Chomsky Normal Form

Prakash Panangaden

18th October 2011

Any context-free language (CFL) has a context-free grammar (CFG) with all the rules in the following form

$$A \longrightarrow BC \text{ or } A \longrightarrow a$$

where upper case letters are variables (non-terminals) and lower case letters are terminals. We allow, as a special case, the rule $S \longrightarrow \varepsilon$ *for the start symbol only*; otherwise we could never have $\varepsilon \in L$. A CFG in this form is said to be in **Chomsky normal form** (CNF).

How do we prove such a thing? We will assume we have a CFG G for our CFL L and we will systematically transform G so that all the rules have the required forms; taking care that each transformation preserves the language. I will not give formal proofs that the transformations preserve the language, it is usually clear, in any case.

We order the variables and process them one by one. We call rules of the form $A \longrightarrow \varepsilon$ null rules. We start by getting rid of null rules. Let us suppose that we have a rule $A \longrightarrow \varepsilon$ that we want to eliminate. We throw it out but we introduce some new rules. If we see any rule with A on the right hand side we proceed as follows. Suppose, for example, there is a rule $X \longrightarrow aABAc$, we keep this rule and add new rules

$$X \longrightarrow aBAc \mid aABc \mid aBc.$$

In general, suppose we have a rule

$$X \longrightarrow \alpha_1 A \alpha_2 A \alpha_3 A \ldots \alpha_k A \beta$$

where the αs and the β are strings of symbols (including terminals and variables) without any occurrence of A. We introduce new rules

$$X \longrightarrow \alpha_1 \alpha_2 A \alpha_3 A \ldots \alpha_k A \beta$$

$$X \longrightarrow \alpha_1 A \alpha_2 \alpha_3 A \ldots \alpha_k A \beta$$

....

$$X \longrightarrow \alpha_1 A \alpha_2 A \alpha_3 A \ldots \alpha_k \beta$$

$$X \longrightarrow \alpha_1 \alpha_2 \alpha_3 A \alpha_4 \ldots \alpha_k A \beta$$

$$X \longrightarrow \alpha_1 \alpha_2 A \alpha_3 \alpha_4 \ldots \alpha_k A \beta$$

....

and so on. We first have to have rules with just one of the As removed, then rules with 2 of the As removed and so on until we have accounted for all possible combinations of As being removed. As you can see writing the most general form is neither pleasant nor particularly illuminating; furthermore it is pretty clear how to do it if someone holds a gun to your head.

We get rid of rules of the form $A \longrightarrow B$ (these are called unit rules) by adding rules of the form $A \longrightarrow \alpha$ whenever we have $B \longrightarrow \alpha$ unless $A \longrightarrow \alpha$ is a unit rule already removed.

We remove rules of the form $X \longrightarrow ab$ by introducing new variables $\langle a \rangle$ and $\langle b \rangle$ and then adding rules of the form

$$X \longrightarrow \langle a \rangle \langle b \rangle \quad \langle a \rangle \longrightarrow a \quad \langle b \rangle \longrightarrow b.$$

Similarly we get rid of rules of the form $A \longrightarrow aB$ by introducing the new variable $\langle a \rangle$ and adding the rules

$$X \longrightarrow \langle a \rangle B \quad \langle a \rangle \longrightarrow a.$$

Finally we are left with rules of the form

$$X \longrightarrow u_1 u_2 u_3 \ldots u_k$$

where $k \geq 3$ and the u_i are either terminal symbols or variables. We introduce new variables called $X_1, X_2, \ldots X_k$ and introducing new rules

$$X \longrightarrow u_1 X_1, X_1 \longrightarrow u_2 X_2, X_3 \longrightarrow u_3 X_3, \ldots$$

Some of these rules are in the right form – the ones where the u_i are variables – but some are not; we get new rules of the form $A \longrightarrow aB$, but we know how to get rid of them. I claim, without proof, that this process will eventually terminate yielding a CFG in CNF.