

# Software Specifications Context-Free Grammars

Cain Susko

Queen's University  
School of Computing

February 4, 2022

## CF Grammar

A context free grammar is a set of rules for rewriting a language, applied recursively. A CF grammar can generate many nonregular languages and is commonly used to specify the syntax of a programming language.

### Example 1

Given the set of balanced strings:

$$A = \{a^i b^i \mid i \geq 0\}$$

Find a Grammar for  $A$

$$S \rightarrow aSb$$

$$S \rightarrow \epsilon.$$

Where  $S$  is the start variable and all expressions to the right of  $S$  are recursive rules for the given set. The  $S$  in the first rule represents a call of this rule again, hence the recursive property.

Combining these rules, we derive the grammar for  $A$ :

$$S \Rightarrow aSb$$

which operates like so:

$$aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow \dots$$

### Example 2

Given the Expression grammar:

$$\diamond \rightarrow \diamond + \diamond$$

$$\diamond \rightarrow \diamond \times \diamond$$

$$\diamond \rightarrow (\diamond)$$

$$\diamond \rightarrow a.$$

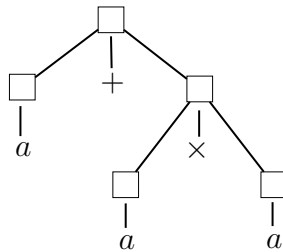
Where  $\diamond = \text{expression}$ .  $\diamond$  is **non-Terminal/Variable**. Note: the Terminal symbols are  $\{+, \times, (, ), a\}$  Thus the derivation of these rules is:

$$\diamond \Rightarrow \diamond + \diamond \Rightarrow \diamond + \diamond \times \diamond \Rightarrow a + \diamond \times \diamond \Rightarrow a + a \times \diamond \Rightarrow a + a \times a$$

Thus, the terminal string is

$$a + a \times a$$

From this derivation and terminal string we can create a Parse Tree:



Note: Grammar is ambiguous such that some terminal strings can have more than one parse tree. For example:

