Assignment: BNF Grammar

Abstract: This assignment asked me to construct BNF grammars and parse trees for 5 languages. The 5 languages include Shapes, SQN (special quaternary numbers), fours, BXR (Boolean-expressions), and CF (color-fun).

Interaction: Shapes

Grammar:

<S> ::= (<list>)

<list> ::= <el1><el2><el3><el4>

<el1> ::= (<part1.1><part1.2>)

<el2> ::= (<part2.1><part2.2>)

<el3> ::= (<part3.1><part3.2>)

<el4> ::= (<part4.1><part4.2>)

<part1.1> ::= size

< part1.2> ::= big | medium | small

< part2.1> ::= color

< part2.2> ::= red | blue | yellow

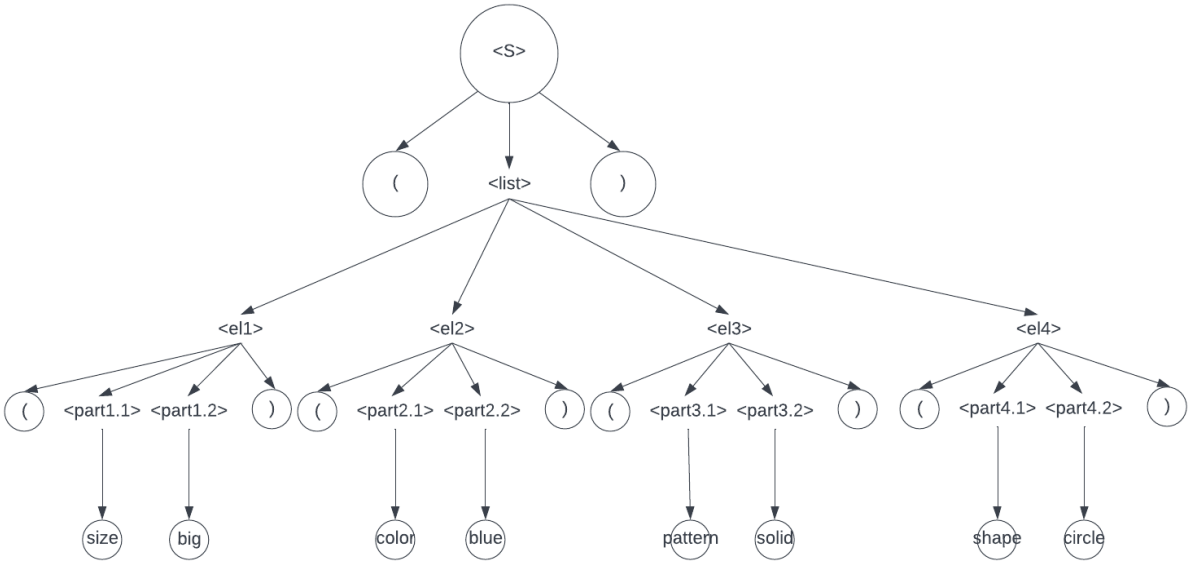
< part3.1> ::= pattern

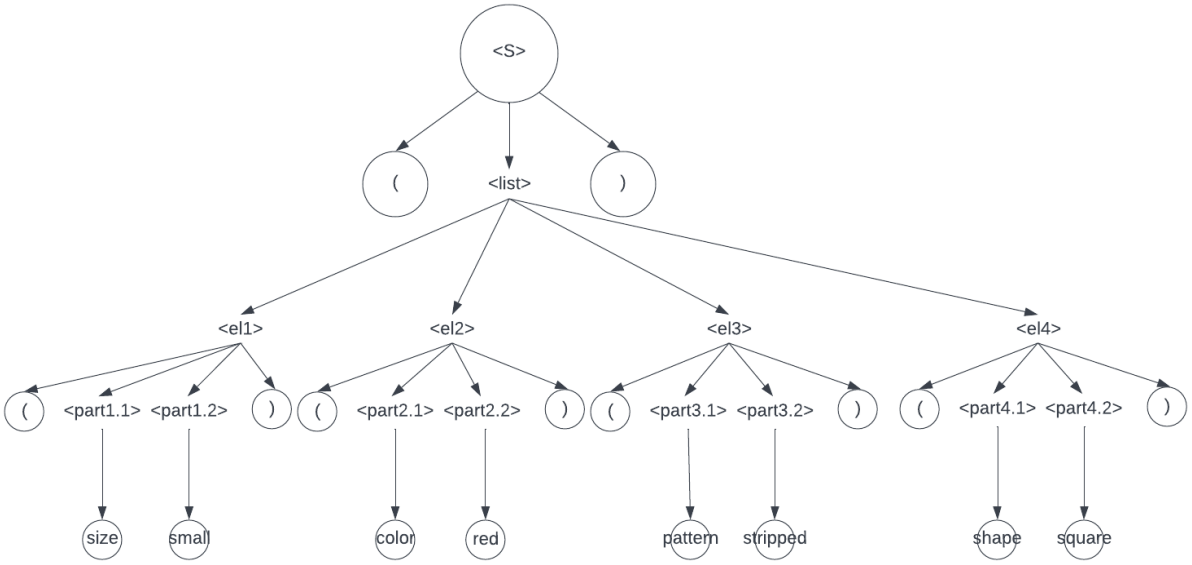
< part3.2> ::= striped | dotted | solid

< part4.1> ::= shape

< part4.2> ::= circle | square | triangle

Parse Trees:





Interaction: SQN

Grammar:

<S> ::= 0 | <start-sequence>

<start-sequence> ::= <1-digit> | <2-digit> | <3-digit>

<0-sequence> ::= <1-digit> | <2-digit> | <3-digit> | <empty>

<1-sequence> ::= <0-digit> | <2-digit> | <3-digit> | <empty>

<2-sequence> ::= <0-digit> | <1-digit> | <3-digit> | <empty>

<3-sequence> ::= <0-digit> | <1-digit> | <2-digit> | <empty>

<empty> ::= ε

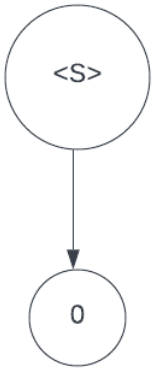
<0-digit> ::= 0 <0-sequence>

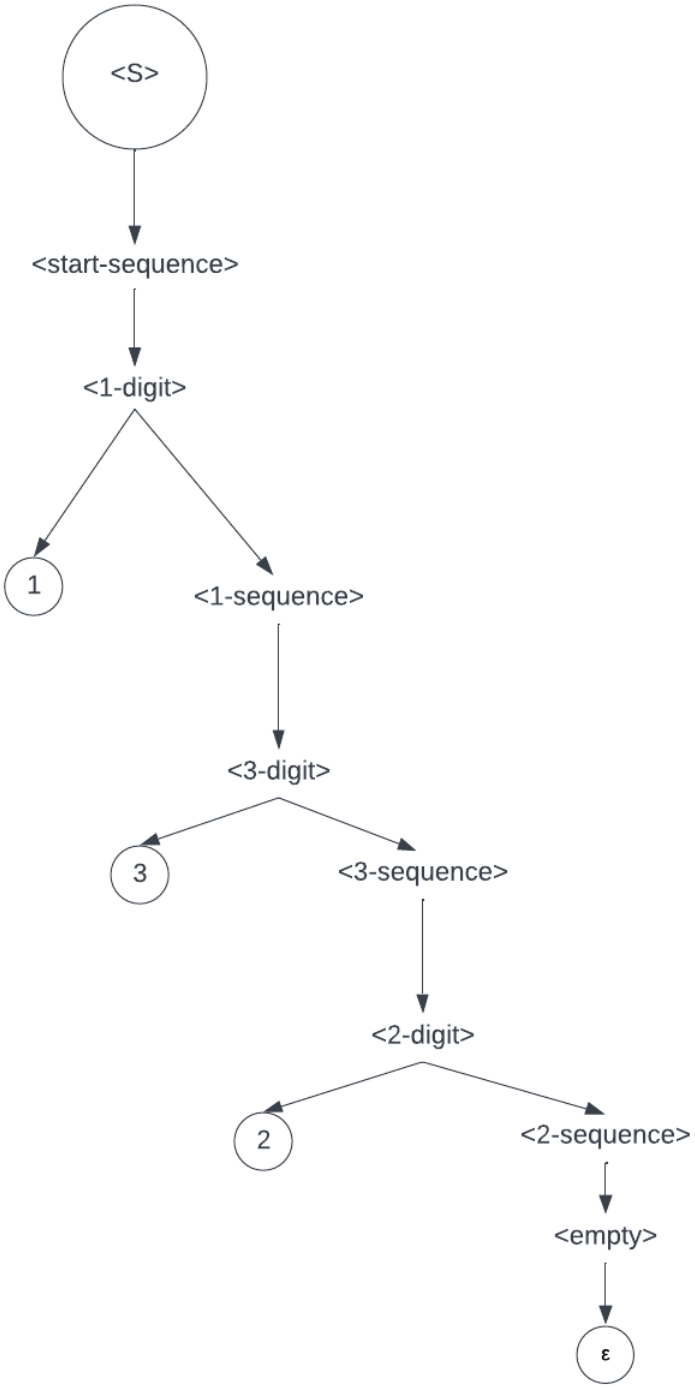
<1-digit> ::= 1 <1-sequence>

<2-digit> ::= 2 <2-sequence>

<3-digit> ::= 3 <3-sequence>

Parse Trees:





Explanation: Why 1223 is not available in the parse tree? Starting at the beginning of sequence we can only choose two options a 0 or a start sequence. The start sequence is the path we will follow which leads us into the next part which is to choose the “1-digit” non-terminal. This non-terminal gives us the 1 in the 1223 sequence and we look into the “1-sequence” non-terminal which has the option to pick “2-digit” to get us our 2 that we seek in the 1223 sequence. A second 2 is not allowed because the “2-

sequence” non-terminal has 4 options, “0-digit” to get a 0, “1-digit” to get a 1, “3-digit" to get a 3, and “empty” to stop the sequence.

Interaction: Fours

Grammar:

<S> ::= <start-1s> | <start-2s> | <start-3s> | <start-4s>

<start-1s> ::= <1s> <1-seq>

<start-2s> ::= <2s> <2-seq>

<start-3s> ::= <3s> <3-seq>

<start-4s> ::= <4s> <4-seq>

<1-seq> ::= <start-1s> | <start-2s> | <start-3s> | <start-4s> | <empty>

<2-seq> ::= <start-2s> | <start-3s> | <start-4s> | <empty>

<3-seq> ::= <start-3s> | <start-4s> | <empty>

<4-seq> ::= <start-4s> | <empty>

<empty> ::= ε

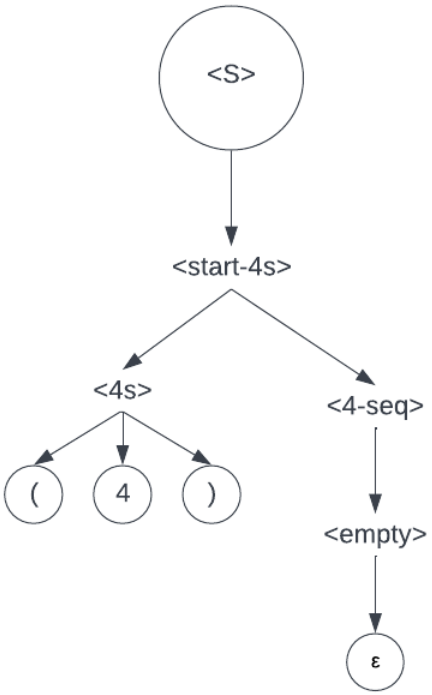
<1s> ::= (1 1 1 1)

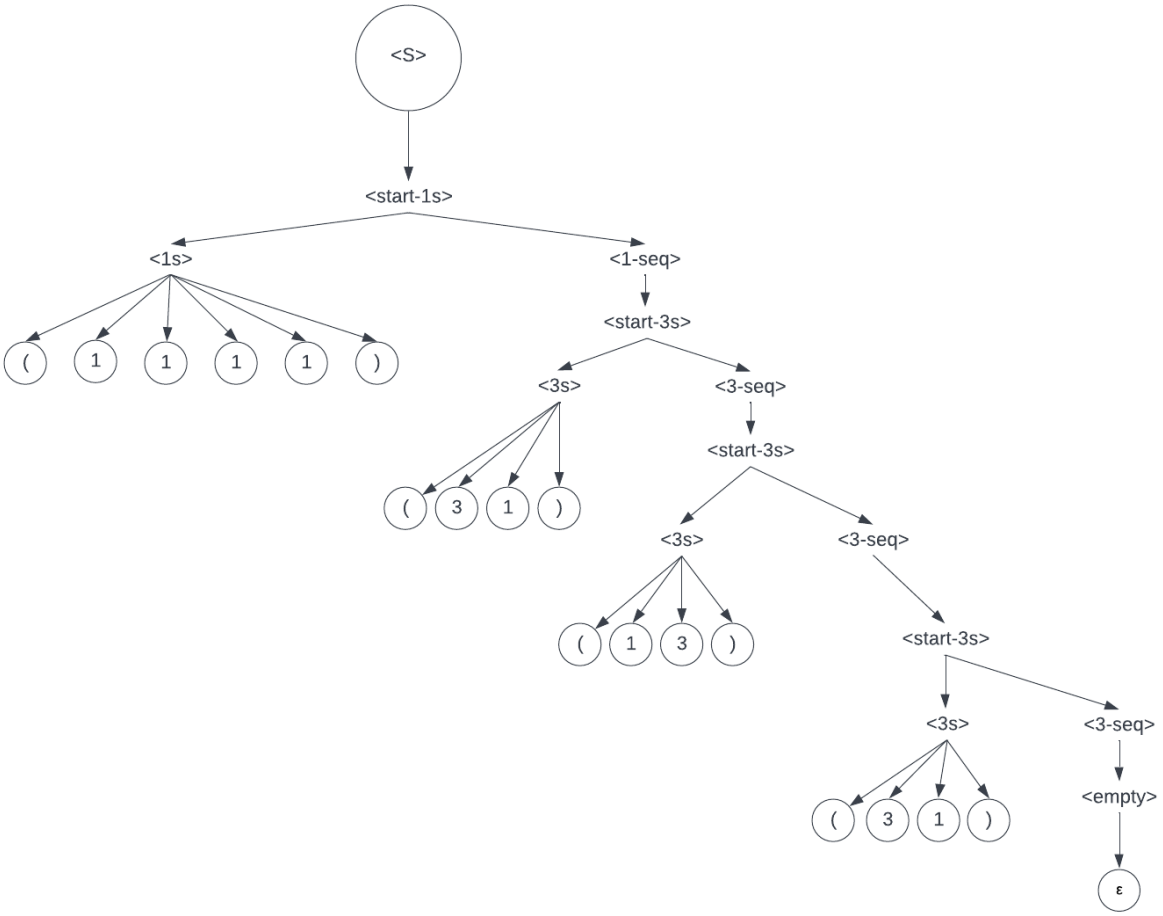
<2s> ::= (1 1 2) | (1 2 1) | (2 1 1)

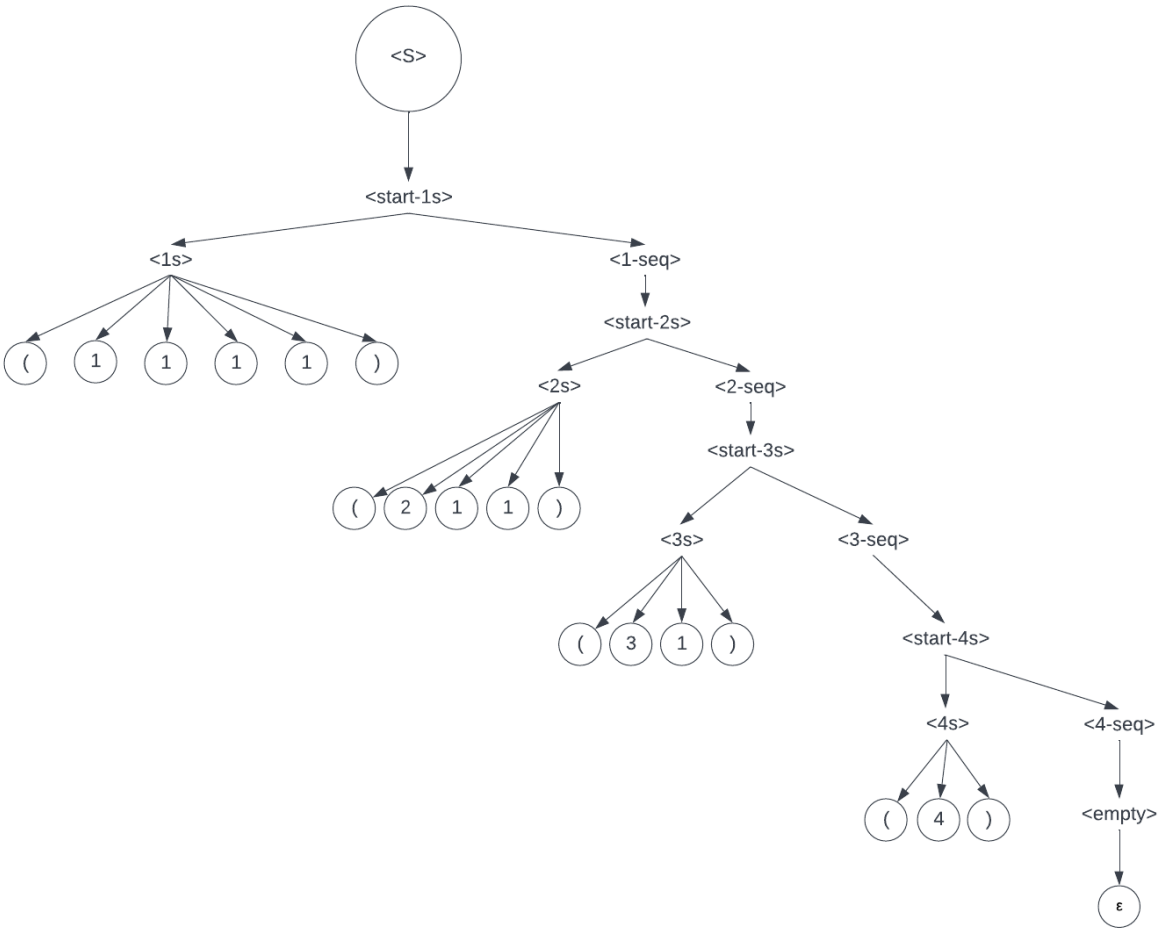
<3s> ::= (3 1) | (1 3)

<4s> ::= (4)

Parse Trees:


<S>
<start-4s>
<4s>
<4-seq>
(
4
)
<empty>
ε


<S>
<start-1s>
<1s>
<1-seq>
(
1
1
1
1
)
<start-3s>
<3s>
<3-seq>
(
3
1
)
<start-3s>
<3s>
<3-seq>
(
1
3
)
<start-3s>
<3s>
<3-seq>
(
3
1
)
<empty>
ε



Interaction: BXR

Grammar:

<S> ::= <and> | <or> | <not>

<and> :: = (and <and-op-list>)

<or> ::= (or <or-op-list>)

<not> ::= (not <not-op-list>)

<and-op-list> ::= <value> <and-op-list> | <empty>

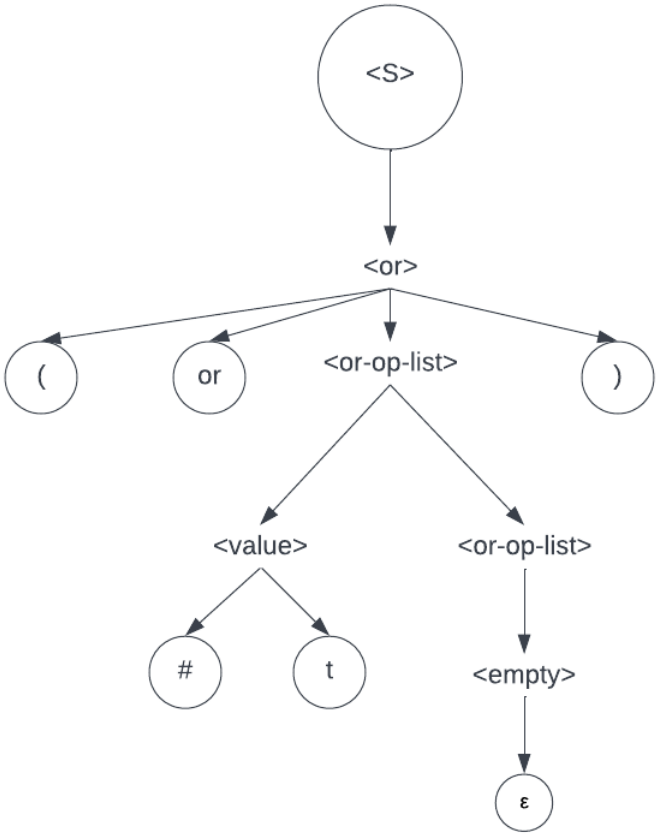
<or-op-list> ::= <value> <or-op-list> | <empty>

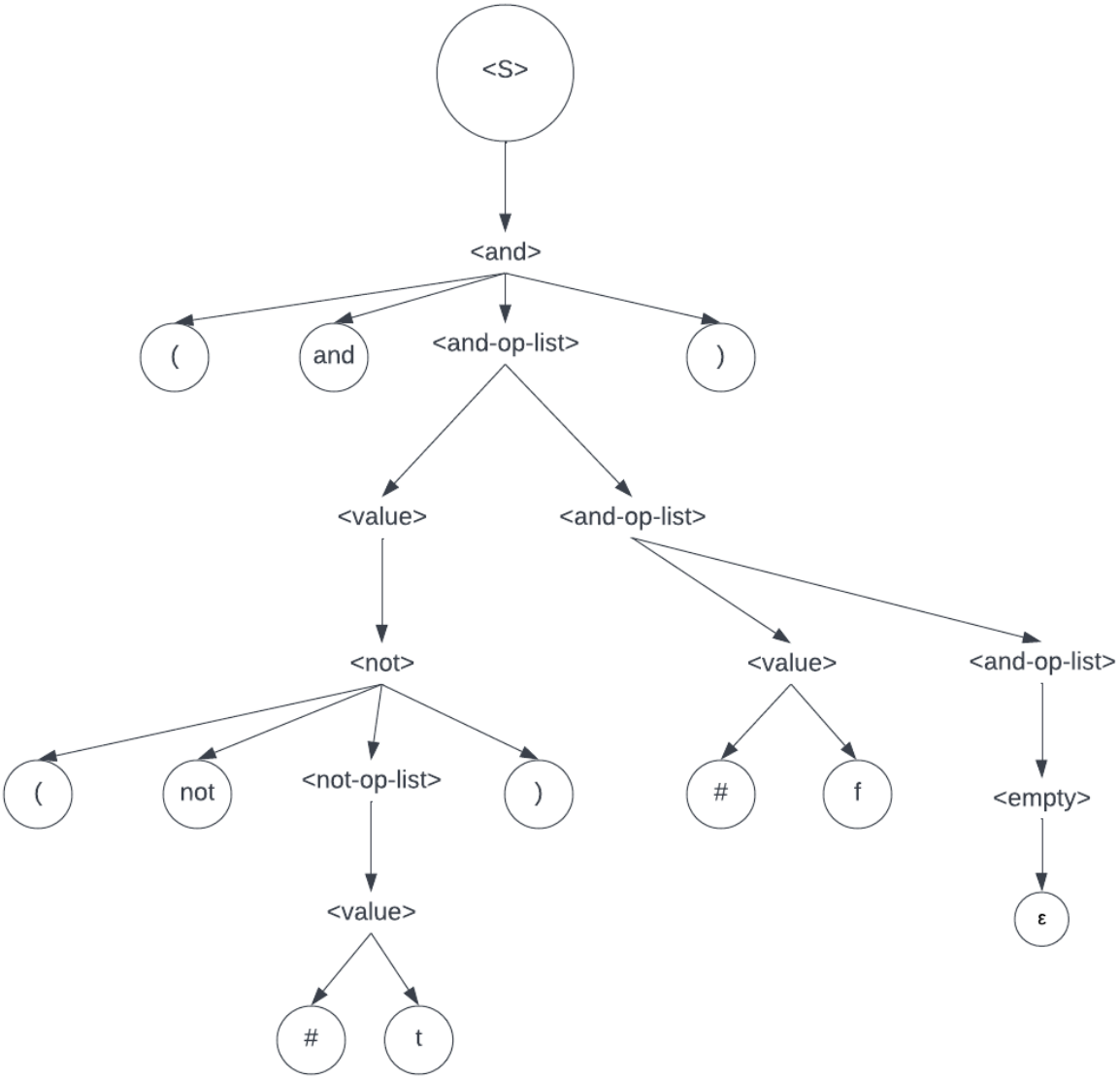
<not-op-list> ::= <value>

<empty> ::= ε

<value> ::= #t | #f | <and> | <or> | <not>

Parse Trees:





Interaction: CF

Grammar:

<S> ::= <operators>

<operators> ::= <add> | <show> | <describe> | <colors>

<add> ::= add <value> <name>

<show> ::= show <name>

<describe> ::= describe <name>

<colors> ::= colors

<value> ::= (<rgb-value> <rgb-value> <rgb-value>) | (<rgb-value> <rgb-value> <rgb-value> <rgb-vaule>)

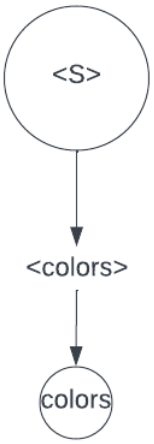
<name> ::= <string>

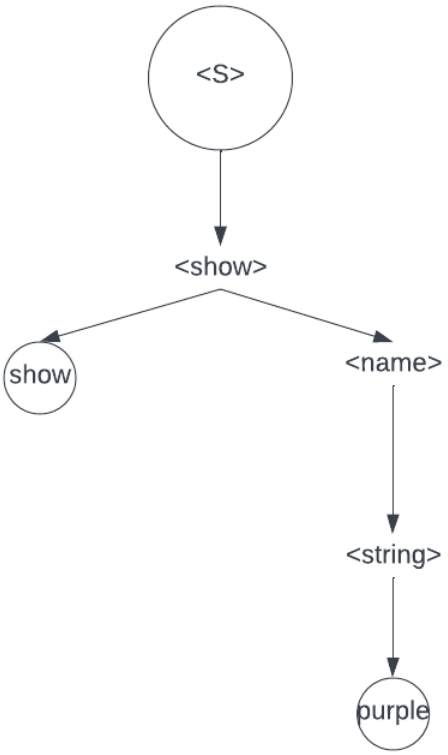
Dictionary:

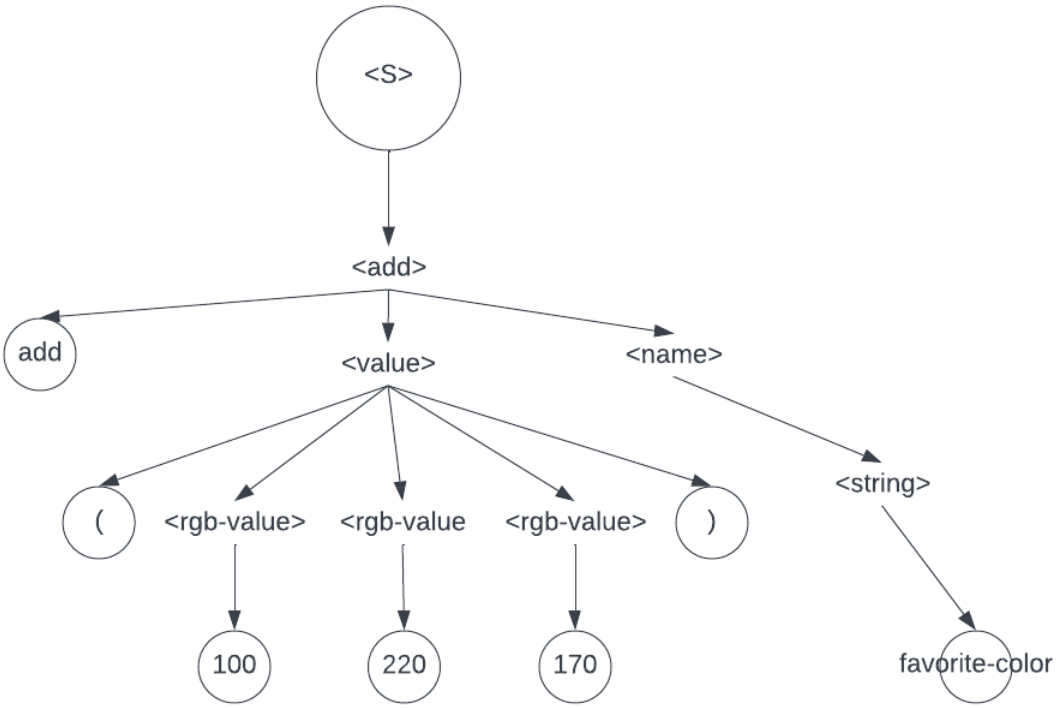
<rgb-value> is defined as any integer value from 0 to 255.

<string> is defined as any sequence of characters.

Parse Trees:




<S>
<show>
show
<name>
<string>
purple


<S>
<add>
add
<value>
<name>
(
<rgb-value>
<rgb-value
<rgb-value>
)
<string>
100
220
170
favorite-color

Interaction: BNF?

What is BNF?

BNF is a way to describe a language that is easier to understand then using a natural language such as English. BNF is a notation that includes the start sequence, rules that define the language which include non-terminal and terminal symbols. When you draw parse-trees using the BNF it will show the reader what is a valid sentence in the language.