Lec 06. More MSO & Properties of Regular Languages

Eunjung Kim

MSO LOGIC ON STRINGS

We first express a string $s \in \Sigma^*$ as a logical structure (often called "relational structure").

STRING w AS A LOGICAL STRUCTURE

Universe $= [n]$, where n is the length of the string.

- That is, each "position" (from 1 to n) in the string is an element in the universe. If $w = \epsilon$, the universe is $\emptyset$.

A binary relation $<$ and $|\Sigma|$ unary relations P_a for all $a \in \Sigma$ on the universe.

- $x < y$: "the x-th position precedes the y-th position in the string."
- $P_0(x)$ is true if "the x-th symbol is 0."

$\tau = \{<\} \cup \{P_a \mid a \in \Sigma\}$ is called the vocabulary on Σ-strings.

MSO LOGIC ON STRINGS

MSO-FORMULA ON Σ-STRINGS

An MSO-formula on strings is a well-formed string that can be constructed using from atomic formulas for (infinite supply of) individual variables $x, y, z \ldots$, and set variables $X, Y, Z \cdots$ i.e.

- $x < y$; note that $< \in \tau$,
- $P_a(x)$ for each $a \in \Sigma$,
- $x = y$, and $x \in X$.

by applying

- the logical connectives $\wedge, \vee, \neg, \rightarrow$; $\varphi_1 \wedge \varphi_2$, $\neg\varphi$, etc,
- the universal and existential quantifier $\forall, \exists$; in the form $\exists x\varphi$, $\exists X\varphi$, etc.

An MSO-formula in which all variables are quantified (by $\forall$ or $\exists$) is called an MSO-sentence.

MSO LOGIC ON STRINGS

A property = the set of all Σ-strings which has the property.

A property on strings as an MSO-sentence

We say that a property $L \subseteq \Sigma^*$ on strings (a.k.a. a language) is expressible, or equivalently definable, in MSO if there is an MSO-sentence φ on Σ-strings such that

$$w \in L \text{ if and only if } w \models \varphi$$

for every string $w \in \Sigma^*$.

$$\varphi = \forall x \forall y \; \big((x < y) \to \big(\exists z \; (x < z < y) \vee P_0(x) \vee P_0(y)\big)\big)$$

MSO LOGIC ON STRINGS, BY EXAMPLE

Let us express the property L on $\{0,1\}$-strings having even number of 1's, i.e.

$$L = \{w \in \{0,1\}^* \mid \text{there are even number of 1's in } w\}.$$

Use the fact that $w \in L$ if and only if

- either $w = \epsilon$,
- or the positions of 1's in w can be "uniquely colored" in RED or BLUE so that the colors alternate, and the first 1 is RED and the last 1 is in BLUE.

MSO LOGIC ON STRINGS, BY EXAMPLE

MSO-FORMULA DEFINING L

- $\varphi_\epsilon = \neg\exists x\,(x = x)$
- $\varphi_{color}(R, B) = \forall x\,(P_1(x) \rightarrow (x \in R \vee x \in B)) \wedge (P_0(x) \rightarrow \neg(x \in R \vee x \in B))$
- $\varphi_{unique}(R, B) = \forall x\,(x \in R \rightarrow \neg x \notin B) \wedge (x \in B \rightarrow \neg x \notin R)$
- $\varphi_{alternate}(R, B) =$??????
- $\varphi_{firstlast}(R, B) =$??????

Finally, we get a sentence φ_L defining L as

$$\varphi_L = \varphi_\epsilon \vee \exists R\,\exists B\,\varphi_{color}(R, B) \wedge \varphi_{unique}(R, B) \wedge \varphi_{alternate} \wedge \varphi_{firstlast}$$

Büchi's Theorem 1960

Recognizability equals definability on strings

A language is regular if and only if it is definable in MSO.

Proof sketch of ($\Rightarrow$).

- Show that for each atomic regular expressions ($\emptyset$, ϵ, a for each $a \in \Sigma$), the corresponding language can be defined in MSO.
- Show that the languages of $R_1 \cup R_2$, $R_1 \circ R_2$ and R_1^* can be defined in MSO, assuming that $L(R_1), L(R_2)$ can be defined in MSO.

Büchi's Theorem 1960

How to define the language of an atomic regular expression in MSO.

- $\emptyset$
- ϵ
- a for each $a \in \Sigma$.

BÜCHI'S THEOREM 1960

How to define the language of $\cup, \circ, *$ in MSO assuming that $L(R_1), L(R_2)$ are defined in MSO.

- $R_1 \cup R_2$
- $R_1 \circ R_2$
- R_1^*

MSO LOGIC ON STRINGS, BY EXAMPLE

Define in MSO

$$L = L_1 \circ L_2$$

where $L_1 = L((00)^+)$ and $L_2 = L((11)^+)$.

Questions to examine

1. Given an NFA M, decide if $L(M) = \emptyset$ or not.
2. Given two regular languages L_1 and L_2, decide if $L_1 = L_2$.
3. Is *Prefix*(L) is regular when L is regular?
4. How about *Suffix*(L)?
5. Quotient of L by a symbol $a \in \Sigma$, denoted by L/a, is regular when L is?
6. How about $a \setminus L$?
7. Fix a DFA M and a state $s \in Q$. The set of all strings w such that the (accepting) computation history of w visits the state s, is it regular?
8. Fix a DFA M. The set of all strings w such that the (accepting) computation history of w visits all the state of M, is it regular?

DECIDING IF $L = \emptyset$

Given a regular language L, we want to decide if $L = \emptyset$ or not.

L IS GIVEN BY NFA N

$L(N) \neq \emptyset$ if and only if there is a directed path from the initial state q_0 to OOOOOOOOOOO in the transition diagram of N.

Recall: $w \in \Sigma^*$ satisfies $\delta^*(q_0, w) = q$ if and only if there is a (q_0, q)-walk in the transition diagram labelled by w (ϵ-label allowed).

Deciding if $L = \emptyset$

Given a regular expression R, we want to decide if $L(R) = \emptyset$ or not.
You can convert R into an NFA and apply the previous criteria, or do the following.

L is given by a regular expression R

If there is no occurrence of $\emptyset$ in R, $L(R) \neq \emptyset$.

Otherwise, check if $L(R) = \emptyset$ inductively:

1. $L(R_1 \cup R_2) = \emptyset$ if and only if $L(R_1) = \emptyset$ and $L(R_2) = \emptyset$.
2. $L(R_1 \cdot R_2) = \emptyset$ if and only if $L(R_1) = \emptyset$ or $L(R_2) = \emptyset$.
3. $L(R^\star) \neq \emptyset$ (even when $R = \emptyset$).

When L is regular, so is $Prefix(L)$?

Given two strings $x, w \in \Sigma^*$, x is a prefix of w if $w = xy$ for some $y \in \Sigma^*$.
For a language $L \subseteq \Sigma^*$, let $Prefix(L) = \{x \in \Sigma^* : x \text{ is a prefix of } w \in L\}$.

If L is regular, $Prefix(L)$ is regular

Let M be an DFA with $L = L(M)$. Notice that

$w \in L$ can be written as $w = xy$ if and only if $\hat{\delta}(q_0, x) = q$ for some state $q \in Q$ such that ....

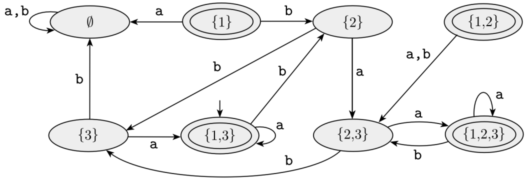

Figure 1.43, Sipser 2012

When L is regular, so is $Prefix(L)$?

Let M be an DFA with $L = L(M)$. Then

$$Prefix(L) = \bigcup_{q \in Q \text{ such that....}} L_q.$$

where $L_q = \{x \in \Sigma^* : \hat{\delta}(q_0, x) = q\}$.

- If L_q is regular, then $Prefix(L)$ is regular (why?)
- Is L_q regular?
- $properPrefix(L)$ be the set of all proper prefixes of some $w \in L$; x is a proper prefix of w if $w = xy$ for some $y \in \Sigma^+$.
- Is $properPrefix(L)$ regular?

When L is regular, so is $\mathit{Suffix}(L)$?

Given two strings $x, w \in \Sigma^*$, x is a suffix of w if $w = yx$ for some $y \in \Sigma^*$.
For a language $L \subseteq \Sigma^*$, let $\mathit{Suffix}(L) = \{x \in \Sigma^* : x \text{ is a suffix of } w \in L\}$.

If L is regular, $\mathit{Suffix}(L)$ is regular

Let *reverse*(L) be the set of all strings each of which is a reversal w^R of some string $w \in$L.

- If L is regular, *reverse*(L) is regular as well; homework.
- $\mathit{Suffix}(L)$ can be obtained by applying ???? and ???? operations on L.

QUOTIENT L/a FOR $a \in \Sigma$

Given a language L over Σ and a symbol $a \in \Sigma$, the quotient of L by a denoted as L/a is the language

$$\{x \in \Sigma^* : xa \in L\}.$$

Is L/a regular?

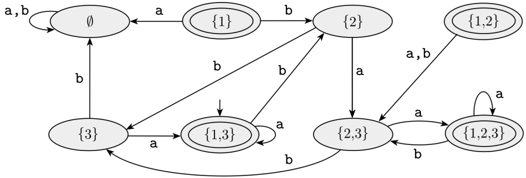

Figure 1.43, Sipser 2012

- For a state $q \in Q$, if $x \in L_q$ satisfies $xa \in L$ for some x, then for all $y \in L_q$ we have $ya \in L$.
- That is, $L_q \subseteq L/a$ or $L_q \cap L/a = \emptyset$.
- How to tell if $L_q \subseteq L/a$?

The language $a \setminus L$ for $a \in \Sigma$

Given a language L over Σ and a symbol $a \in \Sigma$, the language $a \setminus L$ is defined as

$$\{x \in \Sigma^* : ax \in L\}.$$

Is $a \setminus L$ regular?

Idea: Express $a \setminus L$ using the operations we examined so far to immediately conclude.

More exotic language P_s

- Fix a DFA M and a state $s \in Q$.
- Let P_s be the set of all string $w \in L$ such that the accepting computation history of w visits the state s.
- Is P_s regular?

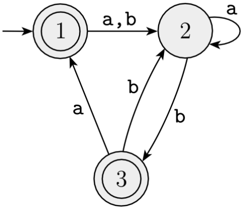

Figure 1.21, Sipser 2012

More exotic language P_s

First approach.

- For any string w, $w \in P_s$ if and only if it can be written as $w = xy$ with $\hat{\delta}(q_0, x) = s$ and $\hat{\delta}(s, y) \in F$.
- That is $P_s = L_s \cdot A_s$, where L_q and A_q are defined for all $q \in Q$ as

$$L_q = \{x \in \Sigma^* : \hat{\delta}(q_0, x) = q\}.$$

$$A_q = \{x \in \Sigma^* : \hat{\delta}(q, x) \in F\}.$$

- Is any one of L_q and A_q regular?

More exotic language P_s

Second approach: use Myhill-Nerode Theorem.

Myhill-Nerode Theorem

L is regular if and only if the number of equivalence classes of $\equiv_L$ is finite.

Idea: use the DFA M recognizing L to identify the equivalence relation $\equiv_{P_s}$, (or a refinement of it) of finite index.

- For $Z \subseteq Q$ and $q \in W$, let $L_{Z,q}$ be the set of all strings w such that the computation history of w on M visits precisely the states in Z and end in q.
- $\Sigma^* = \dot{\bigcup}_{Z \subseteq Q, q \in Z} L_{W,q}$.
- We want to argue that any strings $x, y \in L_{Z,q}$ are indistinguishable by P_s. But for proving this claim, we need to try *all strings z which might potentially distinguish x and y... or do we?*

More exotic language P_s

Second approach: use Myhill-Nerode Theorem and test for a finite number of extensions z (and argue that it suffices).

Myhill-Nerode Theorem, in action

P_s is regular if for any $Z \subseteq Q$ and $q \in Z$,

- any $x, y \in L_{Z,q}$ are indistinguishable by P_s, or equivalently
- for any $x, y \in L_{Z,q}$ and for any $z \in \Sigma^*$, $xz \in P_s$ if and only if $yz \in P_s$.

What are the key property of z which will make $xz \in P_s$ (or not) for $x \in L_{Z,q}$?

MORE EXOTIC LANGUAGE P_s

Second approach: use Myhill-Nerode Theorem and test for a finite number of extensions z (and argue that it suffices).

MYHILL-NERODE THEOREM, IN ACTION

P_s is regular if for any $Z \subseteq Q$ and $q \in Z$,

- any $x, y \in L_{Z,q}$ are indistinguishable by P_s, or equivalently
- for any $x, y \in L_{Z,q}$ and for any $z \in \Sigma^*$, $xz \in P_s$ if and only if $yz \in P_s$.

What are the key property of z which will make $xz \in P_s$ (or not) for $x \in L_{Z,q}$?

1. whether $\delta^*(q, z) \in F$ or not: this dictates whether $xz \in L$.
2. whether the states visited by the computation history of $\delta^*(q, z)$ include s or not: this affects whether the computation history of xz from q_0 visits s or not.

A bit more exotic language

Fix a DFA M. The set of all strings w such that the (accepting) computation history of w visits all the state of M, is it regular?

Even More exotic language

Why do we care about the second approach using Myhill-Nerode theorem when the first approach seems much simpler?

Even more exotic language. Fix two states s_1, s_2 of a DFA M. Let P_{s_1,s_2} be the set of strings $w \in L$ whose computation history visits both s_1, s_2 and visiting s_2 only after visiting s_1.

Is P_{s_1,s_2} regular?