CS 154 Introduction to Automata and Complexity Theory
Handout 16
Winter 1996 R. van Glabbeek February 21, 1996

Midterm Exam—Solutions

Problem 1 (25 points) Construct a 2SM (Stack machine with control over two stacks) the recognizes the language

$$\{x\#x \ : \ x \in (a \cup b)^*\}$$

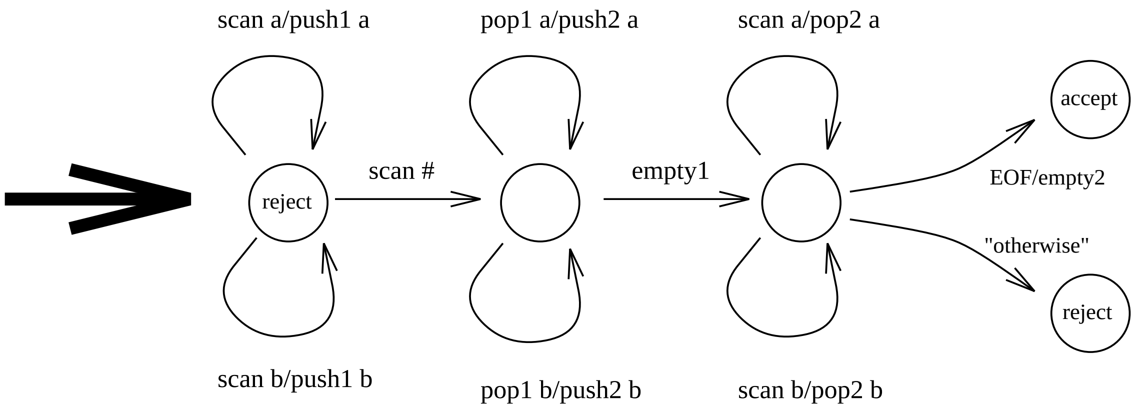


Figure 1: 2SM recognizing the given language

Problem 2 (20 points) Does the given string belong to the given regular set

<u>Yes</u> $01110111 \in (1^*01)^*(11 \cup 0^*)$

<u>No</u> $11100111 \in ((1^*0)^* \cup 0^*11)^*$

<u>No</u> $011100101 \in 01^*10^*(11^*0)^*$

<u>Yes</u> $1000011 \in (10^* \cup 11)^*(0^*1)^*$

Problem 3 (35 points) Are the following languages regular? (Prove your answer.)

1. $\{0^n \ : \ n \text{ is a multiple of } 8 \ \}$

   Yes, $(00000000)^*$ is a regular expression for this language.

2. $\{w \ : \ w \in (0 \cup 1)^* \text{ and } w \text{ contains more 0's than 1's } \}$

   No, with pumping lemma:
   Let N be given. Choose $z = 1^N 0^{N+1}$. If $z = uvw$ with $|v| > 0$ and $|uv| \leq N$, then $v = |1^k|$ with $1 \leq k \leq N$, and uv^2w is not in the language.

3. $\{w \ : \ w \in (0 \cup 1)^* \text{ and } w \text{ contains no substring of the form } xxx \text{ with } |x| \geq 1\}$.
   Answer this question under each of the following assumptions:

   - There exists an infinite string w, such that w contains *no* substring of the form xxx, with $|x| \geq 1$.
   - There exists no such string.

   You *don't* have to decide which of these assumptions is valid.

   Suppose there is an infinite string as mentioned above. Then the language is not regular, as can be shown with the pumping lemma. Let N be given. Choose a string z in the language with $|z| \geq N$. Such a string exists: just take a prefix of the infinite string w. Let $z = uvy$ with $|v| > 0$, then uv^3y is not in the language.

   Suppose there is no infinite string as above. Then the language must be finite, and hence regular. For suppose it is infinite: consider the infinite tree (as in exercise 0.6-29 of FB) whose nodes are the strings of the language, with Λ as root and x as parent of xa for $a \in \{0, 1\}$. This tree is finite-branching and therefore, by König's lemma, has an infinite branch. This infinite branch consist of a chain of strings, each one extending its parent by one letter, that do not contain substrings of the form xxx with $|x| \geq 1$. This gives rise to an infinite string which does not contain such substrings, contradicting the assumption. Therefore the language must be finite. Finally note that any finite language is regular.

Problem 4 (30 points)

1. Minimize the DFA in Figure 2 by determining E_0, E_1, E_2, ... until a fixpoint, as done in Section 4.7.1.
2. Extract the regular expression corresponding to the constructed DFA.

The initial state in Figure 2 is $Q0$, the final states are $Q6$ and $Q7$.

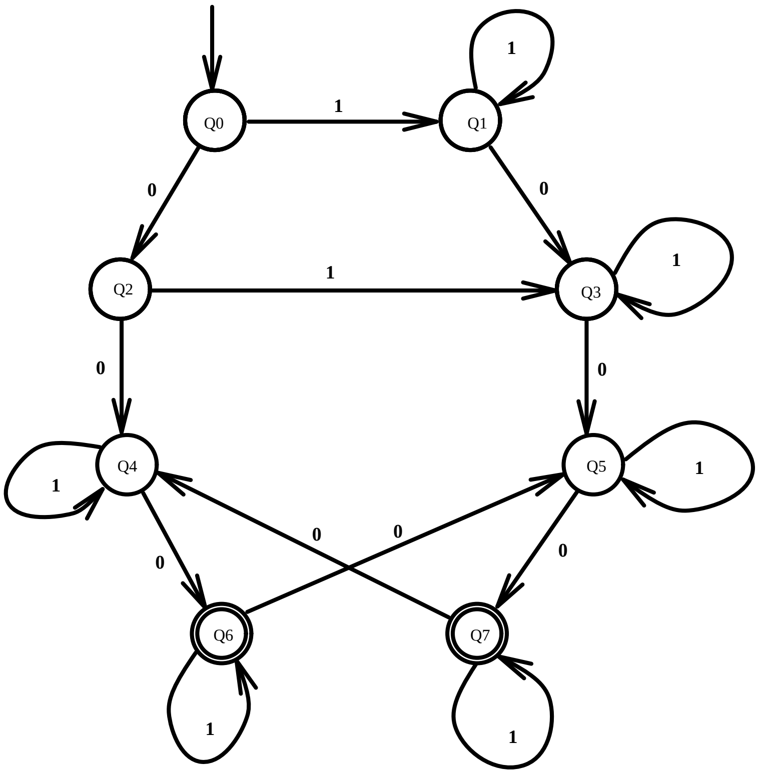


Figure 2: DFA to be minimized

(Please try to be neat with your diagrams)

1. $$\begin{aligned} E_0 &= \{\{Q0, Q1, Q2, Q3, Q4, Q5\}, \{Q6, Q7\}\} \\ E_1 &= \{\{Q0, Q1, Q2, Q3\}, \{Q4, Q5\}, \{Q6, Q7\}\} \\ E_2 &= \{\{Q0, Q1\}, \{Q2, Q3\}, \{Q4, Q5\}, \{Q6, Q7\}\} \\ E_3 &= E_2 \end{aligned}$$

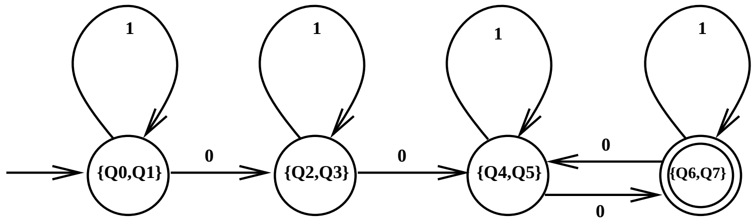


Figure 3: Minimized DFA

The final state is drawn with two circles.

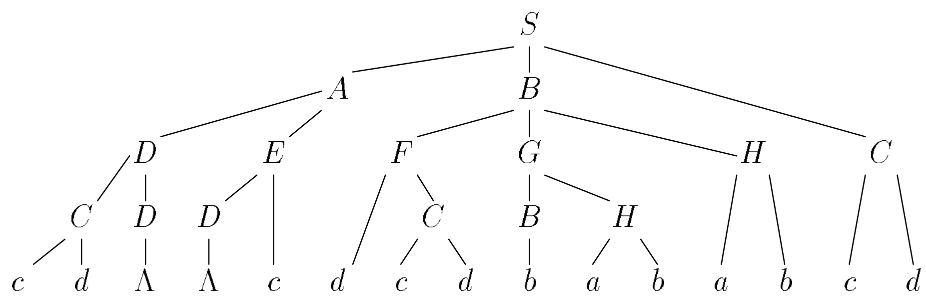
2. A regular expression corresponding to the automata is:

$1^*01^*01^*(01^*01^*)^*01^*$

Problem 5 (45 points) Consider the following CFG:

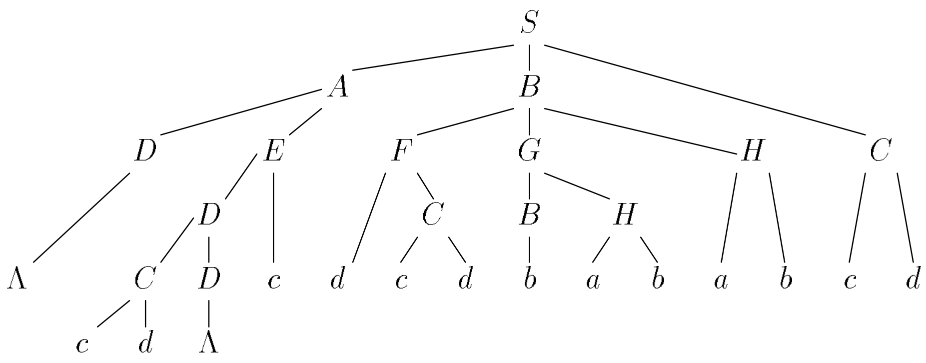
$$
\begin{array}{lcl}
S & \rightarrow & ABC \\
A & \rightarrow & DE \\
B & \rightarrow & FGH \\
B & \rightarrow & b \\
C & \rightarrow & cd \\
D & \rightarrow & CD \\
D & \rightarrow & \Lambda \\
E & \rightarrow & Dc \\
F & \rightarrow & dC \\
G & \rightarrow & BH \\
H & \rightarrow & ab
\end{array}
$$

1. Construct a parse tree for *cdcdcdbababcd*.



2. Is the grammar ambiguous? Explain why, and construct an unambiguous CFG generating the same language.

   Yes, it's ambiguous; here is another parse tree for the same string:



   An unambiguous CFG generating the same language is obtained by changing $A \rightarrow DE$ into $A \rightarrow E$, or $E \rightarrow Dc$ into $E \rightarrow c$ (but not both).

3. Write *cdcdcdbababcd* as $uvwxy$ such that $|u| > 0$, $|vx| > 0$, and for $i \geq 0$ also uv^iwx^iy is in the language generated by the CFG.

cdc	dcd	b	$abab$	cd
u	v	w	x	y

(take subtrees sprouting from B in parse tree).

Problem 6 (30 points)

1. Give a context-free grammar generating the set of all strings of balanced parenthesis [and]. In such a string each "[" has a matching "]", and pairs of matching parenthesis are properly nested.

$$\begin{array}{l} S \to \Lambda \\ S \to [S]S \end{array} \qquad \text{or} \qquad \begin{array}{l} S \to \Lambda \\ S \to SS \\ S \to [S] \end{array}$$

2. Convert the context free grammar from 1. into Chomsky Normal Form.

$$\begin{array}{l} S \to \Lambda \\ S \to T \\ T \to TT \\ T \to BR \\ R \to TE \\ B \to [\\ E \to] \\ R \to] \end{array}$$

Problem 7 (20 points) Prove that $\{xxxw \ : \ x \in (0 \cup 1)^+, w \in (0 \cup 1)^*\}$ is not context free.

With pumping lemma and intersection:

Intersect with the regular language $10^*10^*10^*$. It suffices to prove that the resulting language $L = \{10^n10^n10^{n+m} \mid n, m \geq 0\}$ is not context-free, since CFL's are closed under intersection with a regular language.

Let N be given. Choose $z = 10^N10^N10^N$. If $z = uvwxy$ with $|vx| > 0$, and v or x contains a 1, take $i = 0$ and uv^iwx^iy has less than three 1's and cannot be in L. In the remaining case v and x contain 0's from at most two groups.

- If $|v| = |x|$ and they take 0's from the first two groups, pump up: $uv^2wx^2y = 10^{n+|v|}10^{n+|v|}10^n \notin L$.
- In all other cases pump down: $uwy \notin L$ (either the first two groups of 0's have a different length or the last group contains less 0's then the other groups).

Problem 8 (15 points) Is the class of deterministic counter languages closed under intersection ? Prove your answer or provide a counterexample. A mere *yes* or *no* will get no points.

No: The languages $a^nb^nc^*$ and $a^*b^nc^n$ are both DCLs (easy), but their intersection $a^nb^nc^n$ is not *context free* (FB Example 5.22, p359), and hence not a DCL (as $DCL \subseteq CFL$).

Problem 9 (30 points)

A bisimulation between acceptors $P = (\alpha, \Pi, \omega)$ and $P' = (\alpha', \Pi', \omega')$ is given by a relation of representation satisfying conditions 1–12 below.

		$L(P) = L(P')$	$L(P) \subseteq L(P')$	$\widetilde{L}(P) = \widetilde{L}(P')$	$\widetilde{L}(P) \subseteq \widetilde{L}(P')$
1	$x \overset{\alpha}{\mapsto} C \Rightarrow (\exists C')[x \overset{\alpha'}{\mapsto} C'\rho C]$	$\surd$	$\surd$	$\surd$	
2	$x \overset{\alpha}{\mapsto} C \Leftarrow [x \overset{\alpha'}{\mapsto} C'\rho C]$	$\surd$			
3	$[C_0'\rho C_0 \overset{\Pi}{\mapsto} C_1] \Rightarrow (\exists C_1')[C_0' \overset{\Pi'}{\mapsto} C_1'\rho C_1]$	$\surd$	$\surd$	$\surd$	$\surd$
4	$(\exists C_0)[C_0'\rho C_0 \overset{\Pi}{\mapsto} C_1] \Leftarrow [C_0' \overset{\Pi'}{\mapsto} C_1'\rho C_1]$	$\surd$			
5	$[C_0'\rho C_0 \overset{\omega}{\mapsto} \text{ACCEPT}] \Rightarrow [C_0' \overset{\omega'}{\mapsto} \text{ACCEPT}]$	$\surd$	$\surd$	$\surd$	$\surd$
6	$(\exists C_0)[C_0'\rho C_0 \overset{\omega}{\mapsto} \text{ACCEPT}] \Leftarrow [C_0' \overset{\omega'}{\mapsto} \text{ACCEPT}]$	$\surd$			
7	$x \overset{\alpha'}{\mapsto} C' \Rightarrow (\exists C)[x \overset{\alpha}{\mapsto} C\rho^{-1}C']$			$\surd$	$\surd$
8	$x \overset{\alpha'}{\mapsto} C' \Leftarrow [x \overset{\alpha}{\mapsto} C\rho^{-1}C']$				
9	$[C_0\rho^{-1}C_0' \overset{\Pi'}{\mapsto} C_1'] \Rightarrow (\exists C_1)[C_0 \overset{\Pi}{\mapsto} C_1\rho^{-1}C_1']$			$\surd$	$\surd$
10	$(\exists C_0')[C_0\rho^{-1}C_0' \overset{\Pi'}{\mapsto} C_1'] \Leftarrow [C_0 \overset{\Pi}{\mapsto} C_1\rho^{-1}C_1']$				
11	$[C_0\rho^{-1}C_0' \overset{\omega'}{\mapsto} \text{ACCEPT}] \Rightarrow [C_0 \overset{\omega}{\mapsto} \text{ACCEPT}]$			$\surd$	
12	$(\exists C_0')[C_0\rho^{-1}C_0' \overset{\omega'}{\mapsto} \text{ACCEPT}] \Leftarrow [C_0 \overset{\omega}{\mapsto} \text{ACCEPT}]$				

The first 6 conditions say that ρ is a lockstep simulation of P by P'; the last 6 say that its *converse relation* (pp. 21 FB) ρ^{-1} is a lockstep simulation of P' by P.

Let $\widetilde{L}(P)$ be the language accepted by an acceptor P, according to the alternative definition of acceptance given in Footnote 5 in Section 1.9 of FB (pp. 101): programs accept if *every* sequence of legal transitions leads to acceptance. A usual, $L(P)$ denotes the language accepted by P under the standard definition of acceptance: programs accept if *some* sequence of legal transitions leads to acceptance.

In the course it has been shown that if there exists a relation ρ satisfying conditions 1–6, then $L(P) = L(P')$.

(a) Which of these conditions are needed to guarantee that $L(P) \subseteq L(P')$?

In one of the homeworks you had to show that the existence of a relation satisfying all 12 conditions implies $\widetilde{L}(P) = \widetilde{L}(P')$.

(b) Which of these conditions are actually needed to guarantee this?

(c) Which of these conditions are needed to guarantee that $\widetilde{L}(P) \subseteq \widetilde{L}(P')$?

Give your answers by placing checkmarks in the table above.