Recursively Enumerable Sets

Thus, saying that a set \( B \) is computable or recursive is the same as saying that \( P(x_1, \ldots, x_n) \) is a computable function.

Likewise, \( B \) is a primitive recursive set if \( P(x_1, \ldots, x_n) \) is a primitive recursive predicate.

It follows that:

**Theorem 4.1:** Let the sets \( B, C \) belong to some PRC class \( \mathcal{C} \). Then so do the sets \( B \cup C, B \cap C, \neg B \).

**Proof:** This is an immediate consequence of Theorem 5.1, Chapter 3.

---

Recursively Enumerable Sets

As long as the Gödel numbering functions \([x_1, \ldots, x_m]\) and \((x)_i\) are available, we only need to consider subsets of \( \mathbb{N} \) instead of subsets of \( \mathbb{N}^m \).

Then we have:

**Theorem 4.2:** Let \( \mathcal{C} \) be a PRC class, and let \( B \) be a subset of \( \mathbb{N}^m, m \geq 1 \). Then \( B \) belongs to \( \mathcal{C} \) if and only if \( B' = \{ [x_1, \ldots, x_m] \in \mathbb{N} | (x_1, \ldots, x_m) \in B \} \) belongs to \( \mathcal{C} \).

---

Recursively Enumerable Sets

**Definition:** The set \( B \subseteq \mathbb{N} \) is called *recursively enumerable* if there is a partially computable function \( g(x) \) such that \( B = \{ x \in \mathbb{N} | g(x) \downarrow \} \).

The term recursively enumerable is abbreviated \( r.e. \).

A set is \( r.e. \) just when it is the domain of a partially computable function.

If \( \varphi \) is a program that computes the function \( g \) (see above), then \( B \) is simply the set of all inputs to \( \varphi \) for which \( \varphi \) eventually halts.

---

Recursively Enumerable Sets

**Example:** Let us consider the case \( B \cup C \).

There must be predicates \( P_B \) and \( P_C \) such that:

\[
B = \{ x \in \mathbb{N} | P_B(x) \} \\
C = \{ x \in \mathbb{N} | P_C(x) \}
\]

Then:

\[
B \cup C = \{ x \in \mathbb{N} | P_B(x) \lor P_C(x) \}
\]

Since \( \lor \) is a primitive-recursive function, the predicate \( P_B(x) \lor P_C(x) \) is also in class \( \mathcal{C} \). Then the equation above shows that \( B \cup C \) is in \( \mathcal{C} \) as well.

---

Recursively Enumerable Sets

**Proof:** If \( P_B(x_1, \ldots, x_m) \) is the characteristic function of \( B \), then

\[
P_B(x) \iff P_B(x_1, \ldots, x_m) \land Lk(x) \leq m \land x > 0
\]

is the characteristic function of \( B' \), and \( P_B \) clearly belongs to \( \mathcal{C} \) if \( P_B \) belongs to \( \mathcal{C} \).

On the other hand, if \( P_B(x) \) is the characteristic function of \( B' \), then

\[
P_B(x_1, \ldots, x_m) \iff P_B([x_1, \ldots, x_m])
\]

is the characteristic function of \( B \), and \( P_B \) clearly belongs to \( \mathcal{C} \) if \( P_B \) belongs to \( \mathcal{C} \).

For example, \( \{ [x, y] \in \mathbb{N} | HALT(x, y) \} \) is not a computable set.

---

Recursively Enumerable Sets

If we think of \( \varphi \) as providing an algorithm for testing for membership in \( B \), we see that

- if a number belongs to \( B \), the algorithm will provide a positive answer,
- if a number does not belong to \( B \), the algorithm will never terminate.

Such algorithms are called *semi-decision procedures*.

They can be considered an “approximation” to solving the problem of testing membership in \( B \).
Recursively Enumerable Sets

**Theorem 4.3:** If B is a recursive set, then B is r.e.

**Proof:** Consider the following program $P:

[A] IF $ \neg (X \in B)$ GOTO A

Since B is recursive, the predicate $X \in B$ is computable and $P$ can be expanded to a program of $L$.

Let $P$ compute the function $h(x)$. Then, clearly, $B = \{x \in N | h(x) \downarrow \}$.

---

Recursively Enumerable Sets

**Theorem 4.4:** The set B is recursive if and only if B and $\neg B$ are both r.e.

**Proof:** If B is recursive, then by Theorem 4.1 so is $\neg B$, and hence by Theorem 4.3, they are both r.e. Conversely, if B and $\neg B$ are both r.e., we may write

$B = \{x \in N | g(x) \downarrow \}

\neg B = \{x \in N | h(x) \downarrow \}$,

where g and h are both partially computable.

---

Recursively Enumerable Sets

**Definition:** We write:

$W_n = \{x \in N | \Phi(x, n) \downarrow \}$.

**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.

---

Recursively Enumerable Sets

We further define:

$K = \{n \in N | n \in W_n\}$.

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 $n$.

**Theorem 4.7:**

$K$ is r.e. but not recursive.
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 $\text{¬}K$ would be r.e.
If that were the case, then by the enumeration theorem there would have to be some number $i$ so that $\text{¬}K = W_i$.
But then:
\[ i \in K \iff i \notin K \iff i \in W_i \iff i \in \text{¬}K. \]
Contradiction!

Or:
\[ i \in K \iff i \notin K \iff i \in W_i \iff i \in \text{¬}K. \]
Contradiction!

There are alternative ways of describing r.e. 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
\[ B = \{x \in \mathbb{N} \mid \exists t \ \text{STP}(x, n, t)\}, \]
and $\text{STP}(1)$ is primitive recursive by Theorem 3.2.

**Theorem 4.9:**
Let $S$ be a nonempty r.e. set. Then there is a primitive recursive function $f(u)$ such that $S = \{f(n) \mid n \in \mathbb{N}\} = \{f(0), f(1), f(2), \ldots\}$. 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:

**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.