BNF Assignment

Bryan McLean

Learning Abstract:

This assignment consists of many different BNF grammars and parse trees to navigate the language. This assignment also has a simple explanation of BNF in English.

Problem 1: Shapes

Language

<st> ::= (<p>)

<p> ::= <size> <color> <pattern> <shape>

<size> ::= (size big) | (size medium) | (size small)

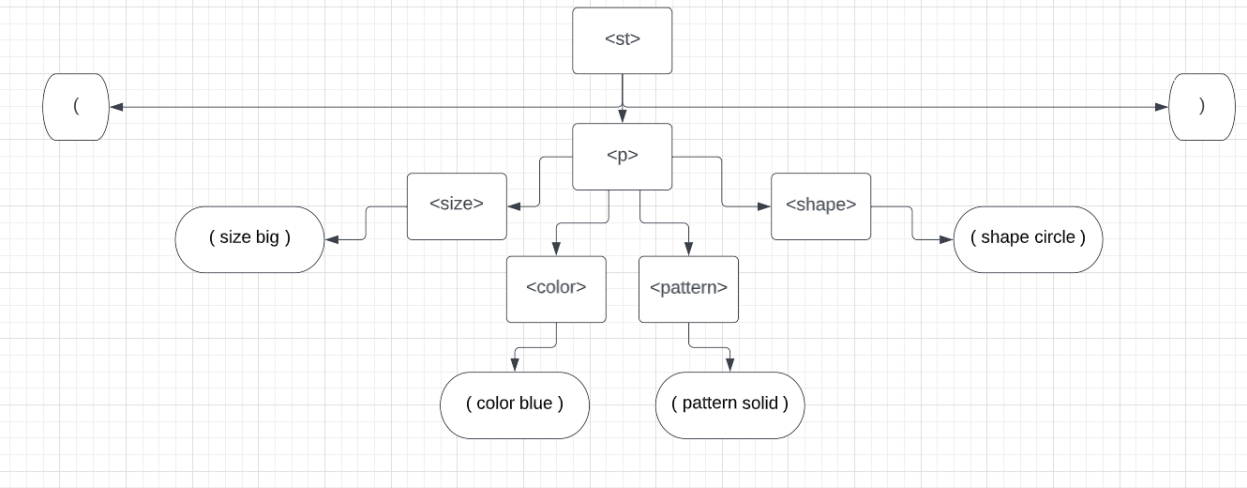
<color> ::= (color red) | (color blue) | (color yellow)

<pattern> ::= (pattern striped) | (pattern dotted) | (pattern solid)

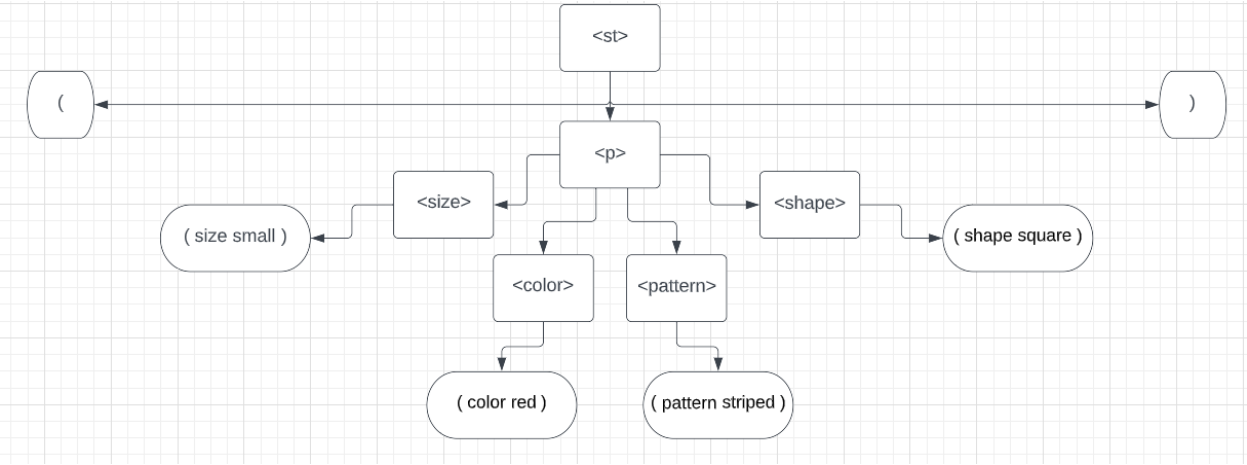
<shape> ::= (shape circle) | (shape square) | (shape triangle)

Parse Trees:

2)



3)



Problem 2: SQN (Special Quaternary Numbers)

Language:

<SQN> ::= 0 | <O> | <TW> |<TH>

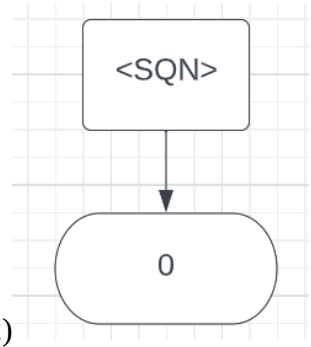
<O> ::= <empty> | 1 <empty> | 1 <TW> | 1 <TH>

<TW> ::= <empty> | 2 <empty> | 2 <O> | 2 <TH>

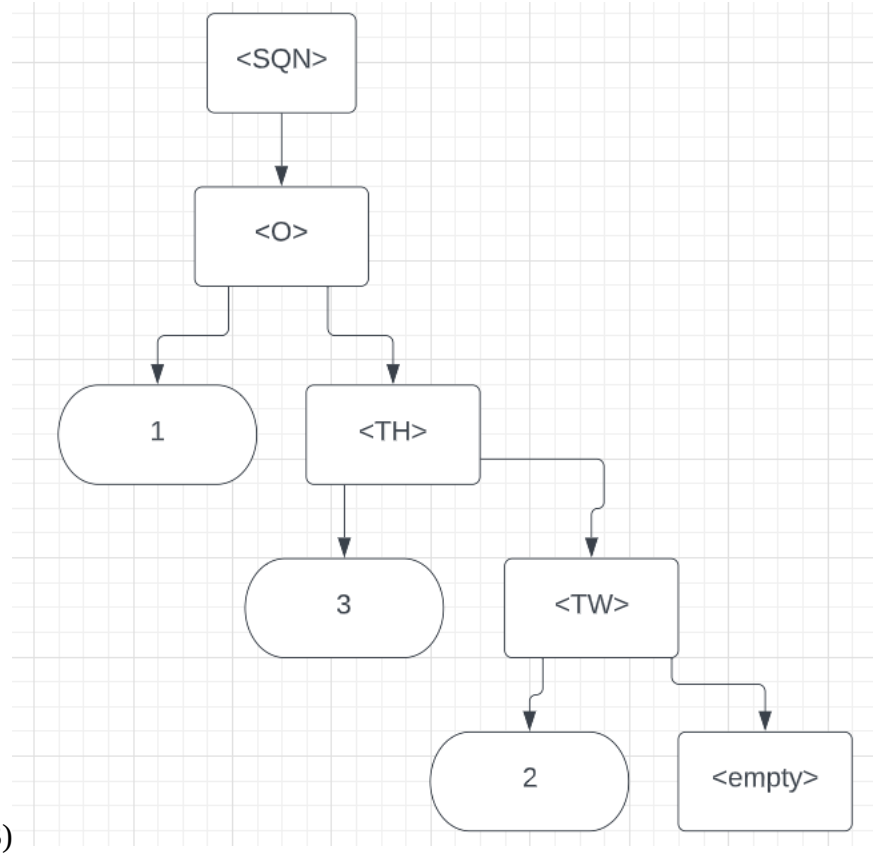
<TH> ::= <empty> | 2 <empty> | 2 <O> | 2 <TW>

Parse Trees:

2)



3)



4) You cannot draw a parse tree for the string 1223 because the language does not allow for back-to-back occurrences of the same number. This string would not work because of the twos that occur next to each other.

Problem 3: Fours

Language:

<Fours> ::= <1> <2> <3> <4>

<1> ::= <empty> | (1 1 1 1) <1>

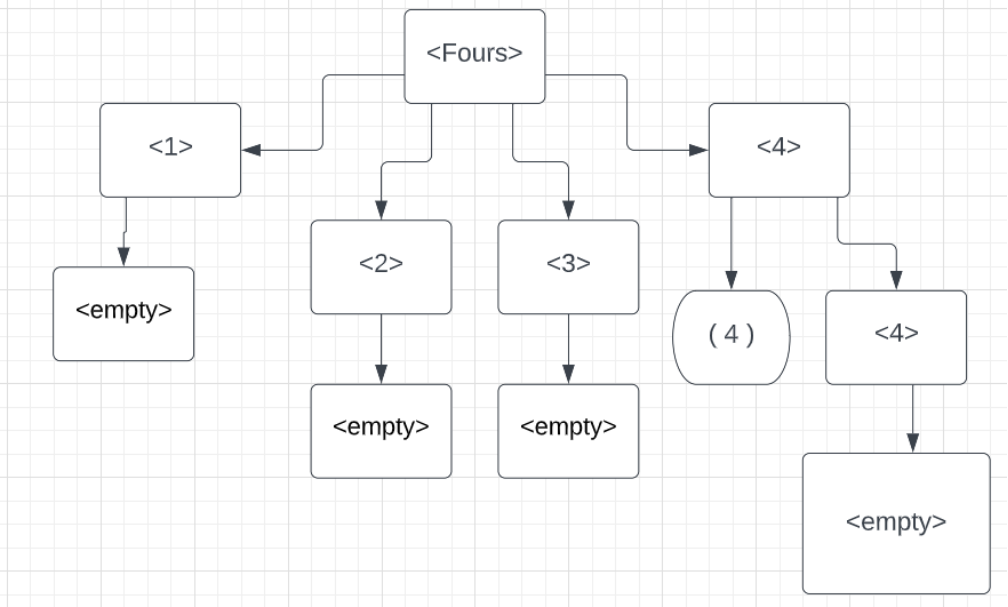
<2> ::= <empty> | (1 1 2) <2> | (1 2 1) <2> | (2 1 1) <2>

<3> ::= <empty> | (3 1) <3> | (1 3) <3>

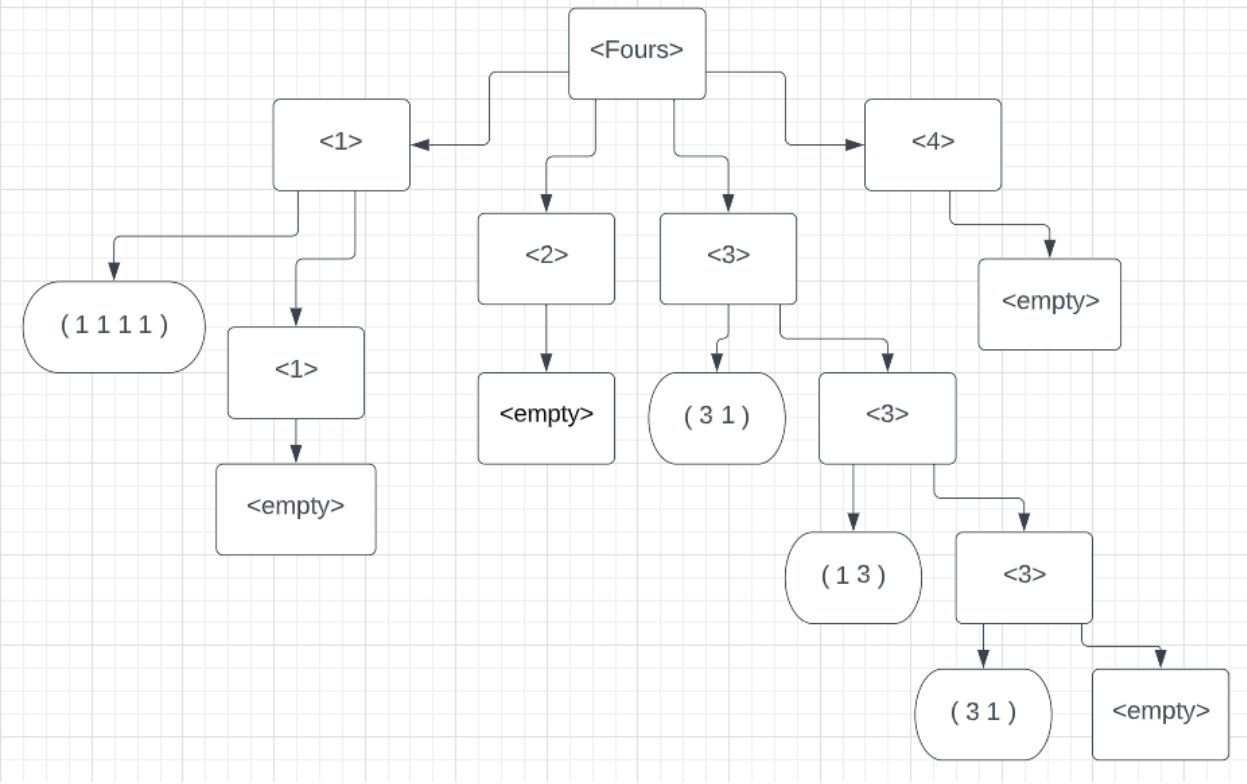
<4> ::= <empty> | (4) <4>

Parse Trees:

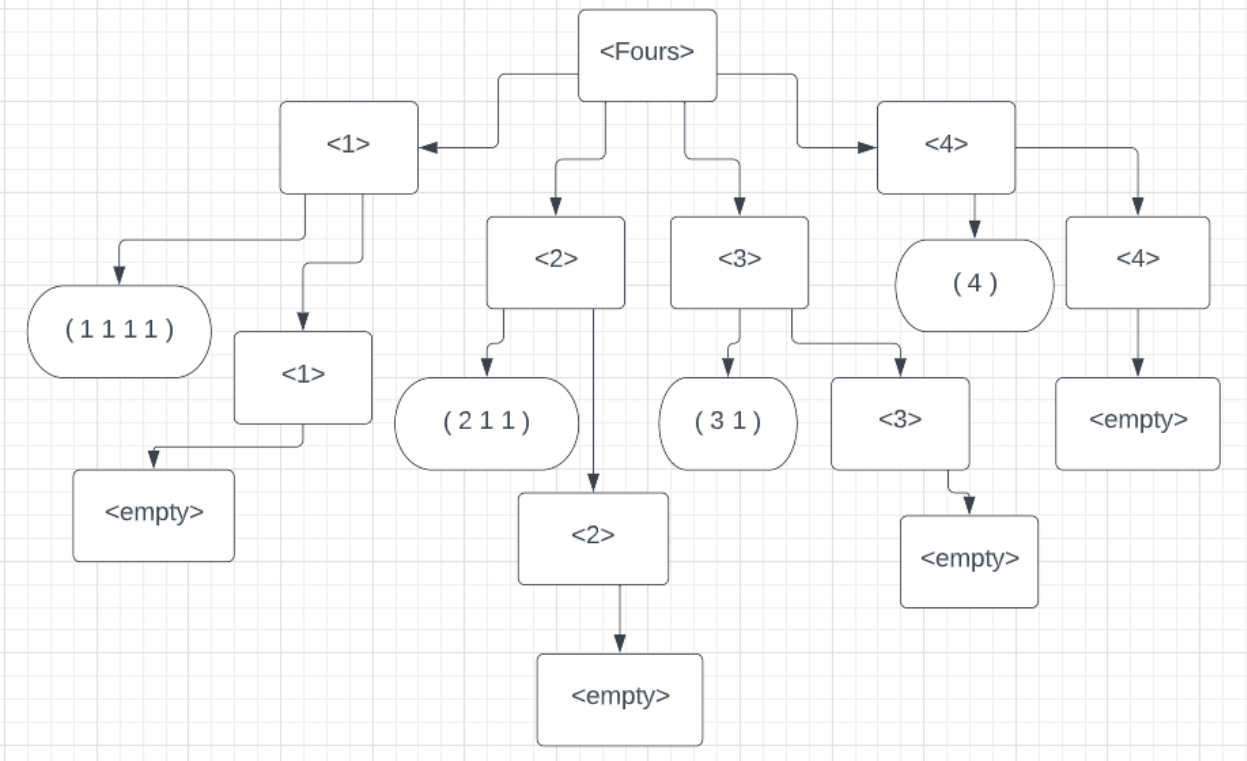
2)



3)



4)



Problem 4: BXR

Language:

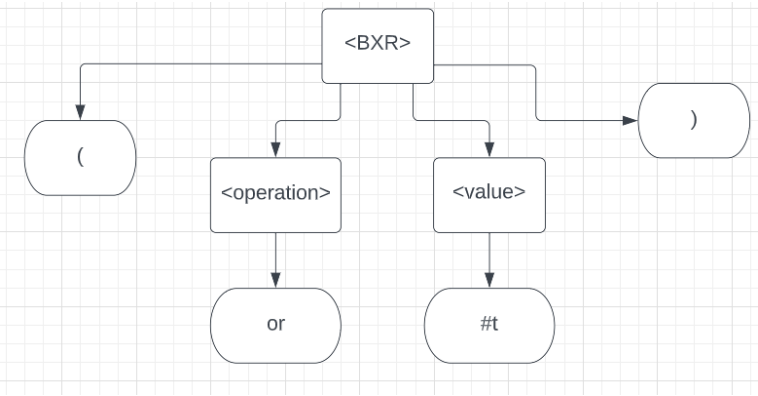
<BXR> ::= (<operation> (<operation> <value>) <value>) | (<operation> <value>)

<value> ::= <empty> | #t | #f | #t <BXR> | #f <BXR>

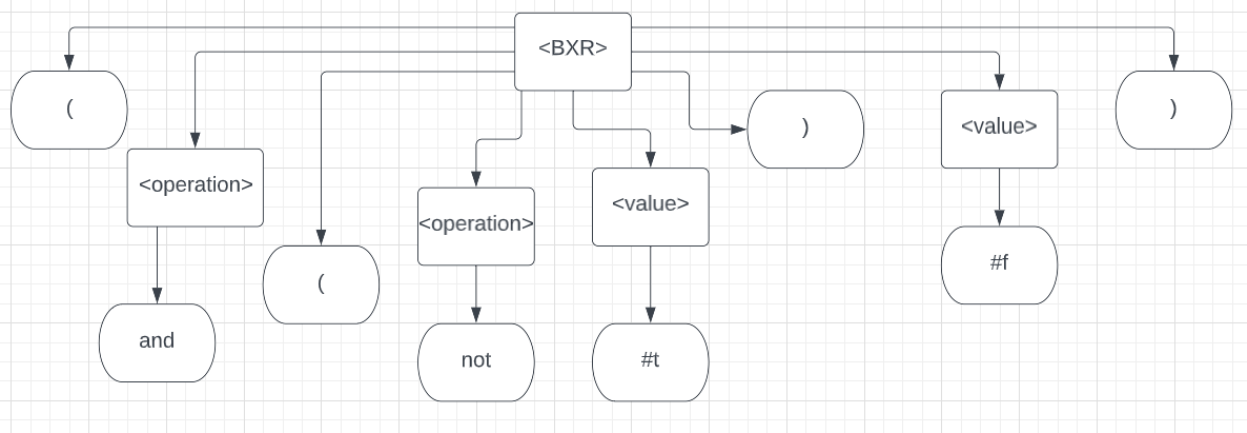
<operation> ::= <empty> | and | or | not | <BXR>

Parse Trees:

2)



3)



Problem 5: Color Fun

Language:

<cf> ::= <add> | <show> | <describe> | colors | <exit>

<add> ::= add (<number> <number> <number>) <color> | add (<number> <number> <number> <number>) <color>

<show> ::= show <color>

<describe> ::= describe <color>

<exit> ::= Goodbye …

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

1 <09> | 2 <09> | 3 <09> | 4 <09> | 5 <09> | 6 <09> | 7 <09> | 8 <09> | 9 <09> |

1 <09> <09> | 2 <04> <09> | 25 <05>

<09>::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<04> ::= 0 | 1 | 2 | 3 | 4

<05> ::= 0 | 1 | 2 | 3 | 4 | 5

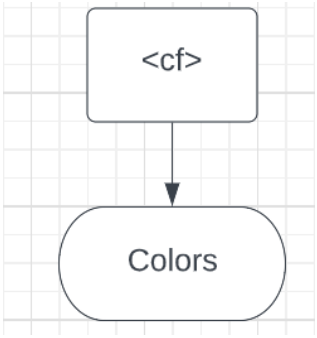
<color> ::= <shade> <name>

<shade> ::= <empty> | light- | dark-

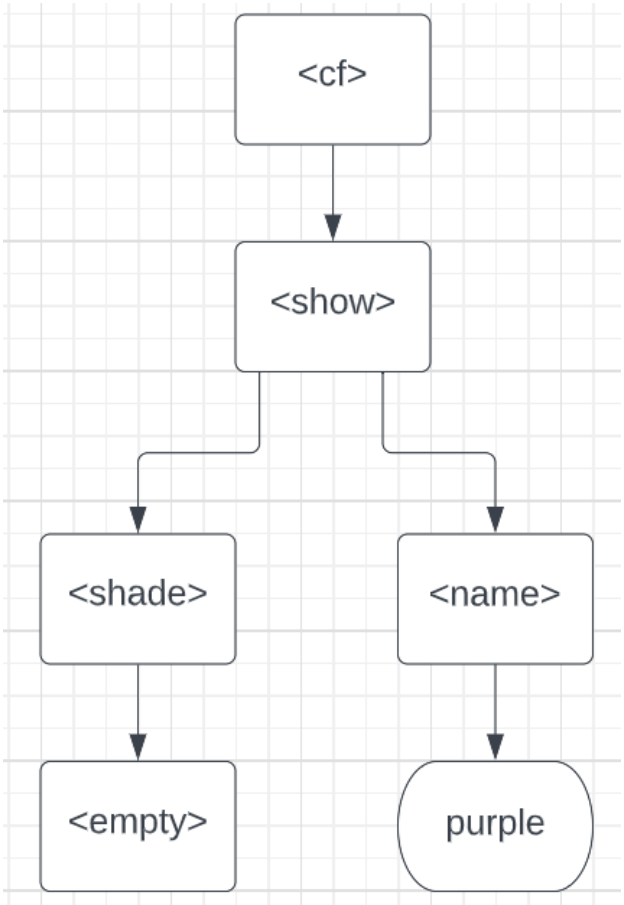
<name> ::= yellow | orange | red | purple | pink | blue | green | black | gray| white | favorite-color

Parse Trees:

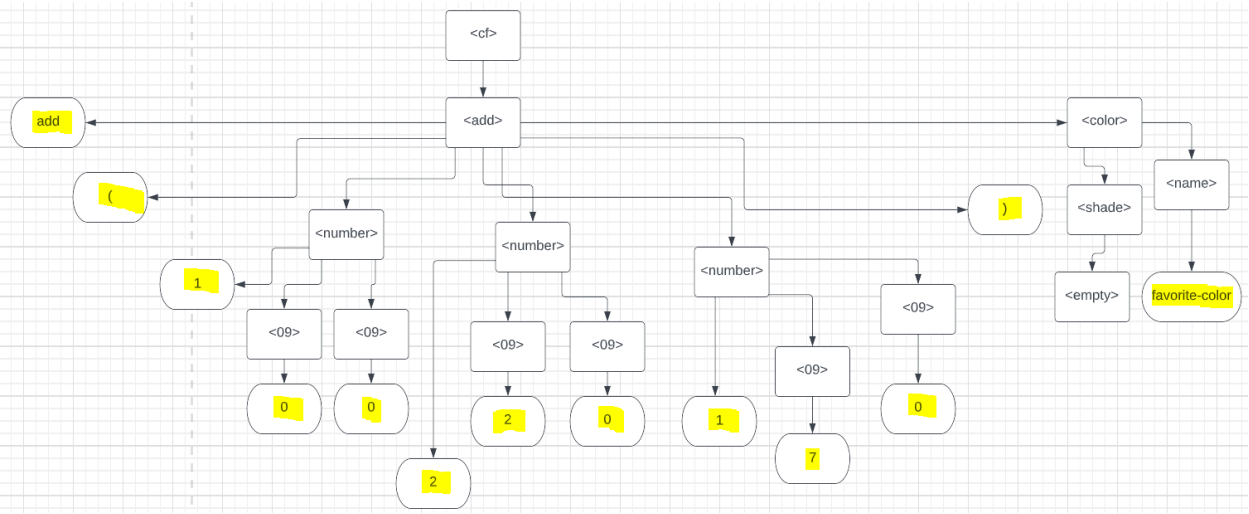
2)



3)



4)



Problem 6: BNF?

BNF or Backus-Naur form is used by many program language developers to specify the rules of a language. BNF uses nonterminal symbols that are not in the language to create production rules and are replaced by expressions that are in the language. BNF was invented by John Backus and Peter Naur in 1960.