FORMAL LANGUAGES AND AUTOMATA, 2025 FALL SEMESTER

Lec 06. Minimum DFA, Myhill-Nerode and MSO logic

Eunjung Kim

Reducing the number of states of DFA

- Why does this procedure works? (i.e. produces an equivalent automaton)
- Given a DFA *M*, the procedure leads to a unique outcome?
- Is this a DFA with the minimum possible number of states?
- Does the procedure leads to the same (minimum) DFA regardless of the starting DFAs?

Why does this procedure works?

We observe

- Any pair marked as distinguishable are indeed distinguishable.
  $\rightsquigarrow$ By induction, we argue that any marked pair has a distinguishing string.

- Any pair unmarked at the end of procedure are indistinguishable.
  $\rightsquigarrow$ Suppose not, and unmarked pair p, q of states is distinguished by a string w of length n. Consider the sequence of states in the computation histories of (p, w) and (q, w)...

Why does this procedure works?

Now the "groups" in Q are indeed the equivalence classes of $\sim$.

- Let $Q_1, \ldots, Q_\ell$ be the equivalence classes.
- Key fact: For $p, p' \in Q_i$ (i.e. $p \sim p'$), $\delta(p, a) \sim \delta(p', a)$ for every $a \in \Sigma$.
- So the "quotient $M/\sim$ of M is well-defined; this is our new DFA.

$$\delta'([p], a) := [\delta(p, a)]$$

  well-defined; $\delta'([p], a) = [\delta(p, a)] = [\delta(q, a)] = \delta'([q], a)$

- Uniqueness of the procedure's outcome from a given DFA follows.
- Check yourself that $L(M) = L(M/\sim)$.

Is this a DFA with the minimum # states?

New states of $M/\sim$ are distinguishable

- Choose two inequivalent states of M, i.e. $q_1 \not\sim q_2$, and let w be a string distinguishing q, q'.
- For any $q_1' \sim q_1$, w also distinguishes q_1' and q_2. (Why?)

$\rightsquigarrow$ every pair of new states in $M/\sim$ are distinguishable.

Is this a DFA with the minimum # states?

Let $p_0, p_1, \ldots, p_\ell$ be the states of $M' = (Q', \Sigma, \delta', p_0, F')$ (our new DFA obtained from M).

Suppose there is another DFA D with $q < \ell$ states.

- Choose ℓ strings $s_1, \ldots, s_\ell \in \Sigma^*$ such that $\hat{\delta}'(p_0, s_i) = p_i$ for each $i \in [\ell]$.
- Such strings exist because every state of M' is accessible from p_0.
- Run D on these ℓ strings; there exist two strings s_i, s_j s.t. D ends up in the same state upon s_i and s_j.
- Note that there is a string distinguishing p_i and p_j for any pair $0 \leq i < j \leq \ell$ by the previous observation.
- What are the states you reach when you run D on $s_i \circ w$ and $s_j \circ w$?

Does the procedure leads to the same (minimum) DFA regardless of the starting DFAs?

- Here, we are asking if there is a unique minimum DFA (up to renaming the states).
- Answer via so-called Myhill-Nerode Theorem.
- Myhill-Nerode Theorem can also be used as an alternative approach for establishing non-regularity of a language.

Myhill-Nerode Theorem

Fix an alphabet Σ and let L be a language over Σ.

Indistiguishability of two strings by L

We say that two strings $x, y \in \Sigma^*$ is indistinguishable by L if for all $z \in \Sigma^*$,

$$x \cdot z \in L \text{ if and only if } y \cdot z \in L,$$

written as $x \equiv_L y$.

Distiguishability of two strings by L

We say that $z \in \Sigma^*$ is a distinguishing extension of two strings $x, y \in \Sigma^*$ for L if

$$x \circ z \in L \text{ and } y \circ z \notin L, \text{ or vice versa.}$$

Note that $x \not\equiv_L y$ if and only if there is a distinguishing extension of them.

Myhill-Nerode Theorem

Myhill-Nerode Theorem

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

($\leftarrow$) Build a DFA $D = (Q, \Sigma, \delta, q_0, F)$ from the equivalence classes of $\equiv_L$. Use the fact that $x \equiv_L y$ implies $x \circ a \equiv_L y \circ a$ for every $a \in \Sigma$ (why?).

- $Q =$ the set of the equivalence classes of $\equiv_L$ (often written as $\Sigma^* / \equiv_L$).
- $q_0 =$???. $[\varepsilon]$
- $\delta([x], a) =$??? $[xa]$ for each $a \in \Sigma$.
- $F \subseteq Q$: $[x] \in F$ for every $x \in L$.

Myhill-Nerode Theorem

Myhill-Nerode Theorem

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

Moreover, the number of equivalence classes equals the number of states in a minimal (minimum) DFA.

($\rightarrow$, also the second part) Consider any DFA M with $L(M) = L$. Note that if $\hat{\delta}(q_0, x) \sim \hat{\delta}(q_0, y)$ for two strings $x, y \in \Sigma^*$, then $x \equiv_L y$.

Myhill-Nerode Theorem for non-regularity

Myhill-Nerode Theorem, in contraposition

L is non-regular if and only if there is an infinite set $S \subseteq \Sigma^*$ consisting of pairwise distinguishable strings.

Myhill-Nerode Theorem for non-regularity

Myhill-Nerode Theorem, in contraposition

L is non-regular if and only if there is an infinite set $S \subseteq \Sigma^*$ consisting of pairwise distinguishable strings.

- Mind that we seek for distinguishable strings, which are not necessarily in L.
- For pairwise distinguishable strings $S = \{s_1, \ldots, s_m, \cdots\}$, a distinguishing extension for (s_i, s_j) might be in general different from a distinguishing extension for (s_j, s_k).

Myhill-Nerode Theorem for non-regularity, example

- $L_1 = \{0^n 1^n \mid n \geq 1\}$
- $L_2 = \{w \in \{0,1\}^* \mid w \text{ is a palindrome}\}$

Strategy: find an infinite subset of Σ^* which consists of pairwise distinguishable (inequivalent) strings.

MSO LOGIC ON STRINGS

We saw several, all equivalent, characterization of regular language.

- DFA / NFA (algorithm)
- Regular expression (composability via basic operations)
- Recognizability by monoid (algebraic property)
- Myhill-Nerode Theorem
- Generated by left/right linear grammar (not covered, yet)
- Definability by Monadic Second Order logic

MSO LOGIC ON STRINGS, BY EXAMPLE

We want to express the language

$$L = \{w \in \{0,1\}^* \mid w \text{ does not contain 11 as a substring}\}$$

with an MSO-sentence.

MSO-SENTENCE

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

Here, $P_0(x)$ is read as "the x-th symbol in the string is 0".

Likewise, $P_1(y)$ is read as "the y-th symbol in in the string is 1".

10010 satisfies φ whereas 1101 not, which we denote as $10010 \models \varphi$ and $1101 \not\models \varphi$.

MSO LOGIC ON STRINGS, BY EXAMPLE

We want to express that

a set S of positions in the given string forms an "interval".

MSO-FORMULA

$$\varphi_{int}(S) = \forall x \, \forall y \, (x \in S \land y \in S \land x \leq y) \rightarrow \big(\forall z \, (x \leq z \leq y) \rightarrow z \in S\big)$$

Note that the validity of $\varphi_{int}(S)$ depends not only on the given string, but also the variable S.

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 $w \in \Sigma^*$.

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 two colors alternate.

MSO LOGIC ON STRINGS, BY EXAMPLE

MSO-FORMULA DEFINING L

- $\varphi_\epsilon = \neg\exists x\,(x = x)$
- $\varphi_{color}(R, B) = \forall x\,(P_1(x) \to (x \in R \vee x \in B)) \wedge (P_0(x) \to \neg(x \in R \vee x \in B))$
- $\varphi_{unique}(R, B) = \forall x\,(x \in R \to \neg x \overset{\in}{\notin} B) \wedge (x \in B \to \neg x \overset{\in}{\notin} R)$
- $\varphi_{alternate}(R, B) =$?????? $\forall x \forall y ((x<y] \wedge (x \in R \wedge y \in R \to \exists z (x<z<y) \wedge z \in R)$

Finally, we get a sentence φ_L defining L as

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

Büchi's Theorem 1960

Recognizability equals definability on strings

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