**PDA ON TEST**

**a) The nodes correspond to the states of the PDA.**

**b) An arrow labeled Start indicates the start state, and doubly circled states are accepting, as for finite automata.**

**c) The arcs correspond to transitions of the PDA in the following sense. An arc labeled a, X/ from state *q* to state *p* means that (q, a, X) contains the pair (p, ), perhaps among other pairs. That is, the arc label tells what input is used, and also gives the old and new tops of the stack.**

**The only thing that the diagram does not tell us is which stack symbol is the start symbol. Conventionally, it is $, unless we indicate otherwise.**

**The diagram below shows the previous PDA represented as a generalized transition diagram.**

**q2**

**, $ / $**

**0, 0 / **

**PDA that recognizes {wwR | w in (0+1)\*}**

**1, 1 / **

**, 0 / 0**

**, 1 / 1**

**, $ / $**

**0, $ / 0$**

**0, 0 / 00**

**1, 1 / 11**

**1, $ / 1$**

**0, 1 / 01**

**1, 0 / 10**

**EXAMPLE 2.11-C (6.3) (ON TEST)**

**(I) Let’s consider the action of the PDA above on input 1111.**

**The start state is q0 and $ is the top of the stack symbol, thus, initial ID (instantaneous description) is (q0, 1111, $). On this input, the PDA has an opportunity to guess wrongly several times. The entire sequence of IDs that the PDA can reach from the initial ID (q0, 1111, $) is shown step by step below.**

**(1) From the initial ID (q0, 1111, $), there are moves as follows:**

**First: The PDA assumes that it’s already at the middle of the input and then is at q0 looking at a  input and at a $ at the bottom of the stack. It transfers to q1 and continues.**

**(q0, 1111, $) → [,$/$] → (q1, 1111, $) → [,$/$] → (q2, 1111, $) reject, the input is not ! continue iterating!**

**or**

**Second: The PDA assumes that it is one step from the middle sees a 1 coming in and a $ on top of the stack. It then pushes the 1 into the stack and stays at q0.**

**(q0, 1111, $) → [1,$/1$] → (q0, 111, 1$) → [,1/1]→ (q1, 111, 1$) → [1,1/] → (q1, 11, $) → [,$/$] → (q2, 11, $) reject, the input is not ! Continue iterating!**

**Or**

**Third: The PDA assumes that it is two steps from the middle sees a 1 coming in and a $ on top of the stack. It then pushes the 1 into the stack and stays at q0. Then it sees another 1 coming in and pushes it onto the stack and stays at q0. Now it transfers to q1 and starts popping the stack. After that it transfers to q2 with no input anymore. It accepts.**

**(q0, 1111, $) → [1,$/1$] → (q0, 111, 1$) → [,1/11]→ (q0, 11, 11$) → [,1/1] → (q1, 11, 11$) → [1,1/] → (q1, 1, 1$) → [1,1/] → (q1, , $) → [,$/$]** **→ (q2, **, $) accept! STOP. THERE IS NO NEED TO CONTINUE ITERATING!**

**(II) Let’s NOW consider the action of the PDA above on input 111.**

**As before, the start state is q0 and $ is the top of the stack symbol, thus, initial ID (instantaneous description) is (q0, 111, $). The entire sequence of IDs that the PDA can reach from the initial ID (q0, 111, $) is shown step by step below.**

**(1) From the initial ID (q0, 111, $), there are moves as follows:**

**First: The PDA assumes that it’s already at the middle of the input and then is at q0 looking at a  input and at a $ at the bottom of the stack. It transfers to q1 and continues.**

**(q0, 111, $) → [,$/$] → (q1, 111, $) → [,$/$] → (q2, 111, $) reject, the input is not ! Continue iterating!**

**Second: The PDA assumes that it is one step from the middle sees a 1 coming in and a $ on top of the stack. It then pushes the 1 into the stack and stays at q0.**

**(q0, 111, $) → [1,$/1$] → (q0, 11, 1$) → [,1/1] → (q1, 11, 1$) → [1,1/] → (q1, 1, $) → [,$/$] → (q2, 1, $) reject, the input is not ! Continue iterating!**

**Third: The PDA PDA assumes that it is two steps from the middle sees a 1 coming in and a $ on top of the stack. It then pushes the 1 into the stack and stays at q0. The process continues as shown below**

**(q0, 111, $) → [1,$/1$] → (q0, 11, 1$)→ [1,1/11] → (q0, 1, 11$) → [,1/1] → (q1, 1, 11$) → [1,1/] → (q1, , 1$) → reject, the process is stack at q1! Continue iterating!  
Fourth: From the state (q0, , 111$), there is only one last choice as follows:**

**(q0, 111, $) → [1,$/1$] → (q0, 11, 1$) → [1,1/11] → (q0, 1, 11$) → [1,1/11] → (q0, , 111$) → [,1/1] → (q1, , 111$) → reject, the process is stack at q1! End of the iterations!**