Pushdown Automata

Determinism

Deterministic PDAs

- As mentioned before
  - Our basic PDA in non-deterministic
  - We can define a Deterministic PDA (DPDA) as follows:
    - Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA
    - M is deterministic if:
      - $\delta(q, a, X)$ has at most one element
      - If $\delta(q, \varepsilon, X) \neq \emptyset$ then $\delta(q, a, X) = \emptyset$ for all $a \in \Sigma$

Deterministic PDAs

- In other words:
  - There is no configuration where the machine has a "choice" of moves
    - Each transition has at most 1 element.
    - If you can make a ε -transition from a state with a given symbol on the stack,
      - You cannot make that same transition on any tape input symbol.

Deterministic PDAs

- A language L is a deterministic context-free language (DCFL) if there is a DPA that accepts L

PDA Example

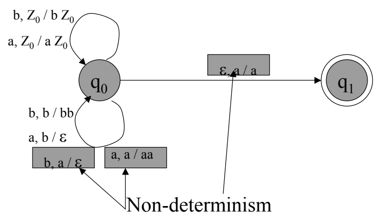
- Example:
  - $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$
  - First using a PDA:
    - Let the stack store the "excess" of one symbol over another
      - If more a's have been read than b's, a's will be on the stack, and via versa
      - If a is on the stack and you read a b, simple match the a with the b.
      - If a is on the stack and you read an a, we have one more extra a – Push it on the stack.
      - An empty stack means the number of a's and b's are equal.

PDA Example

- Example:
  - $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$
  - The PDA will have 2 states:
    - State 0 (start) : where all the work gets done
    - State 1 (accepting) : one you're in here, the machine stops.
  - The machine can "choose" to go into state 1 on a ε transition whenever an a is on the stack.

PDA Example

- Example:
  - $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$



PDA Example

- Let's try on JFLAP

PDA Example

Example:
- $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$

- Removing the non-determinism :
  - Let the stack store 1 minus the "excess" of one symbol over another
  - The state will determine whether you have excess a's or excess b's

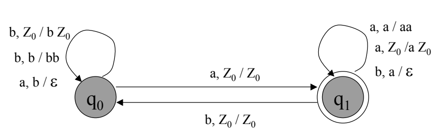
PDA Example

- Example:
  - $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$

  - The PDA will have 2 states:
    - State 0 (start) : when $n_a(x) \leq n_b(x)$
      - Equality or surplus of b's
    - State 1 (accepting) : when $n_a(x) > n_b(x)$
      - Surplus of a's

PDA Example

- Example:
  - $L = \{ x \in \{ a, b \}^* \mid n_a(x) > n_b(x) \}$

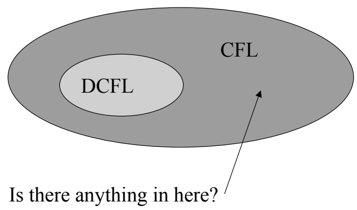


PDA Example

- Let's try on JFLAP

Now you might be wondering…

We know that all DCFLs are CFLs



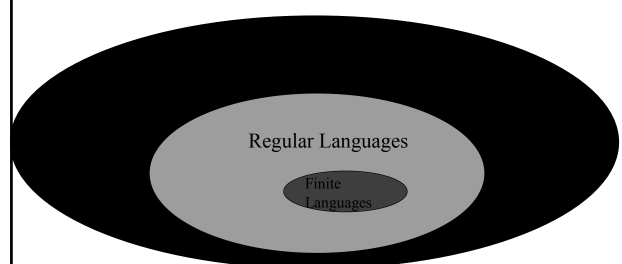
It can be shown…

- That the language pal:
  - pal = $\{ x \in \{ a, b \}^* \mid x = x^r \}$
- Cannot be accepted by any DPDA.

It can also be shown

- That all regular languages can be accepted by a DPDA.
  - Since an DFA is essentially a DPDA that doesn't make use of the stack.

Now our picture looks like



Why DPDAs are important

- A compiler may wish to implement a PDA in software to parse a program given by a given grammar
- DPDAs and ambiguity
  - If L can be accepted by a DPDA, then L can be expressed by an unambiguous CFG
  - Not visa versa
  - Theorems 6.20 / 6.21 in text

Determinism vs. Non-Determinism

- Comparing FAs and PDAs
  - DPDAs allow for ε -transitions
  - DPDAs allow for no moves

  - FAs and NFAs are equivalent
  - PDAs and DPDAs are <u>not</u> equivalent

  - Questions