Context-Free Grammar Practice Problems\*

For each grammar and the given string, draw the parse tree showing how the string may be derived. Whitespace is used for clarity but is not meaningful in these exercises. Non-terminals are written with an initial capital letter.

1. S -> 0 S 1 | 0 1

Generate the string 000111

1. S -> + S S | \* S S | a

Generate the string +\*aaa

1. S -> (S)S | Ɛ

Generate the string ( ( ) ( ) )

1. S -> S + S | S S | (S) | S \* | a

Generate (a + a) \* a

1. S -> a S b S | b S a S | Ɛ

Generate aabbab

1. Id -> Char Rest

Rest -> Symbol Rest | Ɛ

Symbol -> Char | Digit | Underscore

Char -> a | b | c | … | z

Digit -> 0 | 1 | … | 9

Underscore -> \_

Generate:

1. a
2. a1
3. a\_1
4. ax\_1
5. Assume Id is defined as in #6, then add:

Call -> Id ( Optparams )

Optparams -> Params | Ɛ

Params -> Params, Param | Param

Param -> Id

Generate:

1. f()
2. f(x)
3. f(x, y)

\*Most of these are based on examples in Compilers: Principles, Techniques, and Tools by Aho, Lam, Sethi, and Ullman.