Bernd Finkbeiner Date: May 24, 2011

**Automata, Games and Verification: Lecture 7**

---

# 9 Quantified Propositional Temporal Logic (Cont'd)

**Example:** $L = (\emptyset\emptyset)^*\{p\}^\omega$ is QPTL-definable:

$\exists q.\ (\neg q \wedge \square(q \leftrightarrow \bigcirc\neg q) \wedge \square(p \rightarrow \bigcirc p) \wedge \square(\bigcirc p \rightarrow p \vee q) \wedge \diamond p)$ ✚

**Theorem 1** *For every Büchi automaton $\mathcal{A}$ over $\Sigma = 2^{AP}$ there exists a QPTL formula $\varphi$ such that models$(\varphi) = \mathcal{L}(\mathcal{A})$.*

**Proof:**

Let $S = \{s_1, s_2, \ldots, s_n\}$ and $AP' = AP \cup \{at_{s_1}, \ldots, at_{s_n}\}$.

$$\begin{aligned}
\varphi \ := \ \exists at_{s_1}, \ldots, at_{s_n} \quad & . \bigvee_{s \in I} at_s \\
& \wedge \ \square \left( \bigvee_{(s_i, A, s_j) \in T} at_{s_i} \wedge \bigcirc at_{s_j} \wedge \left( \bigwedge_{p \in A} p \right) \wedge \left( \bigwedge_{p \in AP \smallsetminus A} \neg p \right) \right) \\
& \wedge \ \square \left( \bigwedge_{i \neq j} \neg(at_{s_i} \wedge at_{s_j}) \right) \\
& \wedge \ \square \diamond \bigvee_{s_i \in F} at_{s_i}
\end{aligned}$$

■

# 10 Monadic Second-Order Theory of One Successor (S1S)

**Syntax:**

- first-order variable set $V_1 = \{x, y, \ldots\}$
- second-order variable set $V_2 = \{X, Y, \ldots\}$
- Terms $t$:

$$t ::= 0 \mid x \mid S(t)$$

- Formulas $\varphi$:

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

**Abbreviations:**

- $\forall X.\ \varphi \ \ := \ \neg\exists X.\ \neg\varphi$;
- $x \notin Y \ := \ \neg(x \in Y)$;
- $x \neq y \ := \ \neg(x = y)$.

**Semantics:**

- first-order valuation $\sigma_1 : V_1 \to \omega$
- second-order valuation $\sigma_2 : V_2 \to 2^\omega$

Semantics of terms:

- $[0]_{\sigma_1} = 0$
- $[x]_{\sigma_1} = \sigma_1(x)$
- $[S(t)_{\sigma_1}] = [t]_{\sigma_1} + 1$

Semantics of formulas:

- $\sigma_1, \sigma_2 \models t \in X$ iff $[t]_{\sigma_1} \in \sigma_2(X)$
- $\sigma_1, \sigma_2 \models t_1 = t_2$ iff $[t_1]_{\sigma_1} = [t_2]_{\sigma_1}$
- $\sigma_1, \sigma_2 \models \neg\psi$ iff $\sigma_1, \sigma_2 \not\models \psi$
- $\sigma_1, \sigma_2 \models \psi_0 \vee \psi_1$ iff $\sigma_1, \sigma_2 \models \psi_0$ or $\sigma_1, \sigma_2 \models \psi_1$
- $\sigma_1, \sigma_2 \models \exists x.\ \varphi$ iff there is an $a \in \omega$ s.t.

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

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

- $\sigma_1, \sigma_2 \models \exists X.\ \varphi$ iff there is an $A \subseteq \omega$ s.t.

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

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

**Example:**

$$\begin{array}{ll}
X \subseteq Y & :\equiv\ \forall z.\ (z \in X \to z \in Y); \\
X = Y & :\equiv\ X \subseteq Y \wedge Y \subseteq X; \\
\mathit{Suff}(X) & :\equiv\ \forall y.\ (y \in X \to S(y) \in X); \\
x \leq y & :\equiv\ \forall Z.\ (x \in Z \wedge \mathit{Suff}(Z)) \to y \in Z; \\
\mathit{Fin}(X) & :\equiv\ \exists Y.\ (X \subseteq Y \wedge \exists z.\ z \notin Y \wedge \forall z.\ (z \notin Y \to S(z) \notin Y));
\end{array}$$

✚

**Definition 1** *For a S1S formula $\varphi$, models($\varphi$) $= \{\alpha_{\sigma_1,\sigma_2} \in (2^{V_1 \cup V_2})^\omega \mid \sigma_1, \sigma_2 \models \varphi\}$, 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 $\varphi$ with models($\varphi$) $= L$.*

**Theorem 2** *Every QPTL-definable language is S1S-definable.*

**Proof:**

For every QPTL-formula $\varphi$ 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 $P \in AP$;
- $T(\neg\varphi, t) = \neg T(\varphi, t)$;
- $T(\varphi \vee \psi, t) = T(\varphi, t) \vee T(\psi, t)$
- $T(\bigcirc\varphi, t) = T(\varphi, S(t))$
- $T(\varphi \,\mathcal{U}\, \psi, t) = \exists y.(y \geq t \wedge T(\psi, y) \wedge \neg\exists z.(t \leq z < y \wedge T(\neg\varphi, z)))$
- $T(\exists P\ \varphi, t) = \exists P.\ T(\varphi, t)$.

models($\varphi$) = models($T(\varphi, 0)$). ■

**Theorem 3** *Every S1S-definable language is Büchi-recognizable.*

**Proof:**

Let $\varphi$ be a S1S-formula.

1. Rewrite $\varphi$ 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:

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

2. Rename bound variables to obtain unique variables.

**Example:**

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

is rewritten to

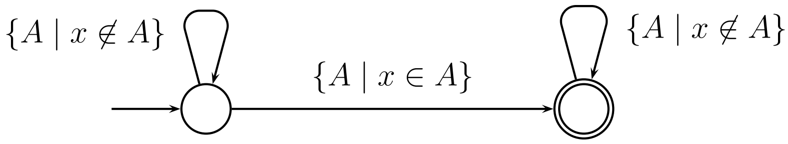
$$\exists x_0.\ \exists x_1.x_0 = S(x_1) \wedge x_1 = S(y) \wedge \exists x_2 \exists x_3.x_3 = S(x_2) \wedge x_3 \in X_0$$

3. Construct Büchi automaton:

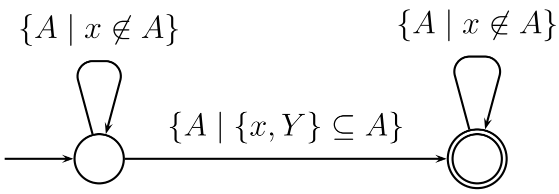
   Base cases:

   - $0 \in X$:

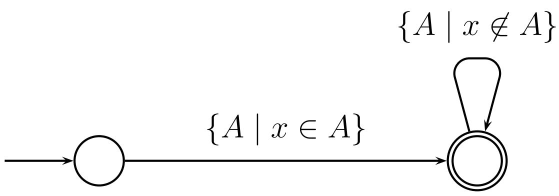

For every $x \in V_1$, intersect with $\mathcal{A}_x$:



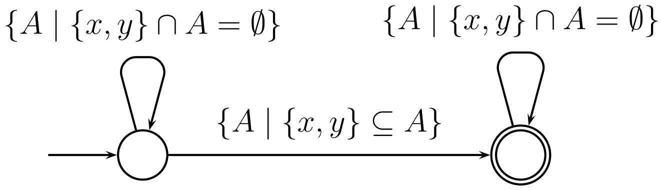

   - $x \in Y$:



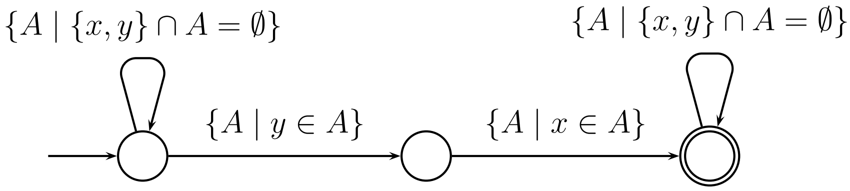

   - $x = 0$:



- $x = y$:




- $x = S(y)$:




Inductive step:

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

■

# 11 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 4** *A language is WS1S-definable iff it is S1S-definable.*

**Proof:**

($\Rightarrow$): Quantifier relativization:

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

(⇐):

- Let $\varphi$ be an S1S-formula.
- Let $\mathcal{A}$ be a Büchi automaton with $\mathcal{L}(\mathcal{A}) = \text{models}(\varphi)$.
- 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, models($\varphi$) is WS1S-definable.

■