CS-331: Assignment 1

Due on 2/1/2024

Prof. Glenn Chappell, Spring 2024, 2/1/2024

Kaleb Burris

A

BE SURE TO DRINK YOUR OVALTINE

B

1. Type checking for C++ is primarily static.
2. This means that type checking is done at compile time instead of runtime.

C

The grammar describes: $[a^n b^m c]$ where $n \geq 1, m \geq 0$.

The strings generated are: 1, 4, 5.

D

The grammar describes all strings that:

- Contains 0 or more x's on the front,
- Contains 0 or more pairs of y's (yy),
- Contains 0 or more z's at the end.

E

The regex is matched by: 3, 4, 5, 7.

F

$$[xyz] * y + [xyz] *$$

G

1. $\underline{S}$

   $\underline{S}S$

   $xy\underline{S}$

   $xyxy$

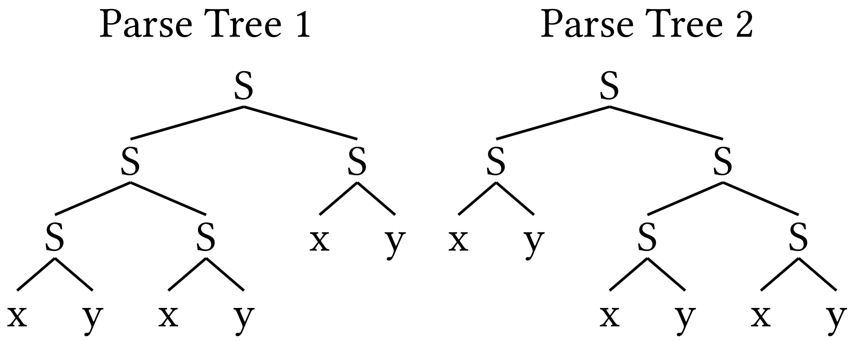
2. $\underline{S}$

   $S\underline{S}$

   $\underline{S}xy$

   $xyxy$

3.



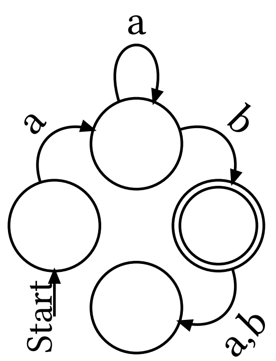
Parse Tree 1

Parse Tree 2

4. $S \to Sxy \mid \varepsilon$

H

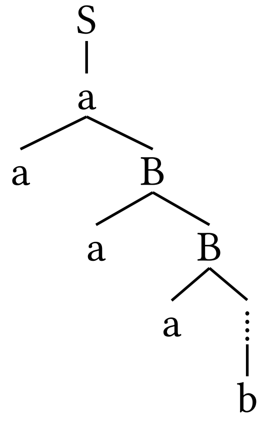
1. $(aa) * b?$
2. This one turned out pretty cool:



3. $S \to aaB$

   $B \to b|aB$

4. My grammar here is not ambiguous as it only has a single, potentially infinite path:



I

I have no faith this works properly.

$$
\begin{aligned}
< \text{reg-exp} > &::= < \text{kleene} > \mid < \text{or} > \mid < \text{char} > \mid \text{"("} < \text{reg-exp} > \text{")"} \\
&\quad \mid < \text{reg-exp} >< \text{reg-exp} > \\
< \text{kleene} > &::= < \text{reg-exp} > \text{"} * \text{"} \\
< \text{or} > &::= < \text{reg-exp} > \text{"|"} < \text{reg-exp} > \\
&\quad \mid \text{"("} < \text{reg-exp} > \text{"|"} < \text{reg-exp} > \text{")"}
\end{aligned}
$$