

Assignment #3

Trevor Strickland, CSC344

September 21, 2022

Learning Abstract: This assignment deals with creating Backus-Noir Forms (BNFs) to describe different languages. BNFs aid in the designing of language syntax and form, and are a (relatively) easy-to-comprehend way of recording how languages function and what requirements there are.

Problem #1: Shapes

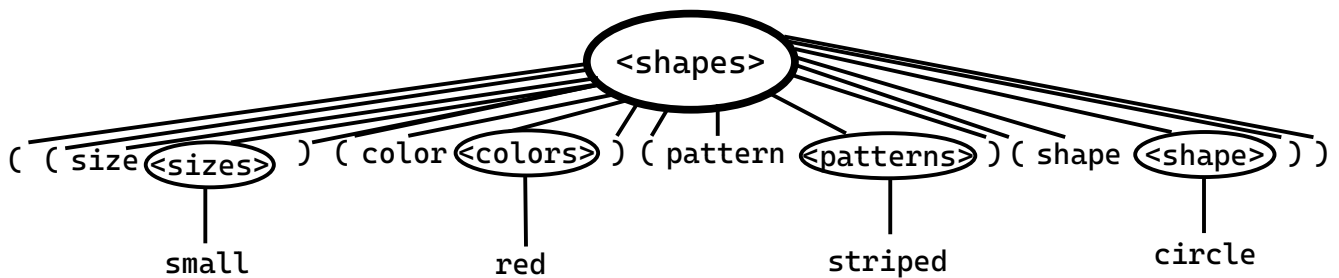
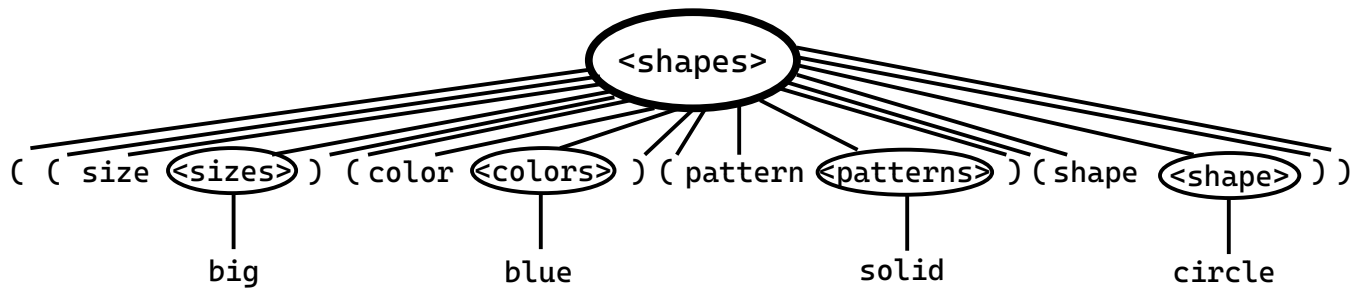
```
<shapes> ::= ( ( size <sizes> ) ( color <colors> ) ( pattern <patterns> )  
( shape <shape> ) )
```

```
<sizes> ::= small | medium | big
```

```
<colors> ::= blue | yellow | red
```

```
<patterns> ::= solid | dotted | striped
```

```
<shape> ::= circle | scircle | square | triangle
```



Problem #2: SQN (Special Quaternary Numbers)

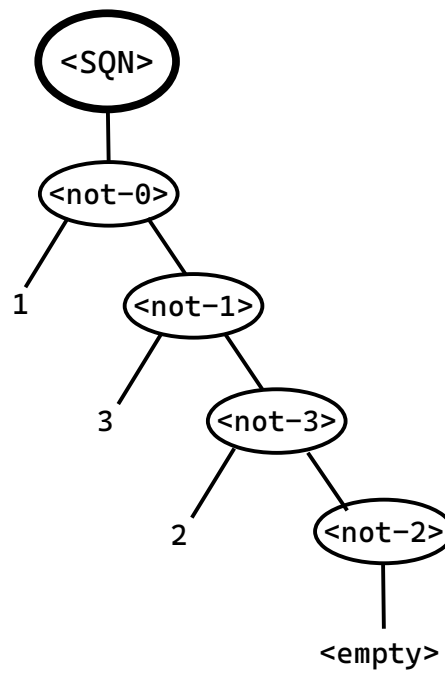
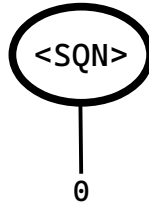
$\langle \text{SQN} \rangle ::= 0 \mid \langle \text{not-0} \rangle$

$\langle \text{not-0} \rangle ::= 1 \langle \text{not-1} \rangle \mid 2 \langle \text{not-2} \rangle \mid 3 \langle \text{not-3} \rangle \mid \langle \text{empty} \rangle$

$\langle \text{not-1} \rangle ::= 0 \langle \text{not-0} \rangle \mid 2 \langle \text{not-2} \rangle \mid 3 \langle \text{not-3} \rangle \mid \langle \text{empty} \rangle$

$\langle \text{not-2} \rangle ::= 0 \langle \text{not-0} \rangle \mid 1 \langle \text{not-1} \rangle \mid 3 \langle \text{not-3} \rangle \mid \langle \text{empty} \rangle$

$\langle \text{not-3} \rangle ::= 0 \langle \text{not-0} \rangle \mid 1 \langle \text{not-1} \rangle \mid 2 \langle \text{not-2} \rangle \mid \langle \text{empty} \rangle$



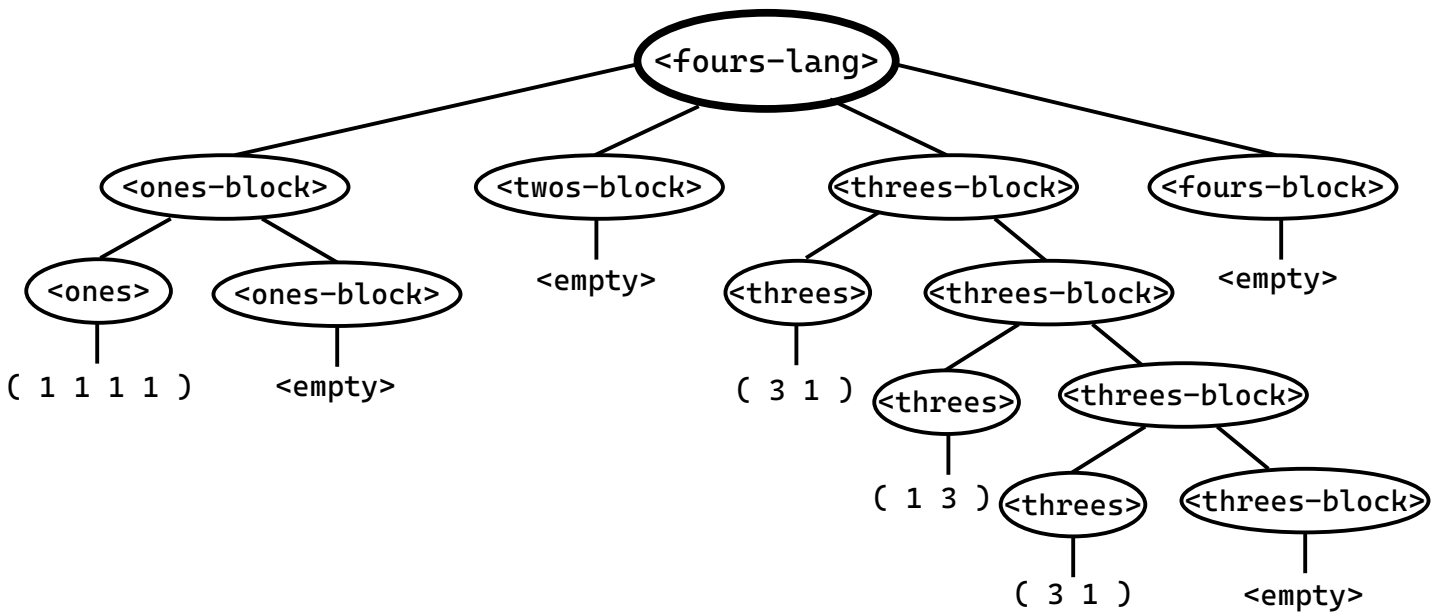
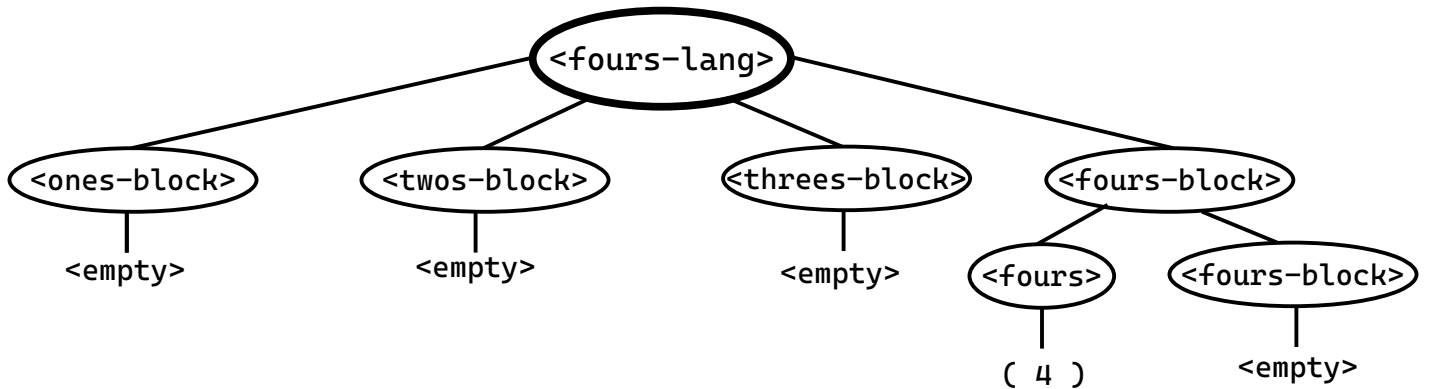
A parse tree for “1223” wouldn’t be possible with this grammar, as the second character 2 requires an item from $\langle \text{not-2} \rangle$, which can either be 0, 1, 3, or the number terminates there.

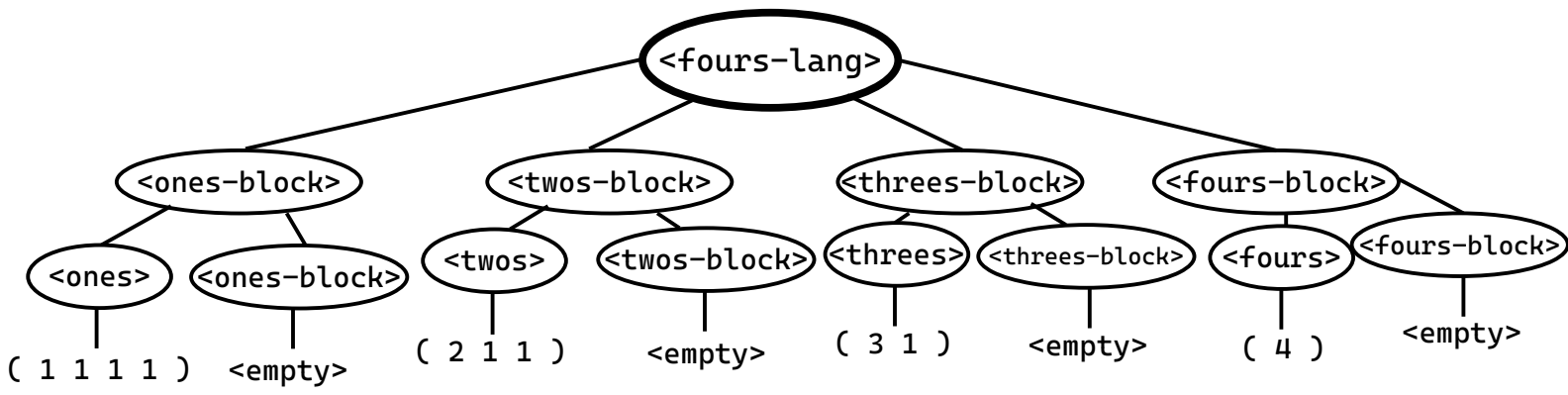
Problem #3: Fours

$\langle\text{fours-lang}\rangle ::= \langle\text{ones-block}\rangle \langle\text{twos-block}\rangle \langle\text{threes-block}\rangle \langle\text{fours-block}\rangle$

$\langle\text{ones-block}\rangle ::= \langle\text{empty}\rangle \mid \langle\text{ones}\rangle \langle\text{ones-block}\rangle$
 $\langle\text{twos-block}\rangle ::= \langle\text{empty}\rangle \mid \langle\text{twos}\rangle \langle\text{twos-block}\rangle$
 $\langle\text{threes-block}\rangle ::= \langle\text{empty}\rangle \mid \langle\text{threes}\rangle \langle\text{threes-block}\rangle$
 $\langle\text{fours-block}\rangle ::= \langle\text{empty}\rangle \mid \langle\text{fours}\rangle \langle\text{fours-block}\rangle$

$\langle\text{ones}\rangle ::= (1 1 1 1)$
 $\langle\text{twos}\rangle ::= (1 1 2) \mid (1 2 1) \mid (2 1 1)$
 $\langle\text{threes}\rangle ::= (3 1) \mid (1 3)$
 $\langle\text{fours}\rangle ::= (4)$





Problem #4: BXR

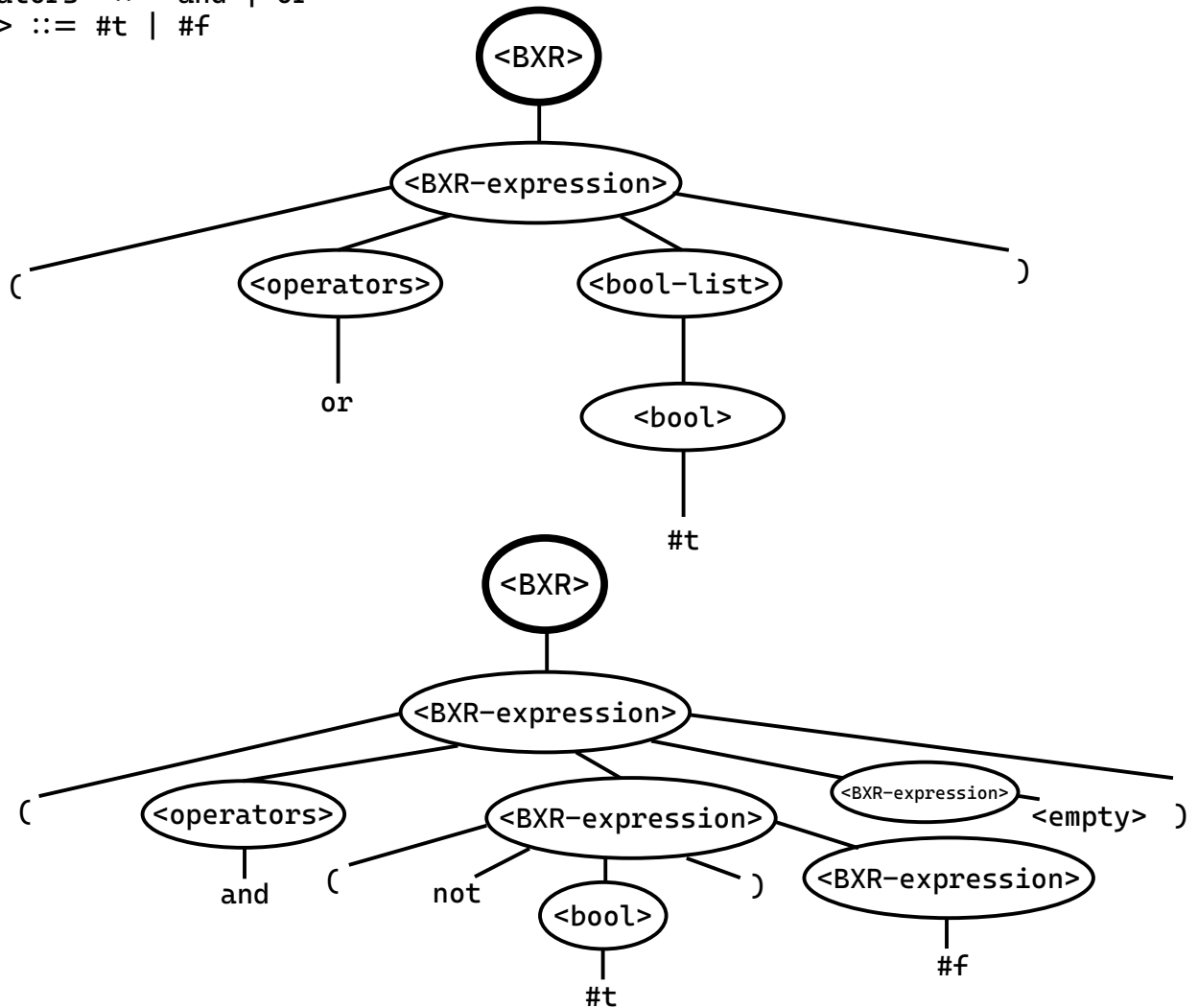
$\langle \text{BXR} \rangle ::= \langle \text{empty} \rangle \mid \langle \text{BXR-expression} \rangle$

$\langle \text{BXR-expression} \rangle ::= (\langle \text{operators} \rangle \langle \text{BXR-expression} \rangle) \langle \text{BXR-expression} \rangle \mid (\langle \text{operators} \rangle \langle \text{bool-list} \rangle) \mid (\text{not } \langle \text{bool} \rangle) \langle \text{BXR-expression} \rangle \mid \langle \text{bool} \rangle \mid \langle \text{empty} \rangle$

$\langle \text{bool-list} \rangle ::= \langle \text{bool} \rangle \langle \text{bool-list} \rangle \mid \langle \text{bool} \rangle$

$\langle \text{operators} \rangle ::= \text{and} \mid \text{or}$

$\langle \text{bool} \rangle ::= \#t \mid \#f$



Problem #5: CF

`<CF> ::= <command>`

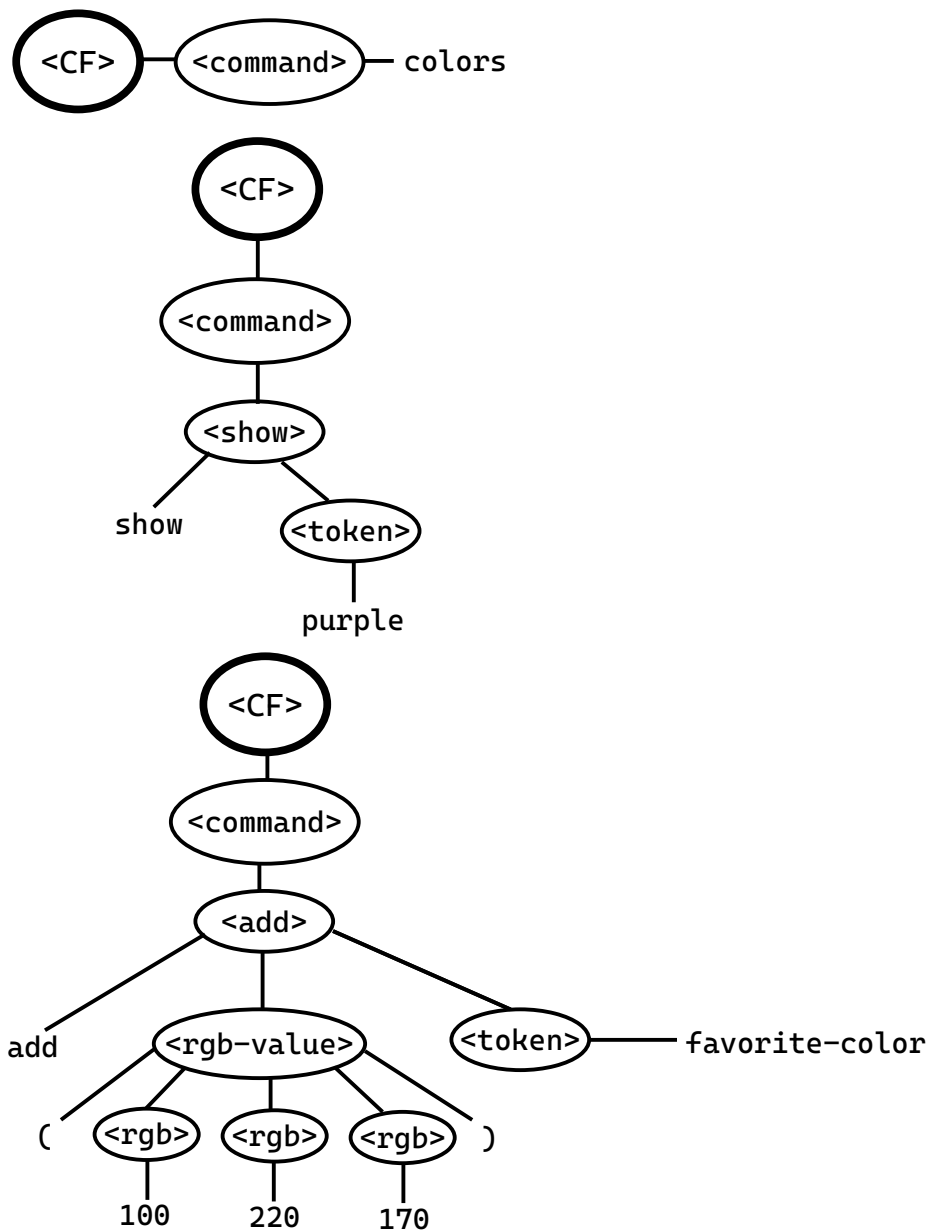
`<command> ::= <add> | colors | <describe> | <show>`

`<add> ::= add <rgb-value> <token> | add color <token>`

`<describe> ::= describe <token>`

`<show> ::= show <token>`

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



Problem #6: BNF?

A BNF (Backus-Normal or Backus-Noir form) is, to put it simply, a syntax for describing syntaxes. It's a way to present the rules of a grammar in a way that's easy to follow and (hopefully) implement. BNFs allow you to easily visualize whether a statement is valid through drawing parse trees to see how one can (or cannot) arrive there via a starting symbol.