Lecture notes for April 8, 2003
Scribe: Andy Dufilie

Chomsky Normal Form Revisited

Suppose we have a context-free grammar with the following rules:

$S \to AB$
$A \to aAb \mid B \mid e$
$B \to BA \mid e$

We want to convert G into G′ in Chomsky Normal Form such that L(G′) = L(G) – {*e*}.

Step 1: Eliminate e-rules

Find $E = \{A \mid A \stackrel{*}{\Rightarrow} e\}$

Here, E = {A, B, S}

Add the following rules:

$S \to A \mid B \mid e$
$A \to ab$
$B \to A$

Eliminate all e-rules:

$S \to e$
$A \to e$
$B \to e$

We get:

$S \to AB \mid A \mid B$
$A \to aAb \mid B \mid ab$
$B \to BA \mid A$

Step 2: Remove "singleton" rules ($A \to B$)

Add the following rules:

$S \to aAb \mid ab \mid BA$
$A \to BA$
$B \to aAb \mid ab$

Remove singleton rules:

$S \to A$
$S \to B$
$A \to B$
$B \to A$

We get:

$S \to AB \mid aAb \mid ab \mid BA$
$A \to aAb \mid ab \mid BA$
$B \to BA \mid aAb \mid ab$

Step 3: Convert all rules obtained at end of step 2 into Chomsky Normal Form

$S \rightarrow AB \mid X_aX_{Ab} \mid X_aX_b \mid BA$
$X_a \rightarrow a$
$X_b \rightarrow b$
$X_{Ab} \rightarrow AX_b$
$A \rightarrow X_aX_{Ab} \mid X_aX_b \mid BA$
$B \rightarrow BA \mid X_aX_{Ab} \mid X_aX_b$

Applications of Chomsky Normal Form

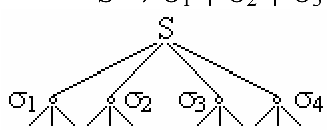
With Chomsky Normal Form, a grammar has binary parse trees in which every internal node on every path except the last one has exactly two children.
Suppose $\alpha \Rightarrow \beta$. Then $|\beta| \geq |\alpha|$
We can use this property to determine the following:

1. Membership problem:

Given a context-free grammar G and a string x, is $x \in L(G)$?
We can first convert G into Chomsky Normal Form G′. Then we want to know whether $x \in L(G')$ if $x \neq e$.
Algorithm: Search tree

Each internal node is a sentential form.

$S \rightarrow \sigma_1 \mid \sigma_2 \mid \sigma_3 \mid \sigma_4$



As soon as x is generated, stop and output "yes."
If a sentential form in the search tree is longer than $|x|$, then we cut off the entire subtree with this form as the root. Since there're only finitely many sentential forms of length $\leq |x|$ and there're no singleton rules, this process will stop. If x is generated, output "yes." Otherwise, output "no."

2. The emptiness problem:

Given a context-free grammar G, is $L(G) = \emptyset$?
Algorithm:

First convert G into Chomsky Normal Form G′.
If $L(G') \neq \emptyset$ then there must be a positive integer n (obtained from the pumping lemma) such that L(G′) contains a string of length $< n$.
We can use a for loop to loop through all possible strings x of length $< n$ and check whether $x \in L(G)$ or not.

3. The finiteness problem:

Given a context-free grammar G, is L(G) finite?
Algorithm:

If L(G) is infinite, then there must be a positive integer n such that L(G) contains a string w with $n \leq |x| < 2n$.