1 Alphabets, Languages, and Grammars

1.1 Terminology

- An **alphabet** (or **vocabulary**) is just a finite, non-empty set. It's elements are called **letters** or **symbols**.
- A **word** (or **string**) is a finite sequence of symbols.
- A word consisting of no symbols is called the **empty word** and denoted λ. (Think `""`.)
- The set of all words using exactly n symbols from V (repetition allowed) is denoted V^n.
- The set of all words using symbols in alphabet V is denoted V^*. (So $V^* = V^0 \cup V^1 \cup V^2 \cdots$.)
- A **language** over V is a subset of V^*.

Exercises

1. Let $V = \{0, 1\}$. What is V^0? What is V^2? What is V^*?
2. Let $V = \{2\}$. What is V^0? What is V^2? What is V^*?
3. Can $\emptyset$ (empty set) be a letter? an alphabet? a word? a language?

1.2 (Phrase-Structure) Grammars

A phase-structure grammar consists of

- an **alphabet**: denoted V here
- a **start symbol** S: S must be (exactly) one of the symbols in the alphabet. We can denote this as $S \in V$.
- **terminal symbols** (T): a set of symbols in the alphabet ($T \subset V$)
  - $N = V - T$ is the set of **nonterminal symbols**
  - so very symbol is either terminal or nonterminal
  - The start symbol is a terminal in some languages and a nonterminal in others
- **production rules** (P): $P \subseteq (V^* - T^*) \times V^*$
  - usually we write elements of P as $u \to v$ rather than (u, v).
  - u: $V^* - T^*$ says that the lefthand side of the rule (u) must **contain at least one nonterminal**.
  - v: The righthand side can be any combinaton of terminals and nonterminals (including the empty string).

Examples

Note: in each of these examples, capital letters are used for nonterminals and lower case letters or digits for terminals. That makes it easy to remember, but it is not a requirement of the definition of a grammar.

Grammar 1 (G_1): alphabet: $\{a, b, A, B, S\}$, terminals: $\{a, b\}$, start symbol: S, production rules:

- $S \to ABa$
- $A \to BB$
- $B \to ab$
- $AB \to b$

Grammar 2 (G_2): alphabet: $\{S, A, a, b\}$, terminals: $\{a, b\}$, start symbol: S, production rules:

- $S \to aA$
- $S \to b$
- $A \to aa$

Grammar 3 (G_3): alphabet: $\{S, 0, 1\}$, terminals: $\{0, 1\}$, start symbol: S, production rules:

- $S \to 11S$
- $S \to 0$

Grammar 4 (G_4): alphabet: $\{A, B, C, D, a, b, c\}$, terminals: $\{a, b, c\}$, start symbol: A, production rules:

- $A \to BC$
- $B \to Da$
- $C \to Ca$
- $C \to Db$
- $C \to b$
- $D \to cb$
- $D \to b$

1.2.1 Derivations and Languages

The rules of a grammar are used to derive strings of terminals (elements of T^*) as follows.

- If $w_0 \to w_1$ is a rule, then $lw_0r \Rightarrow lw_1r$ for any strings l and r.
- $a \overset{*}{\Rightarrow} b$ is defined recursively. $a \overset{*}{\Rightarrow} b$ if either
  - $a \Rightarrow b$, or
  - there is a c such that $a \Rightarrow c \wedge c \overset{*}{\Rightarrow} b$

Basically the production rules are "rewrite rules" that allow us to replace the left side of the rule with the right side. A derivation is a sequence of rewrites.

The language of a grammar G is denoted $L(G)$ and contains all strings of terminals that can be derived from the start symbol:

$$L(G) = \{w \in T^* \mid S \overset{*}{\Rightarrow} w\}$$

Exercises

4. Show that using Grammar 1, $ABa \overset{*}{\Rightarrow} abababa$. Show every step in the process.
5. Is $abababa \in L(G_1)$, the language generated by Grammar 1?
6. What is $L(G_2)$?
7. What is $L(G_3)$?
8. Generate several words using G_4.
9. Create a grammar G_5 such that $L(G_5) = \{0^n1^n \mid n = 0, 1, 2, ...\}$
10. Create a grammar G_6 such that $L(G_6) = \{0^n1^n \mid n \in \mathbb{Z}^+\}$
11. Create a grammar G_7 such that $L(G_7) = \{0^m1^n \mid m, n \in \mathbb{N}\}$