Normal Forms for Context-Free Grammars

Robert M. Keller
Harvey Mudd College
28 September 2003

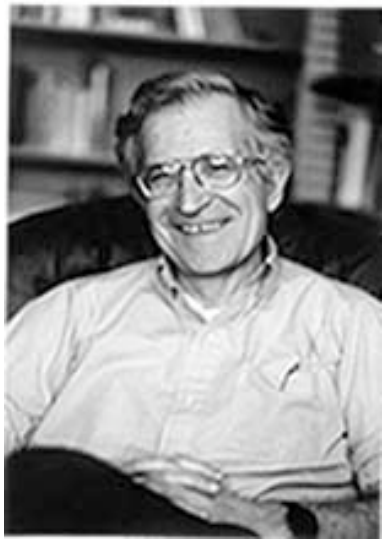
Uses of Normal Forms

- Certain proofs and constructions are simpler if the grammar can be assumed to be in a specific form.

- An example is the proof of the pumping lemma, which used the Chomsky Normal Form.

Chomsky Normal Form

- Each production has one of the following forms:
  - A → BC, where A, B, and C are auxiliaries (not necessarily distinct)
  - A → α, where α is a terminal symbol
- Noam Chomsky is a professor
- of linguistics at M.I.T.



Greibach Normal Form

- Each production has the following form:
  - $A \rightarrow a\beta$, where a is a terminal symbol and β is a string of auxiliary symbols (possibly empty).

- Sheila Greibach is a professor of computer science at UCLA.

- Sheila A. Greibach, A New Normal-Form Theorem for Context-Free Phrase Structure Grammars, Journal of the ACM (JACM), v.12 n.1, p.42-52, Jan. 1965

Example: Normal Forms for the balanced parenthesis language

- Chomsky:

  $S \rightarrow LT$
  $T \rightarrow SR$
  $S \rightarrow LR$
  $S \rightarrow SS$

  $L \rightarrow ($
  $R \rightarrow)$

- Greibach:

  $S \rightarrow (T$
  $S \rightarrow (ST$
  $S \rightarrow (TS$
  $S \rightarrow (STS$
  $T \rightarrow)$

The □ Problem

- Neither Chomsky nor Greibach forms can generate □, so we only apply them to languages not containing □.
- This is not a major issue, since if a language has □ as an element, we can always generate it by the single production

  S □ □

  where S is the start symbol and S does not appear on the RHS of any production.
- In the following discussion, we'll assume languages don't contain □ unless noted.

Chomsky Normal Form Lemma

- Every context-free language (not containing λ) is generated by a grammar in Chomsky Normal Form.

Grammar Cleanup Lemma

- For every context-free grammar (generating a language not containing λ),
  there is an equivalent grammar **not** containing either of the following two types of productions:
  - $A \rightarrow \lambda$ (called λ **productions**)
  - $A \rightarrow B$ (called **unit productions**)

Proof of the Grammar Cleanup Lemma (following Dexter Kozen's book)

- Starting with an arbitrary grammar (without $S \to \varepsilon$), iteratively add additional productions using the following two rules:
  - If $B \to \gamma$, then for any production of the form $A \to \alpha B \beta$, add a production $A \to \alpha\gamma\beta$.
  - If $B \to C$, then for any production of the form $A \to \alpha B \beta$, add a production $A \to \alpha C \beta$.
- The productions added have RHS's no longer than those of the original productions. Hence this process must terminate, because the number of different such strings is finite.
- Adding these productions does not change the language generated.

Continuation of the Proof of the Grammar Cleanup Lemma

- Claim: For any derivation of a terminal string in the modified grammar, there is a derivation not using ε- or unit productions.

- Proof: Suppose there is a derivation of a terminal string using a ε-production $B \rightarrow \varepsilon$:

  $S \Rightarrow^* \alpha B \beta \Rightarrow \alpha\beta \Rightarrow^* x$.

  Then B is not S and must have been introduced by a production of the form $A \rightarrow \alpha B \beta$. But by our rules for augmenting the grammar, we get an equivalent, but shorter, derivation using the rule $A \rightarrow \alpha\beta$ instead.

Continuation (2) of the Proof of the Grammar Cleanup Lemma

- Suppose there is a derivation of a terminal string using a unit production B → C:

  S →* αBβ → αCβ →* x.

  Then B must either be S or have been introduced by a production of the form A → αBβ. But by our rules for augmenting the grammar, we get an equivalent derivation of the same length using the rule A → αCβ instead.
- Since neither λ- or unit productions are needed in derivations in the augmented grammar, all such productions can now be eliminated without changing the language generated.

Example of Grammar Cleanup

- Consider the balanced-parenthesis grammar
  S → (S)
  S → SS
  S → ε
- Augmenting the grammar introduces:
  S → ()
  S → S
- Removing the ε- and unit productions gives:
  S → (S)
  S → ()
  S → SS

This generates the same language without ε.

Proof of the Chomsky Normal Form Lemma

- Start with a cleaned-up grammar.
- For every symbol $\square$ in the terminal alphabet occuring on some RHS of a production, introduce an auxiliary $A_{\square}$ and a production $A_{\square} \rightarrow \square$. Replace $\square$ on the RHS with $A_{\square}$.
- For any production with a RHS longer than 2:
  $A \rightarrow B_1B_2B_3\ldots B_n$,
  add new auxiliaries
  $C_2, C_3, \ldots C_{n-1}$ and replace the production with productions
  $A \rightarrow B_1C_2$, $C_2 \rightarrow B_2C_3$, …., $C_{n-1} \rightarrow B_{n-1}B_n$
- The resulting grammar generates the same language and is in Chomsky Normal Form.

Example of the Chomsky Normal Form Lemma

- Start with the cleaned-up grammar:

  S → (S)
  S → ()
  S → SS

- Add new symbols L and R for (and):

  S → LSR L → (
  S → LR R →)
  S → SS

- Replace S → LSR:

  S → LT
  T → SR L → (
  S → LR R →)
  S → SS