Gradiance Online Accelerated Learning

Zayd

- Home Page
- Assignments Due
- Progress Report
- Handouts
- Tutorials
- Homeworks
- Lab Projects
- Log Out

Help

Submission number:	71226
Submission certificate:	HH147445
Submission time:	2014-03-22 17:00:52 PST (GMT - 8:00)

Number of questions:	6
Positive points per question:	3.0
Negative points per question:	1.0
Your score:	18

Based on Chapter 6 of HMU.

1. Here are the transitions of a determininstic pushdown automaton. The start state is q_0, and *f* is the accepting state.

State-Symbol	a	b	ε
q_0-Z_0	(q_1,AAZ_0)	(q_2,BZ_0)	(f,ε)
q_1-A	(q_1,AAA)	(q_1,ε)	-
q_1-Z_0	-	-	(q_0,Z_0)
q_2-B	(q_3,ε)	(q_2,BB)	-
q_2-Z_0	-	-	(q_0,Z_0)
q_3-B	-	-	(q_2,ε)
q_3-Z_0	-	-	(q_1,AZ_0)

Describe informally what this PDA does. Then, identify below the one input string that the PDA accepts.

a) bbbaabbb

b) bbaa

c) bababbaa

d) babbba

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

This PDA accepts all strings with twice as many *b*'s as *a*'s. In states q_0 and q_1, we push two A's onto the stack for each input *a*, and we pop an A for every input *b*.

push two A's onto the stack for each input a, and we pop an A for every input b. You can interpret state q_1 as saying "we've seen more than half as many a's as b's." In states q_0 and q_2 we push a B for every input b, and (with the help of q_3) we pop two B's for every input a (using q_3 as an intermediate. You can interpret q_2 as "we have seen more than twice as many b's as a's."

2. If we convert the context-free grammar G:

```
S → AS | A
A → 0A | 1B | 1
B → 0B | 0
```

to a pushdown automaton that accepts L(G) by empty stack, using the construction of Section 6.3.1, which of the following would be a rule of the PDA?

a) $\delta(q,\varepsilon,B) = \{(q,0B)\}$

b) $\delta(q,0,B) = \{(q,B), (q,\varepsilon)\}$

c) $\delta(q,\varepsilon,S) = \{(q,AS)\}$

d) $\delta(q,0,0) = \{(q,\varepsilon)\}$

Answer submitted: **d)**

You have answered the question correctly.

Question Explanation:

There is one state, q. The input symbols are 0 and 1, and the stack symbols are {S, A, B, 0, 1}. S is the initial stack symbol. The rules are:

$\delta(q,\varepsilon,S) = \{(q,AS), (q,A)\}$

$\delta(q,\varepsilon,A) = \{(q,0A), (q,1B), (q,1)\}$
$\delta(q,\varepsilon,B) = \{(q,0B), (q,0)\}$
$\delta(q,0,0) = \{(q,\varepsilon)\}$
$\delta(q,1,1) = \{(q,\varepsilon)\}$

3. Suppose one transition rule of some PDA P is $\delta(q,0,X) = \{(p,YZ), (r,XY)\}$. If we convert PDA P to an equivalent context-free grammar G in the manner described in Section 6.3.2 (p. 247), which of the following could be a production of G derived from this transition rule? You may assume s and t are states of P, as well as p, q, and r.

a) $[qXq] \rightarrow 0[rXr][rYq]$

b) $[qXq] \rightarrow 0[rXr][sYq]$

c) $[qXq] \rightarrow [rXr][rYq]$

d) $[qXq] \rightarrow 0[qYr][rZp]$

Answer submitted: **a)**

You have answered the question correctly.

Question Explanation:

If m and n are any states of P, then the fact that (p,YZ) is in δ(q,0,X) says that there will be a production [qXm] → 0[pYn][nZm]. Similarly, the choice (r,XY) says that [qXm] → 0[rXn][nYm] is a production.

4. Consider the pushdown automaton with the following transition rules:

1. $\delta(q,0,Z_0) = \{(q,XZ_0)\}$
2. $\delta(q,0,X) = \{(q,XX)\}$
3. $\delta(q,1,X) = \{(q,X)\}$
4. $\delta(q,\varepsilon,X) = \{(p,\varepsilon)\}$
5. $\delta(p,\varepsilon,X) = \{(p,\varepsilon)\}$
6. $\delta(p,1,X) = \{(p,XX)\}$
7. $\delta(p,1,Z_0) = \{(p,\varepsilon)\}$

The start state is q. For which of the following inputs can the PDA first enter state p with the input empty and the stack containing XXZ_0 [i.e., the ID (p,ε,XXZ_0)]?

a) 0111011

b) 011011011

c) 011001101

d) 111001

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

When in state q, the PDA adds an X to the stack whenever it consumes a 0. The PDA may consume a 1 with no change to the stack, but only if the stack has top symbol X. That is, on inputs beginning with 1 the PDA has no choice of move and can never enter state p. Since entering state p pops an X from the stack, there must be exactly three 0's in the consumed inputs, and any number of 1's. In addition, the first input must be 0.

5. Consider the pushdown automaton with the following transition rules:

1. $\delta(q,0,Z_0) = \{(q,XZ_0)\}$
2. $\delta(q,0,X) = \{(q,XX)\}$
3. $\delta(q,1,X) = \{(q,X)\}$
4. $\delta(q,\varepsilon,X) = \{(p,\varepsilon)\}$
5. $\delta(p,\varepsilon,X) = \{(p,\varepsilon)\}$
6. $\delta(p,1,X) = \{(p,XX)\}$
7. $\delta(p,1,Z_0) = \{(p,\varepsilon)\}$

From the ID $(p,1101,XXZ_0)$, which of the following ID's can NOT be reached?

a) $(p,01,XXXZ_0)$

b) $(p,101,Z_0)$

c) $(p, 01, XXXXXZ_0)$

d) $(p, 101, XXXZ_0)$

Answer submitted: **c)**

You have answered the question correctly.

Question Explanation:

In state p, there is no way to consume a 0 from the input, and no way to leave state p. We can pop X's from the stack spontaneously (on ε input), and by consuming a 1 we can push an X onto the stack (but only if there was already an X on the top of the stack). Finally, with Z_0 at the top of the stack and 1 as the next input, we can pop the Z_0 and consume the 1. Consequently, the accessible ID's can be categorized as follows. All have state p.

1. Input = 1101, stack is XXZ_0, XZ_0, or Z_0.
2. Input = 101, stack is $XXXZ_0$, XXZ_0, XZ_0, Z_0, or ε.
3. Input = 01, stack is $XXXXZ_0$, $XXXZ_0$, XXZ_0, XZ_0, Z_0, or ε.

6. Here are the transitions of a deterministic pushdown automaton. The start state is q_0, and f is the accepting state.

State-Symbol	a	b	ε
q_0-Z_0	(q_1,AAZ_0)	(q_2,BZ_0)	(f,ϵ)
q_1-A	(q_1,AAA)	(q_1,ϵ)	-
q_1-Z_0	-	-	(q_0,Z_0)
q_2-B	(q_3,ϵ)	(q_2,BB)	-
q_2-Z_0	-	-	(q_0,Z_0)
q_3-B	-	-	(q_2,ϵ)
q_3-Z_0	-	-	(q_1,AZ_0)

Describe informally what this PDA does. Then, identify below, the one input string that takes the PDA into state q_3 (with any stack).

a) baabba

b) abbba

c) babbabaa

d) baba

Answer submitted: **b)**

You have answered the question correctly.

Question Explanation:

This PDA accepts all strings with twice as many *b*'s as *a*'s. In states q_0 and q_1, we push two A's onto the stack for each input *a*, and we pop an A for every input *b*. You can interpret state q_1 as saying "we've seen more than half as many *a*'s as *b*'s." In states q_0 and q_2 we push a B for every input *b*, and (with the help of q_3) we pop two B's for every input *a*. You can interpret q_2 as "we have seen more than twice as many *b*'s as *a*'s."

As a result, we enter q_3 when, having previously seen strictly more than twice as many *b*'s as *a*'s, we see an *a* on the input.