12 Parity Tree Automata

Infinite trees:
- finite outdegree
- infinitely many nodes

↳ Only top-down processing.

↳ Extend acceptance for finite trees (≙ leaves reach final states)

↳ Acceptance holds for every path

Goal: Complementation
- A tree is not accepted if there is a path that is not accepting.
- How to encode this ∃-quantifier into the ∀-quantified acceptance condition?

Trick: Parity games

12.1 Parity Tree Automata:

Let (Σ, rk) be a ranked alphabet with $rk(a) > 0$ for all $a \in \Sigma$.

Definition (Infinite trees):
- An infinite tree over (Σ, rk) is a function $t: T \to \Sigma$, where
  - $T \subseteq \mathbb{N}^*$ is a non-empty, prefix-closed set and
  - for all $w \in T$ and all $i < rk(t(w))$: $w.i \in T$.

  The second condition means T is infinite.
- A path in $t: T \to \Sigma$ is a word $\pi \in \mathbb{N}^\omega$ so that every finite prefix u of π is in T.

  The path defines an infinite word $t(\pi) \in \Sigma^\omega$ by $t(\pi)_i := t(\pi_0 \dots \pi_{i-1})$.

Definition (Parity tree automata):
- A parity tree automaton is a tuple $A = ((\Sigma, rk), Q, q_0, \to, \Omega)$

where
- (Σ, rk) is a ranked alphabet

- Q is a finite set of states
  with $q_0 \in Q$ the initial state
- $\Omega : Q \to \mathbb{N}$ assigns priorities to states and
- $\longrightarrow = (\longrightarrow_a)_{a \in \Sigma}$ with

$$\longrightarrow_a \subseteq Q \times Q^{rk(a)}.$$

- A run of $\mathcal{A}$ on $t : T \to \Sigma$ is a function
  $r : T \to Q$
  so that
  - $r(\varepsilon) = q_0$ and
  - for $a = t(w)$ with $rk(a) = n$

$$\underbrace{r(w)}_{q} \longrightarrow_a (\underbrace{r(w.0)}_{q^0}, \ldots, \underbrace{r(w.(n-1))}_{q^{n-1}})$$

- The run is accepting if for all paths
  $\pi = \pi_0 \pi_1 \pi_2 \ldots \in \mathbb{N}^\omega$
  we have: the largest number that occurs infinitely often in
  $\Omega(r(\varepsilon))\Omega(r(\pi_0))\Omega(r(\pi_0\pi_1))\ldots$ is even.
- The language of $\mathcal{A}$ is
  $L(\mathcal{A}) = \{ t : T \to \Sigma \mid$ there is an accepting run of $\mathcal{A}$ on $t \}$.

A tree language L of infinite trees is called regular (or PTA-recognizable)
if $L = L(\mathcal{A})$ for some PTA $\mathcal{A}$.

Complementation needs closure properties.
Consider ranked alphabets (Σ, rk) and (Σ', rk').
A function
$$f : \Sigma \to \Sigma'$$
is called rank-preserving if $rk'(f(a)) = rk(a)$ for all $a \in \Sigma$.

Lemma (Closure properties):
(1) If L and L' are regular, so is $L \cup L'$.
(2) If L over (Σ, rk) is regular and $f : (\Sigma, rk) \to (\Sigma', rk')$
is rank-preserving, then $f(L)$ is regular.

↳ Let (Σ, rk) be a ranked alphabet with maximal rank n.

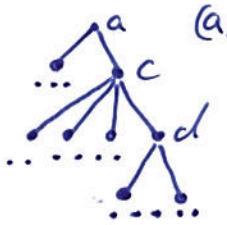
↳ Let $D = \{0, \dots, n-1\}$.

↳ Let $L \subseteq (\Sigma \times D)^\omega$ be an <u>ω-regular</u> language.

Define the language of infinite trees:

$$L^t := \{t : T \to \Sigma \mid \text{every path } \pi \text{ in } t \text{ when written as a word in } \Sigma \times D \text{ is in } L\}$$

<u>Example:</u>



$(a,0).(c,1)(d,3)\dots$

<u>Lemma:</u>

L^t is ω-regular.

<u>Proof:</u>

- Let $L = L(B)$ for B an NBA.
- Every NBA can be turned into a <u>deterministic</u> parity automaton (DPA)
- Hence, we can assume $L = L(A)$ with $A = (\Sigma \times D, Q, q_0, \to, \Omega)$ a DPA.
- To accept L^t, we define the PTA

$$(\Sigma, Q, q_0, \to', \Omega)$$

with $q \xrightarrow{a}{}' (q^0, \dots, q^{n-1})$ if $q \xrightarrow{(a,i)} q^i$ for all $0 \le i < n$. □

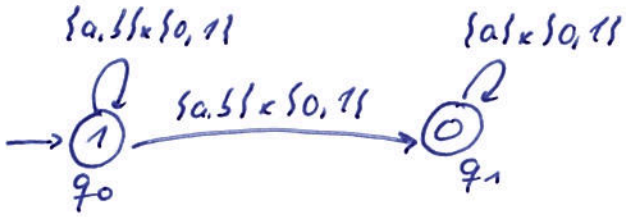
- Note that the resulting PTA is deterministic.
- Indeed, the construction does not work without the determinisation step.

To see that we cannot start from a non-deterministic parity automaton (NPA), consider

$$L = (\Sigma \times D)^* \cdot (\{a\} \times D)^\omega \text{ with } \Sigma = \{a, b\} \text{ and so } D = \{0,1\}.$$

"On every path, there are finitely many 'b's".

- The language is accepted by the NPTA:



- If we adapt the above construction, we obtain the PTA:

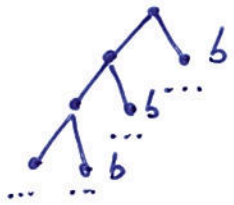
$$q_0 \longrightarrow_{a/b} (q_i, q_j) \text{ with } i,j \in \{0,1\}$$

$$q_1 \longrightarrow a\ (q_1, q_1)$$

Moreover, $\Omega(q_0) = 1$ and $\Omega(q_1) = 0$.

- To show that this PTA does not accept L^t, consider the tree t with

$$t(0^*1) = b \text{ and } t(w) = a \text{ otherwise.}$$



We have $t \in L^t$.

- To see that t is not accepted by the PTA constructed above, assume to the contrary there was an accepting run

$$r: T \longrightarrow Q.$$

Consider the path $\pi = 0^\omega$ in t.

Since r is accepting, the highest priority that occurs infinitely often in

$$\Omega(r(\varepsilon)).\Omega(r(0)).\Omega(r(0^2))\ldots$$

has to be even.

- This means there is $i \in \mathbb{N}$ with

$$r(0^i) = q_1.$$

Since there is only a single transition from q_1,
we get

$$r(0 \cdot 1) = q_1.$$

But since q_1 only accepts "a"s, the PTA fails to accept
the "b" at $t(0 \cdot 1)$.

This means $t \notin L(A)$, where A is the PTA. ↯

The idea behind the construction is that
↳ we force the non-deterministic word automaton
into a decision (enter q_1) and
↳ make the tree automaton dependent on this choice.

Like for top-down tree automata over finite trees,
we obtain:

Lemma:
There are languages of infinite trees that are PTA-recognizable,
but cannot be accepted by a deterministic PTA.

12.2 Complementation:

Trick:
- Understand acceptance of a tree t by a PTA A
as a parity game $G(A,t)$:

$$t \in L(A) \quad \text{iff} \quad A \text{ has a winning strategy for } G(A,t) \text{ (from } (\varepsilon, q_0)).$$

- With positional determinacy:

$$t \notin L(A) \quad \text{iff} \quad P \text{ has a winning strategy}$$
$$(P = \underline{\text{Path finder}}).$$

- Model existence of a winning strategy for P
as a PTA.

Technically: The game construction works as follows:

↳ Put a token on the root of the tree and set A to q_0.

↳ In every round, A selects a transition $q \to_a (q^0, \dots, q^k)$ where $k \le n$, the maximal rank.

↳ Then P moves the token to one of the child nodes. The automaton switches state accordingly.

Definition ($G(A,t)$):

Let $A = ((\Sigma, rk), Q, q_0, \to, \Omega)$ be a PTA and $t: T \to \Sigma$.

The parity-game $G(A,t)$ is defined as follows.

(a) The positions are pairs

for A: (w, q) with $w \in D^*$ a position in t and $q \in Q$.

for P: $(w, \underbrace{(q^0, \dots, q^k)}_{=: \bar{q}})$ with $w \in D^*$ and $\bar{q} \in Q^{\le n}$.

(b) The moves for A are defined as follows:

In (w, q) with $a = t(w)$, player A selects a transition $q \to_a \bar{q}$ and reaches $(w, \bar{q})$.

(c) In position $(w, \bar{q})$, player P selects $i \le rk(t(w))$, and moves to $(w.i, q^i)$ with $q^i = \bar{q}(i)$.

(d) The priority is $\Omega(w, q) := \Omega(q)$ and $\Omega(w, \bar{q}) := 0$.

Lemma:

Player A wins $G(A, t)$ from (ε, q_0) if $t \in L(A)$.

Theorem:
Given a PTA $\mathcal{A}$, we can effectively construct a PTA $\overline{\mathcal{A}}$ with $\mathcal{L}(\overline{\mathcal{A}}) = \overline{\mathcal{L}(\mathcal{A})}$.

Proof:
- If $t \notin \mathcal{L}(\mathcal{A})$, then player A does not have a winning strategy for $\mathcal{G}(\mathcal{A}, t)$ from (ℓ_0, q_0).
- By positional determinacy, this means player P has a positional winning strategy
$$str : T \times Q^* \to D.$$
- We can consider the strategy as a function
$$str : T \to S \quad \text{with} \quad S := Q^n \to D.$$
- With S, we define the new ranked alphabet
$$\Delta := \Sigma \times S \quad \text{where} \quad rk(a, s) := rk(a).$$

Observation:
↳ We have $t \notin \mathcal{L}(\mathcal{A})$
iff there is t' over $\Delta = \Sigma \times S$ so that
- $proj_1(t') = t$ and
- the S-labels define a positional winning strategy for P.

↳ More formally, we define the set $\mathcal{L}'$ of trees t' over $\Delta = \Sigma \times S$ such that
$$str(w, \bar{q}) := (proj_2(t'(w)))(\bar{q})$$
forms a positional winning strategy for P in $\mathcal{G}(\mathcal{A}, proj_1(t'))$.

Point:

- We have
$$\overline{L(A)} = \text{proj}_\Gamma(L').$$
- Hence, if L' is PTR-recognizable, then so is $\overline{L(A)}$, using the closure properties stated above.