COS 487
Fall 2003

Problem Set 3 Solutions

Problem 1:

(a:) Variables: R,S,T,X. Start variable: R.
(b:) ab, ba, aab.
(c:) a, b, aa.
(d:) False.
(e:) True.
(f:) False.
(g:) True.
(h:) True.
(i:) False.
(j:) True.
(k:) True.
(l:) False.
(m:) L(G) contains all strings which are not palindromes.

Problem 2:

(b:) S -> aSb | bY | Ya
Y -> bY | aY | ϵ

(d:) S -> T | Q#T | T#Q | Q#T#Q
T -> P | aTa | bTb | # | #Q#
P -> a | b | aPa | bPb | ϵ
R -> aR | bR | ϵ

Problem 3:

(b:) As long as the next character of the input stream is an "a", read it and push an "a" on the stack. Then, as long as the next character of the input stream is a "b", read it and pop an element from the stack. Accept if another "a" is ever read, if we ever try to pop from an empty stack, or if all of the input is read and the stack is nonempty.

(d:) Nondeterministically guess if the input contains a palindrome or if the string contains $x_i = x_j^R$, $i \neq j$.

If the PDA guessed that the input contains a palindrome, nondeterministically pick an x_k, push the symbols of x_k onto the stack until nondeterministically guessing that this is the midpoint. Then pop the symbols one by one, comparing them to the input. Accept if all match, the stack empties, no # is read during this procedure, and the next symbol of the input is a #.

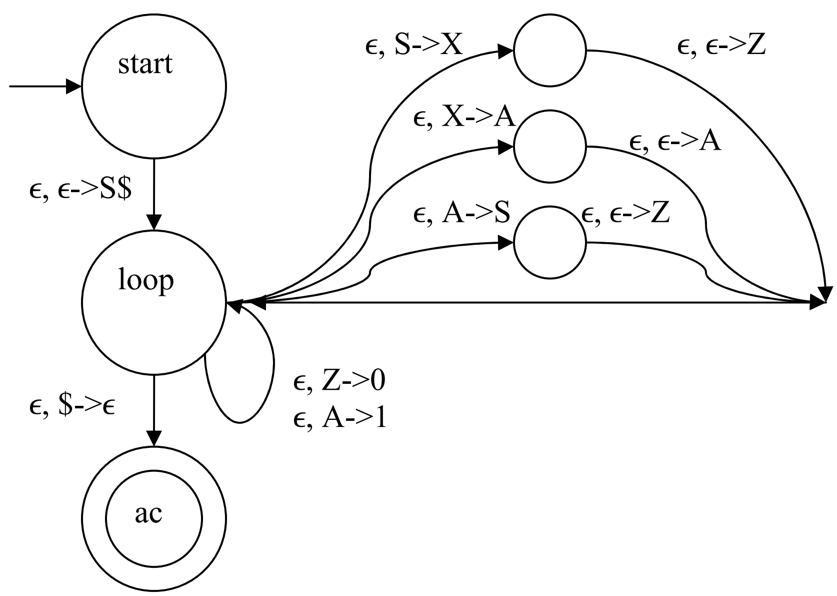
Otherwise, nondeterministically skip to an x_i in the input, and push all of x_i onto the stack, stopping when a # is read. Then, nodeterministically skip to a later x_j, pop the symbols from the stack one by one, and compare them to the input. Accept if all match, and the stack empties.

Problem 4:

(a:) L(G) contains strings which have exactly two # marks, as well as strings which contain some number of zeroes, one pound mark, and twice that number of zeroes.

(b:) Suppose L(G) is regular with pumping length p. Then, applying the pumping lemma to $0^p\#0^{2p}$ shows that $0^{p+i}\#0^{2p}$ must also be in L(G) for i > 0, a contradiction.

Problem 5:



Problem 6:

L(G) is the complement of the language $\{a^n b^n \mid n \geq 0\}$.

Problem 7:

We give a CFG for this language.

```
S -> A1BX | A0CX | D
B -> XBX | X#A0     // Left side is 1 in a position where the right side is 0.
C -> XCX | X#A1     // Right side is 1 in a position where the left side is 0.
D -> R | L | XDX    // The strings on either side are of different lengths.
L -> X# | XL
R-> #X | RX
A -> XA | ϵ         // Anything
X -> 0 | 1
```