Automata and Logic

Radu Iosif, VERIMAG/CNRS

Contents

Chapter 1

Set Theoretic Notions

1.1 Sets, Relations, Functions

Definition 1 *A* set *is a collection of objects.*

$x \in A$ denotes the fact that x is a member of A. $\emptyset$ denotes the empty set. Boolean operations on sets:

- union $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$
- intersection $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$
- complement $\bar{A} = \{x \mid x \notin A\}$

Laws of boolean algebra:

Associativity

$$\begin{aligned} A \cup (B \cup C) &= (A \cup B) \cup C \\ A \cap (B \cap C) &= (A \cap B) \cap C \end{aligned}$$

Commutativity

$$\begin{aligned} A \cup B &= B \cup A \\ A \cap B &= B \cap A \end{aligned}$$

Distributivity

$$\begin{aligned} (A \cup B) \cap C &= (A \cap C) \cup (B \cap C) \\ (A \cap B) \cup C &= (A \cup C) \cap (B \cup C) \end{aligned}$$

Absorption

$$\begin{aligned} A \cup (A \cap B) &= A \\ A \cap (A \cup B) &= A \end{aligned}$$

Idemptotence, Involution, Boundedness

$$A \cup A = A \qquad A \cap A = A$$
$$\bar{\bar{A}} = A$$
$$A \cup \emptyset = A \qquad A \cap \emptyset = \emptyset$$

De Morgan laws

$$\begin{aligned} \overline{A \cup B} &= \bar{A} \cap \bar{B} \\ \overline{A \cap B} &= \bar{A} \cup \bar{B} \end{aligned}$$

Exercise 1 *Prove that $A \cap \bar{A} = \emptyset$. What is $A \cup \bar{A}$?* $\square$

$A \setminus B = A \cap \bar{B}$. $X \subseteq A$ iff $X \cup A = A$. $X = A$ iff $X \subseteq A$ and $A \subseteq X$. 2^A is the *powerset* of A, i.e. $2^A = \{X \mid X \subseteq A\}$. $\|A\|$ is the number of elements in A.

Exercise 2 *Prove $\|2^A\| = 2^{\|A\|}$, for any finite set A.* $\square$

A *partition* of A is a set $\mathcal{P} \subseteq 2^A$ such that $\bigcup_{X \in \mathcal{P}} X = A$ and, for all $X, Y \in \mathcal{P}$ such that $X \neq Y$, we have $X \cap Y = \emptyset$.

Exercise 3 *Give and prove an estimation of the number of partitions of a finite set A.* $\square$

Definition 2 *The Cartesian product of two sets A and B, denoted as $A \times B$, is defined as the set of pairs $\langle x, y\rangle$ where $x \in A$ and $y \in B$. The Cartesian product of n sets $A_1 \times A_2 \times \ldots \times A_n$ is the set of n-tuples $\langle x_1, x_2, \ldots, x_n\rangle$ where $x_i \in A_i$, for all $1 \leq i \leq n$.*

$R \subseteq A \times B$ is a *relation* between two sets A and B. aRb denotes the fact that $\langle a, b\rangle \in R$. A relation $R \subseteq A \times B$ is said to be *binary*. A relation $R \subseteq A_1 \times A_2 \times \ldots \times A_n$ is said to be *n-ary*. A relation R is *functional* iff for all $a \in A$ and $b, b' \in B$, aRb and aRb' implies $b = b'$. A functional relation is also called a *partial function*.

$$\begin{array}{rcl} dom(R) & = & \{x \in A \mid \text{there exists } y \in B \text{ such that } f(x) = y\} \\ rng(R) & = & \{y \in B \mid \text{there exists } x \in A \text{ such that } f(x) = y\} \end{array}$$

A *function* $f : A \to B$ is a functional relation between A and B such that for all $a \in A$ there exists $b \in B$ such that $a\ f\ b$. In the following $a\ f\ b$ is denoted as $f(a) = b$. The set $\{\langle a, b\rangle \in A \times B \mid f(a) = b\}$ is sometimes called the *graph* of f.

Properties of functions:

- f is *one-to-one* (*injective*) if for all $a, b \in A$, $f(a) = f(b)$ implies $a = b$.
- f is *onto* (*surjective*) if for all $b \in B$ there exists $a \in A$ such that $f(a) = b$.
- f is *bijective* iff it is one-to-one and onto.

Example 1 *A* permutation *is a bijective function $\pi : \{1, 2, \ldots, N\} \to \{1, 2, \ldots, N\}$ for some $N \in \mathbb{N}$.* $\square$

The restriction of a function $f : A \to B$ to a set $X \subseteq A$ is denoted as $f\downarrow_X = \{\langle x, y\rangle \mid x \in X \text{ and } f(x) = y\}$.

Definition 3 *An* equivalence relation *on a set A is a relation $R \in A \times A$ that satisfies the following conditions:*

1. reflexivity*: for all $x \in A$. xRx*
2. symmetry*: for all $x, y \in A$. xRy implies yRx*
3. transitivity*: for all $x, y, z \in A$. xRy and yRz implies xRz*

Exercise 4 *Let $n > 0$ be a natural number. Define the relation $\equiv_n \in \mathbb{N} \times \mathbb{N}$ as $x \equiv_n y$ if and only if n divides $|x - y|$. Show that $\equiv_n$ is an equivalence relation.* $\square$

Exercise 5 *Let $A = \{a_1, a_2, \ldots, a_K\}$ and A^n be the set:*

$$\underbrace{A \times A \times \ldots \times A}_{n \text{ times}}$$

and the relation $\sim$ on A^n defined as: $\langle a_{i_1}, a_{i_2} \ldots a_{i_K} \rangle \sim \langle a_{j_1}, a_{j_2} \ldots a_{j_K} \rangle$ iff there exists a permutation π of $\{1, 2, \ldots, K\}$ such that $\pi(i_k) = j_k$, for all $1 \leq k \leq K$. Show that $\sim$ is an equivalence relation on A^n. $\square$

Given $a \in A$, the *equivalence class* of a with respect to R is $[a]_R = \{b \in A \mid aRb\}$. This is called sometimes the *orbit* of a. The *index* of an equivalence relation is the number of equivalence classes.

Exercise 6 *Given R an equivalence relation on A, show that $\{[a]_R \mid a \in A\}$ is a partition of A.* $\square$

If $f : A \to B$ is a function, the *kernel* of f is the relation $ker(f) \subseteq A \times A$ defined as $ker(f) = \{\langle x, y \rangle \mid f(x) = f(y)\}$.

Exercise 7 *Show that $ker(f)$ is an equivalence relation, for any function $f : A \to B$. Next, show that for any equivalence relation $R \subseteq A \times A$ there exists a function f with $dom(f) = A$ such that $R = ker(f)$.* $\square$

1.2 Orders

1.2.1 Partial Orders

Definition 4 *Let A be a set. A binary relation $\preceq \subseteq A \times A$ is called a* partial order *iff it satisfies the following conditions:*

1. reflexivity*: for all $a \in A$. $a \preceq a$*
2. antisymmetry*: for all $a, b \in A$. $a \preceq b$ and $b \preceq a$ implies $a = b$*

3. transitivity*: for all* $a, b, c \in A$ *.* $a \preceq b$ *and* $b \preceq c$ *implies* $a \preceq c$*.*

$\langle A, \preceq \rangle$ *is said to be a* partially ordered set*. We denote by* $a \prec b$ *the fact that* $a \preceq b$ *but* $b \not\preceq a$*.*

A subset $X \subseteq A$ has $l \in A$ as an *upper bound* if for all $l' \in X$ we have $l' \preceq l$ and as a *lower bound* if for all $l' \in X$ we have $l \preceq l'$. A *least upper bound* (lub) l of X is an upper bound that satisfies $l \preceq l'$ for each upper bound l' of X. A *greatest lower bound* (glb) l of X is a lower bound that satisfies $l' \preceq l$ for each lower bound l' of X. When lub's and glb's exists they are unique (because of antisymmetry), and they are denoted by $\sqcup X$ and $\sqcap X$.

Definition 5 *A* complete lattice *is a partially ordered set* $\langle A, \preceq \rangle$ *such that all subsets of* A *have lub's and glb's.*

Exercise 8 *Show that, for any set* A*,* $\langle 2^A, \subseteq \rangle$ *and* $\langle 2^A, \supseteq \rangle$ *are complete lattices.* □

A function $f : A \to A$ is said to be *monotone* if $x \preceq y \Rightarrow f(x) \preceq f(y)$. A *fixed point* of f is $x \in A$ such that $f(x) = x$. The *least fixed point* of f is denoted by $lfp(f)$, and the *greatest fixed point* by $gfp(f)$.

Theorem 1 (Tarski's Fixed Point Theorem) *Let* $\langle A, \preceq \rangle$ *be a complete lattice. If* $f : A \to A$ *is a monotone function, then:*

$$\begin{aligned} lfp(f) &= \sqcap\{l \mid f(l) \preceq l\} \\ gfp(f) &= \sqcup\{l \mid l \preceq f(l)\} \end{aligned}$$

An *ascending chain* in A is a (possibly infinite) sequence $a_1 \preceq a_2 \preceq \ldots$ of elements of A. A *descending chain* in A is a (possibly infinite) sequence $a_1 \succeq a_2 \succeq \ldots$ of elements of A.

Example 2 $\mathbb{N} \supset \mathbb{N} \setminus \{0\} \supseteq \mathbb{N} \setminus \{0, 1\} \supseteq \mathbb{N} \setminus \{0, 1, 2\} \supseteq \ldots$ *is an infinite descending chain in* $2^{\mathbb{N}}$*.* □

A satisfies the ascending (descending) chain condition iff all ascending (descending) chains eventually stabilize, i.e. exists n such that for all $i \geq n$, $a_i = a_{i+1}$.

A *strictly descending chain* in A is a (possibly infinite) sequence $a_1 \succ a_2 \succ \ldots$ of elements of A.

Definition 6 *A partially ordered set* $\langle A, \preceq \rangle$ *is said to be* well-founded *iff every strictly descending chain in* A *is finite. The partial order* $\preceq$ *is also said to be* well-founded *in this case.*

Example 3 $\langle \mathbb{N}, \leq \rangle$ *is a well-founded set.* $\langle \mathbb{Z}, \leq \rangle$ *is not well-founded.* □

Exercise 9 *Let $\langle A, \preceq \rangle$ be a well-founded set. Show that, for any $n \in \mathbb{N}$, the set $\langle A^n, \preceq^n \rangle$ (see Ex. 5) is well-founded, where $\langle a_1, a_2, \ldots, a_n \rangle \preceq^n \langle b_1, b_2, \ldots, b_n \rangle$ iff $a_i \preceq b_i$, for all $1 \leq i \leq n$.* $\Box$

Example 4 *$\langle \mathbb{N}, \leq \rangle$ is well founded, whereas $\langle \mathbb{Z}, \leq \rangle$ is not.* $\Box$

If $\langle A, \preceq \rangle$ is a partially ordered set, $a \in A$ is said to be a minimal (maximal) element iff for no $b \in A$ we have that $b \prec a$ ($a \prec b$).

Exercise 10 *Let $S \subseteq \mathbb{N}^n$ for some $n \in \mathbb{N}$. Show that the set of minimal elements of S is finite.* $\Box$

1.2.2 Linear Orders

Definition 7 *Let $\langle A, \preceq \rangle$ be a partially ordered set. $\preceq$ is said to be a* linear order *iff for all $a, b \in A$ we have either $a \preceq b$ or $b \preceq a$. $\langle A, \preceq \rangle$ is called a* linearly ordered set *in this case.*

Linear orders are sometimes called *total*.

Let $\langle A, \preceq_A \rangle$ and $\langle B, \preceq_B \rangle$ be two linearly ordered sets.

We define the sum $\langle A, \preceq_A \rangle + \langle B, \preceq_B \rangle = \langle A \cup B, \preceq_+ \rangle$, where $a \preceq_+ b$ iff either:

- $a, b \in A$ and $a \preceq_A b$, or
- $a, b \in B$ and $a \preceq_B b$, or
- $a \in A$ and $b \in B$.

Exercise 11 *Let $\omega_0 = \langle \{0\} \times \mathbb{N}, \leq_0 \rangle$ and $\omega_1 = \langle \{1\} \times \mathbb{N}, \leq_1 \rangle$, where $\langle i, n_1 \rangle \leq_i \langle i, n_2 \rangle$ iff $n_1 \leq n_2$, $i = 0, 1$. Prove that $\omega_0 + \omega_1$ is well-founded.* $\Box$

We define the product $\langle A, \preceq_A \rangle \times \langle B, \preceq_B \rangle = \langle A \cup B, \preceq_\times \rangle$, where $\langle a_1, b_1 \rangle \preceq_\times \langle a_2, b_2 \rangle$ iff either:

- $a_1 \preceq_A a_2$ and $a_1 \neq a_2$, or
- $a_1 = a_2$ and $b_1 \preceq_B b_2$.

This is also called *lexicographic order*. The definition of the generalization $\langle A_1, \preceq_1 \rangle \times \langle A_2, \preceq_2 \rangle \times \ldots \times \langle A_n, \preceq_n \rangle$ is left as an exercise.

Exercise 12 *Given two linearly ordered sets $\langle A, \preceq_A \rangle$ and $\langle B, \preceq_B \rangle$, prove that $\langle A, \preceq_A \rangle + \langle B, \preceq_B \rangle$ and $\langle A, \preceq_A \rangle \times \langle B, \preceq_B \rangle$ are linearly ordered sets as well.* $\Box$

Exercise 13 *If $\langle A, \preceq \rangle$ is a linearly ordered set, let $\langle A, \preceq \rangle^* = \Sigma\Pi_{n \in \mathbb{N}} \langle A, \preceq \rangle^n$. Show that $\langle \mathbb{N}, \leq \rangle^*$ is not well-founded.* $\Box$

1.3 Cardinality, Induction, Pigeonhole, Diagonalization

Definition 8 *We say that two sets A and B* have the same size *iff there exists a bijection $f : A \to B$.*

Exercise 14 *Let $\equiv$ be a relation on the set of all sets, defined as $A \equiv B$ if A and B have the same size. Prove that $\equiv$ is an equivalence relation.* $\square$

Exercise 15 *Prove that the following sets have the same size: $\mathbb{N}$, $\mathbb{Z}$, $\{2x+1 \mid x \in \mathbb{N}\}$, $\mathbb{N}^2$, $\mathbb{Q}$ and $\{X \subset \mathbb{N} \mid X \text{ is finite}\}$. Notice that we can have $A \equiv B$ even when $A \subset B$.* $\square$

Definition 9 *A set A is said to be* countable *iff it has the same size as $\mathbb{N}$ or one of its subsets.*

Principle 1 (Natural Induction) *Let P be a property of natural numbers. If both the following hold:*

1. *$P(0)$ is true, and*
2. *for all $n \in \mathbb{N}$: if $P(n)$ is true then $P(n+1)$ is true*

then, for all $n \in \mathbb{N}$, $P(n)$ is true.

Exercise 16 *Show that $n > 0$ lines may split the cartesian plane in at most $1 + \frac{n(n+1)}{2}$ convex polyhedra.* $\square$

Principle 2 (Well-founded Induction) *Let $\langle W, \preceq \rangle$ be a well-founded set, and P a property of the elements of W. If both the following hold:*

1. *P is true for all minimal elements of W,*
2. *for all $x \in W$: if $P(y)$ is true for all $y \prec x$ then $P(x)$ is true*

then, for all $x \in W$, $P(x)$ is true.

Notice that the first condition of the well-founded induction principle is implied by the second one. However, the definition above is common in textbooks.

Exercise 17 *Prove that every natural number has a prime factorization. Prove that this factorization is unique.* $\square$

Example 5 *Consider the set $\mathbb{N} \cup \{\infty\}$, where $n \leq \infty$ for all $n \in \mathbb{N}$, and the property $P(n)$:* the set $[0, n+1) \cap \mathbb{N}$ is finite. *Obviously, for all $n \in \mathbb{N}$ we have that $P(n)$ is true. However, $P(\infty)$ is false. The reason is that $\mathbb{N}^\infty$ is not well-founded (why?).* $\square$

Principle 3 (Pigeonhole) *If A and B are finite sets and $\|A\| \leq \|B\|$, then there is no one-to-one function $f : A \to B$.*

Exercise 18 *Prove the pigeonhole principle correct using natural induction.*$\square$

Principle 4 (Diagonalization) *Let A be a (possibly infinite) set and $R \subseteq A \times A$ be a binary relation on A. For each $a \in A$ let $R_a = \{b \mid b \in A$ and $\langle a, b\rangle \in R\}$. Let $D = \{a \mid \langle a, a\rangle \notin R\}$. Then D is distinct from each R_a.*

Exercise 19 *Prove that $2^{\mathbb{N}}$ is uncountable.*$\square$

Exercise 20 *Prove that $\mathbb{R}$ is uncountable.*$\square$

Exercise 21 *Let R be the set of all sets which are not members of themselves, i.e. $R = \{X \mid X \notin X\}$. Is the statement $R \in R$ true or false?* $\square$

1.4 Words, Trees, Graphs

1.4.1 Words

An *alphabet* is a finite non-empty set of symbols $\Sigma = \{a, b, c, \ldots\}$. A *word* of length n over Σ is an element of Σ^n, represented as $w = a_0 a_1 \ldots a_{n-1}$, where $a_i \in \Sigma$, for all $0 \leq i \leq n-1$. Equivalently, a word is a function $w : \{0, 1, \ldots, n-1\} \to \Sigma$. The length n of the word w is denoted by $|w|$. The *empty word* is denoted by ϵ, i.e. $|\epsilon| = 0$. We denote by Σ^* the set of all finite words over Σ, i.e $\Sigma^* = \bigcup_{n \in \mathbb{N}} \Sigma^n$.

The *concatenation* of two words w and u is denoted as wu. The *prefix relation* $\leq \subseteq \Sigma^* \times \Sigma^*$ is defined as $u \leq w$ iff there exists $v \in \Sigma^*$ such that $uv = w$.

Exercise 22 *Prove that $\langle \Sigma^*, \leq\rangle$ is a well-founded set.*$\square$

An *infinite word* over Σ is an infinite sequence $w = a_1 a_2, \ldots$, where $a_i \in \Sigma$, for all $i > 0$. Equivalently, an infinite word is a function $f : \mathbb{N} \to \Sigma$. We denote by Σ^ω the set of all infinite words over Σ. Let $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$ be the set of all words, either finite or infinite.

Exercise 23 *Let $w \in \Sigma^\omega$ be an infinite word. A prefix of w is a finite word $u \in \Sigma^*$ such that $uv = w$ for some infinite word $v \in \Sigma^\omega$. Let $Pref(w) \subseteq \Sigma^*$ be the set of all prefixes of w. Show that $\langle Pref(w), \leq\rangle$ is a linearly ordered set.* $\square$

A *prefix-closed* set $S \in \Sigma^*$ is a set such that for all $w \in S$ and $u \in \Sigma^*$, if $u \leq w$ then $u \in S$.

1.4.2 Trees

A *tree* over Σ is a partial function $t : \mathbb{N}^* \to \Sigma$ such that $dom(t)$ is a prefix-closed set:

- for each $p \in dom(t)$ for all $q \leq p$ we have $q \in dom(t)$.

A word $p \in dom(t)$ is called a *position*. If $p, q \in dom(t)$ such that $pn = q$ for some $n \in \mathbb{N}$, we say that p is the *parent* of q and that q is the *n-th child* of p.

Exercise 24 *Show that for every tree t, $\langle dom(t), \leq \rangle$ has exactly one minimal element. This position is called the* root *of the tree.* $\square$

A tree t is said to be *finite-branching* iff for any $p \in dom(t)$, the number of children of p is finite. Otherwise t is said to be *infinite-branching*. A tree t is said to be *finite* if $dom(t)$ is finite. Otherwise t is said to be *infinite*. We denote by $\mathcal{T}(\Sigma)$ the set of all finite trees, and by $\mathcal{T}^\omega(\Sigma)$ the set of all infinite trees over Σ. Let $\mathcal{T}^\infty(\Sigma) = \mathcal{T}(\Sigma) \cup \mathcal{T}^\omega(\Sigma)$ be the set of all trees, either finite or infinite.

Given a finite tree $t \in \mathcal{T}(\Sigma)$, the *frontier* of t is the set $fr(t) = \{p \in dom(t) \mid \text{for all } n \in \mathbb{N}\, pn \notin dom(t)\}$. A *path* in t is a maximal subset π of $dom(t)$ linearly ordered by $\leq$. Given $p \in dom(t)$, the *subtree* t_p is defined as $t_p : \{q \in \mathbb{N}^* \mid pq \in dom(t)\} \to \Sigma$, such that $t_p(q) = t(pq)$, for all $q \in dom(t_p)$.

Lemma 1 (König) *A finitely branching tree is infinite if and only if it has an infinite path.*

Proof: Let t be an infinite tree with finite branching. We build the infinite path π recursively. Let ϵ be the first position on the path. Suppose that we have built a prefix $\pi^i = p_0 p_1 \ldots p_{i-1}$. Then one of the children of p_{i-1} must be the root of an infinite subtree of t, otherwise if all children would be roots of finite subtrees, since there is a finite number of them, the entire tree would be finite. Let p_i be this position, and π^{i+1} be $\pi^i \cdot p_i$. Continuing like this we build an infinite path in t. $\square$

Exercise 25 *We denote by $t \preceq t'$ the fact that t is a subtree of t'. Show that $\langle \mathcal{T}(\Sigma), \preceq \rangle$ is a partially ordered set.*$\square$

A *ranked alphabet* $\langle \Sigma, \# \rangle$ is a set of symbols together with a function $\# : \Sigma \to \mathbb{N}$. For $f \in \Sigma$, the value $\#(f)$ is said to be the *arity* of f. A *ranked tree* t over Σ is a partial function $t : \mathbb{N}^* \to \Sigma$ that satisfies the following conditions:

- $dom(t)$ is a finite prefix-closed subset of $\mathbb{N}^*$, and
- for each $p \in dom(t)$, if $\#(t(p)) = n > 0$ then $\{i \mid pi \in dom(t)\} = \{1, \ldots, n\}$.

The *size* and *height* of a tree are defined inductively, as follows:

- $\|t\| = height(t) = 1$ if $t = a$, for some $a \in \Sigma$, $\#(a) = 0$,
- $\|t\| = 1 + \sum_{i \in \{1,\ldots,n\}} \|t_i\|$ and $height(t) = 1 + \max(\{height(t_i) \mid i \in \{1, \ldots, n\}\})$.

A finite tree over a ranked alphabet is also called a *term*. The set of all terms over Σ is denoted by $\mathcal{T}(\Sigma)$.

Let $\sigma \in \Sigma$. By $T \cdot_\sigma T'$ we denote the set of trees obtained from some $t \in T$ by replacing each occurrence of σ on $fr(t)$ by a tree in T'. If $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$, let $T \cdot_\sigma \langle T_1, \ldots, T_m \rangle$ be the set of trees obtained from some $t \in T$ by replacing each occurrence of σ_i on $fr(t)$ by a tree in T_i. We denote by $\langle T_1, \ldots, T_m \rangle^{\omega\sigma}$ the set of infinite trees obtained by the infinite unfolding of the concatenation operation.

Let $X = \{x_1, \ldots, x_n\}$ be a finite set of variables, interpreted over terms. By convention, all variables are symbols of zero arity. A term $t \in \mathcal{T}(\Sigma \cup X)$ is said to be *linear* if each variable occurs in t at most once. A *context* is a linear term $C[x_1, \ldots, x_n]$, and $C[t_1, \ldots, t_n]$ denotes the result of replacing x_i with the term t_i, for all $1 \leq i \leq n$. A context is said to be *trivial* if it is reduced to a variable, and *non-trivial* otherwise.

Exercise 26 *A* binary tree *is a tree over the binary alphabet $B = \{\cdot, \Box\}$, where $\#(\cdot) = 2$ and $\#(\Box) = 0$. Show that, for any (unranked) alphabet Σ the sets $\mathcal{T}^\infty(\Sigma)$ and $\mathcal{T}^\infty(B)$ have the same size.* $\Box$

1.4.3 Graphs

A *graph* is a tuple $G = \langle V, E_1, E_2, \ldots, E_n \rangle$ where:

- V is a (usually finite) set of *vertices*,
- $E_i \subseteq V \times V$ are the *edge relations*, $1 \leq i \leq n$.

We denote $\langle u, v \rangle \in E_i$ by $u \rightarrow_i v$. A graph with edge relations that are of arity $n > 2$ is called a *hypergraph*.

Given an alphabet Σ, a Σ-*labeled graph* is a pair $G = \langle V, E \rangle$, where:

- V is the set of vertices,
- $E \subseteq V \times \Sigma \times V$ is the edge relation.

We denote $\langle u, a, v \rangle \in E$ by $u \xrightarrow{a} v$. Notice that, if Σ is finite, a Σ-labeled graph is a graph.

Two Σ-labeled graphs $G_1 = \langle V_1, E_1 \rangle$ and $G_2 = \langle V_2, E_2 \rangle$ are said to be *isomorphic* iff there exists a bijection $\beta : V_1 \to V_2$ such that, for all $u, v \in V_1$, for all $a \in \Sigma$: $(u, a, v) \in E_1 \iff (\beta(u), a, \beta(v)) \in E_2$.

Given a graph $G = \langle V, E_1, E_2, \ldots, E_n \rangle$ and an equivalence relation $\sim \subseteq V \times V$, we denote by $G_{/\sim} = \langle V_{/\sim}, E_{1/\sim}, E_{2/\sim}, \ldots, E_{n/\sim} \rangle$ the *quotient* of G with respect to $\sim$, where:

- $V_{/\sim} = \{[v] \mid v \in V\}$ is the set of $\sim$-equivalence classes in V,
- $E_{i/\sim} = \{\langle U, V \rangle \in V_{/\sim}^2 \mid \text{there exist } u \in V \text{ and } v \in V \text{ such that } u \to_i v\}$.

Chapter 2

Logics and Theories

2.1 First Order Logic

Syntax

The alphabet of FOL consists of the following symbols:

- *predicate symbols*: $p_1, p_2, \ldots, =$
- *function symbols*: $f_1, f_2, \ldots$
- *constant symbols*: $c_1, c_2, \ldots$
- *first-order variables*: $x, y, z, \ldots$
- *connectives*: $\vee, \wedge, \rightarrow, \leftrightarrow, \neg, \perp, \forall, \exists$

Predicate and function symbols are given as ranked alphabets, i.e. we refer to their arities in the following.

The set of *first-order terms* is defined inductively:

- any constant symbol c is a term,
- any first-order variable x is a term,
- if $t_1, t_2, \ldots, t_n$ are terms and f is a function symbol of arity $n > 0$, then $f(t_1, t_2, \ldots, t_n)$ is a term,
- nothing else is a term.

A term with no variable is said to be a *ground term*. An *atomic proposition* is any proposition of the form $p(t_1, \ldots, t_n)$ or $t_1 = t_2$, where $t_1, t_2, \ldots, t_n$ are terms.

Exercise 27 *An inductively defined set S is any set defined using a number of constructors $f_1, f_2, \ldots f_n$, and axioms of either form:*

1. $f \in S$ if $\#(f) = 0$, *or*
2. if $t_1, t_2, \ldots, t_n \in S$ then $f(t_1, t_2, \ldots, t_n) \in S$, where $t_1, t_2, \ldots, t_n$ are terms and $\#(f) = n > 0$.

Show that any inductively defined set is well-founded. Give an example of a well-founded set that is not inductively defined. $\square$

The set of *first-order formulae* is defined inductively:

- $\bot$ and $\top$ are formulae,
- p is a formula, if $\#(p) = 0$,
- if $t_1, t_2, \ldots, t_n$ are terms and p is a predicate symbol of arity $n > 0$, then $p(t_1, t_2, \ldots, t_n)$ is a formula,
- if t_1, t_2 are terms, then $t_1 = t_2$ is a formula,
- if φ and ψ are formulae, then $\varphi \bullet \psi$, $\neg\varphi$, $\forall x \;.\; \varphi$ and $\exists x \;.\; \varphi$ are formulae, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$,
- nothing else is a formula.

The *language* of a logic LOG is the set of formulae in LOG, denoted as $\mathcal{L}(LOG)$.

The *size* of a formula is the number of subformulae it contains, in other words, the number of nodes in the syntax tree representing the formula. The size of φ is denoted as $|\varphi|$.

Example 6 *The following are FOL formulae:*

$$
\begin{gathered}
x = y \\
\forall x \forall y \;.\; x = y \leftrightarrow y = x \\
\exists x (\forall y \;.\; p(x, y)) \rightarrow q(x) \\
\forall x \;.\; p(x) \rightarrow q(f(x)) \\
\forall x \exists y \;.\; f(x) = y \wedge (\forall z \;.\; f(z) = y \rightarrow z = x)
\end{gathered}
$$

$\square$

The variables within the scope of a quantifier are said to be *bound*. The variables that are not bound are said to be *free*. We denote by $FV(\varphi)$ the set of free variables in φ. If $FV(\varphi) = \emptyset$ then φ is said to be a *sentence*.

Example 7 $FV(\forall x \,.\, x = y \wedge x = z \rightarrow p(x)) = \{y, z\}\Box$

If $x \in FV(\varphi)$, we denote by $\varphi[t/x]$ the formula obtained from φ by substituting x with the term t.

Semantics

A *structure* is a tuple $\mathfrak{m} = \langle U, \bar{p_1}, \bar{p_2}, \ldots, \bar{f_1}, \bar{f_2}, \ldots \rangle$, where:

- U is a (possible infinite) set called the *universe*,
- $\bar{p_i} \subseteq U^{\#(p_i)}$, $i = 1, 2, \ldots$ are the *predicates*,
- $\bar{f_i} : U^{\#(f_i)} \rightarrow U$, $i = 1, 2, \ldots$ are the *functions*,

The elements of the universe are called *individuals*, denoted by $\bar{c_1}, \bar{c_2}, \ldots$.

Note: Each individual $\bar{c}$ has a corresponding constant symbol c. However, not all constant symbols from the set $\{c \mid \bar{c} \in U\}$ need to be in the alphabet of the logic.

The *interpretation* of a ground term t in a structure $\mathfrak{m}$ is denoted as $t^{\mathfrak{m}} \in U$:

$$\begin{array}{rcl} c^{\mathfrak{m}} & = & \bar{c} \in U \\ f(t_1, \ldots, t_n)^{\mathfrak{m}} & = & \bar{f}(t_1^{\mathfrak{m}}, \ldots, t_n^{\mathfrak{m}}) \end{array}$$

The *meaning* of a sentence φ in a structure $\mathfrak{m}$ is denoted as $[\![\varphi]\!]_{\mathfrak{m}} \in \{\text{true}, \text{false}\}$:

$$\begin{array}{rcllll} [\![\bot]\!]_{\mathfrak{m}} & = & \text{false} & & & \\ [\![p(t_1, \ldots, t_n)]\!]_{\mathfrak{m}} & = & \text{true} & \text{iff} & \langle t_1^{\mathfrak{m}}, \ldots, t_n^{\mathfrak{m}} \rangle \in \bar{p} & \\ [\![t_1 = t_2]\!]_{\mathfrak{m}} & = & \text{true} & \text{iff} & t_1^{\mathfrak{m}} = t_2^{\mathfrak{m}} & \\ [\![\neg\varphi]\!]_{\mathfrak{m}} & = & \text{true} & \text{iff} & [\![\varphi]\!]_{\mathfrak{m}} = \text{false} & \\ [\![\varphi \wedge \psi]\!]_{\mathfrak{m}} & = & \text{true} & \text{iff} & [\![\varphi]\!]_{\mathfrak{m}} = [\![\psi]\!]_{\mathfrak{m}} = \text{true} & \\ [\![\exists x \,.\, \varphi]\!]_{\mathfrak{m}} & = & \text{true} & \text{iff} & [\![\varphi[c/x]]\!]_{\mathfrak{m}} = \text{true}, & \text{for some individual } \bar{c} \in U \end{array}$$

Derived meanings:

$$\begin{array}{rcl} [\![\top]\!] & = & [\![\neg\bot]\!] \\ [\![\varphi \vee \psi]\!]_{\mathfrak{m}} & = & [\![\neg(\varphi \wedge \psi)]\!]_{\mathfrak{m}} \\ [\![\varphi \rightarrow \psi]\!]_{\mathfrak{m}} & = & [\![\neg\varphi \vee \psi]\!]_{\mathfrak{m}} \\ [\![\varphi \leftrightarrow \psi]\!]_{\mathfrak{m}} & = & [\![(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)]\!]_{\mathfrak{m}} \\ [\![\forall x \,.\, \varphi]\!] & = & [\![\neg\exists x \,.\, \neg\varphi]\!]_{\mathfrak{m}} \end{array}$$

If $[\![\varphi]\!]_{\mathfrak{m}} = \text{true}$ we say that $\mathfrak{m}$ is a *model* of φ, denoted as $\mathfrak{m} \models \varphi$. If $\mathfrak{m} \models \varphi$ for all structures $\mathfrak{m}$, we say that φ is *valid*, denoted as $\models \varphi$. If φ has at least one model, we say that it is *satisfiable*.

SAT Problem*: Given a sentence φ, is φ satisfiable?*

Exercise 28 *Two problems P and Q are equivalent when a method for solving P is also a method for solving Q, and viceversa. Show that satisfiability and validity of first-order sentences are equivalent problems.* $\square$

Exercise 29 *Prove the validity of the following sentences:*

$$\begin{array}{r}
\forall x \forall y \forall z \;.\; x = y \wedge y = z \rightarrow x = z \\
(\exists x \;.\; \varphi \vee \psi) \leftrightarrow ((\exists x \;.\; \varphi) \vee (\exists x \;.\; \psi)) \\
(\forall x \;.\; \varphi \wedge \psi) \leftrightarrow ((\forall x \;.\; \varphi) \wedge (\forall x \;.\; \psi)) \\
(\exists x \;.\; \varphi \wedge \psi) \rightarrow ((\exists x \;.\; \varphi) \wedge (\exists x \;.\; \psi)) \\
\neg(((\exists x \;.\; \varphi) \wedge (\exists x \;.\; \psi)) \rightarrow (\exists x \;.\; \varphi \wedge \psi)) \\
((\forall x \;.\; \varphi) \vee (\forall x \;.\; \psi)) \rightarrow (\forall x \;.\; \varphi \vee \psi) \\
\neg((\forall x \;.\; \varphi \vee \psi) \rightarrow ((\forall x \;.\; \varphi) \vee (\forall x \;.\; \psi)))
\end{array}$$

Example 8 *Let $\leq$ be a binary predicate symbol, and $\mathfrak{m} = \langle U, \bar{\leq} \rangle$ be a structure. $\mathfrak{m}$ is a partially ordered set if $\mathfrak{m} \models \varphi_1 \wedge \varphi_2$, where:*

$$\begin{array}{rcl}
\varphi_1 & : & \forall x \forall y \;.\; x \leq y \wedge y \leq x \leftrightarrow x = y \\
\varphi_2 & : & \forall x \forall y \forall z \;.\; x \leq y \wedge y \leq z \rightarrow x \leq z
\end{array}$$

Notice that $\models \varphi_1 \rightarrow \forall x \;.\; x \leq x$. $\mathfrak{m}$ is a linearly ordered set if $\mathfrak{m} \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3$, where:

$$\varphi_3 \;:\; \forall x \forall y \;.\; x \leq y \vee y \leq x$$

$\square$

2.1.1 Normal Forms

A formula $\varphi \in \mathcal{L}(FOL)$ is said to be *quantifier-free* iff it contains no quantifiers.

A quantifier-free formula $\varphi \in \mathcal{L}(FOL)$ is said to be in *disjunctive normal form* (DNF) iff $\varphi = \bigvee_i \bigwedge_j \lambda_{ij}$, where λ_{ij} are either atomic propositions or negations of atomic propositions.

A quantifier-free formula $\varphi \in \mathcal{L}(FOL)$ is said to be in *conjunctive normal form* (CNF) iff $\varphi = \bigwedge_i \bigvee_j \lambda_{ij}$, where λ_{ij} are either atomic propositions or negations of atomic propositions.

A quantifier-free formula $\varphi \in \mathcal{L}(FOL)$ is said to be in *negation normal form* (NNF) iff the only subformulae appearing under negation are atomic propositions.

A formula $\varphi \in \mathcal{L}(FOL)$ is said to be in *prenex normal form* (PNF) iff

$$\varphi = Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \;.\; \psi(x_1, x_2, \ldots, x_n)$$

where $Q_i \in \{\exists, \forall\}$ and ψ is a quantifier-free formula. Sometimes ψ is said to be the *matrix* of φ.

Exercise 30 *Show that any quantifier-free formula can be written equivalently in either DNF, CNF or NNF. Show that these representations are unique. If n is the size of the formula, give and prove a bound on the sizes of the DNF, CNF and NNF representations.*

2.1.2 FOL on Words

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet and $w : \{0, 1, \ldots, n-1\} \to \Sigma$ be a finite word, e.g. $w = a_0 a_1 \ldots a_{n-1}$. The structure corresponding to w is $\mathfrak{m}_w = \langle dom(w), \{\bar{p_a}\}_{a \in \Sigma}, \bar{\leq} \rangle$, where:

- $dom(w) = \{0, 1, \ldots, n-1\}$,
- $\bar{p_a} = \{x \in dom(w) \mid w(x) = a\}$,
- $x \bar{\leq} y$ iff $x \leq y$.

We denote by Σ^* the set of all finite words.

Example 9 $\mathfrak{m}_{abbaab} = \langle \{0, \ldots, 5\}, \bar{p_a} = \{0, 3, 4\}, \bar{p_b} = \{1, 2, 5\}, \bar{\leq} \rangle$ □

Let $w : \mathbb{N} \to \Sigma$ be an infinite word. The structure corresponding to w is $\mathfrak{m}_w = \langle \mathbb{N}, \{\bar{p_a}\}_{a \in \Sigma}, \bar{\leq} \rangle$. We denote by Σ^ω the set of all infinite words, and by $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

Example 10 $\mathfrak{m}_{(ab)^\omega} = \langle \mathbb{N}, \bar{p_a} = \{2k \mid k \in \mathbb{N}\}, \bar{p_b} = \{2k+1 \mid k \in \mathbb{N}\}, \bar{\leq} \rangle$ □

Exercise 31 *Write a FOL formula $S(x, y)$ which is valid for all positions $x, y \in \mathbb{N}$ such that $y = x + 1$.* □

Exercise 32 *Write a FOL formula $zero(x)$ which is satisfiable only for $x = 0$.* □

Exercise 33 *Write a FOL sentence whose models are all words with a on even positions and b on odd positions. Next, (try to) write a FOL sentence whose models are all words with a on even positions.* □

Exercise 34 *Write a FOL sentence whose models are all finite words.* □

2.1.3 FOL on Trees

Let $\Sigma = \{f, g, \ldots\}$ be an alphabet and $t : \mathbb{N}^* \to \Sigma$ be a finite tree over Σ. The structure corresponding to t is $\mathfrak{m}_t = \langle dom(t), \{\bar{p_f}\}_{f \in \Sigma}, \bar{\leq}, \{s_n\}_{n \in \mathbb{N}} \rangle$ where:

- $\bar{p_f} = \{p \in dom(t) \mid t(p) = f\}$,
- $\bar{\leq}$ is the prefix order on $\mathbb{N}^*$,
- s_n for any $n \in \mathbb{N}$, is the n-th successor function, defined as follows:

$$s_n(p) = \begin{cases} pn & \text{if } pn \in dom(t), \\ p & \text{otherwise} \end{cases}$$

Example 11 $\mathfrak{m}_{f(f(g,g),g)} = \langle\{\epsilon, 0, 1, 00, 01\}, \bar{p_f} = \{\epsilon, 0\}, \bar{p_g} = \{00, 01, 1\}, \bar{\preceq}, \{s_n\}_{n\in\mathbb{N}}\rangle$. □

Example 12 *The* lexicographic *order on* $\mathbb{N}^*$ *is defined as follows:*

$$x \preceq y \ : \ x \leq y \vee \exists z \ . \ s_0(z) \leq x \wedge s_1(z) \leq y$$

□

Let $t : \mathbb{N}^* \to \Sigma$ be an infinite tree over Σ. The structure corresponding to t is $\mathfrak{m}_t = \langle\mathbb{N}^*, \{\bar{p_f}\}_{f\in\Sigma}, \bar{\leq}, \{s_n\}_{n\in\mathbb{N}}\rangle$.

Exercise 35 *A red-black tree is a tree in which all nodes are either red or black, such that the root is black, and each red node has only black children. Write a FOL sentence whose models are all red-black trees.* □

2.2 Monadic Second Order Logic

Syntax

The alphabet of MSOL consists of:

- all first-order symbols
- *set variables*: $X, Y, Z, \ldots$

The set of MSOL terms consists of all first-order terms, set constants and set variables. The set of MSOL formulae consists of:

- all first-order formulae, i.e. $\mathcal{L}(FOL) \subseteq \mathcal{L}(MSOL)$,
- if t is a term and X is a set variable, then $X(t)$ is a formula,
- if φ and ψ are formulae, then $\varphi \bullet \psi$, $\neg\varphi$, $\forall x \ . \ \varphi$, $\exists x \ . \ \varphi$, $\forall X \ . \ \varphi$ and $\exists X \ . \ \varphi$ are formulae, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$.

$p(t)$ and $X(t)$ is sometimes written $t \in p$ with $\#(p) = 1$, and $t \in X$.

Example 13 *The following are MSOL formulae:*

$$\exists X \forall x \ . \ X(x)$$
$$\forall x \ . \ X(x) \rightarrow Y(x)$$
$$\forall Y \ . \ ((\forall x \ . \ Y(x) \rightarrow X(x)) \wedge \exists x \ . \ X(x) \wedge \neg Y(x)) \rightarrow \forall x \ . \ \neg Y(x)$$

Semantics

Let $\mathfrak{m} = \langle U, \bar{p_1}, \bar{p_2}, \ldots, \bar{f_1}, \bar{f_2}, \ldots \rangle$ be a *structure*. The meaning of a sentence φ in a structure $\mathfrak{m}$ is defined by all FOL rules, and in addition by:

$$[\![\exists X \ . \ \varphi]\!]_{\mathfrak{m}} \quad = \quad \text{true} \quad \text{iff} \quad [\![\varphi[p/X]]\!]_{\mathfrak{m}} = \text{true}, \quad \text{for some set } \bar{p} \subseteq U$$

Example 14 *The following MSOL formula characterizes all partitions $\langle X, Y \rangle$ of the set Z:*

$$partition(X, Y, Z) \ : \ (\forall x \forall y \ . \ X(x) \wedge Y(y) \rightarrow \neg x = y) \wedge (\forall x \ . \ Z(x) \leftrightarrow X(x) \vee Y(x))$$

□

2.2.1 MSOL on Words

MSOL interpreted on words (either finite or infinite) is also called the *sequential calculus* or (W)S1S. Intuitivelly, W(eak) indicates that all quantification of set variables $\exists X$ and $\forall X$ is done over finite sets.

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet. The alphabet of the sequential calculus is composed of:

- the infix predicate symbol $\leq$ denoting the linear ordering of positions,
- the set constants $\{p_a \mid a \in \Sigma\}$ where p_a denotes the set of positions of a
- the first and second order variables and connectives.

Example 15

- *The formula $len(x) : \exists z \forall y \ . \ y \leq z \wedge S(z, x)$ defines the length of a finite word and is unsatisfiable on infinite words. S is the FOL-definable successor relation $S(z, x) \iff x = z + 1$ from Exercise 31. Note that word positions are counted starting with* 0.
- *The set of positions of a word is defined by the formula $pos(X) : \forall x \ . \ X(x)$.*
- *The set of even positions is defined by $even(X) \ : \ \exists z \exists Y, Z \ . \ zero(z) \wedge pos(Z) \wedge X(z) \wedge partition(X, Y, Z) \wedge \forall x, y \ . \ X(x) \wedge S(x, y) \rightarrow Y(y) \wedge \forall x, y \ . \ Y(x) \wedge S(x, y) \rightarrow Y(x)$.*
- *The set of all words having a's on even positions is the set of models of the sentence: $\exists X \ . \ even(X) \wedge \forall x \ . \ X(x) \rightarrow p_a(x)$.*

□

2.2.2 MSOL on Trees

MSOL interpreted on trees (either finite or infinitef) is also called (W)S ω S. If the trees are of finite branching $k \in \mathbb{N}$, then we speak of (W)SkS. Intuitivelly, W(eak) indicates that all quantification of set variables $\exists X$ and $\forall X$ is done over finite sets.

Let $\Sigma = \{a, b, \ldots\}$ be a tree alphabet. The alphabet of (W)SωS is composed of:

- the function symbols $\{s_i \mid i \in \mathbb{N}\}$; $s_i(x)$ denotes the i-th successor of x
- the set constants $\{p_a \mid a \in \Sigma\}$; p_a denotes the set of positions of a
- the first and second order variables and connectives.

Example 16 *Let us consider binary trees, i.e. the alphabet of S2S.*

- *The formula* $closed(X) \ : \ \forall x \ . \ X(x) \rightarrow X(s_0(x)) \wedge X(s_1(x))$ *denotes the fact that* X *is a downward-closed set.*
- *The prefix ordering on tree positions is defined by* $x \leq y \ : \ \forall X \ . \ closed(X) \wedge X(x) \rightarrow X(y)$. *The root of a tree is defined by* $root(x) \ : \ \forall y \ . \ x \leq y$.

□

Exercise 36 *Define the set of binary trees* $t \ : \ \{0,1\}^* \rightarrow \{a, b\}$ *such that* $t(p) = a$ *if* p *is of even length and* $t(p) = b$ *if* p *is of odd length.* □

Exercise 37 *Write a S2S formula* $path(X)$ *that defines the set of all paths in a binary tree.* □

Exercise 38 *Write a SωS sentence whose models are all finite trees.* □

2.3 Temporal Logics

Temporal logics are sub-fragments of MSOL that are used in verification of (finite or infinite) systems.

Let $\mathcal{P} = \{a, b, c, \ldots\}$ be a finite alphabet of *atomic propositions*, and $\mathcal{A} = \{\alpha, \beta, \gamma, \ldots\}$ be a finite set of *actions*. A *Kripke structure* is a tuple $K = \langle S, s_0, \rightarrow, L\rangle$ where:

- S is a set of *states*, and $s_0 \in S$ a designated *initial state*,
- $\rightarrow : \ S \times \mathcal{A} \times S$ is a *transition relation*,
- $L : S \rightarrow 2^{\mathcal{P}}$ is a *labeling function.*

In principle, a Kripke structure is a labeled graph. An *infinite path* in K is a sequence of states and actions $\pi\ :\ s_0, \alpha_0, s_1, \alpha_1, s_2, \alpha_2 \ldots$ such that $s_i \xrightarrow{\alpha_i} s_{i+1}$, for all $i \in \mathbb{N}$. Notice that, if S is finite, then every infinite path must visit at least one state infinitely often. When action names are irrelevant, we drop them from the transition relation, i.e. $s \rightarrow t$ iff $s \xrightarrow{\alpha} t$ for some $\alpha \in \mathcal{A}$.

Model Checking Problem: *Given a Kripke structure K and a formula φ, does $K \models \varphi$?*

The *unfolding* of a Kripke structure K is the (possibly infinite) tree structure $\mathcal{T}_K = \langle S_T, s_0, \rightarrow_T, L_T\rangle$ where:

- $S_T = \{s_0, s_1, \ldots s_n \mid$ for all $0 \leq i < n$ there exists a transition $s_i \xrightarrow{\alpha} s_{i+1}$ in $K\}$,
- $s_0, s_1, \ldots s_k \xrightarrow{\alpha} s_0, s_1, \ldots s_k s_{k+1}$ iff $s_k \xrightarrow{\alpha} s_{k+1}$ is K,
- $L_T(s_0, s_1, \ldots s_k) = L(s_k)$.

2.3.1 Linear Temporal Logic

Linear Temporal Logic (LTL) is interpreted over infinite paths in a Kripke structure.

The alphabet of LTL is composed of:

- atomic proposition symbols $a, b, c, \ldots$,
- boolean connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$,
- temporal connectives $\bigcirc, \Box, \Diamond, \mathcal{U}, \mathcal{R}$.

The set of LTL formulae is defined inductively, as follows:

- any atomic proposition is a formula,
- if φ and ψ are formulae, then $\neg\varphi$ and $\varphi \bullet \psi$, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ are also formulae.
- if φ and ψ are formulae, then $\bigcirc\varphi$, $\Box\varphi$, $\Diamond\varphi$, $\varphi\mathcal{U}\psi$ and $\varphi\mathcal{R}\psi$ are formulae,
- nothing else is a formula.

Let π be a path in a Kripke structure $K = \langle S, \rightarrow, L\rangle$. By $\pi(i)$ we denote the i-th element of π and by π_i the path obtained by removing from π the prefix $\pi(0), \pi(1), \ldots \pi(i-1)$.

The semantics of an LTL formula φ is defined on the structure of φ as follows:

$$\begin{array}{ccc}
K, \pi \models a & \iff & a \in L(\pi(0)) \\
K, \pi \models \neg\varphi & \iff & K, \pi \not\models \varphi \\
K, \pi \models \varphi \wedge \psi & \iff & K, \pi \models \varphi \text{ and } K, \pi \models \psi \\
K, \pi \models \bigcirc\varphi & \iff & K, \pi_1 \models \varphi \\
K, \pi \models \varphi\mathcal{U}\psi & \iff & \text{there exists } k \in \mathbb{N} \text{ such that } K, \pi_k \models \psi \text{ and } K, \pi_i \models \varphi \text{ for all } 0 \leq i < k
\end{array}$$

Derived meanings:

$$\begin{array}{ccc}
K, \pi \models \Diamond\varphi & \iff & K, \pi \models \top\mathcal{U}\varphi \\
K, \pi \models \Box\varphi & \iff & K, \pi \models \neg\Diamond\neg\varphi \\
K, \pi \models \varphi\mathcal{R}\psi & \iff & K, \pi \models \neg(\neg\varphi\mathcal{U}\neg\psi)
\end{array}$$

Given a Kripke structure K and a formula $\varphi \in \mathcal{L}(LTL)$, we say that K is a model of φ, denoted $K \models \varphi$ iff $K, \pi \models \varphi$ for all infinite paths π in K.

Example 17

- *a holds throughout the execution of the system (a is invariant):* $\Box a$
- *whenever a holds, b is bound to hold in the future:* $\Box(a \rightarrow \Diamond b)$
- *a holds infinitely often:* $\Box\Diamond a$
- *a holds forever starting from a certain point in the future:* $\Diamond\Box a$

$\Box$

Exercise 39 *Give a semantic definition for $K, \pi \models \varphi\mathcal{R}\psi$.* $\Box$

Exercise 40 *Prove using the semantic rules above, the validity of the following equivalences:*

$$\begin{array}{ccc}
\varphi\mathcal{U}\psi & \iff & \psi \vee (\varphi \wedge \bigcirc(\varphi\mathcal{U}\psi)) \\
\varphi\mathcal{R}\psi & \iff & \psi \wedge (\varphi \vee \bigcirc(\varphi\mathcal{R}\psi))
\end{array}$$

$\Box$

Exercise 41 *Prove that any LTL formula can be written equivalently in negation normal form, using only $\wedge$, $\vee$, $\bigcirc$, $\mathcal{U}$ and $\mathcal{R}$ as connectives.* $\Box$

2.3.2 Expressiveness of FOL on infinite words, LTL and S1S

$$FOL \equiv LTL < S1S$$

Theorem 2 *LTL and FOL on infinite words have the same expressive power.*

Proof: We give here the (easy) inclusion of LTL into FOL. The other direction, from FOL to LTL uses star-free sets and can be found in [1], Theorem 6.8, page 369. We define the translation function $Tr : \mathcal{L}(LTL) \to \mathcal{L}(FOL)$ as follows. For each formula φ in the language of LTL $Tr(\varphi)$ is a formula in the language of FOL, with one free variable, namely t.

$$\begin{array}{rcc} Tr(a) & = & p_a(t) \\ Tr(\neg\varphi) & = & \neg Tr(\varphi) \\ Tr(\varphi \wedge \psi) & = & Tr(\varphi) \wedge Tr(\psi) \\ Tr(\bigcirc\varphi) & = & Tr(\varphi)[t+1/t] \\ Tr(\varphi \mathcal{U} \psi) & = & \exists x \,.\, Tr(\varphi)[x/t] \wedge \forall y \,.\, y < x \to Tr(\psi)[y/t] \end{array}$$

Any infinite path π in a Kripke structure K maps into an infinite word $w_{K,\pi}$. Next, one proves by induction on the structure of $\varphi \in \mathcal{L}$ that, for any infinite path π we have $K, \pi \models \varphi \iff w_{K,\pi} \models Tr(\varphi)[0/t]$. This step is left as an exercise. □

Exercise 42 *Write a S1S formula whose models are exactly all infinite words starting with an even number of* 0*'s followed by an infinite number of* 1*'s.* □

Definition 10 *A language* $L \subseteq \Sigma^\omega$ *is said to be* non-counting *iff:*

$$\exists n_0 \forall n \geq n_0 \forall u, v \in \Sigma^* \forall \beta \in \Sigma^\omega \,.\, uv^n\beta \in L \iff uv^{n+1}\beta \in L$$

Conversely, a language $L \subseteq \Sigma^\omega$ *is said to be* counting *iff:*

$$\forall n_0 \exists n \geq n_0 \exists u, v \in \Sigma^* \exists \beta \in \Sigma^\omega \,.\, (uv^n\beta \notin L \wedge uv^{n+1}\beta \in L) \vee (uv^n\beta \in L \wedge uv^{n+1}\beta \notin L)$$

Exercise 43 *Show that the language* $(00)^*1^\omega$ *(the language of all infinite words starting with an even number of* 0*'s followed by an infinite number of* 1*'s) is counting.* □

Proposition 1 *Each LTL-definable* ω*-language is non-counting.*

Proof: Let φ be an LTL formula and $uv^n\beta$ be an infinite word. We show that $uv^n\beta \models \varphi \iff uv^{n+1}\beta \models \varphi$. The proof is by induction on the structure of φ:

- $\varphi = a$: $uv^n\beta \models a$ depends on the first letter of u, which is the same for $uv^{n+1}\beta$.
- $\varphi = \neg\psi$: $uv^n\beta \not\models \psi \iff uv^{n+1}\beta \not\models \psi$ (by induction hypothesis)

- $\varphi = \psi_1 \wedge \psi_2$: by induction hypothesis ψ_1, ψ_2 define non-counting languages L_1, L_2 with constants n_1, n_2. Take $n_0 = \max(n_1, n_2)$. The remainder of this step is left to the reader.

- $\varphi = \psi_1 \mathcal{U} \psi_2$: by induction hypothesis ψ_1, ψ_2 define non-counting languages L_1, L_2 with constants n_1, n_2. Take $n_0 = 2\max(n_1, n_2)$. We need to show:

  $$\exists j \,.\, (uv^n\beta)_j \models \psi_2 \wedge \forall i < j \,.\, (uv^n\beta)_i \models \psi_1 \iff \exists j \,.\, (uv^{n+1}\beta)_j \models \psi_2 \wedge \forall i < j \,.\, (uv^{n+1}\beta)_i \models \psi_1$$

  "$\Rightarrow$" Let j be the position chosen in the left-hand side of the implication.

  - if $(uv^n\beta)_j$ contains $\geq \max(n_1, n_2)$ v-segments, then by induction hypothesis we have $(uv^{n+1}\beta)_j \models \psi_2 \wedge \forall i < j \,.\, (uv^{n+1}\beta)_i \models \psi_1$.
  - else, if $(uv^n\beta)_j$ contains $< \max(n_1, n_2)$ v-segments, since $n_0 = 2\max(n_1, n_2)$, then the prefix up to j contains $\geq \max(n_1, n_2)$ v-segments, therefore we can chose $j' < j$ such that $(uv^n\beta)_{j'}$ contains $\geq \max(n_1, n_2)$ v-segments.

□

Theorem 3 *LTL is strictly less expressive than S1S.*

Proof: Use proposition 1 and exercises 42 and 43.□

2.3.3 Propositional μ-Calculus

The propositional μ-calculus is the most expressive temporal logic used to reason about *computation trees.*

The alphabet of the propositional μ-calculus is composed of:

- atomic proposition symbols $a, b, c, \ldots$, false $\perp$ and true $\top$,
- boolean connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$,
- transition connectives $\langle\alpha\rangle$, $[\alpha]$, where $\alpha \in \mathcal{A}$ is an action name,
- set variables $X, Y, Z, \ldots$,
- least μ and greatest ν fixed point operators.

The set of formulae $\mathcal{L}(\mu)$ is defined inductively, as follows:

- any atomic proposition is a formula,
- any set variable is a formula,

- if φ and ψ are formulae, then $\neg\varphi$ and $\varphi \bullet \psi$, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ are also formulae.
- if φ is a formula, then $\langle\alpha\rangle\varphi$ and $[\alpha]\varphi$ are formulae, for some $\alpha \in \mathcal{A}$,
- if φ is a formula with $X \in FV(\varphi)$, then $\mu X\ .\ \varphi$ and $\nu X\ .\ \varphi$ are formulae, provided that φ is *syntactically monotone* in X, i.e. that X occurs under an even number of negations in φ.

Let $\varphi(\mathbf{X})$ be a μ-calculus formula with $FV(\varphi) = \mathbf{X}$, $K = \langle S, s_0, \rightarrow, L\rangle$ be a Kripke structure, and $\iota : \mathbf{X} \rightarrow 2^S$ be a mapping of the free variables into subsets of S. The semantics of φ in K with interpretation ι, denoted by $[\![\varphi]\!]_{K,\iota}$ is a set of states, defined by induction on the structure of φ, as follows:

$$\begin{array}{rcc}
[\![\bot]\!]_{K,\iota} & = & \emptyset \\
[\![\top]\!]_{K,\iota} & = & S \\
[\![a]\!]_{K,\iota} & = & \{s \mid a \in L(s)\} \\
[\![X]\!]_{K,\iota} & = & \iota(X) \\
[\![\neg\varphi]\!]_{K,\iota} & = & S \setminus [\![\varphi]\!]_{K,\iota} \\
[\![\varphi \wedge \psi]\!]_{K,\iota} & = & [\![\varphi]\!]_{K,\iota} \cap [\![\psi]\!]_{K,\iota} \\
[\![\langle\alpha\rangle\varphi]\!]_{K,\iota} & = & \{s \mid \exists t \in S\ .\ s \xrightarrow{\alpha} t \text{ and } t \in [\![\varphi]\!]_{K,\iota}\} \\
[\![\mu X\ .\ \varphi]\!]_{K,\iota} & = & \bigcup_{n\in\mathbb{N}} F^n_\varphi(\emptyset)
\end{array}$$

where $F^n_\varphi(\emptyset)$ is the n-th composition of the function $F_\varphi : 2^S \rightarrow 2^S$ $F_\varphi(P) = [\![\varphi(X)]\!]_{K,\iota[X\leftarrow P]}$ with itself, starting with $\emptyset$, i.e. $F_\varphi(F_\varphi(...F_\varphi(\emptyset)))$. Indeed, if defined, this is the least fixpoint of F_φ, i.e. the smallest set Q such that $X = F_\varphi(Q)$.

Derived meanings:

$$\begin{array}{rcl}
[\![[\alpha]\varphi]\!]_{K,\iota} & = & [\![\neg\langle\alpha\rangle\neg\varphi]\!]_{K,\iota} \\
[\![\nu X\ .\ \varphi]\!]_{K,\iota} & = & [\![\neg\mu X\ .\ \neg\varphi]\!]_{K,\iota}
\end{array}$$

For a sentence $\varphi \in \mathcal{L}(\mu)$ we define the satisfaction relation $K, s \models \varphi$ iff $s \in [\![\varphi]\!]_{K,\iota_\emptyset}$, where $\iota_\emptyset$ is the assignment mapping all variables into $\emptyset$. More generally, we have $K \models \varphi$ iff $K, s_0 \models \varphi$.

Exercise 44 *Given a formula $\varphi(X) \in \mathcal{L}(\mu)$, and a Kripke structure K, prove that $F_\varphi : 2^S \rightarrow 2^S$, defined as $F_\varphi(Y) = [\![\varphi]\!]_{K,\iota[X\leftarrow Y]}$ has a least and a greatest fixpoint.* $\Box$

Example 18

- *a holds in all α-successors, and there is at least one such successor:* $[\alpha]a \wedge \langle\alpha\rangle\top$
- *a holds in every state of the system:* $\nu X\ .\ a \wedge \bigwedge_{\alpha\in\mathcal{A}} \big(\langle\alpha\rangle\top \rightarrow \langle\alpha\rangle X\big)$

- *there exists a path where b holds and a holds all the way up to the point where b holds:* $\mu X \ . \ (b \vee (a \wedge \bigvee_{\alpha \in \mathcal{A}} \langle \alpha \rangle X))$

- *on every path b holds and a holds all the way up to the point where b holds:* $\mu X \ . \ \Big(b \vee (a \wedge \bigwedge_{\alpha \in \mathcal{A}} (\langle \alpha \rangle \top \rightarrow \langle \alpha \rangle X))\Big)$

- *from each state there is a path where a eventually holds:* $\nu X \ . \ (\mu Y \ . \ a \vee \bigvee_{\alpha \in \mathcal{A}} \langle \alpha \rangle Y) \wedge \bigwedge_{\alpha \in \mathcal{A}} (\langle \alpha \rangle \top \rightarrow \langle \alpha \rangle X)$

□

2.3.4 Expressiveness of the μ-calculus and SkS

$$\mu\text{-calculus} \leq SkS$$

Theorem 4 *SkS is at least as expressive as the propositional μ-calculus.*

Proof: First one needs to prove that, for any formula $\varphi \in \mathcal{L}(\mu)$ and for any Kripke structure K we have $K \models \varphi \iff \mathcal{T}_K \models \varphi$, where $\mathcal{T}_K$ is the unfolding of K. This is an easy induction on the structure of φ. Next, we shall code any $\varphi \in \mathcal{L}(\mu)$ into SkS. Let the successor functions of SkS be $\{S_\alpha \mid \alpha \in \mathcal{A}\} \cup \{S_\beta\}$ and set constants $\{P_a \mid a \in \mathcal{P}\}$. We consider the following SkS-definable relation:

$$y = x\alpha\beta^i \text{ for some } i \ : \ \forall X \ . \ (X(S_\alpha(x)) \wedge \forall z \ . \ X(z) \rightarrow X(S_\beta(z))) \rightarrow X(y)$$

The translation function $Tr : \mathcal{L}(\mu) \rightarrow \mathcal{L}(SkS)$ is defined as follows. For each formula $\varphi \in \mathcal{L}(\mu)$, $Tr(\varphi)$ is a formula with a free variables x and M.

$$\begin{array}{ccc}
Tr(X) & = & X(x) \\
Tr(a) & = & P_a(x) \\
Tr(\varphi \wedge \psi) & = & Tr(\varphi) \wedge Tr(\psi) \\
Tr(\neg\varphi) & = & \neg Tr(\varphi) \\
Tr(\langle\alpha\rangle\varphi) & = & \exists y \ . \ y = x\alpha\beta^i \text{ for some } i \wedge Tr(\varphi) \\
Tr(\mu X \ . \ \varphi) & = & \forall X \ . \ (\forall y \ . \ Tr(\varphi)[y/x] \rightarrow X(y)) \rightarrow X(x)
\end{array}$$

It remains to be shown that $K \models \varphi$ iff $\mathcal{T}_K \models Tr(\varphi)$. This is left as an exercise. □

2.3.5 Computational Tree Logic

The alphabet of CTL is composed of:

- atomic proposition symbols $a, b, c, \ldots$,

- boolean connectives $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$,
- temporal connectives $EX, AX, E\Box, A\Box, E\Diamond, A\Diamond, E\mathcal{U}, A\mathcal{U}, E\mathcal{R}, A\mathcal{R}$.

The set of LTL formulae is defined inductively, as follows:

- any atomic proposition is a formula,
- if φ and ψ are formulae, then $\neg\varphi$ and $\varphi \bullet \psi$, for $\bullet \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ are also formulae.
- if φ and ψ are formulae, then $\bullet X\varphi$, $\bullet\Box\varphi$, $\bullet\Diamond\varphi$, $\bullet[\varphi\mathcal{U}\psi]$ and $\bullet[\varphi\mathcal{R}\psi]$ are formulae, where $\bullet \in \{A, E\}$.
- nothing else is a formula.

Let π be a path in a Kripke structure $K = \langle S, \rightarrow, L\rangle$. A *pointed* Kripke structure is a pair $\langle K, s\rangle$, with $s \in S$. The semantics of CTL is defined w.r.t. pointed Kripke structures.

The semantics of an CTL formula φ is defined on the structure of φ as follows:

$$
\begin{array}{ccc}
K, s \models a & \iff & a \in L(s) \\
K, s \models \neg\varphi & \iff & K, s \not\models \varphi \\
K, s \models \varphi \wedge \psi & \iff & K, s \models \varphi \text{ and } K, \pi \models \psi \\
K, s \models EX\varphi & \iff & \text{exists } s_1 \in S \text{ such that } s \rightarrow s_1 \text{ and } K, s_1 \models \varphi \\
K, s \models E\Box\varphi & \iff & \text{exists a path } \pi \text{ in } K \text{ such that } \pi(0) = s, \text{ and } K, \pi \models \Box\varphi \\
K, s \models E(\varphi\mathcal{U}\psi) & \iff & \text{there exists a path } \pi \text{ in } K \text{ such that } \pi(0) = s, \text{ and } K, \pi \models \varphi\mathcal{U}\psi
\end{array}
$$

Derived meanings:

$$
\begin{array}{ccc}
K, s \models AX\varphi & \iff & K, s \models \neg EX\neg\varphi \\
K, s \models E\Diamond\varphi & \iff & K, s \models E[\top\mathcal{U}\varphi] \\
K, s \models A\Box\varphi & \iff & K, s \models \neg E\Diamond\neg\varphi \\
K, s \models A\Diamond\varphi & \iff & K, s \models \neg EG\neg\varphi \\
K, s \models A[\varphi\mathcal{U}\psi] & \iff & K, s \models \neg E[\neg\varphi\mathcal{U}\neg\psi] \wedge \neg E\Box\neg\psi \\
K, s \models A[\varphi\mathcal{R}\psi] & \iff & K, s \models \neg E[\neg\varphi\mathcal{U}\neg\psi] \\
K, s \models E[\varphi\mathcal{R}\psi] & \iff & K, s \models \neg A[\neg\varphi\mathcal{U}\neg\psi]
\end{array}
$$

CTL $<$ μ-calculus

We show strict embedding of CTL into the propositional μ-calculus, by the following translation function $Tr : \mathcal{L}(CTL) \rightarrow \mathcal{L}(\mu)$:

$$
\begin{array}{rcl}
Tr(a) & = & a \\
\neg\varphi & = & \neg Tr(\varphi) \\
\varphi \wedge \psi & = & Tr(\varphi) \wedge Tr(\psi) \\
Tr(E\Diamond\varphi) & = & \bigvee_{\alpha\in\mathcal{A}} \langle\alpha\rangle Tr(\varphi) \\
Tr(E\Box\varphi) & = & \nu Z \ . \ Tr(\varphi) \wedge \bigvee_{\alpha\in\mathcal{A}} \langle\alpha\rangle Z \\
Tr(E[\varphi\mathcal{U}\psi]) & = & \mu Z \ . \ Tr(\psi) \vee (Tr(\varphi) \wedge \bigvee_{\alpha\in\mathcal{A}} \langle\alpha\rangle Z)
\end{array}
$$

2.4 Integer Arithmetic

The (positive) integer arithmetic (IA) is the first order theory of (natural) integer numbers.

The alphabet of the integer arithmetic consists of:

- function symbols $+, \cdot, S$ (S is the successor function $n \mapsto n+1$)
- constant symbol 0

If we remove $\cdot$ from the alphabet of IA we obtain Presburger Arithmetic (PA). If we remove $+$ from the alphabet of IA we obtain Skolem Arithmetic (SA).

The semantics of IA is defined in the structure $\mathfrak{Z} = \langle \mathbb{Z}, +, \cdot, n \mapsto n+1\rangle$, respectivelly $\mathfrak{N} = \langle \mathbb{N}, +, \cdot, n \mapsto n+1\rangle$ for positive integers.

Example 19 *Suppose that the following are interpreted in $\mathfrak{N}$.*

- *The order relation is defined as $x \leq y \ : \ \exists z \ .x + y = z$*
- *The set of even numbers is defined by $even(x) \ : \ \exists y \ . \ x = y + y$*
- *The divisibility relation is defined as $x|y \ : \ \exists z \ . \ y = xz$*
- *The set of prime numbers is defined by $prime(x) \ : \ \forall yz \ . \ x = yz \rightarrow (y = 1 \vee z = 1)$*
- *The Conjecture of Goldbach: $\forall x \ . \ 2 \leq x \wedge even(x) \rightarrow \exists y, z \ . \ prime(y) \wedge prime(z) \wedge x = y + z$*

□

Exercise 45 *The least common multiple relation over positive integers is defined as* $[x, y] = z \ : \ \forall t \ . \ x|t \wedge y|t \leftrightarrow z|t$. *Define the multiplication relation, i.e. write a formula that defines the set of all triples* (x, y, z) *such that* $x \cdot y = z$, *using only lcm,* $+$ *and* S. □

Exercise 46 *Write a formula* $pos(x)$ *interpreted in* $\mathfrak{Z}$, *that defines the set of all positive integers.* □

2.4.1 Peano Arithmetic

An *axiomatic theory* is a set of formulae in which truth is derived from a (possibly infinite) set of *axioms*, e.g. Euclid's geometry is an axiomatic theory.

Peano Arithmetic is the axiomatic theory with the following set of axioms:

1. $0 \neq S(x)$
2. $S(x) = S(y) \rightarrow x = y$
3. $x + 0 = x$
4. $x + S(y) = S(x + y)$
5. $x \cdot 0 = 0$
6. $x \cdot S(y) = x \cdot y + x$
7. $\varphi(0) \wedge \forall x \ . \ [\varphi(x) \rightarrow \varphi(S(x))] \rightarrow \forall x \ . \ \varphi(x)$

Notice that the last point defines an infinite number of axioms, one for each formula $\varphi \in \mathcal{L}(IA)$.

Exercise 47 *Prove using the axioms of Peano arithmetic that* $\forall x, y, z \ . \ x \cdot (y + z) = x \cdot y + x \cdot z$.
□

2.4.2 Presburger Arithmetic

Presburger Arithmetic is the theory of natural (integer) numbers with addition, i.e. $\langle \mathbb{N}, + \rangle$ (respectivelly $\langle \mathbb{Z}, + \rangle$).

Example 20

- *The property "x is even" is definable in* $\langle \mathbb{N}, + \rangle$, *i.e.* $even(x) \ : \ \exists y \ . \ x = y + y$.
- *The order relation* $\leq$ *is definable in* $\langle \mathbb{N}, + \rangle$, *i.e.* $x \leq y \ : \ \exists z \ . \ x + z = y$.
- *The constant* 0 *is definable in* $\langle \mathbb{N}, + \rangle$, *i.e.* $zero(x) \ : \ \forall y \ . \ x \leq y$.

- *The constant* 1 *is definable in* $\langle \mathbb{N}, + \rangle$, *i.e.* $one(x) \ : \ 0 \leq x \wedge 0 \neq x \wedge \forall y \ . \ 0 \leq y \wedge y \leq x \rightarrow y = 0 \vee y = x$

$\square$

Exercise 48 *Is it possible to define the order relation* $\leq$ *in* $\langle \mathbb{Z}, + \rangle$*?* $\square$

2.4.3 Skolem Arithmetic

Skolem arithmetic is the theory of natural (integer) numbers with multiplication, i.e. $\langle \mathbb{N}, \cdot \rangle$ (respectivelly $\langle \mathbb{Z}, \cdot \rangle$).

Example 21

- *The property "x is a perfect square" is definable in* $\langle \mathbb{N}, \cdot \rangle$, *i.e.* $psq(x) : \exists y \ . \ x = y \cdot y$
- *The divisibility relation* $|$ *is definable in* $\langle \mathbb{N}, \cdot \rangle$, *i.e.* $x|y \ : \ \exists z \ . \ y = x \cdot z$
- *The constant* 1 *is definable in* $\langle \mathbb{N}, \cdot \rangle$, *i.e.* $one(x) \ : \ x \cdot x = x$

$\square$

Chapter 3

Automata on Finite Words

3.1 Definition of Automata on Finite Words

Let $\Sigma = \{a_1, a_2, \dots a_n\}$ be a finite input alphabet.

Definition 11 *A* non-deterministic finite automaton *(NFA) over* Σ *is a tuple* $A = \langle S, I, T, F\rangle$ *where:*

- S *is a finite set of* states*,*
- $I \subseteq S$ *is a set of* initial states*,*
- $T \subseteq S \times \Sigma \times S$ *is a* transition relation*,*
- $F \subseteq S$ *is a set of* final states*.*

We denote $T(s, \alpha) = \{s' \in S \mid (s, \alpha, s') \in T\}$. When T is clear from the context we denote $(s, \alpha, s') \in T$ by $s \xrightarrow{\alpha} s'$.

Definition 12 *An automaton* $A = \langle S, I, T, F\rangle$ *is* deterministic *(DFA) iff* $\|I\| = 1$ *and, for each* $s \in S$ *and for each* $\alpha \in \Sigma$*,* $\|T(s, \alpha)\| \leq 1$*.*

If A is deterministic we write $T(s, \alpha) = s'$ instead of $T(s, \alpha) = \{s'\}$.

Definition 13 *An automaton* $A = \langle S, I, T, F\rangle$ *is* complete *iff for each* $s \in S$ *and for each* $\alpha \in \Sigma$*,* $\|T(s, \alpha)\| \geq 1$*.*

Given a finite word $w \in \Sigma^*$, $w = \alpha_1\alpha_2 \dots \alpha_n$, a *run* of A over w is a finite sequence of states $s_1, s_2, \dots, s_n, s_{n+1}$ such that $s_1 \in I$ and $s_i \xrightarrow{\alpha_i} s_{i+1}$ for all $1 \leq i \leq n$. The existence of a run between s_1 and s_{n+1} is denoted as $s_1 \xrightarrow{w} s_{n+1}$. The run is said to be *accepting* iff $s_{n+1} \in F$. If A has an accepting run over w, then we say that A *accepts* w. The language of A, denoted $\mathcal{L}(A)$ is the set of all words accepted by A. A set of words $S \subseteq \Sigma^*$ is *rational* if there exists an automaton A such that $S = \mathcal{L}(A)$.

Proposition 2 *If A is deterministic, then it has at most one run for each input word.*

Proof: Left as an exercise. □

Proposition 3 *If A is complete, then it has at least one run for each input word.*

Proof: Left as an exercise. □

3.2 Determinism and Closure Properties

Theorem 5 *For every NFA A there exists a DFA A_d such that $\mathcal{L}(A) = \mathcal{L}(A_d)$.*

Proof: Let $A_d = \langle 2^S, \{I\}, T_d, \{G \subseteq S \mid G \cap F \neq \emptyset\}\rangle$, where $(S_1, \alpha, S_2) \in T_d$ iff $S_2 = \{s' \mid \exists s \in S_1 \,.\, (s, \alpha, s') \in T\}$. The proof that A_d is deterministic and that $\mathcal{L}(A) = \mathcal{L}(A_d)$ is left as an exercise. □

Lemma 2 *For every NFA A there exists a complete NFA A_c such that $\mathcal{L}(A) = \mathcal{L}(A_c)$.*

Proof: Let $A_c = \langle S \cup \{\sigma\}, I, T_c, F\rangle$, where $\sigma \notin S$ is a new sink state. The transition relation T_c is defined as: $\forall s \in S \forall \alpha \in \Sigma \,.\, (s, \alpha, \sigma) \in T_c \iff \forall s' \in S \,.\, (s, \alpha, s') \notin T$ and $\forall \alpha \in \Sigma \,.\, (\sigma, \alpha, \sigma) \in T_c$. The proof that $\mathcal{L}(A) = \mathcal{L}(A_c)$ is left as an exercise. □

Theorem 6 *Let $A_1 = \langle S_1, I_1, T_1, F_1\rangle$ and $A_2 = \langle S_2, I_2, T_2, F_2\rangle$ be two NFA. There exists automata $\bar{A}_1$, $A_\cup$ and $A_\cap$ that recognize the languages $\Sigma^* \setminus \mathcal{L}(A_1)$, $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$, and $\mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ respectivelly.*

Proof: Let $A' = \langle S', I', T', F'\rangle$ be the complete deterministic automaton such that $\mathcal{L}(A_1) = \mathcal{L}(A')$, and $\bar{A}_1 = \langle S', I', T', S' \setminus F'\rangle$. The proof that $\mathcal{L}(\bar{A}_1) = \Sigma^* \setminus \mathcal{L}(A_1)$ is left as an exercise.

Let $A_\cup = \langle S_1 \cup S_2, I_1 \cup I_2, T_1 \cup T_2, F_1 \cup F_2\rangle$. The proof that $\mathcal{L}(A_\cup) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ is left as an exercise.

Let $A_\cap = \langle S_1 \times S_2, I_1 \times I_2, T_\cap, F_1 \times F_2\rangle$ where $(\langle s_1, t_1\rangle, \alpha, \langle s_2, t_2\rangle) \in T_\cap$ iff $(s_1, \alpha, s_2) \in T_1$ and $(t_1, \alpha, t_2) \in T_2$. The proof that $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ is left as an exercise. □

Let the input alphabet $\Sigma = \Sigma_1 \times \Sigma_2$. Any word $w \in \Sigma^*$ can be uniquely identified to a pair $\langle w_1, w_2\rangle \in \Sigma_1^* \times \Sigma_2^*$ such that $|w_1| = |w_2| = |w|$. The *projection* operations are $pr_1(L) = \{u \in \Sigma_1^* \mid \langle u, v\rangle \in L, \text{ for some } v \in \Sigma_2^*\}$ and $pr_2(L) = \{v \in \Sigma_2^* \mid \langle u, v\rangle \in L, \text{ for some } u \in \Sigma_1^*\}$.

Theorem 7 *If the language $L \subseteq (\Sigma_1 \times \Sigma_2)^*$ is rational, then so are the projections $pr_i(L)$, for $i = 1, 2$.*

Proof: Left as an exercise.□

As a remark, the operations of union, intersection and complement correspond to the boolean $\vee$, $\wedge$ and $\neg$, whereas the projection corresponds to the first-order existential quantifier $\exists x$.

3.2.1 On the Exponential Blowup of Complementation

The construction of automata recognizing the complement of a regular language uses determinization. Notice that the size (number of states) in the deterministic automaton build from a NFA A is $2^{size(A)}$, where $size(A)$ is the number of states in A. Some of these states might not be accessible from the initial state, so they can be removed. The following shows that the exponential bound is tight.

Theorem 8 *For every $n \in \mathbb{N}$, $n \geq 1$, there exists an automaton A, with $size(A) = n + 1$ such that no deterministic automaton with less than 2^n states recognizes the complement of $\mathcal{L}(A)$.*

Proof: Let $\Sigma = \{a, b\}$ be the alphabet and $L = \{uav \mid u, v \in \Sigma^*, |v| = n - 1\}$. There exists a NFA with exactly $n + 1$ states which recognizes L. The construction of this automaton is left as an exercise. Now we show that no automaton with less that 2^n states recognizes $\Sigma^* \setminus L$. Suppose that there exists $B = \langle S, \{s_0\}, T, F\rangle$, with $\|S\| < 2^n$ that accepts $\Sigma^* \setminus L$. Since the number of words of length n in Σ^* is 2^n and $\|S\| < 2^n$, there exists two distinct words uav_1 and ubv_2 that lead from s_0 to the same state s. Notice that $uav_1u \in L$, hence it is not recognized by B, however $ubv_2u \notin L$, hence it is recognized by B. By the fact that B is deterministic, $s \xrightarrow{u} s'$, for some unique state s', which should be both accepting (for ubv_2u) and rejecting (for uav_1u), which leads to a contradiction. $\square$

3.3 Myhill-Nerode Theorem and Minimal Form

Let $A = \langle S, I, T, F\rangle$ be an automaton over the alphabet Σ^*. We define the relation $\sim_A \subseteq \Sigma^* \times \Sigma^*$ as $u \sim_A v$ iff $\forall s, s' \in S \,.\, s \xrightarrow{u} s' \iff s \xrightarrow{v} s'$.

Exercise 49 *Show that $\sim_A$ is an equivalence relation of finite index.* $\square$

Let $L \subseteq \Sigma^*$ be a language. The words $u, v \in \Sigma^*$ are L-equivalent, denoted as $u \sim_L v$ iff for all $w \in \Sigma^*$, $uw \in L \iff vw \in L$.

Theorem 9 (Myhill-Nerode) *A language $L \subseteq \Sigma^*$ is rational iff $\sim_L$ is of finite index.*

Proof: "$\Rightarrow$" Suppose $L = \mathcal{L}(A)$ for some automaton A. By exercise 49 $\sim_A$ is of finite index. One proves that, for all $u, v \in \Sigma^*$ that $u \sim_A v \Rightarrow u \sim_L v$. This is left as an exercise. As a consequence, the index of $\sim_L$ is less than the index of $\sim_A$, thus finite.

"$\Leftarrow$" Suppose that $\sim_L$ is an equivalence relation of finite index, and let $[u]$ denote the equivalence class of $u \in \Sigma^*$. The automaton we are looking after is $A = \langle S, I, T, F\rangle$, where:

- $S = \{[u] \mid u \in \Sigma^*\}$,
- $I = [\epsilon]$,

- $[u] \xrightarrow{\alpha} [v] \iff u\alpha \sim_L v$,

- $F = \{[u] \mid u \in L\}$.

It remains to be shown that $\mathcal{L}(A) = L$, which is left as an exercise. □

A DFA A is said to be *minimal* if the number of states of every DFA recognizing $\mathcal{L}(A)$ is greater or equal to $size(A)$.

Exercise 50 *Consider the automaton A from the proof of the Myhill-Nerode theorem. Show that it is minimal.* □

Two automata $A_i = \langle S_i, I_i, T_i, F_i \rangle$, $i = 1, 2$ are said to be *isomorphic* iff there exists a bijection $h : S_1 \to S_2$ such that, for all $s, s' \in S_1$ and for all $\alpha \in \Sigma$ we have :

- $s \in I_1 \iff h(s) \in I_2$,

- $(s, \alpha, s') \in T_1 \iff (h(s), \alpha, h(s')) \in T_2$,

- $s \in F_1 \iff h(s) \in F_2$.

Exercise 51 *Any minimal automaton recognizing L is isomorphic to the automaton from the proof of the Myhill-Nerode theorem for L.* □

Exercise 52 *Give example of a rational language L and two non-deterministic automata A_1 and A_2 such that $size(A_1) = size(A_2)$, A_1 and A_2 are not isomorphic, and L cannot be recognized by any automaton with size less that A_1 and A_2.* □

3.4 Pumping Lemma and Decidability

Lemma 3 (Pumping) *Let $A = \langle S, I, T, F \rangle$ be a finite automaton with $size(A) = n$, and $w \in \mathcal{L}(A)$ be a word of length $|w| \geq n$. Then there exists three words $u, v, t \in \Sigma^*$ such that $|v| \geq 1$, $w = uvt$ and, for all $k \geq 0$, we have that $uv^k t \in \mathcal{L}(A)$.*

Proof: Left as an exercise.□

Example 22 *We prove that the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$ is not rational by contradiction. Suppose that there exists an automaton A with $size(A) = N$, such that $L = \mathcal{L}(A)$. Consider the word $a^N b^N \in L = \mathcal{L}(A)$. Then, according to the Pumping Lemma, there exists words u, v, w such that $|v| \geq 1$, $uvw = a^N b^N$ and $uv^k w \in L$ for all $k \geq 1$. We distinguish three cases:*

- *$v = a^m$, for some $m \in \mathbb{N}$. Then $uvvw$ contains more a's than b's and cannot be in L, contradiction.*

- $v = a^m b^p$ *for some* $m, p \in \mathbb{N}$. *Then* $uvvw = a \dots a^m b^p a^m b^p \dots b$ *and cannot be in* L*, contradiction.*

- $v = b^m$*, for some* $m \in \mathbb{N}$ *is symmetric to the first case.*

□

Exercise 53 *Prove that the following languages are not rational:*

- $\{a^n b^n c^n \mid n \in \mathbb{N}\}$,

- $\{ww \mid w \in \{a, b\}^*\}$,

- $\{w \mid w = w^r\}$*, where* w^r *is the word corresponding to the reversal of* w.

□

Exercise 54 *If* $w \in \Sigma^*$ *and* $\alpha \in \Sigma$*, let* $\#_\alpha(w)$ *be the number of occurrences of* α *in* w*, and* $p_\alpha(w)$ *be the position of the first occurrence of* α *in* w*. Prove that the following languages are not rational:*

- $\{w \in \{a, b\}^* \mid \#_a(w) = \#_b(w)\}$,

- $\{w \in \{a, b, c\}^* \mid p_b(w) = \#_a(w)\}$.

□

The following problems can be asked about automata. Given automata A and B:

- **Emptiness** $\mathcal{L}(A) = \emptyset$?

- **Equality** $\mathcal{L}(A) = \mathcal{L}(B)$?

- **Infinity** $\|\mathcal{L}(A)\| < \infty$?

- **Universality** $\mathcal{L}(A) = \Sigma^*$?

Theorem 10 *Let* A *be an automaton with* $size(A) = n$*. If* $\mathcal{L}(A) \neq \emptyset$*, then there exists a word of length less than* n *that is accepted by* A.

Proof: Let u be the shortest word in $\mathcal{L}(A)$. If $|u| < n$ we are done. Otherwise, if $|u| \geq n$, according to the Pumping Lemma there exists $u_1, v, u_2 \in \Sigma^*$ such that $|v| > 1$ and $u_1 v u_2 = u$. However $u_1 u_2 \in \mathcal{L}(A)$ and $|u_1 u_2| < |u_1 v u_2|$, which contradicts with the choice of u being the shortest word in $\mathcal{L}(A)$. □

Theorem 11 *The emptiness, equality, infinity and universality problems are decidable for automata on finite words.*

Proof: Left as an exercise. □

3.5 Automata on Finite Words and WS1S

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet. Any finite word $w \in \Sigma^*$ induces the following *finite* sets $p_a = \{p \mid w(p) = a\}$. The logic WS1S is the monadic second order logic interpreted over finite words, with atomic propositions:

- $x \leq y$: x is less than y,
- $S(x) = y$: y is the successor of x,
- $p_a(x)$: a occurs at position x in w

Notice that $\leq$ and S can be defined one from another.

Let $m \in \mathbb{N}$ be the smallest number such that $k \leq 2^m$. Every symbol $a \in \Sigma$ is associated a representation in $\{0,1\}^m$. W.l.o.g. we can assume that $\Sigma = \{0,1\}^m$. Let $w \in \Sigma^*$ be a word, $X_1, \ldots X_p$ be WS1S set variables, $p \leq m$ and $x_{p+1}, \ldots x_m$ be first-order individual variables. The word w induces an interpretation I_w of variables as follows. For all $1 \leq i \leq |w|$, $1 \leq j \leq m$ we have:

- $i \in I_w(X_j)$ iff the j-th element of w_i is 1, and
- $I_w(x_j) = i$ iff w_i has 1 on the j-th position and 0 otherwise.

If $\varphi(X_1, \ldots, X_p, x_{p+1}, \ldots, x_m)$ is a WS1S formula, we denote $I_w \models \varphi$ simply as $w \models \varphi$.

Example 23 *Let $\Sigma = \{a, b, c, d\}$, encoded as $a = (00)$, $b = (01)$, $c = (10)$ and $d = (11)$. Then the word abbaacdd induces the valuation $X_1 = \{5, 6, 7\}$, $X_2 = \{1, 2, 6, 7\}$.* $\square$

A language $L\Sigma^*$ is said to be *definable* in a logic LOG iff there exists a formula $\varphi \in \mathcal{L}(LOG)$ such that $L = \{w \mid w \models \varphi\}$.

3.5.1 From Automata to Formulae

Let $A = \langle S, I, T, F\rangle$ be an automaton with p states ($S = \{s_1, \ldots, s_p\}$) over the alphabet $\Sigma = \{0,1\}^m$. We build a WS1S formula $\Phi_A(X_1, \ldots, X_m)$ such that for any $w \in \Sigma^*$, we have $w \in \mathcal{L}(A) \iff w \models \Phi_A$. Let $\mathbf{X} = \{X_1, \ldots, X_m\}$.

Coding of Σ Let $a \in \{0,1\}^m$. We define the formula $\Phi_a(x, \mathbf{X})$ as the conjunction of:

- $X_i(x)$ if the $a_i = 1$, and
- $\neg X_i(x)$ otherwise.

It follows, that for any $w \in \Sigma^*$ we have $w \models \forall x \;.\; \bigvee_{a \in \Sigma} \Psi_a(x, \mathbf{X})$.

Coding of S Let $\mathbf{Y} = \{Y_1, \ldots, Y_p\}$ be set variables. Intuitivelly, the set variable Y_i, $1 \leq i \leq p$ contains all positions labeled by A with state s_i during the run on some word w. The following formula says that each position is labeled with exactly one state during some run:

$$\Phi_S(\mathbf{Y}) \ : \ \forall z \ . \ \bigvee_{1 \leq i \leq p} Y_i(z) \ \wedge \ \bigwedge_{1 \leq i < j \leq p} \neg \exists z \ . \ Y_i(z) \wedge Y_j(z)$$

Coding of I Every run starts from an initial state:

$$\Phi_I(\mathbf{Y}) \ : \ \exists x \forall y \ . \ x \leq y \wedge \bigvee_{s_i \in I} Y_i(x)$$

Coding of T If A is at position x in w, applying a transition makes A label position $S(x)$:

$$\Phi_T(\mathbf{X}, \mathbf{Y}) \ : \ \bigwedge_{i=1}^{p} \forall x \ . \ x \neq S(x) \wedge Y_i(x) \wedge \Phi_a(x, \mathbf{X}) \rightarrow \bigvee_{(s_i, a, s_j) \in T} Y_j(S(x))$$

Coding of F The last state on the run is a final state:

$$\Phi_F(\mathbf{Y}) \ : \ \exists x \forall y \ . \ y \leq x \wedge \bigvee_{s_i \in F} Y_i(x)$$

The formula Φ_A is $\exists Y_1 \ldots \exists Y_p \ . \ \Phi_S \wedge \Phi_I \wedge \Phi_T \wedge \Phi_F$. It is left as an exercise to verify that indeed $w \in \mathcal{L}(A) \iff w \models \Phi_a$. Notice that, since we consider finite words from Σ^*, the quantification is over finite sets only.

3.5.2 From Formulae to Automata

Let $\Phi(X_1, \ldots, X_p, x_{p+1}, \ldots, x_m)$ be a WS1S formula. We build an automaton A_Φ such that $\mathcal{L}(A) = \{w \in \Sigma^* \mid w \models \Phi\}$.

Exercise 55 *Consider the set variables X_1, X_2 and the individual variables x_3, x_4. Build the automata recognizing the following languages:*

- $\{w \in \{0,1\}^4 \mid w \models X_1(x_3)\}$
- $\{w \in \{0,1\}^4 \mid w \models x_3 \leq x_4\}$
- $\{w \in \{0,1\}^4 \mid w \models X_1 = X_2\}$

□

Consider that the formula Φ contains only $\neg, \vee, \wedge, \exists x, \exists X$ and atomic propositions $X(x)$, $x \leq y$ and $X = Y$. Notice that $\forall x \ . \ \varphi$ is $\neg \exists x \ . \ \neg \varphi$. The automaton A_Φ is built by induction on the structure of Φ:

- for $\Phi = \phi_1 \wedge \phi_2$ we have $\mathcal{L}(A_\Phi) = \mathcal{L}(A_{\phi_1}) \cap \mathcal{L}(A_{\phi_2})$
- for $\Phi = \phi_1 \vee \phi_2$ we have $\mathcal{L}(A_\Phi) = \mathcal{L}(A_{\phi_1}) \cup \mathcal{L}(A_{\phi_2})$
- for $\Phi = \neg\phi$ we have $\mathcal{L}(A_\Phi) = \overline{\mathcal{L}(A_\phi)}$
- for $\Phi = \exists X \ . \ \phi$, assume w.l.o.g. that $X = X_1$. Then we have $\mathcal{L}(A_\Phi) = pr_1(\mathcal{L}(A_\phi))$.

Notice that in all cases, A_Φ can be effectively constructed, as a consequence of Theorems 6 and 7.

Theorem 12 *A language $L \subseteq \Sigma^*$ is definable in WS1S iff it is rational.*

Corollary 1 *The SAT problem for WS1S is decidable.*

Proof: As a consequence of Theorems 12 and 11. □

Lemma 4 *Any WS1S formula ϕ is equivalent to an WS1S formula of the form $\exists X_1 \ldots \exists X_m \ . \ \varphi$, where φ does not contain other set variables than $X_1, \ldots, X_m$.*

Proof: Convert ϕ into an automaton A_ϕ and then obtain the formula Φ_{A_ϕ}. Notice that the latter is of the desired form. □

Chapter 4

Automata on Infinite Words

4.1 Definition of Büchi Automata

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet. By Σ^ω we denote the set of all infinite words over Σ.

Example 24 *If $\Sigma = \{a\}$ then $\Sigma^\omega = \{aaa\ldots\}$ has one element.* $\square$

Exercise 56 *Show that Σ^ω is not countable, if $\|\Sigma\| > 1$.* $\square$

Definition 14 *A* non-deterministic Büchi automaton *(NBA) over Σ is a tuple $A = \langle S, I, T, F\rangle$, where:*

- *S is a finite set of* states*,*
- *$I \subseteq S$ is a set of* initial states*,*
- *$T \subseteq S \times \Sigma \times S$ is a* transition relation*,*
- *$F \subseteq S$ is a set of* final states*.*

Notice that a NBA has the same definition as a NFA. The definition of *deterministic* Büchi automaton (DBA) and *complete* Büchi automata are the same as for DFA. A *run* of a Büchi automaton is defined over an infinite word $w : \alpha_1\alpha_2\ldots$ as an infinite sequence of states $\pi : s_0s_1s_2\ldots$ such that $s_0 \in I$ and $(s_i, \alpha_{i+1}, s_{i+1}) \in T$, for all $i \in \mathbb{N}$. Let $\inf(\pi) = \{s \mid s \text{ appears infinitely often on } \pi\}$.

Definition 15 *Given a Büchi automaton $A = \langle S, I, T, F\rangle$, a run π of A is said to be* accepting *iff $\inf(\pi) \cap F \neq \emptyset$.*

An infinite word w is *accepted* by a Büchi automaton A iff A has an accepting run on w. The language of A, denoted $\mathcal{L}(A) \subseteq \Sigma^\omega$ is the set of all strings accepted by A. A language $L \subseteq \Sigma^\omega$ is said to be *Büchi recognizable* iff there exists a Büchi automaton A such that $L = \mathcal{L}(A)$. These languages are also called ω-regular.

Exercise 57 *Let $\Sigma = \{0, 1\}$. Construct Büchi automata recognizing the languages below:*

- $\{w \mid 0$ *occurs in* w *exactly once*$\}$
- $\{w \mid$ *each* 0 *is followed by* $1\}$
- $\{w \mid w$ *contains finitely many* 1*'s*$\}$
- $\{w \mid w$ *contains infinitely many* 1*'s*$\}$
- $\{(01)^n w \mid n \in \mathbb{N},\ w \in \Sigma^\omega\}$
- $\{w \mid 0$ *occurs on all even positions of* $w\}$

□

4.2 Closure under Union, Intersection and Projection

Proposition 4 *For any Büchi automaton A there exists a complete Büchi automaton A_c such that $\mathcal{L}(A) = \mathcal{L}(A_c)$.*

Proof: Left to the reader.□

Theorem 13 *Let $A_1 = \langle S_1, I_1, T_1, F_1 \rangle$ and $A_2 = \langle S_2, I_2, T_2, F_2 \rangle$ be Büchi automata. Then there exists Büchi automata $A_\cup$ and $A_\cap$ such that $\mathcal{L}(A_\cup) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ and $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.*

Proof: The construction of $A_\cup$ is the same as in the case of finite automata. The construction of $A_\cap = \langle S, I, T, F \rangle$ is given below:

- $S = S_1 \times S_2 \times \{1, 2, 3\}$,
- $I = I_1 \times I_2 \times \{1\}$,
- the definition of T is the following:
  - $((s_1, s_1', 1), a, (s_2, s_2', 1)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1' \notin F_1$
  - $((s_1, s_1', 2), a, (s_2, s_2', 2)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_2' \notin F_2$
  - $((s_1, s_1', 1), a, (s_2, s_2', 2)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_1' \in F_1$
  - $((s_1, s_1', 2), a, (s_2, s_2', 3)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$ and $s_2' \in F_2$
  - $((s_1, s_1', 3), a, (s_2, s_2', 1)) \in T$ iff $(s_i, a, s_i') \in T_i$, $i = 1, 2$
- $F = S_1 \times S_2 \times \{3\}$

The proof that $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ is left as an exercise. □

Exercise 58 *Why the Büchi automaton recognizing the intersection of two Büchi recognizable languages cannot be built using the construction for rational languages?* □

Theorem 14 *If* $\Sigma = \Sigma_1 \times \Sigma_2$ *and* $L \subseteq \Sigma^\omega$ *is Büchi recognizable, then so are the projections* $pr_i(L)$, $i = 1, 2$.

Proof: The construction is the same as for finite automata. □

4.3 The Büchi Characterization Theorems

This section establishes the relation between rational and Büchi recognizable languages.

Lemma 5 *If* $L \subseteq \Sigma^*$ *is a rational language, there exists a DFA* $A = \langle S, \{s_0\}, T, F\rangle$ *such that* s_0 *has no incoming transition, and* $L = \mathcal{L}(A)$.

Proof: Let $A' = \langle S', \{s_0'\}, T', F'\rangle$ be a DFA that recognizes L. Let $pre(s_0')$ be the set of predecessors of s_0', i.e. $\{s \mid s \rightarrow s_0'\}$. If $pre(s_0') = \emptyset$ then we are done. Otherwise, suppose w.l.o.g. that $pre(s_0')$ contains only accessible states. Then every state from $pre(s_0')$ belongs to at least one cycle that contains s_0. The idea is to add new states to S' by unrolling all elementary cycles that contain s_0, once. The new automaton is guaranteed to have no incoming transitions to the initial state. The details are left as an exercise. □

If $W \in \Sigma^*$ is a language of finite words, we denote by W^ω the ω-language $\{w_1 w_2 \ldots \mid w_i \in W, i \in \mathbb{N}\}$.

Theorem 15 *Let* $W, V \subseteq \Sigma^*$ *be rational languages. Then the language* VW^ω *is Büchi recognizable.*

Proof: From the previous lemma we know that W is recognized by an automaton A_W with a single initial state, which has no incoming transition to it. Then use this automaton to build a Büchi automaton A_W^b recognizing W^ω. Next, use the finite automaton A_V recognizing V to build a Büchi automaton that recognizes VW^ω. The details are left as an exercise. □

Exercise 59 *Is it possible that, for some non-rational language* $W \subseteq \Sigma^*$, W^ω *is Büchi recognizable?* □

Let $A = \langle S, I, T, F\rangle$ be a Büchi automaton and $s, s' \in S$ be two states. We define $W_{s,s'} = \{w \in \Sigma^* \mid s \xrightarrow{w} s'\}$. Obviously, the language $W_{s,s'} \subseteq \Sigma^*$ is rational, for any $s, s' \in S$.

Theorem 16 (First Büchi Characterization) *An* ω*-langage* $L \subseteq \Sigma^\omega$ *is Büchi recognizable iff* L *is a finite union of* ω*-languages* VW^ω, *where* $V, W \subseteq \Sigma^*$ *are rational languages.*

Proof: "⇒" Let $A = \langle S, I, T, F \rangle$ be the Büchi automaton recognizing L. We prove $L = \bigcup_{s \in I, s' \in F} W_{s,s'} W^{\omega}_{s',s'}$. "⇐" This direction is a consequence of the previous Theorem and Theorem 13, namely closure of ω-regular languages under union. □

Definition 16 *A rational language $W \subseteq \Sigma^*$ is said to be* closed *iff $W \cdot W \subseteq W$.*

Theorem 17 (Second Büchi Characterization) *An ω-langage $L \subseteq \Sigma^\omega$ is Büchi recognizable iff L is a finite union of ω-languages VW^ω, where $V, W \subseteq \Sigma^*$ are closed languages.*

Proof: "⇒" Let $A = \langle S, I, T, F \rangle$ be the Büchi automaton recognizing L. In the proof of the previous Theorem we have $\mathcal{L}(A) = \bigcup_{s \in I, s' \in F} W_{s,s'} W^{\omega}_{s',s'}$, and it is easy to prove that $W_{s',s'} \cdot W_{s',s'} \subseteq W_{s',s'}$ for all $s' \in F$. □

Exercise 60 *Prove that every non-empty ω-regular language contains an* ultimately periodic *infinite word of the form $uvvvv\ldots$.* □

Exercise 61 *Show that the language $\{w \in \Sigma^\omega \mid \text{every finite word occurs in } w\}$ is not ω-regular.* □

4.4 Complementation of Büchi Automata

Definition 17 *An equivalence relation $R \subseteq \Sigma^* \times \Sigma^*$ is said to be a* left-congruence *iff for all $u, v, w \in \Sigma^*$ we have $u \cong v \Rightarrow wu \cong wv$.*

Definition 18 *An equivalence relation $R \subseteq \Sigma^* \times \Sigma^*$ is said to be a* right-congruence *iff for all $u, v, w \in \Sigma^*$ we have $u \cong v \Rightarrow uw \cong vw$.*

Definition 19 *An equivalence relation $R \subseteq \Sigma^* \times \Sigma^*$ is said to be a* congruence *iff it is both a left- and a right-congruence.*

Exercise 62 *Let $L \in \Sigma^*$ be a rational language. Show that the Myhill-Nerode equivalence $\sim_L$ is a congruence.* □

Let Σ be a finite alphabet and $A = \langle S, I, T, F \rangle$ be a Büchi automaton, for the rest of this section.

For $s, s' \in S$ and $w \in \Sigma^*$, we denote as $s \rightarrow^F_w s'$ the fact that there exists a finite sequence $s = s_1, s_2, \ldots, s_n = s'$ such that $(s_i, w(i), s_{i+1}) \in T$ for all $1 \leq i < n$ and $s_j \in F$ for some $1 \leq j \leq n$. Let $W^F_{s,s'}$ be the set of all words such that $s \rightarrow^F_w s'$.

For any two words $u, v \in \Sigma^*$ we denote $u \cong v$ iff for all $s, s' \in S$ we have:

- $s \xrightarrow{u} s' \iff s \xrightarrow{v} s'$, and

- $s \to_u^F s' \iff s \to_v^F s'$.

Exercise 63 *Show that $\cong$ is an equivalence relation on Σ^*.* $\square$

Lemma 6 *The relation $\cong$ is a congruence of Σ^*.*

Proof: Left as an exercise. $\square$

Let $[w]_\cong$ denote the equivalence class of $w \in \Sigma^*$ w.r.t. $\cong$.

Lemma 7 *For any $w \in \Sigma^*$, $[w]_\cong$ is the intersection of all sets of the form $W_{s,s'}, W_{s,s'}^F, \overline{W_{s,s'}}, \overline{W_{s,s'}^F}$, containing w.*

Proof: Let T_w be the intersection of all sets of the form $W_{s,s'}, W_{s,s'}^F, \overline{W_{s,s'}}, \overline{W_{s,s'}^F}$, containing w. We show that $[w]_\cong = T_w$.

"$\subseteq$" If $u \cong w$ then clearly $u \in T_w$.

"$\supseteq$" Let $u \in T_w$. We distinguish two cases:

- if $s \xrightarrow{w} s'$, then $w \in W_{s,s'}$, hence $u \in W_{s,s'}$, then $s \xrightarrow{u} s'$ as well.
- if $s \not\xrightarrow{w} s'$, the $w \in \overline{W_{s,s'}}$, hence $u \in \overline{W_{s,s'}}$, then $s \not\xrightarrow{u} s'$.

Also,

- if $s \to_w^F s'$, then $w \in W_{s,s'}^F$, hence $u \in W_{s,s'}^F$, then $s \to_u^F s'$ as well.
- if $s \not\to_w^F s'$, then $w \in \overline{W_{s,s'}^F}$, hence $u \in \overline{W_{s,s'}^F}$, then $s \not\to_u^F s'$.

Then $u \cong w$. $\square$
Notice that the lemma above proves also that $\cong$ is of finite index.

Definition 20 *A congruence relation $R \subseteq \Sigma^* \times \Sigma^*$* saturates *an ω-language L iff for all R-equivalence classes V and W, if $VW^\omega \cap L \neq \emptyset$ then $VW^\omega \subseteq L$.*

Lemma 8 *The congruence relation $\cong$ saturates $\mathcal{L}(A)$.*

Proof: Let V and W be $\cong$-equivalence classes, and let $w = v_0v_1v_2 \ldots \in VW^\omega \cap \mathcal{L}(A)$. Hence there exists a successful infinite run $s_0s_1s_2\ldots$ such that $s_i \xrightarrow{v_i} s_{i+1}$, for all $i \geq 0$, and for infinitely many $j > 0$ we have $s_j \to_{v_j}^F s_{j+1}$. Let $w' \in VW^\omega$ be an infinite word. Hence, there exists $v_0' \in V$ and $v_i' \in W$, $i > 0$ such that $w' = v_0'v_1'v_2'\ldots$. Since $s_i \xrightarrow{v_i} s_{i+1}$ and $v_i \cong v_i'$, we have $s_i \xrightarrow{v_i'} s_{i+1}$ and for all $j > 0$ such that $s_j \to_{v_j}^F s_{j+1}$ we have $s_j \to_{v_j'}^F s_{j+1}$. Hence A has the same successful run on w', therefore $w' \in \mathcal{L}(A)$. We have proved that $VW^\omega \subseteq \mathcal{L}(A)$. $\square$

Let $\alpha \in \Sigma^\omega$ be an infinite word for the rest of this section. By $\alpha(n,m)$, we denote the finite subword $\alpha(n)\alpha(n+1)\ldots\alpha(m-1)$, for some $n \leq m$.

Definition 21 *Two positions $k, k' \in \mathbb{N}$ are said to* merge at m, $m > k$ *and* $m > k'$ *iff* $\alpha(k, m) \cong \alpha(k', m)$. *We say that k and k' are $\cong_\alpha$-equivalent, denoted $k \cong_\alpha k'$ iff they merge at m, for some $m > k, k'$.*

Exercise 64 *If k and k' merge at m then they also merge at m', for all $m' \geq m$.* □

Lemma 9 *The relation $\cong_\alpha$ is an equivalence relation of finite index.*

Proof: The fact that $\cong_\alpha$ is an equivalence relation is left to the reader. For the second part, assume that $m_1 < m_2 < \ldots$ is an infinite sequence of numbers that are pairwise $\cong_\alpha$-distinct. Let p the number of equivalence classes of $\cong$. Then the words $\alpha_{m_1,s}, \ldots, \alpha_{m_{p+1},s}$, for some $s > m_{p+1}$ are pairwise $\cong$-distinct, which contradicts with the fact that $\cong$ is of index p. Therefore $\cong_\alpha$ must be of finite index. □

As a consequence of the previous lemma, there exist infinitely many positions $0 < k_0 < k_1 < k_2 < \ldots$ such that $k_i \cong_\alpha k_j$, for all $i, j \in \mathbb{N}$. Since $\cong$ is also an equivalence relation of finite index, then there exists an infinite subsequence $\alpha_{k_0,k_{i_0}}, \alpha_{k_0,k_{i_1}}, \alpha_{k_0,k_{i_2}}, \ldots$ such that all words are $\cong$-equivalent. Now we construct from $i_0, i_1, i_2, \ldots$ an infinite subsequence $p_0, p_1, p_2, \ldots$ such that $p_i \cong_\alpha p_j(p_{i+1})$, for all $0 \leq j \leq i$.

Exercise 65 *Give the construction for the sequence $p_0, p_1, p_2, \ldots$, having the property that $p_i \cong_\alpha p_j(p_{i+1})$, for all $0 \leq j \leq i$.* □

Lemma 10 *Let $V = \{w \mid w \cong \alpha_{0,p_0}\}$ and $W = \{w \mid w \cong \alpha_{p_0,p_1}\}$. Then $\alpha \in VW^\omega$.*

Proof: Obviously, $\alpha_{0,p_0} \in V$. Since $p_0 \cong_\alpha p_i(p_{i+1})$, we have $\alpha_{p_i,p_{i+1}} \cong \alpha_{p_0,p_{i+1}}$. But $\alpha_{p_0,p_1} \cong \alpha_{p_0,p_{i+1}}$, and by transitivity of $\cong$ we obtain $\alpha_{p_i,p_{i+1}} \cong \alpha_{p_0,p_1}$, i.e. $\alpha_{p_i,p_{i+1}} \in W$. Hence $\alpha \in VW^\omega$. □

Theorem 18 *For any Büchi automaton A there exists a Büchi automaton $\overline{A}$ such that $\mathcal{L}(\overline{A}) = \Sigma^\omega \setminus \mathcal{L}(A)$.*

Proof: By Lemma 8 and 10 it is immediate that:

$$\mathcal{L}(A) = \bigcup_{VW^\omega \cap \mathcal{L}(A) \neq \emptyset} VW^\omega$$

where V, W are $\cong$-equivalence classes. Notice that there is a finite number of such sets, since $\cong$ is of finite index. Hence

$$\Sigma^\omega \setminus \mathcal{L}(A) = \bigcup_{VW^\omega \cap \mathcal{L}(A) = \emptyset} VW^\omega$$

By Theorem 16 we have that $\Sigma^\omega \setminus \mathcal{L}(A)$ is Büchi recognizable. □

Exercise 66 *Give and prove an upper bound on the size of $\overline{A}$, given the size of A.* □

In [2] it is shown that $size(\overline{A}) \leq 2^{size(A) \log size(A)}$, and this bound is optimal.

4.5 Decidability

This section gives decidability results for the *emptiness*, *equivalence* and *universality* problems for Büchi automata.

Theorem 19 *The emptiness problem for Büchi automata is decidable.*

Proof: Let $A = \langle S, I, T, F \rangle$ be a Büchi automaton. If $\mathcal{L}(A) \neq \emptyset$, then there exists a run $\pi : s_0 s_1 s_2 \ldots$ over some word $w \in \Sigma^\omega$ such that $\inf(\pi) \cap F \neq \emptyset$. Then there exists a state $s \in F$ which is accessible from s_0 and from itself. Viceversa, if there exists such state, A has an accepting run over some word $w \in \Sigma^\omega$. $\square$

Exercise 67 *Write an algorithm that decides whether a given Büchi automaton is empty.* $\square$

Exercise 68 *Write an algorithm that given a Büchi automaton A, constructs a Büchi automaton B, such that $\mathcal{L}(B) = \overline{\mathcal{L}(A)}$.* $\square$

Theorem 20 *The equivalence and universality problems for Büchi automata are decidable.*

Proof: Left as an exercise. $\square$

4.6 Büchi Automata and S1S

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet. Any finite word $w \in \Sigma^*$ induces the following *possibly infinite* sets $P_a = \{p \mid w(p) = a\}$. The logic WS1S is the monadic second order logic interpreted over finite words, with atomic propositions:

- $x \leq y$: x is less than y,
- $S(x) = y$: y is the successor of x,
- $P_a(x)$: a occurs at position x in w

Notice that $\leq$ and S can be defined one from another.

W.l.o.g. we assume $\Sigma = \{0,1\}^m$ for some suitable $m \in \mathbb{N}$. Let $w \in \Sigma^\omega$ be a word, $X_1, \ldots X_p$ be WS1S set variables, $p \leq m$ and $x_{p+1}, \ldots x_m$ be first-order individual variables. The word w induces an interpretation I_w of variables as follows. For all $1 \leq i \leq |w|$, $1 \leq j \leq m$ we have:

- $i \in I_w(X_j)$ iff the j-th element of w_i is 1, and
- $I_w(x_j) = i$ iff w_i has 1 on the j-th position and 0 otherwise.

If $\varphi(X_1, \ldots, X_p, x_{p+1}, \ldots, x_m)$ is a WS1S formula, we denote $I_w \models \varphi$ simply as $w \models \varphi$.

Example 25 *Let $\Sigma = \{a, b, c\}$, encoded as $a = (00)$, $b = (01)$ and $c = (10)$. Then the word $ab(acb)^\omega$ induces the valuation $X_1 = \{3k + 4 \mid k \geq 0\}$ and $X_2 = \{3k + 2 \mid k \geq 0\}$.* $\square$

4.6.1 From Automata to Formulae

Let $A = \langle S, I, T, F \rangle$ be a Büchi automaton with exactly p states ($S = \{s_1, \dots, s_p\}$) over the alphabet $\Sigma = \{0,1\}^m$. We build a WS1S formula $\Phi_A(X_1, \dots, X_m)$ such that for any $w \in \Sigma^\omega$, we have $w \in \mathcal{L}(A) \iff w \models \Phi_A$. Let $\mathbf{X} = \{X_1, \dots, X_m\}$.

Let $\mathbf{Y} = \{Y_0, \dots, Y_p\}$ be set variables. Intuitivelly, the set variable Y_i, $1 \le i \le p$ contains all positions labeled by A with state s_i during the run on some word w. For the construction of Φ_A, refer to Section 3.5.1. The only difference is in the definition of the accepting condition, which is:

$$\forall x \exists y \;.\; x \le y \wedge x \ne y \wedge \bigvee_{s_i \in F} Y_i(y)$$

4.6.2 From Formulae to Automata

Let $\Phi(X_1, \dots, X_p, x_{p+1}, \dots, x_m)$ be a S1S formula. We build an automaton A_Φ such that $\mathcal{L}(A) = \{w \in \Sigma^\omega \mid w \models \Phi\}$. The construction is the same as in Section 3.5.2, based on the fact that Büchi automata have the same closure properties as finite automata: union, intersection, complement, projection. Decidability of S1S is implied by this fact and by the decidability of the emptiness problem for Büchi automata.

The construction of A_Φ from Φ has non-elementary complexity. If Φ be a formula of the form $\exists X_1 \forall X_2 \exists X_3 \dots \forall X_n \;.\; \Psi$, then A_Φ requires n complementations, hence $size(A_\Phi)$ is bounded tightly by:

$$2^{\cdots 2^{2^{2^{k \log k} k \log k}} \cdots 2^{k \log k} k \log k}$$

where the height of the tower is n.

Theorem 21 *The satisfiability problem for S1S is decidable.*

Proof: Left as an exercise. □

4.7 From LTL to Büchi Automata

As shown in Section 2.3.2, every LTL formula can be translated in S1S with equivalent meaning. By Section 4.6.2, for every LTL formula φ one can build a Büchi automaton A_φ that accepts exactly the same infinite words that satisfy φ. However, in general this construction is non-elementary. In this section we present a direct construction of Büchi automata from LTL, of single exponential complexity.

Definition 22 *Let $\Sigma = \{a, b, \dots\}$ be a finite alphabet. A* generalized Büchi automaton *(GBA) over Σ is a tuple $A = \langle S, I, T, \mathcal{F} \rangle$, where:*

- *S is a finite set of* states*,*

- $I \subseteq S$ *is a set of* initial states*,*
- $T \subseteq S \times \Sigma \times S$ *is a* transition relation*,*
- $\mathcal{F} \subseteq 2^S$ *is a set of* sets of final states*.*

The run of a GBA over a word is defined in the same way as for NBA. Let $\mathcal{F} = \{F_1, \ldots, F_k\}$. A run π of a GBA is said to be *accepting* iff, for all $1 \leq i \leq k$, we have $\inf(\pi) \cap F_i \neq \emptyset$.

Lemma 11 *Generalized Büchi automata and non-deterministic Büchi automata recognize the same languages.*

Proof: Given a GBA $A = \langle S, I, T, \mathcal{F} \rangle$, where $\mathcal{F} = \{F_1, \ldots, F_k\}$. Let $A' = \langle S', I', T', F' \rangle$ be a NBA built as follows:

- $S' = S \times \{1, \ldots, k\}$,
- $I' = I \times \{1\}$,
- $(\langle s, i \rangle, a, \langle t, j \rangle) \in T'$ iff $(s, t) \in T$ and:
  - $j = i$ if $s \notin F_i$,
  - $j = (i \mod k) + 1$ if $s \in F_i$.
- $F' = F_1 \times \{1\}$.

The proof that $\mathcal{L}(A) = \mathcal{L}(A')$ is left as an exercise. □

Let $\mathcal{P} = \{a, b, \ldots\}$ be a finite set of atomic propositions. An LTL formula $\varphi \in \mathcal{L}(LTL)$ is interpreted over an infinite path from a Kripke structure $K = \langle S, \rightarrow, L \rangle$, i.e. $K, \pi \models \varphi$. The pair K, π is equivalently seen as an infinite word $w \in \left(2^{\mathcal{P}}\right)^{\omega}$, i.e. $w_i = L(\pi_i)$, for all $i \in \mathbb{N}$. From now on, assume that each LTL formula is in negation normal form, using only $\wedge, \vee, \bigcirc, \mathcal{U}$ and $\mathcal{R}$ as connectives (see Exercise 41).

Definition 23 *Let $\varphi \in \mathcal{L}(LTL)$ be a formula. The* closure *of φ is the set $cl(\varphi) \in 2^{\mathcal{L}(LTL)}$ defined by the following rules:*

- $\varphi \in cl(\varphi)$
- $\bigcirc\psi \in cl(\varphi) \Rightarrow \psi \in cl(\varphi)$
- $\psi_1 \bullet \psi_2 \in cl(\varphi) \Rightarrow \psi_1, \psi_2 \in cl(\varphi)$*, for all* $\bullet \in \{\wedge, \vee, \mathcal{U}, \mathcal{R}\}$*.*

Example 26

$$cl(\Diamond\neg a) = cl(\top\mathcal{U}\neg a) = \{\Diamond\neg a, \neg a, \top\}$$

□

Definition 24 *Given a formula $\varphi \in \mathcal{L}(LTL)$ and an infinite word $w \in \Sigma^\omega$, a* closure labeling[1] *is an infinite word $\tau_{\varphi,w} \in (cl(\varphi))^\omega$ that satisfies the following conditions, for every $i \geq 1$:*

1. $\perp \notin \tau_{\varphi,w}(i)$,
2. $[a \in \tau_{\varphi,w}(i) \Rightarrow a \in w(i)]$ *and* $[\neg a \in \tau_{\varphi,w}(i) \Rightarrow a \notin w(i)]$,
3. $\psi_1 \wedge \psi_2 \in \tau_{\varphi,w}(i) \Rightarrow \psi_1 \in \tau_{\varphi,w}(i)$ *and* $\psi_2 \in \tau_{\varphi,w}(i)$,
4. $\psi_1 \vee \psi_2 \in \tau_{\varphi,w}(i) \Rightarrow \psi_1 \in \tau_{\varphi,w}(i)$ *or* $\psi_2 \in \tau_{\varphi,w}(i)$,
5. $\bigcirc\psi \in \tau_{\varphi,w}(i) \Rightarrow \psi \in \tau_{\varphi,w}(i+1)$,
6. $\psi_1\mathcal{U}\psi_2 \in \tau_{\varphi,w}(i) \Rightarrow \psi_2 \in \tau_{\varphi,w}(i)$ *or* $[\psi_1 \in \tau_{\varphi,w}(i)$ *and* $\psi_1\mathcal{U}\psi_2 \in \tau_{\varphi,w}(i+1)]$,
7. $\psi_1\mathcal{R}\psi_2 \in \tau_{\varphi,w}(i) \Rightarrow \psi_2 \in \tau_{\varphi,w}(i)$ *and* $[\psi_1 \in \tau_{\varphi,w}(i)$ *or* $\psi_1\mathcal{R}\psi_2 \in \tau_{\varphi,w}(i+1)]$,
8. *if* $\psi_1\mathcal{U}\psi_2 \in \tau_{\varphi,w}(i)$ *then there exists* $j \geq i$ *such that* $\psi_2 \in \tau_{\varphi,w}(j)$.

The intuition is that, for all $i \in \mathbb{N}$ we have $\psi \in \tau_{\varphi,w}(i) \Rightarrow w_i \models \psi$. Notice that rule (6) alone does not meet this condition for $\psi = \psi_1\mathcal{U}\psi_2$, because there is no guarantee that $w_j \models \psi_2$ for some $j \in \mathbb{N}$. This is the role of rule (8). Notice that a closure labeling for a given φ and w may not be unique.

Theorem 22 *Let $\mathcal{P}$ be a set of atomic propositions, $\varphi \in \mathcal{L}(LTL)$ a formula built using atomic propositions from $\mathcal{P}$, and $w \in (2^{\mathcal{P}})^\omega$ be an infinite word. Then $w \models \varphi$ iff there exists a closure labeling $\tau \in (cl(\varphi))^\omega$ such that $\varphi \in \tau_0$.*

Proof: "$\Rightarrow$" Suppose $w \models \varphi$ and define the closure labeling τ such that, for all $i \in \mathbb{N}$ and $\psi \in cl(\varphi)$ we have $\psi \in \tau(i) \iff w_i \models \psi$. By definition of $cl(\varphi)$ we have $\varphi \in cl(\varphi)$. Since $w_0 \models \varphi$ we have $\varphi \in \tau_0$. The fact that τ satisfies Definition 24 is left as an exercise.

"$\Leftarrow$" Suppose there exists a closure labeling $\tau \in (cl(\varphi))^\omega$ such that $\varphi \in \tau_0$. The proof is by induction on the structure of φ. The most interesting case is $\varphi = \psi_1\mathcal{U}\psi_2$. By rule (8) there exists $j \in \mathbb{N}$ such that $\psi_2 \in \tau(j)$. Let j be the smallest such number. By the induction hypothesis $w_j \models \psi_2$. Since $\psi_2 \notin \tau(i)$, by rule (6) we have $\psi_1 \in \tau(i)$, for all $0 \leq i < j$. By the induction hypothesis, $w_i \models \psi_1$. By the semantics of LTL, $w \models \psi_1\mathcal{U}\psi_2$. $\square$

Given a formula $\varphi \in \mathcal{L}(LTL)$, we build a GBA recognizing exactly the infinite words $w \in (2^{\mathcal{P}})^\omega$ such that $w \models \varphi$. Let $A_\varphi = \langle S, I, T, \mathcal{F} \rangle$, where:

- $S = 2^{cl(\varphi)}$,
- $I = \{s \in S \mid \varphi \in s\}$,

[1]Sometimes called *Hintikka sequence*.

- $(s, \alpha, t) \in T$ iff:
  - for all $a \in \mathcal{P}$, $a \in s \Rightarrow a \in \alpha$, and $\neg a \in s \Rightarrow a \notin \alpha$,
  - $\bigcirc\psi \in s \Rightarrow \psi \in t$,
  - $\psi_1 \mathcal{U} \psi_2 \in s \Rightarrow \psi_2 \in s$ or $[\psi_1 \in s$ and $\psi_1 \mathcal{U} \psi_2 \in t]$
  - $\psi_1 \mathcal{R} \psi_2 \in s \Rightarrow \psi_2 \in s$ and $[\psi_1 \in s$ or $\psi_1 \mathcal{U} \psi_2 \in t]$
- let $\phi_1 \mathcal{U} \psi_1, \ldots \phi_n \mathcal{U} \psi_n$ be the "until" subformulae of φ. $\mathcal{F} = \{F_1, \ldots, F_n\}$, where:

  $$F_i = \{s \in S \mid \phi_i \mathcal{U} \psi_i \in s \text{ and } \psi_i \in s\} \cup \{s \in S \mid \phi_i \notin s\}$$

  for all $1 \leq i \leq n$.

The acceptance condition is to prevent the situation in which, for some subformula $\phi \mathcal{U} \psi$ of φ and some word $w \in \mathcal{L}(A_\varphi)$, ϕ holds on every position of w from some point on, without ψ every occurring. The proof of $\mathcal{L}(A_\varphi) = \{w \in (2^{\mathcal{P}})^\omega \mid w \models \varphi\}$ is left as an exercise.

Note that $size(A) = 2^{|\varphi|}$. The automata construction for LTL provides a more efficient solution to the satisfiability problem for LTL, than the translation of LTL into S1S and the (non-elementary) automata construction for S1S.

4.7.1 Automata Theoretic LTL Model Checking

An important application of the automata construction for LTL is the model checking problem. Given a Kripke structure $K = \langle S, s_0, \rightarrow, L\rangle$ over a set of atomic propositions $\mathcal{P}$ and a set of actions $\mathcal{A}$ and an LTL formula φ, we ask whether $K \models \varphi$. Remember that $K \models \varphi$ iff $K, \pi \models \varphi$, for all infinite paths π in K. W.l.o.g. we assume that any path in K can be extended to an infinite one, by assuming that the transition relation of K is total.

Let $A_{\neg\varphi} = \langle S_A, I_A, T_A, F_A\rangle$ be the automaton built for the $\neg\varphi$ formula. First, we compute the product automaton $K \otimes A_{\neg\varphi}$, defined as follows. Let $B = K \otimes A_{\neg\varphi} = \langle S_B, I_B, T_B, F_B\rangle$, where:

- $S_B = S_A \times S$,
- $I_B = I_A \times \{s_0\}$,
- $T_B \subseteq S_B \times 2^{\mathcal{P}} \times S_B$ and we have $(\langle s, t\rangle, P, \langle s', t'\rangle) \in T_B$ iff $s \xrightarrow{\alpha} s'$ for some $\alpha \in \mathcal{A}$, and either:
  - $t \neq t'$ and $(t, P, t') \in T_A$ for some $P \subseteq L(s)$, or
  - $t = t'$ and for no transition rule $(t, P, t'') \in T_A$, we have $P \subseteq L(s)$.
- $F_B = F_B \times S$

Notice the definition of the transition relation, in which K can always make a move, whereas $A_{\neg\varphi}$ may perform a transition only if all the atomic propositions labeling the transition are true in the labeling of source state of K. One can check that $\mathcal{L}(B)$ consists of all infinite paths in K that satisfy $\neg\varphi$. In other words, $\mathcal{L}(B) = \emptyset \iff K \models \varphi$.

4.8 Nondeterminism and the McNaughton Theorem

4.8.1 Deterministic Büchi Automata $\subset$ Büchi Automata

Let $W \subseteq \Sigma^*$ be a language of finite words. Define the ω-language $\overrightarrow{W} = \{\alpha \in \Sigma^\omega \mid \alpha(0, n) \in W$ for infinitely many $n\}$

Theorem 23 *A language $L \subseteq \Sigma^\omega$ is recognizable by a deterministic Büchi automaton iff there exists a rational language $W \subseteq \Sigma^*$ such that $L = \overrightarrow{W}$.*

Proof: If $L = \mathcal{L}(A)$ then $W = \mathcal{L}(A')$ where A' is the DFA with the same definition as A, and with the finite acceptance condition. $\square$

Theorem 24 *There exists a Büchi recognizable language that can be recognized by no deterministic Büchi automaton.*

Proof: Let $\Sigma = \{a, b\}$ and $L = \{\alpha \in \Sigma^\omega \mid \#_a(\alpha) < \infty\} = \Sigma^* b^\omega$ be the ω-regular language of words with a finite number of a's. Suppose $L = \overrightarrow{W}$, for some rational language $W \subseteq \Sigma^*$. Since $b^\omega \in L$ we have $b^{n_1} \in W$ for some $n_1 \in \mathbb{N}$. Since $b^{n_1} a b^\omega \in L$ we have $b^{n_1} a b^{n_2} \in W$ for some $n_2 \in \mathbb{N}$. Continuing in this way we obtain an infinite number of finite words containing $0, 1, 2 \ldots$ a's. Hence an infinite word with an infinite number of a's is contained in $\overrightarrow{W} = L$, contradiction. $\square$

We can also show that Büchi automata are not closed under complement.

Theorem 25 *There exists a DBA A such that no DBA recognizes the language $\Sigma^\omega \setminus \mathcal{L}(A)$.*

Proof: Let $\Sigma = \{a, b\}$ and $L = \{\alpha \in \Sigma^\omega \mid \#_a(\alpha) < \infty\} = \Sigma^* b^\omega$ be the ω-regular language of words with a finite number of a's. Let $V = \Sigma^* a$. There exists a DFA A such that $\mathcal{L}(A) = V$, hence there exists a deterministic Büchi automaton B such that $\mathcal{L}(A) = \overrightarrow{V}$. But $\Sigma^\omega \setminus \overrightarrow{V} = L$ which cannot be recognized by any DBA, according to the previous theorem. $\square$

4.8.2 Müller and Rabin Automata on Infinite Words

Let $\Sigma = \{a, b, \ldots\}$ be a finite alphabet.

Definition 25 (Müller Automaton) *A* Müller automaton *over Σ is $A = \langle S, s_0, T, \mathcal{F} \rangle$, where:*

- S *is the finite* set of states
- $s_0 \in S$ *is the* initial state
- $T : S \times \Sigma \to S$ *is the* transition function
- $\mathcal{F} \subseteq 2^S$ *is the set of* accepting sets

A *run* of a Müller automaton is defined over an infinite word $w : \alpha_1\alpha_2\ldots$ as an infinite sequence of states $\pi : s_0s_1s_2\ldots$ such that $T(s_i, \alpha_{i+1}) = s_{i+1}$, for all $i \in \mathbb{N}$. A run π of A is said to be *accepting* iff $\inf(\pi) \in \mathcal{F}$. A language $L \subseteq \Sigma^\omega$ is *Müller-recognizable* iff there exists a MA A such that $L = \mathcal{L}(A)$.

Theorem 26 *For each deterministic Büchi automaton A there exists a Müller automaton B such that $\mathcal{L}(A) = \mathcal{L}(B)$*

Proof: Let $A = \langle S, \{s_0\}, T, F\rangle$ be a deterministic Büchi automaton. Define the Müller automaton $B = \langle S, s_0, T, \{G \in 2^S \mid G \cap F \neq \emptyset\}\rangle$. The proof that $\mathcal{L}(A) = \mathcal{L}(B)$ is left as an exercise. $\Box$

Theorem 27 *The class of Müller-recognizable languages is closed under union, intersection and complement.*

Proof: Let $A = \langle S, s_0, T, \mathcal{F}\rangle$ be a Müller automaton. The automaton recognizing the complement of $\mathcal{L}(A)$ is defined as $B = \langle S, s_0, T, 2^S \setminus \mathcal{F}\rangle$. The proof that $\mathcal{L}(B) = \Sigma^\omega \setminus \mathcal{L}(A)$.

Let $A_i = \langle S_i, s_{0,i}, T_i, \mathcal{F}_i\rangle$, $i = 1, 2$ be Müller automata. Define $B = \langle S, s_0, T, \mathcal{F}\rangle$ where:

- $S = S_1 \times S_2$,
- $s_0 = \langle s_{0,1}, s_{0,2}\rangle$,
- $T(\langle s_1, s_2\rangle, a) = \langle T(s_1, a), T(s_2, a)\rangle$
- $\mathcal{F} = \{\{\langle s_1, s_1'\rangle, \ldots, \langle s_k, s_k'\rangle\} \mid \{s_1, \ldots, s_k\} \in \mathcal{F}_1 \text{ or } \{s_1', \ldots, s_k'\} \in \mathcal{F}_2\}$

The proof that $\mathcal{L}(B) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ is left as an exercise.

In consequence, Müller automata are also closed under intersection. $\Box$

Theorem 28 *A language $L \subseteq \Sigma^\omega$ is Müller-recognizable iff L is a Boolean combination of sets $\overrightarrow{W}$, $W \subseteq \Sigma^*$, i.e. $L = \bigcup_i \left(\bigcap_j \overrightarrow{W_{ij}} \cap \bigcap_k (\Sigma^\omega \setminus \overrightarrow{W_{ik}})\right)$.*

Proof: "$\Leftarrow$" Any set $\overrightarrow{W_{ij}}$ is recognized by a deterministic Büchi automaton, hence also by a Müller automaton.

"$\Rightarrow$" Let $A = \langle S, s_0, T, \mathcal{F}\rangle$ be a Müller automaton recognizing L. Let $A_q = \langle S, s_0, T, \{q\}\rangle$, for some $q \in S$, and define $W_q = \mathcal{L}(A_q)$. The proof that $L = \bigcup_{Q\in\mathcal{F}} \left(\bigcap_{q\in Q} \overrightarrow{W_q} \cap \bigcap_{q\in S\setminus Q}(\Sigma^\omega \setminus \overrightarrow{W_q})\right)$ is left as an exercise. □

Definition 26 (Rabin Automaton) *A* Rabin automaton *over* Σ *is* $A = \langle S, s_0, T, \Omega\rangle$, *where:*

- S *is the finite* set of states
- $s_0 \in S$ *is the* initial state
- $T : S \times \Sigma \to S$ *is the* transition function
- $\Omega = \{\langle N_1, P_1\rangle, \ldots, \langle N_k, P_k\rangle\}$ *is the set of* accepting pairs*,* $N_i, P_i \subseteq S$*.*

A run π of A is said to be *accepting* iff $\inf(\pi) \cap N_i = \emptyset$ and $\inf(\pi) \cap P_i \neq \emptyset$, for some $1 \leq i \leq k$.

We show now equivalence between Rabin and Müller automata.

Lemma 12 *Given a Rabin automaton* $A = \langle S, s_0, T, \Omega\rangle$*, there exists a Müller automaton* B *such that* $\mathcal{L}(A) = \mathcal{L}(B)$*.*

Proof: Let $\Omega = \{\langle N_1, P_1\rangle, \ldots, \langle N_k, P_k\rangle\}$ be the set of accepting pairs of A. Let $A_i = \langle S, s_0, T, P_i\rangle$, and $B_i = \langle S, s_0, T, N_i\rangle$ be finite state automata. The proof that $\mathcal{L}(A) = \bigcup_{i=1}^{k} \left(\overrightarrow{\mathcal{L}(A_i)} \cap (\Sigma^\omega \setminus \overrightarrow{\mathcal{L}(B_i)})\right)$ is left to the reader. By the characterization of Müller-recognizable languages, this entails the existence of a Müller automaton B such that $\mathcal{L}(A) = \mathcal{L}(B)$. □

Lemma 13 *Given a Müller automaton* $A = \langle S, s_0, T, \mathcal{F}\rangle$*, there exists a Rabin automaton* B *such that* $\mathcal{L}(A) = \mathcal{L}(B)$*.*

Proof: Let $\mathcal{F} = \{Q_1, \ldots, Q_k\}$ be the set of accepting sets of A. Let $B = \langle S', s'_0, T', \Omega'\rangle$ where:

- $S' = 2^{Q_1} \times \ldots \times 2^{Q_k} \times S$
- $s'_0 = \langle \emptyset, \ldots, \emptyset, s_0\rangle$
- $T'(\langle S_1, \ldots, S_k, s\rangle, a) = \langle S'_1, \ldots, S'_k, s'\rangle$, where, for all $1 \leq i \leq k$:
  - $s' = T(s, a)$

- $S'_i = \emptyset$ if $S_i = Q_i$,
- $S'_i = (S_i \cup \{s'\}) \cap Q_i$, otherwise.

- $P_i = \{\langle S_1, \ldots, S_i, \ldots, S_k, s\rangle \mid S_i = Q_i\}$, $1 \le i \le k$
- $N_i = \{\langle S_1, \ldots, S_i, \ldots, S_k, s\rangle \mid s \notin Q_i\}$, $1 \le i \le k$

The proof that $\mathcal{L}(A) = \mathcal{L}(B)$ is left to the reader. □

As an obvious consequence of the above two Lemmas, Rabin automata are also closed under the boolean operations of union, intersection and complement.

4.8.3 The McNaughton Theorem

The McNaughton Theorem shows equivalence of Müller and Büchi automata. In other words, this means that, since deterministic Büchi automata are strictly less powerful than non-deterministic ones, the ω-regular languages can also be recognized by deterministic automata (Müller and Rabin).

Theorem 29 (McNaughton) *For every Büchi automaton there exists a Müller automaton with the same language.*

For the proof, we start by introducing the notion of *flag points*. Let $\alpha \in \Sigma^\omega$ be an infinite word, and $W \subseteq \Sigma^*$ be a rational language, for the rest of this section.

Definition 27 *Position $i \in \mathbb{N}$ is said to be a* flag point *if there exists a position $0 \le j < i$, called an* associated point *such that:*

1. $\alpha(0, j) \in W$
2. $T(s_0, \alpha(j, i)) = T(s_0, \alpha(0, i))$
3. $\forall j < k < i \;.\; T(s_0, \alpha(j, k)) \neq T(s_0, \alpha(0, k))$

Exercise 69 *Prove that, if i_1 and i_2 are distinct flag points of α, then the associated points j_1 and j_2 are distinct.* □

Define $W^+ = \{\alpha \mid \alpha \text{ has infinitely many flag points}\} \subseteq \Sigma^\omega$.

Lemma 14 $W^+ \subseteq W^\omega$.

Proof: Let $\alpha \in W^+$ be an infinite word. α has infinitely many flag points $0 < i_1 < i_2 < \ldots$. Since the associated points $j_1 < j_2 < \ldots$ are distinct, by Exercise 69, we can choose, by passing to a subsequence, the flag points interleaved with the associated points as: $0 < i_1 <$

$j_1 < i_2 < j_2 < \ldots$. By definition of a flag point, we have $\alpha(0, j_1) \in W$. Now we show that $\alpha(j_k, j_{k+1}) \in W$, for all $k \geq 1$.

Let $A = \langle S, T, I, F\rangle$ be the finite automaton recognizing W. We have $T(s_0, \alpha(j_k, j_{k+1})) = T(s_0, \alpha(j_k, i_k) \cdot \alpha(i_k, j_{k+1})) = T(s, \alpha(i_k, j_{k+1})) = T(s_0, \alpha(0, j_{k+1})) \in F$. $\square$

We recall here the definition of closed languages (Def. 16).

Two positions i and j are said to be α-equivalent iff there exists $k > i, j$ such that $T(s_0, \alpha(i, k)) = T(s_0, \alpha(j, k))$. α-equivalence is denoted as $i =_\alpha j$.

Exercise 70 *$=_\alpha$ is an equivalence relation of finite index.* $\square$

Lemma 15 *If W is a closed language, then $W^\omega = W \cdot W^+$.*

Proof: "$\supseteq$" By the preceding Lemma, $W^+ \subseteq W^\omega \Rightarrow W \cdot W^+ \subseteq W \cdot W^\omega \subseteq W^\omega$.

"$\subseteq$" Let $\alpha \in W^\omega$. Then there exists a sequence $0 < j_1 < j_2 < \ldots$ such that $\alpha(0, j_1) \in W$ and $\alpha(j_k, j_{k+1}) \in W$, for all $k \geq 1$. Since $=_\alpha$ is an equivalence relation of finite index, we can go to a subsequence of the original sequence, an suppose w.l.o.g. that all j_k, $k \geq 1$ are α-equivalent. Since W is closed, we can go to a subsequence without losing the fact that $\alpha(j_k, j_{k+1}) \in W$.

We need to show that $\alpha_{j_1} \in W^+$. Since $j_1 =_\alpha j_2$ we have an $m > j_1, j_2$ such that $T(s_0, \alpha(j_1, m)) = T(s_0, \alpha(j_2, m))$. Let m be the minimal such number. Since $\alpha(j_1, j_2) \in W$, we have that m is a flag point for α_{j_1}, whose associated point is j_2. Let j_s be the smallest number in the sequence that is greater or equal to m. We have $\alpha(j_1, j_s) \in W$, and since $j_1 =_\alpha j_s$ there exists $m_1 > j_1, j_s$ such that $T(s_0, \alpha(j_1, m_1)) = T(s_0, \alpha(j_s, m_1))$. Hence $m_1 > m$ is a flag point for $\alpha(j_1, \omega)$, whose associated point is j_s. Continuing in this way, we show that $\alpha_{j_1} \in W^+$. $\square$

Lemma 16 *The set W^+ is Müller-recognizable.*

Proof: We show that $W^+ = \overrightarrow{V}$ for some rational language $V \in \Sigma^*$. $v \in V$ iff $v = v_1 v_2$ such that:

- $v_1 \in W$,
- $T(s_0, v) = T(s_0, v_2)$,
- for no $v_3 \prec v_2$ we have $T(s_0, v) = T(s_0, v_3)$,

The proof $W^+ = \overrightarrow{V}$ is left as an exercise.

Let $A = \langle S, \{s_0\}, T, F\rangle$ be the DFA recognizing W. We define $A' = \langle S', \{s_0\}, T', F'\rangle$ where:

- $S' = S \cup S \times S \cup \{f\}$
- $T' = T \cup T_1 \cup T_2 \cup T_3$ where:
  - $T_1 = \{(s, a, \langle s_0, s_2 \rangle) \mid T(s, a) \in F,\ (s, a, s_2) \in T\}$
  - $T_2 = \{(\langle s_1, s_2 \rangle, a, \langle s_1', s_2' \rangle) \mid (s_1, a, s_1'), (s_2, a, s_2') \in T\}$
  - $T_3 = \{(\langle s, s \rangle, a, f) \mid s \in S\}$
- $F' = \{\langle s, s \rangle \mid s \in S\}$

The proof that $\mathcal{L}(A') = V$ is left as an exercise. □

If $\sigma\ :\ s_0 s_1 s_2 \ldots s_{k-1}$ is a sequence of states, define the *display* $D(s_0 s_1 s_2 \ldots s_{k-1}) = s_{i_1} s_{i_2} \ldots s_{i_m}$ $:\ s_{i_t} \neq s_{i_{t-1}},\ s_{i_t} \neq s_{i_{t-2}},\ \ldots\ s_{i_t} \neq s_{i_0}$. A sequence σ is called a display iff $D(\sigma) = \sigma$. Let $\mu(\sigma)$ be the largest number n such that $\sigma(0, n)$ is a display.

Lemma 17 *Let $W \subseteq \Sigma^*$ be a rational language and $L \subseteq \Sigma^\omega$ be a Müller-recognizable language. Then the language $W \cdot L$ is also Müller-recognizable.*

Proof: Let $A_1 = \langle S_1, s_{0,1}, T_1, F_1 \rangle$ be a DFA recognizing W, and $A_2 = \langle S_2, s_{0,2}, T_2, \Omega_2 \rangle$ be a Müller automaton recognizing L, where $\Omega_2 = \{\langle N_1, P_1 \rangle, \ldots, \langle N_t, P_t \rangle\}$.

If $\sigma = s_0 s_1 \ldots s_n$ is a sequence of states, we denote by $T(\sigma)$ the sequence $T(s_0)T(s_1) \ldots T(s_n)$. We define a Rabin automaton $A = \langle S, I, T, \Omega \rangle$, where:

- $S = \{\langle s_1, \mathbf{s}, i \rangle \mid s_1 \in S_1, D(\mathbf{s}) = \mathbf{s}, i \leq \|S_2\|\}$
- $s_0 = \langle s_{0,1}, \epsilon, 0 \rangle$
- 1. $T(\langle s_1, \mathbf{s}, i \rangle, a) = \langle T_1(s_1, a), D(T_2(\mathbf{s}, a)), \mu(T_2(\mathbf{s}, a)) \rangle$ if $s_1 \notin F_1$
  2. $T(\langle s_1, \mathbf{s}, i \rangle, a) = \langle T_1(s_1, a), D(T_2(\mathbf{s}s_{0,2}, a)), \mu(T_2(\mathbf{s}s_{0,2}, a)) \rangle$ if $s_1 \notin F_1$
- for every $0 \leq k \leq \|S_2\|$ and $\langle N_i, P_i \rangle$ accepting pair of A_2:
  1. $P_{i,k} = \{\langle s_1, \mathbf{s}, j \rangle \mid s_1 \in S_1 \text{ and } \mathbf{s}(k) \in P_i\}$
  2. $N_{i,k} = \{\langle s_1, \mathbf{s}, j \rangle \mid (s_1 \in S_1 \text{ and } \mathbf{s}(k) \in N_i) \text{ or } j < k\}$

The proof that $\mathcal{L}(A) = W \cdot L$ is left as an exercise. □

Theorem 30 (McNaughton) *For every Büchi automaton there exists a Müller (Rabin) automaton recognizing the same language.*

Proof: Let A be Büchi automaton. By the second characterization Theorem for Büchi-recognizable languages, $\mathcal{L}(A) = \bigcup_{i \in I} V_i \cdot W_i^\omega = \bigcup_{i \in I} V_i \cdot W_i \cdot W_i^+$, by Lemma 15. By Lemma 16 we have that W_i^+ are Müller-recognizable, and by Lemma 17 we have that $V_i \cdot W_i \cdot W_i^+$ are Müller-recognizable. Since Müller-recognizable languages are closed under union, we have the result. □

Chapter 5

Automata on Finite Trees

5.1 Bottom Up Tree Automata

Let $\Sigma = \{f, g, h, \ldots\}$ be a finite *ranked alphabet*. A *bottom-up tree automaton* is a tuple $A = \langle S, T, F \rangle$ where:

- S is a finite set of *states*,
- T is a set of *transition rules* of the form:

  $$f(q_1, \ldots, q_n) \to q$$

  where $f \in \Sigma$, $\#(f) = n$, and $q_1, \ldots q_n, q \in S$.
- $F \subseteq S$ is a set of final states.

Notice that there are no initial states. If $\#(f) = 0$ we have rules of the form $f \to q$. Zero-arity symbols are called *constants*, and denoted by $a, b, c, \ldots$.

A *run* of A over a tree $t : \mathbb{N}^* \to \Sigma$ is a mapping $\pi : dom(t) \to S$ such that, for each position $p \in dom(t)$, where $q = \pi(p)$, we have:

- if $\#(t(p)) = n$ and $q_i = \pi(pi)$, $1 \leq i \leq n$, then T has a rule

  $$t(p)(q_1, \ldots, q_n) \to q$$

A run π is said to be *accepting*, if and only if $\pi(\lambda) \in F$. As usual, the *language* of A, denoted as $\mathcal{L}(A)$ is the set of all trees over which A has an accepting run. A set of trees $L \subseteq \mathcal{T}(\Sigma)$ is said to be a *rational tree language* iff there exists a bottom-up tree automaton A such that $\mathcal{L}(A) = L$.

Example 27 *Let $\Sigma = \{f, g, a\}$, where $\#(f) = 2$, $\#(g) = 1$ and $\#(a) = 0$. Let $A = \langle S, T, F \rangle$, where:*

- $S = \{q_f, q_g, q_a\}$, $F = \{q_f\}$, *and*
- $T = \{a \to q_a, g(q_a) \to q_g, g(q_g) \to q_g, f(q_g, q_g) \to q_f\}$

□

Example 28 *Let* $\Sigma = \{red, black, nil\}$ *with* $\#(red) = \#(black) = 2$ *and* $\#(nil) = 0$. *Let* $A_{rb} = \langle \{q_b, q_r\}, T, \{q_b\} \rangle$ *with*

$$T = \{nil \to q_b, black(q_{b/r}, q_{b/r}) \to q_b, red(q_b, q_b) \to q_r\}$$

By using $q_{x/y}$ *within the left-hand side of a transition rule, we mean the set of rules in which either* q_x *or* q_y *take the place of* $q_{x/y}$. □

5.2 Determinism and Closure Properties

A tree automaton is said to be *deterministic* iff there are no two transition rules with the same left-hand side.

Proposition 5 *A deterministic tree automaton has at most one run for each input tree.*

A tree automaton is said to be *complete* iff there exists at least one transition rule $f(q_1, \ldots, q_n) \to q$, for each $f \in \Sigma$, $\#(f) = n$ and $q_1, \ldots, q_n \in S$.

Proposition 6 *A complete tree automaton has at least one run for each input tree.*

Theorem 31 *Let L be a rational tree language. Then there exists a complete deterministic tree automaton A such that $\mathcal{L}(A) = L$.*

Proof: We give the construction for the deterministic tree automaton. The completion part is left as an exercise. Let $A = \langle S, T, F \rangle$ be a tree automaton. We define $A_d = \langle S_d, T_d, F_d \rangle$ as:

- $S_d = 2^S$,
- $f(s_1, \ldots, s_n) \to s \iff s = \{q \in S \mid \exists q_1 \in s_1, \ldots \exists q_n \in s_n \,.\, [f(q_1, \ldots, q_n) \to q] \in T\}$,
- $F_d = \{s \subseteq S \mid s \cap F \neq \emptyset\}$.

It is left as an exercise to prove that $\mathcal{L}(A_d) = \mathcal{L}(A)$. □
Using a similar argument as in Theorem 8, we can show that the exponential blowup is actually necessary. Intuitively, this is also because tree automata are a generalization of word automata.

Exercise 71 *Let* $\Sigma = \{f, a\}$, *with* $\#(f) = 2$ *and* $\#(a) = 0$. *Describe a tree automaton recognizing the language of "combs" of the form* $f(a, f(a, \ldots f(a, a)))$. □

Exercise 72 *Let L be a rational tree language over $\Sigma = \{f, a\}$, with $\#(f) = 2$ and $\#(a) = 0$. Let $L^c \supseteq L$ be the least superset of L that is closed under the following rule:*

- commutativity*: $C[f(t_1, t_2)] \in L^c \Rightarrow C[f(t_2, t_1)] \in L^c$*

Prove that L^c is rational. □

Theorem 32 *The class of rational tree languages is closed under union, complementation and intersection.*

Proof: **Union** Let $A_i = \langle S_i, T_i, F_i \rangle$ for $i = 1, 2$. Suppose that $S_1 \cap S_2 = \emptyset$. Let $A_\cup = \langle S_1 \cup S_2, T_1 \cup T_2, F_1 \cup F_2 \rangle$. The proof that $\mathcal{L}(A_\cup) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ is left as an exercise.

Complementation Let $A = \langle S, T, F \rangle$ be a complete deterministic tree automaton such that $\mathcal{L}(A) = L$. Define $\bar{A} = \langle S, T, S \setminus F \rangle$. The proof that $\mathcal{L}(A^c) = \mathcal{T}(\Sigma) \setminus \mathcal{L}(A)$ is left as an exercise.

Intersection We use the fact that $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. □

Exercise 73 *Give a construction for $A_\cup$ such that, if A_1 and A_2 are deterministic, so is $A_\cup$. Give a direct construction for $A_\cap$ i.e., without using complementation.* □

In order to define the operations of projection, let $\Sigma = \Sigma_1 \times \Sigma_2 = \{(\sigma_1, \sigma_2) \mid \sigma_1 \in \Sigma_1,\ \sigma_2 \in \Sigma_2,\ \#(\sigma_1) = \#(\sigma_2)\}$ be the product ranked alphabet, where for each pair we have $\#(\sigma_1, \sigma_2) = \#(\sigma_1)$. Let $t : \mathbb{N}^* \to \Sigma$ be a tree. We define $pr_1(t) : \mathbb{N}^* \to \Sigma_1$, where $pr_1(t)(p) = \sigma_1$ iff there exist $\sigma_2 \in \Sigma_2$ such that $t(p) = \langle \sigma_1, \sigma_2 \rangle$. $pr_2(t)$ is defined in a similar way.

Theorem 33 *If $L \subseteq \mathcal{T}(\Sigma_1 \times \Sigma_2)$ is a rational tree language, then so are the projections $pr_1(L)$ and $pr_2(L)$.*

Proof: Left as an exercise. □

Exercise 74 *Let $L \subseteq \mathcal{T}(\Sigma_1)$ be a rational tree language, and Σ_2 be another alphabet. The* cylinder *of L is the language $L' \subseteq \Sigma_1 \times \Sigma_2$ such that $t \in L'$ iff $pr_1(t) \in L$. Show that L' is rational.* □

5.3 Minimization of Tree Automata

A relation $\equiv\ \subseteq \mathcal{T}(\Sigma) \times \mathcal{T}(\Sigma)$ is a *congruence* on $\mathcal{T}(\Sigma)$ iff for every context $C[x_1, \ldots x_n]$ we have $u_i \equiv v_i$, for all $1 \le i \le n \Rightarrow C[u_1, \ldots, u_n] \equiv C[v_1, \ldots, v_n]$.

For a given tree language L, we define the relation $\equiv_L$ on $\mathcal{T}(\Sigma)$ as: $u \equiv_L v$ iff for all contexts $C[x]$ we have $C[u] \in L \iff C[v] \in L$.

Theorem 34 (Myhill-Nerode) *A tree language is rational iff the congruence $\equiv_L$ is of finite index.*

Proof: The proof is similar to the word case, see Theorem 9. □

5.4 Pumping Lemma for Rational Tree Languages

Lemma 18 (Pumping) *Let L be a rational tree language. Then there exists a constant $N > 0$ such that, for every $t \in L$ with $height(t) > N$, there exists a context C, a non-trivial context D and a tree u such that $C[D[u]] \in L$, and, for all $n \geq 0$ we have $C[D^n[u]] \in L$.*

Proof: Left as an exercise.$\square$

Corollary 2 *Let $A = \langle S, T, F\rangle$ be a tree automaton.*

1. *$\mathcal{L}(A) \neq \emptyset$ iff there exists $t \in \mathcal{L}(A)$ with $height(t) < \|S\|$,*
2. *$\|\mathcal{L}(A)\| = \omega$ iff there exists $t \in \mathcal{L}(A)$ with $\|S\| < height(t) < 2\|S\|$.*

Exercise 75 *Let $\Sigma = \{f, g, a\}$, $\#(f) = 2$, $\#(g) = 1$ and $\#(a) = 0$. Prove that the language $L = \{f(g^i(a), g^i(a)) \mid i > 0\}$ is not rational.* $\square$

Exercise 76 *Let $\Sigma = \{f, a\}$, $\#(f) = 2$, and $\#(a) = 0$. Prove that the language $L = \{t \mid \|dom(t)\|$ is a prime number$\}$ is not rational.* $\square$

Exercise 77 *Let L be a tree language over $\Sigma = \{f, a, b\}$, with $\#(f) = 2$ and $\#(a) = \#(b) = 0$. Let $L^{ac} \supseteq L$ be the least superset of L that is closed under the following two rules:*

- commutativity*: $C[f(t_1, t_2)] \in L^{ac} \;\Rightarrow\; C[f(t_2, t_1)] \in L^{ac}$*
- associativity*: $C[f(f(t_1, t_2), t_3)] \in L^{ac} \;\Rightarrow\; C[f(t_1, f(t_2, t_3))] \in L^{ac}$*

Show that there exist rational languages L such that the language L^{ac} is not rational. $\square$

5.5 Top Down Tree Automata

A top-down tree automaton is a tuple $A = \langle S, I, T, F\rangle$ where:

- S is a set of *states*,
- $I \subseteq S$ is a set of *initial states*,
- T is a set of *transition rules* of the form

$$q(f) \rightarrow \langle q_1, \ldots, q_n\rangle$$

  where $\#(f) = n$. For $\#(f) = 0$ there are no rules in T.

- F is a set of *final states*

A *run* of A over a tree $t : \mathbb{N}^* \rightarrow \Sigma$ is a mapping $\pi : dom(t) \rightarrow S$ such that, for each position $p \in dom(t)$, where $q = \pi(p)$, we have:

- if $p = \epsilon$ then $q \in I$, and
- if $\#(t(p)) = n$ and $q_i = \pi(pi)$, $1 \leq i \leq n$, then T has a rule

$$q(t(p)) \rightarrow \langle q_1, \ldots, q_n \rangle$$

A run π is said to be *accepting*, if and only if $\pi(p) \in F$, for all $p \in fr(t)$.

Theorem 35 *Bottom up and top down tree automata recognize the same languages.*

Proof: Left as an exercise.□

A top down tree automaton is said to be *deterministic* if it has one initial state and no two rules with the same left-hand side.

Proposition 7 *A deterministic top down tree automaton has at most one run for each input tree.*

Proof: Left as an exercise. □

Proposition 8 *There exists a rational tree language that is not accepted by any top down deterministic tree automaton.*

Proof: Consider $L = \{f(a,b), f(b,a)\}$ for $\Sigma = \{f, a, b\}$, with $\#(f) = 2$ and $\#(a) = \#(b) = 0$. □

5.6 Decidability

- **Emptiness** $\mathcal{L}(A) = \emptyset$?
- **Equality** $\mathcal{L}(A) = \mathcal{L}(B)$?
- **Infinity** $\|\mathcal{L}(A)\| < \infty$?
- **Universality** $\mathcal{L}(A) = \mathcal{T}(\Sigma)$?

Theorem 36 *The emptiness, equality, infinity and universality problems on tree automata are decidable. In particular, emptiness is decidable in time polynomial in the size (number of states) of automata.*

Proof: The decidability of the emptiness problem is a consequence of the pumping lemma. Moreover, emptiness is decidable in time linear in the number of transition rules, by reducing any accepting run to a run in which each rule is used at most once.

The rest is left as an exercise. □

5.7 Tree Automata and WSkS

In the following we refer to non-deterministic top-down tree automata. By Theorem 35, we can reuse the closure properties of bottom-up tree automata here.

Exercise 78 *Prove that non-deterministic top-down tree automata are closed under union, intersection, complement and projection, without using Theorem 35.* $\Box$

We restrict the alphabet to $\Sigma = \Sigma_2 \cup \Sigma_0$, where $\Sigma_2 \cap \Sigma_0 = \emptyset$ and, for all $\sigma \in \Sigma_2$, $\#(\sigma) = 2$, and for all $\sigma \in \Sigma_0$, $\#(\sigma) = 0$. This is done w.l.o.g., since any tree with finite branching can be embedded in a binary tree.

Let $X_1, \ldots X_k, x_{k+1}, \ldots, x_m$ be a sequence of variables, and assume w.l.o.g. that $\Sigma = \{0,1\}^m$, for some $m \in \mathbb{N}$.

5.7.1 From Automata to Formulae

Let $A = \langle S, I, T, F \rangle$ be a non-deterministic top-down tree automaton, where $S = \{s_1, \ldots, s_p\}$. We shall build a formula Φ_A such that $\mathcal{L}(A) = \mathcal{L}(\Phi_A)$.

Coding of Σ Let $\sigma \in \{0,1\}^m$. We define the formula $\Phi_\sigma(x, \mathbf{X})$ as the conjunction of:

- $X_i(x)$ if the $\sigma_i = 1$, and
- $\neg X_i(x)$ otherwise.

It follows, that for any $t \in \mathcal{T}(\Sigma)$, we have $t \models \forall x \ . \ \bigvee_{\sigma \in \Sigma} \Psi_\sigma(x, \mathbf{X})$.

Coding of S Let $\mathbf{Y} = \{Y_1, \ldots, Y_p\}$ be set variables. Intuitivelly, the set variable Y_i, $1 \le i \le p$ contains all tree positions labeled by A with state s_i during the run on some tree. The following formula says that each position is labeled with exactly one state during some run:

$$\Phi_S(\mathbf{Y}) \ : \ \forall z \ . \bigvee_{1 \le i \le p} Y_i(z) \ \wedge \bigwedge_{1 \le i < j \le p} \neg \exists z \ . \ Y_i(z) \wedge Y_j(z)$$

Coding of I Every run starts from an initial state:

$$\Phi_I(\mathbf{Y}) \ : \ \exists x \forall y \ . \ x \le y \wedge \bigvee_{s_i \in I} Y_i(x)$$

Coding of T If A is at position x, then if $t(x) \in \Sigma_2$, the automaton moves simultaneously on $s_0(x)$ and $s_1(x)$.

$$\Phi_T(\mathbf{X}, \mathbf{Y}) \ : \ \bigwedge_{i=1}^{p} \Big(\forall x \ . \ Y_i(x) \wedge \bigvee_{\sigma \in \Sigma_2} \Phi_\sigma(x, \mathbf{X}) \rightarrow \bigvee_{s_i(\sigma) \longrightarrow \langle s_j, s_k \rangle} Y_j(s_0(x)) \wedge Y_k(s_1(x)) \Big)$$

Coding of F If A is at position x and $t(x) \in \Sigma_0$, then the automaton must be in an accepting state.

$$\Phi_F(\mathbf{X}, \mathbf{Y}) \ : \ \forall x \ . \ \bigvee_{\sigma \in \Sigma_0} \Phi_\sigma(x, \mathbf{X}) \to \bigvee_{s_i \in F} Y_i(x)$$

Exercise 79 *Give the construction of* Φ_A *without using* $\sigma \in \Sigma_2$ *or* $\sigma \in \Sigma_0$. □

5.7.2 From Formulae to Automata

Example 29 *Consider the formula* $x_2 \in X_1$. *We define a non-deterministic top-down tree automaton over the alphabet* $\{0,1\}^2$ *as* $A_{x_2 \in X_1} = \langle \{s_0, s_1\}, s_0, T, \{s_1\} \rangle$, *where* T *has the rules:*

$$\begin{array}{rcl}
\langle 0,0 \rangle (s_0) & \to & \{\langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\} \\
\langle 1,0 \rangle (s_0) & \to & \{\langle s_0, s_1 \rangle, \langle s_1, s_0 \rangle\} \\
\langle 1,1 \rangle (s_0) & \to & \langle s_1, s_1 \rangle \\
\langle 0,0 \rangle (s_1) & \to & \langle s_1, s_1 \rangle \\
\langle 1,0 \rangle (s_1) & \to & \langle s_1, s_1 \rangle
\end{array}$$

Notice that the first two non-deterministic rules are needed in order to label the whole frontier of the tree with s_1, *and not just the nodes that belong to the subtree below* $\langle 1,1 \rangle$. □

Example 30 *Consider the formula* $s_0(x_1) = x_2$. *We define a non-deterministic bottom-up tree automaton over the alphabet* $\{0,1\}^2$ *as* $A_{s_0(x_1)=x_2} = \langle \{s_0, s_1, s_2\}, T, \{s_0\} \rangle$, *where* T *has the rules:*

$$\begin{array}{rcl}
\langle 0,0 \rangle & \to & s_2 \\
\langle 0,1 \rangle & \to & s_1 \\
\langle 1,1 \rangle & \to & s_0 \\
\langle 0,0 \rangle (s_2, s_2) & \to & s_2 \\
\langle 0,1 \rangle (s_2, s_2) & \to & s_1 \\
\langle 1,0 \rangle (s_1, s_2) & \to & s_0 \\
\langle 0,0 \rangle (s_0, s_2) & \to & s_0 \\
\langle 0,0 \rangle (s_2, s_0) & \to & s_0
\end{array}$$

□

As in the case of finite automata, A_Φ can be effectively constructed, as a consequence of Theorems 32 and 33.

Theorem 37 *Given a ranked alphabet Σ, a tree language $L \subseteq \mathcal{T}(\Sigma)$ is definable in WSkS iff it is rational.*

Corollary 3 *The SAT problem for WSkS is decidable.*

Proof: As a consequence of Theorems 37 and 36. $\Box$

5.8 Tree Automata and Horn Clauses

A *Horn clause* is a disjunction of literals with at most one positive literal:

$$\neg p_1 \vee \neg p_2 \vee \ldots \vee \neg p_n \vee q$$

or

$$(p_1 \wedge p_2 \wedge \ldots p_n) \rightarrow q$$

Every transition $f(q_1, \ldots, q_n) \rightarrow q$ of a tree automaton is translated into a Horn clause:

$$P_{q_1}(x_1) \wedge \ldots \wedge P_{q_2}(x_2) \wedge P_{q_n}(x_n) \rightarrow P_q(f(x_1, \ldots, x_n))$$

Decidability of Horn clauses is studied in logic programming (Prolog). An algorithm for deciding satisfiability of Horn clauses gives us an efficient method for deciding emptiness of tree automata.

Chapter 6

Automata on Infinite Trees

In this chapter we consider the *infinite complete binary tree* $t : \{0,1\}^* \to \Sigma$ over an unranked alphabet $\Sigma = \{a, b, c, \ldots\}$. The example below shows that any tree (finite or infinite) of any branching ($k \in \mathbb{N}$ or ω) can be coded within t, using an appropriate alphabet.

Example 31 *Let $t : D \to \Sigma$, $D \subseteq \mathbb{N}^*$ be a k-ary tree. We encode t as an infinite binary tree $T : \{0,1\}^* \to \Sigma_\perp$, where:*

- *for all positions $n_1 n_2 \ldots n_p \in D$, $n_i < k$, we have $T(1^{n_1}01^{n_2}0 \ldots 1^{n_p}) = t(n_1 n_2 \ldots n_p)$,*
- *$T(x) = \perp$ if $x \notin \{1^{n_1}01^{n_2}0 \ldots 1^{n_p} \mid n_1 n_2 \ldots n_p \in D\}$.*

$\square$

6.1 Büchi Tree Automata

Definition 28 *A Büchi tree automaton over Σ is $A = \langle S, I, T, F \rangle$, where:*

- *S is a finite set of* states*,*
- *$I \subseteq S$ is a set of* initial states*,*
- *$T : S \times \Sigma \to 2^{S \times S}$ is the* transition function*,*
- *$F \subseteq S$ is the set of* final states*.*

A *run* of A over a tree $t : \{0,1\}^* \to \Sigma$ is a mapping $\pi : \{0,1\}^* \to S$ such that, for each position $p \in \{0,1\}^*$, where $q = \pi(p)$, we have:

- if $p = \epsilon$ then $q \in I$, and
- if $q_i = \pi(pi)$, $i = 0, 1$ then $\langle q_0, q_1 \rangle \in T(q, t(p))$.

If π is a run of A and σ is a path in t, let $\pi_{|\sigma}$ denote the path in π corresponding to σ.

A run π is said to be *accepting*, if and only if for every path σ in t we have:

$$\inf(\pi_{|\sigma}) \cap F \neq \emptyset$$

A tree $t \in \mathcal{T}^\omega(\Sigma)$ is *accepted* by A iff A has an accepting run on t. The language of A, denoted $\mathcal{L}(A)$ is the set of all trees accepted by A. A set of infinite trees $L \in \mathcal{T}^\omega(\Sigma)$ is said to be Büchi-recognizable iff there exists a Büchi tree automaton A such that $L = \mathcal{L}(A)$.

Exercise 80 *Let $\Sigma = \{a, b\}$. Define $t_n : \{0,1\}^* \to \Sigma$ as $t_n(p) = a$ for $p \in \{\epsilon, 0, 00, \ldots, 0^n\}$ and $t_n(p) = b$, otherwise. Construct a Büchi tree automaton that recognizes the language $\{t_n \mid n \in \mathbb{N}\}$.* □

A Büchi automaton $A = \langle S, I, T, F\rangle$ is said to be *complete* iff for any $s \in S$ and $a \in \Sigma$, $T(s, a) \neq \emptyset$.

Lemma 19 *For every Büchi automaton A there exists a complete Büchi automaton A' such that $\mathcal{L}(A) = \mathcal{L}(A')$.*

Proof: Left as an exercise.□

Theorem 38 *The class of Büchi-recognizable tree languages is closed under union, intersection and projection.*

Proof: Let $A_i = \langle S_i, I_i, T_i, F_i\rangle$, $i = 1, 2$, where $S_1 \cap S_2 = \emptyset$.

Let $A_\cup = \langle S_1 \cup S_2, I_1 \cup I_2, T_1 \cup T_2, F_1 \cup F_2\rangle$. The proof that $\mathcal{L}(A_\cup) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ is left as an exercise.

Let $A_\cap = \langle S, I, T, F\rangle$ where:

- $S = S_1 \times S_2 \times \{0, 1, 2\}$
- $I = I_1 \times I_2 \times \{1\}$
- for any $s, s_1, s_2 \in S_1$, $s', s'_1, s'_2 \in S_2$, $a, b \in \{0, 1, 2\}$ and $\langle (s_1, s'_1, b), (s_2, s'_2, b)\rangle \in T((s, s', a), \sigma)$ iff $\langle s_1, s_2\rangle \in T(s, \sigma)$, $\langle s'_1, s'_2\rangle \in T(s', \sigma)$ and:
    1. if $a = 0$ or ($a = 1$ and $s \notin F_1$), then $b = 1$
    2. if ($a = 1$ and $s \in F_1$) or ($a = 2$ and $s \notin F_1$), then $b = 2$
    3. if $a = 2$ and $s \in F_2$, then $b = 0$
- $F = S \times S \times \{0\}$

The proof that $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ is left as an exercise.

The proof for projection is the same as the proof of Theorem 33. □

6.1.1 Decidability of Emptiness

Let $A = \langle S, I, T, F\rangle$ be a Büchi tree automaton where $F = \{s_1, \ldots, s_m\}$, and $\pi : \{0,1\}^* \to S$ be a successful run of A on the tree $t \in \mathcal{T}(\Sigma)$. For any $s \in S$, and any $u \in \{0,1\}^*$, let $d_u^\pi = \{w \in u \cdot \{0,1\}^* \mid \pi(v) \notin F, \text{ for all } u < v < w\}$. By König's lemma, d_u^π is finite for any $u \in \{0,1\}^*$. If $\pi(u) = s$, let t_s^π denote the restriction of t to d_u^π. Notice that the same tree t_s^π may correspond to more than one position $u \in dom(t)$. Let $T_s = \{t_s^\pi \mid \pi \text{ is a successful run of } A \text{ on } t\}$.

If we denote $\mathbf{s} = \langle s_1, \ldots, s_m\rangle$, it can be easily proved that:

$$\mathcal{L}(A) = \bigcup_{s_0 \in I} T_{s_0} \cdot_{\mathbf{s}} \langle T_{s_1}, \ldots, T_{s_m}\rangle^{\omega\mathbf{s}}$$

Conversely, it can be shown that the expression above denotes a Büchi-recognizable tree language.

Theorem 39 *The emptiness problem for Büchi tree automata is decidable.*

Proof: Let $A = \langle S, I, T, F\rangle$ be a Büchi tree automaton. For each $s \in S$ let T_s be the rational tree language defined above. By Theorem 36, emptiness of T_s is decidable. Eliminate from S (and T) all states s such that $T_s = \emptyset$, and let S' be the resulting set of states. We claim that $\mathcal{L}(A) \neq \emptyset \iff S' \cap I \neq \emptyset$.

"$\Rightarrow$" Obvious.

"$\Leftarrow$" If $S' \cap I \neq \emptyset$. Then $T_{s_0} \neq \emptyset$, hence S' contains also final states. Let $S' \cap F = \{s_1, \ldots, s_k\}$ and $\mathbf{s} = \langle s_1, \ldots, s_k\rangle$. By the construction, we have $T_{s_i} \neq \emptyset$, for all $1 \leq i \leq k$. Since, by the argument above, we have $\mathcal{L}(A) \supseteq \bigcup_{s_0 \in I \cap S'} T_{s_0} \cdot_{\mathbf{s}} \langle T_{s_1}, \ldots, T_{s_m}\rangle^{\omega\mathbf{s}}$, we have $\mathcal{L}(A) \neq \emptyset$. $\Box$

6.1.2 The Complementation Problem

We prove that Büchi tree automata are not closed under complement. Let $\Sigma = \{a, b\}$, and $T_0 = \{t \in \mathcal{T}^\omega(\Sigma) \mid \text{some path in } t \text{ has infinitely many } a\text{'s}\}$

Exercise 81 *Describe a Büchi tree automaton that recognizes T_0.* $\Box$

Let $T_1 = \mathcal{T}^\omega(\Sigma) \setminus T_0 = \{t \in \mathcal{T}^\omega(\Sigma) \mid \text{all paths in } t \text{ have finitely many } a\text{'s}\}$

Proposition 9 *The set T_1 is not Büchi-recognizable.*

Proof: Suppose that there exists a Büchi automaton $A = \langle S, I, T, F\rangle$ such that $\|S\| = n-1$, and $\mathcal{L}(A) = T_1$. Let $t_n : \{0,1\}^* \to \Sigma$ be $t_n(p) = a$ if $p \in \{\epsilon, 1^{m_1}0, 1^{m_1}01^{m_2}0, \ldots, 1^{m_1}01^{m_2}0\ldots1^{m_n}0\}$ for some $m_1, \ldots m_n \in \mathbb{N}$, and $t_n(p) = b$ otherwise. Obviously $t_n \in T_1$.

Let $\pi : \{0,1\}^* \to S$ be a successful run of A on t_n. Since $\|S\| = n - 1$, there must be a path in t_n and three positions $u < v < w$ on this path such that $\pi(u) = \pi(w) = s \in F$ and

$t_n(v) = a$. We define t_1 the tree obtained from t_n by deleting t_u, t_2 the tree obtained from t_1 by deleting t_3, and $t_3 = t_w$. We have $t_n = t_1 \cdot t_2 \cdot t_3$. It is easy to show that A has a successful run on $t_1 \cdot t_2^\omega$. But $t_1 \cdot t_2^\omega$ contains a path with an infinite number of a's, which leads to a contradiction. □

6.2 Müller and Rabin Tree Automata

6.2.1 Müller Tree Automata

Definition 29 *A Müller tree automaton* Σ *is* $A = \langle S, I, T, \mathcal{F} \rangle$*, where:*

- S *is a finite set of* states*,*
- $I \subseteq S$ *is a set of* initial states*,*
- $T : S \times \Sigma \to 2^{S \times S}$ *is the* transition function*,*
- $\mathcal{F} \subseteq 2^S$*, is the set of* accepting sets*.*

Let $t \in \mathcal{T}^\omega(\Sigma)$. Runs of Müller tree automata are defined in the same way as for Büchi tree automata. A run π of A over t is said to be *accepting*, if and only if for every path σ in t we have:

$$\inf(\pi_{|\sigma}) \in \mathcal{F}$$

Exercise 82 *Construct Müller automata recognizing the following tree languages over* $\Sigma = \{a, b\}$*:*

1. *all trees with a finite number of b's,*
2. *all trees* t *where* $t(x) = a$ *implies* $t(y) = b$ *for some* $y > x$*.*

□

Theorem 40 *The class of Müller-recognizable tree languages is closed under union and intersection.*

Proof: For union, the proof is exactly as in the case of Büchi automata. For $A_\cup$, the set of accepting sets is the union of the sets $\mathcal{F}_i$, $i = 1, 2$.

For intersection, let $A_\cap = \langle S_1 \times S_2, I_1 \times I_2, T, \mathcal{F} \rangle$, where:

- $\langle (s_1, s_1'), (s_2, s_2') \rangle \in T((s, s'), \sigma)$ iff $\langle s_1, s_2 \rangle \in T(s, \sigma)$ and $\langle s_1', s_2' \rangle \in T(s', \sigma)$, and
- $\mathcal{F} = \{G \in S_1 \times S_2 \mid pr_1(G) \in \mathcal{F}_1$ and $pr_2(G) \in \mathcal{F}_2\}$, where:
  - $pr_1(G) = \{s \in S_1 \mid \exists s' \,.\, (s, s') \in G\}$, and
  - $pr_2(G) = \{s \in S_2 \mid \exists s' \,.\, (s', s) \in G\}$.

The proof that $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ is left to the reader. □

6.2.2 Rabin Tree Automata

Definition 30 *A Rabin tree automaton* Σ *is* $A = \langle S, I, T, \Omega \rangle$*, where:*

- S *is a finite set of* states*,*
- $I \subseteq S$ *is a set of* initial states*,*
- $T : S \times \Sigma \to 2^{S \times S}$ *is the* transition function*,*
- $\Omega = \{\langle L_1, U_1 \rangle, \ldots, \langle L_n, U_n \rangle\}$ *is the set of* accepting pairs*.*

Let $t \in \mathcal{T}^\omega(\Sigma)$. Runs of Rabin tree automata are defined in the same way as for Büchi tree automata. A run π of A over t is said to be *accepting*, if and only if for every path σ in t there exists a pair $\langle L_i, U_i \rangle \in \Omega$ such that:

$$\inf(\pi_{|\sigma}) \cap L_i = \emptyset \text{ and } \inf(\pi_{|\sigma}) \cap U_i \neq \emptyset$$

Exercise 83 *Show that for every Büchi tree automaton A there exists a Rabin tree automaton B, such that* $\mathcal{L}(A) = \mathcal{L}(B)$. $\square$

Exercise 84 *Show that for every Müller tree automaton A there exists a Rabin tree automaton B, such that* $\mathcal{L}(A) = \mathcal{L}(B)$. $\square$

From Exercises 82, 83, 84 and Proposition 9, it follows that the class of Büchi-recognizable tree languages is strictly included in the class of Rabin-recognizable tree languages.

Theorem 41 *The class of Rabin-recognizable tree languages is closed under union.*

Proof: Left as an exercise.

IMPORTANT The next Theorem has one of the most difficult proofs in the domain of formal languages. There are more proofs to this theorem, each of them using different ideas and techniques. We refer the reader to [3] Chapter 5, pp. 249-327 for a proof using games.

Theorem 42 (Rabin '69) *The class of Rabin-recognizable tree languages is closed under complement.*

Lemma 20 *For any Rabin tree automaton* $A = \langle S, I, T, \Omega \rangle$*, there exists a Rabin tree automaton* $A' = \langle S', \{s'_0\}, T', \Omega \rangle$ *with one initial state such that* $\mathcal{L}(A) = \mathcal{L}(A')$*.*

Proof: Suppose $s'_0 \notin S'$ and let $S' = S \cup \{s'_0\}$. Define $T'(s'_0, \sigma) = \bigcup_{s \in I} T(s, \sigma)$ and $T'(s, \sigma) = T(s, \sigma)$ if $s \neq s_0$. $\square$

Definition 31 *Given an alphabet* Σ*, an infinite tree* $t \in \mathcal{T}^\omega(\Sigma)$ *is said to be* regular *if there are only finitely many distinct subtrees* t_u *of* t*, where* $u \in \{0, 1\}^*$*.*

Theorem 43 (Rabin '72)

1. *Any non-empty Rabin-recognizable set of trees contains a regular tree.*
2. *The emptiness problem for Rabin tree automata is decidable.*

Proof: 1. Let $A = \langle S, I, T, \Omega \rangle$ be a Rabin tree automaton over Σ, such that $\mathcal{L}(A) \neq \emptyset$. Let $A' = \langle S \times \Sigma, I \times \Sigma, T', \Omega' \rangle$ be a Rabin tree automaton over the empty alphabet, where:

- $\langle (s_1, \sigma_1), (s_2, \sigma_2) \rangle \in T'((s, \sigma))$ iff $\langle s_1, s_2 \rangle \in T(s, \sigma)$, and $\sigma_1, \sigma_2 \in \Sigma$.
- $\Omega' = \{\langle L_1 \times \Sigma, U_1 \times \Sigma \rangle, \ldots, \langle L_n \times \Sigma, U_n \times \Sigma \rangle\}$, where $\Omega = \{\langle L_1, U_1 \rangle, \ldots, \langle L_n, U_n \rangle\}$.

It is easy to see that the successful runs of A' are pairs (π, t), where $t \in \mathcal{L}(A)$, and π is a successful run of A on t.

Claim 1 *If A' has a successful run, then A' has also a regular successful run.*

Consider a Rabin tree automaton $A = \langle S, s_0, T, \Omega \rangle$ over the empty alphabet. By Lemma 20, we can assume w.l.o.g. that A has a single initial state. A state $s \in S$ is said to be *live* if $s \neq s_0$ and $\langle s_1, s_2 \rangle \in T(s)$ for some $s_1, s_2 \in S$, where either $s_1 \neq s$ or $s_2 \neq s$.

We prove the claim by induction on the number n of live states in A. Let π be a successful run of A. If $n = 0$, the run will be $\pi(\epsilon) = s_0$, and $\pi(p) = s$, for some $s \in S$, for all $p \neq \epsilon$. For the induction step, we have three cases:

1. if some live state in A is missing on π, apply the induction hypothesis.
2. all states of A appear on π, and there is a position $u \in \{0,1\}^*$ such that $\pi(u) = s$ is live, but some live state s' does not appear in π_u. Let π_1 be the run obtained from π by eliminating π_u, and $\pi_2 = \pi_u$. It can be shown that both π_1 and π_2 are runs of automata with $n-1$ live states, hence there exists successful regular runs π'_1 and π'_2 of these automata. The desired run is $\pi'_1 \cdot_s \pi'_2$.

   Let $A'' = \langle S, s, T, \Omega \rangle$. If π is a successful run of A, then π_2 is a successful run of A'', and A'' has $n-1$ live states. By the induction hypothesis, there exists a regular successful run π'_2 of A''. The regular run we are looking for is $\pi'_1 \cdot_s \pi'_2$.
3. all live states appear in any subtree of π. Let σ be a path in π. Note that σ consists of all live states only (why?). Since π is successful, there exists an accepting pair $\langle L, U \rangle \in \Omega$, such that $\inf(\pi_{|\sigma}) \cap L = \emptyset$ and $\inf(\pi_{|\sigma}) \cap U \neq \emptyset$. Then L contains only non-live states. Let $s \in \inf(\pi_{|\sigma}) \cap U$, and u, v be the first and second positions where s appears on σ. Then π_1 is obtained from π by eliminating π_u, and π_2 is obtained from π_u

by eliminating π_v. We define A' and π'_1 as in the previous case, and $A'' = \langle S, s, T'', \Omega \rangle$, where T'' is obtained by unfolding once the sequence of transitions from s to itself.

If π is a successful run of A, then π_2 is a successful run of A'', and A'' has $n-1$ live states. By the induction hypothesis, there exists a regular successful run π'_2 of A''. The regular run we are looking for is $\pi'_1 \cdot_s {\pi'_2}^{\omega s}$. To show that this is indeed a successful run, we consider the path σ on which s appears infinitely often. Then $\langle L, U \rangle$ is an accepting pair, since $s \in \inf(\pi_{|\sigma}) \cap U$ and $\inf(\pi_{|\sigma}) \cap L = \emptyset$. Since L contains only non-live states, there is a non-live state $s' \in \inf(\pi_{|\sigma})$, since s' is non-live, it is the only state appearing infinitely often on σ.

2. We consider again automata on empty alphabet. Let A be an input-free Rabin tree automaton with n live states. By applying the construction above, we derive $A_{n-1}, A_{n-2}, \ldots, A_0$ from A, having $n-1, n-2, \ldots 0$ live states. If A has a successful run, by the above point it has a regular run, composed of runs of $A_{n-1}, A_{n-2}, \ldots, A_0$. So it is enough to check emptiness of $A_{n-1}, A_{n-2}, \ldots, A_0$. For A_0 the test is trivial. For A_i we use proceed inductively, using the emptiness test for A_{i-1}. $\Box$

6.2.3 Rabin Tree Automata and SkS, $k \geq 2$ and SωS

The main application of Rabin tree automata was establishing the equivalence with the logic S2S [4]. The decidability of SkS and SωS follow, as well as other important results [5], e.g. monadic second order logic of one function symbol (why?).

The direction from Rabin tree automata to S2S formulae is almost the same as in Section 5.7. The only difference is the encoding of the acceptance condition. Let $A = \langle S, I, T, \Omega \rangle$ be a Rabin tree automaton, where $S = \{s_1, \ldots, s_p\}$. Let $\mathbf{Y} = \{Y_1, \ldots, Y_p\}$ be set variables. Intuitivelly, the set variable Y_i, $1 \leq i \leq p$ contains all tree positions labeled by A with state s_i during the run on some tree. We use the set predicate $path(X)$ to express the fact that X is an infinite path in the tree. If X denotes a path, state i appears infinitely often in X iff:

$$inf_i(X) \ : \ \forall x \ . \ X(x) \to \exists y \ . \ x \leq y \wedge X(y) \wedge Y_i(y)$$

The formula expressing the accepting condition is:

$$\Phi_\Omega(\mathbf{Y}) \ : \ \forall X \ . \ path(X) \to \bigvee_{\langle L,U \rangle \in \Omega} \Big(\bigwedge_{s_i \in L} \neg inf_i(X) \ \wedge \ \bigvee_{s_i \in U} inf_i(X) \Big)$$

The direction from formulae to automata is the same as in Section 5.7.

Theorem 44 *Given an alphabet Σ, a tree language $L \subseteq \mathcal{T}^\omega(\Sigma)$ is definable in S2S iff it is rational.*

Corollary 4 *The SAT problem for S2S is decidable.*

Let $\Sigma = \{a, b, \ldots\}$ be an alphabet. Let $\mathcal{T}_2^\omega(\Sigma)$ be the set of all complete infinite binary trees, and $\mathcal{T}_\omega^\omega(\Sigma)$ be the set of all infinite trees in which each position has (at most) ω successors. We define a set $D \subset \{0,1\}^*$ and functions f_i, $i \in \mathbb{N}$, such that any tree $t \in \mathcal{T}_\omega^\omega(\Sigma)$ corresponds to exactly one tree $t' \in \mathcal{T}_2^\omega(\Sigma)$ such that:

- $dom(t') = D$, and
- $\forall p \in dom(t),\ i \in \mathbb{N}\ \exists p' \in dom(t')\ .\ pi \in dom(t) \iff f_i(p') \in dom(t')$

Then, if D and $\{f_i\}_{i\in\mathbb{N}}$ are definable in S2S, by Corollary 4, SωS is also decidable. Let $D = \{\epsilon\} \bigcup_{n_1,\ldots,n_k \in \mathbb{N}} \{1^{n_1}01^{n_2}0 \ldots 1^{n_k}0 \mid k \geq 1, n_i \geq 1, 1 \leq i \leq k\}$.

Lemma 21 *D is S2S-definable.*

Proof: Consider the following formula $\Phi_D(x)$:

$$\exists z \forall y\ .\ (z \leq y) \wedge x = z\ \vee\ \Big(s_1(z) \leq x \wedge \exists y\ .\ y < x \wedge s_0(y) = x \wedge \forall y\ .\ \big(s_0(y) < x \rightarrow s_1(s_0(y)) < x\big)\Big)$$

The proof that $\Phi_D(x)$ defines D is left as an exercise. $\Box$

Lemma 22 *$\{f_i\}_{i\in\mathbb{N}}$ are S2S-definable.*

Proof: First define the relations $x \leq_1 y : D(x) \wedge D(y) \wedge x \leq y$ and $x \preceq_1 y : D(x) \wedge D(y) \wedge x \preceq y$. Then define the relation $x \leq_D y$ iff $x \in D$ and $y = x1^n0$, for some $n \in \mathbb{N}$. This is left as an exercise. Finally, we define $f_0, f_1, f_2, \ldots$ by induction:

- $f_0(x) = y\ :\ D(x) \wedge D(y) \wedge x \leq_D y \wedge \forall z\ .\ x \leq_D z \rightarrow y \preceq_1 z$
- $f_{i+1}(x) = y\ :\ D(x) \wedge D(y) \wedge x \leq_D y \wedge \forall z\ .\ x \leq_D z \rightarrow y \preceq_1 z \wedge \bigwedge_{0 \leq k \leq i} y \neq f_k(x)$.

The correctness of the definitions are left as an exercise. $\Box$

Chapter 7

Alternating Automata

7.1 Alternating Automata on Finite Words

Let $S = \{s_1, s_2, \ldots, s_n\}$ be a finite set of states. By $\mathcal{B}^+(S)$, we denote the set of positive boolean combinations of elements of S. We denote by $\models \subseteq 2^S \times \mathcal{B}^+(S)$ the *satisfaction* relation, with the meaning of satisfaction in propositional logic. If $\varphi \in \mathcal{B}^+(S)$, the *dual* of φ, denoted $\tilde{\varphi}$ is the formula obtained by exchanging $\wedge$ with $\vee$ in φ.

Example 32 *Consider* $\phi = s_1 \wedge (s_2 \vee s_3) \wedge (s_2 \vee s_4) \in \mathcal{B}^+(S)$. *We have* $\{s_1, s_2\} \models \phi$ *and* $\{s_1, s_3, s_4\} \models \phi$. *The dual is* $\tilde{\phi} = s_1 \vee (s_2 \wedge s_3) \vee (s_2 \wedge s_4)$. $\Box$

Lemma 23 *Let* $\varphi \in \mathcal{B}^+(S)$ *and* $X \subseteq S$. *Then* $S \setminus X \models \tilde{\varphi}$ *iff* $S \not\models \varphi$.

Proof: By induction on the structure of φ.$\Box$

Definition 32 *Let* $\Sigma = \{a, b, c, \ldots\}$ *be a finite alphabet. An* alternating finite word automaton *is a tuple* $A = \langle S, I, T, F \rangle$ *where:*

- S *is a finite set of* states*,*
- $I \subseteq S$ *is a set of* initial states*,*
- $T : S \times \Sigma \to \mathcal{B}^+(S)$ *is the* transition mapping*,*
- $F \subseteq S$ *is a set of* final states*.*

A *run* of A on a finite word $w = a_0 a_1 a_2 \ldots a_n$ is a finite tree $\pi : \mathbb{N}^* \to S \times \mathbb{N}$ such that:

- $\pi(\epsilon) = \langle s, 0 \rangle$, for some $s \in I$,
- if $\pi(p) = \langle s, i \rangle$ for some position $p \in \mathbb{N}^*$ and $i < n$, then there exists a set $X = \{s_1, \ldots, s_k\} \subseteq S$ such that:

1. $X \models T(s, a_{i+1})$,
2. the successors of p in $dom(\pi)$ are $\{p \cdot 1, p \cdot 2, \ldots, p \cdot k\}$,
3. $\pi(p \cdot j) = \langle s_j, i+1 \rangle$, for all $1 \leq j \leq k$.

A run is said to be *accepting* iff for all $\langle s, n \rangle \in fr(\pi)$ we have either $s \in F$ or $T(s, w_{n+1}) = \top$. The latter is called the *success condition*, and the states s meeting the success condition are called *success states*.

Theorem 45 *The class of languages recognized by alternating finite word automata coincides with the class of rational languages.*

Proof: Left as an exercise. □

7.2 Alternating Automata on Finite Trees

In this section we work with boolean combinations from the set $\mathcal{B}^+(S \times \mathbb{N})$, with the notion of satisfaction adapted accordingly.

Definition 33 *Let $\langle \Sigma, \# \rangle$ be a ranked alphabet. An* alternating tree automaton *over Σ is a tuple $A = \langle S, I, T, F \rangle$ where:*

- *S is a set of* states*,*
- *$I \subseteq S$ is a set of* initial states*,*
- *$T : S \times \Sigma \to \mathcal{B}^+(S \times \mathbb{N})$ is the* transition mapping*, such that for all $s \in S$ and $f \in \Sigma$ we have $T(s, f) \in \mathcal{B}^+(S \times \{1, 2, \ldots, \#(f)\})$,*
- *F is a set of* final states.

Notice that this definition is a generalization of top-down non-deterministic tree automata.

A *run* of A over a tree $t : \mathbb{N}^* \to \Sigma$ is a tree $\pi : \mathbb{N}^* \to S \times \mathbb{N}^*$ such that, for each position $p \in dom(t)$, where $\langle s, \sigma \rangle = \pi(p)$, we have:

- if $p = \epsilon$ then $s \in I$ and $\sigma = \epsilon$,
- if $\#(t(p)) = k$ there exists a set $X = \{\langle s_1, i_1 \rangle, \ldots, \langle s_k, i_k \rangle\} \in S \times \{1, \ldots, k\}$ such that:
  1. $X \models T(s, t(p))$,
  2. the successors of p in $dom(\pi)$ are $\{p \cdot 1, p \cdot 2, \ldots, p \cdot k\}$,
  3. $\pi(p \cdot j) = \langle s_j, p \cdot i_j \rangle$, for all $1 \leq j \leq k$.

A run is said to be *accepting* iff for all $\langle s, \sigma \rangle \in fr(\pi)$ we have either $s \in F$ or for all successors ρ of σ in t we have $T(s, t(\rho)) = \top$ (*success condition*).

7.2.1 Properties of Alternating Automata on Finite Trees

Lemma 24 *Given an alternating tree automaton A over Σ, there exists an automaton A' with one initial state such that $\mathcal{L}(A) = \mathcal{L}(A')$.*

Proof: Let $A = \langle S, I, T, F\rangle$. Then $A' = \langle S', s_0, T', F\rangle$, where $q_0 \notin S$, $S' = S \cup \{s_0\}$ and let, for all $f \in \Sigma$, $T'(s_0, f) = \bigvee_{s \in I} T(s, f)$ and $T'(s, f) = T(s, f)$, for $s \in S$. $\square$

Theorem 46 *The class of languages recognized by alternating tree automata is closed under union, intersection and complement.*

Proof: Let $A_1 = \langle S_1, I_1, T_1, F_1\rangle$ and $A_2 = \langle S_2, I_2, T_2, F_2\rangle$ be alternating tree automata over Σ, with $S_1 \cap S_2 = \emptyset$. By Lemma 24, we can assume w.l.o.g. that $I_1 = \{s_{01}\}$ and $I_2 = \{s_{02}\}$.

Union Let $s_0 \notin S_1 \cup S_2$. We have $A_\cup = \langle S_1 \cup S_2, s_0, T_\cup, F_1 \cup F_2\rangle$ where for all $f \in \Sigma$ we have $T_\cup(s_0, f) = T_1(s_{01}, f) \vee T_2(s_{02}, f)$, for all $s \in S_1$ we have $T_\cup(s, f) = T_1(s, f)$ and for all $s \in S_2$, we have $T_\cup(s, f) = T_2(s, f)$. The proof that $\mathcal{L}(A_\cup) = \mathcal{L}(A_1) \cup \mathcal{L}(A_2)$ is left as an exercise.

Intersection The same construction as in the case of union, except $T_\cap(s_0, f) = T_1(s_{01}, f) \wedge T_2(s_{02}, f)$. The proof that $\mathcal{L}(A_\cap) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$ is left as an exercise.

Complement Let $A = \langle S, s_0, T, F\rangle$. Then define $\bar{A} = \langle S, I, \tilde{T}, S \setminus F\rangle$, where for all $s \in S$ and $f \in \Sigma$, we have $\tilde{T}(s, f) = \widetilde{T(s, f)}$. The proof that $\mathcal{L}(\bar{A}) = \mathcal{T}(\Sigma) \setminus \mathcal{L}(A)$ is by induction on $t \in \mathcal{T}(\Sigma)$.

If t is a constant, then for all $s \in S$, either $T(s_0, t) \in F$, i.e. $t \in \mathcal{L}(A)$ or $\tilde{T}(s_0, t) = T(s_0, t) \in S \setminus F$, i.e. $t \in \mathcal{L}(\bar{A})$.

If $t = f(t_1, \ldots, t_n)$, for some $f \in \Sigma$ with $\#(f) = n$, and let $T(s, f) = \phi$. Let $X = \{\langle s_j, i_j\rangle \mid 1 \leq j \leq n,\ t_{i_j} \in \mathcal{L}(A, s_j)\}$. Here $\mathcal{L}(A, s)$ is the language accepted by A started in s. By Lemma 23, $S \setminus X \models \tilde{\phi}$ iff $S \not\models \phi$, i.e. either $t \in \mathcal{L}(A)$ or $t \in \mathcal{L}(\bar{A})$, but not both. $\square$

Theorem 47 *Given an alternating tree automaton A, there exists a deterministic bottom-up tree automaton B such that $\mathcal{L}(A) = \mathcal{L}(B)$.*

Proof: Let $A = \langle S, I, T, F\rangle$ be an alternating tree automaton over Σ, and let $B = \langle 2^S, T_B, F_B\rangle$ be a bottom-up deterministic tree automaton over Σ, where:

- T_B is the set of rules

$$
\begin{array}{rcl}
f(S_1, \ldots, S_n) & \rightarrow & \{s \in S \mid \bigcup_{i=1}^{n} S_i \times \{i\} \models T(s, f)\} \\
a & \rightarrow & \{s \in S \mid T(s, f) \in F\}
\end{array}
$$

- $F_B = \{X \subseteq S \mid X \cap I \neq \emptyset\}$

We prove $t \in \mathcal{L}(B, X)$ iff $t \in \mathcal{L}(A, s)$ for all $s \in X$, by induction on the structure of t. If t is a constant, t is accepted by all states s such that $T(s,t) \in F$ hence t is accepted by B.

If $t = f(t_1, \ldots, t_n)$, let $S_1, \ldots, S_n$ be the sets such that $t_i \in \mathcal{L}(A, S_i)$, $1 \leq i \leq n$. $t \in \mathcal{L}(A, s)$ iff there exists $S_0 \subseteq S \times \{1, \ldots, n\}$ such that $S_0 \models T(s, f)$, and for every pair $\langle s_i, i\rangle \in S_0$, we have $t_i \in \mathcal{L}(A, s_i)$. Hence $S_0 \subseteq \bigcup_{i=1}^n S_i \times \{i\}$, therefore $\bigcup_{i=1}^n S_i \times \{i\} \models T(s, f)$. We get that $t \in \mathcal{L}(B, S_0)$ by the induction hypothesis. □

7.2.2 Tree Automata versus Alternating Word Automata

We show the equivalence between alternating word automata and non-deterministic bottom-up tree automata on one-letter alphabets.

First, given a non-deterministic bottom-up tree automaton $A = \langle S, T, F\rangle$ over Σ, we build an alternating word automaton $B = \langle S \times \Sigma, F \times \Sigma, T_B, I \times \Sigma\rangle$ over $\{a\}$, where:

$$T_B(\langle s, f\rangle, a) = \bigvee_{f(s_1,\ldots,s_n) \longrightarrow s \in T} \bigwedge_{i=1}^{n} \bigvee_{g \in \Sigma} \langle\langle s_i, g\rangle, i\rangle$$

Conversely, if $A = \langle S, I, T, F\rangle$ is an alternating word automaton over $\{a\}$, we build a non-deterministic bottom-up tree automaton $B = \langle S, T_B, I\rangle$ over Σ where:

- $\Sigma = \{f_i \mid 1 \leq i \leq n\}$, where $n = \max\{\text{number of conjunctions in } T(s, a) \mid s \in S\}$, and
- $T(s, a) = \bigvee_{i=1}^n \bigwedge_{j=1}^{k_i} \langle s_{i,j}, i\rangle$ iff $f_i(s_{i,1}, \ldots, s_{i,k_i}) \to s \in T_B$

Exercise 85 *Prove the correctness of the constructions above.* □

7.3 Alternating Automata on Infinite Trees

(Weak) alternating automata (WAA) on infinite trees are defined in the same way as alternating automata on finite trees, with the corresponding acceptance condition (Büchi, Rabin, Müller, etc.) The "weakness" is in the following requirement.

Let $A = \langle S, s_0, T, F\rangle$ be an alternating tree automaton:

1. there exists $S_1, S_2, \ldots, S_n$ a partition of S.
2. for all S_i, either $S_i \subseteq F$ or $S_i \cap F = \emptyset$

3. there exists a partial order $\leq$ on $\{S_i\}_{i=1}^n$ such that if $q' \in Q_j$ occurs in $T(q, \sigma)$, for $q \in Q_i$, then $Q_j \leq Q_i$.

Every infinite path on a run of A ultimately gets "trapped" in some Q_i. A run is *accepting* iff $Q_i \subseteq F$ for all paths.

To decide emptiness of WAA, first reduce the automaton to an automaton on a singleton alphabet. Then extend the partial order to a linear order $S_1 \leq S_2 \leq \ldots \leq S_n$. Start by labeling all states in S_i according to the order $\leq$:

- $\top$ if $S_i \subseteq F$
- $\bot$ if $S_i \cap F = \emptyset$

Update the transition relation according by replacing states with $\top$ or $\bot$ in the transition formulae. If $s \in Q_i$ and the only states occurring in $T(s, a)$ are from Q_i, then label Q_i as above. It is left as an exercise to prove that $\mathcal{L}(A) \neq \emptyset$ iff s_0 is labeled with $\top$.

7.3.1 Automata Theoretic CTL Model Checking

If φ is a CTL formula, $cl(\varphi)$ is the set of all subformulae of φ.

Example 33 *For $\varphi = A\Box(E\Diamond p)$ we have $cl(\varphi) = \{A\Box(E\Diamond p),\ E\Diamond p,\ p\}$.* $\Box$

We define $A_\varphi = \langle S, s_0, T, F\rangle$ over $\Sigma = 2^{\mathcal{P}}$, where:

- $S = cl(\varphi) = \{\psi_1, \psi_2, \ldots, \psi_n\}$,
- $S_i = \{\psi_i\}$ and $\{\psi_i\} \leq \{\psi_j\}$ if $\psi_i \in cl(\psi_j)$,
- $s_0 = \varphi$,
- F contains all $\psi \in cl(\varphi)$ of the form $A[\varphi U \psi]$, $E[\varphi U \psi]$, $A[\varphi R \psi]$ or $E[\varphi R \psi]$.
- $T : S \times \Sigma \times \mathbb{N}^+ \to \mathcal{B}(S \times \mathbb{N})$ is defined in the following:
  - $T(p, \sigma, k) = \top$, if $p \in \sigma$ and $T(p, \sigma, k) = \bot$, if $p \notin \sigma$,
  - $T(\neg p, \sigma, k) = \top$, if $p \notin \sigma$ and $T(\neg p, \sigma, k) = \bot$, if $p \in \sigma$,
  - $T(\psi_1 \bullet \psi_2, \sigma, k) = T(\psi_1, \sigma, k) \bullet T(\psi_2, \sigma, k)$, where $\bullet \in \{\vee, \wedge\}$,
  - $T(A\Diamond\psi_1, \sigma, k) = \bigwedge_{c=0}^{k-1} \langle \psi_1, c\rangle$,
  - $T(E\Diamond\psi_1, \sigma, k) = \bigvee_{c=0}^{k-1} \langle \psi_1, c\rangle$,
  - $T(A[\psi_1 U \psi_2], \sigma, k) = T(\psi_2, \sigma, k) \ \vee\ \big(T(\psi_1, \sigma, k) \ \wedge\ \bigwedge_{c=0}^{k-1} \langle A[\psi_1 U \psi_2], c\rangle\big)$,
  - $T(E[\psi_1 U \psi_2], \sigma, k) = T(\psi_2, \sigma, k) \ \vee\ \big(T(\psi_1, \sigma, k) \ \wedge\ \bigvee_{c=0}^{k-1} \langle A[\psi_1 U \psi_2], c\rangle\big)$,
  - $T(A[\psi_1 R \psi_2], \sigma, k) = T(\psi_2, \sigma, k) \ \wedge\ \big(T(\psi_1, \sigma, k) \ \vee\ \bigvee_{c=0}^{k-1} \langle A[\psi_1 R \psi_2], c\rangle\big)$,

– $T(E[\psi_1 R \psi_2], \sigma, k) = T(\psi_2, \sigma, k) \ \wedge \ \big(T(\psi_1, \sigma, k) \ \vee \ \bigwedge_{c=0}^{k-1} \langle E[\psi_1 R \psi_2], c\rangle\big)$,

Note the last parameter of the transition function $k \in \mathbb{N}$. The automaton applies a transition $T(s, \sigma, k) = \phi$ in a state s iff the number of $\rightarrow$-successors of s is k. This type of transition relation is also called "amorphous". This modification of the original definition of WAA has however no impact on the closure properties and the decidability of the emptiness problem.

Let $K = \langle S_K, s_{0K}, \rightarrow, L\rangle$ be a Kripke structure, and $A = \langle S_A, s_{0A}, T_A, F_A\rangle$ be a WAA over Σ. We define $A \otimes K = \langle S_K \times S_A, (s_{0K}, s_{0A}), T_{A_K}, S_K \times F_A\rangle$ over $\{a\}$:

- if $\langle s_1, \ldots, s_k\rangle$ are the $\rightarrow$-successors of s in K, and $T(q, \sigma, k) = \theta$, then $T(\langle s, q\rangle, a) = \theta[(\langle s_1, \eta_1\rangle, 1)/(\eta_1, 1) \ldots (\langle s_k, \eta_k\rangle, k)/(\eta_k, k)]$

For any CTL formula φ, we have $\mathcal{L}(A_\varphi \otimes K) \iff K \models \varphi$. The proof is left as an exercise.

Chapter 8

Recognizability of Integer Sets

In this chapter, we work mostly with the theories $\langle \mathbb{N}, + \rangle$ of Presburger arithmetic, and $\langle \mathbb{N}, +, V_p \rangle$, where $p \in \mathbb{N}$ is a prime number and $V_p : \mathbb{N} \to \mathbb{N}$ is the function defined by:

- $V_p(0) = 1$,
- $V_p(x)$ is the greatest power of p dividing x.

Exercise 86 *Prove that, for all $x \in \mathbb{N}$:*

- $V_4(x) = V_4(2x) \Rightarrow V_2(x) = V_4(x)$, *and*
- $V_4(x) \neq V_4(2x) \Rightarrow V_2(x) = 2V_4(x)$.

□

Example 34 *Consider the relation $x \in_2 y$ iff x is a power of 2 and x occurs in the 2-expansion of y. $\in_2$ is defined in $\langle \mathbb{N}, +, V_2 \rangle$ as follows:*

$$V_2(x) = x \wedge \exists z \exists t \ . \ y = z + x + t \wedge z < x \wedge (t = 0 \vee V_2(t) \geq x)$$

□

Example 35 *Consider the predicate $P_p(x)$ which is true iff x is a power of p, i.e. $P_p(x) \ : \ V_p(x) = x$.*

An *m-dimensional sequence* $(s_n)_{n \geq 0}$ is a function $s : \mathbb{N}^m \to \mathbb{N}$, $m \geq 1$. $(s_n)_{n \geq 0}$ is *definable* in a theory $\mathfrak{T}$ iff there exists a first-order arithmetic formula $\varphi(x_1, \ldots, x_m, y)$ in the language of $\mathfrak{T}$ such that for all $n_1, \ldots, n_m, n \in \mathbb{N}^{m+1}$ we have:

$$f(n_1, \ldots, n_m) = n \iff \mathfrak{T} \models \varphi(n_1, \ldots, n_m, n)$$

Example 36 *The sequence* $s_n = n \mod 3$ *is definable in the theory* $\langle \mathbb{N}, + \rangle$. *Let* $\varphi_0(x) : \exists y \,.\, x = 3y$, $\varphi_1(x) : \exists y \,.\, x = 3y+1$ *and* $\varphi_2(x) : \exists y \,.\, x = 3y+2$. *The formula defining* $(s_n)_{n\geq 0}$ *is*

$$(\varphi_0(x) \wedge y = 0) \vee (\varphi_1(x) \wedge y = 1) \vee (\varphi_2(x) \wedge y = 2)$$

□

A sequence $s : \mathbb{N}^m \to \mathbb{N}$ is definable iff all sets $s^{-1}(n), n \in rng(s)$ are definable.

Note that any set $M \subseteq \mathbb{N}^m$ can be represented by the characteristic sequence $s_M : \mathbb{N}^m \to \{0,1\}$, where $s_M(x) = 1$ if $x \in M$ and $s_M(x) = 0$, otherwise.

8.1 Presburger Arithmetic

In this section we show that the theory $\langle \mathbb{N}, + \rangle$ has quantifier elimination, i.e. each formula in the language of $\langle \mathbb{N}, + \rangle$ is equivalent to an effectively constructible formula without quantifiers. We shall work in the theory $\langle \mathbb{N}, +, \{\equiv_i\}_{i\geq 2}\rangle$, where $x \equiv_i y$ stands for $\exists z \,.\, x - y = iz$.

To eliminate the existentially quantified variable x from $\exists x \,.\, \alpha_1 \wedge \ldots \wedge \alpha_n$, one proceeds with the following steps:

1. **Eliminate the negations** Replace $\neg(t_1 = t_2)$ by $t_1 < t_2 \vee t_2 < t_1$, $\neg(t_1 < t_2)$ by $t_1 = t_2 \vee t_2 < t_1$, and $\neg(t_1 \equiv_m t_2)$ by $\bigvee_{i=1}^{m-1} t_1 \equiv_m t_2 + i$. Then rewrite the formula into DNF, i.e. a disjunction of $\exists x \,.\, \beta_1 \wedge \ldots \wedge \beta_n$, where each β_i is one of the following forms:

$$\begin{array}{rcl} nx & = & u - t \\ nx & \equiv_m & u - t \\ nx & < & u - t \\ u - t & < & nx \end{array}$$

2. **Uniformize the coefficients of** x Let p be the least common multiple of the coefficients of x. Multiply each atomic formula containing nx by $\frac{p}{n}$. In particular, $nx \equiv_m u-t$ becomes $px \equiv_{\frac{p}{n}m} \frac{p}{m}(u-t)$.

3. **Eliminate the coefficients of** x Replace all over the formula px by x and add the new conjunct $x \equiv_p 0$

4. **Special case** If $x = u - t$ occurs in the formula, eliminate directly x by replacing it with $u - t$.

5. **Assume = does not occur** We have a formula of the form

$$\exists x \,.\, \bigwedge_{j=1}^{l} r_j - s_j < x \;\wedge\; \bigwedge_{i=1}^{k} x < t_i - u_i \;\wedge\; \bigwedge_{i=1}^{n} x \equiv_{n_i} v_i - w_i$$

Let $M = [n_i]_{i=1}^n$. The formula is equivalent to:

$$\bigvee_{j=1}^{l}\bigvee_{q=1}^{M}\Big[\bigwedge_{i=1}^{l} r_i - s_i < (r_j - s_j) + q \wedge \bigwedge_{i=1}^{k}(r_j - s_j) + q < t_i - u_i \wedge \bigwedge_{i=1}^{n}(r_j - s_j) + q \equiv_{m_i} v_i - w_i\Big]$$

Notice that a the result quantifier elimination in a Presburger formula is equivalent to a disjunction of conjunctions of atomic propositions of the following forms:

$$\begin{array}{rcl} \sum_{i=1}^{n} a_i x_i + b & \geq & 0 \\ \sum_{i=1}^{n} a_i x_i + b & \equiv_n & m \end{array}$$

The quantifier elimination procedure gives the decidability of Presburger arithmetic.

Exercise 87 *Eliminate the quantifier from the following formula:*

$$\exists y\ .\ w < 4y \wedge 2y < u \wedge 3y < v \wedge y \equiv_3 t$$

□

8.2 One-dimensional Integer Sets

In this section we consider one-dimensional sequences $s : \mathbb{N} \to \mathbb{N}$.

Example 37 *Let $p_2 : \mathbb{N} \to \{0, 1\}$ be the characteristic sequence of the powers of 2:*

$$0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0 \ldots$$

□

Definition 34 *Let $p \geq 2$ be an integer. A p-automaton is a complete DFA $A = \langle S, q_0, T, \Lambda\rangle$ over the alphabet $\{0, 1, \ldots, p-1\}$, whose states are labeled with numbers from $\mathbb{N}$ by a function $\Lambda : S \to \mathbb{N}$. A p-automaton defines a function $f : \{0, 1, \ldots, p-1\}^* \to \mathbb{N}$.*

Notice that the final states of a p-automaton are designated by the Λ function.

Given $n \in \mathbb{N}$, its *p-ary expansion* is the word $w \in \{0, 1, \ldots, p-1\}^*$ such that $n = w(0)p^k + w(1)p^{k-1} + \ldots + w(k)p^0$, denoted also by $(n)_p$. Note that the most significant digit is $w(0)$. Conversely, to any word $w \in \{0, 1, \ldots, p-1\}^*$ corresponds its value $[w]_p = w(0)p^k + w(1)p^{k-1} + \ldots + w(k)p^0$. Notice that $[w]_p = [0w]_p = [00w]_p = \ldots$, i.e. the leading zeros don't change the value of a word.

Definition 35 *A sequence s is said to be* p-recognizable *iff there exists a p-automaton $A = \langle S, q_0, T, \Lambda \rangle$ such that, for all $n \in \mathbb{N}$:*

- $q_0 \xrightarrow{(n)_p} q$, *and*
- $\Lambda(q) = s(n)$

From now on, we will always assume that any p-automaton has a loop $q_0 \xrightarrow{0} q_0$.

A set $M \subseteq \mathbb{N}$ is said to be *recognizable* iff its characteristic sequence $s_M : \mathbb{N} \to \{0, 1\}$ is recognizable. Equivalently, M is recognizable iff there is a finite automaton accepting the set $\{w \in \{0, 1, \ldots, p-1\}^* \mid [w]_p \in M\}$.

Proposition 10 *A set $M \subseteq \mathbb{N}$ is p-recognizable iff there exists a finite automaton A over the alphabet $\{0, 1, \ldots, p-1\}$ such that $M = \{[w]_p \mid w \in \mathcal{L}(A)\}$.*

Proof: Left as an exercise.□

Define the equivalence relation:

$$n \sim_{p,M} m \iff [\forall k \geq 0 \ \forall 0 \leq r < p^k \ . \ np^k + r \in M \iff mp^k + r]$$

Proposition 11 *A set $M \subseteq \mathbb{N}$ is p-recognizable iff the equivalence relation $\sim_{p,M}$ is of finite index.*

Proof: Left as an exercise.□

Proposition 12 *A sequence $s : \mathbb{N} \to \mathbb{N}$ is p-recognizable iff for each $n \in rng(s)$, the set $s^{-1}(n)$ is p-recognizable.*

Proof: Left as an exercise.□

Example 38 *The sequence p_2 is 2-recognizable.* □

Definition 36 *A sequence $s : \mathbb{N} \to \mathbb{N}$ is p-definable if, for each $v \in rng(s)$ there exists a first-order formula φ_v of $\langle \mathbb{N}, +, V_p \rangle$ such that:*

$$s^{-1}(v) = \{n \in \mathbb{N} \mid \langle \mathbb{N}, +, V_p \rangle \models \varphi_v(n)\}$$

Example 39 *The sequence p_2 is 2-definable:*

$$\begin{array}{rcl} p_2^{-1}(1) & = & \{n \in \mathbb{N} \mid \langle \mathbb{N}, +, V_2 \rangle \models V_2(n) = n\} \\ p_2^{-1}(0) & = & \{n \in \mathbb{N} \mid \langle \mathbb{N}, +, V_2 \rangle \models V_2(n) \neq n\} \end{array}$$

□

Exercise 88 *Consider the Thue-Morse sequence $t : \mathbb{N} \to \{0, 1\}$:*

$$1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, \ldots$$

$t(n) = 1$ iff $(t)_2$ has an even number of 1's, and $t(n) = 0$ otherwise. Prove that t is both 2-recognizable and 2-definable. □

8.3 Multi-dimensional Integer Sets

In this section we consider m-dimensional sequences $s : \mathbb{N}^m \to \mathbb{N}$, for $m \geq 2$.

Example 40 *Let us consider the sequence $t : \mathbb{N}^2 \to \{0,1\}$ defined as $t(n,m) = 0$ iff for some $k \geq 0$, we have $(n)_2(k) = (m)_2(k) = 1$, and $t(n,m) = 1$ otherwise.* $\square$

Let $(u,v) \in \left(\{0,1,\ldots,p-1\}^2\right)^*$ be a word, where $u,v \in \{0,1,\ldots,p-1\}^*$, $|u| = |v|$. A p-automaton is defined now over $\left(\{0,1,\ldots,p-1\}^2\right)^*$. The definitions of p-recognizability and p-definability are easily adapted to the m-dimensional case.

Example 41 *The sequence t is 2-recognizable.* $\square$

Example 42 *The sequence t is 2-definable:*

$$\begin{array}{rcl} t^{-1}(0) & : & \exists z \,.\, z \in_2 x \wedge z \in_2 y \\ t^{-1}(1) & : & \forall z \,.\, \neg(z \in_2 x) \vee \neg(z \in_2 y) \end{array}$$

$\square$

Theorem 48 *Let $M \subseteq \mathbb{N}^m$, $m \geq 1$ and $p \geq 2$. Then M is p-recognizable if and only if M is p-definable.*

Proof:

From Automata to Formulae

We define the relations $x \in_{j,p} y$, $0 \leq j < p$, meaning that x is a power of p and the coefficient of x in $(y)_p$ is j, i.e. $x = \sum_{x \in_{j,p} y} j \cdot y$. For $y > x$, we consider $y \in_{0,p} x$ to be satisfied. The function $\lambda_p(x)$ denotes the greatest power of p occurring in $(x)_p$. By convention $\lambda_p(0) = 1$.

$$\begin{array}{rcl} x \in_{j,p} y & : & P_p(x) \wedge [\exists z \exists t \,.\, y = z + j \cdot x + t \;\wedge\; z < x \;\wedge\; (x < V_p(t) \vee t = 0)] \\ \lambda_p(x) = y & : & (x = 0 \wedge y = 1) \vee [P_p(y) \;\wedge\; y \leq x \;\wedge\; \forall z \,.\, (P_p(z) \wedge y < z) \to (x < z)] \end{array}$$

Let $A = \langle S, q_0, T, \Lambda \rangle$ be a p-automaton, with $\Lambda : S \to \{0,1\}$. Suppose $S = \{q_0, q_1, \ldots, q_{l-1}\}$ and replace w.l.o.g. q_k by $e_k = \langle 0, \ldots, 1, \ldots, 0 \rangle \in \{0,1\}^l$, the vector with 0 everywhere and 1 on the k-th position. We build a formula of $\langle \mathbb{N}, +, V_p \rangle$ that defines all accepting runs of A.

The tuple $\langle n_1, \ldots, n_m \rangle \in M$ iff $\langle (n_1)_p, \ldots, (n_m)_p \rangle \in \mathcal{L}(A)$ iff there exists a tuple $\langle y_1, \ldots, y_l \rangle$ of integers such that:

- $\langle (y_1)_p(0), \ldots, (y_l)_p(0) \rangle = \langle 1, 0, \ldots, 0 \rangle$:

$$\varphi_1 \;:\; \bigwedge_{j=1}^{l} 1 \in_{q_0(j),p} y_j$$

- $\Lambda(\langle (y_1)_p(0), \ldots, (y_l)_p(0)\rangle) = 1$, i.e. $\langle (y_1)_p(0), \ldots, (y_l)_p(0)\rangle$ is a final state of A, with $p^k \geq \max_{1 \leq j \leq k} \lambda_p(x_j)$:

$$\varphi_2 \;:\; \bigvee_{\Lambda(q)=1} \bigwedge_{j=1}^{l} z \in_{q(j),p} y_j$$

- for all $0 \leq i < k$, $\langle (y_1)_p(i), \ldots, (y_l)_p(i)\rangle \xrightarrow{\langle (x_1)_p(i), \ldots, (x_m)_p(i)\rangle} \langle (y_1)_p(i+1), \ldots, (y_l)_p(i+1)\rangle$:

$$\varphi_3 \;:\; \forall t \;.\; P_p(t) \wedge t < z \wedge \bigwedge_{T(q,(a_1,\ldots,a_m))=q'} \bigwedge_{j=1}^{l} t \in_{q(j),p} y_j \wedge \bigwedge_{j=1}^{m} t \in_{a_j,p} x_j \rightarrow \bigwedge_{j=1}^{l} p \cdot t \in_{q'(j),p} y_j$$

The formula we are looking for is $\Phi_A \;:\; \exists y_1 \ldots \exists y_l \exists z \;.\; P_p(z) \wedge z \geq \max_{1 \leq j \leq m} \lambda_p(x_j) \wedge \varphi_1(y_1, \ldots, y_l) \wedge \varphi_2(y_1, \ldots, y_l, z) \wedge \varphi_3(x_1, \ldots, x_m, y_1, \ldots, y_l, z)$

From Formulae to Automata

For a given formula $\varphi(x_1, \ldots, x_m)$ in theory $\langle \mathbb{N}, +, V_p\rangle$, we build a p-automaton A_φ over $\{0, 1, \ldots, p-1\}^m$, recognizing the set of words $\{\langle (n_1)_p, \ldots, (n_m)_p\rangle \in \{0, 1, \ldots, p-1\}^m \mid \langle \mathbb{N}, +, V_p\rangle \models \varphi(n_1, \ldots, n_m)\}$. The automaton is built by induction on the structure of the formulae, starting with the automata A_+, $A_=$ and A_{V_p}

If φ is $\phi(x_1, \ldots, x_k, x_{k+1}, \ldots, x_l) \vee \psi(x_{k+1}, \ldots, x_l, x_{l+1}, \ldots, x_m)$ then A_φ is obtained as follows:

- any edge of A_ϕ, labeled by $\langle a_1, \ldots, a_l\rangle$ is replaced with the set of edges labeled by $\{\langle a_1, \ldots, a_l\rangle\} \times \{0, 1, \ldots, p-1\}^{m-l}$. Let the result be denoted by A'_ϕ.
- any edge of A_ψ, labeled by $\langle a_{k+1}, \ldots, a_m\rangle$ is replaced with the set of edges labeled by $\{0, 1, \ldots, p-1\}^k \times \{\langle a_{k+1}, \ldots, a_m\rangle\}$. Let the result be denoted by A'_ψ.

A_ϕ is $A'_\phi \cup A'_\psi$.

$A_{\neg\varphi}$ is the complement of A_φ, and $A_{\exists x_k \varphi}$ is obtained by projection of the k-th dimension in A_φ, and by subsequent determinization. A technical problem is related to the potential elimination of the loop of leading zeros by determinization. The solution is left as an exercise. $\square$

Corollary 5 *The theories $\langle \mathbb{N}, +\rangle$ and $\langle \mathbb{N}, +, V_p\rangle$ are decidable.*

Proof: Follows from the decidability of the emptiness problem for p-automata. $\square$

Corollary 6 *Let $f : \mathbb{N}^m \rightarrow \mathbb{N}^n$ be a function over integers. If f is p-definable and $M \subseteq \mathbb{N}^m$ is p-recognizable, then $f(M)$ is p-recognizable.*

Proof: Left as an exercise.$\square$

8.4 Semilinear Sets

If $u = \langle u_1, \ldots, u_n \rangle, v = \langle v_1, \ldots, v_n \rangle \in \mathbb{N}^n$, we define $u \leq v$ iff $u(i) \leq v(i)$ for all $1 \leq i \leq n$. Since $\langle \mathbb{N}, \leq \rangle$ is well-founded we have:

Lemma 25 *Each set of pairwise incomparable elements of $\mathbb{N}^n$ is finite. In consequence, each set $M \subseteq \mathbb{N}^n$ has a finite number of minimal elements.*

Proof: Left as an exercise. □

Given $C, P \in \mathbb{N}^n$, let $L(C, P) = \{x_0 + x_1 + \ldots + x_m \mid x_0 \in C,\ x_1, \ldots, x_n \in P\}$. An element $x \in L(C, P)$ is of the form $x = x_0 + \sum_{i=1}^{m} \lambda_i x_i$, where $x_0 \in C$, $\lambda_i \in \mathbb{N}$ and $x_i \in P$, for all $1 \leq i \leq m$.

A set $M \in \mathbb{N}^n$ is said to be *linear* if $M = L(c, P)$ for $c \in \mathbb{N}^n$ and finite $P \subseteq \mathbb{N}^n$. A set $M \in \mathbb{N}^n$ is said to be *semilinear* if $M = L(C, P)$ for finite $C, P \subseteq \mathbb{N}^n$.

A function $f : \mathbb{N}^n \to \mathbb{N}^m$ is said to be *linear* if for all $x, y \in \mathbb{N}^n$ we have $f(x+y) = f(x)+f(y)$.

Lemma 26 *Let $M \subseteq \mathbb{N}^n$ be a semilinear set and $f : \mathbb{N}^n \to \mathbb{N}^m$ be a linear function. Then $f(M) \subseteq \mathbb{N}^m$ is a semilinear set.*

Proof: Since $f(x_0+\sum_{i=1}^{p} x_i) = f(x_0)+\sum_{i=1}^{p} f(x_i)$, we have $f(L(\{x_0\}, P)) = L(\{f(x_0)\}, f(P))$. □

Lemma 27 *Let $w \in \mathbb{Z}^n$, $u_i, v_j \in \mathbb{N}^n$ and $a_i, b_j \in \mathbb{Z}$, $1 \leq i \leq p$, $1 \leq j \leq q$. The problem of finding all minimal positive integers $a_1, \ldots, a_p, b_1, \ldots, b_q$ such that:*

$$w = \sum_{i=1}^{p} a_i u_i - \sum_{j=1}^{q} b_j v_j$$

is decidable.

Proof:
1. First we prove the decidability of the existence of positive integers, by induction on $p+q$. If $p+q = 1$ then either $w = a_1 u_1$ (or $w = b_1 v_1$), in which case it is possible to decide whether $a_1 > 0$ (or $b_1 > 0$) exists.

We assume the conclusion for $p + q < m$, $m > 1$ and prove it for $p + q = m$. If all u_i and v_j are independent, then the numbers $r_1, \ldots, r_p, s_1, \ldots, s_q \in \mathbb{Q}$, such that $w = \sum_{i=1}^{p} r_i u_i + \sum_{j=1}^{q} s_j v_j$ if they exist, are unique, and can be effectively found. Then one checks whether $r_1, \ldots, r_p, -s_1, \ldots, -s_q \in \mathbb{N} \setminus \{0\}$.

If u_i and v_j are dependent, we have:

$$\sum_{i \in I} r_i u_i - \sum_{j \in J} s_j v_j = \sum_{i \notin I} r_i u_i - \sum_{j \notin J} s_j v_j$$

for some $r_1, \ldots, r_p, s_1, \ldots, s_q \in \mathbb{N}$ and $I \subseteq \{1, \ldots, p\}$, $J \subseteq \{1, \ldots, q\}$. Consequently we have:

$$w = \sum_{i \in I}(a_i - r_i)u_i + \sum_{i \notin I}(a_i + r_i)u_i + \sum_{j \in J}(b_j - s_j)v_j + \sum_{j \notin J}(b_j + s_j)v_j$$

There are two cases:

- for all $i \in I$, $a_i > r_i$ and for all $j \in J$, $b_j > s_j$. If $a_1, \ldots, a_p, b_1, \ldots, b_q$ is a solution to the original equation, then also $\{a_i - r_i \mid i \in I\} \cup \{a_i + r_i \mid i \notin I\} \cup \{b_j - s_j \mid j \in J\} \cup \{b_j + s_j \mid j \notin J\}$ forms a solution, with $a_i - r_i < a_i$ for some $i \in I$, and $b_j - r_j < b_j$ for some $j \in J$. Continuing in this way, we will eventually obtain a solution with some $a_i < r_i$ or some $b_j < s_j$.

- either $a_i < r_i$ for some $i \in I$, or $b_j < r_j$, for some $j \in J$.

Let i_0, (respectively j_0) be the value found above. Since r_{i_0} and s_{j_0} are known to us, we can enumerate all $1 \leq k < r_{i_0}$ (or $1 \leq k < s_{j_0}$) and find the solution to:

$$w - ku_{i_0} = \sum_{i \neq i_0} a_i u_i - \sum_j b_j v_j$$

or respectively

$$w + kv_{j_0} = \sum_i a_i u_i - \sum_{j \neq j_0} b_j v_j$$

By the inductive hypothesis, this is possible.

2. We show that it is possible to enumerate all minimal positive integers, by induction on $p + q$. If $p + q = 1$, there is only one solution, by the point above.

We assume the conclusion for $p+q < m$, $m > 1$ and prove it for $p+q = m$. By the point above, it is possible to find one positive solution, if one exists. Moreover, one can find a minimal positive solution, by enumeration up to the positive solution given by the previous point. Let $\bar{v} = \langle \bar{a}_1, \ldots, \bar{a}_p, \bar{b}_1, \ldots, \bar{b}_q \rangle$ be a positive minimal solution, and $v' = \langle a'_1, \ldots, a'_p, b'_1, \ldots, b'_q \rangle$ be any other positive minimal solution. Since $\bar{v}$ and v' are incomparable, either (i) $\bar{a}_{i_0} > a'_{i_0}$, for some $1 \leq i_0 \leq p$ or (ii) $\bar{b}_{j_0} > b'_{j_0}$, for some $1 \leq j_0 \leq q$. In the first case, we have that $\langle a'_1, \ldots, a'_{i-1}, a'_{i+1}, \ldots, a'_p, b'_1, \ldots, b'_q \rangle$ is a minimal solution of the equation:

$$w - \bar{a}_{i_0} u_{i_0} = \sum_{i \neq i_0} a_i u_i - \sum_j b_j v_j$$

By the induction hypothesis, this solution can be effectively found, and $0 \leq a'_{i_0} < \bar{a}_{i_0}$ can be found by enumeration. The case (ii) is symmetric. □

Theorem 49 *The class of semilinear subsets of* $\mathbb{N}^n$, $n \geq 1$ *is effectively closed under union, intersection and projection.*

Proof: Proving closure of semilinear sets under union and projection is trivial. For intersection, notice that it is enough to prove that the intersection of two linear sets is semilinear. Let $X = L(\{x_0\}, \{x_1, \ldots, x_p\})$ and $X' = L(\{x'_0\}, \{x'_1, \ldots, x'_q\})$. Let

$$A = \{\langle y_1, \ldots, y_p, z_1, \ldots, z_q\rangle \mid x_0 + \sum_{i=1}^{p} y_i x_i = x'_0 + \sum_{i=1}^{q} z_i x'_i\}$$

and

$$B = \{\langle y_1, \ldots, y_p, z_1, \ldots, z_q\rangle \mid \sum_{i=1}^{p} y_i x_i = \sum_{i=1}^{q} z_i x'_i\}$$

Let $f : \mathbb{N}^{p+q} \to \mathbb{N}^n$ defined as $f(\langle y_1, \ldots, y_p, z_1, \ldots, z_q\rangle) = \sum_{i=1}^{p} y_i x_i$. f is a linear function and $X \cap X' = x_0 + f(A)$. By Lemma 26, it is enough to prove that A is semilinear.

Let C and P be the sets of minimal elements of A and $B \setminus 0^{p+q}$, respectively. By Lemma 25, C and P are finite, and by Lemma 27, C and P are effectively computable. We prove that $A = L(C, P)$. The direction $L(C, P) \subseteq A$ is trivial. For the converse, let $\mathbf{y} \cdot \mathbf{z} \in A$. There exists $\mathbf{y}' \cdot \mathbf{z}' \in C$ such that $\mathbf{y}' \cdot \mathbf{z}' \leq \mathbf{y} \cdot \mathbf{z}$. Let $\mathbf{y}'' \cdot \mathbf{z}'' = \mathbf{y} \cdot \mathbf{z} - \mathbf{y}' \cdot \mathbf{z}'$.

$$\begin{aligned}
\sum_{i=1}^{p} y''_i x_i &= \sum_{i=1}^{p} (y_i - y'_i) x_i \\
&= \sum_{i=1}^{p} y_i x_i - \sum_{i=1}^{p} y'_i x_i \\
&= (x'_0 - x_0) + \sum_{i=1}^{q} z_i x'_i - \left[(x'_0 - x_0) + \sum_{i=1}^{q} z'_i x'_i\right] \\
&= \sum_{i=1}^{q} (z_i - z'_i) x'_i \\
&= \sum_{i=1}^{q} z''_i x'_i
\end{aligned}$$

Hence $\mathbf{y}'' \cdot \mathbf{z}'' \in B$. We prove that each element of B is a sum of elements of P, by induction on $\sum_{i=1}^{p} y_i + \sum_{j=1}^{q} z_i$. This is left as an exercise. □

Theorem 50 (Ginsburg-Spanier) *The class of semilinear subsets of $\mathbb{N}^n$ coincides with the class of Presburger definable subsets of $\mathbb{N}^n$.*

Proof: "$\subseteq$" Let $M = L(C, P) \subseteq \mathbb{N}^k$ be a semilinear set, with $C = \{c_1, \ldots, c_n\} \subset \mathbb{N}^k$ and $P = \{p_1, \ldots, p_m\} \subset \mathbb{N}^k$. The Presburger formula defining M is:

$$\phi(x_1, \ldots, x_k) \;:\; \exists y_1 \ldots \exists y_m \,.\, \bigvee_{i=1}^{n} \bigwedge_{j=1}^{k} x_j = c_i + \sum_{j=1}^{m} y_i p_i$$

"$\supseteq$" Let $\phi(x_1, \ldots, x_k)$ be a Presburger formula. By the result of Section 8.1, we can assume that ϕ is a disjunction of conjunctions of atomic propositions of the following forms:

$$\begin{array}{rcl} \sum_{i=1}^{n} a_i x_i + b & \geq & 0 \\ \sum_{i=1}^{n} a_i x_i + b & \equiv_n & m \end{array}$$

Each atomic proposition describes a semilinear set, and by Theorem 49, their intersections and unions are again semilinear sets. $\square$

8.5 The Cobham-Semenov Theorem

The Cobham-Semenov theorem is probably the most important result in the domain of recognizability of integer sets. However, the proofs are intricate and technical. The interested reader is referred to [6] for such a proof.

Definition 37 *Two integers $p, q \in \mathbb{N}$ are said to be* multiplicatively dependent *if there exist $k, l \geq 1$ such that $p^k = q^l$.*

Equivalently, p and q are multiplicatively dependent iff there exists $r \geq 2$ and $k, l \geq 1$ such that $p = r^k$ and $q = r^l$.

Lemma 28 *Let $p, q \geq 2$ be multiplicatively dependent integers. Let $m \geq 1$ and $s : \mathbb{N}^m \to \mathbb{N}$ be a sequence. Then s is p-recognizable iff it is q-recognizable.*

Proof: We prove that, for any $p, k \geq 1$, the structures $\langle \mathbb{N}, +, V_p \rangle$ and $\langle \mathbb{N}, +, V_{p^k} \rangle$ are equivalent. Since, for some $k, l \geq 1$ we have $p^k = q^l$ this implies the equivalence of $\langle \mathbb{N}, +, V_p \rangle$ and $\langle \mathbb{N}, +, V_q \rangle$, and by Theorem 48, the equivalence between p-recognizability and q-recognizability.

p^k-definable $\Rightarrow$ p-definable Let $\phi(x, y) \ : \ P_{p^k}(y) \wedge y \leq V_p(x)$. We have $V_{p^k}(x) = y \iff \phi(x, y) \wedge \forall z \ . \ \phi(x, z) \to z \leq y$. It remains to define P_{p^k} in $\langle \mathbb{N}, +, V_p \rangle$:

$$P_{p^k}(x) \ : \ P_p(x) \wedge \exists y \ . \ x - 1 = (p^k - 1)y$$

Notice that $p^k - 1$ above is a constant. Indeed, if $x = p^{ak}$ then $p^k - 1 | x - 1$. Conversely, if assume x is a power of p but not of p^k, i.e. $x = p^{ak+b}$, for some $0 < b < k$. Then $x - 1 = p^b(p^{ak} - 1) + (p^b - 1)$, and since $p^k - 1 | x - 1$, we have $p^k - 1 | p^b - 1$, contradiction.

p-definable $\Rightarrow$ p^k-definable Consider the following definition:

$$
\begin{array}{lccc}
\text{if} & V_{p^k}(x) = V_{p^k}(p^{k-1}x) & \text{then} & V_p(x) = V_{p^k}(x) \\
\text{else if} & V_{p^k}(x) = V_{p^k}(p^{k-2}x) & \text{then} & V_p(x) = pV_{p^k}(x) \\
 & \cdots & & \\
\text{else if} & V_{p^k}(x) = V_{p^k}(px) & \text{then} & V_p(x) = p^{k-2}V_{p^k}(x) \\
\text{otherwise} & & & V_p(x) = p^{k-1}V_{p^k}(x)
\end{array}
$$

Notice that p^{k-j} and p^{j-1} above are constants. The correctness of the definition above can be established as a generalization of Exercise 86. □

Theorem 51 (Cobham-Semenov) *Let $m \geq 1$, and $p, q \geq 2$ be multiplicatively independent integers. Let $s : \mathbb{N}^m \to \mathbb{N}$ be a sequence. If s is p-recognizable and q-recognizable, then s is definable in $\langle \mathbb{N}, + \rangle$.*

Bibliography

[1] Dominique Perrin and Jean-Eric Pin. *Infinite Words: Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics Series, Vol. 141, Elsevier, 2004.

[2] Shmuel Safra. *Complexity of Automata on Infinite Objects*. PhD Thesis, Weizmann Institute of Science, 1989.

[3] Bakhadyr Khoussainov and Anil Nerode. *Automata Theory and its Applications*. Progress in Computer Science and Applied Logic Series, Vol. 21. Birkhäuser, 2001.

[4] Michael Rabin. *Decidability of Second-Order Theories and Automata on Infinite Trees*. Transactions of the American Mathematical Society, Vol. 141 (1969), pages 1-35.

[5] Michael Rabin. *Decidable Theories* Handbook of Mathematical Logic, Chapter 3, pages 596-629, North-Holland (1977)

[6] Véronique Bruyère, Georges Hansel, Christian Michaux and Roger Villemaire. *Logic and p-Recognizable Sets of Integers*. Bulletin of Belgian Mathematical Society, Vol. 1 (1994), pages 191-238.