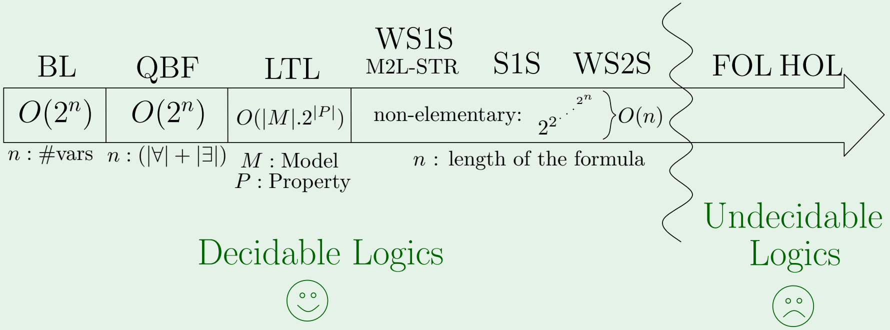
Monadic Second Order Logic

Hossein Hojjat

LARA

November 26, 2010

Quest for Expressiveness/Decidability



Second Order Logic

- FOL is the logic of quantification over the elements of a type
  - $\forall x.\Phi(x)$
    For every individual x, $\Phi(x)$
- Second order logic is the logic of quantification over the predicates
  - $\forall P.\forall x.P(x)$
    For every set of individuals P and for every individual x, $x \in P$
  - $\exists R.\forall x.R(x, x)$
    There exists a relation R such that for every individual x, $R(x, x)$

- *Monadic* Second Order (MSO): The fragment of the second order logic which allows only quantification over sets

S1S

- S1S: Monadic second order logic of one successor
- The fragment of MSO interpreted on discrete linear orders ($\leq$)

- Let $\{x_1, \cdots, x_n\}$ be a family of first-order variables and $\{X_1, \cdots, X_n\}$ a family of second-order monadic variables
- S1S is defined on the signature $(\mathbb{N}, S)$ as the following
  - $t := 0|x_i$
  - $f := S(t,t)|X_i(t)|\neg f|f \wedge f|\exists x_i.f|\exists X_i.f$
- S is the successor predicate
- The predicate S and $\leq$ can be defined from each other

- WS1S: The fragment of S1S which allows only quantification over finite sets

WS1S Semantics

- Signature: Natural numbers $\langle \mathbb{N}, S \rangle$
- Interpretation: $x \xrightarrow{I} n \in \mathbb{N}$ and $X \xrightarrow{I} N \in 2^{\mathbb{N}}$ such that N is finite
- Truth value of a formula with respect to interpretation I

$$
\begin{array}{lll}
I \models Y(x) & \text{iff} & I(x) \in I(Y) \\
I \models S(x,y) & \text{iff} & I(x) + 1 = I(y) \\
I \models \neg\Phi & \text{iff} & I \not\models \Phi \\
I \models \Phi_1 \wedge \Phi_2 & \text{iff} & I \models \Phi_1 \text{ and } I \models \Phi_2 \\
I \models \exists x.\Phi & \text{iff} & I[n/x] \models \Phi, \text{ for some } n \in \mathbb{N} \\
I \models \exists X.\Phi & \text{iff} & I[N/X] \models \Phi, \text{ for some finite } N \in 2^{\mathbb{N}}
\end{array}
$$

Word Model

- Finite alphabet Σ is given
- Word is defined as $\omega = a_0 \cdots a_{n-1}$ where $a_0, \cdots, a_{n-1} \in \Sigma$
- Domain of ω: $dom(\omega) = \{0, \cdots, |\omega| - 1\}$
- A unary predicate P_α is defined for every $\alpha \in \Sigma$ such that $P_\alpha(i)$ if and only if $a_i = \alpha$
- The word ω defines a word model $\underline{\omega} = (dom(\omega), S^\omega, P_{a_0}, \cdots, P_{a_{n-1}})$

Example

Let $\Sigma = \{a, b\}$ and $\omega = aabba$
$dom(\omega) = \{0, 1, 2, 3, 4\}$
$P_a = \{0, 1, 4\}$
$P_b = \{2, 3\}$

MSO on Words

- Given an alphabet Σ, the logic S1S can also be defined on the signature of the words: $\{\leq, (P_\alpha)_{\alpha \in \Sigma}\}$ or $\{S, (P_\alpha)_{\alpha \in \Sigma}\}$

- $\exists x \exists y . P_a(x) \land P_b(y) \land x \leq y \land \neg \exists z.(x < z \land z < y)$
- Word contains the substring *ab*

- $\exists x . P_a(x) \land \neg \exists y.(x < y)$
- The last symbol is *a*: $P_a(last)$

- $\exists X.(X(first) \land \forall y \forall z.(S(y,z) \rightarrow (X(y) \leftrightarrow \neg X(z))) \land \neg X(last))$
- The length of the word is even

MSO_0

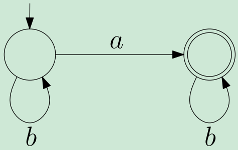
- We can check a set X to see if it is singleton
  $Sing(X) \equiv \exists Y.Y \subseteq X \wedge Y \neq X \wedge \neg(\exists Z.Z \subseteq Y \wedge Z \neq Y)$
  $(X = Y) \equiv X \subseteq Y \wedge Y \subseteq X$
- We can remove all the first-order variables if we allow S and $\leq$ to be applied to singleton sets
- The result belongs to MSO_0

$$\Phi ::= X \subseteq Y | S(X, Y) | \exists X.\Phi | \neg\Phi | \Phi_1 \wedge \Phi_2$$

- $\Phi_{MSO} = \forall x \forall y.(P_a(x) \wedge x < y \rightarrow P_b(y))$
- $\Phi_{MSO_0} = \forall X \forall Y.(Sing(X) \wedge Sing(Y) \wedge X \subseteq P_a \wedge X < Y \rightarrow Y \subseteq P_b)$

Büchi Theorems

- A language $L \subseteq \Sigma*$ is regular if and only if it is expressible in weak monadic second-order logic on words



$\forall x \forall y.(P_a(x) \wedge x < y \rightarrow P_b(y))$

- A language $L \subseteq \Sigma^\omega$ is ω-regular if and only if it is expressible in monadic second-order logic on words

Proof

A language $L \subseteq \Sigma*$ is regular if and only if it is expressible in weak monadic second-order logic on words

Automata $\Rightarrow$ Logic

- Code the execution of an automaton
- A formula with a structure similar to the following
  $\exists X_0 \cdots \exists X_n . \Phi_{\text{partition}} \wedge \Phi_{\text{start}} \wedge \Phi_{\text{transitions}} \Phi_{\text{accept}}$

Logic $\Rightarrow$ Automata

- Construction based on induction on the structure of Φ
- $X_1 \subseteq X_2$, $X_1 \subseteq P_a$, $Sing(X_1)$, $S(X_1, X_2)$, $X_1 < X_2$

WS1S decidability

Decision Procedure

- Given a WS1S formula Φ
- Translate $\neg\Phi$ to an automaton $A_{\neg\Phi} = (Q, \Sigma, \delta, q_0, F)$ accepting ω iff $\omega \models \neg\Phi$
- Output
  - Φ is valid when $A_{\neg\Phi}$ accepts the empty string
  - Return a counter model ω which belongs to the automaton

From Logic to Automaton: Alphabet

- The MSO_0 formula $\Phi(X_1, \cdots, X_n)$ is interpreted in the word model of ω and the sets $K_1, \cdots, K_n$
- $K_i \in \mathrm{dom}(\omega)$ represents a set of positions
- To code the models we use an alphabet $\Sigma \times \{0, 1\}^n$

Let $\Sigma = \{a, b\}$

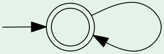
$$\begin{array}{r} \omega \longrightarrow \\ K_1 \longrightarrow \\ K_2 \longrightarrow \\ K_3 \longrightarrow \end{array} \begin{bmatrix} a \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} b \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} b \\ 1 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} a \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\omega = abba \qquad \begin{aligned} K_1 &= \{2, 3\} \\ K_2 &= \emptyset \\ K_3 &= \{0, 3\} \end{aligned}$$

From Logic to Automaton: Base case

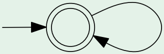
- $\# \in \Sigma$ is an arbitrary symbol
- $* \in \{0,1\}^{n-2}$ is an arbitrary vector

$X_1 \subseteq X_2$



$$\begin{bmatrix} \# \\ 0 \\ 0 \\ * \end{bmatrix}, \begin{bmatrix} \# \\ 0 \\ 1 \\ * \end{bmatrix}, \begin{bmatrix} \# \\ 1 \\ 1 \\ * \end{bmatrix}$$

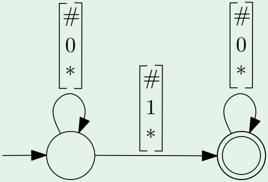
$X_1 \subseteq P_a$



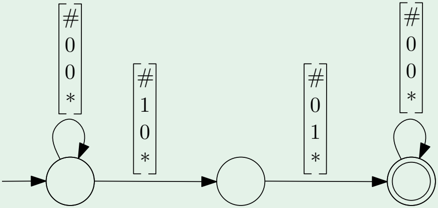
$$\begin{bmatrix} \# \\ 0 \\ * \end{bmatrix}, \begin{bmatrix} a \\ 1 \\ * \end{bmatrix}$$

From Logic to Automaton: Base case

$Sing(X_1)$

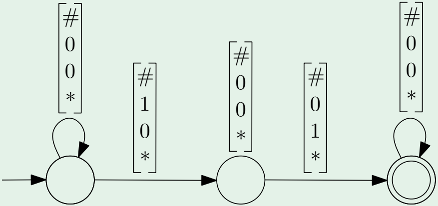


$S(X_1, X_2)$



From Logic to Automaton: Base case

$(X_1 < X_2)$



From Logic to Automaton: Step case

$\neg\Phi$

- Complement the automaton A_Φ by flipping the final and non-final states
- $L(\neg\Phi) = \overline{L(\Phi)} = \overline{L(A_\Phi)} = L(A_{\neg\Phi})$

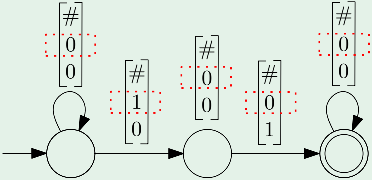
$\Phi_1 \wedge \Phi_2$

- Product construction of A_{Φ_1} and A_{Φ_2}
- $L(\Phi_1 \wedge \Phi_2) = L(\Phi_1) \cap L(\Phi_2) = L(A_{\Phi_1}) \cap L(A_{\Phi_2}) = L(A_{\Phi_1 \wedge \Phi_2})$

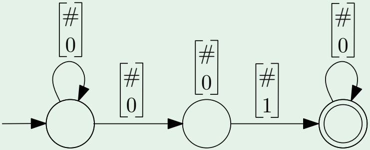
From Logic to Automaton: Step case

$\exists X_i.\Phi$

- $A_{\exists X_i.\Phi}$ acts as A_Φ except that it guesses the values in the set X_i
- Projection on X_i by simply removing its track from the automaton



$\Phi(X_1, X_2) = X_1 < X_2$

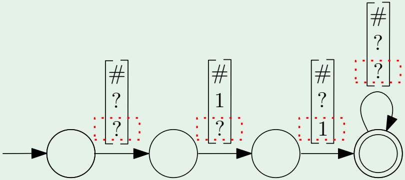


$\exists X_1.\Phi(X_1, X_2)$

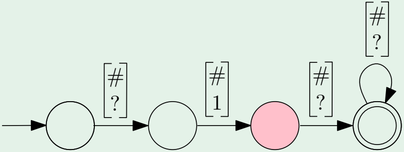
From Logic to Automaton: Step case

$\exists X_i.\Phi$

- We should be careful of $(0^{n-1})*$ suffix after projection from $\Phi(X_1, \cdots, X_n)$ to $\exists X_i.\Phi(X_1, \cdots, X_n)$



$\Phi(X_1, X_2) = X_1(1) \wedge X_2(2)$



$\exists X_2.\Phi(X_1, X_2)$

Doesn't include $X_1 = \{1\}$

From Logic to Automaton: Step case

$\exists X_i.\Phi$

- Right quotient of $L \subseteq \Sigma*$ with $L' \subseteq \Sigma*$
  $L/L' = \{\omega \in \Sigma * | \exists u \in L'.\omega u \in L\}$
- Define the projection function $\Pi_i : (\{0,1\}^n)* \rightarrow (\{0,1\}^{n-1})*$ such that $\Pi_i(\begin{pmatrix} b_1 \\ \vdots \\ b_{i-1} \\ b_i \\ b_{i+1} \\ \vdots \\ b_n \end{pmatrix}) = \begin{pmatrix} b_1 \\ \vdots \\ b_{i-1} \\ b_{i+1} \\ \vdots \\ b_n \end{pmatrix}$
- $L(\exists X_i \Phi) = \Pi_i(L(\Phi))/(\{0\}^{n-1}) = \Pi_i(L(A_\Phi))/(\{0\}^{n-1}) = L(A_{\exists X_i \Phi})$

Translation: Summary

- Correspondence between logical operators and basic automata
- Constructive proof using induction on the formula structure
- Construction results in a trivial DFA that accepts all the acceptable words
- Shows that WS1S formulas define regular languages
- Give a decision procedure for WS1S

State Explosion

- Negation requires determinization
- Existential quantification introduces non-determinism
- Quantifier alternation results in exponential blow-ups
  - $\forall X.\exists Y.\Phi \equiv \neg\exists X.\neg\exists Y.\Phi$
  - If $|A_\Phi| = n$ then $|A_{\neg\exists Y.\Phi}| = O(2^{|n|})$
  - $|A_{\neg\exists X.\neg\exists Y.\Phi}| = O(2^{2^{|n|}})$

Corollary

- Presburger arithmetic is decidable
- We can translate a given formula in Presburger arithmetic to its equivalent in MSO logic
- Idea of encoding:
  - Encode $n \in \mathbb{N}$ as the set of positions in which there is a 1 in its binary representation
    $17 = (10001)_2 \rightsquigarrow \{0, 4\}$
  - Encode the addition of $x_1 \in \mathbb{N}$ and $x_2 \in \mathbb{N}$ as the MSO formula
    $\exists X_{Result}.\exists X_{Carry}.\Phi(X_1, X_2, X_{Result}, X_{Carry})$
  - X_1, X_2, X_{Result} and X_{Carry} represents the bits of x_1, x_2, result and carry during addition

M2L-STR

- Interpretation with respect to k
- Domain is $[k] = \{0, \cdots, k\}$
- Successor relation S restricted to $[k] \times [k]$

Semantics

$k, I \models Y(x)$	iff	$I(x) \in I(Y)$, $I(x) \in [k]$ and $I(X) \subseteq [k]$
$k, I \models S(x, y)$	iff	$I(x) + 1 = I(y)$, $I(y) \in [k]$
$k, I \models \neg\Phi$	iff	$I \not\models \Phi$
$k, I \models \Phi_1 \wedge \Phi_2$	iff	$I \models \Phi_1$ and $I \models \Phi_2$
$k, I \models \exists x.\Phi$	iff	$I[n/x] \models \Phi$, for some $n \in [k]$
$k, I \models \exists X.\Phi$	iff	$I[N/X] \models \Phi$, for some $N \subseteq [k]$

Validity

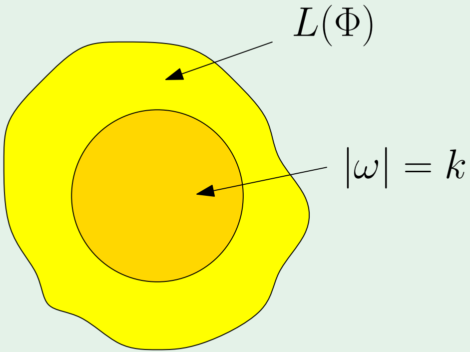
$\models \Phi$ if and only if $k, I \models \Phi$, for all I and $k \in \mathbb{N}$

Satisfiability Examples

	WS1S	M2L-STR
$X \subseteq Y$	$X \mapsto \{1\}, Y \mapsto \{1,2\}$	$k = 3, X \mapsto \{1\}, Y \mapsto \{1,2\}$
$\exists X . \forall p . p \in X$	unsatisfiable	valid $\forall k \in \mathbb{N} . X \mapsto \{0, \cdots, k-1\}$
$\exists X . \exists p . p \in X$	valid	satisfiable for $k > 0$ unsatisfiable for $k = 0$

Bounded Model Construction

- INSTANCE: A formula Φ and $k \in \mathbb{N}$
- QUESTION: Is there ω such that $|\omega| = k$ and ω satisfies Φ?



BMC for M2L-STR

$$\Phi ::= X \subseteq Y | S(X, Y) | \exists X.\Phi | \neg\Phi | \Phi_1 \wedge \Phi_2$$

- Encode $M \subseteq [k]$ by the Booleans $b_0, \cdots, b_{k-1}$ so that $i \in M$ iff b_i is true
- Translation from MSO to QBF with $\lceil . \rceil_k$: MSO $\rightarrow$ QBF

$$\begin{array}{lcl}
\lceil X \subseteq Y \rceil_k & = & \bigwedge_{0 \leq i \leq k-1} (x_i \rightarrow y_i) \\
\lceil S(X, Y) \rceil_k & = & Sing(x_0, \cdots, x_{k-1}) \wedge Sing(y_0, \cdots, y_{k-1}) \wedge \\
 & & \bigvee_{0 \leq i \leq k-1} (x_i \rightarrow y_{i+1}) \\
\lceil \Phi_1 \wedge \Phi_2 \rceil_k & = & \lceil \Phi_1 \rceil_k \wedge \lceil \Phi_2 \rceil_k \\
\lceil \neg\Phi \rceil_k & = & \neg \lceil \Phi \rceil_k \\
\lceil \exists X.\Phi \rceil_k & = & \exists x_0 \cdots x_{k-1}.\lceil \Phi \rceil_k
\end{array}$$

BMC for WS1S

Theorem

- BMC for WS1S is non-elementary
- Proof.
  - Closed formulas are either valid or unsatisfiable
  - Closed formula Φ has a model of length k iff Φ is valid
  - Validity in WS1S is non-elementary

Reference

- "Bounded Model Construction for Monadic Second-Order Logics"; Ayari, Basin - CAV 2000
- "Languages, Automata, and Logic"; Wolfgang Thomas, Chapter 7 of Handbook of Formal Languages vol. 3, 1997