We now turn to techniques that will allow us to prove that some sets are not recursive or r.e. The first one is also used to prove that the real numbers are uncountable...

**Def.** A set $A$ is countable (countably infinite) if it can be put into 1-to-1 correspondence with $N$.

**Theorem.** The set $F = \{ f | f : N \to \{0, 1\} \}$ is uncountable.

**Proof.** Assume it is countable. Then $F = \{ f_0, f_1, f_2, \ldots \}$. We now define a new function $f : N \to \{0, 1\}$ as $f(n) = 1 \cdot f_n(n) (= \text{minus}(1, f(n)))$. Then $f \in F$ and $f \neq f_n$ for every $n \geq 0$. Contradiction.

**Def.** A set $A$ is co-r.e. if its complement $\overline{A}$ is r.e.

**Theorem.** There exists a set $A$ which is co-r.e. but not r.e.

**Proof.** Recall that $\{ W_n | n \geq 0 \}$ is an (effective) enumeration of all the r.e. sets. We need to construct a set $A$ which is different from all of them - and thus not r.e. Let $\Lambda = \{ n | n \notin W_n \}$. Since $A \neq W_n$ for all $n \geq 0$, $A$ cannot be r.e. But $\overline{A} = \{ n | n \in W_n \} = \{ n | (\exists t) \text{halt}(n, n, t) \}$, and so $\overline{A}$ is r.e.

**Corollary 5.20.** a) $K = \{ n | n \in W_n \}$ is r.e. but not recursive. b) $K_0 = \{ \langle y, x \rangle | M_x \text{ halts on input } y \}$ is r.e. but not recursive.

**Proof.** a) $\overline{K}$ is not r.e. $\Rightarrow K$ is not recursive (Thm. 5.15: $K \Leftrightarrow K$ and $\overline{K}$ r.e.). b) $K_0$ r. $\Rightarrow K$ r. By $\chi_{K_0}(x) = \chi_{\overline{K}_0}(x, x)$. Contradiction.

**Def.:** A decision problem is a problem whose answer (output) is yes or no. Every decision problem corresponds to a language consisting of all inputs for which the answer is yes.

**Def.:** a decision problem is undecidable (unsolvable) if the corresponding language is not recursive. It is decidable, if the corresponding language is recursive. If the language is r.e., the problem is usually called provable, by some authors solvable, or partially decidable.

**Theorem.** The problem of determining whether a given Turing Machine $M_x$ halts on a given input string $y$ (= the halting problem) is undecidable.

**Proof.** By Corollary 5.20 b, $K_0 = \{ \langle y, x \rangle | M_x \text{ halts on input } y \}$ is r.e. but not recursive.

A perfectly legitimate question would be: if we have a partial recursive function $f : \{0, 1\}^* \to \{0, 1\}^*$, can we extend it (i.e., define it over the elements of the domain where it is not defined) so that we end up with a recursive function? We could arbitrarily define it to be 0, couldn’t we? The answer turns out to be a maybe… but not in general. The re-definition cannot, in general, be carried out via a Turing machine…
**Theorem.** There exists a partial recursive function \( f : \{0, 1\}^* \to \{0, 1\}^* \) which is not extendable - there is no recursive function \( g \) s.t. \( g(x) = f(x) \) \( \forall x. \)

**Proof.** If \( f \) is extendable, \( f = \phi_n \) for some \( n \), over the domain where \( f \) is defined, and \( \phi_n \) is total. Diagonalization suggests: construct a partial recursive function \( f \), which is different from every (total) recursive function \( \phi_n \). Then it cannot be total (since it is different, at points where it is defined, from every total function), and it cannot be extended to be total (since the extension would then have to be the same as some total function, while differing from every total function). For each \( n \), define \( f(n) = \text{minus}(1, \phi_n(n)) = \text{minus}(1, \Phi_1(n, g(n))) \). \( f \) is partial recursive since \( \Phi_1(x, y) \) is the two argument function computed by the Universal TM and \( \text{minus} \) and 1 are primitive recursive. Furthermore \( f(n) \downarrow \) for all \( n \) such that \( \phi_n \) is recursive, since \( \phi_n(x) \) is then total (in \( x \) and \( \phi_n(n) \)) . If it were extendable, its extension would be a total function \( \phi_n \). But \( f(n) = \text{minus}(1, \phi_n(n)) \neq \phi_n(n) \) \( \forall n \). Contradiction.

We can show that the set of recursive functions - a subset of \( \{ \phi_n \mid n \geq 0 \} \) - is not recursive, in fact, not even recursively enumerable.

**Ex. 5.22.** \( \text{Tor} = \{ n \mid W_n = \{0, 1\}^* \} = \{ n \mid \phi_n \text{ is recursive} \} \) is not r.e.

**Pf.** Assume \( \text{Tor} \) is r.e. **Thm 5.13** \( \exists g, g \text{ recursive}, \text{range}(g) = \text{Tor} \). Thus, for all \( n \geq 0 \), \( \phi_{g(n)} \) is recursive and we have the correspondence \( n \leftrightarrow \phi_{g(n)} \). We now construct a recursive function \( f \neq \phi_{g(n)} \) for all \( n \geq 0 \).

Define \( f(n) = \text{minus}(1, \phi_{g(n)}(n)) \). Since \( g(n) \in \text{Tor} \) \( \forall n \geq 0 \), \( \phi_{g(n)}(n) \) is always defined, and \( f \) is a total function. Since, by definition, \( f(n) = \text{minus}(1, \phi_{g(n)}(n)) \), \( f \) is recursive. The contradiction arises from the fact that \( f(n) \neq \phi_{g(n)}(n) \) \( \forall n \), and this implies that \( f \neq \phi_{g(n)} \) \( \forall n \). We have just constructed an "extra" recursive function...