CS 3311
October 22, 2012
Monday (1)

last time: eliminating lambda-rules

two steps
1. identify NULLable variables
2. remove $A \to \lambda$ rules where $A \neq S$.

big picture: we are trying to transform a given CFG into Chomsky Normal Form (CNF).

$A \to BC$ $A, B, C \in V$
$A \to a$ $a \in \Sigma$
$S \to \lambda$

Chomsky

$A \to w_1 A_1 w_2 A_2 w_3 A_3 w_4$

where $w_i \in (\Sigma \cup V)^*$
and $A_1, A_2, A_3 \in$ NULL

add all productions where a subset of $\{A_1, A_2, A_3\}$ turns to λ.

$A \to w_1 A_1 w_2 A_2 w_3 A_3 w_4 \mid w_1 w_2 A_2 w_3 A_3 w_4 \mid$
$w_1 A_1 w_2 w_3 A_3 w_4 \mid w_1 A_1 w_2 A_2 w_3 w_4 \mid$
$w_1 w_2 w_3 A_3 w_4 \mid w_1 A_1 w_2 w_3 w_4 \mid$
$w_1 w_2 A_2 w_4 w_4 \mid w_1 w_2 w_3 w_4$

simply cross out all the λ productions except at S.

$A \rightarrow aAbBcC$ where $A, B, C \in NULL$

$A \rightarrow aAbAcA$ where $A \in NULL$

$A \rightarrow \lambda$ or

$A \rightarrow BC$

How to remove A chain rule: CHAIN rules?

$A \rightarrow B$ |

a single variable on the right hand side

example

$A \rightarrow \underbrace{aA \mid a}_{\text{not chain rules}} \mid \underbrace{B}_{\text{a chain rule}}$

$B \rightarrow bB \mid c$

$A \rightarrow aA \mid a \mid$

$A \rightarrow aA \mid aB$

$\Rightarrow$

$w_1 A w_2 \Rightarrow w_1 B w_2$

$A \rightarrow aA \mid a \mid bB \mid c$

$B \rightarrow bB \mid c$

} transformed grammar

example:

$\{S, A, B, C\}$ $S \rightarrow A \mid B \mid def$

$\{A, C\}$ $A \rightarrow aA \mid a \mid C$

$\{B\}$ $B \rightarrow bB \mid b$

$\{C\}$ $C \rightarrow cC \mid c$

Remove the CHAIN rules from the above grammar:

Step 1: identify chains of variables

every variable that S is chained to.

$CHAIN(S) = \{S\} \cup \{A, B\} \cup \{C\} = \{S, A, B, C\}$

$CHAIN(A) = \{A\} \cup \{C\} \cup \ldots = \{A, C\}$

$CHAIN(B) = \{B\} \cup \ldots = \{B\}$

$CHAIN(C) = \{C\} \cup \ldots = \{C\}$

(4)

$S \to def \mid aA \mid a \mid bB \mid b \mid cC \mid c$

$\{S, A, B, C\}$ ✓ ✓ ✓

def: S's original

aA | a: all of A's non chain rules

bB | b: all of B's non-chain rules

cC | c: all of C's non-chain rules

$\{A, C\}$ $A \to aA \mid a \mid cC \mid c$ ✓ ✓

$\{B\}$ $B \to bB \mid b$

$\{C\}$ $C \to cc \mid c$

final grammar with no chain rules.

Get rid of useless variables:

- those that do not generate terminals
- those that are not reachable from S.

$G_1 = S \to AB$
$A \to aA$
$B \to bB \mid b$

— A is useless

$S \to \cancel{AB}$
$\cancel{A \to aA}$
$B \to bB \mid b$

$L(G_1) = ?$ $= \{\}$

an infinite number of a's
only b
no strings, $\emptyset$

$L(G_1) = \{\}$

$\Sigma = \{a, b\}$

$G_2 =$ $S \to AB$

$A \to aA \mid Bb$

$B \to bB \mid b$

$L(G_2) \stackrel{?}{=} \{\}$ no

A is no longer useless.

step 1: identify the TERM set

$S \to ABC \mid AC$ $\quad S \to AC$

$A \to aA \mid a$ $\quad A \to aA \mid a$

$B \to bB$

$C \to cC \mid def$ $\quad C \to cC \mid def$

$TERM = \{A, C\} \cup \{S\} = \{A, C, S\}$

$V - TERM = \{B\}$

↓ is the set variables that can generate strings of terminals

step2: remove every rule that has variables in $V - \{TERM\}$

$G = (V, \Sigma, P, S)$

$G' = (TERM, \Sigma, P', S)$

↓ retain the variables in TERM

↓ all the original production rules except the ones that have the variables in $V - \{TERM\}$

transformed grammar every variable can generate strings of terminals

How to remove the variables that are not reachable?

$S \to ABC \mid def$

$A \to aA \mid a \mid DE$

$B \to bB \mid b$

$C \to cC \mid c$

$D \to ghi$

~~$E \to eE \mid e$~~

$REACH = \{S\} \cup \{A, B, C\} \cup \{D\} = \{S, A, B, C, D\}$

on the right-hand side of S "reachable from S"

$G' = (REACH, \Sigma, P', S)$

P' → rules containing variables in V - REACH crossed out