CS21004 Formal Languages and Automata Theory, Spring 2010–11

Class test 2

Maximum marks: 20 | Date: April 11, 2011 | Duration: 1 hour

Roll no: ____________ **Name:** ______________________________

[*Write your answers in the question paper itself. Be brief and precise. Answer all questions.*]

1. Consider the following language over $\Sigma = \{a, b, c\}$:

$$L_1 = \{a^i (bc)^j \mid i, j \geqslant 0 \text{ and } i \geqslant j\}.$$

(a) Design a context-free grammar for L_1. Briefly describe the substrings generated by the non-terminal symbols used in your grammar. **(5)**

Solution The following grammar generates L_1.

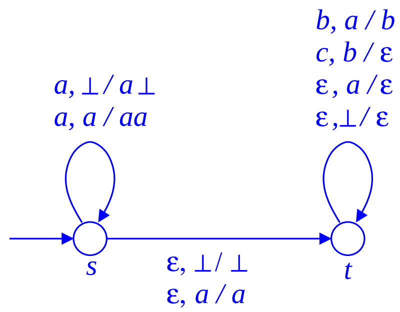
$$\begin{array}{lcll} S & \rightarrow & UV & [S \text{ is the start symbol}] \\ U & \rightarrow & \epsilon \mid aU & [U \text{ generates the excess of } a\text{'s over } (bc)\text{'s}] \\ V & \rightarrow & \epsilon \mid aVbc & [V \text{ generates equal numbers of } a\text{'s and } (bc)\text{'s}] \end{array}$$

A single start symbol S itself can generate L_1 as follows:

$$S \rightarrow \epsilon \mid aS \mid aSbc.$$

(b) Design a PDA $M = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$ to accept L_1. M must contain at most two states. Mention clearly whether your machine M accepts by final state or empty stack or both. **(7)**

Solution We take $Q = \{s, t\}$ with the start state s and no final states ($F = \emptyset$). The machine accepts by empty stack. The alphabets are $\Sigma = \{a, b, c\}$ and $\Gamma = \{a, b, \perp\}$. The transitions are described in the following figure.



In the start state s, the machine reads the initial block of a's. For every a read from the input, an a is pushed to the stack. M subsequently makes an ϵ-transition to the state t to read the block of (bc)'s, that follows the block of a's. Only if a matching a is found at the top of the stack, the reading of one occurrence of bc is initiated. The a at the top of the stack is replaced by b to indicate that the reading of bc is only half-way through. Only if a c is available at the input at this stage, this c is consumed, and the intermediate marker b is popped out of the stack exposing the next a to be matched against the next occurrence of bc.

When an input of L_1 is fully read by M, the stack contains $i - j$ occurrences of a and the bottom marker $\perp$. M uses the transitions $\epsilon, a \,/\, \epsilon$ and $\epsilon, \perp \,/\, \epsilon$ against the loop at the state t, in order to pop the excess a's and the bottom marker $\perp$. If M uses the transition $\epsilon, a \,/\, \epsilon$ more than $i - j$ times before reading all the j occurrences of bc, the machine gets stuck before reading the entire input.

2. Consider the following language over $\Sigma = \{a, b, c\}$:

$$L_2 = \{a^i b^j c^j \mid i, j \geqslant 0 \text{ and } i \geqslant j\}.$$

Prove that L_2 is not context-free. **(8)**

Solution We prove that L_2 is not context-free, using the pumping lemma for CFL's. Suppose that L_2 is context-free. Let k be a pumping-lemma constant for L_2. By definition of L_2, the string $\beta = a^k b^k c^k$ belongs to L_2 and is of length $\geqslant k$. The pumping lemma gives a decomposition $\beta = \beta_1\beta_2\beta_3\beta_4\beta_5$ such that

(i) $|\beta_2\beta_4| > 0$,
(ii) $|\beta_2\beta_3\beta_4| \leqslant k$, and
(iii) $\beta_1\beta_2^i\beta_3\beta_4^i\beta_5 \in L_2$ for all $i \geqslant 0$.

We consider all possible cases for β_2 and β_4, and show that in each case, there is a contradiction. Condition (ii) implies that β_2 and β_4 belong either to the same block or to two adjacent blocks.

Case 1: β_2 or β_4 (or both) contain different symbols. In this case, $\beta_1\beta_2^2\beta_3\beta_4^2\beta_5$ is not of the form $a^*b^*c^*$.

Case 2: Both β_2 and β_4 belong to the block of a's. In this case, $\beta_1\beta_3\beta_5$ contains less a's than b's or c's.

Case 3: Both β_2 and β_4 belong to the block of b's. In this case, $\beta_1\beta_2^2\beta_3\beta_4^2\beta_5$ contains more b's than a's.

Case 4: Both β_2 and β_4 belong to the block of c's. In this case, $\beta_1\beta_2^2\beta_3\beta_4^2\beta_5$ contains more c's than a's.

Case 5: β_2 belongs to the block of a's, and β_4 belongs to the block of b's. If $\beta_4 \neq \epsilon$, consider the string $\beta_1\beta_2^2\beta_3\beta_4^2\beta_5$ which contains more b's than c's. If $\beta_4 = \epsilon$, then $\beta_2 \neq \epsilon$, and $\beta_1\beta_3\beta_5$ contains less a's than c's.

Case 6: β_2 belongs to the block of b's, and β_4 belongs to the block of c's. In this case, $\beta_1\beta_2^2\beta_3\beta_4^2\beta_5$ contains more b's or c's (or both) than a's.