13.3 Emptiness

Goal: To use PTA to derive decidability results, have to be able to decide emptiness.

Trick: Abstract away alphabet and accept a single tree.

Definition:

Let $A = ((\Sigma, rk), Q, q_0, \rightarrow, \Omega)$ be a PTA with maximal rank $n \in \mathbb{N}$ in Σ.

We define $\Sigma^- := \{a\}$ with $rk^-(a) := n$.

Then we set $A^- := ((\Sigma^-, rk^-), Q, q_0, \rightarrow^-, \Omega)$ with

$$q \rightarrow_a^- (q^0, \ldots, \underbrace{q^i, q^i, \ldots, q^i}_{\substack{n-i+1 \\ \text{times}}}) \text{ if } \exists b \in \Sigma \text{ with } rk(b) = i+1 \text{ and } q \rightarrow_b (q^0, \ldots, q^i)$$

Intuitively:

A^- accepts all trees that A accepts without considering the alphabet.

Lemma:

$L(A) \neq \emptyset$ iff. $L(A^-) \neq \emptyset$.

Proof: Homework.

Theorem:

Emptiness for PTA is decidable.

Proof:

- Given a PTA A, we construct A^- from the above lemma.
- We have $L(A) \neq \emptyset$ iff $L(A^-) \neq \emptyset$ iff $t_a \in L(A^-)$.

  Here, t_a is the unique tree where all nodes are labelled by a, the single symbol in Σ^-.
- By a lemma from a previous lecture, $t_a \in L(A)$ iff player A has a winning strategy for $G(A^-, t_a)$ from (ε, q_0).

- Since t_a is labelled by a everywhere, game $G(A^-, t_a)$ does not need information about the node in the tree.
  This means we can represent it as a finite graph:
  $$Pos_A = Q \quad \text{and} \quad Pos_P = Q^n$$
- Use McNZSolve to solve this game. □

<u>Problem</u>: Usually, $L(A) \neq \emptyset$ is not enough, but we want a tree $t \in L(A)$.

<u>Idea</u>: Every regular language over infinite trees contains a tree that is finitely <u>representable</u>.

<u>Definition</u>:
A tree t over (Σ, rk) is finitely <u>representable</u> if there is a system of equations
$$t_1 = a_1(t_{1,1}, \dots, t_{1,rk(a_1)})$$
$$\vdots$$
$$t_m = a_m(t_{m,1}, \dots, t_{m,rk(a_m)})$$
so that $t = t_1$.

<u>Lemma</u>:
$L(A) \neq \emptyset$ iff there is a finitely-representable tree $t \in L(A)$.

<u>Proof</u>:
$\Rightarrow$:
- Let $A = ((\Sigma, rk), Q, q_0, \rightarrow, \Omega)$ with $L(A) \neq \emptyset$.
- Then player A has a positional winning strategy s for $G(A^-, t_a)$.
- This strategy s has the shape
  $$Q \rightarrow Q^n \quad \text{where } n \text{ is the maximal rank.}$$
- We use s to construct a finitely-representable tree.

- To set up the system of equations, consider
$$s(q) = (q^0, \dots, q^k, q^k, \dots, q^k).$$
The move $q \to (q^0, \dots, q^k, q^k, \dots, q^k)$ in $G(A^-, ta)$ exists due to a transition in A:
$$q \to_b (q^0, \dots, q^k) \text{ with } k = rk(b)-1.$$
We add the equation
$$t_q = b(t_{q^0}, \dots, t_{q^k}).$$

↳ We argue that $t_{q_0} \in L(A)$.

- Clearly there is a run of A on t_{q_0} that labels subtree t_q by q.
- To see that the run is accepting, note that <u>every path</u> in this run <u>corresponds to a</u> play of $G(A^-, ta)$ that is conform with s.
- Since s is a winning strategy, the highest priority that occurs infinitely often in this play is even.
- Since the priorities in the play are the priorities of the automata states, the highest priority on each path in the run is even. $\square$

14 Monadic Second Order Logic on Trees

<u>Goal</u>: - Extend MSO by multiple successors and interpret it on trees.
- Prove decidability of satisfiability

<u>Approach</u>: - Büchi, like for finite words
- Employ complementation result.

14.1 Syntax and Semantics

Variables:
- First (V_1) and second (V_2) order variables
- Ranging over positions in an infinite tree
- Note that positions are finite words over a finite alphabet D of directions.

(say $D = \{1,2\}$ for the infinite binary tree).

Definition (MSOT):

Let $V_1 = \{x, y, \ldots\}$ and $V_2 = \{X, Y, \ldots\}$ be countably infinite sets of FO and SO variables.
Formulas in monadic second order logic on trees (MSOT) over V_1, V_2 are defined by

$$\varphi ::= x = y \mid x = \varepsilon \mid x = y.d \mid X(x)$$
$$\mid \varphi_1 \vee \varphi_2 \mid \neg \varphi \mid \exists x. \varphi \mid \exists X. \varphi.$$

Here, $d \in D$, $x, y \in V_1$, $X \in V_2$.

We still use

$\varphi \wedge \psi$, $\varphi \to \psi$, ... as shortcuts.
$\forall x \varphi$, $\forall X. \varphi$

Moreover,

$x \neq y$ means $\neg(x = y)$
and similar for the remaining equalities.

To define the semantics of formulas,
we fix the structure to be the infinite $|D|$-ary tree

$$\mathcal{T}_D := (D^*, (= .d)_{d \in D}).$$

The predicate $x = y.d$ is defined as expected.
It holds for each pair $(x, y) \in D^* \times D^*$ with $x = w.d$ and $y = w$ for some $w \in D^*$.

Since the structure is fixed,
the semantics only depends on the interpretation.

Definition (Semantics of MSOT):

An <u>interpretation</u> is a function

$$\mathcal{I} : V_1 \cup V_2 \longrightarrow D^* \cup \mathbb{P}(D^*)$$

Given an interpretation, the <u>satisfaction relation</u> $\models$
is defined as follows:

$\mathcal{I} \models x = y$ if $\mathcal{I}(x) = \mathcal{I}(y)$

$\mathcal{I} \models x = \varepsilon$ if $\mathcal{I}(x) = \varepsilon$

$\mathcal{I} \models x = y.d$ if $\mathcal{I}(x) = \mathcal{I}(y).d$

$\mathcal{I} \models X(x)$ if $\mathcal{I}(x) \in \mathcal{I}(X)$

$\mathcal{I} \models \varphi \vee \psi$ if $\mathcal{I} \models \varphi$ or $\mathcal{I} \models \psi$

$\mathcal{I} \models \neg\varphi$ if not $\mathcal{I} \models \varphi$

$\mathcal{I} \models \exists x.\varphi$ if there is $w \in D^*$: $\mathcal{I}[^w/_x] \models \varphi$

$\mathcal{I} \models \exists X.\varphi$ if there is $L \subseteq D^*$: $\mathcal{I}[^L/_X] \models \varphi$.

<u>Example</u>:

- Construct a formula $Path(X, x)$ with $x \in V_1$ and $X \in V_2$ free.
- Idea: $\mathcal{I}(X)$ is an infinite set of positions
  that form a path starting in $\mathcal{I}(x)$.
- Simple case: path starts at root.
  Then: ↳ root belongs to X
  ↳ every node in X has precisely one successor in X
  ↳ the successors of all other nodes
  are not part of X.

Use

$$\forall w.\ AuxPath(X, w, \varepsilon)$$

$$
\begin{aligned}
MaxPath(X, w, v) := \;& (w = v \rightarrow X(w)) \\
& \wedge (\neg X(w) \rightarrow \forall y. \bigwedge_{d \in D} y = w.d \rightarrow \neg X(y)) \\
& \wedge (X(w) \rightarrow \exists y. \bigvee_{d \in D} (y = w.d \wedge X(y) \\
& \qquad \wedge \forall z. \bigwedge_{d' \in D \setminus \{d\}} z = w.d' \rightarrow \neg X(z))).
\end{aligned}
$$

↳ Note that we cannot just use

$$Path(X, x) = \forall w. MaxPath(X, w, x).$$

This formula is satisfied by interpretations with arbitrarily many paths in X, one containing x.

↳ Instead we require that every node in X different from x has a predecessor in X.

Hence every subset of X that does not contain x is extended to the root.

By the last conjunct in MaxPath, there is only one such subset.

↳ What remains: X should not form 2 paths if $x \neq \varepsilon$, then ε is not in X.

$$
\begin{aligned}
Path(X, x) = \;& \forall w. (MaxPath(X, w, x) \\
& \wedge (w \neq x \wedge X(w)) \rightarrow \exists v. \bigvee_{d \in D} w = v.d \wedge X(v)) \\
& \wedge x \neq \varepsilon \rightarrow \neg X(\varepsilon).
\end{aligned}
$$

14.2 Decidability:

- A formula uses a finite set of variables $V \subseteq V_1 \cup V_2$.
- Understand interpretations $\mathcal{I}$ as trees $t_{\mathcal{I}}$

  over $\Sigma := \mathbb{P}(V)$ with $rk(x) := |D|$ for all letters:

  $$t_{\mathcal{I}}(w) := \{ X \in V \mid w \in \mathcal{I}(X) \} \cup \{ x \in V \mid w = \mathcal{I}(x) \}.$$

  Note that this is equivalent to $\mathbb{B}^{|V|}$,

  the labelling we used for WMSO.

- By induction on the structure of formulas φ,

  we construct a PTA A_φ

  that accepts the satisfying interpretations:

  ↳ $\exists$-quantifiers: projection (non-determinism)

  ↳ Disjunction: union

  ↳ Negation: complementation

  ↳ Atomic formulas: explicit.

Theorem (Rabin '69):

For every MSOT formula φ there is a PTA A_φ

so that for all interpretations $\mathcal{I}$ of $Vars(\varphi)$

we have

$$t_{\mathcal{I}} \in L(A_\varphi) \quad \text{iff} \quad \mathcal{I} \models \varphi.$$

Corollary:

Satisfiability for MSOT is decidable.