Automata, Games & Verification

Summary #11

Today at 4:15pm in SR 014

Games in Verification and Synthesis

Daniel Dahrendorf: *Counterexample-guided control*

Georg Neis: *Three-valued abstractions of games*

Reminder: Exam on Friday, July 25, 14:00-16:00
in Hörsaal 002, building E 1 3.

The exam will be *open book*, i.e., handwritten notes as well as books and other printed materials are allowed.

Determinacy of parity games

Assumptions:

- arena is finite or countably infinite.
- the number of colors is finite (max color k).

Lemma 1. Merging strategies

Given a parity game $\mathcal{G}$ and a set of nodes $U \subseteq V$, s.t. for every $p \in U$, Player σ has a memoryless strategy $f_{\sigma,p}$ that wins from p, then there is a memoryless winning strategy f_σ that wins from all $p \in U$.

Theorem 1. *Parity games are memoryless determined.*

Nondeterministic Tree Automata

Definition 1. *A nondeterministic tree automaton (over binary Σ-trees) $\mathcal{A} = (S, s_0, M, \varphi)$ consists of*

- S*: finite set of states;*
- $s_0 \in S$*;*
- $M = S \times \Sigma \times S \times S$*;*
- φ*: acceptance condition (Büchi, parity, . . .).*

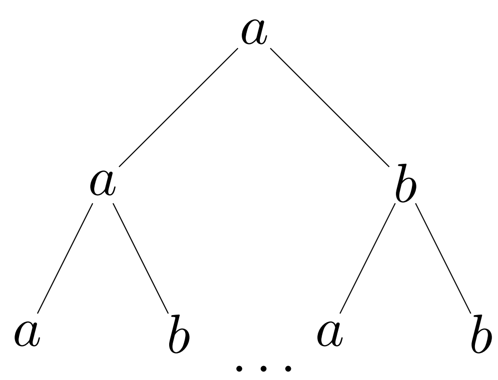
Definition 2. *A run of a nondeterministic tree automaton $\mathcal{A}$ on a Σ-tree*
v
is a S-tree (T, r), s.t.

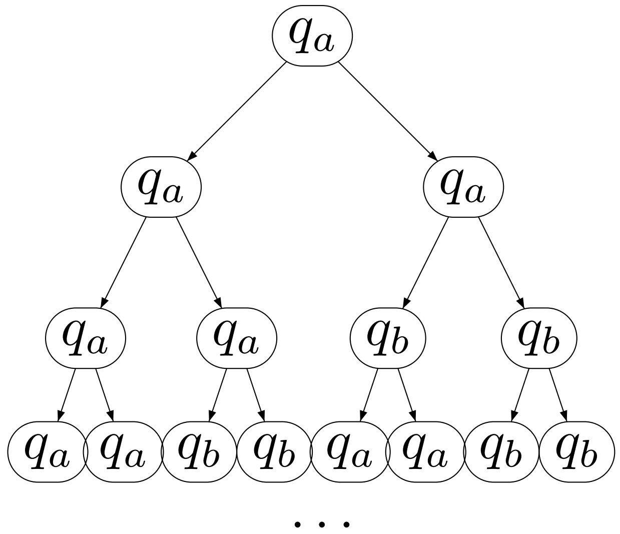
- $r(\epsilon) = s_0$*;*
- $(r(q), v(q), r(q0), r(q1)) \in M$ *for all* $q \in \{0, 1\}^*$*;*

A run is accepting if every branch is accepting.

Example: $\{a, b\}$-trees with infinitely many bs on each path.

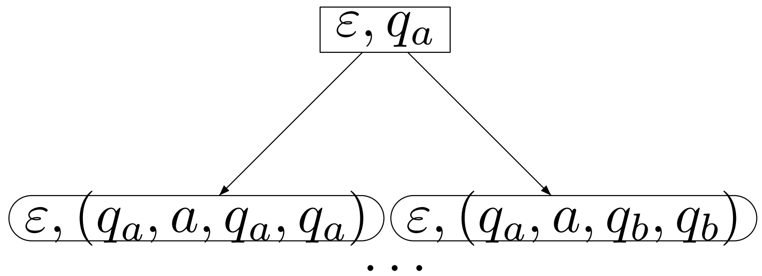
$\mathcal{A} = (S, s_0, M, c); \Sigma = \{a, b\};$
$S = \{q_a, q_b\}; s_0 = q_a;$
$M = \{(q_a, a, q_a, q_a), (q_b, a, q_a, q_a), (q_a, b, q_b, q_b), (q_a, a, q_b, q_b), \ldots\};$
Büchi $F = \{q_b\}$.





Theorem 2. [Acceptance Game] *A parity tree automaton* $\mathcal{A} = (S, s_0, M, c)$ *accepts an input tree* t *iff Player 0 wins the parity game* $\mathcal{G}_{\mathcal{A},t} = (V_0, V_1, E, c')$ *from position* (ε, s_0)*.*

- $V_0 = \{(w, q) \mid w \in \{0,1\}^*, q \in S\}$;
- $V_1 = \{(w, \tau) \mid w \in \{0,1\}^*, \tau \in M\}$;
- $E = \{((w,q),(w,\tau)) \mid \tau = (q, t(w), q_0', q_1'), \tau \in M\}$
  $\cup \{((w,\tau),(w',q')) \mid \tau = (q, \sigma, q_0', q_1')$ *and*
  $((w' = w0$ *and* $q' = q_0')$ *or* $(w' = w1$ *and* $q' = q_1'))\}$;
- $c'(w, q) = c(q)$ *if* $q \in S$;
- $c'(w, \tau) = 0$ *if* $\tau \in M$.



Theorem 3. [Emptiness Game] *The language of a parity tree automaton* $\mathcal{A} = (S, s_0, M, c)$ *is non-empty iff Player 0 wins the parity game* $\mathcal{G}_{\mathcal{A},t} = (V_0, V_1, E, c')$ *from position* s_0*.*

- $V_0 = S$;
- $V_1 = M$;

$\Big\} \leftarrow V$ *is finite!*

- $E = \{(q,\tau) \mid \tau = (q,1,q'_0,q'_1), \tau \in M\}$
  $\cup \{(\tau, q') \mid \tau = (q,1,q'_0,q'_1)$ *and*
  $(q' = q'_0$ *or* $q' = q'_1)\}$;
- $c'(q) = c(q)$ *for* $q \in S$;
- $c(\tau) = 0$ *for* $\tau \in M$.