Lec 05. Pumping Lemma & Minimal DFA

Eunjung Kim

Limit of finite automata and tools for investigation

Which of the following languages are regular?

1. $B = \{0^n 1^n : n \geq 0\}$.
2. $C = \{w : w$ has equal number of 0's and 1's$\}$.
3. $D = \{w : w$ has equal number of 01's and 10's$\}$.

Pumping lemma

Pumping lemma: tool to prove nonregularity

Let A be a regular language. Then there exists a number p (called the pumping length) such that any string $w \in A$ of length at least p, w can be written as $w = xyz$ such that the following holds:

1. $|y| \geq 1$,
2. $|xy| \leq p$,
3. $xy^iz \in A$ for every $i \geq 0$.

Proof idea: DFA for A has a finite (constant) number of states.

Pumping lemma, proof

There exists DFA M with $L(M) = A$.

1. Let p be the number of states of this DFA.
2. Consider the accepting computation history $r_0 = q_0, r_1, \ldots, r_s$ for w (with $r_s \in F$) such that $r_{i+1} = \delta(r_i, w_{i+1})$ for all $i = 0, \ldots, s-1$, where w_i is the i-th symbol of w.
3. In the first $p+1$ states $r_0, \ldots, r_p$, there exist two identical states, say r_a and r_b, with $a \neq b$.
4. Take $x = w_1 \cdots w_a$, $y = w_{a+1} \cdots w_b$ and $z = w_{b+1} \cdots w_s$.
5. It remains to observe that
   - $r_{b+1} = \delta(r_b, w_{b+1}) = \delta(r_a, w_{b+1})$, and thus $w_1 \cdots w_a \cdot w_{b+1} \cdots w_s = x \cdot z = x \cdot y^0 \cdot z$ is accepted with the sequence of states $r_0, \ldots, r_a, r_{b+1}, \ldots, r_s$.
   - Any $x \cdot y^i \cdot z$ is accepted with the sequence

$$r_0, \ldots, r_a, (r_{a+1}, \ldots, r_b)^i, r_{b+1}, \ldots, r_s.$$

Pumping lemma for nonregularity

Pumping lemma

Let A be a regular language. Then there exists a number p such that any string $w \in A$ of length at least p, w can be written as $w = xyz$ such that ....

Recipe: assume that A is regular and p is an unknown (arbitrary) pumping length. Choose a good string s, and show that rewriting $s = xyz$ as required is impossible.

Pumping lemma for nonregularity

Pumping lemma

Let A be a regular language. Then there exists a number p such that any string $w \in A$ of length at least p, w can be written as $w = xyz$ such that ....

Recipe: assume that A is regular and p is an unknown (arbitrary) pumping length. Choose a good string s, and show that rewriting $s = xyz$ as required is impossible.
That is, we use the contraposition of Pumping lemma for proving nonregularity of A

Syntax for showing non-regularity

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz")
4. there exists $i \geq 0$ with $xy^i z \notin A$ ("$\exists i$").

NONREGULARITY OF $B = \{0^n1^n : n \geq 0\}$.

SYNTAX

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz")
4. there exists $i \geq 0$ with $xy^iz \notin A$ ("$\exists i$").

$\{w : w$ HAS EQUAL # OF 0'S AND 1'S$\}$

SYNTAX

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz meeting the conditions")
4. there exists $i \geq 0$ with $xy^iz \notin A$ ("$\exists i$").

Alternative way to show the non-regularity?

$D = \{1^{n^2} : n \geq 0\}$

Syntax

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz")
4. there exists $i \geq 0$ with $xy^i z \notin A$ ("$\exists i$").

$D = \{0^i \cdot 1^j : i > j\}$

Syntax

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz")
4. there exists $i \geq 0$ with $xy^iz \notin A$ ("$\exists i$").

$F = \{ww : w \in \{0,1\}^*\}$

SYNTAX

1. For every positive number p, ("$\forall p$")
2. there exists $w \in A$ of length at least p such that ("$\exists w \in A$")
3. for every split $w = xyz$ with $|y| \geq 1$ and $|xy| \leq p$ ("$\forall$ splits xyz")
4. there exists $i \geq 0$ with $xy^iz \notin A$ ("$\exists i$").

Reducing the number of states of DFA

A DFA recognizes a single (unique) language. But there are more than one (in fact, arbitrarily many) DFAs which recognizes the same language.

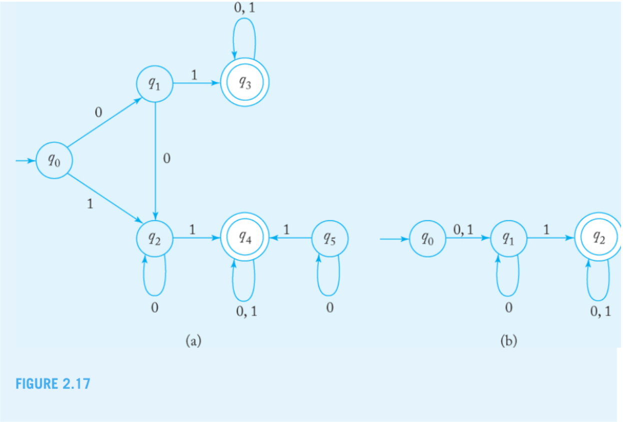

Figure 2.17, Peter Linz 2014.

Reducing the number of states of DFA

Indistinguishable states

Given DFA M, two states $p, q \in Q$ are <u>indistinguishable</u> if for every string $w \in \Sigma^*$,

$$\hat{\delta}(p, w) \in F \text{ if and only if } \hat{\delta}(q, w) \in F.$$

Remark: indistinguishability is an equivalence relation on Q.

Reducing the number of states of DFA

Indistinguishable states

Given DFA M, two states $p, q \in Q$ are indistinguishable if for every string $w \in \Sigma^*$,

$$\hat{\delta}(p, w) \in F \text{ if and only if } \hat{\delta}(q, w) \in F.$$

Remark: indistinguishability is an equivalence relation on Q.

Distinguishing string

We say that a string $w \in \Sigma^*$ distinguishes two states p, q if

$$\hat{\delta}(p, w) \in F \text{ and } \hat{\delta}(q, w) \notin F \text{ or vice versa.}$$

Reducing the number of states of DFA

Procedure for reducing # states of given DFA.

1. Remove all inaccessible (i.e. not accessible from q_0) states.
2. Any pair $(p, q) \in F \times Q \setminus F$ is marked as distinguishable.
3. Mark a pair p, q as distinguishable if there exists $a \in \Sigma$ such that the pair $\delta(p, a), \delta(q, a)$ is already marked as distinguishable.
4. Repeat above until there is no more pair to be marked distinguishable.
5. Group all states which are not marked as indistinguishable; the groups $(\sim)$ form a partition of Q.
6. $M/\sim$ is well-defined; this is our reduced automaton.

Reducing the number of states of DFA

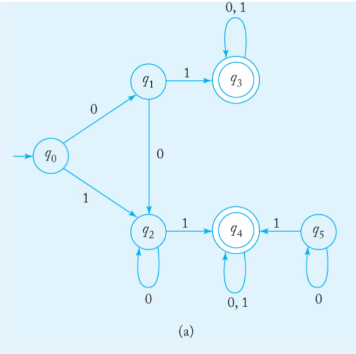

Figure 2.17 (a), Peter Linz 2014.

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.
  $\leadsto$ By induction, we argue that any marked pair has a distinguishing string.

- Any pair unmarked at the end of procedure are indistinguishable.
  $\leadsto$ Suppose not, and unmarked pair p, q 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.
- Uniqueness of the procedure's outcome from a given DFA follows.
- Check yourself that $L(M) = L(M/\sim)$.

Another example

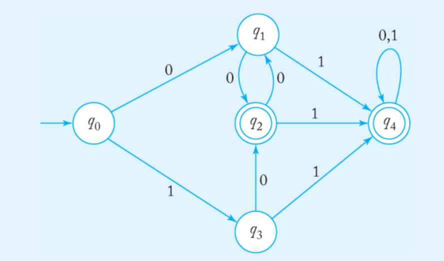

Figure 2.18, Peter Linz 2014.

Is this a DFA with the minimum # states?

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.
- This can also be used as an alternative approach for establishing non-regularity of a language.