THEOREM OF THE DAY

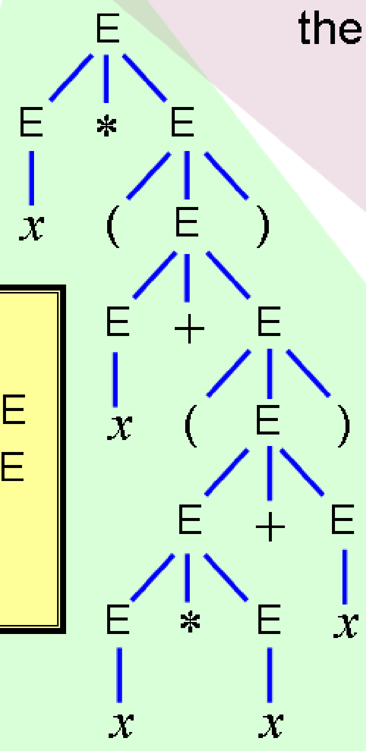
The Greibach Normal Form Theorem *If L is a context-free language not containing the empty word then L is generated by a context-free grammar in Greibach Normal Form, that is, one in which every production rule has the form $V \to t\alpha$, where t is a terminal symbol, and α is a, possibly empty, string of non-terminal symbols.*

Consider the grammars G_1 and G_2 shown on the right: they are *context-free*, meaning that their non-terminal symbols, respectively E, and D, E, F, may be replaced by the right-hand side of a rule regardless of context (surrounding symbols). G_1 generates the context-free language comprising all algebraic expressions in the variable x, with operators $+$ and $*$ and with bracketing. This is confirmed for any particular expression by constructing a *parse tree*, like the left-hand one here, in which the expression $x * (x + (x * x + x))$ is seen to be generated by applying each of the rules (1)-(3) twice and rule (4) five times. Grammar G_2 generates exactly the same language but is in Greibach Normal Form (GNF), since every $\longrightarrow$ is immediately followed by one of the terminal symbols, '+', '$*$', '(', ')' and 'x', which occur nowhere else. A useful side-effect is that a word in the context-free language which consists of n terminal symbols will necessarily be generated by exactly n rules: here we use, going from the root of the tree downwards, and from left to right, rules 8, 3, 5, 8, 9, 1, 5, 8, 9, 4, 7, 1, 7; a total of 13 the same, indeed, as the number of symbols in the generated expression. More significantly, GNF allows the specification of, for instance, functional programming languages, purely in terms of prefix operators; and it sheds light on the relationship between context-free grammars and pushdown automata.

The algebraic expression

$$x * (x + (x * x + x))$$

parsed according to the two grammars G_1 and G_2

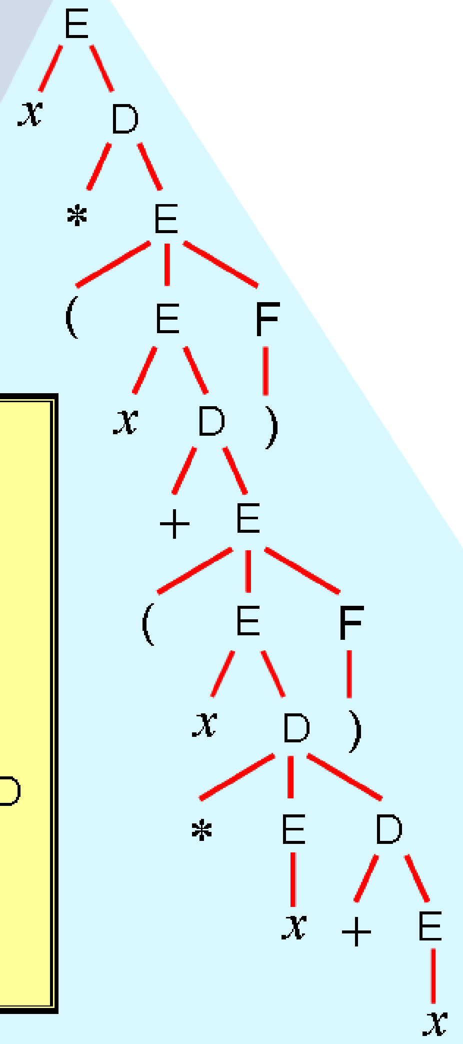


G_1:

1. $E \longrightarrow E + E$
2. $E \longrightarrow E * E$
3. $E \longrightarrow (E)$
4. $E \longrightarrow x$

G_2:

1. $D \longrightarrow + E$
2. $D \longrightarrow + E D$
3. $D \longrightarrow * E$
4. $D \longrightarrow * E D$
5. $E \longrightarrow (E F$
6. $E \longrightarrow (E F D$
7. $E \longrightarrow x$
8. $E \longrightarrow x D$
9. $F \longrightarrow)$



Sheila Greibach proved her normal form theorem in 1962. If the size $|G|$ of a context-free grammar G is the total number of symbols in all its rules, then a 1998 conversion algorithm of Norbert Blum and Robert Koch bounds the size of an equivalent GNF grammar G' to $O(|G|^4)$.

Web link: www.cs.stonybrook.edu/~cse350/slides/cfg3.pdf

Further reading: *Models of Computation and Formal Languages* by R Gregory Taylor, Oxford University Press Inc, USA, 1997, chapter 10.