Other logics

François Laroussinie

29 octobre 2024

draft !

1 S1S

The Monadic Second-Order Logic of one successor ($\mathsf{S1S}$) is an extension of $\mathsf{FO}$with quantifications over sets. Formulas are interpreted over a discrete linear structure with a successor function : $(\mathbb{N}, \mathsf{succ})$. Many elements of this description comes from the slides of Luke Ong (Oxford Univ.) for the TACL summer school, this topics is also addressed in the survey of Vardi and Wilke ("Automata : From Logic to Algorithms").

1.1 Syntax and semantics

The syntax of $\mathsf{S1S}$ formulas is as follows :

$$\begin{array}{l} \mathsf{S1S} \ni \varphi, \psi ::= t \in X \mid \varphi \vee \psi \mid \neg\varphi \mid \ \mathbf{E}x.\varphi \mid \ \mathbf{E}X.\varphi \\ t ::= x \mid \mathsf{succ}(t) \end{array}$$

We use lowercase letters $(x, y, z \ldots)$ to denote first-order variables, and capital letter $(X, Y, Z, \ldots)$ for second-order variables. The universal quantifications is $\neg\exists\neg$.

We write $\varphi(x_1, \ldots, x_m, X1, \ldots, X_n)$ when $x_1, \ldots, x_m$ are free first-order variables and $X_1, \ldots, X_n$ are free second-order variables in φ. A model for $\varphi(x_1, \ldots, x_m, X1, \ldots, X_n)$ is m values $a_i \in \mathbb{N}$ with $1 \le i \le m$ and n values $P_j \subseteq \mathbb{N}$ with $1 \le j \le n$: $a_1, \ldots, a_m, P_1, \ldots, P_n \models \varphi(x_1, \ldots, x_m, X1, \ldots, X_n)$ if the valuation $(x_i \mapsto a_i)_{1\le i\le m}$ and $(X_j \mapsto P_j)_{1\le j\le n}$ satisfies φ.

1.2 Examples of formulas

Here are some useful formulas to illustrate the expressive power of $\mathsf{S1S}$:

— "$x = y$" : $\forall X.(x \in X \ \Leftrightarrow\ y \in X)$

— "$x = 0$" : $\neg\exists y.(x = \mathsf{succ}(y)$

— "$x \le y$" : $\forall X.\Big((x \in X \ \wedge\ \forall z.(z \in X \ \Rightarrow\ \mathsf{succ}(y) \in X)) \ \Rightarrow\ y \in X\Big)$

— "X is finite" : $\exists x.\forall y \ (y \in X \ \Rightarrow\ y \le x)$

A language over the alphabet 2^{AP} with $\mathsf{AP} = \{a_1, \ldots, a_n\}$ is defined with a $\mathsf{S1S}$ formula $\varphi(X_1, \ldots, X_n)$ where X_i describes the set of positions where a_i is true. For example, the models where a is true at every even position can be expressed with :

$$\varphi(X_a) \ = \ \exists X_e.\Big(0 \in X_e \ \wedge\ \forall y.(y \in X_e \ \Leftrightarrow\ \mathsf{succ}(y) \notin X_e) \ \wedge\ \forall y.(y \in X_e \ \Rightarrow\ y \in X_a)\Big)$$

1.3 Büchi automata and S1S

In the automata (or LTL) point of view, we consider infinite words over 2^{AP}, that is $w \in (2^{\mathsf{AP}})^\omega$. With S1S, we consider n predicates P_i, that is a model is in $(2^\omega)^n$. Both points of view are equivalent : a $w \in (2^{\mathsf{AP}})^\omega$ describes the predicates P_i^w with $1 \leq i \leq n$ s.t. $P_i^w = \{j \in \mathbb{N} \mid a_i \in w_j\}$. And a set of n predicates $\mathcal{P} = \{P_1, \ldots, P_n\}$ describes the word w s.t. $w_i^{\mathcal{P}} = \{a_j \in \mathsf{AP} \mid i \in P_j\}$.
In the following we write $\mathcal{L}(\varphi(X_1, \ldots, X_n))$ to denote the set $\{w \in (2^{\mathsf{AP}})^\omega \mid P_1^w, \ldots, P_n^w \models \varphi(X_1, \ldots, X_n)$.

We say that a language $\mathcal{L} \subseteq (2^{\mathsf{AP}})^\omega$ is S1S-definable iff there exists an S1S formula $\varphi(X_1, \ldots, X_n)$ s.t. $\mathcal{L} = \mathcal{L}(\varphi(X_1, \ldots, X_n))$.

S1S and (non-deterministic) Büchi automata have the same expressive power. Indeed we have the two following theorems due to Büchi :

Theorem 1. *For any Büchi automata $\mathcal{A} = (Q, Q_0, \delta, F)$ over 2^{AP} with $|\mathsf{AP}| = n$, there exists an S1S formula $\varphi_{\mathcal{A}}(X_1, \ldots, X_n)$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi_{\mathcal{A}}(X_1, \ldots, X_n))$.*

Proof. Assume $Q = \{q_1, \ldots, q_k\}$. We define φ_A as follows :

$$\varphi_{\mathcal{A}} \; = \; \exists Q_1 \ldots Q_k . \Big[\Psi_Q \; \wedge \; (\bigvee_{q_i \in Q_0} 0 \in Q_i) \; \wedge \; \Big(\forall x \, \exists y . \, (y > x \wedge \bigvee_{q_i \in F} y \in Q_i) \Big) \Big]$$

with Ψ_Q ensuring that for there is one and only one state $q_i \in Q$ associated with every position and that the labelings of two successive positions satisfy the transition function δ :

$$\Psi_Q \; = \; \forall x . \bigvee_{1 \leq i \leq k} \Big(x \in Q_i \; \wedge \; (\bigwedge_{j \neq i} x \notin Q_j) \; \wedge \; \bigvee_{\sigma \in 2^{\mathsf{AP}}} (P_\sigma(x) \; \wedge \; \bigvee_{q_j \in \delta(q_i, \sigma)} \mathsf{succ}(x) \in Q_j) \Big)$$

where $P_\sigma(x)$ stands for $\bigwedge_{a_j \in \sigma} x \in X_j \; \wedge \; \bigwedge_{a_j \notin \sigma} x \notin X_j$ □

And we can also build an automaton from a formula. In that case, we consider an S1S formula $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$ which defines a language over the alphabet 2^{AP_φ} where AP_φ is a set of $m+n$ propositions we will denote with $\{p_1, \ldots, p_m, P_1, \ldots, P_n\}$ in the following. Formally we have :

Theorem 2. *For any S1S formula $\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n)$, there exists a Büchi automata $\mathcal{A}_\varphi$ over 2^{AP_φ} s.t. $\mathcal{L}(\mathcal{A}_\varphi) = \mathcal{L}(\varphi(x_1, \ldots, x_m, X_1, \ldots, X_n))$.*

Proof. The construction of $\mathcal{A}_\varphi$ is done by induction over φ.

- Consider a formula $\varphi = \mathsf{succ}^k(x_i) \in X_j$. We need to verify that the k-th successor of the position labeled by x_i belongs to X_j. If $k = 0$, we use the automaton $\mathcal{A}_{x_i \in X_j}$ and for $k > 0$, we use the automaton $\mathcal{A}_{\mathsf{succ}^k(x_i) \in X_j}$ as described at Figure **??** (there is no need of Büchi condition here and then every state is accepting).
- If $\varphi = \varphi_1 \wedge \varphi_2$, we can build $\mathcal{A}_{\varphi_1}$ and $\mathcal{A}_{\varphi_2}$, and then apply the intersection operation over Büchi automata. And if $\varphi = \neg\psi$, we use the complement operation (with its complexity blow-up!!).

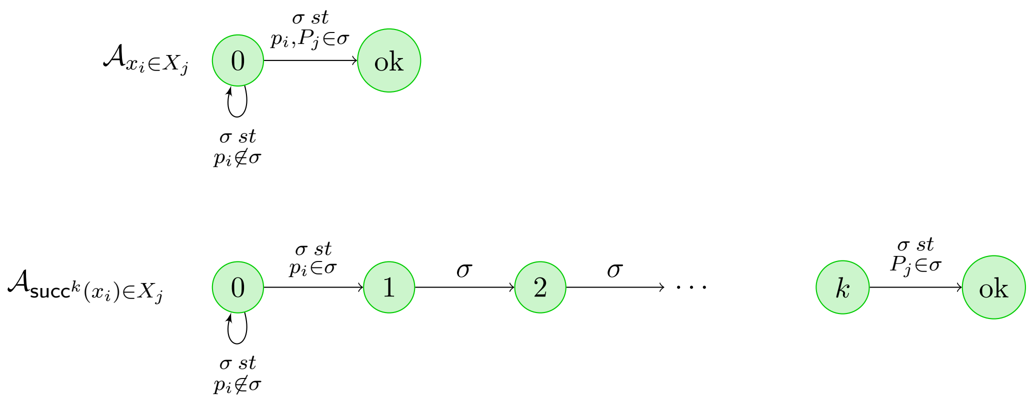

FIGURE 1 – Automata $\mathcal{A}_{x_i \in X_j}$ and $\mathcal{A}_{\mathsf{succ}^k(x_i) \in X_j}$.

— If $\varphi = \exists X_j.\psi$, we first build $\mathcal{A}_\psi = (Q, Q_0, , \delta, F)$ over 2^{AP_ψ}. Note that AP_ψ contains P_j as X_j is a free variable in ψ. More precisely we have $\mathsf{AP}_\psi = \mathsf{AP}_\varphi \cup \{P_j\}$. We then define $\mathcal{A}_{\exists X_j.\psi} = (Q, Q_0, \delta', F)$ over 2^{AP_φ}. For every $\sigma \in 2^{\mathsf{AP}_\varphi}$ we have :

$$\delta'(q, \sigma) \;=\; \delta(q, \sigma) \;\cup\; \delta(q, \sigma \cup \{P_j\})$$

Intuitively it means that a word is accepted by φ iff there a way to label it with P_j in such a way to satisfy ψ.

— If $\varphi = \exists X_j.\psi$, we proceed in a similar way as for the second-order case, but we also have to ensure that the label p_i is used exactly once in δ' (for this we use two copies of Q). Consider $\mathcal{A}_\psi = (Q, Q_0, , \delta, F)$ over 2^{AP_ψ}. The automaton for φ is (Q', Q_0^1, δ', F') with :

— $Q' = Q^1 \cup Q^2$ where every Q^i is the set Q tagged with i.

— And :

$$\begin{array}{rcl} \delta'(q^1, \sigma) & = & \{q'^1 \mid q' \in \delta(q, \sigma)\} \cup \{q'^2 \mid q' \in \delta(q, \sigma \cup \{p_i\})\} \\ \delta'(q^2, \sigma) & = & \{q'^2 \mid q' \in \delta(q, \sigma)\} \end{array}$$

— $F' = \{q^2 \mid q \in F\}$.

$\square$