Push Down Automaton

Atri Rudra

May 10

Announcement

- Mid term
  - If you have not collected yours yet, see me after/before class.

Puzzle of the day

- Design a context free grammar for the following language:
- $\{ xy \mid x,y \in \{0,1\}^* \text{ and } |x|=|y| \text{ but } x \neq y \}$

Last lecture

- Chomsky Normal form
- Grammar $G = \langle V,\Sigma,R,S \rangle$
- All the rules are of the following form
  - $S \rightarrow \varepsilon$
  - $A \rightarrow BC$ $\quad B,C \neq S$
  - $A \rightarrow a$ $\quad a \in \Sigma$

10/2 things Chomsky hates about CFGs

1. S appears on the RHS of a rule
2. There is a rule of the form $A \rightarrow XaY$
   - Either X or $Y \neq \varepsilon$
3. $A \rightarrow B_1B_2\ldots B_k$, $k>2$
4. $A \rightarrow \varepsilon$
5. $A \rightarrow B$

We will get rid of each condition one by one

Running Example

- $P \rightarrow PQP \mid Q \mid \varepsilon$
- $Q \rightarrow 00 \mid \varepsilon$

Step 1: S on RHS

- Fix : Add a new start variable S' and
  - Add rule S'→ S
- S' → P
- P→ PQP | Q | ε
- Q→ 00 | ε

Step 2: Non-single terminal on RHS

- Problem: Rule of the form A→ XaY
- Fix: Remove old rule and
  - Add a new variable Z and the rules
  - Z→ a, A→ XZY
- S' → P
- P→ PQP | Q | ε
- Q→ 00 | ε
- Z→ 0

Step 2: Non-single terminal on RHS

- Problem: Rule of the form A→ XaY
- Fix: Remove old rule and
  - Add a new variable Z and the rules
  - Z→ a, A→ XZY
- S' → P
- P→ PQP | Q | ε
- Q→ ZZ | ε
- Z→ 0

Step 3: Multiple vars. On RHS

- Problem: A→ $B_1B_2\ldots B_k$, k>2
- Fix: Remove the old rule and
  - Add new vars T_2, … T_{k-1} and the rules
  - A→ B_1T_2, T_2→ B_2T_3, … T_{k-1}→ $B_{k-1}B_k$
- S' → P
- P→ PQP | Q | ε
- Q→ ZZ | ε
- Z→ 0
- T_2→ QP

Step 3: Multiple vars. On RHS

- Problem: A→ $B_1B_2\ldots B_k$, k>2
- Fix: Remove the old rule and
  - Add new vars T_2, … T_{k-1} and the rules
  - A→ B_1T_2, T_2→ B_2T_3, … T_{k-1}→ $B_{k-1}B_k$
- S' → P
- P→ PT_2 | Q | ε
- Q→ ZZ | ε
- Z→ 0
- T_2→ QP

Step 4: A→ε

- Fix: Remove all such rules
- Patchwork is slightly more involved
- S' → P
- P→ PT_2 | Q | ε
- Q→ ZZ | ε
- Z→ 0
- T_2→ QP

What are we losing ?

- Maybe $S' \Rightarrow^* \varepsilon$
- If $A \rightarrow BC$ and $B \rightarrow \varepsilon$
  - Earlier: $A \Rightarrow^* C$
  - Now: no longer possible
- The fix: compute $\mathcal{E}$
  - Set of variables that can derive ε
  - If $A \rightarrow \varepsilon$, then put A in $\mathcal{E}$
  - If $B \rightarrow w$, $w \in \mathcal{E}^*$, then put B in $\mathcal{E}$

After computing $\mathcal{E}$

- If $S' \in \mathcal{E}$, then add rule
  - $S' \rightarrow \varepsilon$
- For rule $A \rightarrow BC$ such that $C \in \mathcal{E}$
  - Add rule $A \rightarrow B$
- For rule $A \rightarrow BC$ such that $B \in \mathcal{E}$
  - Add rule $A \rightarrow C$

Applying the fix to our example

- $\mathcal{E} = \{ P, Q, S', T_2 \}$

- $S' \rightarrow P$
- $P \rightarrow PT_2 \mid Q \mid \varepsilon$
- $Q \rightarrow ZZ \mid \varepsilon$
- $Z \rightarrow 0$
- $T_2 \rightarrow QP$

Removing the $A \rightarrow \varepsilon$ rules

- $\mathcal{E} = \{ P, Q, S', T_2 \}$

- $S' \rightarrow P$
- $P \rightarrow PT_2 \mid Q$
- $Q \rightarrow ZZ$
- $Z \rightarrow 0$
- $T_2 \rightarrow QP$

The patchwork

- $\mathcal{E} = \{ P, Q, S', T_2 \}$

- $S' \rightarrow \varepsilon$
- $S' \rightarrow P$
- $P \rightarrow PT_2 \mid Q$
- $Q \rightarrow ZZ$
- $Z \rightarrow 0$
- $T_2 \rightarrow QP$

The patchwork

- $\mathcal{E} = \{ P, Q, S', T_2 \}$

- $S' \rightarrow \varepsilon$
- $S' \rightarrow P$
- $P \rightarrow PT_2 \mid Q \mid P \mid T_2$
- $Q \rightarrow ZZ$
- $Z \rightarrow 0$
- $T_2 \rightarrow QP \mid Q \mid P$

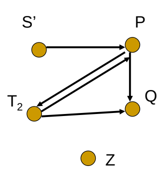
Step 5: A→ B

- Fix: Remove the *unit* rules
- What do we lose ?
  - Say A→ B and B→ X (X is not a single variable)
  - Earlier: A⇒* X
  - Now: not possible
- Patchwork
  - If we knew that A can reach B using unit rules
  - Add a new rule A→ X

Computing the reachability

- Build a directed graph
  - Each node is a variable
  - If A→ B then edge from A to B
- Denote $\mathcal{D}$(A)
  - Set of vars reachable from A in the above graph
- If B→ X and B∈$\mathcal{D}$(A) then add
  - X is *interesting*
  - A→ X

Back to the example



- S'→ ε
- S' → P
- P→ PT_2 | Q | P | T_2
- Q→ ZZ
- Z→ 0
- T_2→ QP | Q | P
- $\mathcal{D}$(P) ={ Q,T_2 }
- $\mathcal{D}$(T_2}={Q,P}, $\mathcal{D}$(S')={P,Q,T_2}, $\mathcal{D}$(Q)=$\mathcal{D}$(Z)=∅

Removing unit rules

- S'→ ε
- S' →
- P→ PT_2 |
- Q→ ZZ
- Z→ 0
- T_2→ QP |
- $\mathcal{D}$(P) ={ Q,T_2 }
- $\mathcal{D}$(T_2}={Q,P}, $\mathcal{D}$(S')={P,Q,T_2}, $\mathcal{D}$(Q)=$\mathcal{D}$(Z)=∅

Completing the patchwork

- S'→ ε
- S' → PT_2 | ZZ | QP
- P→ PT_2 | QP | ZZ
- Q→ ZZ
- Z→ 0
- T_2→ QP | PT_2 | ZZ
- $\mathcal{D}$(P) ={ Q,T_2 }
- $\mathcal{D}$(T_2}={Q,P}, $\mathcal{D}$(S')={P,Q,T_2}, $\mathcal{D}$(Q)=$\mathcal{D}$(Z)=∅

Chomsky is finally happy

- S'→ ε
- S' → PT_2 | ZZ | QP
- P→ PT_2 | QP | ZZ
- Q→ ZZ
- Z→ 0
- T_2→ QP | PT_2 | ZZ

Questions ?

Up next…

- A new model for context free languages
- It is a "machine"
- Why do we need two models ?

$\{0^n1^n \mid n \geq 0\}$

- Problem with DFAs
  - Have finite memory
- The fix: allow machines to have infinite memory
- Machine for the above language:
  - Write all 0s to the tape
  - Once you start seeing 1s
    - For each 1 mark off a 0
  - At the end, check if all 0s are marked off

All is fine…

- Went from DFAs to new machines
- In the Simpson language

DFAs

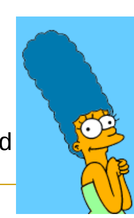
Turing Machines

So Marge is mad…

Let's see how Marge can do it

- What is her most distinctive feature ?
- Her "stack" of hair
- When she sees a 0
  - She sticks it in her hair
- When she sees a 1
  - She pulls the top 0 out
  - Discards both the 0 and the 1
- Accept is no 0s left at the end

0 0 1 1



What the heck is he talking about ?

- Enter Push Down Automatons
  - PDAs