What is a pushdown automata?

An NFA augmented with a stack

Stack:
- Holds stack symbols (stack alphabet, $\tau$ separate from input alphabet, $\Sigma$)
- Can pop symbol from the stack
  - popping empty stack causes this computation to not accept
  - Can only retrieve topmost symbol
- Can push a stack symbol
  - Always goes on top
- No way to explicitly test whether stack is empty
  - But we've got a trick to be able to tell!

State diagram
- labels become: $a$, $b \rightarrow c$
  - means
    - reading $a$ from input
    - $a$ and top of stack is $b$
    - pop $b$
    - push $c$
Example

Language = \{0^n1^n | n \geq 0\}

Example

Language = \{w \in \{0, 1\}^* | w \text{ has equal numbers of 0's and 1's}\}
Example

Language = \{w\#w^R| w\in\{0, 1\}^*\}

Example

Language = \{ww^R| w\in\{0, 1\}^*\}
Example

Language = w not a palindrome (over \{0, 1\}*)

Example

Language: L\{xy \mid |x|=|y|, x \neq y, x, y \in \{0, 1\}*\}
**CFG can be converted to NPDA**

**Easiest to use Chomsky Normal Form**

- Book uses any grammar without conversion; idea is the same

**Let’s look at leftmost derivation with CNF**

- Example grammar
  - \( S \rightarrow TT \mid RT \)
  - \( T \rightarrow 0 \mid TT \mid 1 \)
  - \( R \rightarrow 0 \mid RR \)

- Example string
  - 01101

**CFG can be converted to NPDA**

**Without Chomsky Normal Form**

- Can have arbitrary mix of terminals and non-terminals in sentential form

- Store everything except leading terminals on stack

- Match input symbols to stack terminal symbols

- Example:
  - \( S \rightarrow 0T1 \mid 1 \)
  - \( T \rightarrow T0 \mid \epsilon \)

- Example string:
  - 0001
CFG equivalent to PDA

Still need to show can convert PDA to CFG
  • Believe me:) (In book, if you desire)