Decidability of **S2S**

MSc Thesis (*Afstudeerscriptie*)

written by

Christian Kissig
(born May 29th, 1981 in Riesa, Germany)

under the supervision of **Dr. Yde Venema** and **Dr. Clemens Kupke**, and submitted to the Board of Examiners in partial fulfillment of the requirements for the degree of

MSc in Logic

at the *Universiteit van Amsterdam*.

Date of the public defense:	**Members of the Thesis Committee:**
August 29th, 2007	Prof. Dr. Peter van Emde Boas
	Prof. Dr. Dick de Jongh
	Dr. Clemens Kupke
	Dr. Yde Venema

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Contents

Chapter 1

Introduction

Leibniz hoped for a calculus of truth in a universal sense. Gödel showed with his Incompleteness Theorem that such a calculus can not exist. Gödels result, however, left open which particular logics are (un)decidable. For instance, it was not until the 1970s that the undecidability of Hilbert's Tenth Problem was shown.

Besides decidability, researchers were interested in model checking properties of infinite structures using finite automata. See [CGP99] for reference. In particular Elgot and Büchi pursued the question which properties can be expressed and verified by automata. In 1962, Elgot [Elg61] and Büchi [Büc62] showed that finite state deterministic automata and monadic second-order logic interpreted on finite words, which is the weak monadic second-order logic **WS1S**, are equally expressive. In [Büc62], Büchi showed the same for the monadic second-order logic **S1S** of infinite words. Thereby he obtained an effective translation of formulas of **S1S** into non-deterministic word automata with Büchi acceptance condition. In [McN66] McNaughton generalised Büchi's method and obtained an effective translation of monadic second-order logic **S2S** of binary trees into finite automata.

Using the connection between logics and automata exhibited by Büchi and McNaughton, Rabin proved the decidability of **S2S** by reducing the satisfiability problem to the non-emptiness problem for non-deterministic binary tree automata with Rabin acceptance condition, which was known to be effectively solvable. From this result follow immediately various decidability results such as for Presburger arithmetic, for the monadic second-order logic of trees with arbitrary (but countable) branching, and for the monadic second-order logic of countable linear orderings. The most important contribution, however, is seen in the application of automata theory to logic.

The proof itself has since been considered hard to comprehend by many scholars, which lead to various improvements. In 1982, Gurevich and Harrington [GH82] gave a reduction of **S2S** to non-deterministic automata with Muller acceptance condition. In 1995, Muller and Schupp [MS95] refined the reduction. At the heart of their argument is the non-determinisation of alternating automata. As argued in [CKS81], alternating automata are closer to classical logics with negation like **S2S** than non-deterministic automata.

The behaviour of automata operating on infinite input is given in terms of accep-

tance games. For Rabin and Muller automata, these acceptance games are not historyfree, but bounded memory determined. In order to win a play of such an acceptance game the players need to have a bounded memory of the play. To quantify the memory needed, Gurevich and Harrington introduced the latest appearance record (LAR) which Muller and Schupp adapted as the index appearance record (IAR) for a complex memory framework. However, for parity automata, acceptance games are historyfree determined which follows from a result for parity graph games shown independently by Emerson and Jutla [EJ91], and Mostowski [Mos91]. We elaborate on determinacy of parity graph games in Appendix B. For the proof that acceptance games for parity automata are historyfree determined we refer to the literature.

Both infinite Σ-words and Σ-labelled binary trees are instances of coalgebras for the respective functors $\Sigma \times (-)$ taking objects X to $\Sigma \times X$ and $\Sigma \times ((-) \times (-))$ taking objects X to $\Sigma \times (X \times X)$. Based on this observation, Venema introduced in [Ven04] automata recognising general F-coalgebras for functors F preserving weak pullbacks. In both [KV05] and [KV07] the class of alternating F-coalgebra automata is shown to be closed under union, existential projection and non-determinisation.

In this text we give a comprehensive proof of Rabin's Theorem majorly based on the concepts used in [KV07]. In Chapter 2 we introduce monadic second-order logic **S2S** in a minimal representation and alternating automata. We prove the class of alternating binary tree automata closed under complementation in Chapter 3. In Chapter 4 we prove that alternating and non-deterministic binary tree automata are equally expressive by reducing the non-determinisation of alternating binary tree automata to the determinisation of non-deterministic word automata as shown by Safra in [Saf88]. In Chapter 5 we define the translation of formulas of **S2S** into alternating binary tree automata. In Chapter 7 we give a game-theoretical solution to the non-emptiness problem for alternating binary tree automata.

Chapter 2

Preliminaries

2.1 Monadic Second-Order Logic

Formulas of the monadic second-order logic **S2S** are defined as

$$\phi ::= X \subseteq Y \mid Suc_d(X, Y) \mid \neg\phi \mid \phi \vee \phi \mid \exists X.\phi$$

where X and Y are set variables. A variable occurs **freely** in ϕ if it is not bound by a quantifier. Formally we define the set of free variables in a formula ϕ of **S2S** inductively as follows.

$$\begin{array}{ll} FV(X \subseteq Y) & := \{X, Y\} \\ FV(Suc_d(X, Y)) & := \{X, Y\} \\ FV(\neg\phi) & := FV(\phi) \\ FV(\phi \vee \psi) & := FV(\phi) \cup FV(\psi) \\ FV(\exists X.\phi) & := FV(\phi) \setminus \{X\} \end{array}$$

Formulas ϕ are interpreted with respect to **valuations** of the variables occurring freely in ϕ. A valuation of a set variable X is a set $v_X \subseteq 2^*$ of nodes in a binary tree. A valuation of a set V of set variables is a function $v : V \to \mathcal{P}(2^*)$ taking each set variable in V to its respective interpretation. Note that such a valuation v is isomorphic to a function $T : 2^* \to \mathcal{P}(V)$ by $X \in T(p)$ *iff* $p \in v(X)$ for all variables $X \in dom(v)$ and nodes $p \in 2^*$. Intuitively, T takes each node p in a binary tree to a set of variables X in whose valuation $v(X)$ p is. Such a function T is a full binary $\mathcal{P}(V)$-labelled tree[1]. Thus we refer to T as the **tree representation** of v.

[1] Compare with Definition A.2.

Let ϕ be a formula of **S2S** with free variables from a set V of set variables. Given a valuation v of the variables in V, the **semantics** of ϕ is defined as follows.

$\models_v X \subseteq Y$	iff	$v(X) \subseteq v(Y)$
$\models_v suc_d(X, Y)$	iff	for all $p \in 2^*$, if $p \in v(X)$ then $pd \in v(Y)$
$\models_v \neg\phi$	iff	$\not\models_v \phi$
$\models_v \phi \vee \psi$	iff	$\models_v \phi$ or $\models_v \psi$
$\models_v \exists X.\phi$	iff	$\models_{v[X \mapsto v_X]} \phi$ for some valuation $v_X \subseteq 2^*$ of X

where $v[X \mapsto v_X]$ is the valuation of the variables in $V \cup \{X\}$ defined by $v[X \mapsto v_X](X) := v_X$ and $v[X \mapsto v_X](Y) := v(Y)$ for all variables $Y \in dom(v)$ that are not X. A binary $\mathcal{P}(V)$-labelled tree T satisfies ϕ, written $T \models \phi$, if it is the tree representation of a valuation satisfying ϕ.

A formula ϕ is **satisfiable** if there is a valuation of the free variables of ϕ or equivalently a binary $\mathcal{P}(FV(\phi))$-labelled tree satisfying ϕ.

2.2 Rabin's Theorem

The problem of deciding whether a formula ϕ of **S2S** is satisfiable, is commonly referred to as the **Satisfiability Problem of S2S**.

Theorem 2.1 (Rabin)**.** *The satisfiability problem for the monadic second-order logic **S2S** of two successors is decidable.*

In his proof [Rab69], Rabin exhibited a connection between formulas ϕ of **S2S** and non-deterministic binary tree automata $\mathbb{A}$ such that $\mathbb{A}$ accepts a binary $\mathcal{P}(FV(\phi))$-labelled tree T iff T satisfies ϕ. Thus ϕ is satisfiable, if there is such a binary tree T accepted by $\mathbb{A}$. Whence the satisfiability problem for ϕ reduces to the non-emptiness problem of $\mathbb{A}$. For alternating binary tree automata the non-emptiness problem is known to be effectively decidable.

A formula ϕ of **S2S** as above is interpreted with respect to valuations which are isomorphic to $\mathcal{P}(FV(\phi))$-labelled trees recognised by automata. In the following we introduce binary tree automata not just for the set $\mathcal{P}(FV(\phi))$, but for arbitrary alphabets Σ.

Definition 2.2 (Alphabets)**.** *An **alphabet** is a set whose elements are called **letters**.*

Position	Player	Admissible Moves	Priority
$(p, a) \in 2^* \times A$	$\exists$	$\{(p, \Phi) \mid \Phi \in \delta(a, T(p))\}$	$\Omega(a)$
$(p, \Phi) \in 2^* \times \mathcal{P}(A)$	$\forall$	$\{(p, (a_0, a_1)) \mid (a_0, a_1) \in \Phi\}$	0
$(p, (a_0, a_1)) \in 2^* \times A \times A$	$\forall$	$\{(pd, a_d) \mid d \in 2\}$	0

Table 2.1: Acceptance Game for Alternating Binary Tree Automata

2.3 Binary Tree Automata

We introduce binary tree automata and word automata for the use in subsequent chapters.

Definition 2.3 (Alternating Binary Tree Parity Automata)**.** *Given an alphabet Σ, an alternating binary tree parity automaton is a tuple*

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega : A \to \omega \rangle$$

consisting of

- *a finite set A of states,*
- *a transition function δ,*
- *an initial state $a_I \in A$, and*
- *a priority function Ω assigning automaton states natural numbers*

Alternating binary tree automata $\mathbb{A}$ as above recognise binary Σ-labelled trees. Whether such an automaton $\mathbb{A}$ accepts a tree T is defined in terms of the acceptance game $\mathcal{G}(\mathbb{A}, T)$, a parity graph game which proceeds according to the following rules.

1. At position $(p, a) \in 2^* \times A$, $\exists$ is to choose a set $\Phi \subseteq A$ of states from $\delta(a, T(p))$ which determines the next position (p, Φ).
2. At position $(p, \Phi) \in 2^* \times \mathcal{P}(A)$, $\forall$ is to choose a pair (a_0, a_1) of states from Φ which yields the next position $(p, (a_0, a_1))$.
3. At position $(p, (a_0, a_1)) \in 2^* \times (A \times A)$, $\forall$ is to choose a direction $d \in \{0, 1\}$ and effectively a successor state a_d. The next position is (pd, a_d).

The above steps constitute one round in the acceptance game $\mathcal{G}(\mathbb{A}, T)$. Within each round we distinguish two stages. We call the first stage, which consists of the first two steps, **static** because the position in the input tree is not changed. In the static stage the players determine the next automaton states. We call the second stage, which consists of the third step only, **dynamic** because it determines the next position in the tree.

We call positions from $2^* \times A$ **basic**. The initial position is the basic position (ϵ, a_I). Positions from $2^* \times A$ belong to $\exists$. All other positions from $2^* \times \mathcal{P}(A \times A)$ and $2^* \times (A \times A)$ belong to $\forall$. Basic positions (p, a) are assigned the priority $\Omega(a)$, whereas all other positions are assigned priority 0. $\exists$ ($\forall$) wins a finite play π in $\mathcal{G}(\mathbb{A}, T)$ if the last position in π belongs to $\forall$ ($\exists$) and he (she) can not move. $\exists$ ($\forall$) wins an infinite play $\mathcal{G}(\mathbb{A}, t)$ if the largest priority of a state occurring infinitely often is even (odd).

$\mathbb{A}$ accepts a binary Σ-labelled tree T if the initial position (ϵ, a_I) is winning for $\exists$ in the game $\mathcal{G}(\mathbb{A}, T)$, and rejects T otherwise. The set of binary trees accepted by $\mathbb{A}$ is called the **language** of $\mathbb{A}$ and is denoted by $\mathcal{L}(\mathbb{A})$. Two automata $\mathbb{A}$ and $\mathbb{B}$ are **equivalent** if they have the same language.

Definition 2.4 (Non-Deterministic Binary Tree Automata)**.** *Given an alphabet Σ, a non-deterministic binary tree automaton is a tuple*

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{P}(A \times A), a_I, \Omega \rangle$$

consisting of

- *a finite set A of states,*
- *a transition function δ,*
- *an initial state $a_i \in A$, and*
- *a priority function $\Omega : A \to \omega$ assigning automaton states natural numbers.*

In fact non-deterministic are special cases of alternating automata. An alternating binary tree automaton

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{P}\mathcal{P}(A \times A), a_I, \Omega \rangle$$

is called **non-deterministic** if for all automaton states $a \in A$ and letters $c \in \Sigma$, $\delta(a, c)$ consists of singletons. Additionally we define **co-non-deterministic** binary automata as alternating binary tree automata such that for all $a \in A$ and $c \in \Sigma$, $\delta(a, c)$ is a singleton or empty. Thereby we obtain characterisations of the acceptance games for non-deterministic and co-non-deterministic automata, such that respectively $\forall$'s and $\exists$'s choice in the static stages is bounded in the acceptance games for non-deterministic and co-non-deterministic automata. Deterministic binary tree automata are both, non-deterministic and co-non-deterministic.

2.4 Word Automata

Word automata are defined similarly as binary tree automata. Next we are going to formally introduce word automata.

Definition 2.5 (Non-deterministic Parity Word Automata)**.** *Given an alphabet* Σ*, a non-deterministic parity word automaton is a tuple*

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{P}(A), a_I, \Omega : A \to \omega \rangle$$

consisting of

- *a finite set* A *of states,*
- *a transition function* δ*,*
- *an initial state* $a_I \in A$*, and*
- *a priority function* Ω *assigning automaton states natural numbers.*

A **run** of a non-deterministic parity word automaton on an infinite Σ-word U, is a sequence π of automaton states from A, such that $\pi(0) = a_I$ and for all $i < |\pi| - 1$, $\pi(i+1) \in \delta(\pi(i), U(i))$. Such a run is **accepting** if it is infinite and the largest priority of an automaton state occurring infinitely often is even, and **rejecting** otherwise. A non-deterministic parity word automaton $\mathbb{A}$ accepts an infinite Σ-word U if there is an accepting run of $\mathbb{A}$ on U.

Definition 2.6 (Deterministic Parity Word Automata)**.** *Given an alphabet* Σ*, a deterministic parity word automaton is a tuple*

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to A, a_I, \Omega : A \to \omega \rangle$$

consisting of

- *a finite set* A *of states,*
- *a transition function* δ*,*
- *an initial state* $a_I \in A$*, and*
- *a priority function* Ω *assigning automaton states natural numbers.*

Similarly to non-deterministic word automata acceptance is defined for deterministic word automata. A **run** of a deterministic parity word automaton $\mathbb{A}$ as above on an infinite Σ-word U is a sequence π of automaton states from A, such that $\pi(0) = a_I$ and $\delta(\pi(i), U(i)) = \{\pi(i+1)\}$ for all $i < |\pi| - 1$. Such a run π is **accepting** if it is infinite and the largest priority of an automaton state occurring infinitely often in π is even, and rejecting otherwise. $\mathbb{A}$ accepts an infinite Σ-word U if there is an accepting run of $\mathbb{A}$ on U.

2.5 Size of Automata

We distinguish two measures of word or binary tree automata $\mathbb{A} = \langle A, \Sigma, \delta, a_I, \Omega \rangle$,

1. the size $|A|$ of the state set A, which we refer to as the **size** of $\mathbb{A}$, and
2. the size $|ran(\Omega)|$ of the range of the priority function, which we call the **index** of $\mathbb{A}$.

The index of $\mathbb{A}$ is bounded by the size of $\mathbb{A}$, that is $|ran(\Omega)| \leq |A|$, so that the index of a finite state automaton is finite.

Chapter 3

Complementing Alternating Binary Tree Automata

In [CKS81] Chandra, Kozen, and Stockmeyer argued that alternating automata are closer than non-deterministic automata to classical logic with negation like **S2S**. In this chapter we devise an algorithm for the complementation of binary tree automata which is based on the structure of the acceptance game for alternating binary tree automata. In Section 5.3, we will see that the complementation of automata corresponds to the negation of formulas.

For the rest of this chapter let

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega \rangle$$

be an alternating binary tree automaton.

An automaton $\overline{\mathbb{A}}$ **complementary** to $\mathbb{A}$ is an alternating automaton which accepts precisely the binary trees rejected by $\mathbb{A}$, that is

$$\mathcal{L}(\overline{\mathbb{A}}) = (\Sigma)^{2^*} \setminus \mathcal{L}(\mathbb{A}).$$

In terms of the acceptance games, $\exists$ has for every binary Σ-labelled tree T a winning strategy in the game $\mathcal{G}(\mathbb{A}, T)$ from the initial position iff $\forall$ has a winning strategy in the game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position.

3.1 Dualising Acceptance Games

Acceptance games $\mathcal{G}(\mathbb{A}, T)$ are parity graph games. For an arbitrary parity graph game $\mathcal{G}$ we can define the dual which is a parity graph game $\overline{\mathcal{G}}$ with the following property. $\exists$ has a winning strategy in $\overline{\mathcal{G}}$ from the initial position iff $\forall$ has a winning strategy in $\mathcal{G}$ from the initial position.

Definition 3.1 (Dual Games of Parity Graph Games)**.** *Given a parity graph game*

$$\mathcal{G} = \langle B_\exists, B_\forall, E, b_I, \Omega \rangle,$$

we define the ***dual game***

$$\overline{\mathcal{G}} := \langle \overline{B_\exists}, \overline{B_\forall}, E, b_I, \overline{\Omega} \rangle$$

by swapping the positions of $\exists$ *and* $\forall$

$$\overline{B_\exists} := B_\forall \text{ and } \overline{B_\forall} := B_\exists,$$

and complementing the acceptance condition

$$\overline{\Omega}(b) := \Omega(b) + 1 \text{ for all } b \in B_\exists \cup B_\forall.$$

Note that the move functions and the initial board positions coincide in $\mathcal{G}$ *and* $\overline{\mathcal{G}}$*.*

Lemma 3.2. $\exists$ *has a winning strategy from the initial position* b_I *in the parity graph game* $\mathcal{G}$ *iff* $\forall$ *has a winning strategy from the initial position* b_I *in the dual game* $\overline{\mathcal{G}}$*.*

Proof. Because the move functions of $\mathcal{G}$ and $\overline{\mathcal{G}}$ coincide, every play in $\overline{\mathcal{G}}$ is a play in $\mathcal{G}$. Because furthermore the positions of $\exists$ and $\forall$ are swapped, every strategy of $\exists$ in $\mathcal{G}$ is a strategy for $\forall$ in $\overline{\mathcal{G}}$. Moreover, every play in $\overline{\mathcal{G}}$ from the initial position consistent with a strategy of $\forall$ is a play in $\mathcal{G}$ played by $\exists$ consistently with the same strategy. Because the positions of $\exists$ and $\forall$ are swapped, the last position b of a finite play in $\mathcal{G}$ belongs to $\exists$ iff b belongs to $\forall$ in the game $\overline{\mathcal{G}}$. Every finite play won by $\exists$ in $\mathcal{G}$ is won by $\forall$ in $\overline{\mathcal{G}}$. Because the acceptance condition is complemented, every infinite play in $\mathcal{G}$ won by $\exists$ is won by $\forall$ in $\overline{\mathcal{G}}$. Hence $\exists$'s winning strategy in $\mathcal{G}$ from the initial position is a winning strategy for $\forall$ in the dual game $\overline{\mathcal{G}}$ from the initial position. The converse follows from a symmetric argument. □

However, we do not obtain a definition of $\overline{\mathbb{A}}$ directly from the dual games $\overline{\mathcal{G}(\mathbb{A}, T)}$ of acceptance games $\mathcal{G}(\mathbb{A}, T)$, because the dual games $\overline{\mathcal{G}(\mathbb{A}, T)}$ are in general not of the shape of acceptance games for any alternating binary tree automaton. First, in acceptance games the initial position belongs to $\exists$, whereas in the dual games the initial position belongs to $\forall$. Second, $\forall$ is to choose the direction in acceptance games, whereas $\exists$ is to choose the direction in the dual games. We will see that both can be resolved by trivialising the choice of direction.

3.2 Direction Normal Form

In the following we show how to compute from an alternating binary tree automaton $\mathbb{A}$ an equivalent alternating binary tree automaton $\mathbb{A}' = \langle A', \Sigma, \delta', a_I, \Omega' \rangle$, such that $\forall$'s choice for direction in the dynamic stage in every round of an acceptance game $\mathcal{G}(\mathbb{A}', T)$ is predetermined by the choice of automaton states in the static stage of the same round.

For the construction we introduce the special state $a_\exists$, so that $A' := A \cup \{a_\exists\}$, with transitions $\delta'(a_\exists, c) := \{\{(a_\exists, a_\exists)\}\}$ for every letter $c \in \Sigma$ and an even priority $\Omega'(a_\exists)$. Once entering $a_\exists$ in a play π of an acceptance game $\mathcal{G}(\mathbb{A}', T)$, the automaton remains in the state $a_\exists$ which will then be the only state occurring infinitely often. Because the priority $\Omega'(a_\exists)$ is even, $\exists$ wins any such play π. Consequently, $\forall$ should avoid $a_\exists$, or looses otherwise.

When every pair (a_0, a_1) in every set in the range of the transition function δ is replaced by two pairs $(a_0, a_\exists)$ and $(a_\exists, a_1)$, $\forall$ has to decide for the successor state and for the direction at once in the static stage. We say that an automaton with such a transition function is in direction normal form.

Definition 3.3 (Direction Normal Form)**.** *Let*

$$\mathbb{A}' = \langle A', \Sigma, \delta : A' \times \Sigma \to \mathcal{PP}(A' \times A'), a_I, \Omega' \rangle$$

be an alternating binary tree automaton. $\mathbb{A}'$ *is in* ***direction normal form*** *if*

1. A' *contains the state* $a_\exists$ *with transitions* $\delta'(a_\exists, c) = \{\{(a_\exists, a_\exists)\}\}$ *for all* $c \in \Sigma$ *and with an even priority* $\Omega'(a_\exists)$
2. *the range of* δ' *consists of sets of pairs* (a_0, a_1) *where* $a_0 = a_\exists$ *or* $a_1 = a_\exists$

Notation 3.4. *For convenience we denote pairs* (a_0, a_1) *where* $a_{1-d} = a_\exists$ *by* $\bigcirc_d a_d$ *from now on.*

Lemma 3.5. *Every alternating binary tree automaton is equivalent to one in direction normal form.*

Proof. Let

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega \rangle$$

be an alternating binary tree automaton, we define the alternating binary tree automaton

$$\mathbb{A}' := \langle A', \Sigma, \delta' : A' \times \Sigma \to \mathcal{PP}(A' \times A'), a_I, \Omega[a_\exists \mapsto 0] \rangle^1$$

with states

$$A' := A \cup \{a_\exists\},$$

and transitions

$$\delta'(a_\exists, c) := \{\{(a_\exists, a_\exists)\}\}$$

for all $c \in \Sigma$, and

$$\delta'(a, c) := \{\{\bigcirc_d a_d \mid (a_0, a_1) \in \Phi, d \in 2\} \mid \Phi \in \delta(a, c)\}$$

for all $a \in A$ and $c \in \Sigma$. The automaton $\mathbb{A}'$ is in direction normal form. We inductively show for any binary Σ-labelled tree that $(*)$ $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{A}, T)$ from the initial position (ϵ, a_I) iff she has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{A}', T)$ from the initial postion (ϵ, a_I).

[1] By $\Omega[a_\exists \mapsto 0]$ we denote the priority function mapping $a_\exists \mapsto 0$ and $a \mapsto \Omega(a)$ for all $a \in dom(\Omega)$ not being $a_\exists$.

$$(p,a) \xrightarrow{\exists} (p,\Phi) \xrightarrow{\forall} (p,(a_0,a_1)) \xrightarrow{\forall} (pd,a_d)$$

$$(p,a) \xrightarrow{\exists} (p,\Phi') \xrightarrow{\forall} (p,\bigcirc_d a_d) \xrightarrow{(\forall)} (pd,a_d)$$

Figure 3.1: A round of $\mathcal{G}(\mathbb{A},T)$ and $\mathcal{G}(\mathbb{A}',T)$

For the left-to-right implication, suppose $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{A},T)$ from the initial position, we derive a strategy for her in the game $\mathcal{G}(\mathbb{A}',T)$ such that every play consistent with the derived strategy is over the same basic positions as a play consistent with $\exists$'s winning strategy in the game $\mathcal{G}(\mathbb{A},T)$.

Initially all plays in $\mathcal{G}(\mathbb{A}',T)$ and $\mathcal{G}(\mathbb{A},T)$ are in the same position. Let π' be a finite partial play in $\mathcal{G}(\mathbb{A}',T)$ from the initial position consistent with the derived strategy such that the last position is a basic position (p,a). By the induction hypothesis there is a partial play π of $\mathcal{G}(\mathbb{A},T)$ over the same basic positions as π'.

Let Φ be the set $\exists$ chooses from $\delta(a,T(p))$ at position (p,a) in the game $\mathcal{G}(\mathbb{A},T)$ consistently with her winning strategy. By construction of δ', in $\delta'(a,T(p))$ there is a set $\Phi' = \{\bigcirc_d a_d \mid (a_0,a_1) \in \Phi \text{ and } d \in 2\}$. At the basic position (p,a) of the game $\mathcal{G}(\mathbb{A}',T)$, let $\exists$ choose Φ'. Let $\bigcirc_d a_d$ be the pair that $\forall$ chooses from Φ'. We assume $\forall$ to choose d for the direction as he needs to avoid loss. Then the next basic position in π' is (pd,a). Then there is a pair (a_0,a_1) in Φ. The basic position (pd,a_d) is thus also a continuation of the play π consistent with $\exists$'s winning strategy.

We have established that for every play π' of $\mathcal{G}(\mathbb{A}',T)$ from the initial position played consistently with the derived strategy, there is a play π of $\mathcal{G}(\mathbb{A},T)$ from the initial position consistent with $\exists$'s winning strategy which is over the same basic positions. Because $\exists$ wins π, the largest priority of an automaton state occurring infinitely often in π is even. Thus the largest priority of an automaton state occurring infinitely often in π' is even as well, so that $\exists$ wins π', which proves the derived strategy winning for her.

The right-to-left implication of $(*)$ follows analogously. Suppose $\exists$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}',T)$ from the initial position, we derive a strategy for her in the game $\mathcal{G}(\mathbb{A},T)$ from the initial position such that for every play in $\mathcal{G}(\mathbb{A},T)$ from the initial position consistent with the derived strategy there is a play in $\mathcal{G}(\mathbb{A}',T)$ from the initial position consistent with $\exists$'s winning strategy.

Initially all plays of $\mathcal{G}(\mathbb{A}',T)$ and $\mathcal{G}(\mathbb{A},T)$ are in the same position. Let π be a finite partial play of $\mathcal{G}(\mathbb{A},T)$ from the initial position consistent with the derived strategy such that the last position is a basic position (p,a). By the induction hypothesis there is a partial play π' in the game $\mathcal{G}(\mathbb{A}',T)$ from the initial position consistent with $\exists$'s winning strategy over the same basic positions as π. Let Φ' be the set that $\exists$ chooses at the position (p,a) consistent with her winning strategy. By the construction of δ' there is a set Φ in $\delta(a,c)$ such that $\Phi' = \{\bigcirc_d a_d \mid (a_0,a_1) \in \Phi \text{ and } d \in 2\}$. Let $\exists$ choose Φ in last position (p,a) of the play π. From Φ, $\forall$ draws a pair (a_0,a_1) and determines the direction d, the next basic position is then (pd,a_d). Then there is pair $\bigcirc_d a_d$ in Φ', so

that (pd, a_d) is an admissible continuation of the play π' consistent with $\exists$'s winning strategy.

We have established that for every play π of $\mathcal{G}(\mathbb{A}, T)$ from the initial position consistent with the derived strategy, there is a play π' of $\mathcal{G}(\mathbb{A}', T)$ from the initial position consistent with $\exists$'s winning strategy over the same basic positions. Because $\exists$ wins π', it satisfies that the largest priority of an automaton state occurring infinitely often in π' is even. Because π and π' are over the same basic positions, the largest priority of an automaton state occurring infinitely often is even as well, so that $\exists$ wins π. Thus the derived strategy is winning for $\exists$. □

As a result of the previous lemma, we can and indeed will assume $\mathbb{A}$ to be in direction normal form.

3.3 Complementation

In the following we are going to construct an automaton $\overline{\mathbb{A}}$ accepting the binary Σ-labelled trees which are rejected by $\mathbb{A}$. In terms of the acceptance games, $\exists$ will have a winning strategy in the acceptance game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position iff $\forall$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{A}, T)$ from the initial position which is iff $\exists$ has a winning strategy in the dual game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position, by Lemma 3.2.

Moreover, given a winning strategy for $\exists$ in the game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position, we want $\exists$ to have a strategy in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$, consistent with which every play is over the same basic positions as a play in $\mathcal{G}(\mathbb{A}, T)$ consistent with $\exists$'s winning strategy. Given a winning strategy for $\exists$ in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position, we want $\exists$ to have a winning strategy in the game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position such that every play from the initial position consistent with that strategy is over the same basic positions as a play in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$ consistent with $\exists$'s winning strategy.

Let $\overline{\mathbb{A}}$ have the same initial state as $\mathbb{A}$, then the initial positions coincide in the acceptance games $\mathcal{G}(\overline{\mathbb{A}}, T)$ and $\overline{\mathcal{G}(\mathbb{A}, T)}$ for any binary tree T.

Suppose $\exists$ has a winning strategy in the acceptance game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position. Let (p, a) be a basic position in a play consistent with her winning strategy. For any Φ that $\forall$ could choose from $\delta(a, T(p))$, $\exists$ has a winning choice for a pair (a_0, a_1). This winning choice can be though of as a substrategy $f : \mathcal{P}(A \times A) \to (A \times A)$ of $\exists$'s winning strategy local to the node p. The range of f is a set Φ' consisting of the pairs of automaton states that are chosen consistently with $\exists$'s winning strategy. $\forall$'s choice for Φ then determines at least one pair of automaton states which is an element of $\Phi \cap \Phi'$. The construction of Φ' gives rise to the following definition of the complementary automaton $\overline{\mathbb{A}}$.

Definition 3.6 (Complementation of an Alternating Binary Tree Automaton)**.** *Let*

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega \rangle.$$

be an alternating binary tree automaton in direction normal form, we define the complementation

$$\overline{\mathbb{A}} := \langle A, \Sigma, \overline{\delta} : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \overline{\Omega} \rangle$$

where the transition function $\overline{\delta}$ *is defined such that for all* $a \in A$ *and all* $c \in \Sigma$*,*

$$\overline{\delta}(a, c) := \{ran(f) \mid f : \delta(a, c) \to \bigcup \delta(a, c) \text{ such that } \forall \Phi \in \delta(a, c). f(\Phi) \in \Phi\}.$$

If $\delta(a, c)$ is empty, then there is precisely one such choice function, videlicet the empty one. Then $\overline{\delta}(a, c)$ contains precisely the empty set $\emptyset$. If $\delta(a, c)$ contains the empty set then there is no choice function f, so that $\overline{\delta}(a, c)$ is empty.

The complement $\overline{\mathbb{A}}$ rejects precisely the trees which are accepted by $\mathbb{A}$.

Proposition 3.7. *Let* $\mathbb{A}$ *be in direction normal form,* $\mathbb{A}$ *accepts precisely the binary* Σ*-labelled trees rejected by* $\overline{\mathbb{A}}$*.*

Proof. In the following fix a binary Σ-labelled tree T. We show that $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position (ϵ, a_I) iff $\forall$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}, T)$ from the initial position (ϵ, a_I). By Lemma 3.2, $\forall$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}, T)$ from the initial position iff $\exists$ has a winning strategy in the dual game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position. It thus remains to show that $(*)$ $\exists$ has a winning strategy in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position iff she has a winning strategy in the game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position.

For the left-to-right direction of $(*)$ suppose $\exists$ has a winning strategy in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$ from the initial position (ϵ, a_I), we derive a strategy for her in the game $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position (ϵ, a_I) with the following property. Every play in $\mathcal{G}(\mathbb{A}, T)$ from the initial position consistent with the derived strategy is over the same basic positions as a play in $\mathcal{G}(\overline{\mathbb{A}}, T)$ consistent with $\exists$'s winning strategy.

Initially, the plays of $\overline{\mathcal{G}(\mathbb{A}, T)}$ and $\mathcal{G}(\overline{\mathbb{A}}, T)$ are in the same basic position. Let π be an finite partial play in $\mathcal{G}(\overline{\mathbb{A}}, T)$ from the initial position consistent with the derived strategy and let (p, a) be the last basic position. By the induction hypothesis there is a partial play π' of $\overline{\mathcal{G}(\mathbb{A}, T)}$ consistent with $\exists$'s winning strategy over the same basic positions as π. For every set Φ that $\forall$ could choose at position (p, a) in $\overline{\mathcal{G}(\mathbb{A}, T)}$, $\exists$ has a winning choice for an element $f_\Phi \in \Phi$. The set $\Phi' = \{f_\Phi \mid \Phi \in \delta(a, T(p))\}$ of all her winning choices is an element of $\overline{\delta}(a, T(p))$ by construction. Let $\exists$ indeed choose Φ' and let $\bigcirc_d a'$ be $\forall$'s choice from Φ'. Then he should choose the direction d or looses otherwise. The next basic position in the play of $\mathcal{G}(\overline{\mathbb{A}}, T)$ would then be (pd, a').

The pair $\bigcirc_d a'$ chosen by $\forall$ is a winning choice f_Φ of $\exists$ for some $\Phi \in \delta(a, T(p))$ which $\forall$ could have chosen at position (p, a) in the game $\overline{\mathcal{G}(\mathbb{A}, T)}$. At position $(p, \bigcirc_d a')$, $\exists$ is bound to choose the direction d or looses otherwise. We assume the choice for d is part of her winning strategy. In the play of $\overline{\mathcal{G}(\mathbb{A}, T)}$, (pd, a') would thus be a next basic position consistent with $\exists$'s winning strategy.

$$(p,a) \xrightarrow{\forall} (p,\Phi) \xrightarrow{\exists} (p,\bigcirc_d a') \xrightarrow{(\exists)} (pd,a')$$

$$(p,a) \xrightarrow{\exists} (p,\Phi') \xrightarrow{\forall} (p,\bigcirc_d a') \xrightarrow{(\forall)} (pd,a')$$

Figure 3.2: A round of $\overline{\mathcal{G}(\mathbb{A},T)}$ and $\mathcal{G}(\overline{\mathbb{A}},T)$

We have established that every play π of $\mathcal{G}(\overline{\mathbb{A}},T)$ from the initial position consistent with the derived strategy is over the same basic positions as a play π' of $\overline{\mathcal{G}(\mathbb{A},T)}$ consistent with $\exists$'s winning strategy. Because $\exists$ wins π', it satisfies that the largest priority of an automaton state occurring infinitely often is even. So does the play π which is then won by $\exists$. The derived strategy is thus winning.

For the right-to-left direction of $(*)$, suppose $\exists$ has a winning strategy in the game $\mathcal{G}(\overline{\mathbb{A}},T)$ from the initial position, we derive a winning strategy for her in the game $\overline{\mathcal{G}(\mathbb{A},T)}$ from the initial position, such that every play in $\overline{\mathcal{G}(\mathbb{A},T)}$ consistent with the derived strategy is over the same basic position as a play in the game $\mathcal{G}(\overline{\mathbb{A}},T)$ consistent with her winning strategy.

Initially all plays in $\overline{\mathcal{G}(\mathbb{A},T)}$ and $\mathcal{G}(\overline{\mathbb{A}},T)$ are in the same basic position. Let π' be a finite partial play of $\overline{\mathcal{G}(\mathbb{A},T)}$ from the initial position consistent with the derived strategy and let (p,a) be the last position of π'. By the induction hypothesis there is a partial play π of $\mathcal{G}(\overline{\mathbb{A}},T)$ consistent with $\exists$'s winning strategy over the same basic positions as π'. Let Φ' be the set which $\exists$ chooses at position (p,a) in π consistently with her winning strategy. Suppose at position (p,a) of the play of $\overline{\mathcal{G}(\mathbb{A},T)}$, $\forall$ chooses the set $\Phi \in \delta(a,T(p))$. By construction of $\overline{\mathbb{A}}$, Φ properly intersects with Φ'. Let $\exists$ choose any element $\bigcirc_d a'$ from $\Phi \cap \Phi'$. $\exists$ should choose the direction d, so that the next basic position is (pd,a').

Because $\bigcirc_d a'$ is an element of Φ' as well, $\forall$ could choose $\bigcirc_d a'$ in $\mathcal{G}(\overline{\mathbb{A}},T)$. For the direction he should choose d or looses otherwise. In the play of $\mathcal{G}(\overline{\mathbb{A}},T)$, (pd,a') is thus a possible next basic position consistent with $\exists$'s winning strategy.

We have established that for every play π' of $\overline{\mathcal{G}(\mathbb{A},T)}$ from the initial position consistent with the derived strategy, there is a play π of $\mathcal{G}(\overline{\mathbb{A}},T)$ consistent with $\exists$'s winning strategy which is over the same basic positions. π satisfies that the largest priority of an automaton state occurring infinitely often is even. So does π', so that $\exists$ wins π'. We have proven the derived strategy winning for $\exists$. □

In $\overline{\mathbb{A}}$, $a_\exists$ has transitions $\delta(a_\exists,c) = \{\{(a_\exists,a_\exists)\}\}$ for every letter $c \in \Sigma$ and an odd priority $\overline{\Omega}(a_\exists)$. Whence we call $a_\exists$, $a_\forall$ in the complementation of $\mathbb{A}$. Similarly to $a_\exists$, once entering the state $a_\forall$, the automaton remains in that state. Because the priority $\overline{\Omega}(a_\forall)$ is odd, $\exists$ looses every play from $a_\forall$ and has thus to avoid $a_\forall$.

For alternating automata, complementation preserves (almost) the size and index of an automaton.

Remark 3.8. *The complement* $\overline{\mathbb{A}}$ *of* $\mathbb{A}$ *is at most of size* $|A|+1$ *and of index* $|ranOmega+1|$.

For the use in subsequent chapters we mention two observations on the complementation of non-deterministic and co-non-deterministic automata.

Proposition 3.9. *The complementation of a non-deterministic binary tree automaton in direction normal form is co-non-deterministic.*

Proof. Recall that we can see non-deterministic automata as alternating automata with a transition function whose range consists of singletons, only. Let $\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega\rangle$ be a non-deterministic binary tree automaton in direction normal form. The range of the transition function δ consists of singletons $\{\bigcirc_d a\}$, only. For every automaton state $a \in A$ and letter $c \in \Sigma$, there is a unique choice of an element from each set $\Phi \in \delta(a, c)$, so that $\overline{\delta}(a, c)$ consists of one element which is the empty set if $\delta(a, c)$ is empty. Hence the complementation $\overline{\mathbb{A}}$ is co-non-deterministic. □

Proposition 3.10. *The complementation of a co-non-deterministic binary tree automaton is non-deterministic.*

Proof. Let $\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega\rangle$ be a co-non-deterministic binary tree automaton. Replacing each pair (a_0, a_1) in every set in the range of δ by the pairs $\bigcirc_0 a_0$ and $\bigcirc_1 a_1$, yields an equivalent co-non-deterministic binary tree automaton, so that we may without loss of generality assume that $\mathbb{A}$ is in direction normal-form. Then for every state $a \in A$ and letter $c \in \Sigma$, $\delta(a, c)$ is a singleton $\{\Phi\}$ or empty. Every choice function picks precisely one element from Φ. In the complementary automaton $\overline{\mathbb{A}}$, $\overline{\delta}(a, c)$ thus consists of singletons, only. If Φ is empty, then there is no such choice function, so that $\overline{\delta}(a, c)$ is empty. The automaton $\overline{\mathbb{A}}$ is thus non-deterministic. □

Chapter 4

Non-Determinising Alternating Binary Tree Automata

Universal choice in the acceptance game of an alternating binary tree automaton complicates the definition of operations, such as existential projection which we will introduce in the next chapter, or decision procedures, such as for the non-emptiness problem of binary tree automata which we will discuss in Chapter 7. In this chapter we will see that alternating binary tree automata and non-deterministic ones are equally expressive.

Theorem 4.1. *For every alternating binary tree automaton there is an equivalent non-deterministic automaton.*

We show how to compute a non-deterministic automaton $\mathbb{B}$ from an alternating binary tree automaton

$$\mathbb{A} = \langle A, \Sigma, \delta : A \times \Sigma \to \mathcal{PP}(A \times A), a_I, \Omega \rangle$$

such that $\mathbb{A}$ and $\mathbb{B}$ are equivalent. We refer to the procedure as **non-determinisation**.

4.1 Towards Relational Macrostates

At a basic position (p, a) in the acceptance game $\mathcal{G}(\mathbb{A}, T)$ for a binary tree T, $\exists$ is to choose a set Φ of automaton states, from which $\forall$ is to choose a particular pair of states. In the acceptance game for a non-deterministic binary tree automaton, $\exists$ is to choose a pair of states.

The main idea of non-determinisation is to group all of $\forall$'s possible choices for pairs of automaton states from Φ into a pair of sets of automaton states, the $\mathfrak{B}$-redistribution of Φ.

Definition 4.2 ($\mathfrak{B}$-redistribution)**.** *The* $\mathfrak{B}$-***redistribution*** *of a set* $\Phi \in \mathcal{P}(A \times A)$ *of pairs of automaton states is the pair* $(\pi_0[\Phi], \pi_1[\Phi]) \in \mathcal{P}(A) \times \mathcal{P}(A)$ *of sets of automaton states.*[1]

Let $(\pi_0[\Phi], \pi_1[\Phi])$ be the $\mathfrak{B}$-redistribution of Φ. The direction d chosen by $\forall$ determines a whole set of successor states, videlicet $X := \pi_d[\Phi]$. Each state in X is a_d in a pair (a_0, a_1) of states in Φ, and is thus an admissible next state of $\mathbb{A}$ in the acceptance game $\mathcal{G}(\mathbb{A}, T)$.

In the next round of the game, $\exists$ has to choose for every state $a' \in X$ from $\delta(a', T(pd))$ a set $\Phi_{a'}$ with the $\mathfrak{B}$-redistribution $(\pi_0[\Phi_{a'}], \pi_1[\Phi_{a'}])$. The choice of the $\mathfrak{B}$-redistribution for every state a' is coded into binary A-relations R_0 and R_1 such that $R_d[a'] := \pi_d \Phi_{a'}$ for both directions $d \in 2$. The set of all binary A-relations is denoted by $Rel(A)$.

If for some state $a' \in X$, $\delta(a', T(pd))$ is empty, then there is no $\Phi_{a'} \in \delta(a', T(pd))$, so that R_0 and R_1 are not defined. If $\delta(a', T(pd))$ is empty, then $\exists$ can not move at the basic position (pd, a'). If instead $\delta(a', T(pd)) := \{\{(a_\forall, a_\forall)\}\}$, then $\exists$ looses the continuing infinite play from position (pd, a). So define $\delta(a, c) := \{\{(a_\forall, a_\forall)\}\}$ whenever $\delta(a, c)$ is empty for a state $a \in A$ and a letter $c \in \Sigma$. In the following we assume $\delta(a, c)$ to be non-empty for all states $a \in A$ and letters $c \in \Sigma$.

A strategy of $\exists$ in the game $\mathcal{G}(\mathbb{A}, T)$ from the initial position determines for each infinite path $p \in 2^\omega$, a sequence $\rho \in (Rel(A))^\omega$ of binary A-relations. The sequence of automaton states occurring in a play consistent with $\exists$'s strategy constitutes a trace through ρ.

Definition 4.3 (Traces through Sequences of Relations)**.** *A* ***trace*** *through a sequence* $\rho \in (Rel(A))^\star$ *of binary relations over A is a sequence* $\tau \in A^\star$ *of automaton states from A such that for each* $i < |\tau| - 1$, $(\tau(i), \tau(i+1)) \in \rho(i)$*. We call a trace* τ *through* ρ ***full*** *if* $|\tau| = |\rho| + 1$.

Given a priority function $\Omega : A \to \omega$*, a trace* τ *through an infinite* $Rel(A)$*-sequence* ρ *is* ***bad*** *with respect to* Ω *if* τ *is infinite and the largest priority* $\Omega(a)$ *of an automaton state a occurring infinitely often in* τ *is odd.*

Bad traces satisfy that the largest priority of a state occurring infinitely often is odd. By intuition bad traces thus correspond to infinite plays won by $\forall$. Finite traces correspond to finite plays. By the assumption, that $\delta(a, c)$ is non-empty for all $a \in A$ and $c \in \Sigma$, $\exists$ is able to move at every basic position, so that $\forall$ looses all finite plays. Intuitively, that is why finite traces are not bad.

The language NBT_Ω of infinite $Rel(A)$-words with **n**o **b**ad **t**races with respect to $\mathbf{\Omega}$ beginning at a_I is in fact regular, that is NBT_Ω is recognisable by a deterministic word automaton which we are going to define next.

[1] We write $\pi_d[\Phi]$ for the elementwise projection $\{a_d \mid (a_0, a_1) \in \Phi\}$ of Φ to d.

4.2 Bad Traces through Sequences of Relations

In a first step we define a non-deterministic word automaton $\mathbb{M}$ which accepts precisely the infinite $Rel(A)$-words which contain a bad trace with respect to Ω. Intuitively, $\mathbb{M}$ non-deterministically "guesses" a bad trace through a $Rel(A)$-word.

Definition 4.4. *Formally, define*

$$\mathbb{M} := \langle A, Rel(A), \lambda : A \times Rel(A) \to \mathcal{P}(A), a_I, \overline{\Omega} \rangle$$

with transitions

$$a' \in \lambda(a, R) \text{ iff } (a, a') \in R \text{ for any } R \subseteq A \times A$$

and priorities

$$\overline{\Omega}(a) := \Omega(a) + 1 \text{ for all } a \in A.$$

Property 4.5. $\mathbb{M}$ *accepts an infinite* $Rel(A)$*-word* $\rho \in (Rel(A)^\omega$ *iff it contains a bad trace with respect to* Ω *which begins with* a_I*.*

Proof. In the following fix an infinite $Rel(A)$-word ρ.

For the left-to-right implication, we need to prove that any accepting run π of $\mathbb{M}$ on ρ is a bad trace through ρ beginning with a_I. Every run of $\mathbb{M}$ on ρ begins with a_I. Moreover, because $\mathbb{M}$ is non-deterministic, π must be infinite. In order to prove π a trace through ρ, it thus remains to show that for all $i \in \omega$, $(\pi(i), \pi(i+1)) \in \rho(i)$ which follows immediately from $\pi(i+1) \in \delta(\pi(i), \rho(i))$ by construction of δ. We have established that π is a trace through ρ. Because π is an accepting run, the largest priority $\overline{\Omega}(a)$ of an automaton state a occurring infinitely often in π is even. Among the states occurring infinitely often, a has also the largest priority $\Omega(a)$ which is odd, so that π is indeed a bad trace.

For the right-to-left implication, we show that every bad trace τ through ρ beginning with a_I is an accepting run of $\mathbb{M}$ on ρ. Because τ is a bad trace, τ is infinite. In order to prove π a run of $\mathbb{M}$ on ρ, it remains to show that for all $i \in \omega$, $\tau(i+1) \in \delta(\tau(i), \rho(i))$ which is immediate because $(\tau(i), \tau(i+1)) \in \rho(i)$ by definition of traces. Because τ is bad, it satisfies that the largest priority $\overline{\Omega}(a)$ of an automaton state a occurring infinitely often is odd. The state a has also the largest priority $\Omega(a)$ among the states occurring infinitely often in τ. $\Omega(a)$ is even, so that τ is an accepting run. □

In [Saf88] Safra presented a procedure to compute from an equivalent deterministic word automaton $\mathbb{N}$ from a non-deterministic word automaton $\mathbb{M}$, that is to determinise $\mathbb{M}$. Let

$$\mathbb{N} = \langle N, Rel(A), \lambda' : N \times Rel(A) \to N, n_I, \Omega' \rangle$$

be a deterministic word automaton accepting precisely the infinite $Rel(A)$-words accepted by $\mathbb{M}$. The following is immediate.

Property 4.6. $\mathbb{N}$ *accepts an infinite Rel*(A)*-word* U *iff* U *contains a bad trace with respect to* Ω *beginning with* a_I.

We are looking for a deterministic word automaton accepting precisely the infinite *Rel*(A)-words which do not contain a bad trace with respect to Ω which begins with a_I, that is the complement of the language of $\mathbb{N}$. Because $\mathbb{N}$ is deterministic, complementing $\mathbb{N}$ amounts to complementing the acceptance condition. Formally, define

$$\overline{\mathbb{N}} := \langle N, \mathit{Rel}(A), \lambda' : N \times \Sigma \to \mathcal{P}(N), n_I, \overline{\Omega'} \rangle.$$

such that for every state $n \in N$, $\overline{\Omega'}(n) = \Omega'(n) + 1$.

Property 4.7. $\overline{\mathbb{N}}$ *accepts an infinite Rel*(A)*-word* U *iff* U *does not contain a bad trace with respect to* Ω *which begins with* a_I.

Proof. Because $\mathbb{N}$ and $\overline{\mathbb{N}}$ have the same state set, the same initial state, and the same transition function, every run of $\mathbb{N}$ is a run of $\overline{\mathbb{N}}$ and vice versa. Because $\overline{\mathbb{N}}$ is deterministic, for every *Rel*(A)-word ρ there is a unique run π of $\overline{\mathbb{N}}$ on ρ which is accepting for $\overline{\mathbb{N}}$ iff π is a rejecting run of $\mathbb{N}$ on ρ. Hence $\overline{\mathbb{N}}$ accepts the complement of $\mathcal{L}(\mathbb{N})$ and thus also the complement of $\mathcal{L}(\mathbb{M})$. Because $\mathbb{M}$ accepts precisely the infinite *Rel*(A)-words containing a bad trace which begins with a_I, $\overline{\mathbb{N}}$ accepts precisely the infinite *Rel*(A)-words not containing a bad trace which begins with a_I. □

4.3 Finding Sequences of Relations without Bad Traces

Because $\mathbb{A}$ is alternating, $\exists$ has in general multiple choices for Φ in every basic position of the acceptance game $\mathcal{G}(\mathbb{A}, T)$. Each possible choice for Φ corresponds to a possible choice for the pair (R_0, R_1) of relations. Using $\mathbb{A}$ and $\overline{\mathbb{N}}$ we define a non-deterministic binary tree automaton $\mathbb{B}$ which "guesses" the relations R_0 and R_1.

Definition 4.8. *Define the non-deterministic binary tree automaton*

$$\mathbb{B} := \langle N, \Sigma, \delta' : N \times \Sigma \to \mathcal{P}(N \times N), n_I, \overline{\Omega'} \rangle$$

with transitions

$$(n_0, n_1) \in \delta'(n, c)$$

iff

- *there are binary relations* R_0 *and* R_1 *such that for each* $a \in A$, $(R_0[a], R_1[a])$ *is the* $\mathfrak{B}$*-redistribution for some* $\Phi \in \delta(a, c)$
- *and* $\lambda'(n, R_0) = n_0$ *and* $\lambda'(n, R_1) = n_1$

The transition function λ uniquely maps relations R_0 and R_1 to successor states n_0 and n_1, respectively. For every choice of a pair (n_0, n_1) of states from $\delta'(n, c)$ there is thus a supporting choice of relations (R_0, R_1) such that $n_d = \lambda(n, R_d)$. Below we will thus speak of a choice of relations, when referring to a choice of successor states.

Proposition 4.9. $\mathbb{A}$ *and* $\mathbb{B}$ *are equivalent.*

Proof. We show that $(*)$ $\mathbb{A}$ accepts a binary $\mathcal{P}(V)$-labelled tree T iff $\mathbb{B}$ accepts T.

For the left-to-right implication, suppose $\mathbb{A}$ accepts a binary $\mathcal{P}(V)$-labelled tree T, so that $\exists$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}, T)$ from the initial position (ϵ, a_I). We derive a winning strategy for her in the game $\mathcal{G}(\mathbb{B}, T)$ from the initial position (ϵ, n_I). Along a play of $\mathcal{G}(\mathbb{B}, T)$, $\exists$ chooses binary A-relations which determine the automaton states as given in the definition 4.8 above. Let ρ be the sequence of A-relations chosen by $\exists$ in a play of $\mathcal{G}(\mathbb{B}, T)$ from the initial position consistently with the derived strategy. We show that every trace through ρ beginning with a_I is the sequence of automaton states in the basic positions of a play in $\mathcal{G}(\mathbb{A}, T)$ from the initial position played consistently with $\exists$'s winning strategy.

Let π' be a finite partial play of $\mathcal{G}(\mathbb{B}, T)$ from the initial position and the last basic position of π' be (p, n). Let furthermore ρ be the sequence of relations which $\exists$ chose along the play π'. By the induction hypothesis, any full trace through ρ with the last state a corresponds to a play of $\mathcal{G}(\mathbb{A}, T)$ from the initial position with the last basic position (p, a). Let Φ_a be the set which $\exists$ chooses at position (p, a) in $\mathcal{G}(\mathbb{A}, T)$ consistently with her winning strategy. Because $\mathcal{G}(\mathbb{A}, T)$ is historyfree determined, Φ_a is well-defined. At position (p, n) let $\exists$ choose relations R_0 and R_1 with the following properties. For every state a terminal in a full trace through ρ, $R_d[a]$ is the $\mathfrak{B}$-redistribution of Φ_a which $\exists$ would choose at position (p, a) in $\mathcal{G}(\mathbb{A}, T)$ consistently with her winning strategy. For all other states b, $R_d[b]$ be the $\mathfrak{B}$-redistribution of some set $\Phi \in \delta(b, T(p))$. The relations R_0 and R_1 then uniquely determine the next automaton states n_0 and n_1, respectively.

Let d be the direction which $\forall$ chooses, then the next basic position in $\mathcal{G}(\mathbb{B}, T)$ is (pd, n_d). By definition of $\mathfrak{B}$-redistributions, every automaton state a' in $R_d[a]$ is the state a_d in a pair (a_0, a_1) from Φ_a, so that (pd, a') is an admissible next position in $\mathcal{G}(\mathbb{A}, T)$ consistent with $\exists$'s winning strategy.

The largest priority of an automaton state occurring infinitely often in a play in $\mathcal{G}(\mathbb{A}, T)$ from the initial position consistent with $\exists$'s winning strategy is even, so that every trace through ρ beginning with a_I is not bad and thus ρ is accepted by $\overline{\mathbb{N}}$ in a unique run π such that $\pi(i+1) = \lambda(\pi(i), \rho(i))$, for all $i \in \omega$. Moreover the automaton states in π and the play π' of $\mathcal{G}(\mathbb{B}, T)$ coincide by definition 4.8, so that $\exists$ wins the play π'.

For the right-to-left direction of $(*)$ suppose that $\mathbb{B}$ accepts a binary $\mathcal{P}(V)$-labelled tree T, so that $\exists$ has a winning strategy in the game $\mathcal{G}(\mathbb{B}, T)$ from the initial postion (ϵ, n_I). We derive a strategy for her in $\mathcal{G}(\mathbb{A}, T)$, such that every play consistent with the derived strategy corresponds to a trace through the relations chosen by $\exists$ in $\mathcal{G}(\mathbb{B}, T)$ consistently with her winning strategy.

Let π be a finite partial play of $\mathcal{G}(\mathbb{A}, T)$ from the initial position consistent with the derived strategy and let (p, a) be the last basic position of π. By the induction hypothesis, a is the last state in a trace τ through the sequence ρ of A-relations chosen by $\exists$ in a finite partial play π' of $\mathcal{G}(\mathbb{B}, T)$ from the initial position consistent with her winning strategy. Let the basic position (p, n) be the last position in π'. At position (p, n), $\exists$ chooses relations R_0 and R_1 which determine the next automaton states n_0 and n_1. By definition of δ, for each $a \in A$, $(R_0[a], R_1[a])$ is the $\mathfrak{B}$-redistribution of some Φ in $\delta(a, T(p))$. Thus at position (p, a), $\exists$ can and should choose such Φ. $\forall$ chooses a pair (a_0, a_1) from Φ and a direction d. The next basic position in $\mathcal{G}(\mathbb{A}, T)$ is then (pd, a_d).

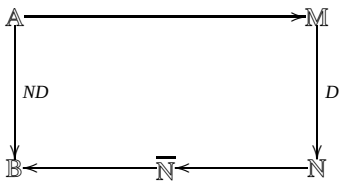

Figure 4.1: Reduction of the Non-Determinisation (*ND*) of Alternating Binary Tree Automata to the Determinisation (*D*) of Word Automata

Because $a_d \in \pi_d\Phi$, it is $(a, a_d) \in R_d$. So that τa_d, that is τ with a_d appended, is a trace through ρR_d, that is ρ with R_d appended.

Hence every play π in $\mathcal{G}(\mathbb{A}, T)$ consistent with $\exists$'s winning strategy is over the automaton states of a trace through the relations chosen in $\mathcal{G}(\mathbb{B}, T)$ consistent with $\exists$'s winning strategy. Every such trace satisfies that the largest priority of an automaton state occurring infinitely often is even, so that $\exists$ wins the corresponding play π. □

If $\mathbb{A}$ is co-non-deterministic, the non-determinisation of $\mathbb{A}$ is deterministic.

Proposition 4.10 (Non-Determinisation of Co-Non-Deterministic Automata)**.** *The non-determinisation $\mathbb{B}$ of a co-non-deterministic automaton $\mathbb{A}$ is deterministic.*

Proof. Because $\mathbb{A}$ is co-non-deterministic, $\delta(a, c)$ is a singleton for all $a \in A$ and $c \in \Sigma$. By the definition of the non-determinisation, there is precisely one pair (R_0, R_1) in $\delta'(R, c)$, so that $\mathbb{B}$ is deterministic. □

4.4 Complexity

Summarising the construction of non-determinisation in this chapter, we have reduced the non-determinisation of alternating binary tree automata to the determinisation of non-deterministic word automata as schematised in Figure 4.1. In the following we will recollect the details of the reduction, and moreover prove the converse, namely that the determinisation of non-deterministic word automata reduces to the non-determinisation of alternating binary tree automata. We obtain that the non-determinisation of alternating binary tree automata is as complex as the determinisation of non-deterministic word automata.

In a first step we obtained from $\mathbb{A}$, a non-deterministic word automaton $\mathbb{M}$ which accepts an infinite $Rel(A)$-word ρ iff ρ contains a bad trace with respect to Ω beginning with a_I. $\mathbb{M}$ has the same size and index as $\mathbb{A}$. Determinising $\mathbb{M}$ yielded an equivalent deterministic word automaton $\mathbb{N}$. Complementing $\mathbb{N}$ we obtained the deterministic automaton $\overline{\mathbb{N}}$ of the same size and index as $\mathbb{N}$. $\overline{\mathbb{N}}$ accepts precisely the infinite $Rel(A)$-words without bad traces with respect to Ω beginning with a_I. Finally we constructed the non-deterministic binary tree automaton $\mathbb{B}$ equivalent to $\mathbb{A}$. $\mathbb{B}$ is of the same size and index as $\overline{\mathbb{N}}$. Hence $\mathbb{A}$ has the same size and index as $\mathbb{M}$, and $\mathbb{B}$ has the same size and

index as $\mathbb{N}$. We have established that the non-determinisation of binary tree automata is at most as hard as the determinisation of word automata.

In the remainder of this chapter we give a reduction of the determinisation of non-deterministic word automata to the non-determinisation of binary tree automata as shown in Figure 4.2. Essentially, determinising a word automaton eliminates $\exists$'s choice in the static stage of the acceptance game, whereas non-determinising a binary tree automaton eliminates $\forall$'s choice in the static stage. We will use the idea from Chapter 3 that complementing an alternating binary tree automaton effectively swaps the roles of $\exists$ and $\forall$.

From a given non-deterministic word automaton $\mathbb{M}$ we derive a non-deterministic binary tree automaton $\mathbb{A}$ which accepts precisely the binary trees T_U whose leftmost path is labelled in order with letters from a Σ-word U which is accepted by $\mathbb{M}$. Essential to the construction is that, for each automaton state of $\mathbb{M}$ and letter from Σ, we fix a direction, so that we obtain the non-deterministic binary tree automaton in direction normal form. Because $\mathbb{A}$ is non-deterministic and in direction normal form, complementation yields a co-non-deterministic automaton $\overline{\mathbb{A}}$ by Proposition 3.10. Non-determinising the co-non-deterministic binary tree automaton $\overline{\mathbb{A}}$ yields a deterministic automaton $\mathbb{B}$ by Proposition 4.10. From $\mathbb{B}$ we derive a deterministic word automaton $\mathbb{N}$ which rejects precisely the infinite Σ-words U for which $\mathbb{B}$ rejects the left U-tree T_U. Finally we show that the automaton $\overline{\mathbb{N}}$ complementary to the automaton $\mathbb{N}$ is equivalent to $\mathbb{M}$.

For the rest of this section, let

$$\mathbb{M} = \langle M, \Sigma, \lambda : M \times \Sigma \to \mathcal{P}(M), m_I, \Omega \rangle$$

be a non-deterministic word automaton. To $\mathbb{M}$ we add the special states $m_\exists$ and $m_\forall$ with

- transitions $\lambda(m_\exists, c) = \{m_\exists\}$ and $\lambda(m_\forall, c) = \{m_\forall\}$ for all letters $c \in \Sigma$
- and priorities $\Omega(m_\exists) = 0$ and $\Omega(m_\forall) = 1$.

Once entering the state $m_\exists$, $\mathbb{M}$ remains in the state $m_\exists$ which is then the only state occurring infinitely often in a play π of an acceptance game $\mathcal{G}(\mathbb{M}, U)$. Because the priority of $m_\exists$ is even, $\exists$ wins every such play π. Hence $\forall$ should avoid $m_\exists$. Similarly, once $\mathbb{M}$ has entered the state $m_\forall$, $\mathbb{M}$ remains in $m_\forall$ which is then the only automaton state occurring infinitely often. Because $m_\forall$ has an odd priority, $\forall$ wins every play with the state $m_\forall$, so that $\exists$ should avoid $m_\forall$.

$\mathbb{M}$ recognises infinite words over Σ. For an infinite Σ-word U we define left U-trees as follows.

Definition 4.11 (Left U-Tree)**.** *Given an infinite Σ-word $U \in \Sigma^\omega$, a **left U-tree** T_U is a binary Σ-labelled tree, whose leftmost path is labelled with U, that is for all $p \in 0^*$, $T_U(p) = U(|p|)$.*

From $\mathbb{M}$ we construct an alternating binary tree automaton $\mathbb{A}$ accepting every left U-tree for every word U accepted by $\mathbb{M}$. Define

$$\mathbb{A} := \langle M, \Sigma, \delta : M \times \Sigma \to \mathcal{P}\mathcal{P}(M \times M), m_I, \Omega \rangle$$

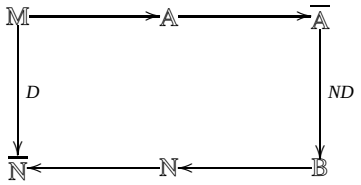

Figure 4.2: Reduction of the Determinisation (*D*) of Non-Deterministic Word Automata to the Non-Determinisation (*ND*) of Alternating Binary Tree Automata

with transitions $\delta(m, c) := \{(m, m_\exists)\} \mid m' \in \lambda(m, c)\}$, for all $m \in M$ and $c \in \Sigma$.

Note that $\forall$'s choice for direction is predetermined in each round, because he should avoid the automaton state $m_\exists$. For convenience we keep the notation $\bigcirc_0 m$ for the pair $(m, m_\exists)$ as introduced in Chapter 3.

Claim 4.12. *$\mathbb{M}$ accepts an infinite Σ-word U iff $\mathbb{A}$ accepts any left U-tree T_U.*

Proof. Suppose $\mathbb{M}$ accepts an infinite Σ-word U, then there is an accepting run $\pi \in M^\omega$ of $\mathbb{M}$ on U. The run π yields a winning strategy for $\exists$ in the game $\mathcal{G}(\mathbb{A}, T_U)$ from the initial position (ϵ, m_I), such that there is only one play which is over the same automaton states as π. Initially the play in $\mathcal{G}(\mathbb{A}, T)$ is at position (ϵ, m_I) and m_I is the initial state of π. Let the play consistent with the derived strategy be at the basic position (p, m) such that $\pi(|p|) = m$. The successor $\pi(|p| + 1)$ of $\pi(|p|)$, is an element of $\lambda(m, U(p))$. Then there is a pair $\bigcirc_0 m$ in $\delta(m, T_U(p))$, which $\exists$ should choose at the basic position (p, m) of the game $\mathcal{G}(\mathbb{A}, T_U)$. $\forall$ is then bound to choose the direction $d = 0$, so that the next basic position is $(p0, \pi(|p0|)$. Because π is accepting, the largest priority of an automaton state occurring infinitely often in π is even. Hence $\exists$ wins the play consistent with the derived strategy.

For the right-to-left direction, suppose $\exists$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}, T_U)$ from the initial position (ϵ, m_I). Then there is only one play consistent with her winning strategy, because $\mathbb{A}$ is direction normal form, and this play is infinite. We show that the automaton states in the play form an accepting run of $\mathbb{M}$ on U. Initially, every play in $\mathcal{G}(\mathbb{A}, T_U)$ is at position (ϵ, m_I) and m_I is the initial state of any run of $\mathbb{M}$ on U. Let $\bigcirc_0 m'$ be the pair which $\exists$ chooses at a basic position (p, m) for which $\pi(|p| - 1) = m$. $\forall$ is bound to choose the direction 0, so that the next basic position is $(p0, m')$. The successor state m' is then an element of $\lambda(m, U(|p|))$ by definition of δ, and hence an admissible continuation of $\pi|_{|p|}$. Moreover π is accepting because the play consistent with $\exists$'s winning strategy satisfies that the largest priority of an automaton state occurring infinitely often is even. □

Because $\mathbb{A}$ is non-deterministic and in direction normal form, the complement

$$\overline{A} = \langle M, \Sigma, \overline{\delta} : A \times \Sigma \to \mathcal{PP}(M \times M), m_I, \overline{\Omega} \rangle$$

is co-non-deterministic as in Proposition 3.9. For all $m \in M$ and $c \in \Sigma$, $\overline{\delta}(m, c)$ is thus a singleton or empty. In the latter case, if there are m and c such that $\overline{\delta}(m, c)$ is empty,

$\exists$ can not move at the basic position (p, m) of an acceptance game $\mathcal{G}(\overline{\mathbb{A}}, T)$ if $T(p) = c$. Then define $\overline{\delta}(m, c) := \{\{(m_\forall, m_\forall)\}\}$, so that $\exists$ looses the continuing infinite play from (p, m). We may thus assume without loss of generality that $\overline{\delta}(m, c)$ is a singleton for all $m \in M$ and $c \in \Sigma$.

By Proposition 4.10, non-determinising $\overline{\mathbb{A}}$ then yields a deterministic binary tree automaton

$$\mathbb{B} = \langle B, \Sigma, \delta' : B \times \Sigma \to \mathcal{P}(B \times B), b_I, \Omega' \rangle,$$

which rejects precisely the left U-trees. From $\mathbb{B}$ we can easily obtain a deterministic word automaton.

$$\mathbb{N} = \langle B, \Sigma, \lambda' : N \times \Sigma \to N, b_I, \Omega' \rangle$$

such that

$$\lambda'(b, c) = b' \text{ if } \pi_0(\delta'(b, c)) = \{b'\}.$$

Then $\mathbb{N}$ rejects the infinite Σ-words for which $\mathbb{B}$ rejects all left U-trees. Note that $\mathbb{B}$ and $\mathbb{N}$ are of the same size and index.

Claim 4.13. *$\mathbb{N}$ rejects a word U iff $\mathbb{B}$ rejects all left U-trees.*

Proof. For the left-to-right implication suppose that $\mathbb{N}$ rejects an infinite Σ-word U. Because $\mathbb{N}$ is deterministic, there is a unique run π of $\mathbb{N}$ on U which is rejecting. We show inductively that if $\forall$ steadily chooses the left direction in a play of $\mathcal{G}(\mathbb{B}, T)$ from the initial position, then the play consists of the states from the run of $\mathbb{N}$ on U. Initially, the play is at position (ϵ, b_I), the initial state of the run is b_I. Let the play be in position (p, b). By the induction hypothesis, the run is in state b at index $|p|$, that is $\pi(|p|) = b$. By definition of left U-trees, $T_U(p) = U(|p|)$. Because $\lambda'(b, U(|p|)) = \pi(|p| + 1)$, by construction of $\mathbb{N}$, it is $\delta'(b, T_U(|p|) = \{(\pi(|p| + 1), b_1)\}$, so that $\exists$ is bound to choose the pair $((\pi(|p| + 1), b_1)$ of states. Let $\forall$ choose the left direction, then the next basic position is $(p0, \pi(|p| + 1)$. We have established, that the play in $\mathcal{G}(\mathbb{B}, T_U)$ contains the states of the run π. Because π is rejecting, the largest priority of an automaton state occurring infinitely often in the play is odd, so that $\forall$ wins the play of $\mathcal{G}(\mathbb{B}, T_U)$ from the initial position and $\mathbb{B}$ rejects T_U.

For the right-to-left implication, we need an additional property of the automaton $\mathbb{B}$. Therefor we firstly digress on the complementation and non-determinisation as we have introduced in the previous and the current chapter.

By construction, $\mathbb{A}$ is non-deterministic and in direction normal form, so that for each $a \in A$ and $c \in \Sigma$, $\delta(a, c)$ consists of singletons $\{\bigcirc_0 a'\}$. Then $\overline{\mathbb{A}}$ is co-non-deterministic such that for each $a \in A$ and $c \in \Sigma$, $\overline{\delta}(a, c) = \{\{\bigcirc_0 a' \mid \{\bigcirc_0 a'\} \in \delta(a, c)\}\}$. The right side of the $\mathfrak{B}$-redistribution $(\pi_0\{\bigcirc_0 a'\} \in \delta(a, c)\}, \pi_1\{\bigcirc_0 a'\} \in \delta(a, c)\})$ is then the singleton $\{a_\exists\}$. Moreover $\overline{\delta}(a_\exists, c)$ contains for any $c \in \Sigma$, only the singleton $\{(a_\exists, a_\exists)\}$ whose $\mathfrak{B}$-redistribution is $(\{a_\exists\}, \{a_\exists\})$. Thus any trace through the relations chosen by $\exists$ in the acceptance $\mathcal{G}(\mathbb{B}, T)$ for any binary tree T on a path with is not strictly leftmost is infinite and contains $a_\exists$ as the only automaton state occurring infinitely often. Any such trace can thus not be bad. Along any play of $\mathcal{G}(\mathbb{B}, T)$ from the initial position, where $\forall$ does not strictly choose the left direction, $\exists$ chooses relations which do not have bad traces. $\forall$ will thus loose every such play. Hence we can and will assume that $\forall$ prefers to choose the left direction.

Suppose $\mathbb{B}$ rejects a left U-tree T_U, so that $\forall$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{B}, T_U)$ from the initial position (ϵ, b_I). By the previous argument we can assume $\forall$ to choose the left direction throughout the play. Because furthermore $\mathbb{B}$ is deterministic, the strategy is trivial, so that there is precisely a play π from the initial position winning for $\forall$. We show inductively that the basic positions π contain automaton states of $\mathbb{N}$ which in order constitute a run of $\mathbb{N}$ on U. For the base case, initially the run of $\mathbb{N}$ is in state b_I, the initial position of $\mathcal{G}(\mathbb{B}, T_U)$ is (ϵ, b_I). Suppose we are at position (p, b) in the play π, where $\delta'(b, T_U(p)) = (b_0, b_1)$. By the previous argument $\forall$ chooses the left direction 0, so that the next position is $(p0, b_0)$. By the induction hypothesis, the run of $\mathbb{N}$ on U is in state b at index $|p|$. By definition of left U-trees, $U(|p|) = T_U(p)$. By construction of $\mathbb{N}$, $\lambda'(b, U(p)) = b_0$, so that the next state in the run of $\mathbb{N}$ on U is b_0.

Because $\mathbb{B}$ is deterministic, π is infinite. Because furthermore π is winning for $\forall$, the largest priority $\Omega'(b)$ of an automaton state b occurring infinitely often in π is odd. Thus the largest priority of an automaton state occurring infinitely often in the corresponding run of $\mathbb{N}$ on U, is odd as well. The run is thus rejecting. Because $\mathbb{N}$ is deterministic, there is only one run of $\mathbb{N}$ on U, so that $\mathbb{N}$ rejects U. □

Complementing $\mathbb{N}$ yields the deterministic word automaton $\overline{\mathbb{N}}$ which accepts a word U iff $\mathbb{N}$ rejects U. Thus $\mathbb{N}$ accepts precisely the words U for which $\mathbb{B}$ and thus also $\mathbb{A}$ reject all left U-trees which is iff $\overline{\mathbb{A}}$ accept a left U-tree which is iff $\mathbb{M}$ accepts U. Hence $\overline{\mathbb{N}}$ and $\mathbb{M}$ are equivalent.

Theorem 4.14. *The non-determinisation of alternating automata has a complexity constant in order of the complexity of the determinisation of non-deterministic word automata.*

In [Pit06] Piterman shows an upper bound for complexity of the determinisation of non-deterministic word automata with Streett acceptance condition.

Recall that a non-deterministic word automaton

$$\mathbb{M} = \langle M, \Sigma, \lambda : M \times \Sigma \to \mathcal{P}(M), m_I, \Omega \rangle$$

with parity acceptance condition accepts an infinite Σ-word U iff there is an accepting run π of $\mathbb{M}$ on U. Such a run π is a sequence of automaton states, such that π is accepting if the largest priority of an automaton state occurring infinitely often in π is even. Thus the priority function Ω distinguishes the set of accepting runs within the set of all M-sequences.

The set of accepting runs can also be described a **Streett acceptance condition**, which consists of a set $Acc \subseteq \mathcal{P}(M) \times \mathcal{P}(M)$ of pairs (E, F) of sets of automaton states, the **Streett acceptance set**. A run is accepting with respect to the Streett acceptance condition if for every pair (E, F) in Acc and for every automaton state $m \in E$ occurring infinitely often, there is a state $m' \in F$ occurring infinitely often.

From a parity acceptance condition we can derive an equivalent Streett acceptance condition.

Proposition 4.15. *For every priority function Ω there is a Streett acceptance set Acc such that a run $\pi \in M^\star$ is accepting with respect to the parity acceptance condition iff*

π is accepting with respect to the Streett condition. Moreover Acc has the same size as the domain dom(Ω) of Ω.

Proof. Let $\mathbb{M} = \langle M, \Sigma, \lambda, m_I, \Omega \rangle$ be a non-deterministic parity word automaton with priority function $\Omega : M \to \omega$, we define a Streett acceptance set *Acc* such that for every automaton state $m \in M$ with odd priority $\Omega(m)$, there is a pair $(\{m\}, \{m' \mid \Omega(m) < \Omega(m')$ *and* $\Omega(m')$ *even*$\})$ in *Acc*.

Let π be a run of $\mathbb{M}$ on some infinite Σ-word accepting with respect to the parity acceptance condition. The largest priority of an automaton state m occurring infinitely often in π is thus even, that is for every state with odd priority which occurs infinitely often, there is a state of larger and even priority which occurs infinitely often. Thus for every pair $(E, F) \in Acc$ with $E = \{m\}$, whenever m with odd priority occurs infinitely often, so does a state $m' \in F$ with even priority. Hence π is accepting with respect to the Streett acceptance condition as well.

Conversely, if π is accepting with respect to the Streett acceptance condition, then for every pair $(E, F) \in Acc$ with $E = \{m\}$, whenever m occurs infinitely often in π, so does a state in F. Because for every state m with odd priority, there is such a pair $(E, F) \in Acc$, whenever a state with odd priority occurs infinitely often, so does a state with larger and even priority. Hence π is accepting with respect to the parity acceptance condition as well. □

Let $\mathbb{M}'$ be a non-deterministic word automaton of size n and with Streett acceptance condition with an acceptance set *Acc* of cardinality k. Nir Piterman showed that there is an equivalent deterministic word automaton with parity acceptance condition which has size $n^{n(k+2)+2}(k+1)^{2n(k+1)}$ and index $2n(k+1)$. Together with the result from Proposition 4.15, we obtain the following corollary.

Corollary 4.16 (An Upper Bound for the Complexity of the Determinisation of Non-Deterministic Parity Word Automata)**.** *For every non-deterministic parity word automaton of size n and index k there is an equivalent parity word automaton of size* $n^{n(k+2)+2}(k+1)^{2n(k+1)}$ *and index* $2n(k+1)$.

Using Theorem 4.14 we obtain from the above corollary an upper bound for the complexity of the non-determinisation of alternating binary tree parity automata.

Corollary 4.17 (An Upper Bound for the Complexity of the Non-Determinisation of Alternating Binary Tree Parity Automata)**.** *For every alternating binary tree automaton of size n and index k there is an equivalent binary tree parity automaton of size* $n^{n(k+2)+2}(k+1)^{2n(k+1)} + 2$ *and index* $2n(k+1) + 2$.

Chapter 5

Formulas of S2S and Alternating Binary Tree Automata

In this chapter we explicate the connection between formulas of **S2S** and alternating binary tree automata, which establishes the reduction of the satisfiability problem for **S2S** to the non-emptiness problem for alternating binary tree automata. The connection is manifested in the following central theorem.

Theorem 5.1. *For every formula ϕ of **S2S** there is an alternating automaton $\mathbb{A}$ accepting precisely the full binary $\mathcal{P}(FV(\phi))$-labelled trees satisfying ϕ.*

Let ϕ be a formula of **S2S**. By structural induction on ϕ we construct for any set V of variables containing the free variables of ϕ, an alternating binary tree automaton $\mathbb{A}^V_\phi$ accepting precisely the binary $\mathcal{P}(V)$-labelled trees satisfying ϕ.

In particular in Section 5.3, we will make use of the complementation of alternating binary tree automata which we introduced in Chapter 3. Therefor we want each automaton $\mathbb{A}^V_\phi$ to have the special states $a_\exists$ and $a_\forall$ as introduced in Chapter 3.

For the base cases, we define in Sections 5.1 and 5.2 automata $\mathbb{A}^V_{X\subseteq Y}$ and $\mathbb{A}^V_{Suc_d(X,Y)}$ accepting the binary trees satisfying $X \subseteq Y$ and $Suc_d(X,Y)$, respectively. Given automata $\mathbb{A}^V_\phi$ and $\mathbb{A}^V_\psi$ accepting the binary trees satisfying formulas ϕ or ψ, respectively, we define automata $\mathbb{A}^V_{\neg\phi}$, $\mathbb{A}^V_{\phi\vee\psi}$, and $\mathbb{A}^V_{\exists X.\phi}$ accepting binary $\mathcal{P}(V)$-labelled trees satisfying formulas $\neg\phi$, $\phi\vee\psi$, or $\exists X.\phi$, respectively, in Sections 5.3, 5.4, and 5.5.

5.1 The Atom $X \subseteq Y$

Let X and Y be variables. For any set V of variables containing X and Y, we define an automaton $\mathbb{A}^V_{X\subseteq Y}$ accepting the binary $\mathcal{P}(V)$-labelled trees T satisfying $X \subseteq Y$. Such a binary tree T satisfies $X \subseteq Y$ iff it is the representation T of a valuation v satisfying

$X \subseteq Y$. By definition of the semantics of **S2S**, v satisfies $X \subseteq Y$ iff $v(X) \subseteq v(Y)$, which is iff all nodes $p \in 2^*$ satisfy $Y \in T(p)$ whenever $X \in T(p)$.

Definition 5.2. *Define the alternating binary tree automaton*

$$\mathbb{A}^V_{X \subseteq Y} := \langle A, \mathcal{P}(V), \delta : A \times \mathcal{P}(V) \to \mathcal{P}\mathcal{P}(A \times A), a, \Omega \rangle$$

with states

$$A := \{a, a_\exists, a_\forall\},$$

and transitions

$$\delta(a, Z) := \begin{cases} \{\{(a,a)\}\} & \text{if } X \notin Z \text{ or } Y \in Z \\ \emptyset & \text{otherwise} \end{cases}$$

$$\delta(a_\exists, Z) := \{\{(a_\exists, a_\exists)\}\}$$

$$\delta(a_\forall, Z) := \{\{(a_\forall, a_\forall)\}\}$$

for all sets $Z \subseteq V$, and priorities

$$\Omega(a) := 0 \text{ and } \Omega(a_\exists) := 0 \text{ and } \Omega(a_\forall) := 1.$$

The automaton $\mathbb{A}^V_{X \subseteq Y}$ has size 3 and index 2.

Intuitively, in a play of the acceptance game $\mathcal{G}(\mathbb{A}^V_{X \subseteq Y}, T)$, $\mathbb{A}^V_{X \subseteq Y}$ remains in state a. For any node p where $X \in T(p)$ and $Y \notin T(p)$, $\delta(a, T(p))$ is empty, so that $\exists$ can not move at position (p, a). If otherwise $X \notin T(p)$ or $Y \in T(p)$, then $\delta(a, T(p))$ contains precisely the singleton $\{(a, a)\}$, so that for both directions the automaton remains in the state a.

Claim 5.3. *For any set V containing X and Y, $\mathbb{A}^V_{X \subseteq Y}$ accepts a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies $X \subseteq Y$.*

Proof. In the following fix a binary $\mathcal{P}(V)$-labelled tree T.

For the right-to-left implication, suppose T satisfies $X \subseteq Y$, that is for all nodes $p \in 2^*$, $Y \in T(p)$ whenever $X \in T(p)$. We show that every play of the acceptance game $\mathcal{G}(\mathbb{A}^V_{X \subseteq Y}, T)$ from the initial position is infinite and over basic positions with the automaton state a, only. Because the priority of a is even, every such play would be won by $\exists$, so that $\mathbb{A}^V_{X \subseteq Y}$ would accept T.

In any basic position (p, a), and thus also in the initial position (ϵ, a), $\delta(a, T(p))$ is the singleton $\{\{(a, a)\}\}$. At position (p, a), $\exists$ can only choose the set $\{(a, a)\}$ from which $\forall$ can only choose the pair (a, a). Given $\forall$'s choice d for the direction, the next basic position is (pd, a) where the argument recurs.

For the left-to-right implication, we prove the contrapositive. Suppose that there is a node $p \in 2^*$, where $X \in T(p)$, but $Y \notin T(p)$. We may assume p closest to the root with this property, because the ancestorship relation is well-founded. We define a winning strategy for $\forall$ in the game $\mathcal{G}(\mathbb{A}^V_{X \subseteq Y}, T)$ from the initial position (ϵ, a_I) directing the play into the position (p, a), where $\exists$ can not move and looses. Then $\mathbb{A}^V_{X \subseteq Y}$ would reject T.

Let the play of $\mathcal{G}(\mathbb{A}^V_{X\subseteq Y}, T)$ be at the basic position (p', a) for any ancestor p' of p. Thus p' may also be the root node ϵ. By assumption p' satisfies $Y \in T(p')$ if $X \in T(p')$, so that $\delta(a, T(p'))$ is the singleton $\{\{(a, a)\}\}$ from which $\exists$ can only choose the set $\{(a, a)\}$. From the latter $\forall$ can only choose the pair (a, a). For the direction let him choose $p(|p'|)$ which is the next direction in the path towards p. The next basic position is then $(p|_{|p'|+1}, a)$.

Eventually, the play will arrive at position (p, a). Because $X \in T(p)$ and $Y \notin T(p)$, $\delta(a, T(p))$ is empty, so that $\exists$ can not move in (p, a) and looses. □

5.2 The Atom $Suc_d(X, Y)$

Let X and Y be variables and let $d \in 2$ be a direction. For any set V of variables containing X and Y, we define the automaton $\mathbb{A}^V_{Suc_d(X,Y)}$ accepting the binary $\mathcal{P}(V)$-labelled trees T satisfying $Suc_d(X, Y)$. T satisfies $Suc_d(X, Y)$ iff it is the tree representation of a valuation ν satisfying $Suc_d(X, Y)$ which is iff $pd \in \nu(Y)$ whenever $p \in \nu(X)$. Thus T satisfies $Suc_d(X, Y)$ iff for all nodes $p \in 2^*$, $Y \in T(pd)$ whenever $X \in T(p)$.

Definition 5.4. *Define the alternating binary tree automaton*

$$\mathbb{A}^V_{Suc_d(X,Y)} := \langle A, \mathcal{P}(V), \delta : A \times \mathcal{P}(V) \to \mathcal{P}\mathcal{P}(A \times A), a, \Omega \rangle$$

with states

$$A := \{a, a_s, a_\exists, a_\forall\}$$

with transitions

$$\delta(a, Z) = \begin{cases} \{\{(a_0, a_1) \mid a_d = a_s, a_{1-d} = a\}\} & \text{if } X \in Z \\ \{\{(a, a)\}\} & \text{otherwise} \end{cases}$$

$$\delta(a_s, Z) = \begin{cases} \emptyset & \text{if } Y \notin Z \\ \{\{(a_0, a_1) \mid a_d = a_s, a_{1-d} = a\}\} & \text{if } Y \in Z \text{ and } X \in Z \\ \{\{(a, a)\}\} & \text{otherwise} \end{cases}$$

$$\delta(a_\exists, Z) := \{\{(a_\exists, a_\exists)\}\}$$

$$\delta(a_\forall, Z) := \{\{(a_\forall, a_\forall)\}\}$$

for all sets $Z \subseteq V$, and priorities

$$\Omega(a) := 0,\ \Omega(a_s) := 0,\ \Omega(a_\exists) := 0,\ \text{and } \Omega(a_\forall) := 1.$$

The automaton $\mathbb{A}^V_{Suc_d(X,Y)}$ has size 4 and index 2.

Claim 5.5. *For any set V of variables containing both X and Y, $\mathbb{A}^V_{Suc_d(X,Y)}$ accepts a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies $Suc_d(X, Y)$.*

Proof. In the following fix a binary $\mathcal{P}(V)$-labelled tree T.

For the right-to-left implication, suppose for all nodes $p \in 2^*$, $Y \in T(pd)$ whenever $X \in T(p)$. We show that any play of the acceptance game $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ is infinite and

over basic positions with the automaton states a and a_s, only. Because the priorities of a and a_s are even, $\exists$ would win every such play, so that $\mathbb{A}^V_{Suc_d(X,Y)}$ would accept T.

Let the last basic position of a finite partial play of $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ be (p, a). If $X \notin T(p)$, then $\delta(a, T(p))$ is the singleton $\{\{(a, a)\}\}$. At position (p, a), $\exists$ can only choose the set $\{(a, a)\}$ from which $\forall$ can only choose the pair (a, a). If $\forall$ chooses d' for the direction, the next basic position is (pd', a), where the argument recurs. If $X \in T(p)$, then $\delta(a, T(p))$ is the singleton $\{\{(a_0, a_1)\}\}$ where $a_d = a_s$ and $a_{1-d} = a$. At position (p, a), $\exists$ can only choose the set $\{(a_0, a_1)\}$ from which $\forall$ can only choose (a_0, a_1). If he chooses direction $1 - d$, the next basic position is $(p(1 - d), a)$ with the automaton state a. Otherwise the next basic position is (pd, a_s) with the automaton state a_s.

Let the last basic position of a finite partial play of $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ be (p, a_s). By assumption p satisfies $Y \in T(p)$. If $X \notin T(p)$, $\delta(a_s, T(p))$ is the singleton $\{\{(a, a)\}\}$. At position (p, a_s), $\exists$ can only choose the set $\{(a, a)\}$ from which $\forall$ can only choose (a, a). Given choice d' of $\forall$ for the direction, the next basic position is (pd', a), where the argument above recurs. If otherwise $X \in T(p)$, then it has to be verified for pd whether $Y \in T(pd)$. The set $\delta(a_s, T(p))$ consists such of the singleton $\{\{(a_0, a_1)\}\}$. At position (p, a_s), $\exists$ then only choose the set $\{(a_0, a_1)\}$ from which $\forall$ can only choose (a_0, a_1). If $\forall$ chooses the direction d, the next basic position is (pd, a_s) where the argument recurs, and otherwise $(p(1 - d), a)$ where the argument above recurs.

For the left-to-right implication we prove the contrapositive. Suppose that there is a node p for which $X \in T(p)$, but $Y \notin T(p)$. Because the ancestorship relation is well-founded for binary trees, we may assume p to be closest to the root with this property. Then we derive a strategy for $\forall$ according to which he directs the play towards one of the basic positions (p, a) or (p, a_s) where $\exists$ can not move and looses. Then $\mathbb{A}^V_{Suc_d(X,Y)}$ would reject T.

Let the last basic position of a finite partial play of $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ from the initial position be (p', a) for p' an ancestor of p. Note that p' can also be the root node ϵ. By the assumption above p' satisfies that $Y \in T(p')$ whenever $X \in T(p')$. If $X \notin T(p')$ then $\delta(a, T(p'))$ is the singleton $\{\{(a, a)\}\}$, so that at position (p', a), $\exists$ can only choose the set $\{(a, a)\}$ from which $\forall$ can then only choose the pair (a, a). For the direction let $\forall$ choose $p(|p'|)$ which is the next direction in the path towards p. The next basic position is then $(p|_{|p'|+1}, a)$. If $X \in T(p')$ then $\delta(a, T(p'))$ is the singleton $\{\{(a_0, a_1)\}\}$ where $a_d = a_s$ and $a_{1-d} = a$. At position (p', a), $\exists$ can only choose the set $\{(a_0, a_1)\}$ from which $\forall$ can only choose the pair (a_0, a_1). Let $\forall$ choose $p(|p'|)$ for the direction. The next basic position is $(p|_{|p'|+1}, a_{p(|p'|)})$.

Let the last basic position of a finite partial play of $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ be (p', a_s) for p' an ancestor of p. By assumption p' satisfies $Y \in T(p')$. If $X \notin T(p')$ then $\delta(a, T(p'))$ is the singleton $\{\{(a, a)\}\}$. At position (p', a_s), $\exists$ can only choose the set $\{(a, a)\}$ from which $\forall$ can only choose the pair (a, a). For the direction let $\forall$ choose $p(|p'|)$ so that the next basic position is $(p|_{|p'|+1}, a)$. If $X \in T(p')$, then $\delta(a, T(p'))$ is the singleton $\{\{(a_0, a_1)\}\}$ where $a_d = a_s$ and $a_{1-d} = a$. At position (p', a_s), $\exists$ can only choose the singleton $\{(a_0, a_1)\}$ from which $\forall$ can only chooses the pair (a_0, a_1). For the direction let $\forall$ choose $p(|p'|)$ so that the next basic position is $(p|_{|p'|+1}, a_{p(|p'|)})$.

Eventually the play will arrive at position (p, a) or (p, a_s). Because $X \in T(p)$, $\delta(a, T(p))$ or respectively $\delta(a_s, T(p))$ is the singleton $\{\{(a_0, a_1)\}\}$ where $a_d = a_s$ and

$a_{1-d} = a$. $\exists$ can only choose the singleton $\{(a_0, a_1)\}$ from which $\forall$ can only choose (a_0, a_1). For the direction let $\forall$ choose d, so that the next basic position is (pd, a_s). Because $Y \notin T(pd)$, $\delta(a, T(pd))$ is empty. So that $\exists$ can not move at position (pd, a_s) and looses.

We have proven the derived strategy for $\forall$ winning, so that $\exists$ does not have a winning strategy in $\mathcal{G}(\mathbb{A}^V_{Suc_d(X,Y)}, T)$ from the initial position. □

5.3 Negation and Complementation

Let ϕ be a formula of **S2S**. For any set V of variables containing the free variables of $\neg\phi$, we define an automaton $\mathbb{A}^V_{\neg\phi}$ accepting a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies $\neg\phi$ which is iff T does not satisfy ϕ.

By the induction hypothesis, for any set V of variables containing the free variables of ϕ, there is an automaton $\mathbb{A}^V_\phi$ which accepts a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies ϕ. Because ϕ and $\neg\phi$ have the same free variables, every set containing the free variables of $\neg\phi$ contains the free variables of ϕ. We can thus define $\mathbb{A}^V_{\neg\phi} := \overline{\mathbb{A}^V_\phi}$. By Proposition 3.7, $\overline{\mathbb{A}^V_\phi}$ accepts a binary Σ-labelled tree T iff $\mathbb{A}^V_\phi$ rejects T which is iff T does not satisfy ϕ. The following is immediate.

Corollary 5.6. *For any set V of variables containing the free variables of $\neg\phi$, $\mathbb{A}^V_{\neg\phi}$ accepts a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies $\neg\phi$.*

By Remark 3.8 complementation preserves the size and index of automata having the special state $a_\exists$. Hence $\mathbb{A}^V_{\neg\phi}$ has the same size and index as $\mathbb{A}^V_\phi$. Moreover, in the complemented automaton $a_\forall$ takes over the role of $a_\exists$ and vice versa, so that we can assume $\mathbb{A}_{\neg\phi}$ to have both special states.

5.4 Disjunction and Union

Let ϕ and ψ be formulas of **S2S**. For any set V of variables containing the free variables of $\phi \vee \psi$ we construct an alternating binary tree automaton $\mathbb{A}^V_{\phi\vee\psi}$ accepting a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies $\phi \vee \psi$. By the semantics, T satisfies $\phi \vee \psi$ iff T satisfies ϕ or ψ.

By the induction hypothesis there is for every set V of variables containing the free variables of ϕ an alternating binary tree automaton

$$\mathbb{A}^V_\phi = \langle A_\phi, \mathcal{P}(V), \delta_\phi : A_\phi \times \mathcal{P}(V) \to \mathcal{PP}(A_\phi \times A_\phi), a^\phi_I, \Omega_\phi \rangle$$

accepting a binary $\mathcal{P}(V)$-labelled tree T iff T satisfies ϕ. Similarly, there is for ψ an alternating automaton

$$\mathbb{A}^V_\psi = \langle A_\psi, \mathcal{P}(V), \delta_\psi : A_\psi \times \mathcal{P}(V) \to \mathcal{PP}(A_\psi \times A_\psi), a^\psi_I, \Omega_\psi \rangle.$$

Because any set containing the free variables of $\phi \vee \psi$ contains the free variables of both ϕ and ψ, we can define $\mathbb{A}^V_{\phi\vee\psi}$ as follows.

Definition 5.7. *Define the alternating binary tree automaton*

$$\mathbb{A}^V_{\phi\vee\psi} := \langle A, \mathcal{P}(V), \delta : A \times \mathcal{P}(V) \to \mathcal{P}\mathcal{P}(A \times A), a_I, \Omega\rangle$$

with states

$$A := \{a_I, a_\exists, a_\forall\} \uplus (A_\phi \setminus \{, a_\exists, a_\forall\}) \uplus (A_\psi \setminus \{a_\exists, a_\forall\})$$

and transitions

- $\delta(a_I, Z) := \delta_\phi(a^\phi_I, Z) \cup \delta_\psi(a^\psi_I, Z)$
- *for any state* $a \in A_\phi \cap A$, $\delta(a, Z) := \delta_\phi(a, Z)$
- *for any state* $a \in A_\psi \cap A$, $\delta(a, Z) := \delta_\psi(a, Z)$

for all subsets $Z \subseteq V$, *and priorities*

- $\Omega(a_I) := 0$
- *for any state* $a \in A_\phi \cap A$, $\Omega(a) := \Omega_\phi(a)$
- *for any state* $a \in A_\psi \cap A$, $\Omega(a) := \Omega_\psi(a)$

Because $\mathbb{A}^V_\phi$ and $\mathbb{A}^V_\psi$ share the states $a_\exists$ and $a_\forall$, $\mathbb{A}^V_{\phi\vee\psi}$ contains only one of each. The size of $\mathbb{A}^V_{\phi\vee\psi}$ is thus at most $|\mathbb{A}^V_\phi| + |\mathbb{A}^V_\phi| - 3$. The index of $\mathbb{A}^V_{\phi\vee\psi}$ is at most $|\Omega_\phi| + |\Omega_\psi| - 2$, because a_I can be assigned an arbitrary priority.

Then the automaton $\mathbb{A}^V_{\phi\vee\psi}$ has the desired property.

Claim 5.8. *For any set* V *of variables containing the free variables of* $\phi \vee \psi$, $\mathbb{A}^V_{\phi\vee\psi}$ *accepts the binary* Σ*-labelled tree* T *iff* T *satisfies* $\phi \vee \psi$.

Proof. For the right-to-left implication, suppose $\mathbb{A}^V_\phi$ accepts a binary $\mathcal{P}(V)$-labelled tree T, so that $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{A}^V_\phi, T)$ from the initial position (ϵ, a^ϕ_I). Let Φ be the set $\exists$ chooses from $\delta_\phi(a^\phi_I, T(\epsilon))$ at the initial position consistently with her winning strategy. By construction of δ, Φ is an element of $\delta(a_I, T(p))$ as well. At the initial position (ϵ, a_I) of the game $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$ let $\exists$ choose Φ. Every pair (a_0, a_1) which $\forall$ can choose from Φ in $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$, he could choose in $\mathcal{G}(\mathbb{A}^V_\phi, T)$. Thus any admissible next basic position (d, a_d) in $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$ consistent with the derived strategy, is an admissible next basic position in $\mathcal{G}(\mathbb{A}^V_\phi, T)$ consistent with $\exists$'s winning strategy. Because $\delta_{\phi\vee\psi}$ and δ_ϕ coincide on the same automaton states from A_ϕ, $\exists$ can apply her winning strategy in $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$ from the basic position (d, a_d). Every play in $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$ from (d, a_d) consistent with $\exists$'s strategy is a play in $\mathcal{G}(\mathbb{A}^V_\phi, T)$ from (d, a_d) consistent with $\exists$ winning strategy. Thus $\exists$ wins any such play in $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$. Hence the automaton $\mathbb{A}^V_{\phi\vee\psi}$ accepts T. The case where $\exists$ has a winning strategy in $\mathcal{G}(\mathbb{A}^V_\psi, T)$ is symmetric.

For the left-to-right implication, suppose that $\mathbb{A}^V_{\phi\vee\psi}$ accepts a binary $\mathcal{P}(V)$-labelled tree T, so that $\exists$ has a winning strategy in the game $\mathcal{G}(\mathbb{A}^V_{\phi\vee\psi}, T)$. Let Φ be the set which $\exists$ chooses at the initial position (ϵ, a_I) consistently with her winning strategy. By construction of δ, Φ is an element of either $\delta_\phi(a^\phi, T(\epsilon))$ or $\delta_\psi(a^\psi, T(\epsilon))$. For the

moment suppose the first. At the initial position (ϵ, a_I^ϕ) of $\mathcal{G}(\mathbb{A}_\phi^V, T)$ let $\exists$ choose the set Φ. Every pair (a_0, a_1) that $\forall$ can choose from Φ in $\mathcal{G}(\mathbb{A}_\phi^V, T)$, he could choose in $\mathcal{G}(\mathbb{A}_{\phi\vee\psi}^V, T)$. So every next basic position (d, a_d) in $\mathcal{G}(\mathbb{A}_\phi^V, T)$ consistent with $\exists$'s strategy is an admissible next basic position in $\mathcal{G}(\mathbb{A}_{\phi\vee\psi}^V, T)$ consistent with $\exists$'s winning strategy. Because δ and δ_ϕ coincide on all automaton states from A_ϕ, every play π in $\mathcal{G}(\mathbb{A}_\phi^V, T)$ from (d, a_d) consistent with the derived strategy is over the same positions as a play in $\mathcal{G}(\mathbb{A}_{\phi\vee\psi}^V, T)$ consistent with $\exists$'s winning strategy. $\exists$ thus wins any such play π, so that the derived strategy is winning. Hence $\mathbb{A}_\phi^V$ accepts T. The case where $\Phi \in \delta_\psi(a_I^\psi, T(\epsilon))$ is analogous. □

5.5 Existential Quantification and Existential Projection

In the following we construct for all sets V of variables containing the free variables of $\exists X.\phi$, an alternating binary tree automaton $\mathbb{A}_{\exists X.\phi}^V$ accepting precisely the binary $\mathcal{P}(V)$-labelled trees satisfying $\exists X.\phi$. Every binary $\mathcal{P}(V)$-labelled tree $T : 2^* \to \mathcal{P}(V)$ represents a valuation $v : V \to 2^*$, such that T satisfies $\exists X.\phi$ iff v satisfies $\exists X.\phi$. By the semantics of **S2S**, such a valuation v satisfies $\exists X.\phi$ iff there is a valuation $v_X \subseteq 2^*$ such that $v[X \mapsto v_X]$ satisfies ϕ, where $v[X \mapsto v_X]$ is the valuation of the variables in $V \cup \{X\}$ mapping $X \mapsto v_X$ and coinciding with v on all other variables in V. Because V contains the free variables in $\exists X.\phi$, $V \cup \{X\}$ contains the free variables of ϕ. Thus the induction hypothesis applies and yields an automaton $\mathbb{A}_\phi^{V\cup\{X\}}$ accepting precisely the binary $\mathcal{P}(V \cup \{X\})$-labelled trees satisfying ϕ.

Chosen arbitrarily, V can contain the variable X. Let v_1 and v_2 be the two valuations of V which agree on the variables in $V \setminus \{X\}$. Because $v_1[X \mapsto v_X] = v_2[X \mapsto v_X]$ for any valuation $v_X \subseteq 2^*$, v_1 and v_2 are equivalent relative to $\exists X.\phi$, that is v_1 satisfies $\exists X.\phi$ iff v_2 does. Thus it makes sense to focus on valuations v of the variables in $V \setminus \{X\}$, that is $v = v_1|_{V\setminus\{X\}}$. We call such a valuation v the **projection** of v_1 (v_2) to $V \setminus \{X\}$. Subsequently we denote the projection of v_1 (v_2) to $V \setminus \{X\}$ by πv_1 (πv_2). Let T_1 be the tree representation of v_1, then the tree representation T of v is defined by $T(p) := T_1(p) \setminus \{X\}$ for all nodes $p \in 2^*$. Analogously, we call T the **projection** of T_1 (T_2) to $V \setminus \{X\}$ which we denote by πT_1 (πT_2).

We define $\mathbb{A}_{\exists X.\phi}^V$ as the **existential projection** $\pi^\exists_{V\setminus\{X\}}\mathbb{A}_\phi^{V\cup\{X\}}$ of $\mathbb{A}_\phi^{V\cup\{X\}}$ which is an alternating binary tree automaton accepting a binary $\mathcal{P}(V)$-labelled tree T' iff there is a $\mathcal{P}(V \cup \{X\})$-labelled tree T accepted by $\mathbb{A}_\phi^{V\cup\{X\}}$ of which T' is the projection to $V \setminus \{X\}$, that is $T' = \pi T$.

We define the existential projection $\pi^\exists_{V\setminus\{X\}}\mathbb{A}_\phi^{V\cup\{X\}}$ in two steps. Firstly, we obtain from $\mathbb{A}_\phi^{V\cup\{X\}}$ an equivalent non-deterministic binary tree automaton $\mathbb{B}$ as described in Chapter 4. Secondly, for $\mathbb{B}$ we define the existential projection as follows.

Definition 5.9 (Existential Projection of Binary Tree Automata)**.** *Let*

$$\mathbb{B} = \langle B, V \cup \{X\}, \delta\text{`} : B \times (V \cup \{X\}) \to \mathcal{P}(B \times B)$$

be a non-deterministic binary tree automaton recognising $\mathcal{P}(V \cup \{X\})$-labelled trees. The existential projection of $\mathbb{B}$ to $V \setminus \{X\}$ is the non-deterministic binary tree automaton

$$\pi^{\exists}_{V\setminus\{X\}}\mathbb{B} := \langle B, \mathcal{P}(V), \pi^{\exists}_{V\setminus\{X\}}\delta : B \times \mathcal{P}(V \setminus \{X\}) \to \mathcal{P}(B \times B), b_I, \Omega\rangle$$

with transitions

$$\pi^{\exists}_{V\setminus\{X\}}\delta(b, Z) := \delta(b, Z) \cup \delta(b, Z \cup \{X\})$$

for all $Z \subseteq V$.

Proposition 5.10. *Let $\mathbb{B}$ be a non-deterministic binary tree automaton recognising $\mathcal{P}(V \cup \{X\})$-labelled trees, and let $\pi^{\exists}_{X\setminus\{X\}}\mathbb{B}$ be the projection of $\mathbb{B}$ to $V \setminus \{X\}$. Then $\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}$ accepts a binary $\mathcal{P}(V \setminus \{X\})$-labelled tree T iff T is the projection of some binary $\mathcal{P}(V \cup \{X\})$-labelled tree T' accepted by $\mathbb{B}$.*

Proof. Suppose $\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}$ accepts a binary $\mathcal{P}(V \setminus \{X\})$-labelled tree T, then $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ from the initial position (ϵ, b_I). We construct a binary $\mathcal{P}(V\cup\{X\})$-labelled tree T' accepted by $\mathbb{B}$ and derive a winning strategy for $\exists$ inductively on a play π' of $\mathcal{G}(\mathbb{B}, T')$ from the initial position consistent with the derived strategy, such that π is a play of $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ consistent with $\exists$'s winning strategy.

Initially all plays in $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ and $\mathcal{G}(\mathbb{B}, T')$ are in the same position, videlicet (ϵ, b_I). Let π be a finite partial play of $\mathcal{G}(\mathbb{B}, T')$ from the initial position consistent with $\exists$'s winning strategy and let the basic position (p, b) be the last position in π. By the induction hypothesis, π is a play of $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$. Let (b_0, b_1) be the pair of states which $\exists$ chooses at position (p, b) consistently with her winning strategy. By construction of $\pi^{\exists}_{V\setminus\{X\}}\delta$, the pair (b_0, b_1) is either in $\delta(b, T(p))$ or in $\delta(b, T(p) \cup \{X\})$. In the first case, let $T'(p) := T(p)$. In the latter case, let $T'(p) := T(p) \cup \{X\}$. Then $\exists$ can choose the same pair (b_0, b_1) of states at position (p, b) in the game $\mathcal{G}(\mathbb{B}, T')$. Let d be the direction which $\forall$ chooses, then (pd, b_d) is the next position in $\mathcal{G}(\mathbb{B}, T')$, and an admissible continuation of the play π in $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ consistently with $\exists$'s winning strategy.

Because $\exists$'s strategy in $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ is winning, she is able to move in every basic position of a play from the initial position consistent with her winning strategy. Also $\forall$ is able to move. The set of all plays consistently with $\exists$'s winning strategy thus completely determines T'.

For the converse direction suppose that $\exists$ has a winning strategy from the initial position (ϵ, b_I) in the game $\mathcal{G}(\mathbb{B}, T')$ for a binary $\mathcal{P}(V\cup\{X\})$-labelled tree T'. We derive a winning strategy for her in the game $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ from the initial position where T is the projection of T' to $V \setminus \{X\}$, such that every play π from the initial position consistent with the derived strategy is a play of $\mathcal{G}(\mathbb{B}, T')$ consistent with $\exists$'s winning strategy.

Initially all plays in $\mathcal{G}(\mathbb{B}, T')$ and $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ are in the same basic position, that is (ϵ, b_I). Let π be a finite partial play of $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ from the initial position consistent with the derived strategy. And let the basic position (p, b) be the last position of π. By

the induction hypothesis π is a play of $\mathcal{G}(\mathbb{B}, T')$. Let (b_0, b_1) be the pair of states which $\exists$ chooses at position (p, b) consistently with her winning strategy. By construction of $\pi^{\exists}_{V\setminus\{X\}}\delta$, the same pair (b_0, b_1) is an element of $\pi^{\exists}_{V\setminus\{X\}}\delta(b, T'(p) \setminus \{X\})$ which equals $\pi^{\exists}_{V\setminus\{X\}}\delta(b, T(p))$. At position (p, b) in the game $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$, let $\exists$ choose the same pair (b_0, b_1). Given a choice d of $\forall$ for the direction, the next position in $\mathcal{G}(\pi^{\exists}_{V\setminus\{X\}}\mathbb{B}, T)$ is (pd, b_d) which is an admissible continuation of the play π in $\mathcal{G}(\mathbb{B}, T')$ consistently with $\exists$'s winning strategy.

□

By construction, $\mathbb{B}$ and $\pi^{\exists}_{X}\mathbb{B}$ are of the same size and index. However, due to the non-determinisation $\mathbb{B}$ is of size exponential in the size of $\mathbb{A}_{\phi}^{V\cup\{X\}}$. Concretely, let n be the size and let k be the index of $\mathbb{A}_{\phi}^{V\cup\{X\}}$, then $\mathbb{A}^{V}_{\exists X.\phi}$ has size at most $n^{n(k+2)+2}(k+1)^{2n(k+1)}$ and index at most $2n(k+1)$ by Corollary 4.17.

Definition 5.11. *Define*

$$\mathbb{A}^{V}_{\exists X.\phi} := \pi^{\exists}_{V\setminus\{X\}}\mathbb{B}$$

where $\mathbb{B}$ is the non-determinisation of $\mathbb{A}_{\phi}^{V\cup\{X\}}$.

Claim 5.12. *For any set V of variables containing the free variables of $\exists X.\phi$, $\mathbb{A}^{V}_{\exists X.\phi}$ accepts precisely the binary $\mathcal{P}(V)$-labelled trees T satisfying $\exists X.\phi$.*

Proof. By Proposition 5.10, $\pi^{\exists}_{X\setminus\{X\}}\mathbb{B}$ accepts precisely the projections πT of binary trees T accepted by $\mathbb{B}$ and thus $\mathbb{A}_{\phi}^{V\cup\{X\}}$. Thus $\pi^{\exists}_{X\setminus\{X\}}\mathbb{B}$ accepts the tree representation T of a valuation v iff there is a valuation v_X of X such that $\mathbb{B}$ accepts the tree representation of $v[X \mapsto v_X]$. By Proposition 4.9, $\mathbb{A}^{V\cup\{X\}}$ and $\mathbb{B}$ are equivalent, so that the following is immediate. $\mathbb{A}^{V}_{\exists X.\phi}$ accepts the tree presentation T of a valuation v iff there is a valuation v_X such that $\mathbb{A}_{\phi}^{V\cup\{X\}}$ accepts the tree representation of $v[X \mapsto v_X]$, that is iff T satisfies $\exists X.\phi$. □

In the next Chapter we will make use of the following observation. Therefor recall from the preliminaries that non-deterministic automata can be seen as special cases of alternating automata.

Remark 5.13. *The existential projection of an alternating binary tree automaton is non-deterministic.*

Chapter 6

Complexity of the Translation

We devote this chapter to the question how efficient our translation is.

6.1 Complexity of the Translation

Before, we define the notion of size for formulas of **S2S**.

Definition 6.1 (Size of **S2S** Formulas)**.** *The size $|\phi|$ of a formula ϕ of **S2S** is inductively defined as*

$$
\begin{array}{ll}
|X \subseteq Y| & := 1 \\
|Suc_d(X, Y)| & := 1 \textit{ for both } d \in 2 \\
|\phi_1 \vee \phi_2| & := |\phi_1| + |\phi_2| + 1 \\
|\exists X.\phi| & := |\phi| + 1
\end{array}
$$

We recollect the complexity results we obtained for the individual constructions.

$$
\begin{array}{ll}
|A^V_{X\subseteq Y}| = 3 & |\Omega_{X\subseteq Y}| = 2 \\
|A^V_{Suc_d(X,Y)}| = 3 & |\Omega_{Suc_d(X,Y)}| = 2 \\
|A^V_{\neg\phi}| = |\mathbb{A}^V_\phi| & |\Omega_{\neg\phi}| = |\Omega_\phi| \\
|A^V_{\phi\vee\psi}| = |\mathbb{A}^V_\phi| + |\mathbb{A}^V_\psi| + 1 & |\Omega_{\phi\vee\psi}| \leq |\Omega_\phi| + |\Omega_\psi| \\
|A^V_{\exists X.\phi}| = |\mathbb{A}^V_\phi|^{|\mathbb{A}^V_\phi|*(|\Omega_\phi|+2)+2}(|\Omega_\phi| + 1)^{2|\mathbb{A}^V_\phi|(|\Omega_\phi|+1)} & |\Omega_{\exists X.\phi}| = 2|\mathbb{A}^V_\phi|(|\Omega_\phi| + 1)
\end{array}
$$

Except for the existential quantification, the translation is linear. Existential projection, however, necessitates non-determinisation which yields an automaton exponential in the size of the alternating automaton. Whence the translation of **S2S** into alternating automata is of non-elementary complexity. In the rest of this Chapter we justify this property.

6.2 A Non-Elementary Lower-Bound for the Translation

In [Rei02] Reinhardt establishes a non-elementary lower bound for the complexity of the translation of formulas of **S1S** into alternating word automata. We adapt his argument to **S2S**. In the following we construct a formula ϕ_n^P of size linear in n which specifies a property of binary trees which can be recognised by an alternating binary tree automaton of size non-elementary in n, only.

We construct by induction over n a formula ϕ_n^P satisfied by a binary $\mathcal{P}(\{P\})$-labelled tree T iff there are precisely two $\{P\}$-labelled nodes along one path. Moreover the nodes have a distance of $F(n)$, videlicet one node is $(F(n) - 1)$-accessible from the other, where $F(n)$ is a non-elementary function defined by

$$F(0) = 1 \textit{ and } F(n+1) = F(n) * 2^{F(n)}.$$

Convention 6.2. *We call the node in P which is closer to the root x, the other node y.*

In the induction basis, we construct a formula ϕ_1^P which establishes that there are (1) precisely (2) two nodes x and y labelled with $\{P\}$ such that y is an immediate successor of x (3).

$$\begin{array}{lll} \phi_0^P := & \exists X.\exists Y. & Sing(X) \wedge X \subseteq P \wedge Sing(Y) \wedge Y \subseteq P \wedge \quad (1) \\ & & (\forall Z.Z \subseteq P \rightarrow Z \subseteq X \vee Z \subseteq Y) \wedge \quad (2) \\ & & Suc_0(X,Y) \vee Suc_1(X,Y) \quad (3) \end{array}$$

For the induction hypothesis we assume for an $n \in \mathbb{N}$ a formula ϕ_n^P with a free variable P which holds in a binary $\mathcal{P}(\{P\})$-labelled tree T iff there are precisely two $\{P\}$-labelled nodes with a distance of $F(n)$.

Given ϕ_n^P, we define a formula ϕ_{n+1}^P with a free variable P which is satisfied by a $\mathcal{P}(\{P\})$-labelled tree iff there are precisely two $\{P\}$-labelled nodes with a distance of $F(n) * 2^{F(n)}$.

	X					Y
Block	0	1	2	...	$2^{F(n)}$	

The segment of a path from x to y can be thought of as being split into $2^{F(n)}$ blocks. Let the blocks be numbered such that the block closest to the root has the smallest number. Each such number can be coded binarly with $F(n)$ bits. Each block is supposed to consist of $F(n)$ nodes so that we can assign each block a bit, such that by convention the node closest to the root holds the least significant bit of the block's address.

	X					Y
Block	0	1	2	...	$2^{F(n)}$	
C	0000...0	1000...0	0100...0	...	1111...1	
B	1000...0	1000...0	1000...0	...	1000...0	

The assignment with bits is realised through a variable C, such that the valuation of C consists of all nodes which are labelled with bit value 1. For auxiliary purposes we will use a set variable B whose valuations consists of the nodes which mark the beginning of each block. In the following we specify the property which the valuations of B and C have to satisfy.

For auxiliary purposes, we introduce two set variables, I and J, which are assigned singletons of nodes with a distance of $F(n)$. The latter is realised through a formula ϕ_n^P. Therefor let P be a doubleton containing the nodes of I and J, only. We refer to I and J as pointers, because we will use them to express properties of any two nodes with a distance $F(n)$. The property *Pointers* specifies that P holds precisely (2) the two nodes from I and J (1) which have distance $F(n)$ (3).

$$\begin{array}{lll} \mathit{Pointers} := & \exists P. & \mathit{Sing}(I) \wedge \mathit{Sing}(J) \wedge I \subseteq P \wedge J \subseteq P \wedge \quad (1) \\ & & (\forall K. K \subseteq P \rightarrow \mathit{Empty}(K) \vee I \subseteq K \vee J \subseteq K) \wedge \quad (2) \\ & & \phi_n^P \quad (3) \end{array}$$

Sing is defined as in Appendix C. See the appendix also for the definitions of *Empty* and $=$.

If we have ensured that the pointers I and J have distance $F(n)$, we can specify the valuation of B. B has to contain X and Y (1) and all nodes in B have to have a distance multiple in $F(n)$, that is whenever I is in B, so is J and vice versa (2) and all nodes K in between are not in B (3).

$$\begin{array}{ll} \mathit{PropB} := & X \subseteq B \wedge Y \subseteq B \wedge \quad (1) \\ & (I \subseteq B \leftrightarrow J \subseteq B) \wedge \quad (2) \\ & \forall K. I < K \wedge K < J \rightarrow \mathit{Empty}(K) \vee K \notin B \quad (3) \end{array}$$

Thereby $\leq$ denotes the reflexive ancestor relation and $<$ its strict part. X is an ancestor of Y, $X \leq Y$, if both are in a "family" F (1) of which X is the origin (2) and the "parents" P of every "group" G within F is contained in F (3), or in other terms: F is closed under descendance.

$$\begin{array}{lll} X \leq Y := & \exists F. & X \subseteq F \wedge Y \subseteq F \wedge \quad (1) \\ & & \forall G. G \subseteq F \wedge \neg \mathit{Empty}(G) \rightarrow \neg(\mathit{Suc}_0(G, X) \vee \mathit{Suc}_1(G, X)) \wedge \quad (2) \\ & & \forall G. G \subseteq F \rightarrow \exists P. (\mathit{Suc}_0(G, P) \vee \mathit{Suc}_1(G, P)) \wedge P \subseteq F \quad (3) \end{array}$$

We define $<$ as the strict part of $X \leq Y$.

$$X < Y := \quad X \leq Y \wedge X \neq Y$$

For brevity we will write $I < K < J$ for $I < K \wedge K < J$ and similarly $I \leq K < J$ and $I < K \leq J$.

Connecting to our previous argument, we distinguish four defining properties of the construction. The first block has to code 0, so none of the nodes in the first block

should be contained in the valuation of C Respectively, the last block should code $F(n)$, so that the valuation of C should contain all nodes in the last block. We need to ensure that the least significant bit changes from one block to the next as the initial effect of incrementing the binary coded number. The second effect is that 1's are carried over to the next bit. The properties will be established respectively by *First*, *Last*, *Start* and *Carry*.

We specify the property *PropC* of the valuation of C. Let I and J satisfy *Pointers*. *First* specifies that all nodes in the first block, which begins with X, are not labelled C (1). *Start* specifies that the first nodes of two consecutive blocks are with respect to C not equally labelled (2). If I is C-labelled and J is not, then the block that I belongs to has a higher value on the bit pointed to by J then the block of J in the corresponding bit pointed to by J (3). Whence '1' needs to be carried over to the bit succeeding the one J points to. Let J^+ be an immediate successor of J and I^+ an immediate successor of I (4), unless I^+ has exceeded the boundaries of the block, J^+ is obtained by adding 1 to the bit of I^+ modulo 2 (5). Analogously to *First*, *Last* specifies that the nodes in the last block whose end is marked by Y are labelled with C.

$$
\begin{array}{lll}
\mathit{First} := & I \subseteq X \leftrightarrow \forall K.(I \leq K < J) \rightarrow \mathit{Empty}(K) \vee K \not\subseteq C & (1)\\
\mathit{Start} := & I \subseteq B \rightarrow \neg(I \subseteq C \leftrightarrow J \subseteq C) & (2)\\
\mathit{Carry} := & (I \subseteq C \wedge J \not\subseteq C) \leftrightarrow & (3)\\
& \forall I^+.\forall J^+. \quad \mathit{Sing}(I^+) \wedge (\mathit{Suc}_0(I, I^+) \vee \mathit{Suc}_1(I, I^+)) \wedge & \\
& \qquad \mathit{Sing}(J^+) \wedge (\mathit{Suc}_0(J, J^+) \vee \mathit{Suc}_1(J, J^+)) \rightarrow & (4)\\
& \qquad I^+ \subseteq B \vee (I^+ \not\subseteq C \leftrightarrow J^+ \subseteq C)) & (5)\\
\mathit{Last} := & J \subseteq Y \leftrightarrow \forall K.(I \leq K < J) \rightarrow K \subseteq C & (6)
\end{array}
$$

The valuation of C is determined by all of the four properties above.

$$\mathit{PropC} := \mathit{First} \wedge \mathit{Start} \wedge \mathit{Carry} \wedge \mathit{Last}$$

Now we can put the pieces together and formulate ϕ^P_{n+1} which specifies that there are precisely (2) two nodes X and Y in P (1) such that the segment of a path from X to Y is B and C labelled according to *PropB* and *PropC* (3), and is thus of length $F(n) * 2^{F(n)}$.

$$
\begin{array}{lll}
\phi^P_{n+1} := & \exists X.\exists Y. \quad \mathit{Sing}(X) \wedge X \subseteq P \wedge \mathit{Sing}(Y) \wedge Y \subseteq P \wedge & (1)\\
& \qquad (\forall Z.Z \subseteq P \rightarrow \mathit{Empty}(Z) \vee X \subseteq Z \vee Y \subseteq Z) \wedge & (2)\\
& \qquad \exists B.\exists C.\forall I.\forall J.\mathit{Pointers} \rightarrow \mathit{PropB} \wedge \mathit{PropC} & (3)
\end{array}
$$

Note that ϕ^P_n is not clean for $n > 1$. However, as we already argued in the preliminaries, ϕ^P_n can be brought into a clean form by distinctively renaming the bound variables.

Let $\mathbb{A}$ be an alternating binary tree automaton accepting precisely the $\mathcal{P}(\{P\})$-labelled trees satisfying ϕ^P_n. Then $\mathbb{A}$ has to have at least $F(n)$ states. Otherwise, while processing a tree T, one automaton state a would occur more than once between the nodes x and y. Because the acceptance condition can not distinguish between finite numbers of occurrences of automaton states, a could occur finitely, but arbitrarily often. Hence $\mathbb{A}$ could not recognise the distance between x and y.

However, ϕ_n^P is by construction linear in n. We obtain the following non-elementary lower bound for the translation of formulas into automata.

Proposition 6.3. *There is a formula ϕ of **S2S** for which there is no alternating binary tree parity automaton $\mathbb{A}$ of size elementary in $|\phi|$ such that $\mathbb{A}$ accepts a binary tree T iff T satisfies ϕ.*

In Chapter 5 we have seen that the existential projection of an alternating automaton is non-deterministic. Consequently, non-determinisation is needed only once in the translation of a sequence of nested existential quantifications. In fact, negation is the only logical operation that spoils non-determinism.

In ϕ_{n+1}^P, the alternation between the existential quantification over X, Y, B and C and the universal quantification over P makes non-determinisation of the automaton necessary in each induction step, which leads to the non-elementary complexity.

For non-deterministic automata, complementation increases the size of the automaton exponentially. Then, as Reinhard argues in [Rei02], negation is responsible or the non-elementary lower bound. Note that our result does not contradict Reinhard's observation.

Chapter 7

Non-Emptiness of Alternating Binary Tree Automata

The **non-emptiness problem** for an alternating binary tree automaton $\mathbb{A}$ consists in deciding whether $\mathbb{A}$ accepts some binary Σ-labelled tree, that is whether the language $\mathcal{L}(\mathbb{A})$ is non-empty. We reduce the non-emptiness problem for alternating automata to the non-emptiness problem for non-deterministic automata, for which we give a game theoretic solution.

Theorem 7.1. *The non-emptiness problem for alternating automata is decidable.*

As described in Chapter 4 we obtain from an alternating binary tree automaton $\mathbb{A}$ a non-deterministic binary tree automaton

$$\mathbb{B} = \langle B, \Sigma, \delta : B \times \Sigma \to \mathcal{P}(B \times B), b_I, \Omega \rangle$$

accepting precisely the same binary Σ-labelled trees as $\mathbb{A}$. Hence $\mathcal{L}(\mathbb{A})$ is non-empty iff $\mathcal{L}(\mathbb{B})$ is non-empty.

7.1 Non-Emptiness of Non-Deterministic Binary Tree Automata

Suppose there is a binary tree T for which $\exists$ has a winning strategy from the initial position in the game $\mathcal{G}(\mathbb{B}, T)$. At a basic position (p, b) of the game $\mathcal{G}(\mathbb{B}, T)$, $\exists$ chooses a pair (b_0, b_1) from $\delta(b, T(p))$ consistently with her winning strategy. Hence there is a letter $c \in \Sigma$, videlicet $T(p)$, for which $\exists$ can choose a pair (b_0, b_1) from $\delta(b, c)$, such that the next position, (pd, b) for both $d \in 2$, is winning for $\exists$. This idea underlies the construction of the **non-emptiness game**, $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$, a parity graph game, which proceeds according to the following rules.

Position	Player	Admissible Moves	Priority
$b \in B$	$\exists$	$\bigcup_{c\in\Sigma} \delta(b,c)$	$\Omega(b)$
$(b_0, b_1) \in B \times B$	$\forall$	$\{b_d \mid d \in 2\}$	0

Table 7.1: The Non-Emptiness Game $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$

1. At a position $b \in B$, $\exists$ chooses a letter $c \in \Sigma$ and a pair (b_0, b_1) from $\delta(b, c)$. The next position is then (b_0, b_1).

2. At a position $(b_0, b_1) \in B \times B$, $\forall$ chooses a direction $d \in 2$ which determines the next position b_d.

We call positions from the set B **basic**. Basic positions b are assigned the priority $\Omega(b)$, all other positions are assigned priority 0.

$\exists$ ($\forall$) wins a finite play π of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$, if the last position of π belongs to $\forall$ ($\exists$) and he (she) can not move. $\exists$ ($\forall$) wins an infinite play if the largest priority of an automaton state occurring infinitely often is even (odd). The non-emptiness game $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ is a parity graph game, and thus enjoys history-free determinacy. We refer to Appendix B for more details.

Proposition 7.2. *$\mathbb{B}$ accepts some binary Σ-labelled tree iff $\exists$ has a winning strategy in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ from the initial position b_I.*

Proof. For the left-to-right implication, suppose there is a binary Σ-labelled tree T for which $\exists$ has a winning strategy in the acceptance game $\mathcal{G}(\mathbb{B}, T)$ from the initial position. We derive a winning strategy for her in the non-emptiness game $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ from the initial position b_I, such that every play consistent with the derived strategy is over the same automaton states as a play in $\mathcal{G}(\mathbb{B}, T)$ consistent with $\exists$'s winning strategy. Initially the play in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ is at position b_I, any play in $\mathcal{G}(\mathbb{B}, T)$ is in (ϵ, b_I). Suppose the play of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ is at position b, then there is by the induction hypothesis a play of $\mathcal{G}(\mathbb{B}, T)$ consistent with $\exists$'s winning strategy over the same automaton states as the play in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$. Suppose $\exists$ chooses the pair (b_0, b_1) from $\delta(b, T(p))$, in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ let her choose the same pair. Any direction d which $\forall$ chooses in response, he could have chosen in $\mathcal{G}(\mathbb{B}, T)$, so that (pd, b_d) is a continuation of π consistent with $\exists$'s winning strategy. $\exists$ wins the play in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ because the largest priority of an automaton state occurring infinitely often in the play of $\mathcal{G}(\mathbb{B}, T)$ and thus also in the play of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ is even. Thus the derived strategy is winning for $\exists$.

For the right-to-left implication, suppose $\exists$ has a winning strategy in the non-emptiness game $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ from the initial position. We derive a binary Σ-labelled tree and a winning strategy for $\exists$ in the game $\mathcal{G}(\mathbb{B}, T)$, such that every play in $\mathcal{G}(\mathbb{B}, T)$ is over the automaton states of a play in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ consistent with $\exists$'s winning strategy. Initially, the play in $\mathcal{G}(\mathbb{B}, T)$ is at position (ϵ, b_I), any play of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ in b_I. Let the play in $\mathcal{G}(\mathbb{B}, T)$ be at position (p, b), then there is by the induction hypothesis as a play of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ over the same automaton states as the play in $\mathcal{G}(\mathbb{B}, T)$. Let (b_0, b_1) be the pair of automaton states which $\exists$ chooses at position b in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ consistent with her winning strategy. Then there is a letter $c \in \Sigma$ for which $(b_0, b_1) \in \delta(b, c)$. Define $T(p) := c$.

Let $\exists$ choose (b_0, b_1) at position (p, b) of $\mathcal{G}(\mathbb{B}, T)$. $\forall$ chooses a direction d, determining the next basic position (pd, b_d). $\forall$ could have chosen the same direction in $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$, so that basic position continuing π is b_d. We have established that for every play of $\mathcal{G}(\mathbb{B}, T)$ there is a play of $\mathcal{G}_{\neq\emptyset}(\mathbb{B})$ over the same automaton states. Thus $\exists$ wins the play in $\mathcal{G}(\mathbb{B}, T)$ because the largest priority of an automaton state occurring infinite often is even. Hence the derived strategy is winning for $\exists$. □

7.2 Complexity of the Non-Emptiness Problem

Because $\mathbb{B}$ has finitely many states, the non-emptiness game is a finite parity graph game. Deciding whether the initial position b_I of such a finite parity graph game is winning for $\exists$ or $\forall$ is called the parity graph game problem.

Definition 7.3 (The Parity Graph Game Problem)**.** *For a finite parity graph game $\mathcal{G}$, the parity graph game problem for $\mathcal{G}$ consists in determining whether a board position b is a winning position of $\exists$ or $\forall$.*

Emerson and Julta showed in [EJ91], that the parity graph game problem is at most exponential in the number of board positions. Given a partition of the board into winning regions, that is sets of winning positions, for $\exists$ and $\forall$, it can be verified in polynomial time, whether there is a consistent winning strategy from a particular board position. Hence the parity graph game problem is in NP, and in co-NP due to the symmetry of the problem. Moreover, as shown in [Jur98] that there is for every parity graph game precisely one such partition, so that the parity graph game problem is in U-NP and co-U-NP.

Chapter 8

Conclusions

This text is the first comprehensive recollection of Rabin's original argument improved in various aspects. Alternation facilitates complementation, though at the price of non-determinisation which is needed for the existential projection of alternating automata. Because of the parity acceptance condition the acceptance game is historyfree determined, which is crucial for the construction of the non-emptiness game.

The definitions of non-determinisation, existential projection and non-emptiness game are based on the more general coalgebraic constructions used in [KV05] and [KV07] for F-coalgebra automata. The relational approach to non-determinisation as given in Chapter 4 stems from [AN01]. Essential for the construction is the $\mathfrak{B}$-redistribution, an instance of the F-redistribution for the binary tree functor $\mathfrak{B}$ taking objects X to pairs $X \times X$. For the functor $\mathfrak{B}$, the F-redistribution is an easy construction. We expect a similar experience with other polynomial functors.

In Chapter 3 we have given an algorithm to compute the complement of a binary tree automaton. It is unlikely that the method used allows the generalisation to arbitrary F-coalgebra automata. In fact the question whether for every F-coalgebra automaton there is a complementary one is still open. Recall that we have defined universal quantification as the dual of existential quantification using negation. The weaker question, whether F-coalgebra automata are closed under universal projection, the dual of existential projection, is still open as well.

Appendix A

Sequences and Binary Trees

For any **alphabet** Σ we define sequences, or words, of letters from Σ and binary trees labelled with letters from Σ.

A.1 Sequences

Definition A.1 (Sequences)**.** *Given an alphabet* Σ*, a sequence of letters from* Σ*, or* Σ*-sequence or* Σ*-word for short, is a partial function* $U \dot{:} \omega \to \Sigma$ *such that if* $U(i)$ *is defined for some* $i \in \omega$*, so is* $U(j)$ *for all* j *below* i*.*

The **length** $|U|$ of a Σ-sequence is the number of indices i where $U(i)$ is defined. The empty Σ-sequence ϵ has length 0. A Σ-sequence is finite if there is an index $i \in \omega$ such that $U(i)$ is undefined, and infinite otherwise. The set of finite Σ-sequences is denoted as Σ^*, the set of infinite Σ-sequences is denoted as Σ^ω, and the set of all Σ-sequences is denoted as $\Sigma^\star$, so that $\Sigma^\star = \Sigma^* \cup \Sigma^\omega$.

We denote the last letter of a finite Σ-sequence U as $\mathit{Last}(U)$ and define it as $\mathit{Last}(U) := U(|U| - 1)$.

A Σ-sequence U' is an **initial subsequence**, or subsequence for short, of a Σ-sequence U if $|U'| \leq |U|$ and for all i where $U'(i)$ is defined, $U(i) = U'(i)$. A subsequence U' of a Σ-sequence U is proper if additionally $|U| < |U'|$.

Given a finite Σ-sequence U and an arbitrary Σ-sequence U', we define the **concatenation** UU' of U and U' such that for all $i < |U|$, $(UU')(i) := U(i)$ and for all $i \geq |U|$, $(UU')(i) := U'(i - |U|)$.

A.2 Binary Trees

In binary trees we distinguish two directions, "left" and "right", which are respectively denoted by 0 and 1 and collected in the two-element set 2. Finite paths through binary trees are then finite 2-sequences. We identify nodes with the respective path from the root node. The root node itself is then the empty sequence ϵ. A Σ-labelling associates nodes with letters from Σ. We identify full binary Σ-labelled trees with their labellings.

Definition A.2 (Binary Trees)**.** *Full binary Σ-labelled trees are functions*

$$T : 2^* \to \Sigma.$$

Given a node p, we call a node p' a child of p if p' is the left or right successor of p, that is $p' = p0$ or $p' = p1$. A node p' is a **descendent** of p, or p is an **ancestor** of p', if p is a proper initial subsequence of p'.

Remark A.3. *Note that in the literature, binary trees are often described as rooted acyclic graphs with out-degree 2, that is tuples $\langle V, E, r, \lambda\rangle$ consisting of a set V of vertices, a mapping $E : V \to V \times V$, a root node r, and a labelling $\lambda : V \to \Sigma$. Let $V = 2^*$, $E : p \mapsto (p0, p1)$ for all $p \in 2^*$, $r = \epsilon$, and $\lambda : p \mapsto T(p)$, then the definition by graphs is easily seen to be equivalent to our definition above.*

Appendix B

Game Theory

In the following we will formally introduce two-player parity graph games, or parity graph games for short. For a more concise account of these games the reader is referred to [GTW02].

Parity graph games are played by two players, $\exists$ (Eloise) and $\forall$ (Abelard) who oppose each other.

Definition B.1 (Parity Graph Games)**.** *A two-player parity graph game is a tuple* $\mathcal{G} = \langle B_\exists, B_\forall, E, b_I, \Omega \rangle$ *with*

- *disjoint sets* $B_\exists$ *and* $B_\forall$ *of board positions of* $\exists$ *and* $\forall$*, respectively*
- *a move relation* $E \subseteq B \times B$ *mapping positions to admissible successors*
- *an initial position* $b_I \in B_\exists \cup B_\forall$

The set $B = B_\exists \cup B_\forall$ is called the **board** of $\mathcal{G}$, the tuple $\langle B_\exists, B_\forall, E \rangle$ the **arena** of $\mathcal{G}$. We say positions from $B_\exists$ belong to $\exists$ and positions from $B_\forall$ belong to $\forall$.

Definition B.2 (Plays)**.** *A* ***partial play****, or play for short, is a sequence of board positions. We call a partial play* ***total*** *if it is infinite or if it is finite and there is no admissible successor for the last position. Otherwise a partial play is* ***initial****.*

A finite play π is won by $\exists$ ($\forall$) if the last position $Last(\pi)$ belongs to $\forall$ ($\exists$) and there is no admissible successor of $Last(\pi)$. We then say that $\forall$ ($\exists$) can not move in the last position of π.

The outcome of infinite plays of parity graph games is determined by the priorities of the board positions occurring in the play. $\exists$ wins an infinite play of a parity graph game if the largest priority occurring infinitely is even, $\forall$ wins if the largest priority occurring infinitely is odd.

Definition B.3 (Strategies)**.** *A* ***strategy*** *of* $\exists$ *($\exists$) in a parity graph game* $\mathcal{G}$ *as above from a position* b_0 *is a partial mapping taking each initial play* $b_0 b_1 \dots b_n$*, where* b_n *belongs to* $\exists$ *($\forall$), to an admissible successor of* b_n*.*

We say a play π is **played consistently** with a strategy σ of $\exists$ if for all proper initial plays π' of π, that is proper subsequences π' of π, where the last position of π' belongs to $\exists$, the next position equals the position determined by the strategy, that is $\pi(|\pi'|) = \sigma(\pi')$. The same definition applies to strategies of $\forall$ as well.

A strategy σ of $\exists$ is winning for $\exists$ if she wins any play consistent with σ. Similarly for strategies of $\forall$.

We call a strategy σ **positional** if for any two initial plays π and π' for which the last positions coincide, that is $Last(\pi) = Last(\pi')$, $\sigma(\pi)$ equals $\sigma(\pi')$. In informal terms, a positional strategy depends on the last position, only.

In parity graph games, plays consistent with strategies of a player constitute sets in a Borel algebra. Martin showed in [Mar75] that such sets are **determined**. As a consequence, in every position one of the players has a winning strategy. We call a position from which $\exists$ ($\forall$) has a winning strategy, a **winning position** of $\exists$ ($\forall$).

In 1991 it was shown independently by Emerson and Jutla in [EJ91], and Mostowski in [Mos91], that parity graph games are **historyfree determined**, that is they enjoy the following property. Whenever $\exists$ ($\forall$) has a winning strategy in a parity graph game $\mathcal{G}$ from a position b, then she (he) has a positional winning strategy from b.

Appendix C

Monadic Second-Order Logic for Binary Trees

In the following we define the monadic second-order logic **S2S** for binary trees, and the minimal second-order only language $\mathbf{S2S_{SO}}$ which we then prove equivalent to **S2S**. The latter entitles us to use $\mathbf{S2S_{SO}}$ as a replacement for **S2S** throughout the text.

C.1 S2S

Convention C.1. *We write individual variables in lower case, set variables in upper case.*

Definition C.2 (Syntax of **S2S**)**.** *Given individual variables x and y and a set variable X, we define formulas of **S2S** as follows.*

$$\phi ::= X(x) \mid suc_d(x,y) \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists x.\phi(x) \mid \exists X.\phi(X)$$

Moreover we define the usual abbreviations $\phi \wedge \psi := \neg\phi \vee \neg\psi$, $\phi \rightarrow \psi := \neg\phi \vee \psi$, $\phi \leftrightarrow \psi := (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$, $\forall x.\phi(X) := \neg\exists x.\neg\phi(x)$, and $\forall X.\phi(X) := \neg\exists X.\neg\phi(X)$.

A variable occurs freely in ϕ if it is not bound by a quantifier. Formally the set $FV_{FO}(\phi)$ of free individual variables and the set $FV_{SO}(\phi)$ of free set variables in a formula ϕ of **S2S** are defined as follows.

$$\begin{array}{ll}
FV_{FO}(X(x)) := \{x\} & FV_{SO}(X(x)) := \{X\} \\
FV_{FO}(suc_d(x,y)) := \{x,y\} & FV_{SO}(suc_d(x,y)) := \emptyset \\
FV_{FO}(\neg\phi) := FV_{FO}(\phi) & FV_{SO}(\neg\phi) := FV_{SO}(\phi) \\
FV_{FO}(\phi \vee \psi) := FV_{FO}(\phi) \cup FV_{FO}(\psi) & FV_{SO}(\phi \vee \psi) := FV_{SO}(\phi) \cup FV_{SO}(\psi) \\
FV_{FO}(\exists x.\phi) := FV_{FO}(\phi) \setminus \{x\} & FV_{SO}(\exists x.\phi) := FV_{SO}(\phi) \\
FV_{FO}(\exists X.\phi) := FV_{FO}(\phi) & FV_{SO}(\exists X.\phi) := FV_{SO}(\phi) \setminus \{X\}
\end{array}$$

Definition C.3 (Valuations)**.** *A valuation of an individual variable is a node $p \in 2^*$ in a binary tree, a valuation of all variables in a set W of individual variables is a function $w : W \to 2^*$. A valuation of a set variable X is a set $v_X \subseteq 2^*$, a valuation of a set V of set variables is a function $v : V \to \mathcal{P}(2^*)$ taking each set variable in V to its valuation. Note that such valuations v of sets of set variables are isomorphic to $\mathcal{P}(V)$-labelled trees $T : 2^* \to \mathcal{P}(V)$ by $X \in T(p)$ iff $p \in v(X)$ for all $p \in 2^*$. We call such a tree T the* ***tree representation*** *of the valuation v.*

Definition C.4 (Semantics of **S2S**)**.** *Let ϕ be a formula of* ***S2S****, and let W be a set containing the free individual variables, V a set containing the free set variables in ϕ. Given valuations $w : W \to 2^*$ and $v : V \to \mathcal{P}(2^*)$ of the variables in W and in V, respectively, the semantics of ϕ is defined inductively as follows.*

$$
\begin{array}{lll}
\Vdash_{w,v} X(x) & \textit{iff} & v(X) \subseteq v(Y) \\
\Vdash_{w,v} suc_d(x,y) & \textit{iff} & w(y) = w(x)d \\
\Vdash_{w,v} \neg\phi & \textit{iff} & \nVdash_{w,v} \phi \\
\Vdash_{w,v} \phi \vee \psi & \textit{iff} & \Vdash_{w,v} \phi \text{ or } \Vdash_{w,v} \psi \\
\Vdash_{w,v} \exists x.\phi & \textit{iff} & \Vdash_{w[x\mapsto w_x],v} \phi \text{ for some valuation } w_x \in 2^* \text{ of } x \\
\Vdash_{w,v} \exists X.\phi & \textit{iff} & \Vdash_{w,v[X\mapsto v_X]} \phi \text{ for some valuation } v_X \subseteq 2^* \text{ of } X
\end{array}
$$

Every formula ϕ of $\mathbf{S2S_{SO}}$ is definable with second-order constructions, only. In the following we introduce the language $\mathbf{S2S_{SO}}$, for binary trees which contains second-order constructions, only.

C.2 $\mathbf{S2S_{SO}}$

Definition C.5 (Syntax of $\mathbf{S2S_{SO}}$)**.** *Given set variables X and Y, we define formulas of* $\mathbf{S2S_{SO}}$ *inductively as*

$$\phi ::= X \subseteq Y \mid Suc_d(X,Y) \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid \exists X.\phi(X).$$

We also define the usual abbreviations $\phi \wedge \psi := \neg\phi \vee \neg\psi$, $\phi \to \psi := \neg\phi \vee \psi$, $\phi \leftrightarrow \psi := (\phi \to \psi) \wedge (\psi \to \phi)$, and $\forall X.\phi(X) := \neg\exists X.\neg\phi(X)$.

The atom $Suc_d(X,Y)$ means "all nodes in X have a successor in Y".

For a formula ϕ of $\mathbf{S2S_{SO}}$ we define the set $FV(\phi)$ of free set variables inductively such that

$$
\begin{array}{ll}
FV(X \subseteq Y) & := \{X, Y\} \\
FV(Suc_d(X,Y)) & := \{X, Y\} \\
FV(\neg\phi) & := FV(\phi) \\
FV(\phi \vee \psi) & := FV(\phi) \cup FV(\psi) \\
FV(\exists X.\phi) & := FV(\phi) \setminus \{X\}
\end{array}
$$

Definition C.6 (Semantics of $\mathbf{S2S_{SO}}$)**.** *Let ϕ be a formula of $\mathbf{S2S_{SO}}$, and let V be a set containing the free set variables in ϕ. Given a valuation $v : V \to \mathcal{P}(2^*)$ of the variables in V, the semantics of ϕ is defined inductively as follows.*

$$
\begin{array}{lll}
\Vdash_v X \subseteq Y & \textit{iff} & v(X) \subseteq v(Y) \\
\Vdash_v Suc_d(X, Y) & \textit{iff} & pd \in v(Y) \textit{ whenever } p \in v(X) \\
\Vdash_v \neg\phi & \textit{iff} & \nVdash_v \phi \\
\Vdash_v \phi \vee \psi & \textit{iff} & \Vdash_v \phi \textit{ or } \Vdash_v \psi \\
\Vdash_v \exists X.\phi & \textit{iff} & \Vdash_{v[X \mapsto v_X]} \phi \textit{ for some valuation } v_X \subseteq 2^* \textit{ of } X
\end{array}
$$

Let ϕ be a formula of $\mathbf{S2S_{SO}}$ with free set variables from a set V of variables. We say a valuation $v : V \to \mathcal{P}(2^*)$ of the variables in V **satisfies** ϕ, written $\Vdash_v \phi$ iff $[\![\phi]\!]_v$. A binary $\mathcal{P}(V)$-labelled tree T satisfies ϕ, $T \models \phi$ iff T is the tree representation of a valuation v satisfying ϕ.

C.3 Equivalence of S2S and $\mathbf{S2S_{SO}}$

In the following we show that **S2S** and $\mathbf{S2S_{SO}}$ are equivalent, that is mutually definable.

Given a formula ϕ in **S2S** with free individual variables from a set W and free set variables from a set V, we inductively define a formula ϕ_{SO} with free variables from $V \cup V'$ where V' is a set consisting precisely of set variables Y for which there is an individual variable y in W. To leverage the definition we assume that for no individual variable $y \in W$ there is a set variable $Y \in V$ in upper case, so that V and V' are disjoint.

Intuitively, we obtain ϕ_{SO} by replacing free individual variables y in ϕ whose valuations are nodes $p \in 2^*$ by free set variables Y whose valuations are singletons $\{p\}$. For the latter we define the singleton predicate *Sing* in $\mathbf{S2S_{SO}}$

$$Sing(X) := \neg Empty(X) \wedge \forall Y.Y \subseteq X \Rightarrow Empty(Y) \vee X = Y$$

where

$$Empty(X) := \forall Z.X \subseteq Z$$

and

$$X = Y := X \subseteq Y \wedge Y \subseteq X$$

Definition C.7 (Translation of **S2S** into $\mathbf{S2S_{SO}}$)**.** *We define the inductive translation of formulas ϕ of **S2S** into formulas ϕ_{SO} of $\mathbf{S2S_{SO}}$ as follows.*

$$
\begin{array}{ll}
(X(y))_{SO} & := Y \subseteq X \\
(suc_d(y_1, y_2))_{SO} & := Suc_d(Y_1, Y_2) \\
(\neg\phi(W, V))_{SO} & := \neg(\phi(W, V)_{SO} \\
(\phi(W, V) \vee \psi(W, V))_{SO} & := (\phi(W, V))_{SO} \vee (\psi(W, V))_{SO} \\
(\exists y.\phi)_{SO} & := \exists Y.Sing(Y) \wedge (\phi)_{SO} \\
(\exists X.\phi)_{SO} & := \exists X.(\phi)_{SO}
\end{array}
$$

The given translation preserves the semantics in the following sense.

Proposition C.8. *Let $V' := \{Y \mid y \in FV_{FO}(\phi)\}$ be a set of set variables each of which is an upper case correspondent of a free individual variable in ϕ. Every valuation $w : FV_{FO}(\phi) \to 2^*$ of the individual variables free in ϕ can be turned into a valuation v' of the set variables in V' such that*

$$v'(Y) := \{w(y)\}.$$

*A formula ϕ of **S2S** is then equivalent to a formula ϕ_{SO} of* $\mathbf{S2S_{SO}}$ *iff $\Vdash_{w,v} \phi$ iff $\Vdash_{v\cup v'} \phi_{SO}$.*

Proof. We show that $\Vdash_{w,v} \phi$ iff $\Vdash_{v\cup v'} \phi_{SO}$ by induction on ϕ.

In the first base case let $\phi = X(y)$. The valuations w and v satisfy $X(y)$ iff $w(y) \in v(X)$ by the semantics of **S2S**, which is iff $\{w(y)\} \subseteq v(X)$. By definition of v', the latter is iff $v'(Y) \subseteq v(X)$ which is iff $(v \cup v')(Y) \subseteq (v \cup v')(X)$. By the semantics of $\mathbf{S2S_{SO}}$, the latter is iff $\Vdash_{v\cup v'} Y \subseteq X$.

In the second base case let $\phi = suc_d(x, y)$. The valuations w and v satisfy $(suc)_d(x, y)$, $\Vdash_{w,v} suc_d(x, y)$, iff $w(y)$ is the successsor of $w(x)$ in direction d, $w(y) = (w(x))d$, which is iff $pd \in v'(Y)$ whenever $p \in v'(X)$. Because we assume V and V' to be disjoint, the union $v \cup v'$ is consistent, so that by the semantics of $\mathbf{S2S_{SO}}$ the latter is equivalent to $\Vdash_{v\cup v'} Suc_d(X, Y)$.

The boolean cases are immediate. The valuations w and v satisfy $\neg\phi$, $\Vdash_{w,v} \neg\phi$, iff $\nVdash_{w,v} \phi$ iff $\nVdash_{v\cup v'} \phi_{SO}$, by the induction hypothesis, which is iff $\Vdash_{v\cup v'} \neg\phi_{SO}$. For the disjunction, w and v satisfy $\phi \vee \psi$, $\Vdash_{w,v} \phi \vee \psi$, iff $\Vdash_{w,v} \phi$ or $\Vdash_{w,v} \psi$ which is iff $\Vdash_{v\cup v'} \phi_{SO}$ or $\Vdash_{v\cup v'} \psi_{SO}$ by the induction hypothesis. By the semantics of $\mathbf{S2S_{SO}}$ the latter is iff $\Vdash_{v\cup v'} \phi_{SO} \vee \psi_{SO}$

For the existential first-order quantification, w and v satisfy $\exists x.\phi$, $\Vdash_{w,v} \exists x.\phi$ iff there is a valuation w_x of the individual variable x such that $\Vdash_{w[x\mapsto w_x],v} \phi$. By the induction hypothesis, there is such a valuation w_x iff there is a valuation v_X, videlicet $v_x = \{w_x\}$, such that $\Vdash_{(v\cup v')[X\mapsto v_X]} \phi_{SO}$ and $\Vdash_{(v\cup v')[X\mapsto v_X]} Sing(X)$, which is, by the semantics of $\mathbf{S2S_{SO}}$, iff $\Vdash_{v\cup v'} \exists X.Sing(X) \wedge \phi_{SO}$.

For the existential second-order quantifation, w and v satisfy $\exists X.\phi$, $\Vdash_{w,v} \exists X.\phi$ iff there is a valuation v_X of the set variable X such that $\Vdash_{w,v[X\mapsto v_X]} \phi$ which is, by the induction hypothesis, iff $\Vdash_{(v\cup v')[X\mapsto v_X]} \phi_{SO}$. By the semantics of $\mathbf{S2S_{SO}}$ the latter is iff $\Vdash_{v\cup v'} \exists X.\phi_{SO}$. □

In an analogous manner we obtain a converse translation of $\mathbf{S2S_{SO}}$ into **S2S**.

Definition C.9 (Translation of $\mathbf{S2S_{SO}}$ into **S2S**)**.** *Given a formula ϕ of* $\mathbf{S2S_{SO}}$ *with free variables from a V of set variables, we define by induction a formula ϕ_{S2S} in **S2S** with free variables from V.*

$$\begin{array}{ll}
(X_1 \subseteq X_2)_{S2S} & := \forall y.X_1(y) \to X_2(y) \\
(Suc_d(X_1, X_2))_{S2S} & := \forall y_1.X_1(y_1) \to \exists y_2.suc_d(y_1, y_2) \wedge X_2(y_2) \\
(\neg\phi)_{S2S} & := \neg\phi_{S2S} \\
(\phi \vee \psi)_{S2S} & := \phi_{S2S} \vee \phi_{S2S} \\
(\exists X.\phi)_{S2S} & := \exists X.\phi_{S2S}
\end{array}$$

The given translation preserves the semantics in the following sense.

Proposition C.10. *Let ϕ be a formula of* $\mathbf{S2S_{SO}}$*, then for every valuation $v : FV(\phi) \to 2^*$ of the set variables free in ϕ, $\Vdash_v \phi$ iff $\Vdash_{\emptyset,v} \phi_{S2S}$.*

Proof. We show by induction on ϕ that $\Vdash_v \phi$ iff $\Vdash_{\emptyset,v} \phi_{S2S}$.

In the first base case let $\phi = X_1 \subseteq X_2$. The valuation v satisfies $X_1 \subseteq X_2$, $\Vdash_v X_1 \subseteq X_2$, iff $v(X_1) \subseteq v(X_2)$, that is for all nodes p, $p \in v(X_2)$ whenever $p \in v(X_1)$. By the semantics of **S2S**, the latter is iff $\Vdash_{\emptyset,v} \forall y.X_1(y) \to X_2(y)$.

In the second base case let $\phi = Suc_d(X_1, X_2)$. The valuation v satsifies $Suc_d(X_1, X_2)$ iff for all nodes $p \in v(X_1)$, $pd \in v(X_2)$, that is there is a node p', successor of p in direction d, which is in $v(X_2)$. By the semantics of **S2S**, the latter is iff $\Vdash_{\emptyset,v} \forall y_1.X_1(y_1) \to \exists y_2.suc(y_1, y_2) \wedge X_2(y_2)$.

The boolean cases are immediate. For the negation, $\Vdash_v \neg\phi$ iff $\nVdash_v \phi$ iff $\nVdash_{\emptyset,v} \phi_{FO}$ by the induction hypothesis which is iff $\Vdash_{\emptyset,v} \neg\phi_{S2S}$. For the disjuntion, $\Vdash_v \phi \vee \psi$ iff $\Vdash_v \phi$ or $\Vdash_v \psi$ which is by the induction hypothesis iff $\Vdash_{\emptyset,v} \phi_{S2S}$ or $\Vdash_{\emptyset,v} \psi_{S2S}$ which in turn is iff $\Vdash_{\emptyset,v} \phi_{S2S} \vee \psi_{S2S}$.

For the existential second-order quantification, $\Vdash_v \exists X.\phi$ iff there is a valuation v_X of X such that $\Vdash_{v[X\mapsto v_X]} \phi$ which is by the induction hypothesis iff $\Vdash_{\emptyset,v[X\mapsto v_X]} \phi_{S2S}$. By the semantics of **S2S** the latter is iff $\Vdash_{\emptyset,v} \exists X.\phi_{S2S}$. □

Index

Bibliography

[AN01] Andre Arnold and Damien Niwinsky, editors. *Rudiments of μ-Calculus*, volume 540-28739v. 146 of *Studies in Logic and the Foundations of Mathematics*. North Holland, February 2001.

[Büc62] Julius Richard Büchi. On a decision method in restricted second-order arithmetic. In *International Congress on Logic, Methodology, and Philosophy of Science*, pages 1–11. Stanford University Press, 1962.

[CGP99] Edmund M. Clark, Orna Grumberg, and Doron A. Peled, editors. *Model Checking*. MIT Press, 1999.

[CKS81] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.

[EJ91] E. Allen Emerson and Charanijt S. Jutla. Tree automata, μ-calculus and determinacy. In *Proceedings of the 32nd annual symposium on Foundations of computer science*, pages 368–377, Los Alamitos, CA, USA, 1991. IEEE Computer Society Press.

[Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. pages 21–51, 1961.

[GH82] Yuri Gurevich and Leo Harrington. Trees, Automata, and Games. In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 60–65, New York, NY, USA, 1982. ACM Press.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[Jur98] Marcin Jurdzinski. Deciding the winner in parity games is in UP and co-UP. *Information Processing Letters*, 68(3):119–124, 1998.

[KV05] Clemens Kupke and Yde Venema. Closure properties of coalgebra automata. In *LICS '05: Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS' 05)*, pages 199–208, Washington, DC, USA, 2005. IEEE Computer Society.

[KV07] Clemens Kupke and Yde Venema. Coalgebraic automata theory: Basic results. Technical report, CWI Amsterdam, 2007.

[Mar75] Donald A. Martin. Borel determinacy. pages 363–371, 1975.

[McN66] Robert McNaughton. Testing and generating infinite sequences by a finite automata, 1966.

[Mos91] Andrzej Mostowski. Games with forbidden positions. Technical Report 78, University of Gdansk, 1991.

[MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra. pages 69–107, 1995.

[Pit06] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. *lics*, 0:255–264, 2006.

[Rab69] Michael Oser Rabin. *Decidability of Second-Order Theories and Automata on Infinite Trees*, pages 1–35. 1969.

[Rei02] Klaus Reinhardt. *The Complexity of Translating Logic to Finite Automata*, pages 231–238. Springer Berlin/Heidelberg, 2002.

[Saf88] Shmuel Safra. On the complexity of ω-automata. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science FoCS '88*, pages 319–327. IEEE Computer Society Press, 1988.

[Ven04] Yde Venema. Automata and fixed point logics for coalgebras. *Electronic Notes in Computer Science*, 106:355–375, 2004.