


L1.0 Pushdown Automata (Introduction)

Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar 

- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar

- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

A stack is a way we arrange elements one on top of another

A stack does two basic operations:

PUSH: A new element is added at the Top of the stack

POP: The Top element of the stack is read and removed

Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar

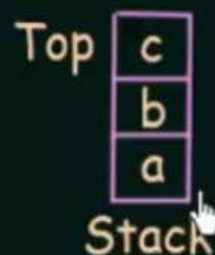
- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

A stack is a way we arrange elements one on top of another

A stack does two basic operations:

PUSH: A new element is added at the Top of the stack

POP: The Top element of the stack is read and removed



Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar

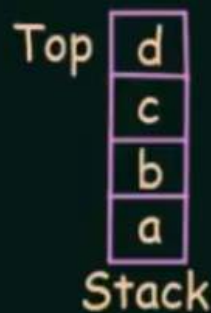
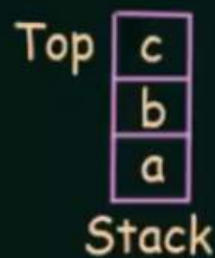
- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

A stack is a way we arrange elements one on top of another

A stack does two basic operations:

PUSH: A new element is added at the Top of the stack

POP: The Top element of the stack is read and removed



Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar

- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

A stack is a way we arrange elements one on top of another

A stack does two basic operations:

PUSH: A new element is added at the Top of the stack

POP: The Top element of the stack is read and removed



Pushdown Automata (Introduction)

A Pushdown Automata (PDA) is a way to implement a Context Free Grammar in a similar way we design Finite Automata for Regular Grammar

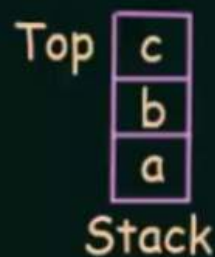
- > It is more powerful than FSM
- > FSM has a very limited memory but PDA has more memory
- > PDA = Finite State Machine + **A Stack**

A stack is a way we arrange elements one on top of another

A stack does two basic operations:

PUSH: A new element is added at the Top of the stack

POP: The Top element of the stack is read and removed



A Pushdown Automata has 3 components:

- 1) An input tape
- 2) A Finite Control Unit
- 3) A Stack with infinite size

