Introduction to Tree Language Theory

Hitoshi Ohsaki

National Institute of
Advanced Industrial Science and Technology (AIST)

seminar talk (1/10)

2009

I. Words

Alphabet

A finite set $\Sigma = \{ s_0, \ldots, s_n \}$ of symbols (i.e. concrete and visibly or otherwise recognizable representations) is called an alphabet if every non-empty finite sequence over Σ is uniquely decomposed to elements in Σ

Elements in the alphabet Σ are called characters or letters

Example

Consider $\Sigma_1 = \{ 1, 11 \}$ $\Sigma_2 = \{ 01, 11 \}$ $\Sigma_3 = \{ 0, 01, 10 \}$

Σ_1 is not an alphabet since 11 is formed by either 11 or 1 (and) 1

Σ_2 is an alphabet

Σ_3 is not an alphabet, e.g., since 010 is decomposed in two ways 0 10 and 01 0

Binary & unary encoding

Consider the alphabet $\Sigma_1 = \{\, \text{a},\ \text{b},\ \text{c} \,\}$

a b c

$\Psi_1 \downarrow$ let Ψ_1 : mapping of $\text{a} \mapsto 1,\ \text{b} \mapsto 10,\ \text{c} \mapsto 100$

110100 (binary encoding over $\Sigma_2 = \{\, 0,\ 1 \,\}$)

$\Psi_2 \downarrow$ let Ψ_2 : mapping of $n \mapsto 1^{\text{decimal}(n)}$ (sequence of 1's)

$1^{\text{decimal}(110100)}$ (unary encoding over $\Sigma_3 = \{\, 1 \,\}$)

Remark

Ψ_1 : $\Psi_1^{-1}(\Psi_1(v)) = v$ & $\Psi_1(\Psi_1^{-1}(w)) = w$ if $w \in \{\, 1, 10, 100 \,\}^*$

Ψ_2 : $\Psi_2^{-1}(\Psi_2(v)) = v$ & $\Psi_2(\Psi_2^{-1}(w)) = w$

Finite automata

finite automaton $(\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$

- Σ : alphabet
- Q : finite set of state symbols
- q_0 : initial state such that $\mathsf{q}_0 \in Q$
- Q_{fin} : final states such that $Q_{\mathsf{fin}} \subseteq Q$
- Δ : finite set of transition rules with the following forms

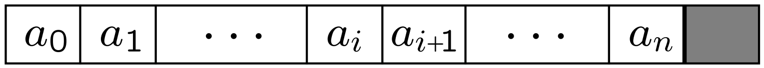
$$p \xrightarrow{a} q \qquad\qquad p \longrightarrow q \qquad\qquad (p, q \in Q,\ a \in \Sigma)$$

Note

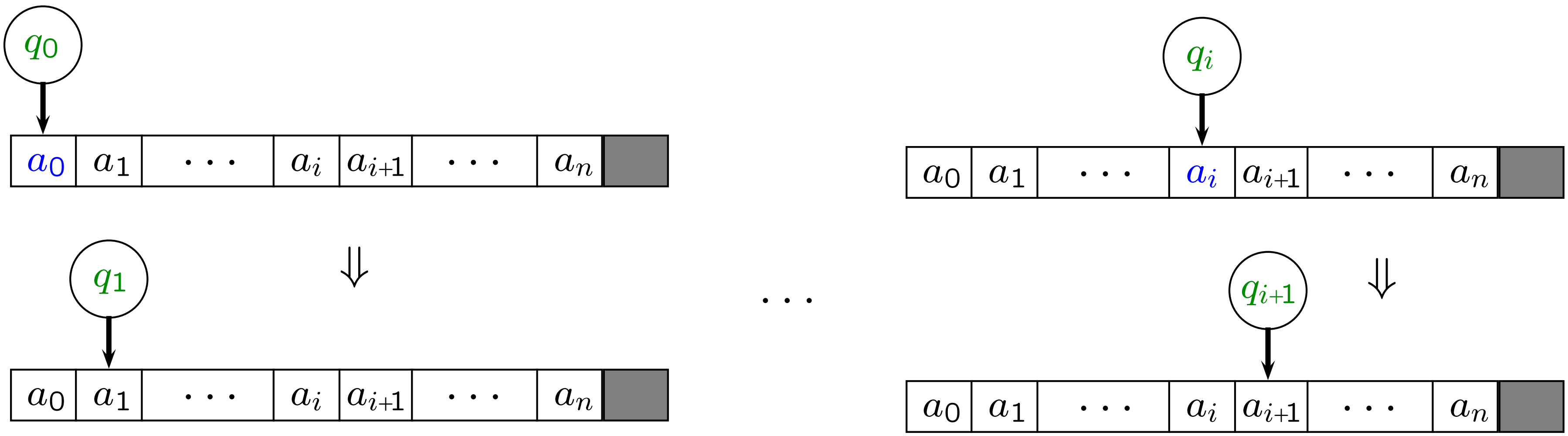
$p \xrightarrow{a} q$ means "state p moves to q reading character a"

$p \longrightarrow q$ means "state p moves to q"

Finite automata (as machines)

Given a tape 

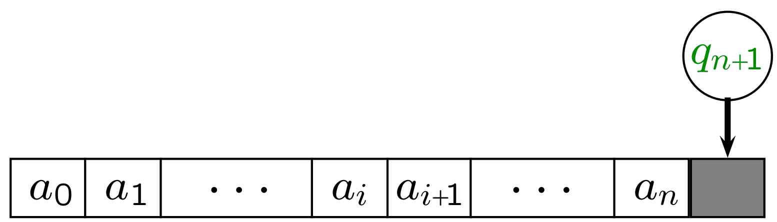
and finite automaton $\mathcal{A}$



1st step: $q_0 \xrightarrow{a_0} q_1$

such that q_0 is initial state

i-th step: $q_i \xrightarrow{a_i} q_{i+1}$

If the above move ends in 

and

q_{n+1} is a final state, we say "the input tape is accepted by $\mathcal{A}$"

Words and Languages

Given an alphabet Σ

word over Σ	$\rangle\rangle$	a finite sequence of characters from Σ
empty word ε	$\rangle\rangle$	empty sequence of character
language over Σ	$\rangle\rangle$	a subset of words over Σ
Σ^*	$\rangle\rangle$	a set of all words over Σ
Σ^+	$\rangle\rangle$	a set of all non-empty words, $\Sigma^* - \{ \varepsilon \}$

Given a finite automaton $\mathcal{A}$ with Σ

word accepted by $\mathcal{A}$	$\rangle\rangle$	word, as input tape, accepted by $\mathcal{A}$
language accepted by $\mathcal{A}$	$\rangle\rangle$	a set of words accepted by $\mathcal{A}$

For instance, we say "a language L over Σ is accepted by $\mathcal{A}$," denoted $\mathcal{L}(\mathcal{A})$, if $L = \{ w \in \Sigma^* \mid w \text{ is accepted by } \mathcal{A} \}$

Example

Define transition rules

$$\Delta_1 \;:\; q_0 \xrightarrow{a} q_0 \qquad q_0 \xrightarrow{b} q_0 \qquad q_0 \xrightarrow{c} q_0$$

$$\Delta_2 \;:\; \emptyset$$

$$\Delta_3 \;:\; q_0 \xrightarrow{a} q_0 \qquad q_1 \xrightarrow{b} q_1 \qquad q_2 \xrightarrow{c} q_2$$

$$q_0 \longrightarrow q_1 \qquad q_1 \longrightarrow q_2$$

for the finite automata with the alphabet $\Sigma = \{ a, b, c \}$. Let

$$\mathcal{A}_1 = (\Sigma, \{ q_0 \}, q_0, \{ q_0 \}, \Delta_1)$$

$$\mathcal{A}_2 = (\Sigma, \{ q_0 \}, q_0, \{ q_0 \}, \Delta_2)$$

$$\mathcal{A}_3 = (\Sigma, \{ q_0, q_1, q_2 \}, q_0, \{ q_2 \}, \Delta_3)$$

then

$$\mathcal{L}(\mathcal{A}_1) = \Sigma^* \qquad \mathcal{L}(\mathcal{A}_2) = \{ \varepsilon \} \qquad \mathcal{L}(\mathcal{A}_3) = \{ a^\ell b^m c^n \mid \ell, m, n \geqslant 0 \}$$

Deterministic finite automata (DFA)

finite automaton $(\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$ is deterministic if

1. Δ contains transition rules with the following form only:

$$p \xrightarrow{a} q$$

2. there are no transition rules $p \xrightarrow{a} q_1$ and $p \xrightarrow{a} q_2$ with $q_1 \not\equiv q_2$

Proposition

Given a finite automaton $\mathcal{A}$, one can construct a DFA $\mathcal{B}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$ i.e., DFA $\mathcal{B}$ accepts the language accepted by FA $\mathcal{A}$

Proof

Let $\mathcal{A} = (\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$, the procedure for constructing DFA consists of two steps:

(1) eliminating rules of the form $p \longrightarrow q$ from Δ

(2) eliminating non-determinisity in rules of $p \xrightarrow{a} q_1$ and $p \xrightarrow{a} q_2$ with $q_1 \not\equiv q_2$

(proof cont'd)

(1) Repeat the following computation until no more rules can be added to Δ:

if there is a pair of transition rules in Δ either $p \overset{a}{\to} q,\ q \to r$ or $p \to q,\ q \overset{a}{\to} r$, then add $p \overset{a}{\to} r$ to Δ

Note 1

The above computation terminates because the number of transition rules of the form $p \overset{a}{\to} q$ in Δ must be $|Q|^2 \times |\Sigma|$ or less, and this is the upper limit on the number of the loops.

(2) Take 2^Q, which is the *power set* of Q, and let $Q_{\mathrm{d\,fin}} = \{S \in 2^Q \mid S \cap Q_{\mathrm{fin}} \neq \varnothing\}$. If $\mathcal{A}$ accepts ε, add $\{\mathsf{q}_0\}$ to $Q_{\mathrm{d\,fin}}$. Define finite automaton $\mathcal{B} = (\Sigma, 2^Q, \{\mathsf{q}_0\}, Q_{\mathrm{d\,fin}}, \Delta_{\mathrm{d}})$ based on $\mathcal{A}$ with Δ obtained in (1), where for each two states S, T in 2^Q and character a in Σ,

$$S \xrightarrow{a} T \text{ in } \Delta_{\mathrm{d}}$$

if $T = \{\, q \mid p \in S$ and $p \xrightarrow{a} q \in \Delta \,\}$ and $T \neq \varnothing$.

Note 2

In the above construction, the reverse ($p_i \to p_{i+1},\ p_m \overset{a}{\to} q_1,\ \ q_j \to p_{j+1}$ in Δ $(1 \leqslant i < m,\ 1 \leqslant j < n)$ implies $S \xrightarrow{a} T$ in Δ_{d}) does not hold in general. Why? □

Example

Consider $\mathcal{A} = (\{\mathsf{a}, \mathsf{b}\}, \{\mathsf{p}, \mathsf{q}\}, \mathsf{p}, \{\mathsf{q}\}, \Delta)$ where

$\{\mathsf{a}, \mathsf{b}\}$: alphabet

$\{\mathsf{p}, \mathsf{q}\}$: state symbols (p: initial state, q: final state)

Δ : $\mathsf{p} \xrightarrow{\mathsf{a}} \mathsf{p} \quad \mathsf{p} \longrightarrow \mathsf{q} \quad \mathsf{q} \xrightarrow{\mathsf{b}} \mathsf{q}$

Step (1)

Add the transition rules

$\mathsf{p} \xrightarrow{\mathsf{a}} \mathsf{q}$ as $\mathsf{p} \xrightarrow{\mathsf{a}} \mathsf{p}$ & $\mathsf{p} \longrightarrow \mathsf{q}$

$\mathsf{p} \xrightarrow{\mathsf{b}} \mathsf{q}$ as $\mathsf{p} \longrightarrow \mathsf{q}$ & $\mathsf{q} \xrightarrow{\mathsf{b}} \mathsf{q}$

Since no more transition rules can be added, all epsilon rules can be eliminated

Step (2)

Define $Q_{\mathsf{d}} = \{\{\mathsf{p}\}, \{\mathsf{q}\}, \{\mathsf{p},\mathsf{q}\}\}$ * $\varnothing$ (empty set) is eliminated for optimization

and Δ_{d} to be

$\{\mathsf{p}\} \xrightarrow{\mathsf{a}} \{\mathsf{p},\mathsf{q}\} \quad \{\mathsf{p},\mathsf{q}\} \xrightarrow{\mathsf{a}} \{\mathsf{p},\mathsf{q}\} \quad \{\mathsf{p}\} \xrightarrow{\mathsf{b}} \{\mathsf{q}\} \quad \{\mathsf{q}\} \xrightarrow{\mathsf{b}} \{\mathsf{q}\} \quad \{\mathsf{p},\mathsf{q}\} \xrightarrow{\mathsf{b}} \{\mathsf{q}\}$

Since ε is accepted by $\mathcal{A}$, $\{\mathsf{p}\}$ is also the final state, so $Q_{\mathsf{d\,fin}} = \{\{\mathsf{p}\}, \{\mathsf{q}\}, \{\mathsf{p},\mathsf{q}\}\}$

Let $\mathcal{A}_{\mathsf{d}} = (\{\mathsf{a},\mathsf{b}\}, Q_{\mathsf{d}}, \{\mathsf{p}\}, Q_{\mathsf{d\,fin}}, \Delta_{\mathsf{d}})$, then one can show that $\mathcal{L}(\mathcal{A}_{\mathsf{d}}) = \mathcal{L}(\mathcal{A})$

Closure properties

Let

$\mathsf{C}(\mathsf{FA}_\Sigma)$: set of languages over Σ accepted by finite automata

op_n : n-ary function $2^{\Sigma^*} \times \cdots \times 2^{\Sigma^*} \to 2^{\Sigma^*}$

where 2^{Σ^*} means the set of all subsets of Σ^*

e.g. the example for op_n :

op_1 : $(\ )^{\mathsf{c}}$ (complement)

op_2 : $\cup$ (union) $\cap$ (intersection) $\cdot$ (concatenation)

For all languages $L_1, \ldots, L_n$ in $\mathsf{C}(\mathsf{FA}_\Sigma)$, if $\mathsf{op}_n(L_1, \ldots, L_n) \in \mathsf{C}(\mathsf{FA}_\Sigma)$,

we say "the class $\mathsf{C}(\mathsf{FA}_\Sigma)$ is closed under op_n"

E.g. $\mathsf{C}(\mathsf{FA}_\Sigma)$ is closed under union iff for all L_1, L_2 in $\mathsf{C}(\mathsf{FA}_\Sigma)$, $L_1 \cup L_2$ in $\mathsf{C}(\mathsf{FA}_\Sigma)$.

Proposition

The class $\mathsf{C}(\mathsf{FA}_\Sigma)$ is closed under union, intersection, complement

Proof for $\cup$

Suppose $\mathcal{A}_1 = (\Sigma, P, \mathsf{p}_0, P_{\mathsf{fin}}, \Delta_1)$ and $\mathcal{A}_2 = (\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta_2)$ are finite automata whose sets P, Q of state symbols are disjoint to each other. Let r_0 be a fresh state symbol, then define $\mathcal{B} = (\Sigma, P \cup Q \cup \{\mathsf{r}_0\}, \mathsf{r}_0, P_{\mathsf{fin}} \cup Q_{\mathsf{fin}}, \Delta_1 \cup \Delta_2 \cup \{\mathsf{r}_0 \to \mathsf{p}_0,\ \mathsf{r}_0 \to \mathsf{q}_0\})$. By construction, trivially $\mathcal{B}$ accepts a word w if and only if $\mathcal{A}_1$ or $\mathcal{A}_2$ accepts w. $\square$

Proof for $\cap$

From the previous proposition about DFA, we suppose $\mathcal{A}_1 = (\Sigma, P, \mathsf{p}_0, P_{\mathsf{fin}}, \Delta_1)$ and $\mathcal{A}_2 = (\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta_2)$ are already DFA. Define $\mathcal{C} = (\Sigma, P \times Q, \langle \mathsf{p}_0, \mathsf{q}_0 \rangle, P_{\mathsf{fin}} \times Q_{\mathsf{fin}}, \Delta)$ where

$$\begin{array}{ccl} P \times Q & : & \{\, \langle p, q \rangle \mid p \in P,\ q \in Q \,\} \\ P_{\mathsf{fin}} \times Q_{\mathsf{fin}} & : & \{\, \langle p, q \rangle \mid p \in P_{\mathsf{fin}},\ q \in Q_{\mathsf{fin}} \,\} \\ \Delta & : & \{\, \langle p_1, q_1 \rangle \xrightarrow{a} \langle p_2, q_2 \rangle \mid p_1 \xrightarrow{a} p_2 \in \Delta_1,\ q_1 \xrightarrow{a} q_2 \in \Delta_2 \,\} \end{array}$$

$\mathcal{C}$ is DFA and it simulates the transition moves of $\mathcal{A}_1$ and $\mathcal{A}_2$, simultaneously. $\square$

Proof for $\cap$ (another version)

Suppose $\mathcal{A}_1 = (\Sigma, P, \mathsf{p}_0, P_{\mathsf{fin}}, \Delta_1)$ and $\mathcal{A}_2 = (\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta_2)$ are finite automata whose sets P, Q of state symbols are disjoint. Let r, s be fresh state symbols, then define $\mathcal{C}' = (\Sigma, P \times Q, \langle \mathsf{p}_0, \mathsf{q}_0 \rangle, P_{\mathsf{fin}} \times Q_{\mathsf{fin}}, \Delta')$ where

$$\begin{aligned}\Delta' \quad : \quad & \{\, \langle p_1, q_1 \rangle \xrightarrow{a} \langle p_2, q_2 \rangle \mid p_1 \xrightarrow{a} p_2 \in \Delta_1,\, q_1 \xrightarrow{a} q_2 \in \Delta_2 \,\} \cup \\ & \{\, \langle p_1, q \rangle \longrightarrow \langle p_2, q \rangle \mid p_1 \longrightarrow p_2 \in \Delta_1,\, q \in Q \,\} \cup \\ & \{\, \langle p, q_1 \rangle \longrightarrow \langle p, q_2 \rangle \mid q_1 \longrightarrow q_2 \in \Delta_2,\, p \in P \,\}\end{aligned}$$

$\mathcal{C}'$ simulates the transition moves of $\mathcal{A}_1$ and $\mathcal{A}_2$, simultaneously, whenever Δ_1 contains $p_1 \xrightarrow{a} p_2$ and Δ_2 contains $q_1 \xrightarrow{a} q_2$. For the transition move by $p_1 \longrightarrow p_2$ of $\mathcal{A}_1$, the other move in $\mathcal{A}_2$ is suspended by assuming that for each state q in Q, there exists a transition rule $q \longrightarrow q$ in Δ_2. For the move by $q_1 \longrightarrow q_2$ of $\mathcal{A}_2$, we assume a similar condition for $\mathcal{A}_1$. $\square$

Proof for $(\ )^{\mathsf{c}}$

The basic idea to construct a finite automaton that accepts for a given finite automaton $\mathcal{A}$ the complement of $\mathcal{L}(\mathcal{A})$ is similar to the construction of DFA from finite automata. See 6–7 in Exercise. $\square$

Example

Consider $\mathcal{A}_1 = (\{\mathsf{a}\}, P, \mathsf{p}_0, \{\mathsf{p}_0\}, \Delta_1)$ and $\mathcal{A}_2 = (\{\mathsf{a}\}, Q, \mathsf{q}_0, \{\mathsf{q}_0\}, \Delta_2)$ where

$$\Delta_1 \;:\; \mathsf{p}_0 \xrightarrow{\mathsf{a}} \mathsf{p}_1 \quad \mathsf{p}_1 \xrightarrow{\mathsf{a}} \mathsf{p}_0$$

$$\Delta_2 \;:\; \mathsf{q}_0 \xrightarrow{\mathsf{a}} \mathsf{q}_1 \quad \mathsf{q}_1 \xrightarrow{\mathsf{a}} \mathsf{q}_2 \quad \mathsf{q}_2 \xrightarrow{\mathsf{a}} \mathsf{q}_0$$

Note that

$$\mathcal{L}(\mathcal{A}_1) = (\mathsf{aa})^* \quad \text{sequence of a's of multiple of 2}$$

$$\mathcal{L}(\mathcal{A}_2) = (\mathsf{aaa})^* \quad \text{sequence of a's of multiple of 3}$$

Now we define by product construction that

$$Q_\times \;:\; (\mathsf{p}_0,\mathsf{q}_0) \quad (\mathsf{p}_0,\mathsf{q}_1) \quad (\mathsf{p}_0,\mathsf{q}_2) \quad (\mathsf{p}_1,\mathsf{q}_0) \quad (\mathsf{p}_1,\mathsf{q}_1) \quad (\mathsf{p}_1,\mathsf{q}_2)$$

$$Q_{\times\,\mathsf{fin}} \;:\; (\mathsf{p}_0,\mathsf{q}_0)$$

$$\Delta_\times \;:\; (\mathsf{p}_0,\mathsf{q}_0) \xrightarrow{\mathsf{a}} (\mathsf{p}_1,\mathsf{q}_1) \quad (\mathsf{p}_0,\mathsf{q}_1) \xrightarrow{\mathsf{a}} (\mathsf{p}_1,\mathsf{q}_2) \quad (\mathsf{p}_0,\mathsf{q}_2) \xrightarrow{\mathsf{a}} (\mathsf{p}_1,\mathsf{q}_0)$$

$$(\mathsf{p}_1,\mathsf{q}_0) \xrightarrow{\mathsf{a}} (\mathsf{p}_0,\mathsf{q}_1) \quad (\mathsf{p}_1,\mathsf{q}_1) \xrightarrow{\mathsf{a}} (\mathsf{p}_0,\mathsf{q}_2) \quad (\mathsf{p}_1,\mathsf{q}_2) \xrightarrow{\mathsf{a}} (\mathsf{p}_0,\mathsf{q}_0)$$

Let $\mathcal{A}_\times = (\{\mathsf{a}\}, Q_\times, \{(\mathsf{p}_0,\mathsf{q}_0)\}, Q_{\times\,\mathsf{fin}}, \Delta_\times)$, then one can show $\mathcal{L}(\mathcal{A}_\times) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$

Decidability

Computable function (by Turing 1936) :

program, which is purely mechanical process, that for some of the input for an instance of a given problem, terminates and gives the correct yes/no answer   * computable function is total if it is defined for every input

Cf. effective method by Rosser 1939

effectively calculable function by Kleene 1952 (Church 1936 for informal use)

effective procedure by Minsky 1967

but, algorithm is named after Abu Abdullah Muhammad ibn Musa al-Khwarizmi 825

Decidable problem :

A decision problem (*Entscheidungsproblem*) is called decidable if there exists a total computable function which solves the problem

Undecidable problem :

If there does not exist a total computable function solving the problem, the problem is called undecidable.

Proposition

The following problems are decidable for the class of finite automata :

$w \in \mathcal{L}(\mathcal{A})$?	(membership problem)
$\mathcal{L}(\mathcal{A}) = \emptyset$?	(emptiness problem)
$\mathcal{L}(\mathcal{A}) = \Sigma^*$?	(universality problem)
$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?	(inclusion problem)
$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$?	(equivalence problem)

Proof

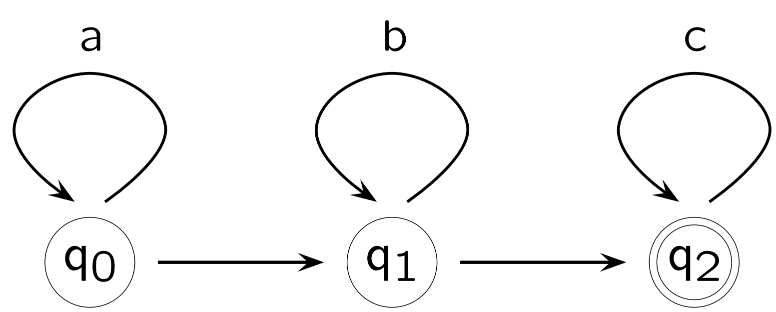
Omitted (proofs in a more general framework will be found in seminar talk 3). Note that if the emptiness problem is decidable, the other problems are decidable, due to the property that for finite automata $\mathcal{A}, \mathcal{B}$, one can construct finite automata, each of which accepts the union $\mathcal{L}(\mathcal{A}) \cup \mathcal{L}(\mathcal{B})$, the intersection $\mathcal{L}(\mathcal{A}) \cap \mathcal{L}(\mathcal{B})$, the complement $(\mathcal{L}(\mathcal{A}))^{\mathsf{c}}$. For instance, the membership problem can be rephrased to the question if $\{w\} \cap \mathcal{L}(\mathcal{A}) \neq \emptyset$. $\square$

Finite state diagrams

directed graph with

Σ	:	alphabet
Q	:	finite set of vertices with labels
(q0)	:	start vertex such that (q0) in Q
Q_{fin}	:	final vertices such that $Q_{\text{fin}} \subseteq Q$
Δ	:	Σ-labeled or unlabeled arrows connecting vertices

e.g. this finite state diagram accepts $\mathsf{a}^\ell \mathsf{b}^m \mathsf{c}^n \quad (\ell, m, n \geqslant 0)$



Cf. Moore machine, Mealy machine

Regular grammar

regular grammar $\mathcal{G} = (\,\Sigma,\, T,\, N,\, \mathsf{q}_0,\, \Delta\,)$

Σ	:	alphabet
T	:	set of terminal symbols such that $T \subseteq \Sigma$
N	:	set of non-terminal symbols such that $N = \Sigma - T$
q_0	:	start symbol such that $\mathsf{q}_0 \in N$
Δ	:	finite set of production rules with the following forms

$$p \longrightarrow a \qquad p \longrightarrow a\,q \qquad p \longrightarrow \varepsilon \qquad (p, q \in N,\;\; a \in T)$$

word generated by $\mathcal{G}$: word over T reachable from q_0

language generated by $\mathcal{G}$: set of words generated by $\mathcal{G}$

(called regular language)

Exercise

1. Can we determine if a given finite set is an alphabet or not? Specifically, given words $w_1, \ldots, w_n$ over an alphabet Σ, is it decidable whether $\{w_1, \ldots, w_n\}$ is an alphabet?

2. [McMillan's Theorem] Let $w_1, \ldots, w_n$ be n non-empty words over the alphabet $\Sigma = \{a_1, \ldots, a_k\}$. If $\{w_1, \ldots, w_n\}$ is an alphabet, then
$$\sum_{i=1}^{n} k^{-|w_i|} \leqslant 1$$
where $|w_i|$ is the length of word w_i. Prove the above statement.

3. Let $\Sigma = \{a_1, \ldots, a_k\}$ be the alphabet. Show that for n natural numbers $p_1, \ldots, p_n$ that possibly include the same numbers, there exist n non-empty words $w_1, \ldots, w_n$ over Σ of length $p_1, \ldots, p_n$, respectively, such that $\{w_1, \ldots, w_n\}$ is the alphabet if and only if
$$\sum_{i=1}^{n} k^{-|w_i|} \leqslant 1.$$

Exercise (cont'd)

4. Show that for every finite set L of words over an alphabet Σ, one can construct a finite automaton $\mathcal{A}$ over Σ that accepts L.

5. For a finite automaton $\mathcal{A}_{\mathsf{d}}$ obtained from $\mathcal{A}$ in page 10, show that $\mathcal{A}_{\mathsf{d}}$ is DFA and $\mathcal{L}(\mathcal{A}_{\mathsf{d}}) = \mathcal{L}(\mathcal{A})$.

6. Even if the condition "$T \neq \varnothing$ for Δ_{d}" in the step (2) is eliminated, $\mathcal{B}$ is DFA and $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{A})$. Verify this statement.

7. Let $\mathcal{D} = (\Sigma, 2^Q, \{\mathsf{q}_0\}, 2^Q - Q_{\mathsf{d}_{\mathsf{fin}}}, \Delta_{\mathsf{d}})$ associated to the above $\mathcal{B}$ in 6. Show that $\mathcal{L}(\mathcal{D}) = \Sigma^* - \mathcal{L}(\mathcal{A})$.

8. Show that the class $\mathrm{C}(\mathrm{FA}_\Sigma)$ is closed under concatenation. The concatenation of language L_1 to language L_2 over the same alphabet, denoted $L_1 \cdot L_2$, is defined as $L_1 \cdot L_2 = \{\, u\,w \mid u \in L_1,\, w \in L_2 \,\}$.

9. Show that languages accepted by finite automata are regular languages, and regular languages are accepted by finite automata.

Appendix : Post's correspondence problem (PCP)

Given an alphabet Σ

instance of PCP of size n	:	n pairs of words v_i, w_i $(i \leqslant n)$ over Σ
solution to this instance of length k	:	sequence $i_1\, i_2 \ldots i_k$ of indices such that $v_{i_1} v_{i_2} \ldots v_{i_k} \equiv w_{i_1} w_{i_2} \ldots w_{i_k}$

Question if there is a solution to a given instance is undecidable [1], even for size 7 [2], but decidable for 2 [3]. Decidability for size 3−6 is unknown so far.

---

[1] E.L. Post: *A Variant of a Recursively Unsolvable Problem*, Bulletin of the American Mathematical Society 52, pp.264–268, 1946

[2] Y. Matiyasevich & G. Senizergues: *Decision Problems for Semi-Thue Systems with a Few Rules*, Proc. of 11th LICS, pp.523-531, 1996

[3] A. Ehrenfeucht, J. Karhumaki, G. Rozenberg: *The (Generalized) Post Correspondence Problem with Lists Consisting of Two Words is Decidable*, TCS 21, pp.119-144, 1982

Advanced topics : MSO and regular languages

vocabulary (F, R) of second-order logic :

- F : a finite set of function symbols (with arity)
- R : a finite set of relation symbols (with arity)

variables (V_1, V_2) :

- V_1 : a set of first-order variables
- V_2 : a set of second-order variables (with non-zero arity)

terms T ::= x if $x \in V_1$
| $f(T, \ldots, T)$ if $f \in F$

formulas Ψ ::= $T = T$
| $r(T, \ldots, T)$ if $r \in R$
| $\exists x\, \Psi$ | $\forall x\, \Psi$ if $x \in V_1$
| $X(T, \ldots, T)$ if $X \in V_2$
| $\exists X\, \Psi$ | $\forall X\, \Psi$
| $\Psi \vee \Psi$ | $\Psi \wedge \Psi$ | $\neg\Psi$

Examples

1. Given a finite set A, define an SO-sentence ψ_1 such that $|A|$ is even iff ψ_1 has a model $\mathcal{A}$ whose carrier is A. We take ψ_1 to be

$$\begin{aligned}\exists X\,\exists F\,[\;&\forall x\,\forall y\,(F(x,y)\;\Rightarrow\;X(x)\wedge\neg X(y))\;\wedge\\ &\forall x\,\exists y\;F(x,y)\;\;\wedge\;\;\forall y\,\exists x\;F(x,y)\;\;\wedge\\ &\forall x_1\,\forall x_2\,\forall y_1\,\forall y_2\,(F(x_1,y_1)\wedge F(x_2,y_2)\;\Rightarrow\;(y_1=y_2\;\Leftrightarrow\;x_1=x_2))\,]\end{aligned}$$

Then, ψ_1 has a model $\mathcal{A}_1$ whose carrier is A_1 iff $|A_1|$ is even ($\because$ F is a bijective mapping from X to $A_1 - X$).

2. Graph connectivity : Given a finite directed graph $\mathcal{G}$ whose predicate $\mathrm{e}(\_,\_)$ indicates the edge relation, the SO-sentence ψ_2 defined below satisfies that $\mathcal{G}$ is connected iff $(G, \varnothing, \{\mathrm{e}\}) \models \psi_2$:

$$\forall X\;\neg[\;\exists x\;X(x)\;\;\wedge\;\;\exists x\;\neg X(x)\;\;\wedge\;\;\forall x\,\forall y\;(X(x)\wedge\neg X(y)\;\Rightarrow\;\neg\mathrm{e}(x,y))\;]$$

3. Graph 3-colorability : Similarly, given a finite directed graph $\mathcal{G}$, we define the SO-sentence ψ_3 below such that $\mathcal{G}$ is 3-colorable iff $(G, \varnothing, \{\mathrm{e}\}) \models \psi_3$:

$$\begin{aligned}\exists X\,\exists Y\,\exists Z\,[\;&\forall x\;\{\,\textstyle\bigvee_{U\in\{X,Y,Z\}}(U(x)\;\Leftrightarrow\;\bigwedge_{V\in\{X,Y,Z\}-\{U\}}\neg V(x))\,\}\;\wedge\\ &\forall x\,\forall y\;\{\,\textstyle\bigwedge_{U\in\{X,Y,Z\}}(\mathrm{e}(x,y)\wedge U(x)\;\Rightarrow\;\neg U(y))\,\}\,]\end{aligned}$$

SO vs. FO

SO is strictly more expressive than FO (first-order logic).

Proof

FO $\subseteq$ SO is obvious; we show below that the inclusion is strict, by using the following two theorems :

[Compactness] A set S of sentences (i.e. closed formulas) over a vocabulary (F, R) has a model iff every finite subset of S has a model.

[Downward Löwenheim-Skolem] If a set S of sentences over a vocabulary (F, R) has a infinite model, then S has a countable model.

Given a vocabulary (F, R) whose F and R are empty, consider the (F, R)-structure $\mathcal{A} = (A, \emptyset, \emptyset)$, where A is the carrier. Suppose, for leading to contradiction, that there exists an FO-sentence ψ such that $\mathcal{A} \models \psi$ iff $|A| \bmod 2 = 0$. Note that this property is definable in SO. Let $\phi_k = \exists x_1 \ldots \exists x_k \bigwedge_{i \neq j} \neg(x_i = x_j)$ for all $k \geqslant 0$, and define two sets of sentences, $S_1 = \{\psi\} \cup \{\phi_k \mid k \geqslant 0\}$ and $S_2 = \{\neg\psi\} \cup \{\phi_k \mid k \geqslant 0\}$. By Compactness theorem, each of S_1 and S_2 has a model. Let $\mathcal{A}_i$ be a model of S_i ($i \in \{1, 2\}$). Since each model must be infinite, by Downward Löwenheim-Skolem theorem, $\mathcal{A}_i = (A_i, \emptyset, \emptyset)$ such that A_i is a countable set ($i \in \{1, 2\}$). Thus, $\mathcal{A}_1$ and $\mathcal{A}_2$ are isomorphic. However, $\mathcal{A}_1 \models \psi$ and $\mathcal{A}_2 \models \neg\psi$, leading to the contradiction. $\square$

Monadic second-order logic (MSO)

$$\text{MSO} \quad : \quad \forall X \in V_2 : X \text{ is 1-ary}$$

$$\exists\text{MSO} \quad : \quad \forall \psi \in \Psi : \psi \text{ is MSO formula } \exists X_1 \ldots \exists X_n \phi \text{ such that}$$
$$\phi \text{ does not contain } \exists X \text{ or } \forall X$$

$$\forall\text{MSO} \quad : \quad \forall \psi \in \Psi : \psi \text{ is MSO formula } \forall X_1 \ldots \forall X_n \phi \text{ such that}$$
$$\phi \text{ does not contain } \exists X \text{ or } \forall X$$

Note 1

$$\text{FO} \subseteq \exists\text{MSO}, \forall\text{MSO} \subseteq \text{MSO} \subseteq \text{SO}$$

It is not obvious, however, whether each inclusion is strict or not, though FO $\subsetneq$ SO.

Note 2

- "Graph connectivity" is definable in $\forall$MSO (Example 2 in page 23).
- "Graph 3-colorability" is definable in $\exists$MSO (Example 3 in page 23).

SO-definable languages

For each word w over Σ, define

$\mathcal{A}_w$: finite structure $(\{1,\ldots,|w|\}, \varnothing, \{\_<\_\} \cup \{\mathsf{p}_c(\_)\}_{c\in\Sigma})$

where $<$: linear order on $\mathbb{N}$

p_c : $\mathsf{p}_c(i) = \text{true}$ if $w = ucv$ $(u, v \in \Sigma^*)$ & $|uc| = i$

(e.g. if $w = \text{aabcb}$, then $\mathsf{p}_\mathsf{a}(1) = \text{true}$, but $\mathsf{p}_\mathsf{b}(1) = \text{false}$).

A language defined by an SO-sentence ψ, denoted $\mathcal{L}(\psi)$:

$$\mathcal{L}(\psi) = \{ w \in \Sigma^* \mid \mathcal{A}_w \models \psi \}$$

Example

Consider

$$\begin{aligned}\psi = \exists X\, [\; & \forall x\, (\forall y\, (x = y \vee x < y) \;\Rightarrow\; X(x)) \wedge \\ & \forall x\, (\forall y\, (x = y \vee y < x) \;\Rightarrow\; \neg X(x)) \wedge \\ & \forall x\, \forall y\, \{ x < y \wedge \neg\exists z\, (x < z \wedge z < y) \;\Rightarrow\; X(x) \Leftrightarrow \neg X(y) \}\,],\end{aligned}$$

then

$\mathcal{L}(\psi) = (\text{ab})^*, (\text{ba})^*, \ldots$ over the alphabet $\{\text{a}, \text{b}\}$

Büchi-Elgot-Trakhtenbrot's theorem

A language is definable in MSO iff the language is regular.

Proof

First we show the "if" part : Given a DFA $\mathcal{A} = (\Sigma, Q, \mathsf{q}_0, Q_{\mathsf{fin}}, \Delta)$, we define below an $(\exists)$MSO-formula $\psi_{\mathcal{A}}$ such that $\mathcal{L}(\psi_{\mathcal{A}}) = \mathcal{L}(\mathcal{A})$. Let $Q = \{\mathsf{q}_0, \ldots, \mathsf{q}_n\}$. We introduce SO-variables $X_{\mathsf{q}0}, \ldots, X_{\mathsf{q}n}$ to indicate a state of the tape-head such that $X_q(x) = \mathsf{true}$ iff the head is in position q. So, at the beginning, if $\mathsf{q}_0 \xrightarrow{c} q$, then

$$\psi_{\mathsf{init}} = \forall x \bigwedge_{c \in \Sigma} [\, \mathsf{p}_c(x) \wedge \forall y\, (x = y \vee x < y) \;\Rightarrow\; X_q(x)\,].$$

For each transition $p \xrightarrow{c} q$, we have

$$\psi_{\mathsf{tran}} = \forall x\, \forall y \bigwedge_{p \in Q} \bigwedge_{c \in \Sigma} [\, X_p(x) \wedge \mathsf{p}_c(x) \wedge (x < y \wedge \neg \exists z\, (x < z \wedge z < y)) \;\Rightarrow\; X_q(y)\,].$$

When accepting an input by the transition $p \xrightarrow{c} q$ $(q \in Q_{\mathsf{fin}})$,

$$\psi_{\mathsf{fin}} = \forall x\, [\forall y\, (x = y \vee y < x) \;\Rightarrow\; \bigvee_{q \in Q_{\mathsf{fin}}} X_q(x)\,].$$

We take the conjunction of the above three formulas to be $\psi_{\mathcal{A}}(X_{\mathsf{q}0}, \ldots, X_{\mathsf{q}n})$. Then, by construction, it is not difficult to see $\mathcal{L}(\exists X_{\mathsf{q}0} \ldots \exists X_{\mathsf{q}n}\, \psi_{\mathcal{A}}(X_{\mathsf{q}0}, \ldots, X_{\mathsf{q}n})) = \mathcal{L}(\mathcal{A})$.

Next, to prove the "only if" part, we use a result obtained as a consequence of MSO Ehrenfeucht-Faïssé theorem (e.g. [4]). The result is explained in the next page.

---

[4] L. Lipkin: *Elements of Finite Model Theory*, EATCS, Springer-Verlag, 2004.

Proof (cont'd)

Let ψ be an MSO-formula, we write rank$(\psi) = k$ if the depth of quantifier nesting is at most k ($k \in \mathbb{N}$), and we write MSO$[k]$ for the set $\{\psi \mid \text{rank}(\psi) \leqslant k\}$. Given two structures $\mathcal{A}$ and $\mathcal{B}$, we say $\mathcal{A}$ and $\mathcal{B}$ are elementary MSO-equivalent up to k, denoted $\mathcal{A} \equiv_k^{\mathsf{MSO}} \mathcal{B}$, when $\mathcal{A} \models \psi$ iff $\mathcal{B} \models \psi$ for all $\psi \in$ MSO$[k]$. Then it is known that the following property hold :

Lemma 1 For every vocabulary (F_0, R), where F_0 is a finite set of constant symbols, with finite fixed numbers m, n of MSO and FO free variables, MSO$[k]$ can be partitioned to $S_1, \ldots, S_\ell$ and contains formulas $\psi_1(\vec{x}_m, \vec{X}_n), \ldots, \psi_\ell(\vec{x}_m, \vec{X}_n)$, such that

1. for every structure $\mathcal{A}$ with the carrier A, and elements $\vec{a}_m \in A^m$ and $\vec{U}_n \in (2^A)^n$, there exists i such that $\mathcal{A} \models \psi_i(\vec{a}_m, \vec{U}_n)$ iff $\mathcal{A} \models \theta(\vec{a}_m, \vec{U}_n)$ for all $\theta \in S_i$,
2. for every $\theta \in$ MSO$[k]$, there exists J such that θ is equivalent to $\bigvee_{j \in J} \psi_j$.

We suppose that a language L over Σ is defined by an MSO sentence δ with rank$(\delta) = k$. According to the above lemma, over a vocabulary $(\varnothing, \{<\} \cup \{\mathsf{p}_c\}_{c \in \Sigma})$ with no SO or FO free variable, MSO$[k]$ can be partitioned to $S_1, \ldots, S_\ell$, and MSO$[k]$ contains sentences $\psi_1, \ldots, \psi_\ell$ that satisfy the above conditions 1 and 2. This implies that there exists a subset F of $\{1, \ldots, \ell\}$ such that $\delta \equiv \bigvee_{i \in F} \psi_i$. Moreover, there exists some S_e that contains a sentence logically equivalent to $\neg \exists x\, (x = x)$. Now we define the finite automaton $\mathcal{A}_\delta = (\Sigma, \{1, \ldots, \ell\}, e, F, \Delta)$. (Proof cont'd)

Proof (cont'd)

The transition rules in Δ of $\mathcal{A}_\delta$ are defined as follows :

$$i \xrightarrow{c} j \qquad \text{if } \mathcal{B}_w \models \theta \text{ for all } \theta \in S_i \text{ and } \mathcal{B}_{wc} \models S_j \text{ for all } \theta \in S_j.$$

We show below that for every word w, after reading w, the automaton $\mathcal{A}_\delta$ ends in some state i ($1 \leqslant i \leqslant \ell$) such that $\mathcal{B}_w \models \theta$ for all $\theta \in S_i$. We use induction on the length of w. The base case is obvious, because $\mathcal{A}_\delta$ ends in the initial state e. For induction step, we suppose $w = vc$ for some $c \in \Sigma$. By induction hypothesis, after reading w, $\mathcal{A}_\delta$ is in state i and $\mathcal{B}_w \models \theta$ for all $\theta \in S_i$. If the next character on the tape is c, there is some j such that $\mathcal{B}_{wc} \models \theta$ for all $\theta \in S_j$. That means, $\mathcal{A}_\delta$ ends in such j after reading wc. Note that $\exists x\,(x = x)$ ($\equiv$ true) is in MSO[k]. Finally, we consider the language accepted by $\mathcal{A}_\delta$. From the above observation, the language $\mathcal{L}(\mathcal{A}_\delta)$ is $\{w \in \Sigma^* \mid \exists i \in F \colon \mathcal{B}_w \models \theta \text{ for all } \theta \in S_i\}$, which is equivalent to $\{w \in \Sigma^* \mid \exists i \in F \colon \mathcal{B}_w \models \psi_i\}$. Hence, because $\delta = \bigvee_{i \in F} \psi_i$, we have $\mathcal{L}(\delta) = \mathcal{L}(\mathcal{A}_\delta)$. □

Corollary

MSO $=$ $\exists$MSO over words.

Proof

According to the proof of the previous theorem, every language definable in MSO is regular, and every regular language is expressible in $\exists$MSO. □

MSO $\subsetneq$ SO

There exists a language definable in SO, but not in MSO.

Proof

We show that the language $L_= = \{\, w \in \{\mathsf{a}, \mathsf{b}\}^* \mid |w|_\mathsf{a} = |w|_\mathsf{b} \,\}$ is definable in SO. Here $|w|_\mathsf{a}$ means the number of occurrences of a in w. Define $\psi_=$ to be

$$\exists X \, \exists F \, [\, \forall x \, (X(x) \Leftrightarrow \mathsf{p_a}(x) \;\wedge\; \neg X(x) \Leftrightarrow \mathsf{p_b}(x)) \;\wedge\; \forall x \, \forall y \, (F(x,y) \Rightarrow X(x) \wedge \neg X(y)) \;\wedge\; \forall x \, \exists y \, F(x,y) \;\wedge\; \forall y \, \exists x \, F(x,y) \;\wedge\; \forall x_1 \, \forall x_2 \, \forall y_1 \, \forall y_2 \, (F(x_1,y_1) \wedge F(x_2,y_2) \;\Rightarrow\; (y_1 = y_2 \Leftrightarrow x_1 = x_2) \,].$$

The above sentence $\psi_=$ specifies that for each word w over the alphabet $\{\mathsf{a}, \mathsf{b}\}$, the number of occurrences of a and b in w are the same iff $\mathcal{A}_w \models \psi_=$. Thus, $\mathcal{L}(\psi_=) = L_=$. However, $L_=$ is not a regular language, which will be explained later in Exercises 5,6 in the next seminar talk. Indeed, $L_=$ is a context-free language (See page 5, seminar talk 2). Therefore, by Büchi-Elgot-Trakhtenbrot's theorem (page 27), $L_=$ is not definable in MSO. □

Note

Over the alphabet $\Sigma = \{\mathsf{a}\}$, a language is context-free iff it is definable in MSO. (Cf. Exercise 7, seminar talk 2)

FO $\subsetneq$ MSO

There exists a language definable in MSO, but not in FO.

Proof

In order to show the statement, one should introduce the following result :

[Gurevich, 1984] Given two linear ordered structures $\mathcal{A} = (A, \varnothing, \{<_A\} \cup R_A)$ and $\mathcal{B} = (B, \varnothing, \{<_B\} \cup R_B)$, where $<_X$ is a linear order over X $(X \in \{A, B\})$, then $|A|, |B| \geqslant 2^k$ implies $\mathcal{A} \equiv_k^{\mathsf{FO}} \mathcal{B}$.

Using the above property, we show that (aa)* is not definable in FO. Suppose, for leading to contradiction, that there exists an FO-sentence ψ such that $\mathcal{L}(\psi) = (\mathsf{aa})^*$ and rank$(\psi) = k$. Let $\mathcal{A}_w$ be a structure associated to a word w in a*, and let $\mathcal{B}_{w\mathsf{a}}$ be a structure associated to wa. Obviously, $\mathcal{A}_w \models \psi$ iff $\mathcal{B}_{w\mathsf{a}} \not\models \psi$. However, according to Gurevich's claim, if $|w| \geqslant 2^k$, then $\mathcal{A}_w \equiv_k^{\mathsf{FO}} \mathcal{B}_{w\mathsf{a}}$, leading to the contradiction. □

Corollary

There is no FO-sentence ψ_{even} such that for every linear ordered structure $\mathcal{A} = (A, \varnothing, \{<\} \cup R)$, $|A|$ is even iff $\mathcal{A} \models \psi_{\mathsf{even}}$.

($\because$ If ψ_{even} exists, the language (aa)* is definable by ψ_{even}.)

Exercise for advanced topics

1. Show that the Hamiltonicity of finite undirected graphs can be expressed in SO, i.e. construct an SO-sentence ψ_{Ham} over graph structures such that a finite undirected graph $\mathcal{G}$ is Hamiltonian iff $\mathcal{G} \models \psi_{\mathsf{Ham}}$. (Note that a finite undirected graph is Hamiltonian iff there exists a path in the graph which visits every vertex exactly once.)

2. Show that a^*b^* is definable in FO, i.e. construct an FO-sentence ψ such that $\mathcal{L}(\psi) = a^*b^*$. Likewise, show that Σ^* and $\varnothing$ are definable in FO.

3. Construct an MSO-sentence that defines $a^*(bb)^*a^*$.

4. Show that there is no FO-sentence ψ_{conn} such that for every finite directed graph $\mathcal{G} = (G, \varnothing, \{e\})$, $\mathcal{G}$ is connected iff $\mathcal{G} \models \psi_{\mathsf{conn}}$. (Hint : Use the previous corollary guaranteeing that there is no FO-sentence which expresses the linear ordered set is even.)

5. Show that FO $\subsetneq \exists$MSO and FO $\subsetneq \forall$MSO.

National Institute of Advanced Industrial Science and Technology (AIST) – Senri-site, AIST Kansai.

Office: Shin-Senri Nishi 1–2–14 (MSK bldg. 5th floor), Toyonaka, Osaka 560–0083, Japan

URL: **http://staff.aist.go.jp/hitoshi.ohsaki/**