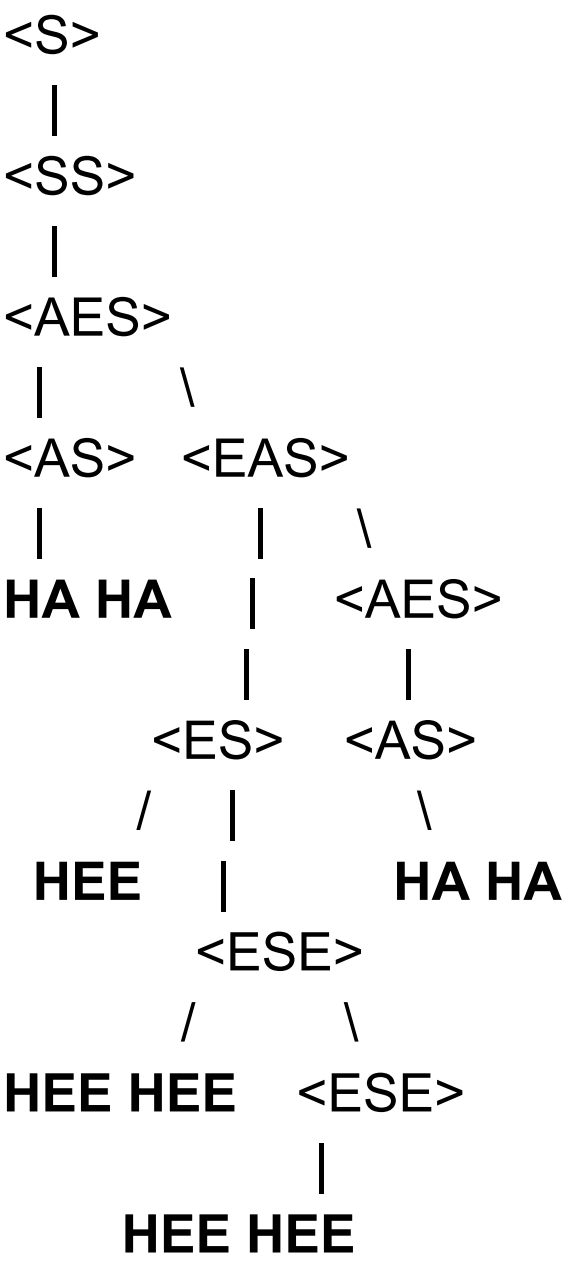
Timothy Germakovski Csc344 BNF Assignment

Task 1: “Laugher”

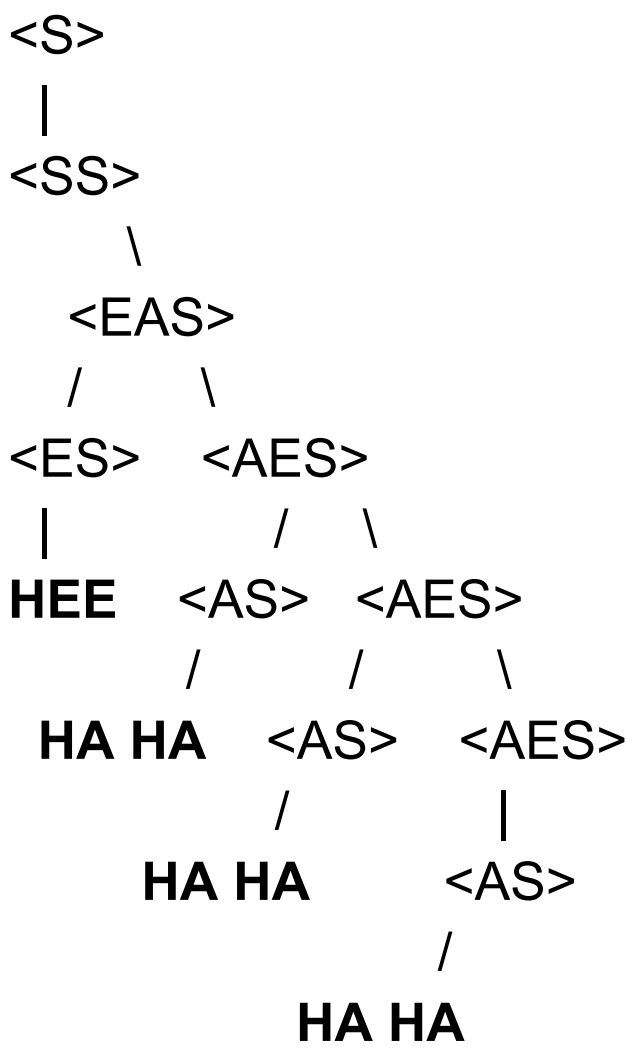
1.1

<S> ::= <AS> | <ES> | <SS>
<SS> ::= <AES> | <EAS>
<AES> ::= <AS> | <AS><EAS>
<EAS> ::= <ES> | <ES><AES>
<AS> ::= HA HA | HA HA <AS>
<ES> ::= HEE | HEE <ESE>
<ESE> ::= HEE HEE | HEE HEE <ESE>

1.2



1.3



Task 2: “SQN”

2.1

<S> ::= 0 | <ZR>
<ZR> ::= 1 <ON> | 2 <TW> | 3 <TR> | 4 <FR>
<ZR> ::= 1 | 2 | 3 | 4
<ON> ::= 2 <TW> | 3 <TR> | 4 <FR> | 0 <ZR>
<ON> ::= 2 | 3 | 4 | 0
<TW> ::= 3 <TR> | 4 <FR> | 0 <ZR> | 1 <ON>
<TW> ::= 3 | 4 | 0 | 1
<TR> ::= 4 <FR> | 0 <ZR> | 1 <ON> | 2 <TW>
<TR> ::= 4 | 0 | 1 | 2
<FR> ::= 0 <ZR> | 1 <ON> | 2 <TW> | 3 <TR>
<FR> ::= 0 | 1 | 2 | 3

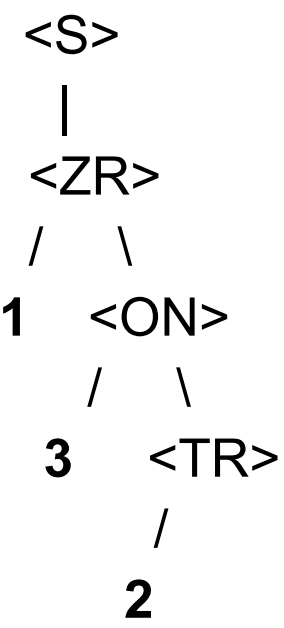
2.2

<S>

|

0

2.3



2.4

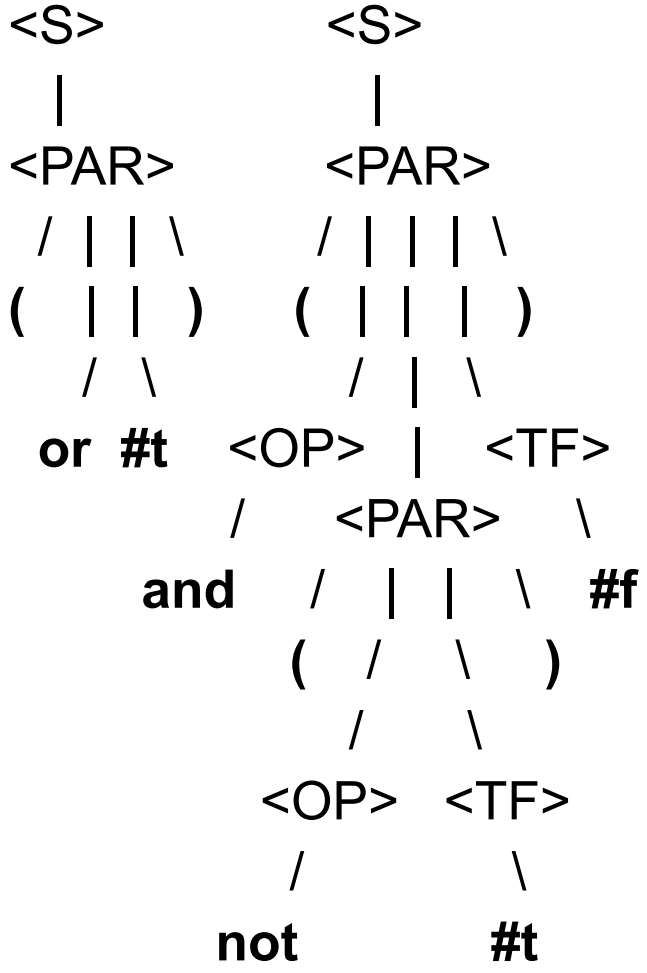
You cannot write 1223 because the only nonterminals that have a 2 on the right side of the ::= sign are of the form 2 <TW> (where <TW> is any quaternary or sequence of quaternaries that does not start with 2) or simply 2, with nothing after. Therefore any string of quaternaries containing a double 2 is impossible.

Task 3: “BXR”

3.1

<S> ::= <TF> | <PAR>
<TF> ::= #t | #f
<OP> ::= and | or | not
<PAR> ::= (<OP> <TF>)
<PAR> ::= (<OP> <PAR> <TF>)
<PAR> ::= (<OP> <TFS>)
<PAR> ::= (<OP> <PAR> <TFS>)
<TFS> ::= <TF> | <TF> < TFS>

3.2



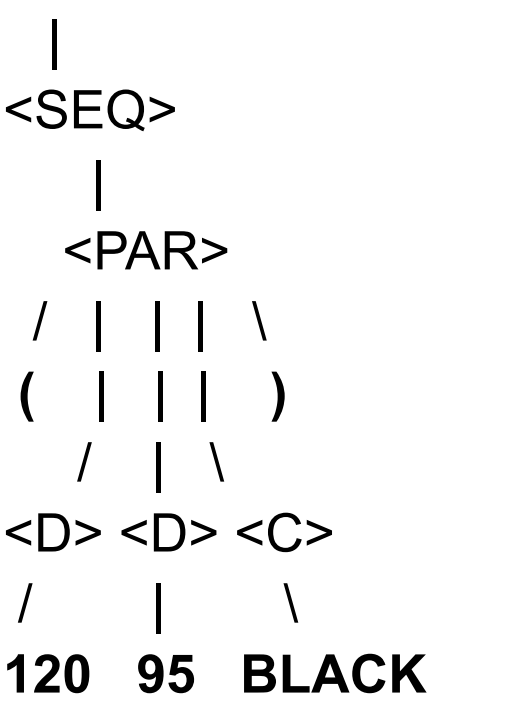
3.3

Task 4: “Line Segment Seq”

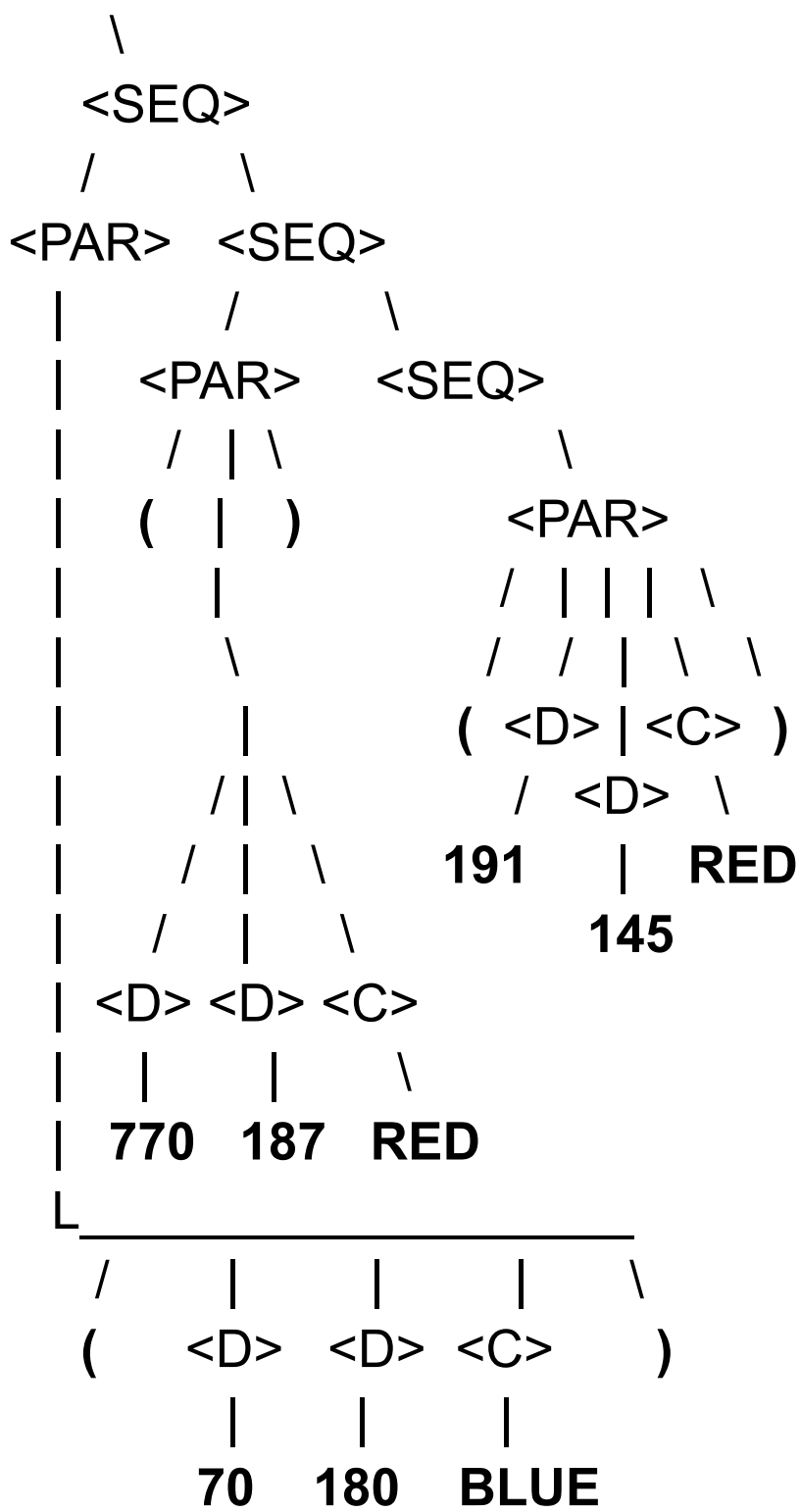
4.1

<S> ::= <SEQ> | <empty>
<SEQ> ::= <PAR> | <PAR><SEQ>
<PAR> ::= (<D> <D> <C>)
<D> ::= * some number above zero *
<C> ::= RED | BLACK | BLUE

4.2



4.3

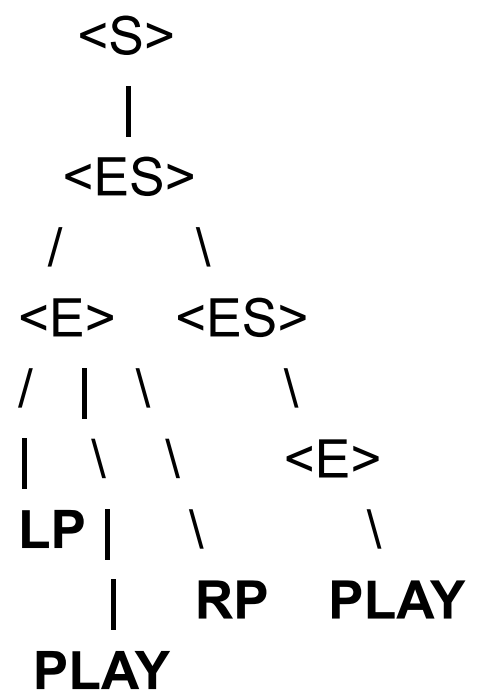

<S>
<SEQ>
<PAR>
<SEQ>
<PAR>
<SEQ>
(
)
<PAR>
(<D> | <C>)
<D>
191
RED
145
<D> <D> <C>
770
187
RED
(<D> <D> <C>)
70
180
BLUE

Task 5: “M-Lines”

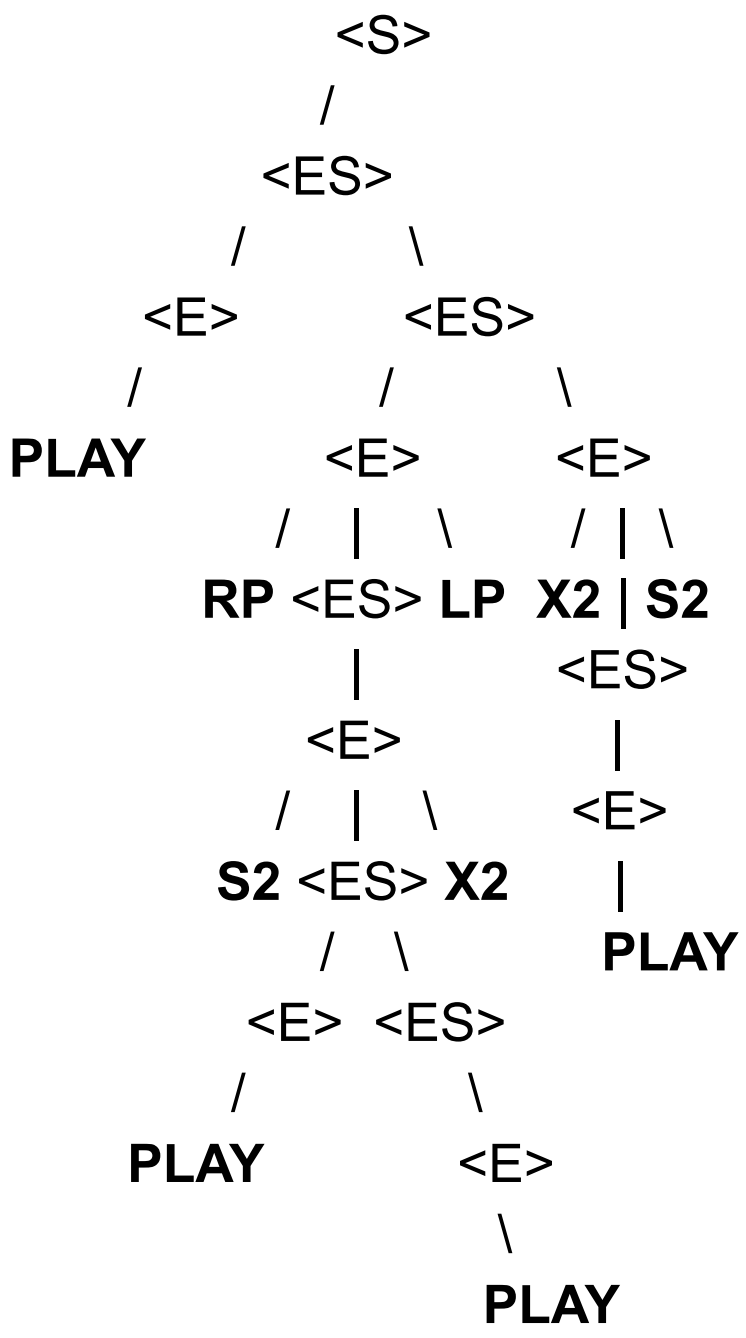
5.1

<S> ::= <ES> | <empty>
<ES> ::= <E> | <E> <ES>
<E> ::= PLAY | REST
<E> ::= RP <ES> LP | LP <ES> RP
<E> ::= S2 <ES> X2 | X2 <ES> S2
<E> ::= X3 <ES> S3 | S3 <ES> X3
<E> ::= X2 <ES> X2

5.2



5.3



Task 6:

BNF is a system for breaking down the syntax of some language or language-like system. It categorizes tokens (pieces of text) into a hierarchy/network of types of tokens (non-terminals). Ideally, a good BNF allows you to construct every possible sentence (given the limitations of the BNF) using these tokens and nonterminals