Faculty of Arts & Sciences
Department of Computer Science
Theory of Computation
Fall 2016–17
Thursday, Nov 10th, 2016

CMPS 257

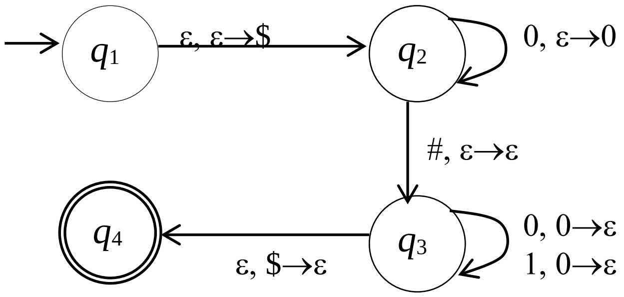
Solution of Quiz 2

Problem 1 (20 Points)

Let Σ = {0,1, #}, and consider the following language:

$A = \{\ 0^n\#u :\ |u| = n,\ n \geq 0\ \}$.

a. (10) Show that *A* is context free by giving the state diagram of a *PDA* that recognizes *A*, with as small a number of states as possible.



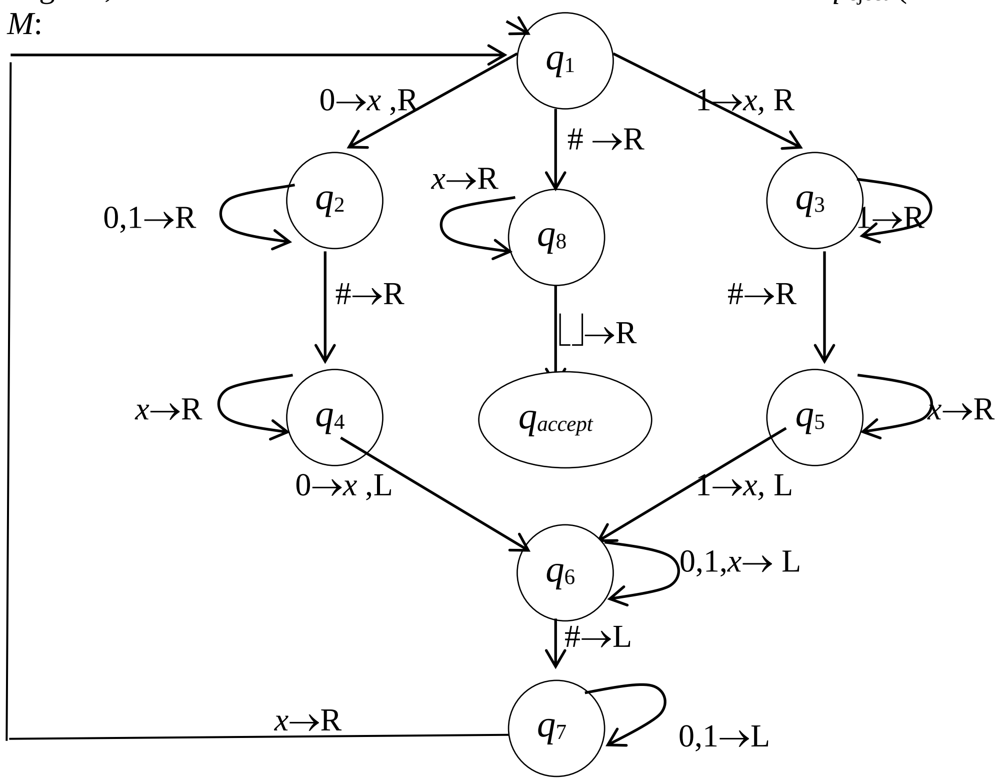
b. (10) Give a context free grammar that generates *A*, having as small a number of variable symbols as possible.

$S \to 0S0 \mid 0S1 \mid \#$

Problem 2 (20 Points)

Let Σ = {0,1, #}, and consider again the language $A = \{\ 0^n\#u :\ |u| = n,\ n \geq 0\ \}$of the previous problem. We have studied a Turing machine *M* whose state diagram is shown below, which decides $\{\ u\#u : u \in \{0,1\}^*\ \}$. In this diagram, we use the convention that "absent arrows" lead to the q_{reject} (which is not shown as well.)

M:



a. (5) Let δ denote the transition function of this Tm. Then each arrow (together with its label) represent a transition rule; e.g. the arrow from q_2 to q_4 with label #→R represents the rule $\delta(q_2, \#) = (q_4, \#, R)$. Complete the following:

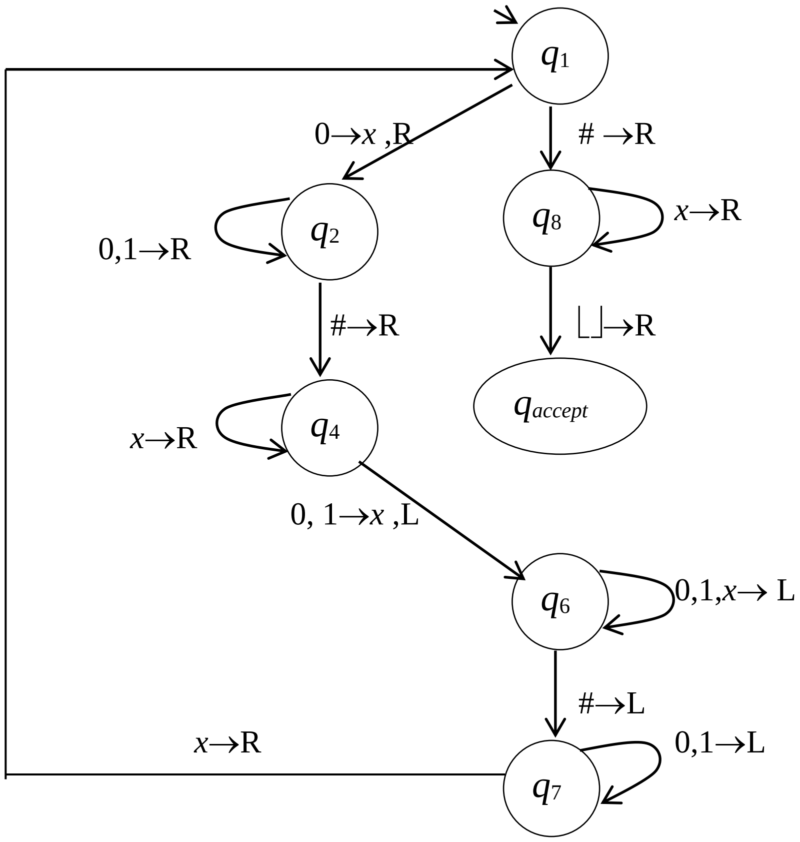
$\delta(q_4, x) = (q_4, x, R)$

$\delta(q_4, \sqcup) = (q_{reject}, \sqcup, R)$
(Note: in the above, ⊔, R are arbitrary! Important thing is to get to q_{reject})
The purpose of the rule $\delta(q_4, \sqcup)$ is to reject strings of the form: $u\#v$ where v is a proper prefix of u.
i.e. $u = vz$, where $z \neq \varepsilon$

b. Motivated by the above, make (minimal) adjustments to construct a Turing machine T that decides the language A .

i. (6) Give a "high level description" of the changes that you want to make
Two main changes:
- Since we only have 0's before the #, then q_3 and q_5 are not needed anymore, and exclude 1 from the transition rule at q_2
- Matching a 0 before the # with a 0 after the # should become matching a 0 before the # with a 0 **OR** a 1 after the # (The length of u should be equal to the number of 0's before the #)

ii. (10) Give the state diagram of your machine T



Problem 3 (30 Points)

Let $\Sigma=\{0,1\}$. Consider the context free grammar (CFG) $G(V, \Sigma, R, S)$, whose rules are:

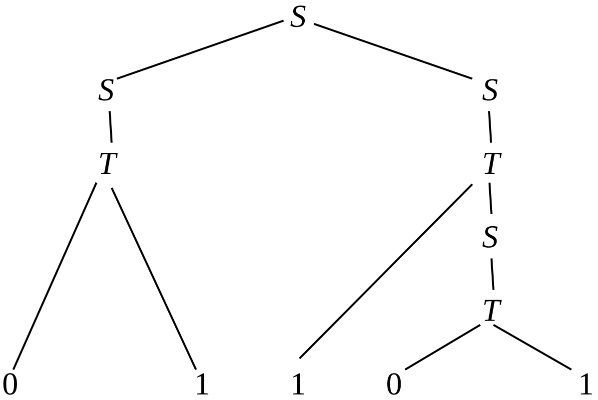
$S \rightarrow S\,S \mid T$

$T \rightarrow 0T \mid 1S \mid 01$

a. (3) According to the Pumping Theorem, the pumping length p =8. Why ?
   $p = b^{|V|+1}$, where $b = \max\{\ |u|\ ;\ A\rightarrow u$ is a rule of $G\} = \max\{2, 1, 2, 2, 2\} = 2$, and $|V| = |\ \{S, T\}\ | = 2$
   So $p = 2^{2+1} = 2^3$=8.

b. (4) Suggest a minimal height for a parse tree that will guarantee the repetition of one of the variables on a path from the root to a leaf node.
   It should be $|V| + 1 = 2+1 = 3$.

Consider the string s = 01101. $s\in L$. and a parse tree for it is :



c. (5) Clearly, repetition of T occurs on the path <S, S, T, S, T, 0>. Based on this repetition, and as in the proof of the Pumping Theorem, we should get a subdivision of s into 5 substrings, $s = uvxyz$, such that $uv^ixy^iz\in L$, for all i.= 0, 1, 2, … What are these 5 substrings?
   u = 01, v = 1, x = 01 , $y = \varepsilon$, $z = \varepsilon$

d. (3) Suggest another decomposition of s into 5 substrings that allows for pumping, this time based on the repetition of the symbol S at the root, and in between the two nodes labeled by T .
   $u = \varepsilon$, v =011, $x = x$ = 01 , $y = \varepsilon$, $z = \varepsilon$

e. (5) Give the leftmost derivation that corresponds to the parse tree above.
   $S \Rightarrow SS \Rightarrow ST \Rightarrow 01S \Rightarrow 01T \Rightarrow 011S \Rightarrow 011T \Rightarrow 01101$

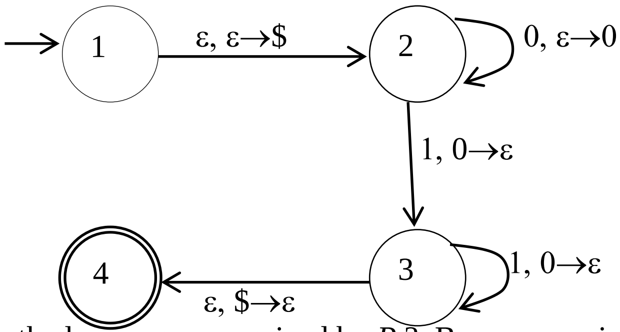
f. (4) The number of **interior** nodes of the parse tree shown above, and the number of steps in the derivation in € must be the same. Why?
   Starting with S
   each interior node of the parse tree labeled with a A and whose children constitute u where $A\rightarrow u$
   corresponds to
   a step in the derivation in which the leftmost symbol A was replaced by u
   and vice-versa.

   Thus the number of interior nodes = the number of steps in the leftmost derivation

g. (6) Using the string s show that the grammar is ambiguous.
   We give a second left most derivation of *s, and thus s is ambiguously derived, so G is ambiguous*:
   $S \Rightarrow T \Rightarrow 0T \Rightarrow 01S \Rightarrow 01T \Rightarrow 011S \Rightarrow 011T \Rightarrow 01101$

Problem 4 (15 Points)
Consider the PDA P whose state diagram is:

P:



a. (4) What is $L(P)$, the language recognized by P ? Be very precise, and note that there is only one accept state..

$A = \{ 0^n 1^n : \ n \geq 1 \}$. (ε is not accepted!!)

b. (2) In Theorem 2.12 we have proved that if a language is recognized by a PDA then it is a context free language, by constructing a CFG $G(V,\Sigma, R, S)$. To facilitate the construction, three assumptions were made:

i. P has exactly one accept state
ii. P accepts a string with an empty stack
iii. Every transition rule of P is either a push or a pop operation

Does the PDA above satisfy these assumptions?

Yes all assumptions are valid

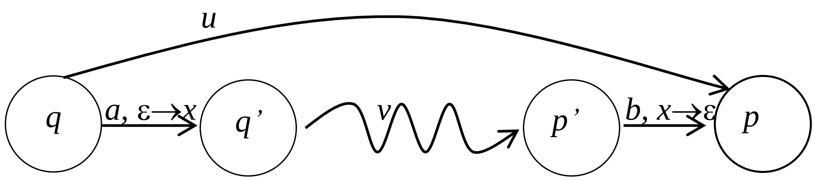
c. (3)What is the purpose of assumption (i) in the development of the proof?

Assumption i is to simplify the choice of the start symbol from among the variable symbols:
$S = A_{sf}$, where s is the start state of the PDA and f is the accept state (there is only one accept state!!)

d. (6) The construction will yield the grammar $G(V,\Sigma, R, S)$ with the following:

$V = \{ A_{11}, A_{12}, A_{13}, A_{14}, A_{21}, A_{22}, A_{23}, A_{24}, A_{31}, A_{32}, A_{33}, A_{34}, A_{41}, A_{42}, A_{43}, A_{44} \}$ $S = A_{14}$

The generic diagram below has 3 instances, as in the table.:



	q	q'	p'	p	a	B	x	$A_{qp} \to a\, A_{q'p'}\, b$
1	1	2	3	4	ε	ε	$	$A_{14} \to A_{23}$
2	2	2	2	3	0	1	0	$A_{23} \to 0\, A_{22}\, 1$
3	2	2	3	3	0	1	0	$A_{23} \to 0\, A_{23}\, 1$

The rules to be added (3 of them) appear in the last column, with the details as “to how”are in the previous columns.

In addition, there is a rule $A_{22} \to \varepsilon$. So the grammar essentially has the following 4 rules:

$A_{14} \to A_{23}$ $A_{23} \to 0\, A_{22}\, 1$ $A_{23} \to 0\, A_{23}\, 1$ $A_{22} \to \varepsilon$.

Problem 5 (15 Points)

Let $\Sigma = \{0,1\}$, and consider the language $D = \{ 0^k \mid k = n^2, n>0 \}$, *i.e. the number of 0 is a perfect square* i.e. k = 0, 1, 4, 9, 16, ….

a. (4) Show that any infinite context free language L must have a sequence of strings whose length form an arithmetic progression of the form $a + id$, i = 0, 1, 2, …. Where $0 < d \le p$, p being the pumping length.
Let p be the pumping length for L. Since L is infinite, there must be a string s in L such that $|s| > p$. The pumping Lemma applies to s. So there must be a decomposition $s = uvxyz$ such that $uv^i xy^i z$ is in L for all i=0, 1, 2, 3,… and $|vy| > 0$, and $|vxy| \le p$. Let $a = |uxz|$ and $d = |vy|$. Then $0 < d \le p$, and the sequence of strings $s_0, s_1, s_2, \ldots, s_i, \ldots$ where $s_i = uv^i xy^i z$, i=0, 1, 2, 3,…will be a sequence of strings in L whose lengths are a, $a+d$, $a+2d$, …. $a+id$, ... which is an arithmetic progression as required.

b. (4) Using (a) argue that D cannot be context free
If *D is a context free language, and being infinite, then it must contain a sequence of strings whose lengths form an arithmetic progression.* But this cannot happen, because the string in D must have lengths that are perfect squares. The sequence of perfect squares is 1, 4, 9, 16, … where the gap between the successive squares n^2 and $(n+1)^2$ is $2n$ which depends on n and keeps growing!! So we cannot fit in it a sequence which forms an arithmetic progression.

c. (7) Be more precise, and argue using the pumping lemma that D is not context free.
(Hint: What is the next perfect square that comes after p^2 ?)

Assume D is context free and let p be the pumping length. Let $s = 0^k$ where $k = p^2$. $|s| = p^2 > p$. So by the pumping lemma there must be a decomposition of $s = uvxyz$ such that the three conditions hold. Now pumping with i=2 , $uv^2 xy^2 z = 0^{k+d,}$ where $d = |vy|$. But $|vxy| \le p$, so $d=|vy| \le |vxy| \le p$, and $|vy| > 0$ since not both v and y are empty. So $0 < d \le p$. $k+d = p^2 + d \le p^2 + p < p^2 + p + p + 1 = (p+1)^2$. So $k+d = p^2 + d$ cannot be a perfect square being "sandwiched" between two successive perfect squares: $p^2 < p^2 + d < (p+1)^2$. Thus $uv^2 xy^2 z = 0^{k+d} \notin D$. Contradiction!!

Problem 6 (10 Points)

Let $\Sigma = \{0,1\}$, and consider the language $D = \{ 0^k \mid k = n^2, n > 0 \}$,
i.e. the number of 0's is a perfect square i.e. k = 0, 1, 4, 9, 16, ….
Give a high level description of a Turing machine that decides D.
Hint: You may want to use the Turing machine M that decides $C = \{a^i b^j c^k \mid i \times j = k$, and $i, j, k \ge 1\}$ repeatedly!!

T = "On input 0^k,

1. If the input is ε or 0 then <u>*accept*</u>. $\{ k = 0$, or $k = 1\}$
2. Let $m = \lfloor k/2 \rfloor$
3. Transform the input to become $0^{m-1}abc^k$,
4. If there are no 0's left, then <u>*reject*</u>
5. Cross off a 0 from the left.
6. Increment the a's and increment the b's each by 1.
7. Reposition the head at the first a.
8. Run M. If M accepts, then <u>*accept*</u>
9. Restore the a's, b's and c's and go to step 4."

How can we get $k/2$: Cross off every other 0!!