Summary of Chapter 6

A **context-free grammar** (CFG) consists of a set of **productions** which allow one to replace a **variable** by a string of variables and **terminals**.

The **language** of a grammar is the set of strings it generates.

A language is **context-free** if there is a CFG for it.

Each string in the language has a **leftmost derivation** and a **derivation tree**.

If these are unique for all strings, then the grammar is called **unambiguous**.

Summary of Chapter 7

A **pushdown automaton** (PDA) is an FA with a **stack** added for storage.

We choose to draw these as flowcharts where the character Δ indicates both empty stack and end-of-input.

A PDA is nondeterministic by definition.

Summary of Chapter 8

There is an algorithm to convert a CFG to an equivalent PDA: the PDA guesses the leftmost derivation.

The algorithm to convert from PDA to CFG is more complex.

Summary of Chapter 9

A **regular grammar** is one where every production has the form $A \to \texttt{b}C$ or $A \to \texttt{a}$.

The **Chomsky hierarchy** also includes context-sensitive grammars and unrestricted grammars.

There are special forms for CFGs such as **Chomsky Normal Form**, where every production has the form $A \to BC$ or $A \to \mathsf{c}$.

The algorithm to convert to Chomsky Normal Form is
(1) determine all nullable variables and get rid of all ε-productions,
(2) get rid of all variable unit productions,
(3) break up long productions, and
(4) move terminals to unit productions.

Context-free languages are closed under the Kleene operations.

They are not closed under intersection or complementation.

The **Pumping Lemma** can be used to prove that a language is not context-free.