CSC 344 BNF First Assignment

Learning Abstract

This assignment focuses on BNF grammars for given languages. This is my first attempt in creating BNF grammars and drawing parse trees.

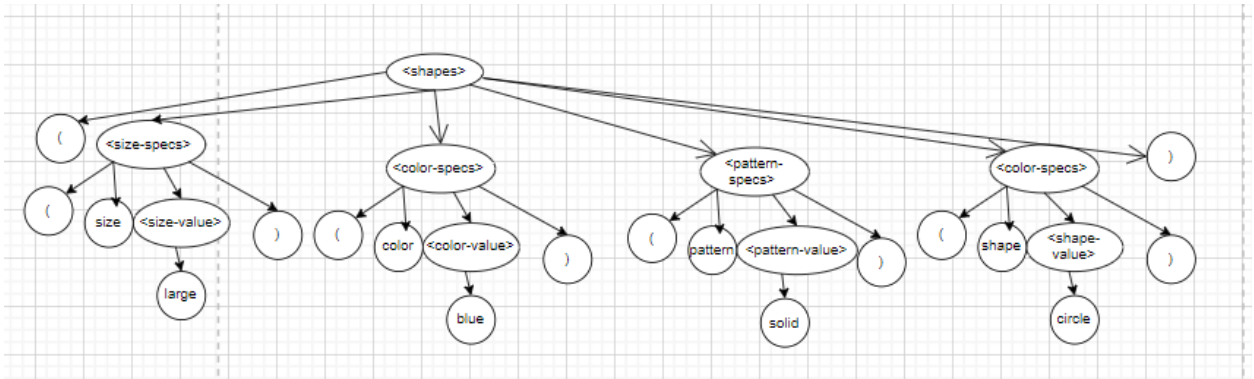
Problem 1 - Shapes

BNF Grammer - Shapes

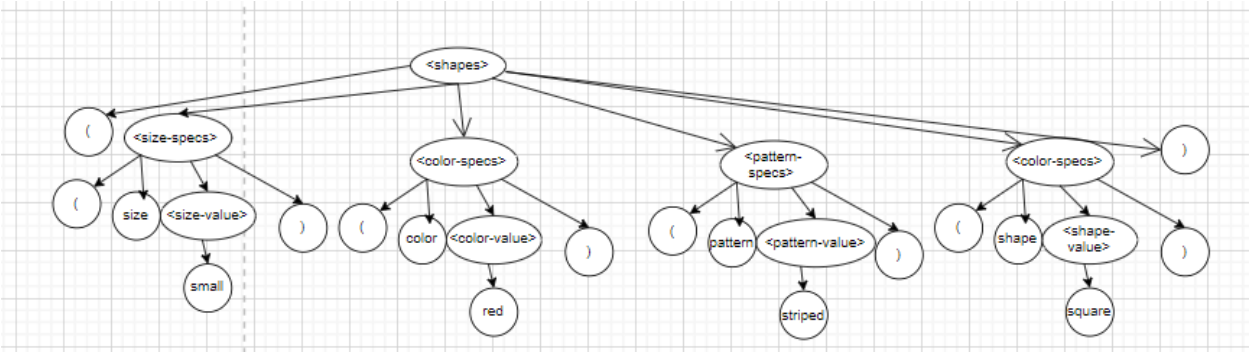
<shapes> : : = (<size-spec> <color-spec> <pattern-spec> <shape-spec>)
<size-spec> : : = (size <size-value>)
<color-spec> : : = (color <color-value>)
<pattern> : : = (pattern <pattern-spec>)
<shape-spec> : : = (shape <shape-value>)
<size-value> : : = large | medium | small
<color-value>: : = red | blue | yellow
<pattern-value> : : = striped | dotted | solid
<shape-value> : : = circle | square | triangle

BNF Parse Trees

1. ((size big) (color blue) (pattern solid) (shape circle))



2. ((size small) (color red) (pattern striped) (shape square))



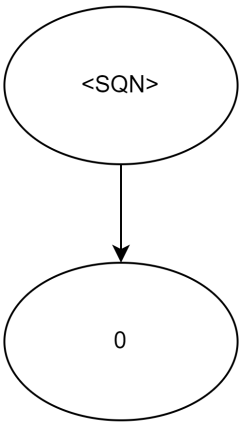
Problem 2 - SQN

BNF Grammer - SQN

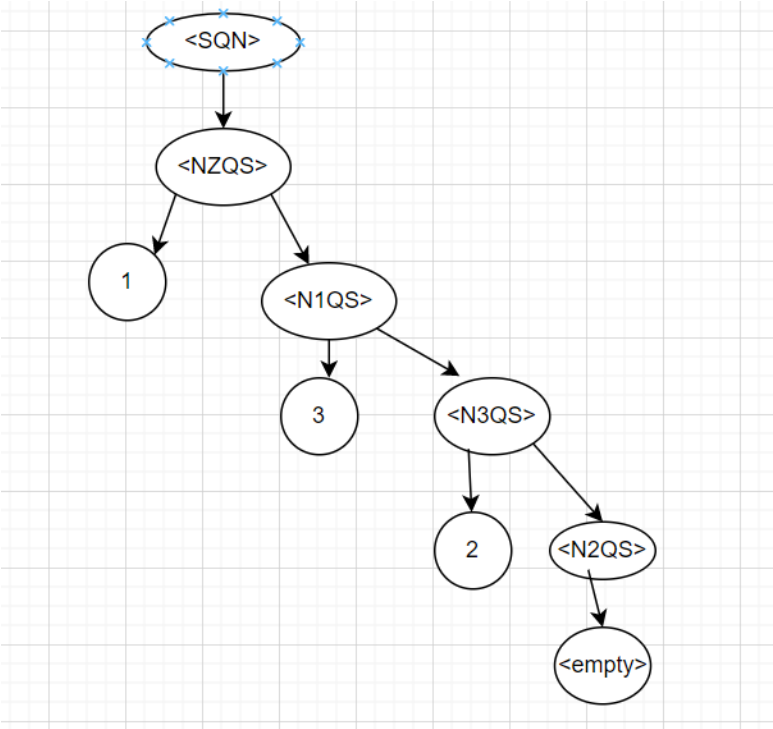
<SQN> : : = <NZQS> | 0
<NZQS> : : = 1 <N1QS> | 2 < N2QS> | 3 <N3QS> | <empty>
<N0S> : : = 1 <N1QS> | 2 < N2QS> | 3 <N3QS> | <empty>
<N1QS> : : = 0 <N0S> | 2 < N2QS> | 3 <N3QS> | <empty>
<N2QS> : : = 0 <N0S> | 1 < N1QS> | 3 <N3QS> | <empty>
<N3QS> : : = 0 <N0S> | 1 < N1QS> | 2 <N2QS> | <empty>

BNF Parse Trees - SQN

1. 0



2. 132



3. We cannot draw a parse tree for 1223 because N2QS does not contain the number 2. It contains 0,1,3 or empty.

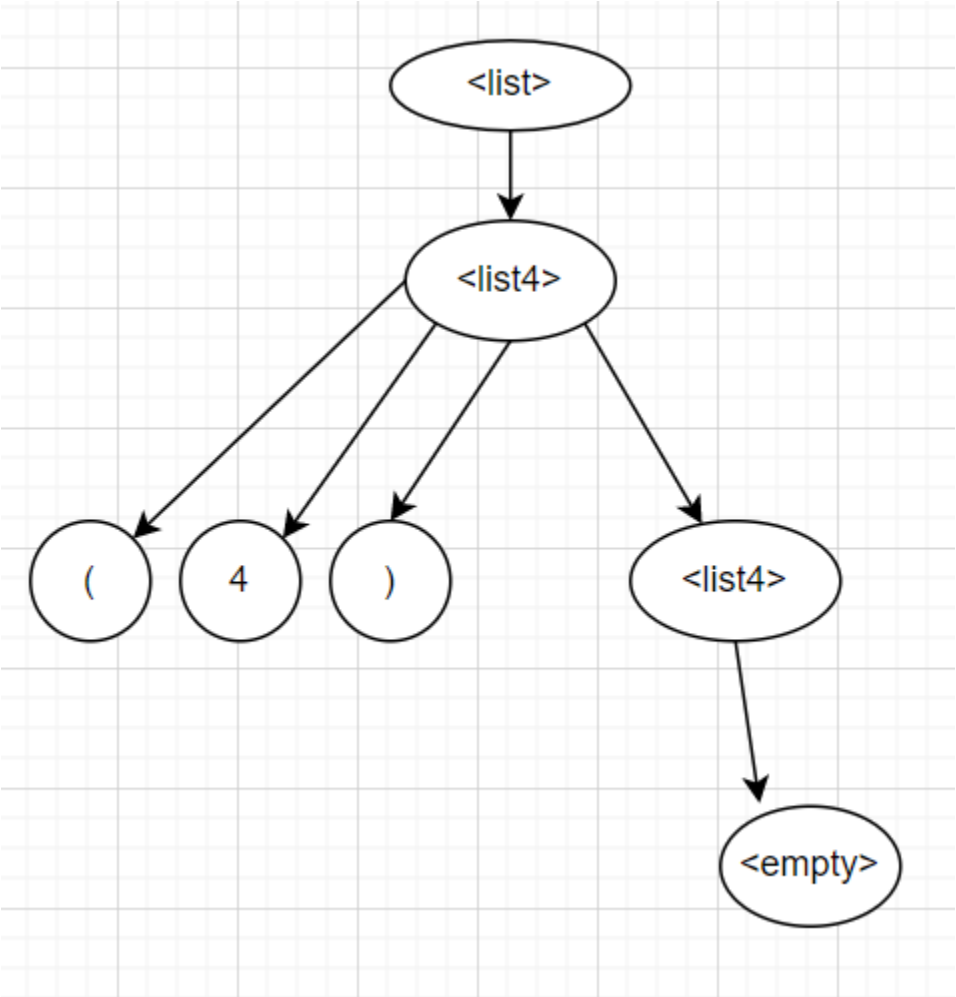
Problem 3 - Fours

BNF Grammer - Fours

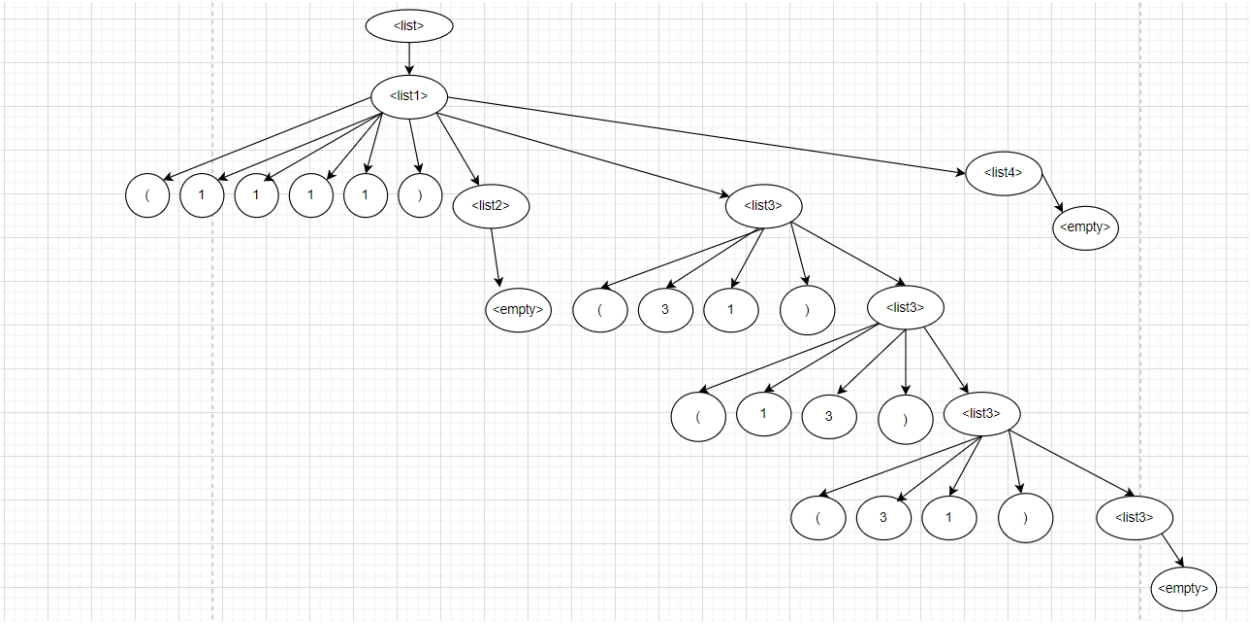
<list> : : = <list1> | <list2> | <list3> | <list4>
<lis1> : : = (1 1 1 1) | (1 1 1 1) <list2> < list3> <list4> | <empty>
<list2> : : = (1 1 2) <list2> | (1 2 1) <list2> | (2 1 1) <list2> | <empty>
<list3> : : = (3 1) <list3> | (1 3) <list3> | <empty>
<list4> : : = (4) <list4> | <empty>

BNF Parse Trees - Fours

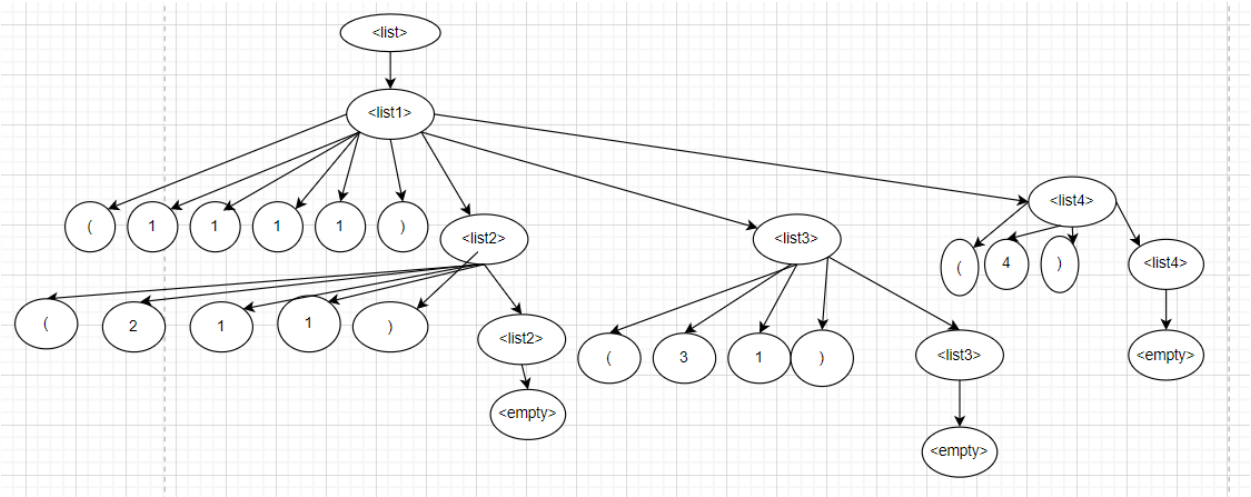
1. (4)



2. (1 1 1 1)(3 1)(1 3)(3 1)



3. (1 1 1 1)(2 1 1)(3 1)(4)



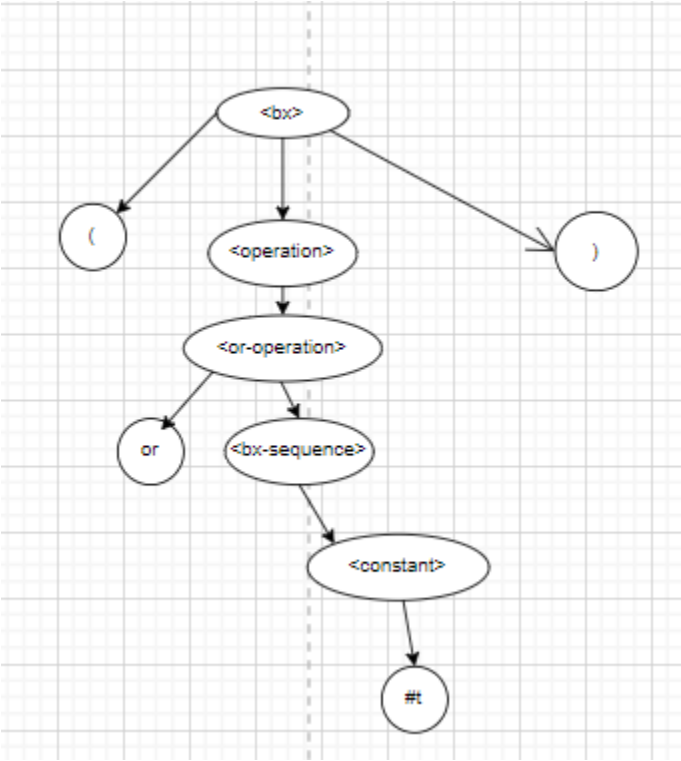
Problem 4 - BXR

<bx> : := <constant> | <operation>
<contant> : := #t | #f

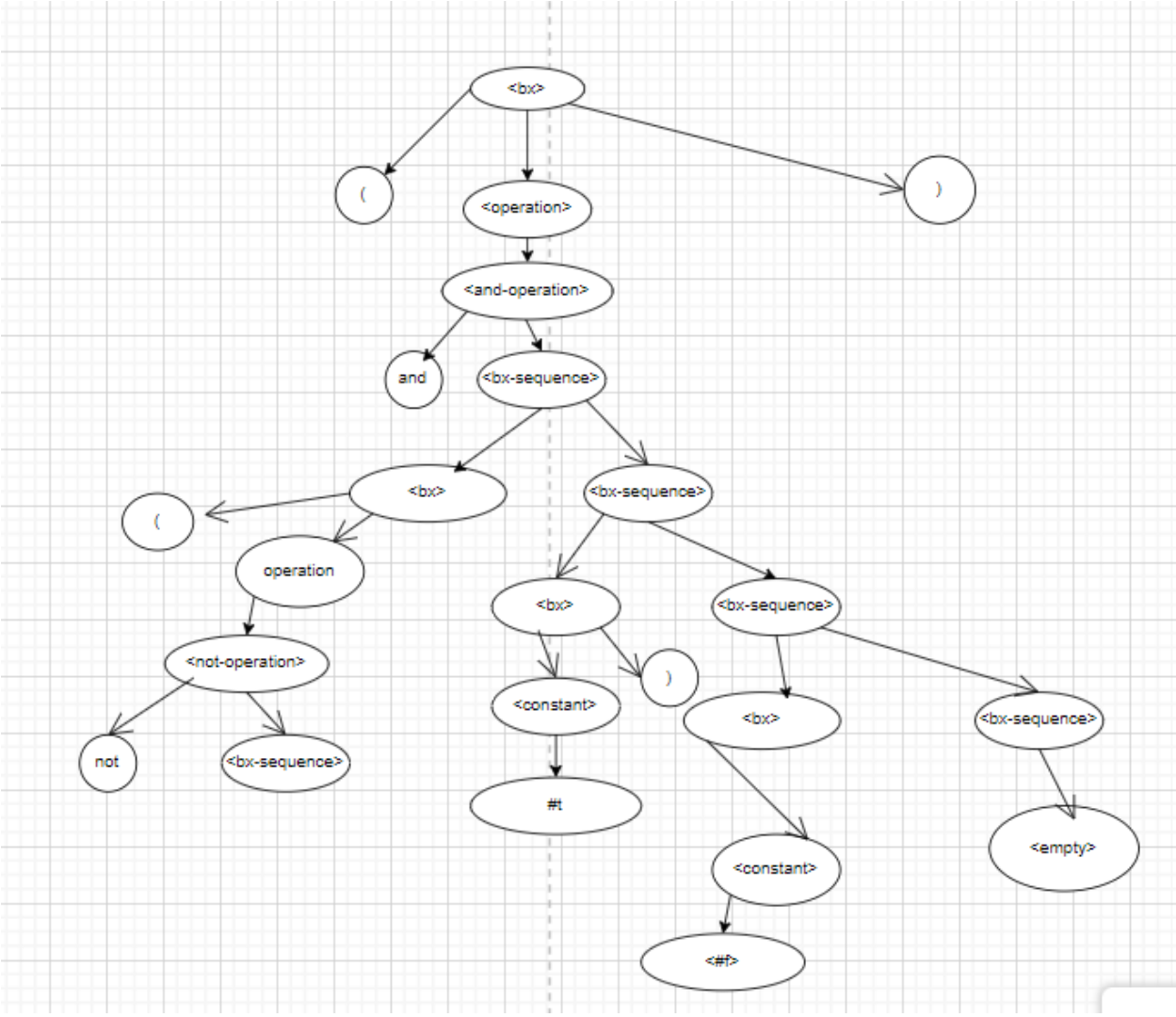
<operation> : := <and-operation> | < or-operation> | <not-operation>
<and-operation> : := (and <bx-sequence>)
<or-operation> : := (or <bx-sequence>)
<not-operation> : : = (not <bx>)
<bx-sequence> : : = <empty> | <bx> <bx-sequence>

BNF Parse Trees - BXR

1. (or #t)



2. (and (not #t) #f)



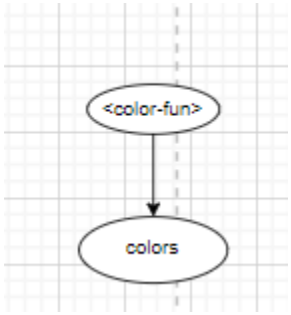
Problem 5 - CF (Color Fun) solution

BNF Grammer - CF

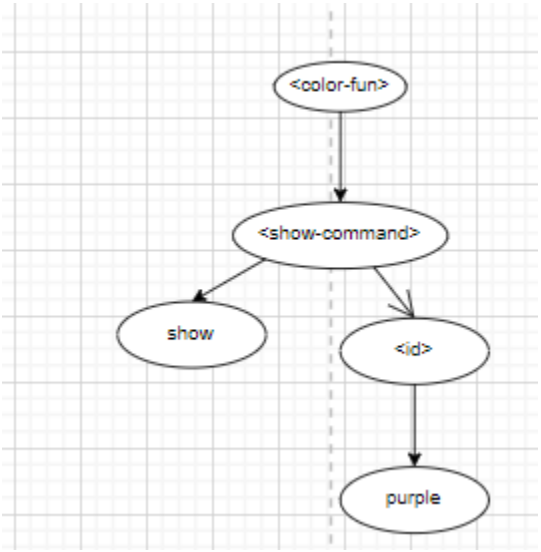
<color-fun> : : = < add-command> | colors | <show-command> | describe-command> | exit
<add-command> : : = add <color> <id> | add color <id>
<show-command> : : = show <id>
<describe-command> : : = describe <id>
<color> : : = (<rgb> <rgb> <rgb>) | (<rgb> <rgb> <rgb> <rgb>)

BNF Parse Trees - CF

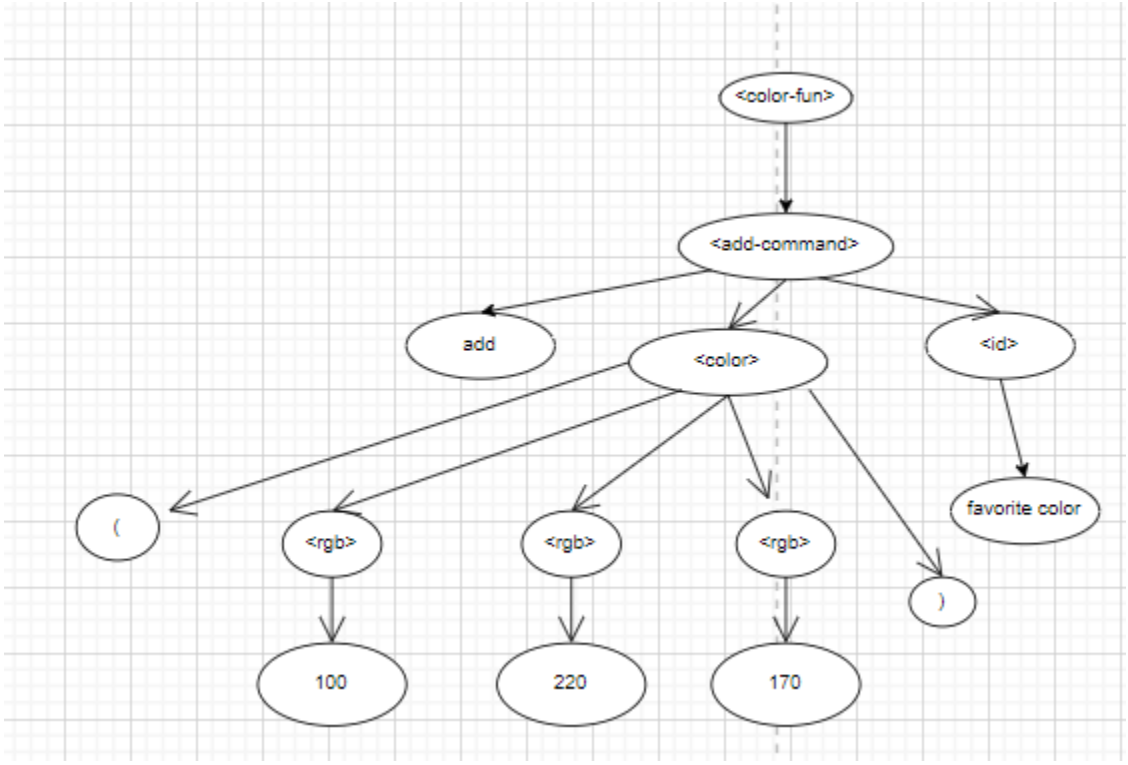
1. colors



2. show purple



3. add (100 220 170) favorite-color



Problem 6 - BNF?

BNF is the way to describe a languages syntax. It is used to represent ambiguos program languges with structure using a set of non-terminal, a set of terminal and rules of production.