Bernd Finkbeiner Date: January 8, 2013

Automata, Games, and Verification: Lecture 11

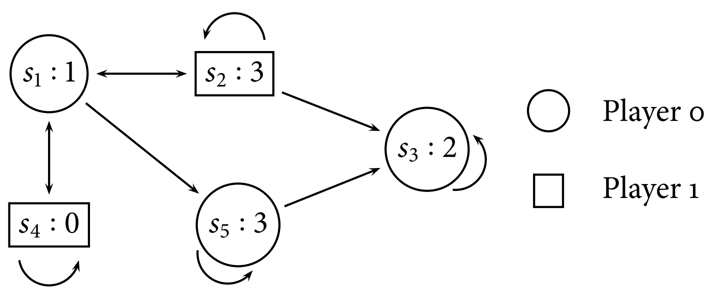
15 Parity Games

Definition 1 *A* parity game $\mathcal{G} = (\mathcal{A}, c)$ *consists of an arena* $\mathcal{A}$ *and a coloring function* $c : V \to \{0, \ldots, k\}$*. Player 0 wins play* π *if* $\max\{c(q) \mid q \in In(\pi)\}$ *is even, otherwise Player 1 wins.*

Assumptions:

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

Example:



Player 0 wins from s_1, s_3, s_4, and s_5.

Theorem 1 *Parity games are memoryless determined.*

Proof:

Induction on k:

- $k = 0$: $W_0 = V, W_1 = \emptyset$. Memoryless winning strategy: fix arbitrary order on V. $f_0(p) = \min\{q \mid (p, q) \in E\}$.
- $k + 1$:
  - If $k + 1$ is even, consider player $\sigma = 0$, otherwise $\sigma = 1$.
  - Let $W_{1-\sigma}$ be the set of positions where Player $(1 - \sigma)$ has a memoryless winning strategy. We show that Player σ has a memoryless winning strategy from $V \setminus W_{1-\sigma}$.
  - Consider subgame $\mathcal{G}'$:
    * $V'_0 = V_0 \setminus W_{1-\sigma}$;
    * $V'_1 = V_1 \setminus W_{1-\sigma}$;
    * $E' = E \cap (V' \times V')$;
    * $c'(p) = c(p)$ for all $p \in V'$.

- $\mathcal{G}'$ is still a game:
  - for $p \in V'_\sigma$, there is a $q \in V \setminus W_{1-\sigma}$ with $(p,q) \in E'$, otherwise $p \in W_{1-\sigma}$;
  - for $p \in V'_{1-\sigma}$, for all $q \in V$ with $(p,q) \in E$, $q \in V \setminus W_{1-\sigma}$, hence there is a $q \in V'$ with $(p,q) \in E$.
- Let $C'_i = \{p \in V' \mid c'(p) = i\}$.
- Let $Y = Attr'_\sigma(C'_{k+1})$. ($Attr'$: Attractor set on $\mathcal{G}'$)
- Let f_A be the attractor strategy on $\mathcal{G}'$ into C'_{k+1}.
- Consider subgame $\mathcal{G}''$:
  - $V''_0 = V'_0 \setminus Y$;
  - $V''_1 = V_1 \setminus Y$;
  - $E'' = E' \cap (V'' \times V'')$;
  - $c'' : V'' \to \{0, \ldots, k\}; c''(p) = c'(p)$ for all $p \in V''$.
- $\mathcal{G}''$ is still a game.
- Induction hypothesis: $\mathcal{G}''$ is memoryless determined.
- Also: $W''_{1-\sigma} = \emptyset$ (because $W''_{1-\sigma} \subseteq W_{1-\sigma}$: assume Player $(1-\sigma)$ had a winning strategy from some position in V''. Then this strategy would win in $\mathcal{G}$, too, since Player σ has no chance to leave $\mathcal{G}''$ other than to $W_{1-\sigma}$.)
- Hence, there is a winning memoryless winning strategy f_{IH} for player σ from V''.
- We define:

$$f_\sigma(p) = \begin{cases} f_{IH}(p) & \text{if } p \in V''; \\ f_A(p) & \text{if } p \in Y \setminus C'_{k+1}; \\ \text{min. successor in } V \setminus W_{1-\sigma} & \text{if } p \in Y \cap C'_{k+1}; \\ \text{min. successor in } V & \text{otherwise.} \end{cases}$$

- f_σ is winning for Player σ on $V \setminus W_{1-\sigma}$.
  Consider a play that conforms to f_σ:
  - Case 1: Y is visited infinitely often.
    $\Rightarrow$ Player σ wins (inf. often even color $k+1$).
  - Case 2: Eventually only positions in V'' are visited.
    $\Rightarrow$ Since Player σ follows f_{IH}, Player σ wins.

■

McNaughton's Algorithm

McNaughton's algorithm solves finite parity games based on the ideas from the proof.

McNaughton$(\mathcal{G})$

1. m := highest color in $\mathcal{G}$
2. <u>if</u> $m = 0$ or $V = \emptyset$
   <u>then</u> <u>return</u> $(V, \emptyset)$
3. set σ to m mod 2

4. set $W_{1-\sigma}$ to $\varnothing$

5. repeat

   (a) $\mathcal{G}' := \mathcal{G} \setminus Attr_\sigma(c^{-1}(m))$

   (b) $(W_0', W_1') := McNaughton(\mathcal{G}')$

   (c) if $(W_{1-\sigma}' = \varnothing)$ then

      i. $W_\sigma := V \setminus W_{1-\sigma}$

      ii. return (W_0, W_1)

   (d) $W_{1-\sigma} := W_{1-\sigma} \cup Attr_{(1-\sigma)}(W_{1-\sigma}')$

   (e) $\mathcal{G} := \mathcal{G} \setminus Attr_{(1-\sigma)}(W_{1-\sigma}')$

Example: We apply McNaughton's algorithm to the example shown earlier.

- $m = 3$, $c^{-1}(3) = \{s_2, s_5\}$, $\sigma = 1$
- $\mathcal{G}' := \mathcal{G} \setminus Attr_1(\{s_2, s_5\}, \mathcal{G}) = \{s_1, s_4, s_3\}$
- $(\{s_3\}, \{s_1, s_4\}) = McNaughton(\mathcal{G}')$
- $W_0 := \varnothing \cup Attr_0(\{s_3\}, \mathcal{G}) = \{s_1, s_3, s_5\}$
- $\mathcal{G} := \mathcal{G} \setminus Attr_0(\{s_3\}, \mathcal{G}) = \{s_2, s_4\}$
- $\mathcal{G}' := \mathcal{G} \setminus Attr_1(\{s_2\}, \mathcal{G}) = \{s_4\}$
- $(\{s_4\}, \varnothing) = McNaughton(\mathcal{G}')$
- $W_0 = \{s_1, s_3, s_5\} \cup \{s_4\}$
- $\mathcal{G} := \mathcal{G} \setminus \{s_4\} = \{s_2\}$
- $\mathcal{G}' = \mathcal{G} \setminus Attr_1(\{s_2\}, \mathcal{G}) = \varnothing$
- $(\varnothing, \varnothing) = McNaughton(\mathcal{G}')$
- $W_1 = V \setminus \{s_1, s_3, s_4, s_5\} = \{s_2\}$
- return $(\{s_1, s_3, s_4, s_5\}, \{s_2\})$

✚

16 Tree Automata

Binary Tree: $T = \{0, 1\}^*$.
Notation: T_Σ : set of all binary Σ-trees

Definition 2 *A tree automaton (over binary Σ-trees) is a tuple $\mathcal{A} = (S, s_0, M, \varphi)$:*

- *S: finite set of states*
- $s_0 \in S$

- $M \subseteq S \times \Sigma \times S \times S$
- *φ: acceptance condition (Büchi, parity, …)*

Definition 3 *A run of a tree automaton $\mathcal{A}$ on a Σ-tree v is a S-tree (T, r), s.t.*

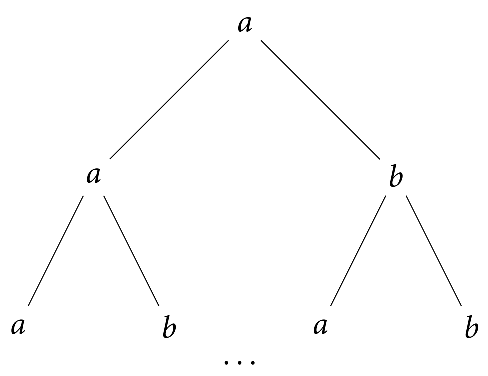
- $r(\varepsilon) = s_0$
- $(r(q), v(q), r(q0), r(q1)) \in M$ *for all* $q \in \{0,1\}^*$

Definition 4 *A run is accepting if every branch is accepting (by φ). A Σ-tree is accepted if there exists an accepting run.*
$\mathcal{L}(A)$:= set of accepted Σ-trees.

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

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

Σ-tree:



run: