Theory of Computation

Lecture 6

Nilesh Bansal

July 31, 2003

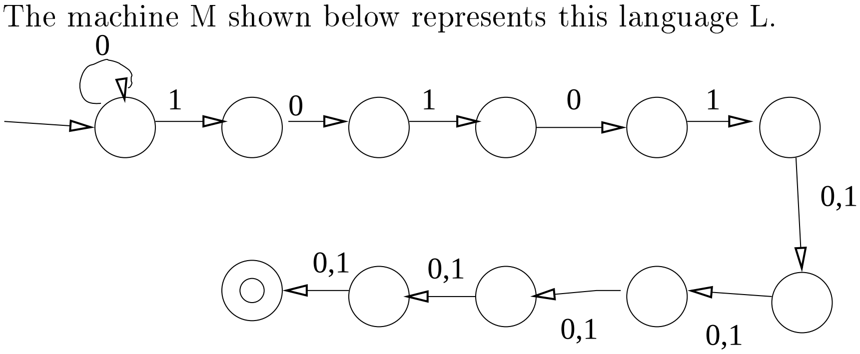
Advantage of NFA over DFA.

When NFA doesn't add any more power to DFA, why should one use NFA at all? Following example will make this point clear.

Consider a language L

L = $\{w \mid w \in \{0,1\}^*$, last 10^{th} to 6^{th} symbols are 10101 $\}$

The machine M shown below represents this language L.



It can be easily seen that
L(M) $\subseteq$ L
and L $\subseteq$ L(M)
which means L = L(M)

all strings of form $x10101a_1a_2a_3a_4a_5$ will belong to L.

We will now try constructing DFA for such a language. Since NFA doesn't add any power and since NFA exists, we can say DFA for L also exists.

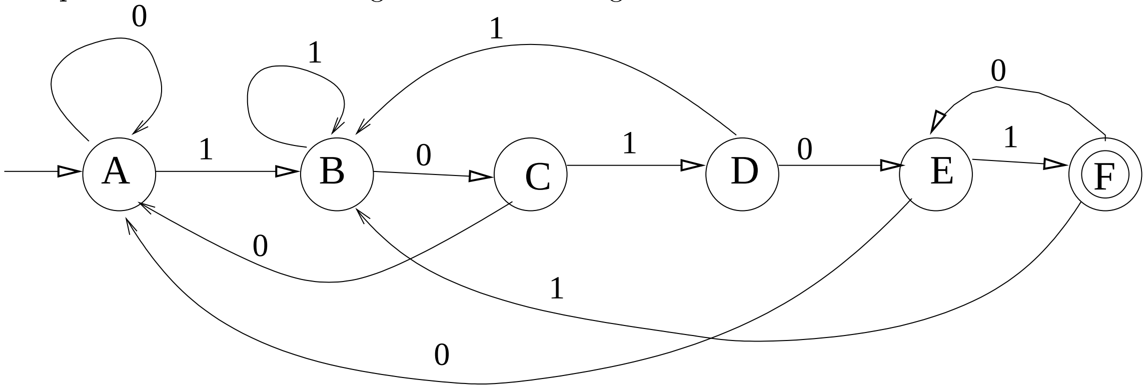
But constructing DFA will not be as simple as constructing NFA. For this language L, we not only have to remember the pattern 10101 but also have to remember position at which this pattern occured in string. This may force us to remember all of last 10 states which will lead to a DFA with 2^9 or more states. Although by a careful observation of L, the problem can possibly be solved using fewer states, but still solution will be very complicated as compared to its NFA.

1 0 1 0 1 0 1 1 0 1 0 0

1 2 3 4 5 6 7 8 9 10 11 12

In the above string, pattern 10101 starts both at 1^{st} and 3^{rd} position. Depending on where the string ends, we have to keep track of all occurences of 10101 in last 5 states that the DFA has seen.

Figure below shows a DFA that takes care of pattern of 10101 but not of relative position within the string from end of string.



Consider a DFA which remembers the last time it has seen each of the states A,B,C,D,E and F. Let $X_i, i \leq 4$ denote that i symbols earlier, DFA was in state X; and X_5 denotes that the state X was not encountered in previous 4 states. In such a DFA total number of states will rise to 6^5. Every state in this case will be a 6 tuple like $(A, B_2, C_4, D_1, E_5, F_3)$.

But it is not necessary to remember all states as its enough to remember when last time DFA was in (the accepting) state F. This will lead to $2^5 = 32$ states because in each DFA state F_1, F_2, F_3, F_4, F_5 may be or may not be present. If we observe the problem more carefully then even this number can be reduced by introducing more constrains like F_i and F_j can not be present in a state if $i - j = 1$, i.e. DFA cannot be in F on seeing 2 consecutive input symbols. But even after this constraint, resulting DFA will be far more complicated than corresponding NFA.

Hence, although NFA is not more powerful than DFA, it is simpler to construct and understand.

Equality of NFA and DFA

To construct DFA from NFA with n states, We construct a DFA with 2^n states, each corresponding to a subset of states of NFA.
Consider a NFA with starting state s_0, and string w. Prove that for all strings $w \epsilon \Sigma^*$

$\widehat{\delta}_N(s_0, w) = \{p_1, p_2, .., pr\}$ | NFA
iff
$\widehat{\delta}_D(\{s_0\}, w) = \{p_1, p_2, .., pr\}$ | DFA
where s_0 represents corresponding state in DFA
Call this statement $P(w)$ for the string w

Proof: by induction on length of string

Base Case: By definition of construction of DFA from NFA :
in the DFA, the state corresponding to state q of NFA is defined as $\{q\}$ iff $\widehat{\delta}_N(q, w) = \{p_1, p_2, .., pr\} = \widehat{\delta}_D(\{q\}, w)$

Inductive Hypothesis: Assume that $P(w)$ is true for all strings w with $|w| \leq k$, $k \geq 0$. We now need to prove $P(w)$ is true for all w with $|w| = k + 1$.

let $w = x.a$
with $|x| = k$, $x \epsilon \Sigma^*$ and $a \epsilon \Sigma$
then from definition of NFA

$$\widehat{\delta}_N(s_0, w) = \bigcup_{p \epsilon \widehat{\delta}_N(s_0, x)} \widehat{\delta}_N(p, a)$$

and from definition of DFA

$$\widehat{\delta}_D(\{s_0\}, w) = \widehat{\delta}_D(\widehat{\delta}_D(\{s_0\}, x), a)$$

also, since $P(x)$ is true for $|x| = k$ we have
$\widehat{\delta}_N(s_0, w) = \widehat{\delta}_D(\{s_0\}, w) = \{q_1, q_2, .., q_r\}(say)$
Now the proof can be divided in 2 parts, $LHS \Rightarrow RHS$ and $RHS \Rightarrow LHS$.

Part 1: LHS $\Rightarrow$ RHS
$\widehat{\delta}_N(s_0, w) = \{p_1, p_2, .., pr\} \Longrightarrow$

$$\widehat{\delta}_N(s_0, w) = \bigcup_{p \epsilon \widehat{\delta}_N(s_0,x)} \widehat{\delta}_N(p, a) = \bigcup_{p \epsilon \{q_1, q_2, .., qr\}} \widehat{\delta}_N(p, a) = \{p_1, p_2, .., pr\}$$

now $\widehat{\delta}_D(\{s_0\}, w) = \widehat{\delta}_D(\widehat{\delta}_D(\{s_0\}, x), a) = \widehat{\delta}_D(\{q_1, q_2, .., q_r\}, a)$ which means $\widehat{\delta}_D(\{s_0\}, w) = \{p_1, p_2, .., pr\}$ from definition of conversion of NFA to DFA.

Part 2: RHS $\Rightarrow$ LHS
$\widehat{\delta}_D(\{s_0\}, w) = \{p_1, p_2, .., pr\} \Longrightarrow \widehat{\delta}_D(\{s_0\}, w) = \widehat{\delta}_D(\widehat{\delta}_D(\{s_0\}, x), a) = \widehat{\delta}_D(\{q_1, q_2, .., q_r\}, a) = \{p_1, p_2, .., pr\}$
now

$$\widehat{\delta}_N(s_0, w) = \bigcup_{p \epsilon \widehat{\delta}_N(s_0,x)} \widehat{\delta}_N(p, a) = \bigcup_{p \epsilon \{q_1, q_2, .., qr\}} \widehat{\delta}_N(p, a)$$

which means $\widehat{\delta}_N(s_0, w) = \{p_1, p_2, .., pr\}$ again from definition of conversion between DFA and NFA.

Therefore, for $|w| = k + 1$, $LHS \Leftrightarrow RHS$ is true when it is assumed true for $|w| = k$. Base case is also true. Thus by principle of mathematical induction $P(w)$ is true for all $w \epsilon \Sigma^*$.