Bernd Finkbeiner Date: January 22, 2013

Automata, Games, and Verification: Lecture 13

Theorem 1 *For each parity tree automaton $\mathcal{A}$ over Σ there is a parity tree automaton $\mathcal{A}'$ with $\mathcal{L}(\mathcal{A}') = T_\Sigma - \mathcal{L}(\mathcal{A})$.*

Proof:

- $\mathcal{A}$ does *not* accept some tree t iff Player 1 has a winning memoryless strategy f in $\mathcal{G}_{\mathcal{A},t}$ from (ε, s_0)
- Strategy
$$f : \{0,1\}^* \times M \to \{0,1\}^* \times S$$
can be represented as
$$f' : \{0,1\}^* \times M \to \{0,1\}$$
(where $f(u,(q,\sigma,q'_0,q'_1)) = (u \cdot i, q'_i)$ iff $f'(u,\tau) = i$).
- f' is isomorphic to
$$g : \{0,1\}^* \to (M \to \{0,1\})$$
($M \to \{0,1\}$ is the finite "local strategy")
- Hence, $\mathcal{A}$ does not accept t iff

  (1) there is a $(M \to \{0,1\})$-tree ν such that
  (2) for all $i_0, i_1, i_2, \ldots \in \{0,1\}^\omega$
  (3) for all $\tau_0, \tau_1, \ldots \in M^\omega$
  (4) <u>if</u>
  - for all j,
    $\tau_j = (q, a, q'_0, q'_1)$
    $\Rightarrow a = t(i_0, i_1, \ldots, i_j)$ and
  - $i_0 i_1 \ldots = \nu(\varepsilon)(\tau_0)\nu(i_0)(\tau_1)\ldots$

  <u>then</u> the generated state sequence $q_0 q_1 \ldots$
  with $q_0 = s_0$, $(q_j, a, q^0, q^1) = \tau_j$,
  $q_{j+1} = q^{\nu(i_0,\ldots,i_{j-1})(\tau_j)}$ for all j
  violates c.
- Condition **(4)** is a property of words over
$$\Sigma' = \underbrace{(M \to \{0,1\})}_{\nu} \times \underbrace{\Sigma}_{t} \times \underbrace{M}_{\tau} \times \underbrace{\{0,1\}}_{i}$$
and can be checked by a parity word automaton $\mathcal{A}_4 = (S_4, \{s_4\}, T_4, c_4)$:

- $S_4 = S \cup \{\bot\}$;
- $s_4 = s_0$;
- $T_4 = \{(q, (f, a, (q, a, q'_0, q'_1), i), q'_i) \mid q \in S, f : M \to \{0,1\},$
  $(q, a, q'_0, q'_1) \in M, i = f(q, a, q'_0, q'_1)\}$
  $\cup\ \{(q, (f, a, (q, a', q'_0, q'_1), i), \bot) \mid a \neq a' \text{ or } i \neq f(q, a', q'_0, q'_1)\}$
  $\cup\ \{(\bot, a, \bot) \mid a \in \Sigma'\}$;
- $c_4(q) = c(q) + 1$ for $q \in S$;
- $c_4(\bot) = 0$.

- Condition **(3)** is a property of words $(M \to \{0,1\}) \times \Sigma \times \{0,1\}$ which results from **(4)** by universal quantification (= complement; project; complement) $\Rightarrow$ there is a deterministic parity word automaton $\mathcal{A}_3$ that checks **(3)**.
- Condition **(2)** defines a property of $(M \to \{0,1\}) \times \Sigma$-trees. It can be checked by a tree automaton $\mathcal{A}_2 = (S_2, s_2, M_2, c_2)$, simulating $\mathcal{A}_3$ along each path:
  - $S_2 = S_3$;
  - $s_2 = s_3$;
  - $M_2 = \{(q, (f, a), q'_0, q'_1) \mid (q, (f, a, 0), q'_0) \in T_3, (q, (f, a, 1), q'_1) \in T_3\}$;
  - $c_2 = c_1$.
- Condition **(1)** is a property on Σ-trees: Use nondeterminism to guess $M \to \{0,1\}$ label: $\mathcal{A}_1 = (S_1, s_1, M_1, c_1)$, where
  - $S_1 = S_2$;
  - $s_1 = s_2$;
  - $M_1 = \{(q, a, q'_0, q'_1) \mid \exists f : M \to \{0,1\}.(q, (f, a), q'_0, q'_1) \in M_2\}$;
  - $c_1 = c_2$.

■

18 Monadic Second-Order Theory of Two Successors (S2S)

Syntax:

- first-order variable set $V_1 = \{x_0, x_1, \ldots\}$
- second-order variable set $V_2 = \{X_0, X_1, \ldots\}$
- Terms t:

$$t ::= \varepsilon \mid x \mid t0 \mid t1$$

- Formulas φ:

$$\varphi ::= t \in X \mid t_1 = t_2 \mid \neg\varphi \mid \varphi_0 \vee \varphi_1 \mid \exists x.\varphi \mid \exists X.\varphi$$

Semantics:

- first-order valuation $\sigma_1 : V_1 \to \mathbb{B}^*$

- second-order valuation $\sigma_2 : V_2 \to 2^{\mathbb{B}^*}$

Semantics of terms:

- $[\![\varepsilon]\!] = \varepsilon$
- $[\![x]\!]_{\sigma_1} = \sigma_1(x)$
- $[\![t0]\!]_{\sigma_1} = [\![t]\!]_{\sigma_1} 0$
- $[\![t1]\!]_{\sigma_1} = [\![t]\!]_{\sigma_1} 1$

Semantics of formulas:

- $\sigma_1, \sigma_2 \vDash t \in X$ iff $[\![t]\!]_{\sigma_1} \in \sigma_2(X)$
- $\sigma_1, \sigma_2 \vDash t_1 = t_2$ iff $[\![t_1]\!]_{\sigma_1} = [\![t_2]\!]_{\sigma_1}$
- $\sigma_1, \sigma_2 \vDash \neg\varphi$ iff $\sigma_1, \sigma_2 \nvDash \varphi$
- $\sigma_1, \sigma_2 \vDash \varphi_0 \vee \varphi_1$ iff $\sigma_1, \sigma_2 \vDash \varphi_0$ or $\sigma_1, \sigma_2 \vDash \varphi_1$
- $\sigma_1, \sigma_2 \vDash \exists x_i.\varphi$ iff there is a $a \in \mathbb{B}^*$ s.t.

$$\sigma_1'(y) = \begin{cases} \sigma_1(y) & \text{if} x \neq y, \\ a & \text{otherwise;} \end{cases}$$

  and $\sigma_1', \sigma_2 \vDash \varphi$
- $\sigma_1, \sigma_2 \vDash \exists X_i.\varphi$ iff there is a $A \subseteq \mathbb{B}^*$ s.t.

$$\sigma_2'(Y) = \begin{cases} \sigma_2(Y) & \text{if} X \neq Y \\ A & \text{otherwise;} \end{cases}$$

  and $\sigma_1, \sigma_2' \vDash \varphi$

Examples:

- "node x is a prefix of node y"

$$x \leq y \quad \Leftrightarrow \quad \forall X.((y \in X \wedge \forall z(z0 \in X \Rightarrow z \in X) \wedge \forall z.(z1 \in X \Rightarrow z \in X)) \Rightarrow x \in X)$$

- "X is linearly ordered by $\leq$"

$$\mathrm{Chain}(X) \quad \Leftrightarrow \quad \forall x.\forall y.((x \in X \wedge y \in X) \Rightarrow (x \leq y \vee y \leq x))$$

- "X is a path"

$$\begin{aligned} \mathrm{Path}(X) &\Leftrightarrow \mathrm{Chain}(X) \wedge \neg\exists Y.\ (X \subseteq Y \wedge X \neq Y \wedge \mathrm{Chain}(Y)) \\ X \subseteq Y &\Leftrightarrow \forall z.(z \in X \Rightarrow z \in Y) \\ X = Y &\Leftrightarrow X \subseteq Y \wedge Y \subseteq X \end{aligned}$$

- "X is infinite"

$$\mathrm{Inf}(X) \quad \Leftrightarrow \quad \exists Y.(Y \neq \varnothing \wedge \forall y \in Y.\exists y' \in Y.\exists x' \in X.\ (y < y' \wedge y < x'))$$

Theorem 2 *For each Muller tree automaton $\mathcal{A} = (S, s_0, M, \mathcal{F})$ over $\Sigma = 2^{V_2}$ there is a S2S formula φ over V_2 s.t. $t \in \mathcal{L}(\mathcal{A})$ iff $\sigma_2 \vDash \varphi$ where $\sigma_2(P) = \{q \in \{0,1\}^* \mid P \in t(q)\}$.*

Proof:

Use $\overline{R} = (R_q)_{q \in S}$ to encode the run tree.

$$
\begin{aligned}
\varphi &\Leftrightarrow \exists \overline{R}.(\mathrm{Part} \wedge \mathrm{Init} \wedge \mathrm{Trans} \wedge \mathrm{Accept}) \\
\mathrm{Part} &\Leftrightarrow \forall x. \bigvee_{q \in S} \mathrm{State}_q(x) \\
\mathrm{State}_q(x) &\Leftrightarrow x \in R_q \wedge \bigwedge_{q' \in S \setminus \{q\}} \neg(x \in R_{q'}) \\
\mathrm{Init} &\Leftrightarrow \mathrm{State}_{s_0}(\varepsilon) \\
\mathrm{Trans} &\Leftrightarrow \forall x. \bigvee_{(q,A,q_0',q_1') \in M} (\mathrm{State}_q(x) \wedge (\bigwedge_{V \in A} x \in V \wedge \bigwedge_{V \notin A} \neg(x \in V)) \wedge \\
&\quad \wedge \mathrm{State}_{q_0'}(x0) \wedge \mathrm{State}_{q_1'}(x1)) \\
\mathrm{InfOcc}_q(P) &\Leftrightarrow \exists Q.(Q \subseteq P \wedge Q \subseteq R_q \wedge \mathrm{Inf}(Q)) \\
\mathrm{Muller}(P) &\Leftrightarrow \bigvee_{F \in \mathcal{F}} (\bigwedge_{q \in F} \mathrm{InfOcc}_q(P) \wedge \bigwedge_{q \notin F} \neg \mathrm{InfOcc}_q(P)) \\
\mathrm{Accept} &\Leftrightarrow \forall P.(\mathrm{Path}(P) \Rightarrow \mathrm{Muller}(P))
\end{aligned}
$$

■

Theorem 3 *For every S2S formula φ over V_1, V_2 there is a Muller tree automaton $\mathcal{A}$ over $\Sigma = 2^{V_1 \cup V_2}$ such that $t \in \mathcal{L}(\mathcal{A})$ iff $\sigma_1, \sigma_2 \vDash \varphi$ where*

$$
\begin{aligned}
\sigma_1(x) &= q \text{ iff } x \in t(q); \\
\sigma_2(X) &= \{q \in \{0,1\}^* \mid X \in t(q)\}.
\end{aligned}
$$

Proof:

First, we rewrite S2S formulas to a normal form, for which we only have the following types of equalities:

$$x = \varepsilon,\ x = y0,\ x = y1,\ x \in Y,\ x = y$$

Next we inductively translate S2S formulas to tree automata. (Analogous to the proof for S1S in Lecture 8.) For example:

- $x \in Y$:
  - $S = \{q_0, q_1\}$
  - $s_0 = q_0$
  - $M = \{(q_0, A, q_0, q_1) \mid x \notin A\}$
    $\cup \{(q_0, A, q_1, q_0) \mid x \notin A\}$
    $\cup \{(q_0, A, q_1, q_1) \mid x \in A, Y \in A\}$
    $\cup \{(q_1, A, q_1, q_1) \mid x \notin A\}$
  - $\mathcal{F} = \{\{q_1\}\}$

■