PDA Neso Academy

November 12, 2020 2:35 PM

* Whats a PDA?

-> A Pushdown Automata (PDA) is a way to implemeat a Context-Free grammar in a similar way that we design Finite Automatas for regular grammars.

-> PDA is more powerful than NFA/DFA because it has more memory.

↳ In an NFA/DFA (FSM) we had very limited memory to use and relied on the loping to represent all strings accepted by a machine/ are in the language. For example in FSM we couldn't track the the number of times some symbol appeared. Like in the assignment. Some languages that we can't represent by a FSM we now can with a PDA.

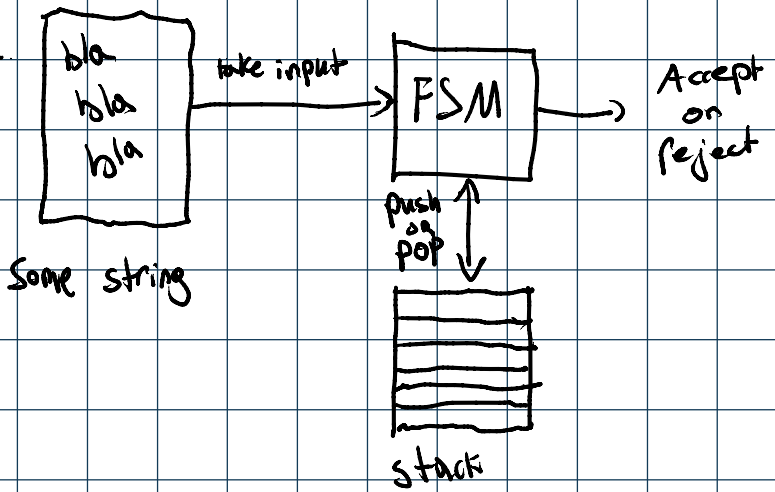
↳ A PDA is FSM + A stack. A stack has infinite memory.

* Whats a Stack?

-> A stack is the same Data structure we used in 3b2. We put elements on top of each other. Basic operations are Push() adds element. Pop() remove and return

* What does PDA consist of?

1) Input tape/string
2) A Finite control unit (NFA)
3) A stack with infinite size.



* How is a PDA formally defined?

-> We defined an NFA with the 5-tuples $(Q, \Sigma, \delta, q_0, F)$ where

Q : set of states $\{q_0, q_1, q_2\}$ etc

Σ : input alphabet $\{a, b, c\}$ $\{0, 1\}$ etc

δ : transitional functions $Q \times \Sigma \rightarrow 2^Q$. Every state can lead to multiple states given an input.

q_0 : start state $\in Q$

F : set of final states.

-> Now we define a PDA with the 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

Q : set of states

Σ : input alphabet

New Γ : A finite stack alphabet.

different δ : transitional functions. Different from DFA and NFA transition functions

q_0 : start state $\in Q$

New Z_0 : The start stack symbol

F : set of final states.

↳ δ takes 3 values as its arguments. $\delta(q, a, X)$ where:

q : A state in Q

a : Input symbol in Σ or λ

X : a stack symbol in Γ

δ returns a set of pairs (p, Y) where:

p : new state

Y : a string of stack symbols that replace X at top of stack

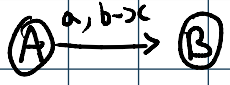
example: 1) if $Y = \lambda$, stack is empty, popped.

2) if $Y = X$, stack is unchanged

3) if $Y = YZ$, X was popped, replaced by Z, then Y pushed to the stack.

* How to graphically represent a PDA

-> FSM : 

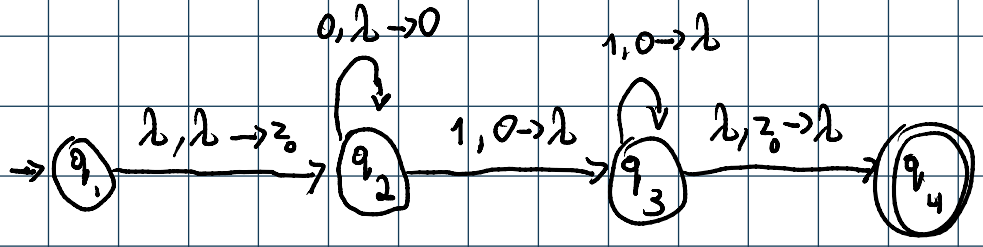
-> PDA : 

a is the input symbol, can be λ

b is the symbol on top of stack, that is always popped, if its λ stack won't read or popped.

c is the symbol that is pushed to the stack. λ means nothing is pushed.

* Example: Construct a PDA that accepts $L = \{0^n 1^n \mid n \geq 0\}$

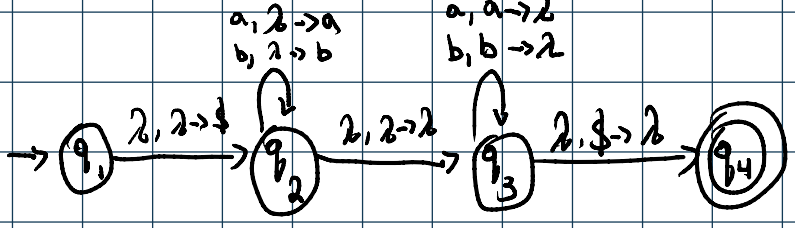


$ or Z_0 is the symbol we push first, stays at the bottom of stack so we know when we exhaust it.

λ cases to accept a string:
1) Reach final state
AND
2) stack is empty

Example: Even palindromes $L = \{ww^R \mid w = (a+b)^+\}$

$(a+b)^+$ contain at least 1 symbol that isn't λ



if we leave q_2, we assume that we've reached the midpoint of our string