Nathaniel Wolf – CSC 344

Problem Set 1 - BNF

Learning Abstract

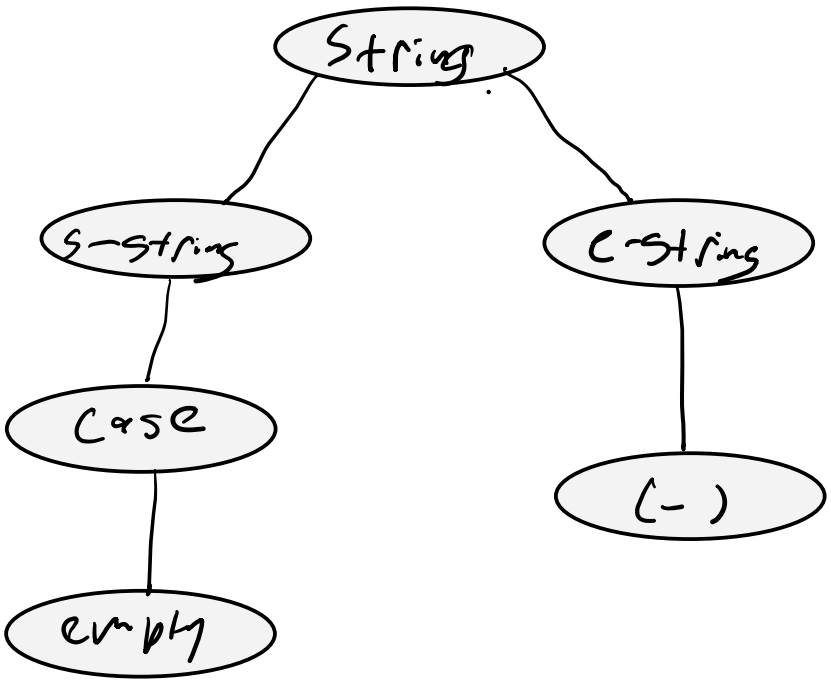
The purpose of this assignment is to gain an understanding of how to reduce a programming language to Backus-Naur Form (which will be referred to as BNF moving forward), diagram parse trees for said language, and then describe concisely what BNF is.

Task 1: RB4B

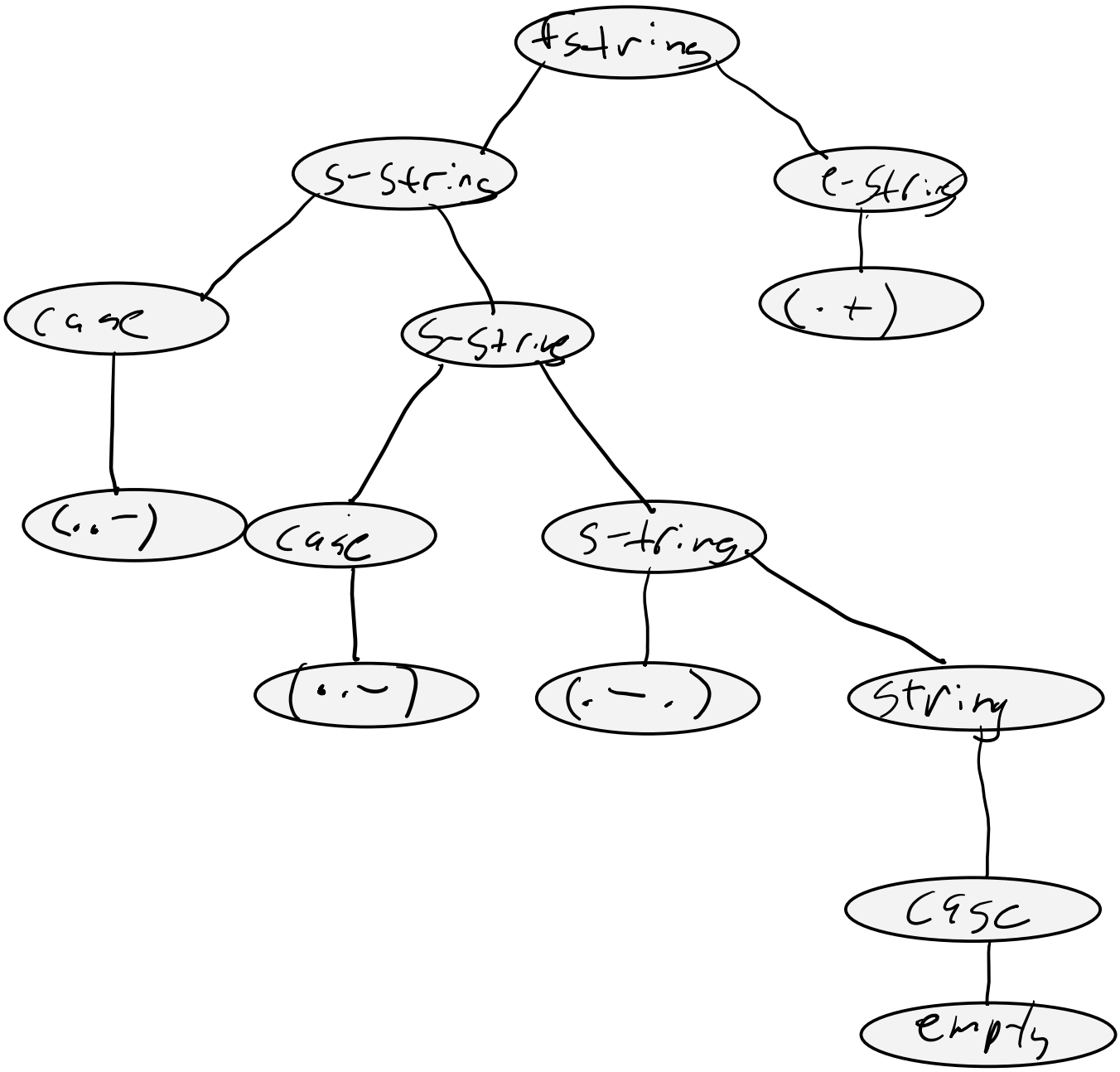
Q1

```
<f-string> ::= <s-string><e-string>
<s-string> ::= <case><-string>
<case> ::= (-) | (--) | (-..) | (..-)
    (.-.) | (.+) | <empty>
<e-string> ::= (-) | (-+)
```

Q2



Q3

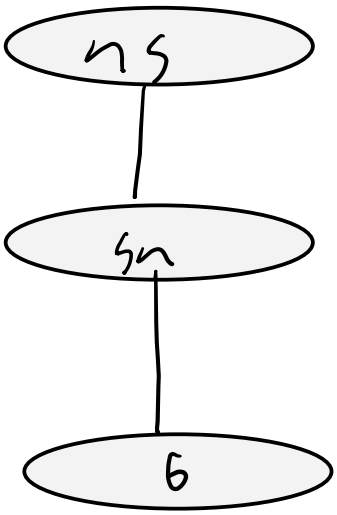

t-string
s-string
e-string
case
s-string
(. +)
(.. -)
case
s-string
(.. ~)
(. — .)
string
case
empty

Task 2: <QNS>

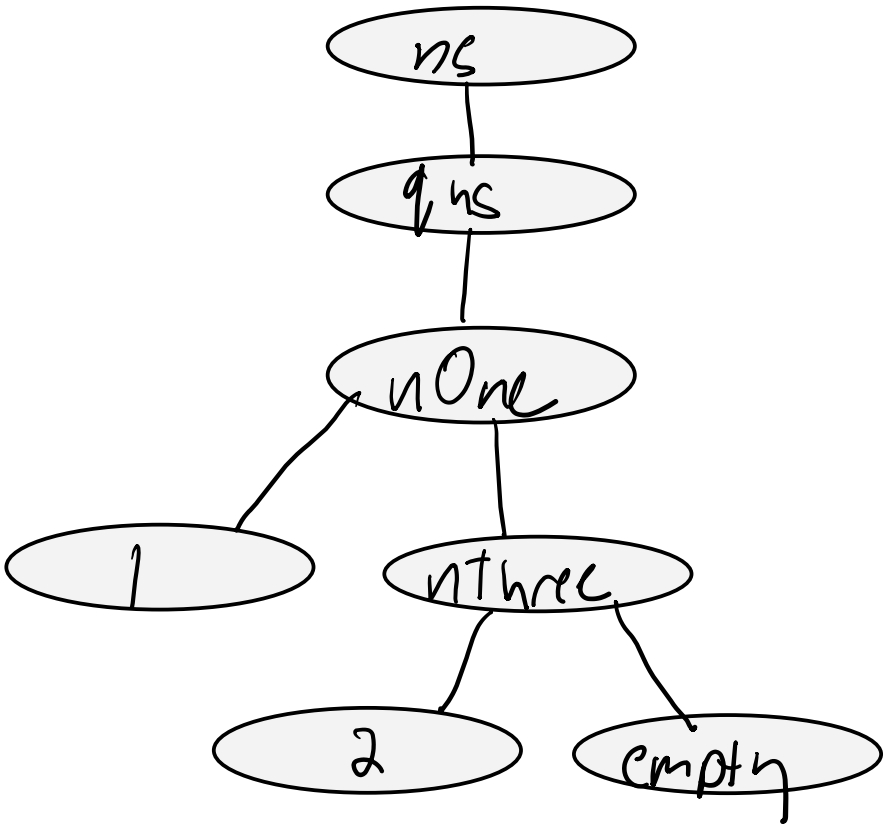
Q1

```
<ns> ::= <sn> | <qns>
<sn> ::= 0 || <qns>
<qns> ::= <nZero> | <nOne> | <nTwo> | <nThree>
<nZero> ::= 0 <nOne> | 0 <nTwo> | 0 <nThree> | 0 <empty>
<nOne> ::= 1 <nZero> | 1 <nTwo> | 0 <nThree> | 1 <empty>
<nTwo> ::= 2 <nZero> | 2 <nOne> | 2 <nTwo> | 2 <empty>
<nThree> ::= 3 <nZero> | 3 <nOne> | 3 <nTwo> | 3 <empty>
```

Q2



Q3



Q4

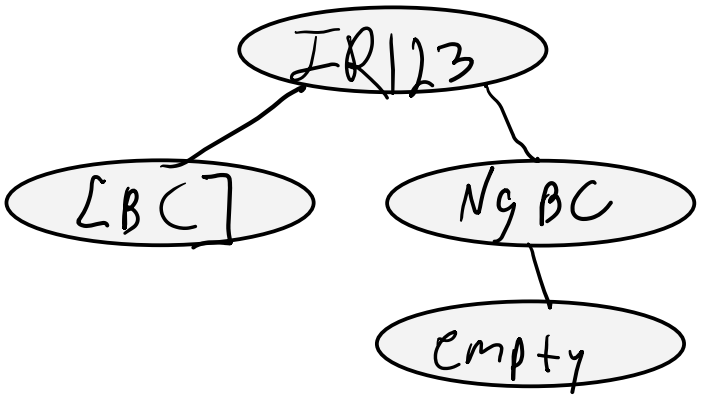
The grammar as defined does not allow for the repetition of digits in a manner that would allow you to have a string of 1223, after the first token of 2 you would have a non-terminal that directs you to choose 0 1 or three.

Task 3: IR123

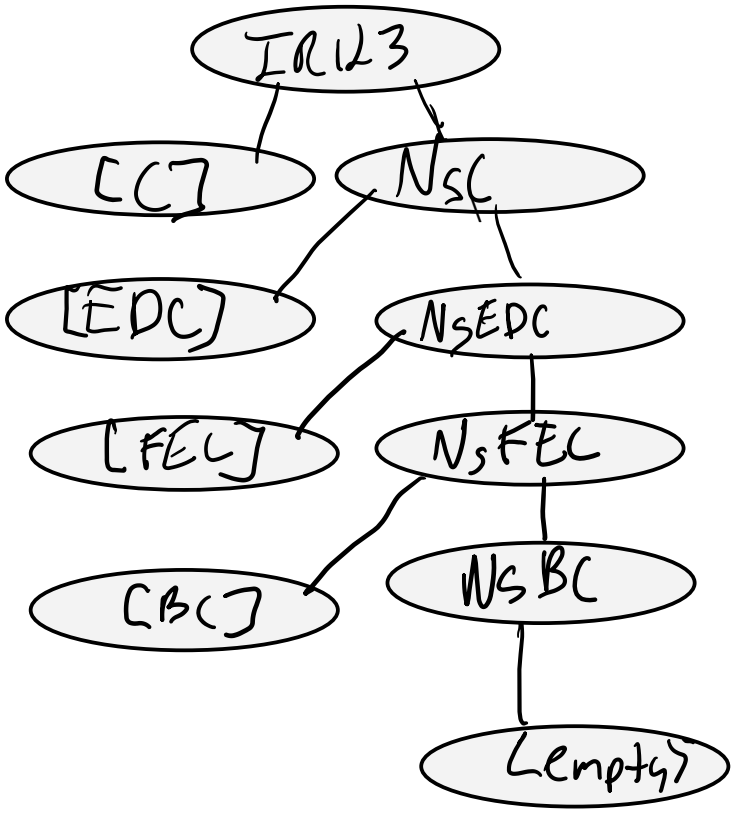
Q1

```
<IR123> ::= [C] <NsC> | [DC] <NsDC> | [BC] <NsBC>
    | [EDC] <NsEDC> | [FEC] <NsFEC> | [GFC] <NsGFC>
<NsC> ::= [DC] <NsDC> | [BC] <NsBC> | [EDC] <NsEDC>
    | [FEC] <NsFEC> | [GFC] <NsGFC> | <empty>
<NsDC> ::= [C] <NsC> | [BC] <NsBC> | [EDC] <NsEDC>
    | [FEC] <NsFEC> | [GFC] <NsGFC> | <empty>
<NsBC> ::= [C] <NsC> | [DC] <NsDC> | [EDC] <NsEDC>
    | [FEC] <NsFEC> | [GFC] <NsGFC> | <empty>
<NsEDC> ::= [C] <NsC> | [DC] <NsDC> | [BC] <NsBC>
    | [FEC] <NsFEC> | [GFC] <NsGFC> | <empty>
<NsFEC> ::= [C] <NsC> | [DC] <NsDC> | [BC] <NsBC>
    | [EDC] <NsEDC> | [GFC] <NsGFC> | <empty>
<NsGFC> ::= [C] <NsC> | [DC] <NsDC> | [BC] <NsBC>
    | [EDC] <NsEDC> | [FEC] <NsFEC> | <empty>
```

Q2



Q3



Q4

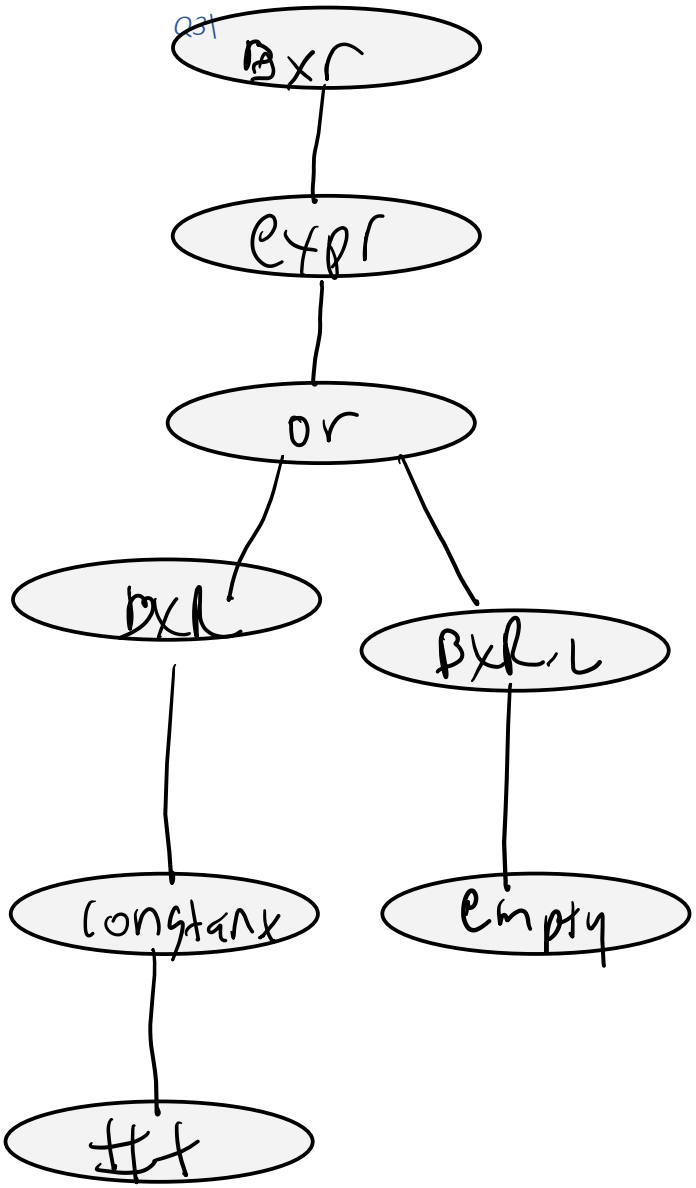
The strings [BC][BC] are repeated is sequence. Per the BNF grammar for this language you cannot have repeating strings after a non-terminal. After BC you have to a select a NsBC token next.

Task 3

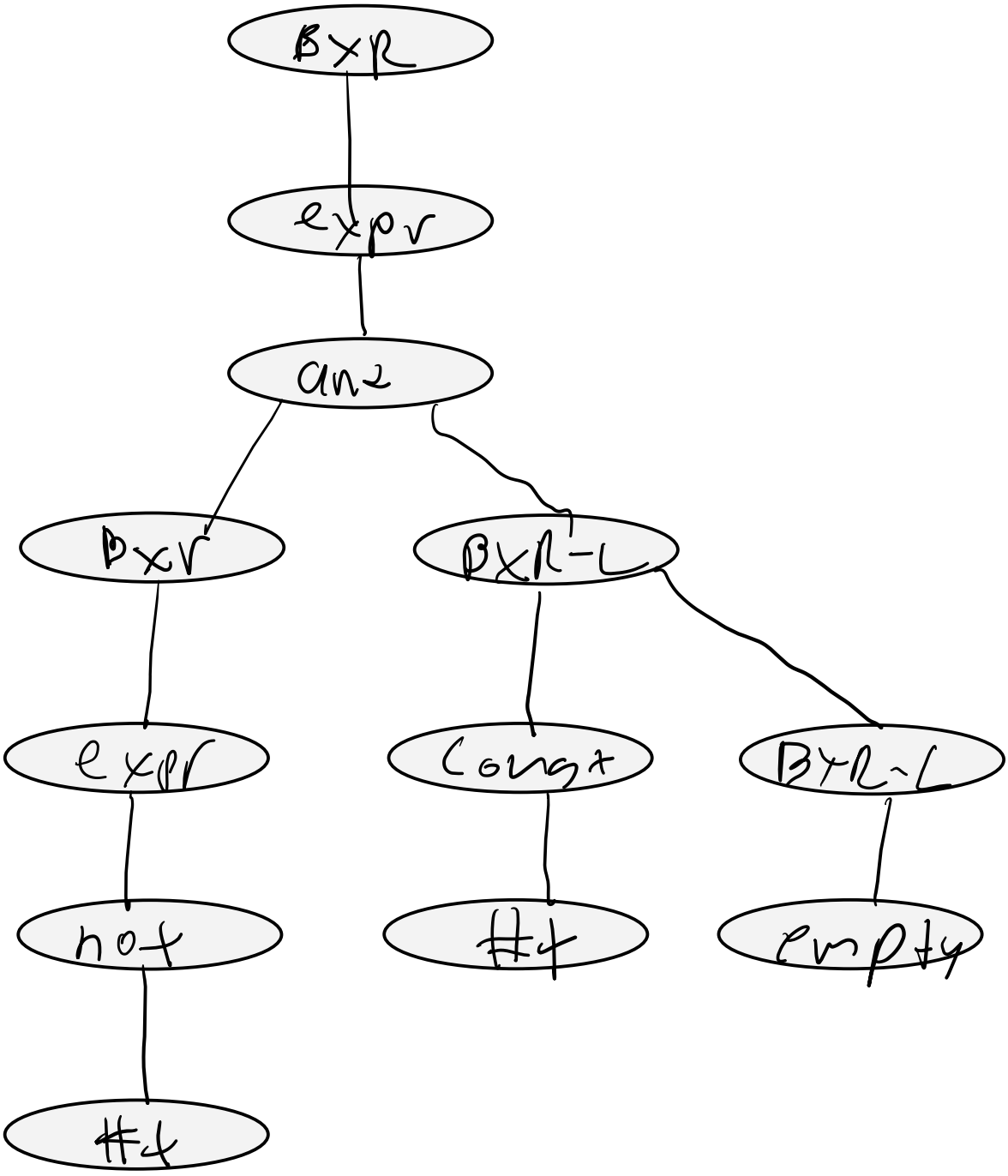
Q1 BXR

```
<BXR> ::= <expr> | <const>
<BXR-1> ::= <BXR> | <empty>
<expr> ::= <or> | <and> | <not>
<and> ::= (and <BXR> <BXR-1)
<or> ::= (or <BXE> <BXR-1)
<not> ::= ( not <const> | not <expr> )
<const> ::= #t | #f
```

Q2



Q4

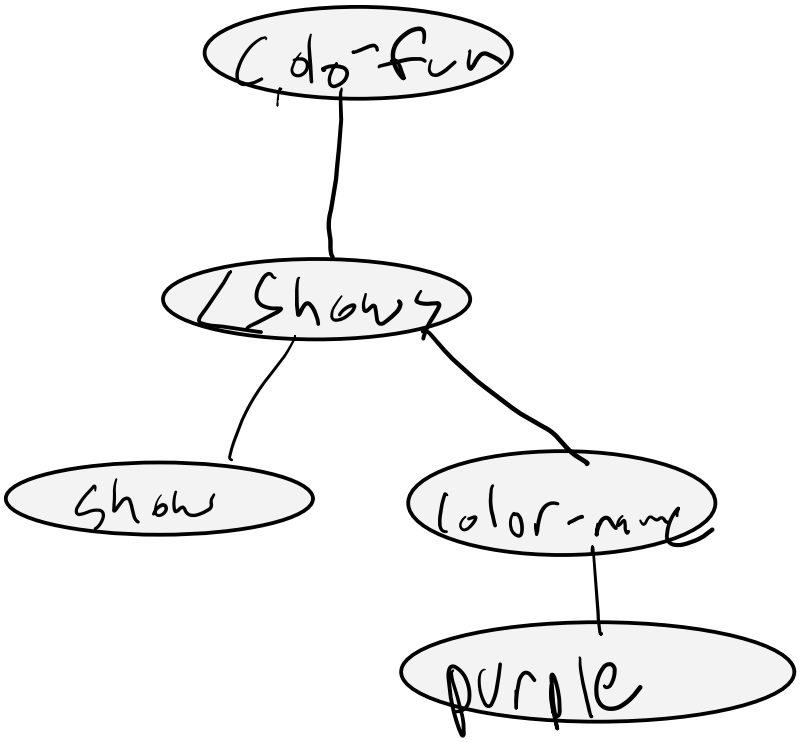


Task 5: Color Fun

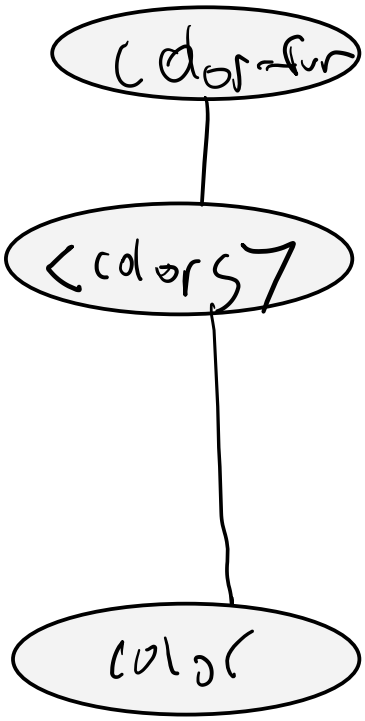
Q1

```
<color-fun> ::= <add> | <show> | <desc> | <colors> | <exit>
<add-color> ::= add <RGB> color-name
<RGB> ::= <0...255><0...255><0..255>|<0...255><0...255><0..255>
<show> ::= show color-name
<desc> ::= describe color-name
<colors> ::= <color-l> | <empty>
<colors-l> ::= color-name | <colors-l> | <empty>
<exit> ::= /terminates program
```

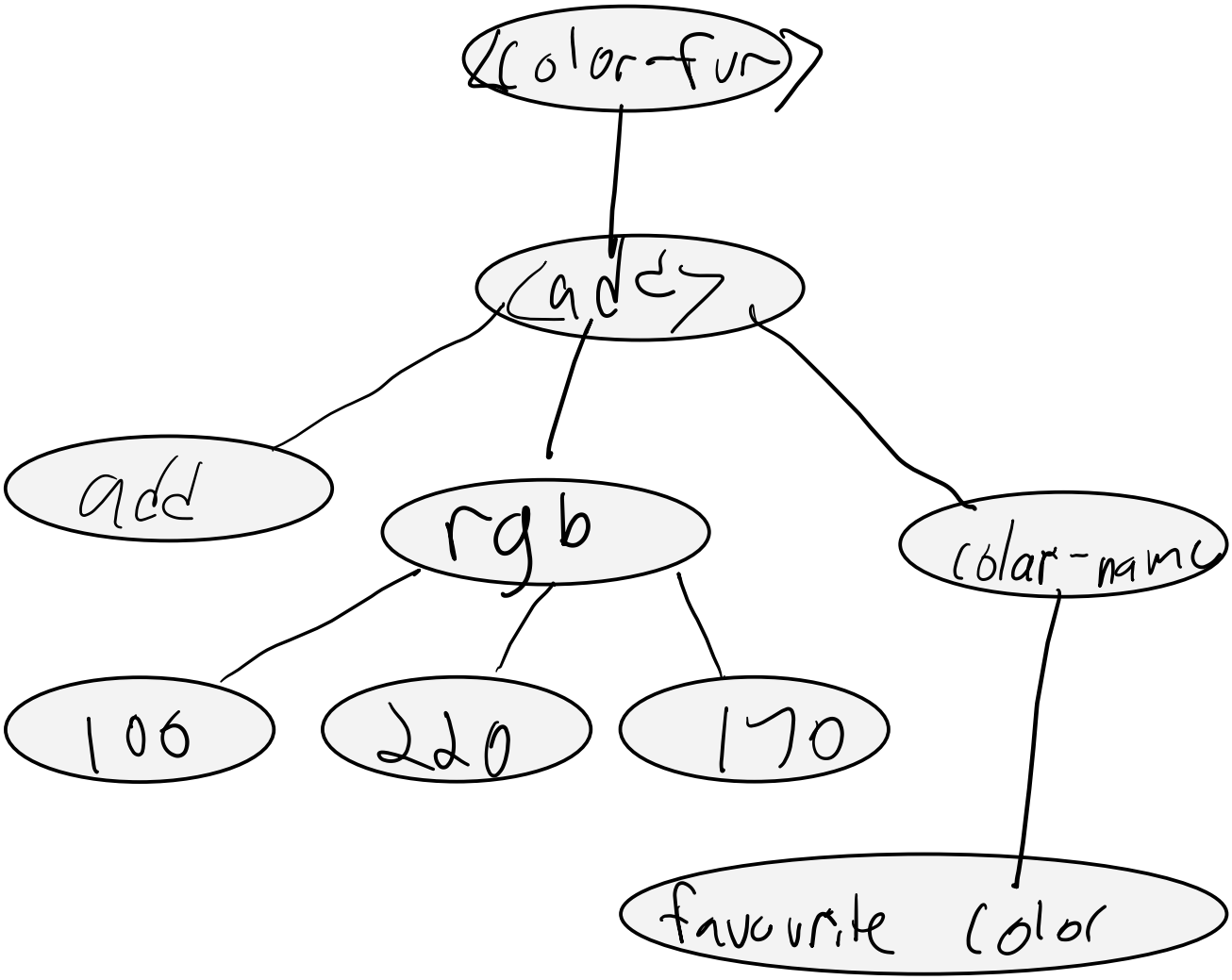
Q4



Q3



Q4


color-fun
addy
add
rgb
colar-name
106
220
170
favourite color

Task 6: Explain BNF Form in your own words

Backus-Naur Form (BNF) is a mechanism that we use to describe a programming languages meta syntax and how the language essentially works on a basic level. Using BNF we can strip away as many high and low order functions of a particular language to determine how things like variables, statements, expressions (assuming any of those exist to said language, they don't necessarily have to ascribe to having a particular concept to be considered a programming language). In theory what remains of this stripping would be a grammar language corresponding with the general principles of putting a language in BNF.