a))/b \mid b \in A],$$

and

$$\Psi\big(a,z\big) := \Omega(z).$$

defines $\Psi : A \times Z \to \omega$. $\lhd$

Intuitively, the automaton $\mathbb{A} \odot \mathbb{Z}$ behaves like $\mathbb{A}$, with the stream automaton $\mathbb{Z}$ following and directly processing the path through $\mathbb{A}$ taken during a match of the acceptance game. More precisely, when the automaton $\mathbb{A}$ moves from state a to b, the corresponding moves of $\mathbb{A} \odot \mathbb{Z}$ are from any position (a,z) to $(b,\zeta(z,a))$, where $\zeta(z,a)$ is the state obtained from z by processing the 'letter' a. Formally, this is established by the transition structure Θ^{ζ} of the automaton $\mathbb{A} \odot \mathbb{Z}$ as follows: $\Theta^{\zeta}\big((a,z),c\big)$ is obtained from $\Theta(a,c)$ by substituting every occurrence of a $b \in A$ by the ('formal') variable $(b,\zeta(z,a)) \in A \times Z$.

Theorem 10.61 *Let $\mathbb{Z} = \langle Z, \zeta, \Omega, z_I \rangle$ be a deterministic parity stream automaton, and let $\mathbb{A} = \langle A, \Theta, Acc, a_I \rangle$ be a disjunctive modal automaton such that $Acc = L_{\omega}(\mathbb{Z})$. Then $\mathbb{A}$ and $\mathbb{A} \odot \mathbb{Z}$ are equivalent.*

▶ `Proof of Theorem 10.61 to be supplied`

Finally, for the proof of the Simulation Theorem we need to combine various results obtained in this Chapter.

Proof of Theorem 10.50. It follows from the Propositions 10.49, 10.56 and 10.59 that every modal automaton can be simulated by a disjunctive, regular automaton. Then the Simulation Theorem follows by combining this observation with Theorem 10.61. QED

Notes

▶ `TBS`

Exercises

Exercise 10.1 Show that the 'slow' acceptance discussed in Remark 10.10 is equivalent to the standard acceptance game of Definition 10.5.

Exercise 10.2 Give a direct, game-theoretic argument proving Theorem 10.12. That is, show that modal automata are bisimulation invariant.

Exercise 10.3 Show the equivalence of the two notions of disjunctive modal automata as discussed in Remark 10.14. That is, give a construction that transforms an arbitrary disjunctive modal automaton into a $\mathtt{1DML}_r$-automaton.

Exercise 10.4 Let $\mathbb{A}$ be a disjunctive modal automaton, and let $(\mathbb{S}, r)$ be a finite pointed Kripke model. Show that $\mathbb{S}, r \Vdash \mathbb{A}$ iff there is a *finite* pointed model $(\mathbb{S}', r')$ such that $\mathbb{S}, r \leftrightarroweq (\mathbb{S}', r')$ and $\mathbb{S}', r' \Vdash_s \mathbb{A}$.

Exercise 10.5 Show that the one-step languages $\mathtt{1FO}$ and $\mathtt{1FOE}$ are closed under taking boolean duals.

Exercise 10.6 Prove Proposition 10.38

Exercise 10.7 Prove Proposition 10.57.

Exercise 10.8 Prove equivalence (120) in the proof of Proposition 10.56.