Generally useful information.

- The notation **z = <x,y>** denotes the pairing function with inverses **x = <z>**$_1$ and **y = <z>**$_2$.
- The minimization notation **μ y [P(…,y)]** means the least **y** (starting at **0**) such that **P(…,y)** is true. The bounded minimization (acceptable in primitive recursive functions) notation **μ y (u≤y≤v) [P(…,y)]** means the least **y** (starting at **u** and ending at **v**) such that **P(…,y)** is true. Unlike the text, I find it convenient to define **μ y (u≤y≤v) [P(…,y)]** to be **v+1**, when no **y** satisfies this bounded minimization.
- The tilde symbol, **~,** means the complement. Thus, set **~S** is the set complement of set **S**, and predicate **~P(x)** is the logical complement of predicate **P(x).**
- A function **P** is a predicate if it is a logical function that returns either **1** (**true**) or **0** (**false**). Thus, **P(x)** means **P** evaluates to **true** on **x**, but we can also take advantage of the fact that **true** is **1** and **false** is **0** in formulas like **y** × **P(x)**, which would evaluate to either **y** (if **P(x)**) or **0** (if **~P(x)**).
- A set **S** is recursive if **S** has a total recursive characteristic function χ_S, such that **x** ∈ **S** ⇔ χ_S**(x)**. Note χ_S is a predicate. Thus, it evaluates to **0** (**false**), if **x** ∉ **S**.
- When I say a set **S** is re, unless I explicitly say otherwise, you may assume any of the following equivalent characterizations:
  1. **S** is either empty or the range of a total recursive function f_S.
  2. **S** is the domain of a partial recursive function g_S.
- If I say a function **g** is partially computable, then there is an index **g** (I know that's overloading, but that's okay as long as we understand each other), such that $\Phi_g(x) = \Phi(x, g) = g(x)$. Here Φ is a universal partially recursive function.
  Moreover, there is a primitive recursive function **STP**, such that
  STP(x, g, t) is **1** (true), just in case **g**, started on **x**, halts in **t** or fewer steps.
  STP(x, g, t) is **0** (false), otherwise.
  Finally, there is another primitive recursive function **VALUE**, such that
  VALUE(x, g, t) is **g(x)**, whenever **STP(x, g, t)**.
  VALUE(x, g, t) is defined but meaningless if **~STP(x, g, t)**.
- The notation **f(x)**↓ means that **f** converges when computing with input **x**, but we don't care about the value produced. In effect, this just means that **x** is in the domain of **f**.
- The notation **f(x)**↑ means **f** diverges when computing with input **x**. In effect, this just means that **x** is **not** in the domain of **f**.
- The **Halting Problem** for any effective computational system is the problem to determine of an arbitrary effective procedure **f** and input **x**, whether or not **f(x)**↓. The set of all such pairs is a classic re non-recursive one.
- The **Uniform Halting Problem** is the problem to determine of an arbitrary effective procedure **f**, whether or not **f** is an algorithm (halts on all input). The set of all such function indices is a classic non re one.