The Church-Turing Thesis (Section 12.2)

The Church-Turing Thesis

- The Church-Turing Thesis: Anything that is intuitively computable can be computed by a Turing machine.
- It is a *thesis* rather than a theorem because it relates the informal notion of intuitively computable to the formal notion of a Turing machine.

Computational Models

- A *computational model* is a characterization of a computing process that describes the form of a program and describes how the instructions are executed.
- *Example:* The Turing machine computational model describes the form of TM instructions and how to execute them.
- *Example:* If X is a programming language, the X computational model describes the form of a program and how each instruction is executed.

Equivalence of Computational Models

- Two computational models are *equivalent* in power if they solve the same class of problems.
- Any piece of data for a program can be represented by a string of symbols and any string of symbols can be represented by a natural number.
- So, even though computational models may process different kinds of data, they can still be compared with respect to how they process natural numbers.
- We will briefly discuss a few models of computation that are equal in power to the TM model.

The Simple Programming Language

- This imperative programming model processes natural numbers. The language is defined as follows:
  - Variables have type $\mathbb{N}$.
  - Assignment statements $X := 0; X := \mathrm{succ}(Y); X := \mathrm{pred}(Y);$ (assume pred(0) $=$ 0)
  - Composition of statements: $S_1; S_2;$
  - **while** $X \neq 0$ **do** S **od**
- This simple language has the same power as the Turing machine model.
- For input and output use the values of the variables before and after execution.

Demonstrate power of language

Define macros to demonstrate the power of the language:

$$
\begin{array}{l|l}
X := Y & X := \mathrm{succ}(Y); X := \mathrm{pred}(X). \\
X := 2 & X := 0; X := \mathrm{succ}(X); X := \mathrm{succ}(X). \\
Z := X + Y & Z := X; C := Y; \textbf{while } C \neq 0 \textbf{ do} \\
 & Z := \mathrm{succ}(Z); C := \mathrm{pred}(C) \textbf{ od}
\end{array}
$$

Other models

- Other models that are discussed in Section 12.2 include:
  - Partial recursive functions
  - Markov algorithms
  - Post algorithms
  - Post Production systems
- We're not going to discuss these models.