Alternating Parity Tree Automata (All Flavors of Finite-State Automata)

Goal: Generalize NFAs on finite words.

Three dimensions:

↳ Alternation / angelic and demonic non-determinism rather than just angelic non-determinism.
Can be used to simultaneously check several conditions.

↳ Automata on trees rather than words.
Process input top-down.

↳ Automata on infinite objects.
Acceptance by parity condition:
highest priority that repeats infinitely often has to be even.

1. Syntax of Alternating Parity Tree Automata

Definition:

Let X be a finite set.

- The set $B^+(X)$ of positive Boolean formulas over X is defined by

$$\Theta ::= true \mid false \mid x \mid \Theta \wedge \Theta \mid \Theta \vee \Theta,$$

where $x \in X$.

- A subset $Y \subseteq X$ satisfies Θ, if assigning true to the elements in Y and false to the elements in $X \setminus Y$ makes Θ true.

Definition (Syntax of Alternating Parity Tree Automata):

An alternating parity tree automaton (APTA) is a tuple

$$A = (\Sigma, Q, \delta, q_I, \Omega),$$

where
- Σ is a ranked alphabet (finite).
  We write $f/_k \in \Sigma$ if $f \in \Sigma$ and $arity(f) = k$.
  Let m be largest arity of a letter from Σ.
- Q is a finite set of states
  with $q_I \in Q$ the initial state.
- $\delta : Q \times \Sigma \longrightarrow B^+(\{1, \dots, m\} \times Q)$
  is the transition function, satisfying
  $$\forall q \in Q. \forall f/_k \in \Sigma. \ \delta(q, f) \in B^+(\{1, \dots, k\} \times Q).$$
- $\Omega : Q \longrightarrow \mathbb{N}$ is the priority function
  used to define acceptance.

Instead of jumping into the semantics,
we illustrate the behavior of APTA on special cases.

2. Parity Word Automata

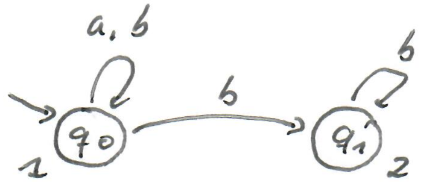
A parity word automaton is an APTA

$$A = (\Sigma, Q, \delta, q_I, \Omega),$$

where
- all symbols in Σ have arity 1,
- the Boolean formulas $\delta(q, f)$ do not use conjunction $\wedge$.

Example:

$\mathcal{A}$:



"Finitely many a's."

This is the APTA

$$(\{a/1, b/1\}, \{q_0, q_1\}, \delta, q_0, \{q_0 \mapsto 1, q_1 \mapsto 2\})$$

with

$$\delta(q_0, a) := (1, q_0)$$

Does not matter as the alphabet is unary.

$$\delta(q_0, b) := (1, q_0) \vee (1, q_1)$$

$$\delta(q_1, a) := \text{false}$$

$$\delta(q_1, b) := (1, q_1).$$

The language will be

$\mathcal{L}(\mathcal{A}) = \{a, b\}^{*} . b^{\omega}$, written as

a(b(a(... ...b(b(b(...

Why?

To accept, the highest priority that occurs infinitely often has to be even.

This eventually forces a move to q_1.

3. Alternating Parity Word Automata

An alternating parity word automaton is an APTA

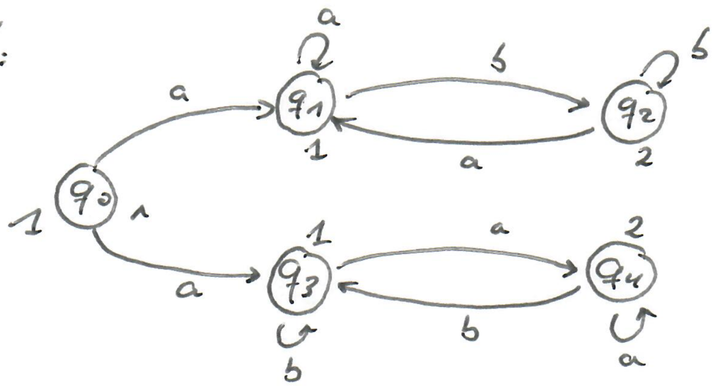
$$\mathcal{A} = (\Sigma, Q, \delta, q_I, \Omega),$$

where all symbols in Σ have arity 1.

Note that we admit general positive Boolean formulas in the transition function.

Example:

$\mathcal{A}$:



"Infinitely many as and infinitely many bs starting from a."

This is an APTA with the transition relation:

$$\delta(q_0, a) = (1, q_1) \wedge (1, q_3)$$

$$\delta(q_0, b) = \text{false}$$

$$\delta(q_1, a) = (1, q_1) \qquad \delta(q_3, a) = (1, q_4)$$

$$\delta(q_1, b) = (1, q_2) \qquad \delta(q_3, b) = (1, q_3)$$

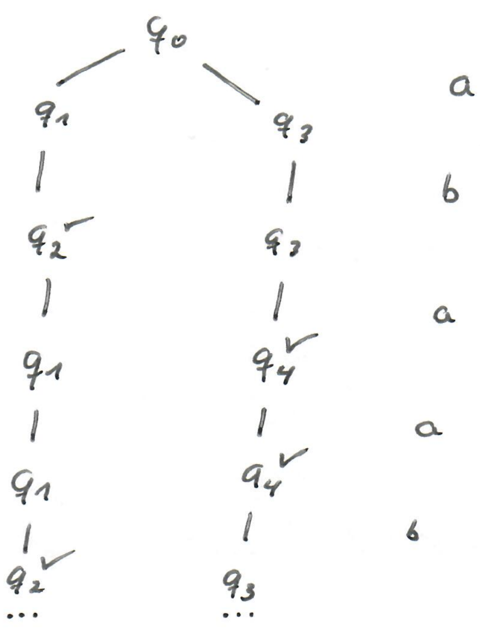
$$\delta(q_2, a) = (1, q_1) \qquad \delta(q_4, a) = (1, q_4)$$

$$\delta(q_2, b) = (1, q_2) \qquad \delta(q_4, b) = (1, q_3).$$

Consider the word

$$w = abaabaaab\ldots \in L(\mathcal{A}).$$

A run of $\mathcal{A}$ on w will be a tree, where all branches act on w:



The run is accepting as all branches are accepting.

4. Parity Tree Automata

A *parity tree automaton* is an APTA

$$\mathcal{A} = (\Sigma, Q, \delta, q_I, \Omega),$$

where the Boolean formulas

$$\delta(q, f) = \bigvee_{o} \bigwedge_{p} (i_{o,p}, q_{o,p})$$

are in a disjunctive normal form that satisfies the following:

every co-clause $\bigwedge_{p} (i_{o,p}, q_{o,p})$ contains precisely one entry (i, q) for all $1 \le i \le arity(f)$.

Note that we admit symbols of arbitrary arity.

Example: Let $\Sigma = \{a/_2, b/_1, c/_0\}$.

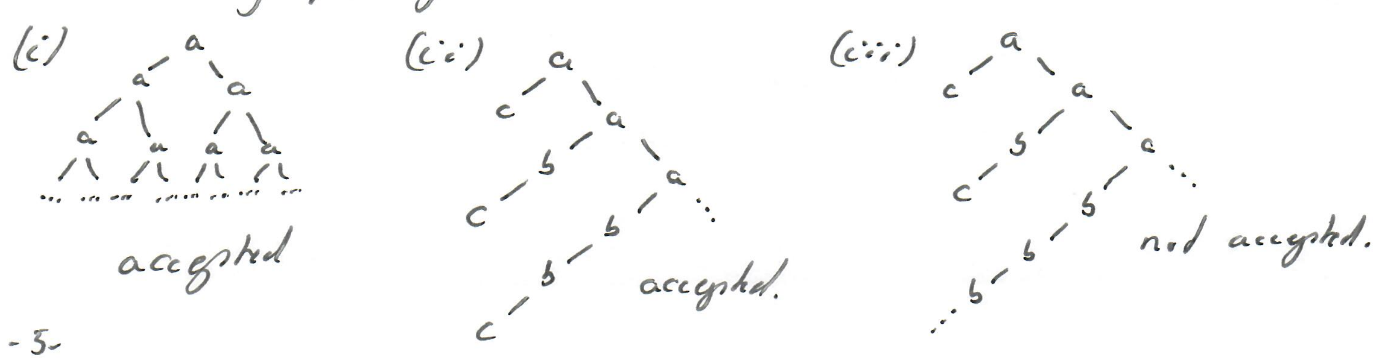
- Let $\mathcal{A}_1 = (\Sigma, \{q_0, q_1\}, \delta_1, q_0, \{q_0 \mapsto 2, q_1 \mapsto 1\})$, where for each $q \in \{q_0, q_1\}$:

$$\delta_1(q, a) = (1, q) \wedge (2, q)$$

$$\delta_1(q, b) = (1, q_1)$$

$$\delta_1(q, c) = \text{true}.$$

Then $\mathcal{A}_1$ accepts a Σ-labelled tree t iff in every path of t where b occurs c eventually occurs.



- Let $A_2 = (\Sigma, \{q_0, q_1\}, \delta_2, q_0, \{q_0 \mapsto 2, q_1 \mapsto 1\})$,

where for all $q \in \{q_0, q_1\}$:

$$\delta_2(q, a) = (1, q_1) \wedge (2, q)$$

$$\delta_2(q, b) = (1, q)$$

$$\delta_2(q, c) = \text{true}.$$

Now A_2 accepts a Σ-labelled tree t iff
every subtree of t that takes a left branch
of an a-labelled node
is finite, ending on c.

5. Semantics of Alternating Parity Tree Automata

<u>Definition</u>:

- A <u>run tree</u> of an APTA $A = (\Sigma, Q, \delta, q_I, \Omega)$
over a Σ-labelled tree t
is a $(\text{dom}(t) \times Q)$-labelled unranked tree r
so that
  - $\varepsilon \in \text{dom}(r)$ and $r(\varepsilon) = (\varepsilon, q_I)$.
    // Run starts at the root
    and in the initial state.
  - for all $\beta \in \text{dom}(r)$ with $r(\beta) = (\alpha, q)$
there is a set S
    ↳ that satisfies $\delta(q, a)$ with $a = t(\alpha)$ and
    ↳ for each $(i, q') \in S$ there is j so that
    $\beta . j \in \text{dom}(r)$ and $r(\beta . j) = (\alpha . i, q')$.

- Let $\pi = \pi_1 \pi_2 \ldots$ be an infinite path in r.

  For each $i \geq 0$, let

  the state label of node $\pi_1 \ldots \pi_i$ be q_{n_i}.

  Note that for q_{n_0}, the state label of ε,

  we have $q_{n_0} = q_I$.

  We say that <u>π satisfies the parity condition</u>,

  if

  the highest priority that occurs ω-often

  in $\Omega(q_{n_0}) \Omega(q_{n_1}) \ldots$ is even.

- A run is <u>accepting</u>,

  if every infinite path in it

  satisfies the parity condition.

- The APTA <u>A accepts t</u>,

  if there is an accepting run tree of A on t.

<u>Note:</u>

- Finite paths in a run tree end on true.
- They are accepting by default.