

A VOCABULARY is a finite set of symbols.

A SENTENCE is a finite sequence of symbols taken from a vocabulary.

A LANGUAGE is a (possibly infinite) set of sentences.

A GRAMMAR G is a 4-tuple $G = \langle P, V, v, S \rangle$ where

P = a finite set of rules (called production rules)

V = a vocabulary of non-terminal symbols

v = a vocabulary of terminal symbols

S = a distinguished symbol – the start symbol

A production rule is of the form: $L \rightarrow R$, where L and R are finite sequences of symbols and the symbol " \rightarrow " means "can be rewritten as."

A production rule $L \rightarrow R$ is CONTEXT-FREE if L is a single non-terminal.

A grammar G is context-free if all its production rules are context-free.

A DERIVATION using grammar G is a finite sequence of sentences,

$D(G) = S_0, S_1, S_2, \dots, S_n$, such that:

1. S_0 is the start symbol of G
2. Each S_i is obtained from $S_{(i-1)}$ by application of a rule from G .

I.e., $S_{(i-1)}$ is a sentence of the form aLb ,

S_i is a sentence of the form aRb ,

and G contains the rule $L \rightarrow R$.

We say that $D(G)$ is a derivation of S_n .

$D(G)$ is a CANONICAL derivation if each application of a rule is performed on the leftmost non-terminal in $S_{(i-1)}$ to produce S_i .

Notation: To indicate that sentence S_j can be derived from sentence S_i ,

we write $S_i \xRightarrow[G]{*} S_j$

If S_n consists of only terminal symbols, then S_n is in the language generated by G .

The language generated by G is denoted by $L(G)$;

$L(G) = \{ w \mid S \xRightarrow[G]{*} w, w \text{ is terminals only} \}$

The Language of Balanced Parentheses

Examples: $(())$ $()(())$ $(())(())()$

A grammar that defines the language whose sentences are strings of balanced parentheses.

Terminals: $\{ (,) \}$

Non-terminals: $\{ S, X \}$ ($S =$ start symbol)

Production Rules:

$S \rightarrow X$ $X \rightarrow ()$ $X \rightarrow (X)$ $X \rightarrow XX$

Derive

$((() ()) (())) ()$

S
 X
 $X X$
 $X ()$
 $(X) ()$
 $(X X) ()$
 $(X (X)) ()$
 $(X (())) ()$
 $((X) (())) ()$
 $((X X) (())) ()$
 $((() X) (())) ()$
 $((() ()) (())) ()$

Integers without leading zeros

Examples: 712, +44, -8787, 900801

Grammar:

Terminals: $\{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Non-terminals: $\{S, X, D, Y, Z\}$

Production rules:

$S \rightarrow 0$	$Y \rightarrow Z$	$D \rightarrow 1$
$S \rightarrow X$	$Y \rightarrow ZY$	\vdots
$S \rightarrow +X$	$X \rightarrow D$	$D \rightarrow 9$
$S \rightarrow -X$	$X \rightarrow DY$	$Z \rightarrow 0$
		$Z \rightarrow D$

S Start Sym. – all integer constants

X Any integer without leading zeros

Y all strings of digits

D all non-zero digits

Z all digits

Derivation

S
+X
+DY
+4Y
+4Z
+4D
+44

Parse Trees

A PARSE TREE for a sentence S_n in $L(G)$, where G is a context-free grammar, is a tree in which:

1. Each node in the tree is a symbol from G
2. The root is the start symbol for G
3. If node L has children R_1, R_2, \dots, R_n , then G contains the rule $L \rightarrow R$ where $R = R_1 R_2 \dots R_n$
4. If node L is a leaf, then L is a terminal symbol.
5. If the leaves of the tree are L_1, L_2, \dots, L_n in left-to-right order, then $S_n = L_1 L_2 \dots L_n$

A context-free grammar G is UNAMBIGUOUS if for every sentence w in $L(G)$, there is a unique parse tree for w (hence a unique canonical derivation).

A given language will in general be generated by many grammars; some may be ambiguous and some may not. An INHERENTLY AMBIGUOUS language is one for which there does not exist an unambiguous grammar.

Terminals: $\{+, -, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Non-terminals: $\{S, X, D, Y, Z\}$

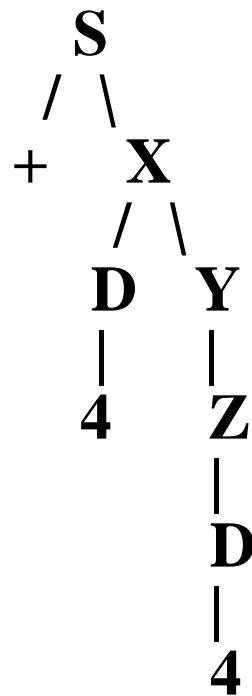
Grammar:

$S \rightarrow 0$	$Y \rightarrow Z$	$D \rightarrow 1$
$S \rightarrow X$	$Y \rightarrow ZY$	\vdots
$S \rightarrow +X$	$X \rightarrow D$	$D \rightarrow 9$
$S \rightarrow -X$	$X \rightarrow DY$	$Z \rightarrow 0$
		$Z \rightarrow D$

Canonical Derivation

S
+X
+DY
+4Y
+4Z
+4D
+44

Parse Tree



Backus-Naur Form (BNF)

This is just some convenient notation for describing a context-free grammar more succinctly.

- Non-terminal symbols are recognized by being enclosed in angle-brackets.
- Terminal symbols are recognized by NOT being enclosed in angle-brackets.
- The "|" symbol is used on the right hand side of rules to indicate alternative choices.
- There is no special declaration of the start symbol; it is usually obvious.

Example: $G = \langle P, V, v, S \rangle$ where

$$\begin{array}{l} P = \{X \rightarrow A, \\ \quad X \rightarrow B, \\ \quad X \rightarrow C\} \quad V = \{X\} \quad S = X \\ v = \{A, B, C\} \end{array}$$

In BNF, we can specify G by the one rule:

$$\langle X \rangle \rightarrow A \mid B \mid C$$

Integers without leading zeros using BNF

$$\langle S \rangle \rightarrow \langle X \rangle \mid +\langle X \rangle \mid -\langle X \rangle \mid 0$$
$$\langle X \rangle \rightarrow \langle D \rangle \mid \langle D \rangle \langle Y \rangle$$
$$\langle Y \rangle \rightarrow \langle Z \rangle \mid \langle Z \rangle \langle Y \rangle$$
$$\langle D \rangle \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
$$\langle Z \rangle \rightarrow 0 \mid \langle D \rangle$$

Convenient extensions to BNF

Optional item:	[]
Unbounded repetition:	...
Nonterminal symbol:	<i>Different font</i>

$$\mathbf{S} \rightarrow \mathbf{X} \mid +\mathbf{X} \mid -\mathbf{X} \mid 0$$
$$\mathbf{X} \rightarrow \mathbf{D} [\mathbf{Y}]$$
$$\mathbf{Y} \rightarrow \mathbf{Z} \dots$$
$$\mathbf{D} \rightarrow 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9$$
$$\mathbf{Z} \rightarrow 0 \mid \mathbf{D}$$