CS 381 Homework 9 solutions

October 26, 2000

Problem 21.1 Construct a PDA that accepts the set of strings in $\{a, b\}^*$ with equally many a's and b's.
Here is a one-state automata that accepts by empty-stack:

$$\begin{aligned}
\delta(q, a, \bot) &= (q, A\bot) \\
\delta(q, b, \bot) &= (q, B\bot) \\
\delta(q, a, A) &= (q, AA) \\
\delta(q, a, B) &= (q, \epsilon) \\
\delta(q, b, A) &= (q, \epsilon) \\
\delta(q, b, B) &= (q, BB) \\
\delta(q, \epsilon, \bot) &= (q, \epsilon)
\end{aligned}$$

Problem 21.2 Give an NPDA accepting the set of all palindeomes over $\{a, b\} = \mathrm{PAL}$

We have 2 states: the first reads in and stores the input, until we non-deterministically decide we've reached the halfway point. Then the second state checks that we correctly guessed the halfway point. (Note that there are 17 transitions; we can eliminate a lot of redundancy if we consider transitions of the form $\delta(q, \sigma, \epsilon) = (p, \gamma)$ where the ϵ indicates we don't care what's on the stack; it is ignored and not popped)

$\delta(s, a, \bot) = (s, A)$	
$\delta(s, b, \bot) = (s, B)$	
$\delta(s, a, A) = (s, AA)$	
$\delta(s, a, B) = (s, AB)$	
$\delta(s, b, A) = (s, BA)$	
$\delta(s, b, B) = (s, BB)$	
$\delta(s, a, A) = (t, A)$	string is odd-length, middle is 'a'
$\delta(s, a, B) = (t, B)$	string is odd-length, middle is 'a'
$\delta(s, a, \bot) = (t, \epsilon)$	string is 'a'
$\delta(s, b, A) = (t, A)$	string is odd-length, middle is 'b'
$\delta(s, b, B) = (t, B)$	string is odd-length, middle is 'b'
$\delta(s, b, \bot) = (t, \epsilon)$	string is 'b'
$\delta(s, a, A) = (t, \epsilon)$	string is even-length, middle is 'aa'
$\delta(s, b, B) = (t, \epsilon)$	string is even-length, middle is 'bb'
$\delta(s, \epsilon, \bot) = (t, \epsilon)$	
$\delta(t, a, A) = (t, \epsilon)$	
$\delta(t, b, B) = (t, \epsilon)$	

Problem 22.1 Give a CFG G in GNF that generates $\mathrm{PAL} - \{\epsilon\}$. Convert G into an NPDA.

$$\begin{aligned} S &\to aSA|bSB|aA|bB|a|b \\ A &\to a \\ B &\to b \end{aligned}$$

$$\begin{aligned} \delta(s,\epsilon,\perp) &= (s,S) \\ \delta(s,a,S) &= (s,SA) \\ \delta(s,a,S) &= (s,A) \\ \delta(s,a,S) &= (s,\epsilon) \\ \delta(s,b,S) &= (s,SB) \\ \delta(s,b,S) &= (s,B) \\ \delta(s,b,S) &= (s,\epsilon) \\ \delta(s,a,A) &= (s,\epsilon) \\ \delta(s,b,B) &= (s,\epsilon) \end{aligned}$$

$(s, abba, \perp) \to (s, abba, S) \to (s, bba, SA) \to (s, ba, BA) \to (s, a, A) \to (s, \epsilon, \epsilon)$
$(s, ababa, \perp) \to (a, ababa, S) \to (s, baba, SA) \to (s, aba, SBA) \to (s, ba, BA) \to (s, a, A) \to (s, \epsilon, \epsilon)$
$(s, aabb, \perp) \to (s, aabb, S) \to (s, abb, \epsilon) \not\to$
$(s, aabb, \perp) \to (s, aabb, S) \to (s, abb, A) \to (s, bb, \epsilon) \not\to$
$(s, aabb, \perp) \to (s, aabb, S) \to (s, abb, SA) \to (s, bb, AA) \not\to$
$(s, aabb, \perp) \to (s, aabb, S) \to (s, abb, SA) \to (s, bb, SAA) \to (s, b, BAA) \to (s, \epsilon, AA) \not\to$
$(s, aabb, \perp) \to (s, aabb, S) \to (s, abb, SA) \to (s, bb, SAA) \to (s, b, SBAA) \to (s, \epsilon, BAA) \not\to$
$aabb$ is not accepted because there is no valid tracing $(s, aabb, \perp) \to^* (s, \epsilon, \epsilon)$

Problem 23.1

Following the algorithm, we get the following one-state NPDA (each set of rules comes from a single rule above, in order):

$$\begin{aligned} \delta'(*, a, \langle s \perp s\rangle) &= (*, \langle sAs\rangle) \\ \delta'(*, a, \langle s \perp t\rangle) &= (*, \langle sAt\rangle) \end{aligned}$$

$$\begin{aligned} \delta'(*, b, \langle s \perp s\rangle) &= (*, \langle sBs\rangle) \\ \delta'(*, b, \langle s \perp t\rangle) &= (*, \langle sBt\rangle) \end{aligned}$$

$$\begin{aligned} \delta'(*, a, \langle sAs\rangle) &= (*, \langle sAs\rangle\langle sAs\rangle) \\ \delta'(*, a, \langle sAs\rangle) &= (*, \langle sAt\rangle\langle tAs\rangle) \\ \delta'(*, a, \langle sAt\rangle) &= (*, \langle sAs\rangle\langle sAt\rangle) \\ \delta'(*, a, \langle sAt\rangle) &= (*, \langle sAt\rangle\langle tAt\rangle) \end{aligned}$$

$$\begin{aligned} \delta'(*, a, \langle sBs\rangle) &= (*, \langle sAs\rangle\langle sBs\rangle) \\ \delta'(*, a, \langle sBs\rangle) &= (*, \langle sAt\rangle\langle tBs\rangle) \\ \delta'(*, a, \langle sBt\rangle) &= (*, \langle sAs\rangle\langle sBt\rangle) \\ \delta'(*, a, \langle sBt\rangle) &= (*, \langle sAt\rangle\langle tBt\rangle) \end{aligned}$$

$$\delta'(*, b, \langle sAs\rangle) = (*, \langle sBs\rangle\langle sAs\rangle)$$
$$\delta'(*, b, \langle sAs\rangle) = (*, \langle sBt\rangle\langle tAs\rangle)$$
$$\delta'(*, b, \langle sAt\rangle) = (*, \langle sBs\rangle\langle sAt\rangle)$$
$$\delta'(*, b, \langle sAt\rangle) = (*, \langle sBt\rangle\langle tAt\rangle)$$

$$\delta'(*, b, \langle sBs\rangle) = (*, \langle sBs\rangle\langle sBs\rangle)$$
$$\delta'(*, b, \langle sBs\rangle) = (*, \langle sBt\rangle\langle tBs\rangle)$$
$$\delta'(*, b, \langle sBt\rangle) = (*, \langle sBs\rangle\langle sBt\rangle)$$
$$\delta'(*, b, \langle sBt\rangle) = (*, \langle sBt\rangle\langle tBt\rangle)$$

$$\delta'(*, a, \langle sAs\rangle) = (*, \langle tAs\rangle)$$
$$\delta'(*, a, \langle sAt\rangle) = (*, \langle tAt\rangle)$$

$$\delta'(*, a, \langle sBs\rangle) = (*, \langle tBs\rangle)$$
$$\delta'(*, a, \langle sBt\rangle) = (*, \langle tBt\rangle)$$

$$\delta'(*, a, \langle s \perp t\rangle) = (*, \epsilon)$$

$$\delta'(*, b, \langle sAs\rangle) = (*, \langle tAs\rangle)$$
$$\delta'(*, b, \langle sAt\rangle) = (*, \langle tAt\rangle)$$

$$\delta'(*, b, \langle sBs\rangle) = (*, \langle tBs\rangle)$$
$$\delta'(*, b, \langle sBt\rangle) = (*, \langle tBt\rangle)$$

$$\delta'(*, b, \langle s \perp t\rangle) = (*, \epsilon)$$

$$\delta'(*, a, \langle sAt\rangle) = (*, \epsilon)$$

$$\delta'(*, b, \langle sBt\rangle) = (*, \epsilon)$$

$$\delta'(*, \epsilon, \langle s \perp t\rangle) = (*, \epsilon)$$

$$\delta'(*, a, \langle tAt\rangle) = (*, \epsilon)$$

$$\delta'(*, b, \langle tBt\rangle) = (*, \epsilon)$$

A grammar for the above (after tossing out impossible rules, like $\langle tBs\rangle$, and unreachable rules, like $\langle s \perp s\rangle$):

$$\begin{aligned}
S &\to \langle s \perp t\rangle \\
\langle s \perp t\rangle &\to a\langle sAt\rangle | b\langle sBt\rangle | a | b | \epsilon \\
\langle sAt\rangle &\to a\langle sAs\rangle\langle sAt\rangle | a\langle sAt\rangle\langle tAt\rangle | b\langle sBs\rangle\langle sBt\rangle | b\langle sBt\rangle\langle tAt\rangle | a\langle tAt\rangle | b\langle tAt\rangle | a \\
\langle sBt\rangle &\to a\langle sAs\rangle\langle sBt\rangle | a\langle sAt\rangle\langle tBt\rangle | b\langle sBs\rangle\langle sBt\rangle | b\langle sBt\rangle\langle tBt\rangle | a\langle tBt\rangle | b\langle tBt\rangle | b \\
\langle sAs\rangle &\to a\langle sAs\rangle\langle sAs\rangle | b\langle sBs\rangle\langle sAs\rangle \\
\langle sBs\rangle &\to a\langle sAs\rangle\langle sBs\rangle | b\langle sBs\rangle\langle sBs\rangle \\
\langle tAt\rangle &\to a \\
\langle tBt\rangle &\to b
\end{aligned}$$

$$(*, ababa, \langle s \perp t\rangle) \to (*, baba, \langle sAt\rangle) \to (*, aba, \langle sBt\rangle\langle tAt\rangle) \to (*, ba, \langle tBt\rangle\langle tAt\rangle) \to (*, a, \langle tAt\rangle) \to (*, \epsilon, \epsilon)$$