GRIFFITH UNIVERSITY
School of Computing and Information Technology
3130CIT Theory of Computation

Second mid-semester Examination, Semester 2, 2002

Answers

1. First, notice that there is no state pair that are connected directly through s. Thus, we can remove state s and related arcs and get the following DFA A_1:

	a	b
$\rightarrow p$	q	r
$* q$	r	
$* r$	r	q

For final state q, we need to eliminate r:

	$a + ba^*b$	a^+b
$\rightarrow p$	q	
$* q$		q

Transforming this to a regular expression gives:

$(a + ba^*b)(a^+b)^*$

Similarly, for final state r, we eliminate state q:

	$aa + b$	$a + ba$
$\rightarrow p$	r	
$* r$		r

Transforming this to a regular expression gives:

$(aa + b)(a + ba)^*$

Combining the two cases, we get the equivalent regular expression

$(a + ba^*b)(a^+b)^* + (aa + b)(a + ba)^*$.

2. $G = (V, T, P, S)$ where $V = \{S\}$, $T = \{a, b\}$ and P is as follows:

$S \rightarrow a \mid aS \mid bSS \mid SbS \mid SSb$

Notice that different forms of CFG can also accept the same language.

3. The idea is to compare the number of a's and the number of c's by employing stack: To push m occurences of a onto a stack and then pop the stack according to the occurences of c's. A string is accpeted if the stack sysmbol Z_0 is reached.

Here is a PDA that implements the above idea:

$A = (\{p_0, p, q, r, s\}, \{a, b, c\}, \{a, Z_0\}, \delta, p_0, Z_0, \{s\})$ where

- $\delta(p_0, a, Z_0) = \{(p, aZ_0)\}$
- $\delta(p, a, a) = \{(p, aa)\}$
- $\delta(p, b, a) = \{(q, a)\}$
- $\delta(q, b, a) = \{(q, a)\}$

- $\delta(q, c, a) = \{(r, \epsilon)\}$
- $\delta(r, c, a) = \{(r, \epsilon)\}$
- $\delta(r, \epsilon, Z_0) = \{(s, Z_0))\}$

Then A accepts L_{mn} by final state (s).

The PDA A is deterministic.

4. Suppose that L_{abc} is context-free, then there is a pumping-lemma constant n. Consider $w = a^n b^{n+1} c^{n+2}$. We may write $w = uvwxy$, where v and x, may be "pumped," and $|vwx| \leq n$.

   If vwx does not contain c's, then uv^3wx^3y has at least $n + 2$ a's or b's, and thus could not be in the language.

   If vwx contain a c, then it could not have a a, because its length is limited to n. Thus, uwy has n a's, but no more than $2n + 2$ b's and c's in total. Thus, it is not possible that uwy has more b's than a's and also has more c's than b's.

   We conclude that uwy is not in the language, and have a contradiction no matter how z is broken into $uvwxy$.

5. (a), (b), (d), (e), (f)

6. Let $H_2(P) = H_1(P, P)$, then

   Now consider $H_2(H_2)$: If $H_2(H_2)$ halts, then $H_2(H_2)$ loops forever according the definition of H_2 and similarly, if $H_2(H_2)$ loops forever, then it halts. This is a contradiction.

7. The language L_n is a bit different from the language $L'_n = \{ a^n b^n \mid n > 0 \}$, since $\epsilon \in L_n$. The textbook provides a TM for L'_n (see pp.322–324).

   To make our TM accept ϵ, we could simply have an additional arc labeled B from the initial state to the final state. However, this will also change other actions of the machine. For instance, the string 001 would be accepted by the modified TM. For this reason, we introduce a new initial state q with a transition on blank (empty input) to the final state q_4 and with a transition on a to state q_1 as in the original TM.

   Here is the transition table for the new TM:

	a	b	X	Y	B
$\rightarrow q$					(q_4, B, R)
q_0	(q_1, X, R)			(q_3, Y, R)	
q_1	$(q_1, 0, R)$	(q_2, Y, L)		(q_1, Y, R)	
q_2	$(q_2, 0, L)$		(q_0, X, R)	(q_2, Y, L)	
q_3				(q_3, Y, R)	(q_4, B, R)
$* q_4$					

8. (a) A language L is computably enumerable if there is a Turing machine M such that $L = L(M)$.

   (b) A language L is computable if there is a Turing machine M such that $L = L(M)$ and M always halts.

(c) The string $a_m \ldots a_1$ is stored on one stack and the string $b_1 \ldots b_n$ is stored on a separate stack.

(d) A queue machine has the *same* expressive power as a Turing machine (stated in lectures but not covered in the text). Nothing has more expressive power than a Turing machine, so a queue machine with two queues has the *same* expressive power as a Turing machine.