CSC2428: Automata and Logic
Lecture #4: Feb. 3, 2005

Lecturer: Prof. Leonid Libkin
Scribe Notes by: Yuan An

February 4, 2005

1 Decidability of WS1S

The Weak Second-order theory of One Successor ($WS1S$) consists of the structure $(\mathbb{N}, succ)$, where $succ$ is the successor function $succ(x) = x+1$, and MSO formulas in which the second-order quantifiers range over finite sets. We have seen the application of $WS1S$ which is the decidability of the FO theory of $(\mathbb{N}, +)$.

Now we will construct an automaton when given a $WS1S$ formula such that the formula is true iff the automaton accepts some string corresponding to the formula.

Suppose we have a $WS1S$ formula $\phi(X_1, X_2, .., X_n)$ such that $X_i \subseteq_{finite} \mathbb{N}$. Let $S[\overrightarrow{X}]$ denote a string over $(\{0,1\}^n)^*$. We translate $X_1, .., X_n$ to $S[\overrightarrow{X}]$ as follows: translate X_i into a string over $\{0,1\}^n$ by putting 1 to the string if a position $k \in \mathbb{N}$ is in X_i, otherwise putting 0. See Figure 1.

Now we can construct an automaton $\mathcal{A}_\phi$ such that $\phi(X_1, .., X_n)$ is true iff $\mathcal{A}_\phi$ accepts $S[\overrightarrow{X}]$. How does it work?

First of all, if ϕ is a sentence, then ϕ is true iff $\mathcal{A}_\phi$ accepts the empty string, that is decidable.

Suppose ϕ is a formula in $MSO_0(MSO = MSO_0)$, then the construction is pretty much like that in *Büchi* theorem. Induction on formulas: the basis is atomic formulas; induction is on $\vee$ and $\neg$. The only case needs to check is the existential quantifier.

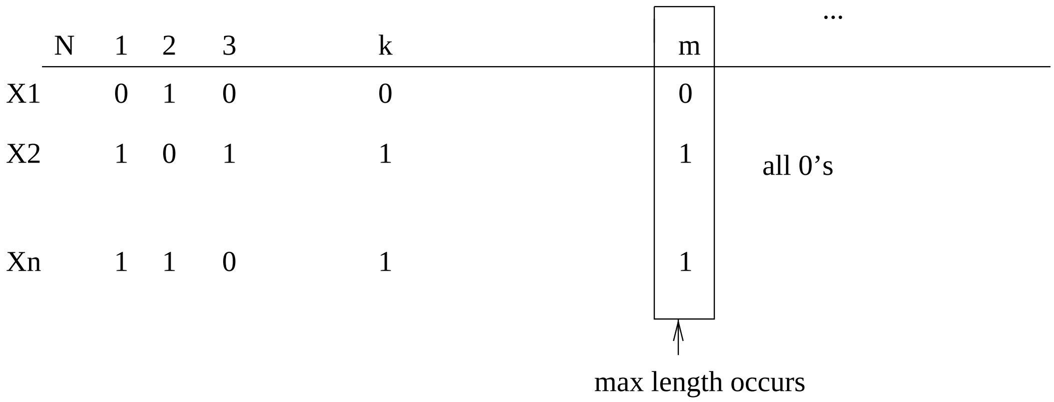


Figure 1: String translation.

Suppose $\phi(X_1,..,X_n) = \exists X_{n+1}\psi(X_1,..,X_{n+1})$. As in *Büchi* theorem, we guess values of X_{n+1} and apply $\mathcal{A}_\psi$, where $\mathcal{A}_\psi$ is the automaton constructed form the formula ψ. However, carefulness must be taken because the length of X_{n+1} may exceed the maximum length of X_i $i = 1,..,n$. Let m be the length of X_i $i = 1,..,n$ and let l be length of X_{n+1}, when $\mathcal{A}_\phi$ reaches m at state q, then look at $\mathcal{A}_\psi$ to see whether it can reach its final state q_f from q with $l-m$ input strings of the form $\begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 0 \end{bmatrix}$.

If it does, then make q the final state of $\mathcal{A}_\phi$.

Therefore, $WS1S$ is decidable; however, the complexity is non-elementary.

2 MSO over Graphs

We have known that MSO is equivalent to regular string and tree languages. We also have known that MSO is decidable and $MSO = EMSO$ over strings and trees. Then what happens if we just look at graphs?

Given a graph $G = (V, E)$, we can say that G is connected in MSO.

The following formula states that G is not connected.

$$\exists X \left(\begin{array}{c} \exists x X(x) \wedge \neg \overline{X}(x),\ X, \overline{X} \neq \emptyset \\ \forall x \forall y (X(x) \wedge \neg X(y)) \rightarrow \neg E(x,y). \end{array} \right)$$

Negating the above formula gives us the formula that G is connected. The negation is in the form of $\forall X \phi$, where $\phi \in FO$. Therefore, connectivity is in universal MSO.

Theorem 2.1 (Fagin 1974) *Graph connectivity is not definable in $EMSO$.*

The corollary is that over graphs $MSO \neq EMSO$.

What is about decidability? Let us consider a simple type of graph – Grids.

A grid is a structure $([0,..,n-1]\times[0,..,n-1], \prec_h, \prec_v)$ consisting the universe $[0,..,n-1]\times[0,..,n-1]$ and two predecessors: a horizontal predecessor $\prec_h$ and a vertical predecessor $\prec_v$, such that $(i,j) \prec_h (i+1,j)$ and $(i,j) \prec_v (i,j+1)$.

Proposition 2.1 *MSO theory of finite grids is undecidable.*

Undecidable problem: given MSO sentence ϕ in vocabulary $(\prec_h, \prec_v)$, is there a finite grid G such that $G \models \phi$?

Proof Sketch. Consider the undecidable problem: *Tiling problem.*

Given a set of T of tiles and two relations: $H, V \subseteq T \times T$, a tiling of $n \times m$ is a map $f : [0,..,n-1] \times [0,..,m-1] \rightarrow T$ such that $(f(i,j), f(i+1,j)) \in H$ and $(f(i,j), f(i,j+1)) \in V$. The undecidable problem is:

$\forall n, m$, is there is a tiling of $n \times m$?

($\exists n, m$, is there a tiling of $n \times m$?)

We can reduce the *tiling problem* to the grid satisfiability problem of MSO sentence by translating a (T, H, V) tiling system to a MSO sentence $\phi_{T,H,V}$ such that

$G_{n,m} \models \phi_{T,H,V} \iff$ there is no tiling of $G_{n,m}$, where $G_{n,m}$ is a grid.

Suppose $T = \{T_1, .., T_l\}$, the translation looks like:

$$\exists X_1 \exists X_2 .. \exists X_l \left(\begin{array}{c} if\, X_1, ..X_l\, partition\, the\, universe\, [0, .., n-1] \times [0, .., m-1] \rightarrow \\ \exists i, j \left(\begin{array}{c} (i,j) \in X_{k_1}, (i, j+1) \in X_{k_2}, (i+1, j) \in X_{k_3} \rightarrow \\ either\, k_1, k_3\, violate\, the\, H\, condition \\ or\, k_1, k_2\, violate\, the\, V\, condition. \end{array} \right) \end{array} \right)$$

□

Theorem 2.2 (Fagin 1974) *$ESO = NP$.*

Move to monadic: $3-colorability \in EMSO$; however, $connectivity \notin Monadic NP$.

Is Hamiltonicity in MSO ($EMSO$)? the answer is no.

Proof Sketch. Let $K_{n,m}$ be a complete bipartite graph between two partitions V_1 and V_2 such that V_1 has n vertices and V_2 has m vertices. We can check that $K_{3,3}$ is a Hamiltonian graph, but $K_{2,3}$ is not. In fact , we can prove that $K_{n,m}$ is a Hamiltonian graph iff $n = m$.

Suppose Hamiltonicity is definable by MSO sentence ϕ, let us look at the structure $\mathfrak{M}_s$, where the string s is in $\{a, b\}^*$. Define a new relation $\alpha(x, y) = (P_a(x) \wedge P_b(y)) \vee (P_b(x) \wedge P_a(y))$. If s has n a's and m b's, then α defines $K_{n,m}$. If the Hamiltonicity of $K_{n,m}$ is definable in MSO, that means in MSO we can check whether the number of a's is equal to the number of b's in the string s. Then the language $L = \{s \in \{a,b\}^* | \#(a)'s = \#(b)'s\}$ is MSO definable. Hence it is regular. It contradicts the fact that L is not regular. □

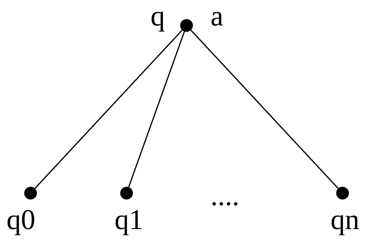
3 Unranked Tree

Definition 3.1 *An unranked tree domain is a prefix-closed subset D of $\mathbb{N}^*$ such that if $s \cdot i \in D$ and $j < i$, then $s \cdot j \in D$.*

Definition 3.2 *An unranked tree is a tuple $(D, \prec_{ch}, \prec_{sbl}, (P_a)_{a \in \Sigma})$, where D is a unranked tree domain, $\prec_{ch}$ is the child relation such that $(s, s \cdot i) \in \prec_{ch}$, $\prec_{sbl}$ is the sibling order such that $(s \cdot i, s \cdot i + 1) \in \prec_{sbl}$, and P_a is a unary predicate.*

Definition 3.3 *A top-down Unranked Non-deterministic Tree Automaton ($UNTA_\downarrow$) is a tuple $(Q, Q_0, \delta : Q \times \Sigma \rightarrow 2^{Q^*})$, where Q_0 is the initial states, the transition function δ is: $\delta(q, a) \subseteq Q^*$ where Q^* is a regular language.*

What is a run? Consider the following segment of tree.



The run indicates that $q_0, .., q_n \in \delta(q, a)$.

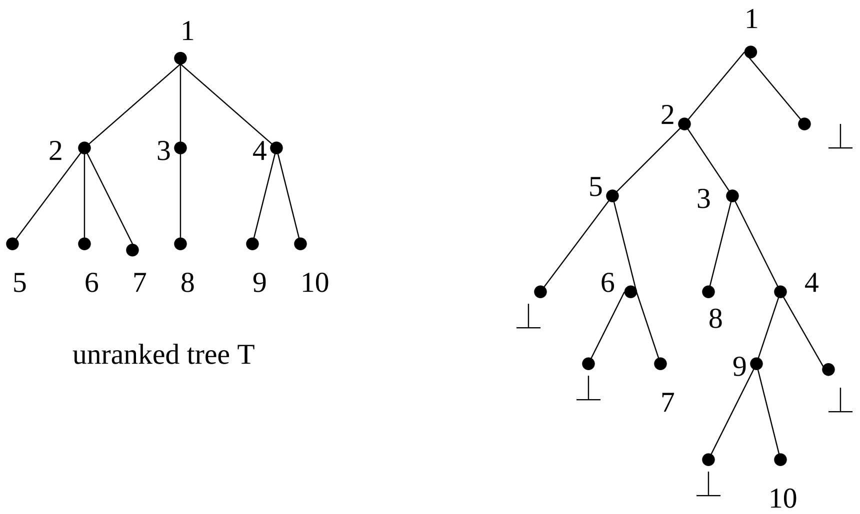
A run is successful if each leaf labeled a is assigned state q s.t. $q \in \delta(q, a)$.

Definition 3.4 *A set of unranked trees is regular iff it is accepted by a* $UNTA_{\downarrow}$.

Fact 3.1 *Over unranked trees,* $UNTA_{\downarrow}$=$UNTA_{\uparrow}$, *where* $UNTA_{\uparrow}$ *stands for bottom-up Unranked Non-deterministic Tree Automaton.*

How to translate an unranked tree T to a ranked tree T^0 (binary tree)?

The general translation rule is that for a vertex $s \in T$, make a vertex $s \in T^0$ such that the first child s_1 of s in T becomes the first child s_1 of s in T^0 and the next sibling s_2 of s in T becomes the second child s_2 of s in T^0. If the first child s_1 or the next sibling s_2 of s in T does not exist, then put the symbol $\perp$ as the corresponding child of s in T^0. For example, the unranked tree T is translated to the binary tree T' in the following diagram.



unranked tree T

binary tree T'

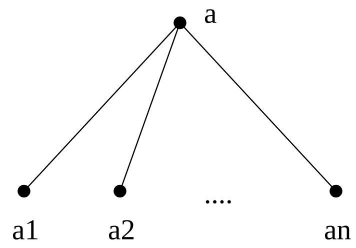
To get bake the unranked tree from a binary tree, we just look at the first child s_1 of a vertex $s \in T^0$, then get all right descendants $s'_1, .., s'_n$ of s_1 and put $s_1, s'_1, .., s'_n$ to be the children of $s \in T$.

Now we can translate an $NTA_{\downarrow}$ to an $UNTA_{\downarrow}$.

4 XML and DTD

Definition 4.1 *A DTD d over Σ is a set of pairs (a, e) such that $a \in \Sigma$ and e is a regular expression over Σ.*

A tree T satisfies a DTD d, $T \models d$, if for each pair $(a, e) \in d$, there is a subtree



such that $a_1, .., a_n \in L(e)$.

Definition 4.2 *An Extended DTD (EDTD) d' is a tuple (Σ, Σ', f, d) such that d is a DTD over Σ', $\Sigma \subseteq \Sigma'$, and f is a function $f : \Sigma' \to \Sigma$.*

A Σ labeled tree T satisfies d', $T \models d'$, iff there is a Σ' labeled tree T' such that $T' \models d$ and $T = f(T')$.

Let $SAT(d') = \{T | T \models d'\}$.

Lemma 4.1 *A set of unranked trees is of the form $SAT(d')$, where d' is an EDTD, iff it is regular.*

5 Infinite Strings

An infinite $\omega - string$ over Σ is a mapping: $\mathbb{N} \to \Sigma$.

A set of all $\omega - string$ is denoted by Σ^ω.

$\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

How to define an automaton over infinite strings?

Definition 5.1 *An automaton $\mathcal{A}$ defined over infinite strings is a tuple $(Q, Q_0, \delta : Q \times \Sigma \to 2^Q, F)$ such that $Q_0 \subseteq Q$ is a set of initial states, $F \subseteq Q$ is a set of final states.*

Definition 5.2 *A run of $\mathcal{A}$ on $S \in \Sigma^\omega$ is a function $\rho : \mathbb{N} \to Q$ such that*

- *$\rho(n+1) \in \delta(\rho(n), a)$, if the $(n+1)th$ symbol is a.*
- *$\rho(0) \in \delta(q_0, a)$ for some $q_0 \in Q_0$ if the $1st$ symbol is a.*

$s \in \Sigma^\omega$ is accepted if for some run ρ, $Inf(\rho) \cap F \neq \emptyset$, where $Inf(\rho) = \{q | |\rho^{-1}(q)| = \omega\}$.