CSC236H: Introduction to the Theory of Computation

Homework 5 Solutions

---

1. Give a context-free grammar for each of the following languages.

   (a) $L_1 = \{0^n 1^m 0^m 1^n \mid n, m \geq 0\}$.
   Solution.

$$S \to 0S1 \mid C$$
$$C \to 1C0 \mid \epsilon$$

   (b) $L_2 = \{a^n b^m c^k \mid n, m, k \geq 0 \text{ and } n = m + k\}$.
   Solution.

$$S \to aSc \mid B$$
$$B \to aBb \mid \epsilon$$

   (c) $L_3 = \{a^n b^m c^k \mid n, m, k \geq 0 \text{ and } n = 2m + 3k\}$.
   Solution.

$$S \to aaaSc \mid B$$
$$B \to aaBb \mid \epsilon$$

   (d) $L_4 = \{a^n b^m \mid 0 \leq n \leq m \leq 2n\}$.
   Solution.

$$S \to aSbb \mid B$$
$$B \to aBb \mid \epsilon$$

2. In a string $w \in \{0,1\}^*$, define a *block* to be a maximal length consecutive sequence of 0s or 1s. For example, in the string 0111000011001 there are six blocks, three blocks of 0s of lengths 1, 4, and 2, and three blocks of 1s of lengths 3, 2, and 1.

   Let L be the language consisting of all strings that have (somewhere) two blocks of zeros of the same length. So, for example, $011001110 \in L$ but $000110011 \notin L$.

   (a) Prove that L is not regular.
   Solution. By pumping lemma, assuming parameter n:
   Let $x = 0^n 10^n$. Any partition of $x = uvw$ should satisfy $|uv| \leq n$ and $|v| \geq 1$. Therefore, we should have $u = 0^j$, $v = 0^k$ and $w = 0^\ell 10^n$, such that $j + k + \ell = n$, and $k \geq 1$. If we pump v to 0, then we will get the string $x' = 0^{j+\ell}10^n$, and since $k \geq 1$, we know that $j + \ell \neq n$. Therefore, by definition, $x' \notin L$. Hence, L does not satisfy the pumping lemma, and therefore is not regular.

   (b) Show that L is context-free by giving a grammar for L.
   Solution. There are different ways of doing this; here is one (let us call this grammar G):

$$S \to B \mid A1B \mid B1A \mid A1B1A$$
$$B \to 0B0 \mid 010 \mid 01A10$$
$$A \to 1A \mid 0A \mid \epsilon$$

   (c) Prove that your grammar of part (b) is correct.
   Solution. The goal is to prove that $L = L(G)$. We will do this by showing the following two subgoals:

   (1) $L(G) \subseteq L$: The goal is to prove that every string that G derives has two blocks of zeros of equivalent length. Again, there are several ways of doing this. I like the following approach because of its nice and clean structure:
   lemma 1: A generates all possible strings over the alphabet $\{0,1\}$ (basically, the set $\{0,1\}^*$).
   proof of lemma 1: This is an easy induction proof. The induction is on the length of the string.

The base case is ϵ, and the induction step is proved by considering the two simple cases of wether the strings begins with a 0 or a 1.
lemma 2: B generates all strings with two blocks of zeros of the same length where the first one appears at the beginning and the second one appears at the end of the string (let's call this property P).
proof of lemma 2: by induction on the number of derivations n:
Basis: $n = 1$, $B \Rightarrow 010$, which satisfies P.
Induction Hypothesis: If $B \Rightarrow_G^n v$ then v satisfies P.
Induction Step: prove that $B \Rightarrow_G^{n+1} w$ then w satisfies P.
If $B \Rightarrow_G^{n+1} w$ then $B \Rightarrow u \Rightarrow_G^n w$ (separating the first step from the rest). There are two options (ruling out 010 which was covered by the basis) for what u can be, based on the production rules for B:

i. $u = 0B0$: By induction hypothesis, $0B0 \Rightarrow_G^n 0v0$ where v satisfies P. But, if v satisfies p then $0v0$ (which is w) also satisfies P (by definition of P). So, this case is fine.
ii. $u = 01A10$: By lemma 1, A generates some string u in $\{0,1\}^*$ (in any number of derivation steps, including n). Therefore, $B \Rightarrow_G^{n+1} 01u10$. But, $01u10$ satisfies P since it has a block of zeros of length 1 at the beginning and at the end. So, this case is fine as well.

The main goal: S generates strings which have two blocks of zeros of equivalent length somewhere (meaning they belong to L).
proof: Goal: assume $S \Rightarrow_G^* w$. Then, $S \Rightarrow u \Rightarrow_G^* w$. There are four options for what u can be:

i. $u = B$: By lemma 2, we know then that $B \Rightarrow_G^n w$ where w has two blocks of zeros of equivalent length (specifically at the beginning and at the end), and therefore $w \in L$.
ii. $u = A1B$: By lemma 1, we know that $A \Rightarrow_G^* x$ where $x \in \{0,1\}^*$. By lemma 2, we know that $B \Rightarrow_G^* v$ such that v has two equivalent length blocks of zeros at the beginning and at the end. We have $w = x1v$. The block of zeros at the end of v remains intact, and the 1 in the middle makes sure that the block of zeros at the beginning of v remains a block of zeros (of the same length) in $x1v$. Thus, w has two equivalent length blocks of zeros, and therefore, $w \in L$.
iii. $u = B1A$: Similar to the above case.
iv. $u = A1B1A$: Similar to the above case.

Note that this last case did not need and induction proof, since the induction happens in the proofs of lemmas 1 and 2.

(2) $L \subseteq L(G)$: we will need the reverse of lemmas 1 and 2 for this direction.
Reverse Lemma 1: any string $w \in \{0,1\}^*$ can be generated by A.
proof: by simple induction on the length of the string.
Reverse Lemma 2: any string w with equivalent length blocks of zeros at its beginning and end can be generated by B.
proof: By strong induction on the length of the string.
Basis: $|w| = 0$, is vacuously true.
Induction Hypothesis: If v satisfies P (from lemma 2) and $|v| < n$, then $B \Rightarrow_G^* v$.
Induction step: If w satisfies P and $|w| = n$, then there are three options for what w can look like:

i. The equivalent length blocks of zeros in w have length exactly 1. There are two sub-cases for this case:
   A. $w = 010$: in this case $B \Rightarrow_G^w$ by using the production rule $B \rightarrow 010$.
   B. $w = 01v10$ such that $v \in \{0,1\}^*$. In this case, $B \Rightarrow_G^* w$, by using the production rule $B \rightarrow 01A10$ and reverse lemma 1.
ii. The equivalent length blocks of zeros in w have length greater than 1: in this case, we can remove one zero from the beginning and one zero from the end of w, and the remaining string will still have two equivalent length blocks of zeros at the beginning and at the end. In other words, we can write $w = 0v0$ such that v satisfies P, and $|v| < n$. Therefore, by induction hypothesis, $B \Rightarrow_G^* v$. Using the production rule $B \rightarrow 0B0$, we know that $B \Rightarrow_G^* w$.

The main goal: The goal is to prove that if $w \in L$ can be generated by G (that is, $S \Rightarrow_G^* w$). By definition, w has two blocks of zeros of equivalent length (let us call this ℓ). Therefore, we have $w = x0^\ell y0^\ell z$ such that one of the following is true:

i. $x = \epsilon \wedge y = \epsilon$: Then $w = 0^\ell y0^\ell$ is a string with equivalent length blocks of zeros at its beginning and end, and therefore it can be generated by B. Since we have the production rule $S \rightarrow B$, then it can be generated by S as well.

ii. $x = x'1 \wedge y = \epsilon$, where $x' \in \{0,1\}^*$: Then $w = x'10^\ell y0^\ell$. Since we have the production rule $S \to A1B$, and A can derive x' (by reverse lemma 1), and B can derive $0^\ell y0^\ell$ (by reverse lemma 2), then S can derive w.

iii. $x = \epsilon \wedge y = 1y'$, where $y' \in \{0,1\}^*$: similar to the above case.

iv. $x = x'1 \wedge y = 1y'$, where $x', y' \in \{0,1\}^*$: similar to the above case.

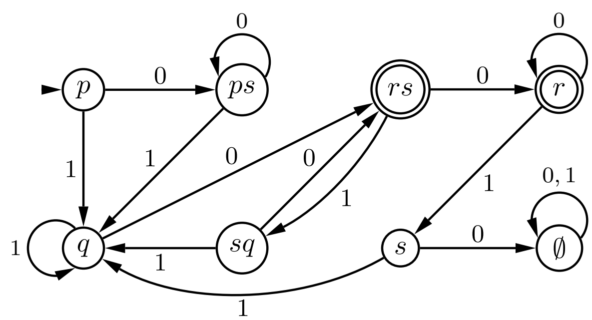
Note that this did not need and induction proof, since the induction happens in the proofs of reverse lemmas 1 and 2.

3. Consider the NFA defined by the transition table below, with p the initial state and r the only accepting state.

state	input 0	input 1
p	$\{p,s\}$	$\{q\}$
q	$\{r,s\}$	$\{q\}$
r	$\{r\}$	$\{s\}$
s	$\{\}$	$\{q\}$

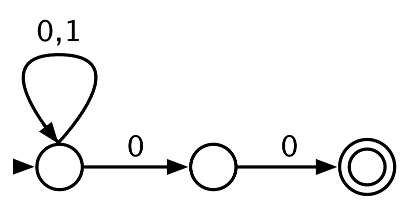
Construct a DFA that accepts the same language using the technique discussed in class. The states of your deterministic machine should be the sets of states of the nondeterministic machine.

Solution. In the diagram below, we have shortened the set representation by removing the braces and ","s. Thus, for example, the label rs corresponds to the set $\{r,s\}$. Also notice that not all subsets of $\{p,q,r,s\}$ have corresponding states in this figure. This is because we have removed the states that were unreachable from the start state as they were useless.

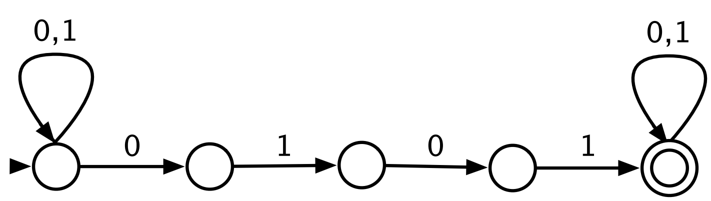


4. Give NFAs with the specified number of states recognizing each of the following languages.

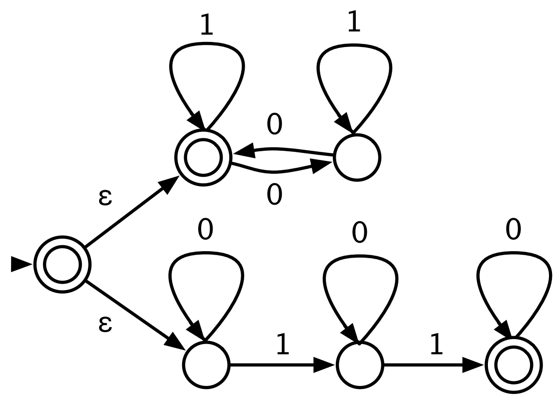
(a) The language $\{w \mid w \text{ ends with } 00\}$ with three states.
Solution.



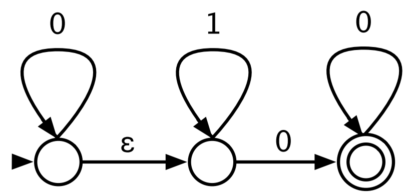
(b) The language $\{w \mid w \text{ contains the substring } 0101\text{, i.e., } w = x0101y \text{ for some } x \text{ and } y\}$ with five states.
Solution.



(c) The language $\{w \mid w \text{ contains an even number of 0s, or exactly two 1s}\}$ with six states.
Solution.



(d) The language $0^*1^*0^*0$ with three states.
Solution.



5. Give regular expressions for:

(a) All binary strings with exactly two 1's.
Solution.

$$0^*10^*10^*$$

(b) All binary strings with a double symbol (contains 00 or 11) somewhere.
Solution.

$$(0+1)^*(00+11)(0+1)^*$$

(c) All binary strings that contain both 00 and 11 as substrings.
Solution.

$$(0+1)^*00(0+1)^*11(0+1)^* + (0+1)^*11(0+1)^*00(0+1)^*$$

(d) All binary strings without a double symbol anywhere.
Solution.

$$(1+\epsilon)(01)^*(0+\epsilon)$$