Bernd Finkbeiner Date: December 4, 2012

Automata, Games, and Verification: Lecture 8

Definition 1 *For a S1S formula* φ, $\mathcal{L}(\varphi) = \{\alpha_{\sigma_1,\sigma_2} \in (2^{V_1 \cup V_2})^\omega \mid \sigma_1, \sigma_2 \models \varphi\}$, *where* $x \in \alpha(j)$ *iff* $j = \sigma_1(x)$, *and* $X \in \alpha(j)$ *iff* $j \in \sigma_2(X)$.

Definition 2 *A language L is LTL/QPTL/S1S-definable if there is a LTL/QPTL/S1S formula* φ *with* $\mathcal{L}(\varphi) = L$.

Theorem 1 *Every QPTL-definable language is S1S-definable.*

Proof:

For every QPTL-formula φ over AP and every S1S-term t over $V_1 = \emptyset$, we define a S1S formula $T(\varphi, t)$ over $V_2 = AP$ such that, for all $\alpha \in (2^{AP})^\omega$,

$$\alpha[[t]_{\sigma_1}..] \models_{\text{QPTL}} \varphi \qquad \text{iff} \qquad \sigma_1, \sigma_2 \models_{\text{S1S}} T(\varphi, t),$$

where $\sigma_2 : P \mapsto \{i \in \omega \mid P \in \alpha(i)\}$.

- $T(P, t) = t \in P$, for $P \in AP$;
- $T(\neg\varphi, t) = \neg T(\varphi, t)$;
- $T(\varphi \vee \psi, t) = T(\varphi, t) \vee T(\psi, t)$
- $T(\mathrm{X}\varphi, t) = T(\varphi, S(t))$
- $T(\varphi\, \mathcal{U}\, \psi, t) = \exists y.(y \geq t \wedge T(\psi, y) \wedge \neg\exists z.(t \leq z < y \wedge T(\neg\varphi, z)))$
- $T(\exists P\, \varphi, t) = \exists P.\ T(\varphi, t)$.

$\mathcal{L}(\varphi) = \mathcal{L}(T(\varphi, 0))$. ■

Theorem 2 *Every S1S-definable language is Büchi-recognizable.*

Proof:

Let φ be a S1S-formula.

1. Rewrite φ into normal form
   $\varphi ::= 0 \in X \mid x \in Y \mid x = 0 \mid x = y \mid x = S(y) \mid$
   $\neg\varphi \mid \varphi \vee \psi \mid \exists x.\ \varphi \mid \exists X.\ \varphi$.
   using the following rewrite rules:

$$\begin{aligned} S(t) \in X &\mapsto \exists y.\ y = S(t) \wedge y \in X \\ S(t) = S(t') &\mapsto t = t' \\ S(t) = x &\mapsto x = S(t) \\ t = S(S(t')) &\mapsto \exists y.\ y = S(t') \wedge t = S(y) \end{aligned}$$

2. Rename bound variables to obtain unique variables.

   Example:

$$\exists x.(S(S(y)) = x \land \exists x\ (S(x) \in X_0))$$

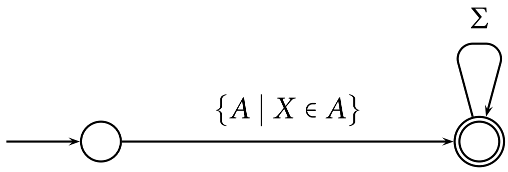
   is rewritten to

$$\exists x_0.\ \exists x_1.x_0 = S(x_1) \land x_1 = S(y) \land \exists x_2 \exists x_3.x_3 = S(x_2) \land x_3 \in X_0$$

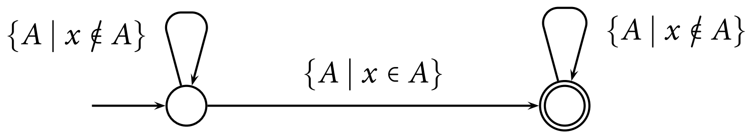
3. Construct Büchi automaton:

   Base cases:

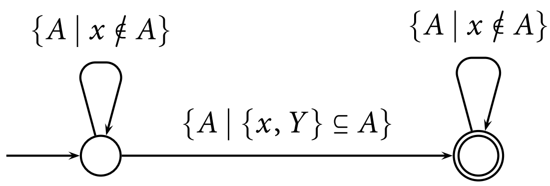
   - $0 \in X$:



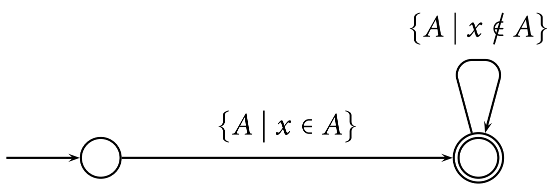
   For every $x \in V_1$, intersect with $\mathcal{A}_x$:



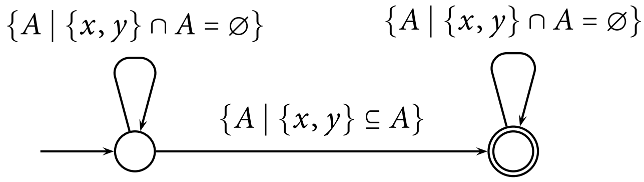
   - $x \in Y$:



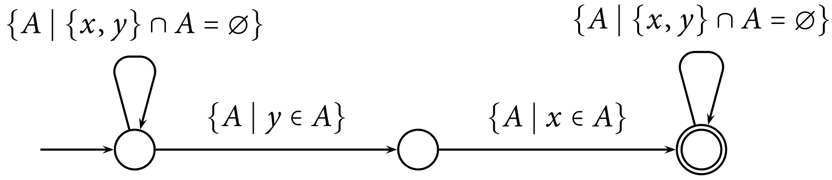
   - $x = 0$:



- $x = y$:



- $x = S(y)$:



Inductive step:

- $\varphi \vee \psi$: language union,
- $\neg\varphi$: complement and intersection with all $\mathcal{A}_x$,
- $\exists x.\ \varphi$, $\exists X.\ \varphi$: projection

■

10 Weak Monadic Second-Order Theory of One Successor (WS1S)

Syntax: same as S1S;

Semantics: same as S1S; except:
$\sigma_1, \sigma_2 \models \exists X.\varphi$ iff there is a **finite** $A \subseteq \omega$ s.t.

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

and $\sigma_1, \sigma_2' \models \varphi$.

Theorem 3 *A language is WS1S-definable iff it is S1S-definable.*

Proof:

($\Rightarrow$): Quantifier relativization:

$$\begin{array}{rcl} \forall X \ldots & \mapsto & \forall X.\ \mathrm{Fin}(X) \rightarrow \ldots \\ \exists X \ldots & \mapsto & \exists X.\ \mathrm{Fin}(X) \wedge \ldots \end{array}$$

(⇐):

- Let φ be an S1S-formula.
- Let $\mathcal{A}$ be a Büchi automaton with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$.
- Let $\mathcal{A}'$ be a deterministic Muller automaton with $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$.
- By the characterization of deterministic Muller languages, $\mathcal{L}(\mathcal{A}')$ is a boolean combination of languages $\overrightarrow{W}$, where W is finite-word recognizable.
- For a finite-word language W, recognizable by a finite automaton $\mathcal{A} = (S, I, T, F)$, where $S = \{s_1, s_2, \ldots, s_n\}$, we define a WS1S formula $\psi_W(y)$ over $V_2 = AP \cup \{At_{s_1}, \ldots, At_{s_n}\}$ that defines the words whose prefix up to position y is in W:

$$\begin{aligned}\psi_W(y) \;:=\; & \exists At_{s_1}, \ldots, At_{s_n} \,. \\ & \bigvee_{s \in I} 0 \in At_s \\ & \wedge\; \forall x < y \left(\bigvee_{(s_i, A, s_j) \in T} \Big(x \in At_{s_i} \wedge S(x) \in At_{s_j} \wedge \bigwedge_{P \in A} x \in P \wedge \bigwedge_{P \in AP \setminus A} x \notin P \Big) \right) \\ & \wedge\; \forall x \leq y \left(\bigwedge_{i \neq j} \neg \Big(x \in At_{s_i} \wedge x \in At_{s_j} \Big) \right) \\ & \wedge\; \bigvee_{s_i \in F} y \in At_{s_i}\end{aligned}$$

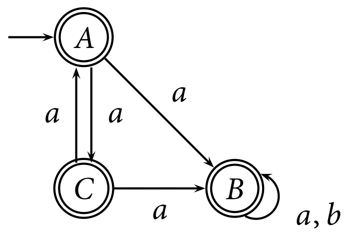
- then, the WS1S formula $\varphi_W := \forall x.\, \exists y.\, (x < y \wedge \psi(y))$ defines the words in $\overrightarrow{W}$.
- Hence, $\mathcal{L}(\varphi)$ is WS1S-definable.

■

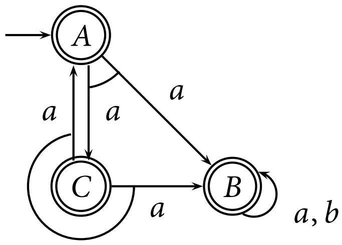
11 Alternating Automata

Example:

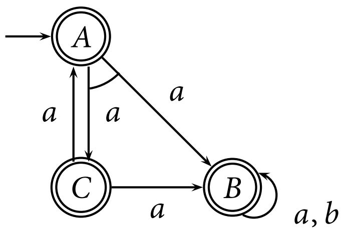
- Nondeterministic automaton, $L = a(a+b)^\omega$, disjunctive branching mode:



- universal automaton, $L = a^\omega$, conjunctive branching mode:



- Alternating automaton, both branching modes (arc between edges indicates universal branching mode), $L = aa(a+b)^\omega$



✚

Definition 3 *The* positive Boolean formulas *over a set X, denoted* $\mathbb{B}^+(X)$*, are the formulas built from elements of X, conjunction* $\wedge$*, disjunction* $\vee$*, true and false.*

Definition 4 *A set* $Y \subseteq X$ satisfies *a formula* $\varphi \in B^+(X)$*, denoted* $Y \vDash \varphi$*, iff the truth assignment that assigns true to the members of Y and false to the members of* $X \setminus Y$ *satisfies* φ*.*

Definition 5 *An* alternating Büchi automaton *is a tuple* $\mathcal{A} = (S, s_0, \delta, F)$*, where:*

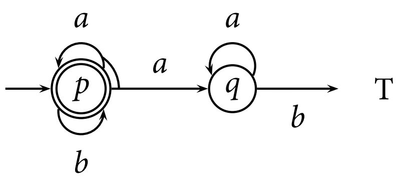
- *S is a finite set of states,*
- $s_0 \in S$ *is the initial state,*
- $F \subseteq S$ *is the set of accepting states, and*
- $\delta : S \times \Sigma \to \mathbb{B}^+(S)$ *is the transition function.*

A tree T over a set of *directions* D is a prefix-closed subset of D^*. The empty sequence ε is called the *root*. The children of a node $n \in T$ are the nodes children$(n) = \{n \cdot d \in T \mid d \in D\}$. A Σ-labeled tree is a pair (T, l), where $l : T \to \Sigma$ is the labeling function.

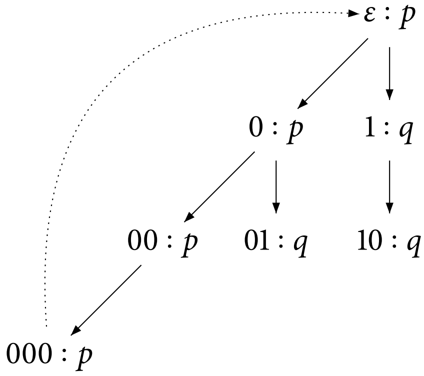
Definition 6 *A* run *of an alternating automaton on a word* $\alpha \in \Sigma^\omega$ *is an S-labeled* tree $\langle T, r\rangle$ *with the following properties:*

- $r(\varepsilon) = s_0$ *and*
- *for all* $n \in T$*, if* $r(n) = s$*, then* $\{r(n') \mid n' \in children(n)\}$ *satisfies* $\delta(s, \alpha(|n|))$*.*

Example: $L = (\{a, b\}^* \, b)^\omega$



$S = \{p, q\}$
$F = \{p\}$
$\delta(p, a) = p \wedge q$
$\delta(p, b) = p$
$\delta(q, a) = q$
$\delta(q, b) = \mathrm{T}$
example word $w = (aab)^\omega$ produces this run:



(The dotted line means that the same tree would repeat there. Note that, in general, an alternating automaton may also have more than one run on a particular word—or no run at all.)

Definition 7 *A* branch *of a tree T is a maximal sequence of words $n_0 n_1 n_2 \ldots$ such that $n_0 = \varepsilon$ and n_{i+1} is a child of n_i for $i \geq 0$.*

Definition 8 *A run (T, r) is* accepting *iff, for every infinite branch $n_0 n_1 n_2 \ldots$,*

$$In(r(n_0)r(n_1)r(n_2)\ldots) \cap F \neq \varnothing.$$