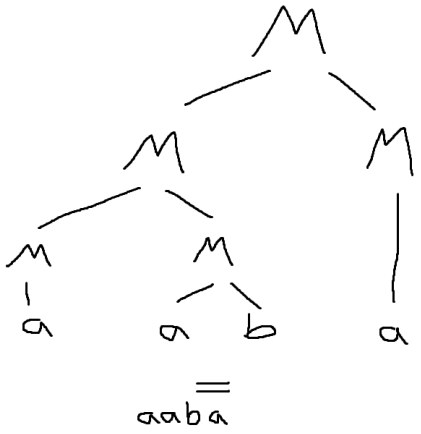
Colton Roach
COSC 4785
September 26, 2022
Exam 1

1.
    a. (11)(1|0)*(000)
    b. S -> 11A000
       A -> 0A | 1A | ∈

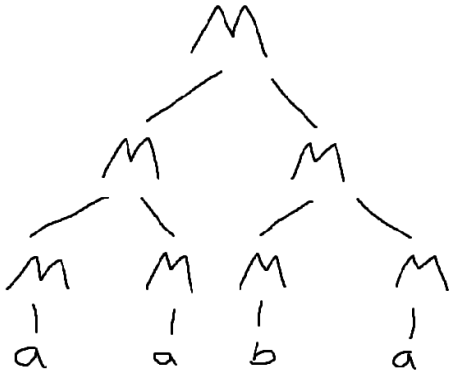
2. L-values are locations, r-values are actual values. This means that when generating intermediate code representations it's important that we assign l-values registers and don't just put the r-values in there. Because the contents of the l-value can change while the r-value cannot.

3. Records for variable names and the attributes for that name like type, storage allocated, scope, number of arguments, etc.

4.
    a.

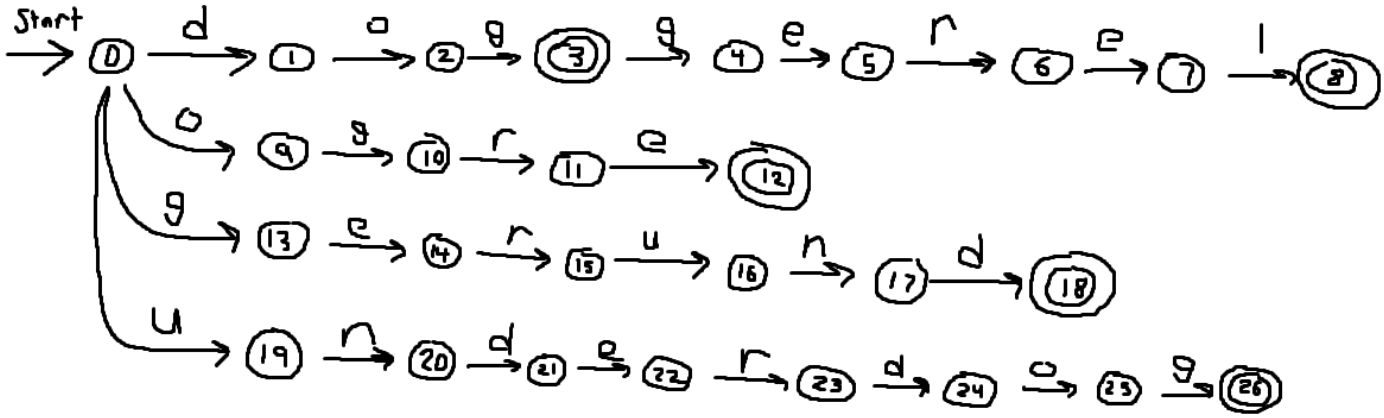


b. Ambiguous, here's another way to get aaba.



5. For block languages we can use the most-closely nested rule. This means that we need to chain our symbol tables together. In other words we will have a program wide symbol table, then if there is a block we create another symbol table attached to the first, then if there's another block inside the first block we create another symbol table attached to the previous one, etc. This creates "environments" in the program that determines the scope of identifiers. To look an entry up, we start in the current environment (symbol table) and if we don't find what we are looking for, we go up a level to the next environment (symbol table).

6.



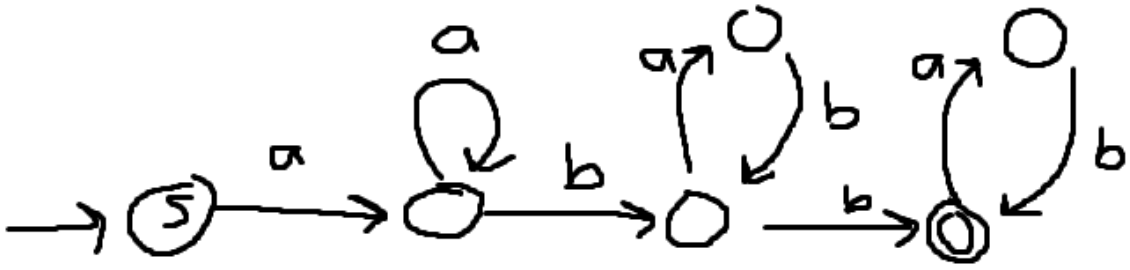
s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
f(s)	0	9	10	13	14	15	0	0	0	13	0	0	0	0	0	19	20	21	0	0	1	0	0	1	2	3

7. The order should be lexical analysis -> syntactic analysis -> semantic analysis
    a. Semantic analysis comes third in the compilation process because it takes information from the syntax tree and the symbol table and makes sure that

all operators' arguments are 'typed' correctly. It gathers the type information and saves it in the symbol table or directly in the syntax tree. It also checks the syntax tree for semantic consistency with the language definition.

b. Syntactic analysis comes second in the compilation process because it takes tokens from the lexical analysis phase and constructs a syntax tree to represent the structure of the token stream. In other words it breaks the token stream down into steps with operator nodes and then argument nodes that are the operator nodes' children
c. Lexical analysis comes first in the compilation process because it reads the stream of characters from the source program and groups them into lexemes. For each lexeme it produces a token that it passes to the syntactic analysis phase. These tokens build up the symbol table.

8. A character class is a part of regular expressions. It is represented by a set of characters, and when Flex is matching patterns it will only match 1 character from the set.
9.



10.

a. It can be a lot easier to break the translation from source code to target code into intermediate code representations than it would be to go straight from one to the other. It allows us to break steps up and analyze each step more easily.
b. Since 3-address code has only one operator on the right side, it makes it very easy to fix the order in which operations have to be done.