CS 205

Lecture 20

2-3-21

Ambiguous grammar : CFG G with two different parse trees for some $w \in L(G)$.

Decision Problems

(1) given a grammar G is G ambiguous?

(2) given CFG G find a CFG G' s.t. $L(G) = L(G')$ and G' is not ambiguous.

Undecidable !

Inherently Ambiguous Language

A language L s.t. every grammar for L is ambiguous.

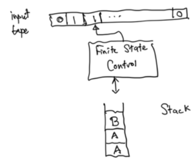
Ex $L = \{a^n b^n c^m d^m \mid m,n \geq 1\} \cup \{a^n b^m c^m d^n \mid m,n \geq 1\}$

Fact : Every CFG for L will have two (or more) derivations for strings $a^n b^n c^n d^n$.

So far : CFL/CFG, derivations, parse trees, ambiguity

Pushdown Automata (PDA)

Extensions of ε-NFA with a stack for storage



Stack size is not limited.

One step of computation

: 1. Read the input symbol and TOS

2. Based on current state, input symbol and TOS

(i) Pop the TOS symbol

(ii) Push a finite strings onto the stack

- symbols pushed from right to left
- String may be ε (POP)
- Same as the one popped (Unchanged)
- or could be some other string

(iii) Change to a new state

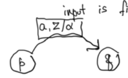
(iv) Advance the input head

Nondeterministic with ε-transitions.

Ex $L = \{ww^R \mid w \in (0+1)^*\}$

PDA :
- start in q_0
- push input symbols onto the stack
- at some point guess that all of w has been read and move to state q_1
- Keep reading input symbols and check they are the same as TOS ; Keep popping the stack
- Accept when the stack is empty and the input is finished

Graphical Presentation



a : input symbol

Z : TOS

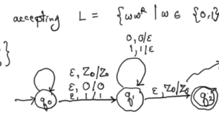
α : new string to replace Z on TOP

If in state p with input a and TOS Z then can make a transition to state q, replacing Z by string α and advancing the input head

PDA for accepting $L = \{ww^R \mid w \in \{0,1\}^*\}$

$\boxed{a, Z \mid \alpha}$

$0, Z_0/0Z_0$
$1, Z_0/1Z_0$
$0,0/00$
$0,1/01$
$1,0/10$
$1,1/11$



Assume : Initially the stack contains Z_0.

Def : A PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ where

Q : finite set of states

Σ : " input alphabet

Γ : " stack alphabet

q_0 : start state

Z_0 : initial stack symbol, $Z_0 \in \Gamma \setminus \Sigma$

$F \subseteq Q$: accepting / final states

$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow$ Set of finite subsets of $Q \times \Gamma^*$

Transition function

Recall ε-NFA : $\delta : Q \times \Sigma \cup \{\varepsilon\} \longrightarrow \boxed{2^Q}$

$\mathcal{P}(X)$: powerset of X

$\mathcal{P}_f(X)$: Set of finite subsets of X

Meaning of Transition function

$$\delta(q, a, Z) = \{\langle p_1, \gamma_1 \rangle, \langle p_2, \gamma_2 \rangle, \dots, \langle p_k, \gamma_k \rangle\}$$

where $a \neq \varepsilon$

means

if in state q, input symbol is a, and TOS is Z, then nondeterministically choose some i, $1 \leq i \leq k$ and
- go to state p_i
- Pop Z
- Push γ_i
- move input head one step

Similarly

$$\delta(q, \varepsilon, Z) = \{\dots\}$$

... if in state q regardless of the input symbol ... do not advance the head