Recursively Enumerable Sets

Definition: We write:

\[ W_n = \{ x \in \mathbb{N} \mid \Phi(x, n) \downarrow \}. \]

Theorem 4.6 (Enumeration Theorem):
A set \( B \) is r.e. if and only if there is an \( n \) for which \( B = W_n \).
This is an immediate consequence of the definition of \( \Phi(x, n) \).
The theorem gets its name from the fact that the sequence \( W_0, W_1, W_2, \ldots \) is an enumeration of all r.e. sets.

Theorem 4.7:

\[ K = \{ n \in \mathbb{N} \mid n \in W_n \}. \]

Then
\[ n \in W_n \iff \Phi(n, n) \downarrow \iff \text{HALT}(n, n). \]

\( K \) is the set of all numbers \( n \) such that program number \( n \) eventually halts on input \( n \).

Recursively Enumerable Sets

Proof:
Since \( K = \{ n \in \mathbb{N} \mid \Phi(n, n) \downarrow \} \), and by the universality theorem (Theorem 3.1), \( \Phi(n, n) \) is partially computable, \( K \) is obviously r.e.

If \( K \) were recursive, then \( \neg K \) would be r.e.
If that were the case, then by the enumeration theorem there would have to be some number \( i \) so that \( \neg K = W_i \).

But then:
\[ \begin{align*}
i \in K & \iff i \in W_i \iff i \in \neg K. \\
i \in \neg K & \iff i \in W_i \iff i \in K. \end{align*} \]

Contradiction!

Recursively Enumerable Sets

Theorem 4.8:

Let \( B \) be an r.e. set. Then there is a primitive recursive predicate \( R(x, t) \) such that \( B = \{ x \in \mathbb{N} \mid (\exists t) R(x, t) \} \).

Proof:

Let \( B = W_n \). Then
\[ B = \{ x \in \mathbb{N} \mid (\exists t) \text{STP}^{(1)}(x, n, t) \} , \]
and \( \text{STP}^{(1)} \) is primitive recursive by Theorem 3.2.

Recursively Enumerable Sets

Theorem 4.9:

Let \( S \) be a nonempty r.e. set. Then there is a primitive recursive function \( f(u) \) such that \( S = \{ f(n) \mid n \in \mathbb{N} \} \).
In other words, \( S \) is the range of \( f \).

Theorem 4.10:

Let \( f(x) \) be a partially computable function, and let \( S = \{ f(x) \mid f(x) \downarrow \} \) so \( S \) is the range of \( f \).
Then \( S \) is r.e.

If we combine Theorems 4.9 and 4.10, we get:

Recursively Enumerable Sets

Theorem 4.11:

Consider a set \( S \neq \emptyset \). The following statements are all equivalent:
1. \( S \) is r.e.;
2. \( S \) is the range of a primitive recursive function;
3. \( S \) is the range of a recursive function;
4. \( S \) is the range of a partial recursive function.

This theorem motivates the term recursively enumerable.
A nonempty r.e. set is enumerated by a recursive function.
Proof Practice

Prove or disprove:
The empty set \( \emptyset \) is recursive.
Proof that it is recursive:
In order to prove that a set \( B \) is recursive, we have to show that there is a computable predicate \( P_B \) such that:
\[
B = \{ x \in \mathbb{N} \mid P_B(x) \}
\]
For the empty set, \( P_\emptyset \) has to be false for every \( x \).
\( P_\emptyset \) is computable, because it is computed, for instance, by the empty program.

Proof Practice

Prove or disprove:
The set of all natural numbers \( \mathbb{N} \) is recursive.
Proof that it is recursive:
In order to prove that a set \( B \) is recursive, we have to show that there is a computable predicate \( P_B \) such that:
\[
B = \{ x \in \mathbb{N} \mid P_B(x) \}
\]
For the set \( \mathbb{N} \), \( P_\mathbb{N} \) has to be true for every \( x \).
\( P_\mathbb{N} \) is computable, because it is computed, for instance, by the following program:
\[
Y \leftarrow Y + 1
\]

Proof Practice

Prove or disprove:
If \( A \) and \( B \) are recursive sets, then \( A \cap B \) is also a recursive set.
Proof that \( A \cap B \) is a recursive set:
If \( A \) and \( B \) are recursive, then there must be computable predicates \( P_A \) and \( P_B \) such that:
\[
A = \{ x \in \mathbb{N} \mid P_A(x) \}
\]
\[
B = \{ x \in \mathbb{N} \mid P_B(x) \}
\]
Then we need to show that there is a computable predicate \( P_{A \cap B} \) such that:
\[
A \cap B = \{ x \in \mathbb{N} \mid P_{A \cap B}(x) \}
\]

Proof Practice

The following program computes \( P_{A \cap B} \):
\[
\begin{align*}
\text{IF } \neg P_A(X) & \text{ GOTO E} \\
\text{IF } \neg P_B(X) & \text{ GOTO E} \\
Y & \leftarrow Y + 1
\end{align*}
\]
Or this one:
\[
Y \leftarrow P_A(X) \cdot P_B(X)
\]

Proof Practice

Prove or disprove:
The set \( B \) of all natural numbers whose associated programs halt on inputs 4 and 7 is r.e.
Proof that it is r.e.:
In order to prove that \( B \) is r.e., we have to show that there is a partially computable function \( f_B(x) \) such that:
\[
B = \{ x \in \mathbb{N} \mid f_B(x) \downarrow \}
\]
The following program computes \( f_B(x) \):
\[
\begin{align*}
Z & \leftarrow \nu_X(4, X) \\
Z & \leftarrow \nu_X(7, X)
\end{align*}
\]

Proof Practice

Prove or disprove:
The set \( B \) of all natural numbers whose associated programs do not halt on input 5 is r.e.
Proof that it is not r.e.:
Assume that \( B \) is r.e. This would require that we can compute for a given program whether it halts on a given input (here: input 5).
In that case, the predicate HALT\((x, y)\) would be computable. However, we already proved that HALT\((x, y)\) is not computable.
This contradiction shows that \( B \) cannot be r.e.
The Parameter Theorem

The parameter theorem is also called iteration theorem and s-m-n theorem.

It is important to the theory of computation as it relates the functions $\phi^{(n)}(x_1, ..., x_n, y)$ for different values of $n$.

**Theorem 5.1 (Parameter Theorem):**
For each $n, m > 0$ there is a primitive recursive function $S_m^n(u_1, ..., u_n, y)$ such that

$$\phi^{(m+n)}(x_1, ..., x_m, y) = \phi^{(m)}(x_1, ..., x_m, S_m^n(u_1, ..., u_n, y)).$$

**Proof:**
Suppose that the values for $u_1, ..., u_n, y$ are fixed. Then the left side of the equation is a partially computable function of the $m$ arguments $x_1, ..., x_m$.

Let the number of the program that computes this function be $q$. Then we have:

$$\phi^{(m+n)}(x_1, ..., x_m, u_1, ..., u_n, y) = \phi^{(m)}(x_1, ..., x_m, q).$$

The parameter theorem tells us that there exists such a $q$ that can be obtained from $u_1, ..., u_n, y$ by a primitive recursive function.

---

The Recursion Theorem

**Theorem 6.1 (Recursion Theorem):**
Let $g(z, x_1, ..., x_m)$ be a partially computable function of $(m + 1)$ variables. Then there is a number $e$ such that

$$\phi^e(m)(x_1, ..., x_m) = g(e, x_1, ..., x_m).$$

**Proof:** Consider the partially computable function $g(S_m^{m+1}(z, v), x_1, ..., x_m)$ where $S_m^{m+1}$ is the function that occurs in the parameter theorem.

Clearly, there must be a program that takes inputs $x_1, ..., x_m$ and $v$ and computes function $g$. Let the number of such a program be $z_0$. Then we have:

$$g(S_m^{m+1}(v, x_1, ..., x_m)) = \phi^{(m+1)}(x_1, ..., x_m, v, z_0).$$

---

**Corollary 6.2:**
There is a number $e$ such that for all $x$:

$$\phi^e(x) = e.$$ 

**Proof:**
Consider the computable projection function $g(z, x) = u_1^2(z, x) = z$.

With the help of the Recursion Theorem, we can find a number $e$ such that

$$\phi^e(x) = g(e, x) = e.$$ 

Such programs generate copies of themselves.