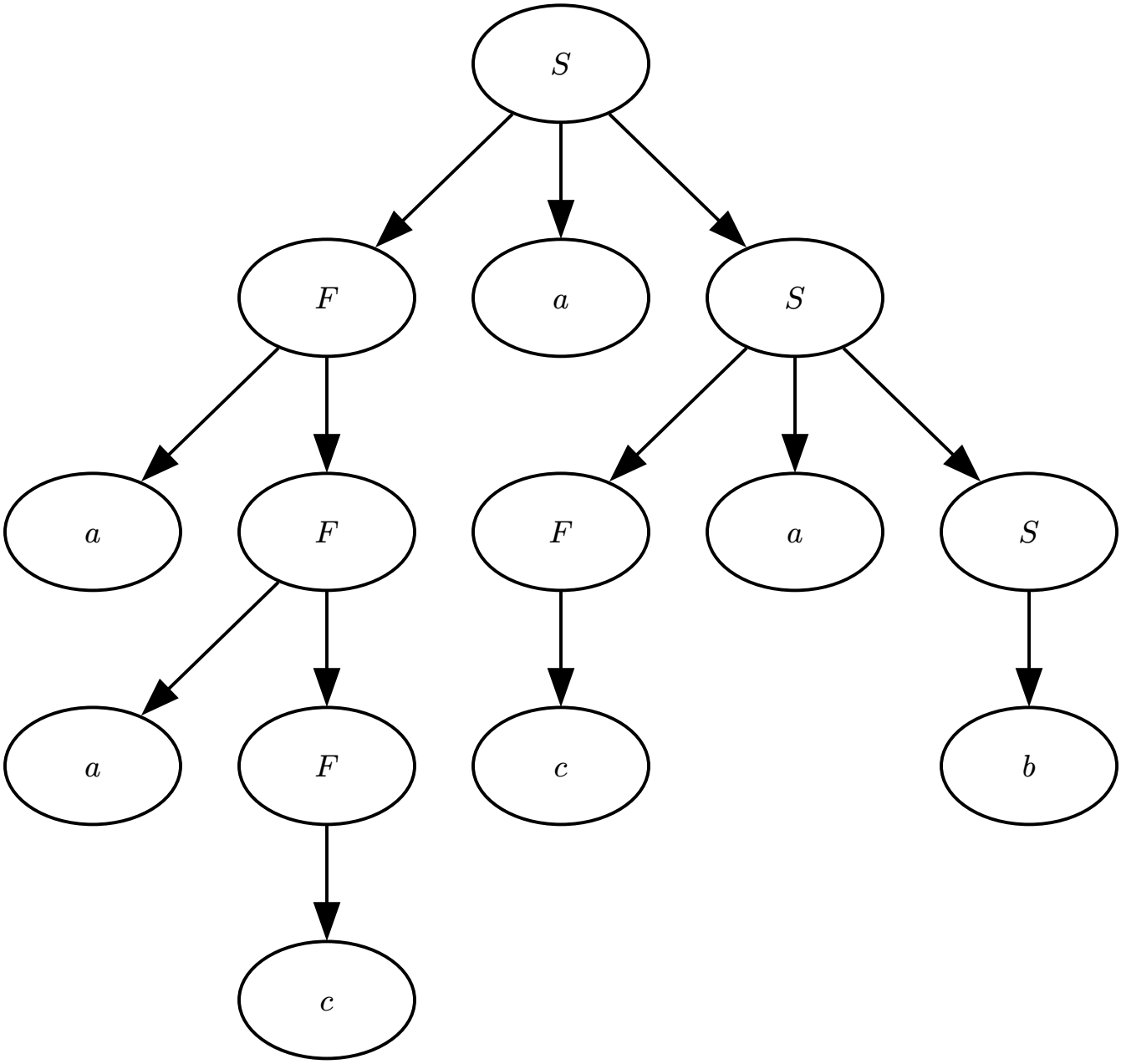
CS 4337 — Programming Language Paradigms, Homework 2

https://iniyanjoseph.com | iniyanijoseph@gmail.com

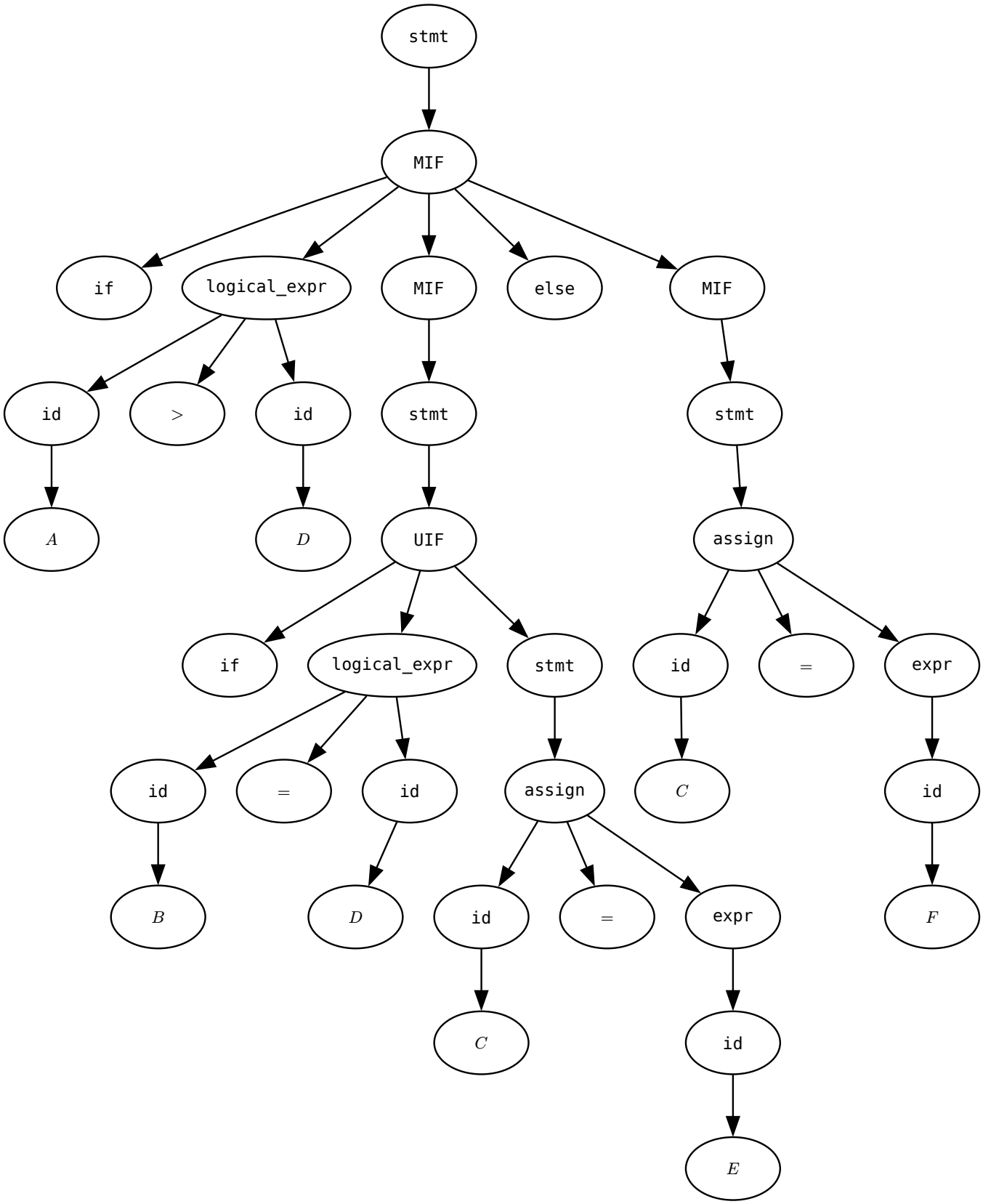
Show a parse tree for string `aacacab` according to the following grammar, where the start symbol is $\langle S \rangle$

$$\langle S \rangle \to \langle F \rangle a \langle S \rangle \mid b$$

$$\langle F \rangle \to a \langle F \rangle \mid c$$



Given the grammar, draw a parse tree for the sentence `if (A > D) if (B = D) C = E else C = F`



Consider the following grammar, which of the following sentences are in the language generated by this grammar? Explain your answer

abcd

Yes. Since $A \to b$ and $B \to d$, S can reach all the terminals without creating other nonterminals

acccbd

No. It is not possible to genrate a "b" after a "d"

acccbcc

Yes. All of the characters can be reached in order

acd

No. It is not possible to generate an empty S, and so the first S is unable to make a string less than 4 characters long.

Rewrite the following BNF to give + precedence over * and force + to be right associative

$$\langle \text{assign} \rangle \to \langle \text{id} \rangle = \langle \text{expr} \rangle$$
$$\langle \text{id} \rangle \to A \mid B \mid C$$
$$\langle \text{expr} \rangle \to \langle \text{expr} \rangle * \langle \text{term} \rangle$$
$$\langle \text{term} \rangle$$
$$\langle \text{term} \rangle \to \langle \text{factor} \rangle + \langle \text{factor} \rangle$$
$$\langle \text{factor} \rangle$$
$$\langle \text{factor} \rangle \to (\langle \text{expr} \rangle)$$
$$\langle \text{id} \rangle$$

Convert the following BNF to EBNF.

$$\langle \text{program} \rangle \to \text{begin } \langle \text{stmt_list} \rangle \text{ end}$$
$$\langle \text{stmt_list} \rangle \to \langle \text{stmt} \rangle \{; \langle \text{stmt} \rangle\}$$
$$\langle \text{stmt} \rangle \to \langle \text{var} \rangle = \langle \text{expression} \rangle\}$$
$$\langle \text{var} \rangle \to A \mid B \mid C$$
$$\langle \text{expression} \rangle \to \langle \text{var} \rangle [(+ \mid -) \langle \text{var} \rangle]$$

Write the EBNF descripiton for the C switch statement

$$\langle \text{switch_stmt} \rangle \to \text{switch } (\langle \text{expr} \rangle) \{ \langle \text{case_list} \rangle \}$$
$$\langle \text{case_list} \rangle \to \langle \text{case} \rangle \{, \langle \text{case} \rangle\} \text{ default: } \langle \text{stmt_list} \rangle$$
$$\langle \text{case} \rangle \to \text{case: } \langle \text{stmt_list} \rangle$$

Rewrite the following EBNF grammar in BNF

$$\langle \text{decl} \rangle \to \langle \text{indent_list} \rangle : \text{int}$$
$$\mid \langle \text{indent_list} \rangle : \text{float}$$
$$\mid \langle \text{indent_list} \rangle : \text{char}$$
$$\langle \text{ident_list} \rangle \to \langle \text{identifier} \rangle, \langle \text{ident_list} \rangle$$

Consider the following BNF grammar

$$\langle S \rangle \rightarrow \langle S \rangle + \langle S \rangle \mid a \mid b \mid c$$

Prove that this grammar is ambiguous.

This grammar is ambiguous. Take the following string: $a + b + c$. It in unclear if the parse tree should take $(a + b) + c$ or $a + (b + c)$

Convert this grammar into an unambiguous grammar.

$$\langle S \rangle \rightarrow \langle \text{term} \rangle S$$
$$\langle \text{term} \rangle \rightarrow a \mid b \mid c$$

Consider the following BNF grammar for a language with two infix operators represented by # and $. Note The order of the rules in a grammar is not significant.

What is the associativity of the # operator?

Right associative.

What is the associativity of the $ operator?

Left associative.

Which operator has higher precedence?

#

The grammar is

Both left and right recursive

Draw a parse tree for the string: $a\#b\#c\$a\$b$

