Metalanguage - BNF

My Classes

CSC 600 : Programming Languages

CSC 621 : Biomedical Imaging & Analysis

CSC 648 : Software Engineering

CSC 675 : Intro to Database Systems

CSC 699 : Learning by Teaching

My Classes

CSC 600 : Programming Languages

CSC 621 : Biomedical Imaging & Analysis

CSC 648 : Software Engineering

CSC 675 : Intro to Database Systems

CSC 699 : Learning by Teaching

Metalanguage - Concept

Object language

- Language that we are analyzing

Metalanguage

- Language used to describe an object language

Metalanguage should be different from the object language

Metalanguages

BNF

- Backus-Naur Form

EBNF

- Extended BNF

Syntax Diagrams

Backus-Naur Form

A grammar that defines any programming language

- How an interpreter/compiler knows what words and numbers mean in your program!

Used by interpreter to break the strings (lines of code) we wrote into their parts

You can use this if you ever want to create your own programming language!

Backus-Naur Form

Metasymbols

- < x > means an object called x
- | means exclusive or
  - <digit> | <letter> means either a digit or a letter, not both
- ::= is used as the assignment operator

Basic Components of BNF

<digit> ::= 0|1|2|3|4|5|6|7|8|9

Terminal Symbol

- Symbols that represent themselves (e.g. parentheses, +, -, if, else, etc.)

BNF

Object name

You can then use <digit> as a metasymbol to replace the tokens 0,…,9

Sequences

<letter> ::= a | b

<digit> ::= 1 | 2 | 3

<var> ::= <letter> <digit> <letter>

Sequences

<letter> ::= a | b

<digit> ::= 1 | 2 | 3

<var> ::= <letter> <digit> <letter>

Valid sequences:

a1a a2a a3b

b1b b1a b2a b3b

Recursion in BNF

<variable_name> ::= <letter> | <variable_name> <char>

<char> ::= <digit> | <letter>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

Recursion in BNF

<variable_name> ::= <letter> | <variable_name> <char>

<char> ::= <digit> | <letter>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

Wrong to use “left-recursion”, e.g.:

<variable_name> ::= <letter> | <char> <variable_name>

Recursion in BNF

<variable_name> ::= <letter> | <variable_name> <char>

<char> ::= <digit> | <letter>

<digit> ::= 0|1|2|3|4|5|6|7|8|9

<letter> ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|r|s|t|u|v|w|x|y|z

Wrong to use “left-recursion”, e.g.:

<variable_name> ::= <letter> | <char> <variable_name>

Java BNF

Entire Java language can be defined in BNF

https://goo.gl/SyoVc6

Look for :

<result type> ::= *<type>* | void

Syntax Diagram

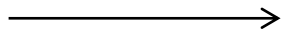
Graphical representation of BNF

Easy to understand

Flowchart concept

Syntax Diagram

Terminal symbols are circled 

Arrows show the path through the diagram 

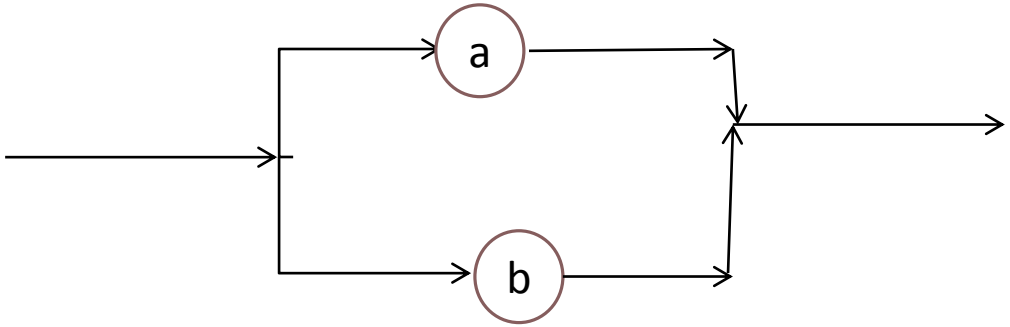
Rectangles mean metasymbols 

Diagram shows every possible path, from entry to exit, for the specific BNF definition

Syntax Diagram

first a and then b

Sequence ab



Exclusive or

a or b