Recursive and Recursively Enumerable Sets

Xiaofeng Gao

Department of Computer Science and Engineering
Shanghai Jiao Tong University, P.R.China

CS363-Computability Theory

* Special thanks is given to Prof. Yuxi Fu for sharing his teaching materials.
Outline

1. Recursive Sets
   - Decidable Predicate
   - Reduction
   - Rice Theorem

2. Recursively Enumerable Set
   - Partial Decidable Predicates
   - Theorems

3. Special Sets
   - Productive Sets
   - Creative Set
   - Simple Sets
Recursive Sets
- Decidable Predicate
- Reduction
- Rice Theorem

Recursively Enumerable Set
- Partial Decidable Predicates
- Theorems

Special Sets
- Productive Sets
- Creative Set
- Simple Sets
The following emphasizes the importance of the subsets of $\mathbb{N}$:

$$\text{Decision Problems} \iff \text{Predicates on Number} \iff \text{Sets of Numbers}$$

A central theme of recursion theory is to look for sensible classification of number sets.

Classification is often done with the help of reduction.
Recursive Set

Let $A$ be a subset of $\mathbb{N}$. The characteristic function of $A$ is given by

$$c_A(x) = \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{if } x \notin A. \end{cases}$$

$A$ is recursive if $c_A(x)$ is computable.
A recursive set is (the domain of) a **solvable** problem.

It is important to know if a problem is solvable.
Examples

The following sets are recursive.

(a) $\mathbb{N}$.

(b) $\mathbb{E}$ (the even numbers).

(c) Any finite set.

(d) The set of prime numbers.
Unsolvable Problem

Here are some important unsolvable problems:

\[
K = \{x \mid x \in W_x\},
\]
\[
Fin = \{x \mid W_x \text{ is finite}\},
\]
\[
Inf = \{x \mid W_x \text{ is infinite}\},
\]
\[
Cof = \{x \mid W_x \text{ is cofinite}\},
\]
\[
Rec = \{x \mid W_x \text{ is recursive}\},
\]
\[
Tot = \{x \mid \phi_x \text{ is total}\},
\]
\[
Ext = \{x \mid \phi_x \text{ is extensible to a total recursive function}\}.
\]
Cofinite

\[ \text{Cof} = \{ x \mid W_x \text{ is cofinite} \} \text{ means the set whose complement is finite.} \]
Cofinite

\[ \text{Cof} = \{ x \mid W_x \text{ is cofinite} \} \text{ means the set whose complement is finite.} \]

**Example 1:** \( \{ x \mid x \geq 5 \} \) is cofinite.
Cofinite

\[ Cof = \{ x \mid W_x \text{ is cofinite} \} \] means the set whose complement is finite.

Example 1: \( \{ x \mid x \geq 5 \} \) is cofinite.

Not every infinite set is cofinite.
Cofinite

\[ Cof = \{ x \mid W_x \text{ is cofinite} \} \text{ means the set whose complement is finite.} \]

**Example 1:** \( \{ x \mid x \geq 5 \} \) is cofinite.

Not every infinite set is cofinite.

**Example 2:** \( \mathbb{E}, \emptyset \) are not cofinite.
Extensible Functions

\[ Ext = \{ x \mid \phi_x \text{ is extensible to a total recursive function} \}. \]
Extensible Functions

\[ Ext = \{ x \mid \phi_x \text{ is extensible to a total recursive function} \}. \]

**Example:** \( f(x) = \phi_x(x) + 1 \) is not extensible.
Extensible Functions

Ext = \{ x \mid \phi_x \text{ is extensible to a total recursive function} \}.

Example: \( f(x) = \phi_x(x) + 1 \) is not extensible.

Proof: Assume \( f(x) \) is extensible, then define total recursive function

\[
g(x) = \begin{cases} 
\psi_U(x,x) + 1 & \text{if } \psi_U(x,x) \text{ is defined.} \\
\star & \text{otherwise}
\end{cases}
\]

Let \( \phi_m \) be the Gödel coding of \( g(x) \), then \( \phi_m \) is a total recursive function.

When \( x = m \), \( \phi_m(m) = \psi_U(m,m) \) by universal problem.

However, \( \phi_m(m) = g(m) = \psi_U(m,m) + 1 \) by equation (1). A contradiction.

\[ \square \]
Extensible Functions

Ext = \{x \mid \phi_x \text{ is extensible to a total recursive function}\}.

**Example:** \(f(x) = \phi_x(x) + 1\) is not extensible.

**Proof:** Assume \(f(x)\) is extensible, then define total recursive function

\[
g(x) = \begin{cases} 
\psi_U(x, x) + 1 & \text{if } \psi_U(x, x) \text{ is defined.} \\
\mathbf{x} & \text{otherwise}
\end{cases} \tag{1}
\]

Let \(\phi_m\) be the Gödel coding of \(g(x)\), then \(\phi_m\) is a total recursive function.

When \(x = m\), \(\phi_m(m) = \psi_U(m, m)\) by universal problem.

However, \(\phi_m(m) = g(m) = \psi_U(m, m) + 1\) by equation (1). A contradiction.

**Comment:** Not every partial recursive function can be obtained by restricting a total recursive function.
A predicate $M(x)$ is **decidable** if its characteristic function $c_M(x)$ given by

$$c_M(x) = \begin{cases} 
1, & \text{if } M(x) \text{ holds,} \\
0, & \text{if } M(x) \text{ does not hold.}
\end{cases}$$

is computable.

The predicate $M(x)$ is **undecidable** if it is not decidable.

Recursive Set $\iff$ Solvable Problem $\iff$ Decidable Predicate.
Theorem. If $A$, $B$ are recursive sets, then so are the sets $\overline{A}$, $A \cap B$, $A \cup B$, and $A \setminus B$. 
Algebra of Decidability

**Theorem.** If $A$, $B$ are recursive sets, then so are the sets $\overline{A}$, $A \cap B$, $A \cup B$, and $A \setminus B$.

**Proof.**

\[
\overline{c_A} = 1 - c_A.
\]

\[
c_{A \cap B} = c_A \cdot c_B.
\]

\[
c_{A \cup B} = \max(c_A, c_B).
\]

\[
c_{A \setminus B} = c_A \cdot c_{\overline{B}}.
\]
Reduction between Problems

A reduction is a way of defining a solution of a problem with the help of the solutions of another problem.

In recursion theory we are only interested in reductions that are computable.

There are several ways of reducing a problem to another.

The differences between different reductions from $A$ to $B$ consists in the manner and extent to which information about $B$ is allowed to settle questions about $A$. 
Many-One Reduction

The set $A$ is **many-one reducible**, or **m-reducible**, to the set $B$ if there is a **total** computable function $f$ such that

$$x \in A \text{ iff } f(x) \in B$$

for all $x$.

We shall write $A \leq_m B$ or more explicitly $f : A \leq_m B$.

If $f$ is injective, then it is a **one-one reducibility**, denoted by $\leq_1$. 
Many-One Reduction

1. $\leq_m$ is reflexive and transitive.

2. $A \leq_m B$ iff $\overline{A} \leq_m \overline{B}$.

3. $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; $A \leq_m \emptyset$ iff $A = \emptyset$.

4. $\mathbb{N} \leq_m A$ iff $A \neq \emptyset$; $\emptyset \leq_m A$ iff $A \neq \mathbb{N}$.
Non-Recursive Set

**Proposition.** $K = \{ x \mid x \in W_x \}$ is not recursive.
Non-Recursive Set

**Proposition.** \( K = \{ x \mid x \in W_x \} \) is not recursive.

**Proof.** If \( K \) were recursive, then the characteristic function

\[
c(x) = \begin{cases} 1, & \text{if } x \in W_x, \\ 0, & \text{if } x \notin W_x, \end{cases}
\]

would be computable.

Then the function \( g(x) \) defined by

\[
g(x) = \begin{cases} 0, & \text{if } c(x) = 0, \\ \text{undefined,} & \text{if } c(x) = 1. \end{cases}
\]

would also be computable.

Let \( m \) be an index for \( g \). Then

\[
m \in W_m \text{ iff } c(m) = 0 \text{ iff } m \not\in W_m.
\]
Proposition. Neither $\text{Tot} = \{ x \mid \phi_x \text{ is total} \}$ nor $\{ x \mid \phi_x \simeq 0 \}$ is recursive.
Non-Recursive Set

**Proposition.** Neither $\text{Tot} = \{ x \mid \phi_x \text{ is total} \}$ nor $\{ x \mid \phi_x \simeq 0 \}$ is recursive.

**Proof.** Consider the function $f$ defined by

$$f(x, y) = \begin{cases} 
0, & \text{if } x \in W_x, \\
\text{undefined}, & \text{if } x \notin W_x.
\end{cases}$$

By S-m-n Theorem there is a primitive recursive function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x, y)$.

It is clear that $k : K \leq_m \text{Tot}$ and $k : K \leq_m \{ x \mid \phi_x \simeq 0 \}$. 

CSC363-Computability Theory@SJTU Xiaofeng Gao
Rice Theorem

Henry Rice.

Rice Theorem. (1953)

If $\emptyset \subset B \subset C_1$, then $\{x \mid \phi_x \in B\}$ is not recursive.
Rice Theorem. (1953)

If \( \emptyset \subset B \subset C_1 \), then \( \{ x \mid \phi_x \in B \} \) is not recursive.

**Proof.** Suppose \( f_\emptyset \notin B \) and \( g \in B \). Let the function \( f \) be defined by

\[
f(x, y) = \begin{cases} 
g(y), & \text{if } x \in W_x, \\
\text{undefined}, & \text{if } x \notin W_x. 
\end{cases}
\]

By S-m-n Theorem there is some primitive recursive function \( k(x) \) such that \( \phi_{k(x)}(y) \simeq f(x, y) \).

It is clear that \( k \) is a many-one reduction from \( K \) to \( \{ x \mid \phi_x \in B \} \).
According to Rice Theorem the following sets are non-recursive:

- **$Fin = \{x \mid W_x \text{ is finite}\}$**, 
- **$Inf = \{x \mid W_x \text{ is infinite}\}$**, 
- **$Cof = \{x \mid W_x \text{ is cofinite}\}$**, 
- **$Rec = \{x \mid W_x \text{ is recursive}\}$**, 
- **$Tot = \{x \mid \phi_x \text{ is total}\}$**
Rice Theorem deals with programme independent properties. It talks about classes of computable functions rather than classes of programmes.

All non-trivial semantic problems are algorithmically undecidable.

It is of no help to a proof that the set of all polynomial time Turing Machines is undecidable.
The partial characteristic function of a set $A$ is given by

$$\chi_A(x) = \begin{cases} 
1, & \text{if } x \in A, \\
\text{undefined}, & \text{if } x \notin A.
\end{cases}$$

$A$ is recursively enumerable if $\chi_A(x)$ is computable.

**Notation 1:** $A$ is also called semi-recursive set, semi-computable set.

**Notation 2:** subsets of $\mathbb{N}^n$ can be defined as r.e. by coding to r.e. subsets of $\mathbb{N}$. 
A predicate $M(x)$ of natural number is partially decidable if its partial characteristic function

$$\chi_M(x) = \begin{cases} 
1, & \text{if } M(x) \text{ holds,} \\
\text{undefined}, & \text{if } M(x) \text{ does not hold,}
\end{cases}$$

is computable.
A problem $f : \mathbb{N} \to \{0, 1\}$ is partially decidable if $\text{dom}(f)$ is r.e.
Partially Decidable Problem $\iff$ Partially Decidable Predicate $\iff$ Recursively Enumerable Set
Quick Review

**Theorem.** A predicate $M(x)$ is partially decidable iff there is a computable function $g(x)$ such that $M(x) \iff x \in Dom(g)$.

**Theorem.** A predicate $M(x)$ is partially decidable iff there is a decidable predicate $R(x, y)$ such that $M(x) \iff \exists y. R(x, y)$.

**Theorem.** If $M(x, y)$ is partially decidable, so is $\exists y. M(x, y)$.

**Corollary.** If $M(x, y)$ is partially decidable, so is $\exists y. M(x, y)$.

**Theorem.** $M(x)$ is decidable iff both $M(x)$ and $\neg M(x)$ are partially decidable.

**Theorem.** Let $f(x)$ be a partial function. Then $f$ is computable iff the predicate ‘$f(x) \simeq y$’ is partially decidable.
Some Important Decidable Predicates

For each $n \geq 1$, the following predicates are primitive recursive:

1. $S_n(e, x, y, t) \overset{\text{def}}{=} \text{‘}P_e(x) \downarrow \text{ in } t \text{ or fewer steps’}.$

2. $H_n(e, x, t) \overset{\text{def}}{=} \text{‘}P_e(x) \downarrow \text{ in } t \text{ or fewer steps’}.$
Some Important Decidable Predicates

For each $n \geq 1$, the following predicates are primitive recursive:

1. $S_n(e, x, y, t) \overset{\text{def}}{=} 'P_e(x) \downarrow y \text{ in } t \text{ or fewer steps}'.

2. $H_n(e, x, t) \overset{\text{def}}{=} 'P_e(x) \downarrow \text{ in } t \text{ or fewer steps}'.

They are defined by

\[
S_n(e, x, y, t) \overset{\text{def}}{=} j_n(e, x, t) = 0 \land (c_n(e, x, t))_1 = y,
\]
\[
H_n(e, x, t) \overset{\text{def}}{=} j_n(e, x, t) = 0.
\]
Example

1. The halting problem is partially decidable. Its partial characteristic function is given by

\[ \chi_{H}(x, y) = \begin{cases} 
1, & \text{if } P_x(y) \downarrow, \\
\text{undefined}, & \text{otherwise}. 
\end{cases} \]
Example

1. The halting problem is partially decidable. Its partial characteristic function is given by

\[ \chi_H(x, y) = \begin{cases} 1, & \text{if } P_x(y) \downarrow, \\ \text{undefined}, & \text{otherwise}. \end{cases} \]

2. \( K = \{ x \mid x \in W_x \} \) is r.e., but not recursive.

**Proof:** \( \chi_K(x) = 1(\psi_U(x, x)) \).
Example

1. The halting problem is partially decidable. Its partial characteristic function is given by

\[ \chi_{H}(x, y) = \begin{cases} 
1, & \text{if } P_x(y) \downarrow, \\
\text{undefined}, & \text{otherwise}.
\end{cases} \]

2. \( K = \{x \mid x \in W_x\} \) is r.e., but not recursive.

Proof: \( \chi_K(x) = 1(\psi_U(x, x)) \).

3. \( \overline{K} = \{x \mid x \not\in W_x\} \) is not r.e., (also not recursive).

Proof: If yes, then define \( f(x) = \begin{cases} 
1 & \text{if } x \not\in W_x \\
\uparrow & \text{if } x \in W_x
\end{cases} \)

Then \( x \in \text{Dom}(f) \iff x \not\in W_x \). \( f \) is computable while \( \text{Dom}(f) \) doesn’t equal to any computable function. Contradiction!
Example (Cont.)

4. Any recursive set is r.e.
4. Any recursive set is r.e.

5. \( \{x \mid W_x \neq \emptyset \} \) is r.e.

**Proof:** \( W_x \neq \emptyset \iff \exists y \exists t (P_x(y) \downarrow \text{ in } t \text{ steps}) \).
Example (Cont.)

4. Any recursive set is r.e.

5. \( \{ x \mid W_x \neq \emptyset \} \) is r.e.

   **Proof:** \( W_x \neq \emptyset \iff \exists y \exists t (P_x(y) \downarrow \text{ in } t \text{ steps}) \).

6. If \( f \) is a computable function, then \( \text{Ran}(f) \) is r.e.

   **Proof:** Let \( \phi_m \) be the Gödel coding of \( f \).

   \[
   x \in E_m \iff \exists y \exists t (P_m(y) \downarrow x \text{ in } t \text{ steps}).
   \]

   \( x \in E_m \) is partial decidable \( \iff \text{Ran}(f) \) is r.e.
Theorem. A set is r.e. iff it is the domain of a unary computable function.
**Index Theorem**

**Theorem.** A set is r.e. iff it is the domain of a unary computable function.

**Proof:**
“⇒”: A is r.e. ⇒ $\chi_A$ is computable ⇒ “$x \in A \iff x \in \chi_A$".

Thus $A$ is the domain of unary computable function $\chi_A$.

“⇐”: If $f$ is a unary computable function, let $A = Dom(f)$.

Then $\chi_A = 1(f(x))$, which is computable.
Theorem. A set is r.e. iff it is the domain of a unary computable function.

Proof:
“⇒”: A is r.e. ⇒ $\chi_A$ is computable ⇒ “$x \in A \Leftrightarrow x \in \chi_A$". Thus A is the domain of unary computable function $\chi_A$.

“⇐”: If $f$ is a unary computable function, let $A = \text{Dom}(f)$. Then $\chi_A = 1(f(x))$, which is computable.

Notation (Index for Recursively Enumerable Set): $W_0, W_1, W_2, \ldots$ is a repetitive enumeration of all r.e. sets. $e$ is an index of $A$ if $A = W_e$, end every r.e. set has an infinite number of indexes.
Normal Form Theorem

**Theorem.** The set $A$ is r.e. iff there is a primitive recursive predicate $R(x, y)$ such that $x \in A$ iff $\exists y. R(x, y)$. 
Normal Form Theorem

**Theorem.** The set $A$ is r.e. iff there is a primitive recursive predicate $R(x, y)$ such that $x \in A$ iff $\exists y. R(x, y)$.

**Proof.** "\(\Leftarrow\)" If $R(x, y)$ is primitive recursive and $x \in A \iff \exists y. R(x, y)$, then define $g(x) = \mu y R(x, y)$. Then $g(x)$ is computable and $x \in A \iff x \in \text{Dom}(g)$. 
Theorem. The set $A$ is r.e. iff there is a primitive recursive predicate $R(x, y)$ such that $x \in A$ iff $\exists y. R(x, y)$.

Proof. "$\Leftarrow$": If $R(x, y)$ is primitive recursive and $x \in A \iff \exists y. R(x, y)$, then define $g(x) = \mu y R(x, y)$.
Then $g(x)$ is computable and $x \in A \iff x \in \text{Dom}(g)$.

"$\Rightarrow$": suppose $A$ is r.e., then $\chi_A$ is computable. Let $P$ be program to compute $\chi_A$ and $R(x, y)$ be

$$ P(x) \downarrow \text{ in } y \text{ steps.} $$

Then $R(x, y)$ is primitive recursive (decidable) and $x \in A \iff \exists y. R(x, y)$. 
**Theorem** (Applying the Normal Form Theorem). If $M(x, y)$ is partially decidable, so is $\exists y. M(x, y)$ ($\{x | \exists y. M(x, y)\}$ is r.e.).
**Quantifier Contraction Theorem**

**Theorem** (Applying the Normal Form Theorem). If $M(x, y)$ is partially decidable, so is $\exists y. M(x, y)$ ($\{x \mid \exists y. M(x, y)\}$ is r.e.).

**Proof.** Let $R(x, y, z)$ be a primitive recursive predicate such that

$$M(x, y) \iff \exists z. R(x, y, z).$$

Then $\exists y. M(x, y) \iff \exists y. \exists z. R(x, y, z) \iff \exists u. R(x, (u)_0, \cdots, (u)_{m+1}).$

($u = 2^{y_1} 3^{y_2} \cdots p_{m+1}^{y_m}, p_{m+1}^{z}$, if $y = (y_1, \cdots, y_m)$).

By Normal Form Theorem, $\exists y. M(x, y)$ is partially decidable, and $\{x \mid \exists y. M(x, y)\}$ is r.e.
Uniformisation Theorem

**Theorem** (Applying the Normal Form Theorem). If $R(x, y)$ is partially decidable, then there is a computable function $c(x)$ such that $c(x) \downarrow$ iff $\exists y. R(x, y)$ and $c(x) \downarrow$ implies $R(x, c(x))$. 
**Theorem** (Applying the Normal Form Theorem). If $R(x,y)$ is partially decidable, then there is a computable function $c(x)$ such that $c(x) \downarrow$ iff $\exists y. R(x,y)$ and $c(x) \downarrow$ implies $R(x,c(x))$.

We may think of $c(x)$ as a choice function for $R(x,y)$. The theorem states that the choice function is computable.
Complementation Theorem

**Theorem.** $A$ is recursive iff $A$ and $\overline{A}$ are r.e.
Theorem. A is recursive iff \( A \) and \( \overline{A} \) are r.e.

Proof. "\( \Rightarrow \)" : If A is recursive, then \( \chi_A \) and \( \chi_{\overline{A}} \) are computable. Thus \( \Rightarrow \) A and \( \overline{A} \) are r.e.
Theorem. A is recursive iff $A$ and $\overline{A}$ are r.e.

Proof. “$\Rightarrow$”: If $A$ is recursive, then $\chi_A$ and $\chi_{\overline{A}}$ are computable. Thus $\Rightarrow A$ and $\overline{A}$ are r.e.

“$\Leftarrow$”: Suppose $A$ and $\overline{A}$ are r.e. Then some primitive recursive predicates $R(x, y), S(x, y)$ exist such that

$$x \in A \iff \exists y R(x, y),$$
$$x \in \overline{A} \iff \exists y S(x, y).$$

Now let $f(x) = \mu y (R(x, y) \lor S(x, y))$.

Since either $x \in A$ or $x \in \overline{A}$ holds, $f(x)$ is total and computable, and $x \in A \iff R(x, f(x))$. Thus $x \in A$ is decidable $\Rightarrow A$ is recursive.
The Hardest Recursively Enumerable Set

**Fact.** If $A \leq_m B$ and $B$ is r.e. then $A$ is r.e.
The Hardest Recursively Enumerable Set

**Fact.** If $A \leq_m B$ and $B$ is r.e. then $A$ is r.e..

**Theorem.** $A$ is r.e. iff $A \leq_m K$. 
**Fact.** If $A \leq_m B$ and $B$ is r.e. then $A$ is r.e..

**Theorem.** $A$ is r.e. iff $A \leq_m K$.

**Proof.** Suppose $A$ is r.e. Let $f(x, y)$ be defined by

$$f(x, y) = \begin{cases} 1, & \text{if } x \in A, \\ \text{undefined}, & \text{if } x \notin A. \end{cases}$$

By S-m-n Theorem there is a total computable function $s(x)$ such that $f(x, y) = \phi_{s(x)}(y)$. It is clear that $x \in A$ iff $s(x) \in K$. 
The Hardest Recursively Enumerable Set

**Fact.** If $A \leq_m B$ and $B$ is r.e. then $A$ is r.e..

**Theorem.** $A$ is r.e. iff $A \leq_m K$.

**Proof.** Suppose $A$ is r.e. Let $f(x, y)$ be defined by

$$f(x, y) = \begin{cases} 
1, & \text{if } x \in A, \\
\text{undefined,} & \text{if } x \notin A.
\end{cases}$$

By S-m-n Theorem there is a total computable function $s(x)$ such that $f(x, y) = \phi_{s(x)}(y)$. It is clear that $x \in A$ iff $s(x) \in K$.

No r.e. set is more difficult than $K$. 
Proposition. If \( A \) is r.e. but not recursive, then \( \overline{A} \not\leq_m A \not\leq_m \overline{A} \).
Proposition. If $A$ is r.e. but not recursive, then $\overline{A} \not \leq_m A \not \leq_m \overline{A}$.

It contradicts to our intuition that $A$ and $\overline{A}$ are equally difficult.
**Theorem.** Let $f(x)$ be a partial function. Then $f(x)$ is computable iff the predicate ‘$f(x) \simeq y$’ is partially decidable iff $\{ \pi(x, y) \mid f(x) \simeq y \}$ is r.e.
**Graph Theorem**

**Theorem.** Let $f(x)$ be a partial function. Then $f(x)$ is computable iff the predicate ‘$f(x) \simeq y$’ is partially decidable iff \{π(x, y) | f(x) \simeq y\} is r.e.

**Proof.** If $f(x)$ is computable by $P(x)$, then

$$f(x) \simeq y \iff \exists t. (P(x) \downarrow y \text{ in } t \text{ steps}).$$

The predicate ‘$P(x) \downarrow y \text{ in } t \text{ steps}$’ is primitive recursive.
**Theorem.** Let $f(x)$ be a partial function. Then $f(x)$ is computable iff the predicate ‘$f(x) \simeq y$’ is partially decidable iff $\{ \pi(x, y) \mid f(x) \simeq y \}$ is r.e.

**Proof.** If $f(x)$ is computable by $P(x)$, then

$$f(x) \simeq y \iff \exists t. (P(x) \downarrow y \text{ in } t \text{ steps}).$$

The predicate ‘$P(x) \downarrow y \text{ in } t \text{ steps}$’ is primitive recursive.

Conversely let $R(x, y, t)$ be such that

$$f(x) \simeq y \iff \exists t. R(x, y, t).$$

Now $f(x) = \mu y. R(x, y, \mu t. R(x, y, t))$. 
Listing Theorem. \( A \) is r.e. iff either \( A = \emptyset \) or \( A \) is the range of a unary **total** computable function.
**Listing Theorem.** $A$ is r.e. iff either $A = \emptyset$ or $A$ is the range of a unary total computable function.

*Proof.* Suppose $A$ is nonempty and its partial characteristic function is computed by $P$. Let $a$ be a member of $A$. The total function $g(x, t)$ given by

$$g(x, t) = \begin{cases} x, & \text{if } P(x) \downarrow \text{ in } t \text{ steps}, \\ a, & \text{if otherwise.} \end{cases}$$

is computable. Clearly $A$ is the range of $h(z) = g((z)_1, (z)_2)$. 
Listing Theorem. A is r.e. iff either $A = \emptyset$ or $A$ is the range of a unary total computable function.

Proof. Suppose $A$ is nonempty and its partial characteristic function is computed by $P$. Let $a$ be a member of $A$. The total function $g(x, t)$ given by

$$g(x, t) = \begin{cases} x, & \text{if } P(x) \downarrow \text{ in } t \text{ steps}, \\ a, & \text{if otherwise.} \end{cases}$$

is computable. Clearly $A$ is the range of $h(z) = g((z)_1, (z)_2)$.

The converse follows from Graph Theorem. Suppose $A = \text{Ran}(h)$, then

$$x \in A \iff \exists y (h(y) \simeq x) \iff \exists y \exists t (P(y) \downarrow x \text{ in } t \text{ steps})$$
It gives rise to the terminology **recursively enumerable**.

The elements of a r.e. set can be effectively generated. E.g., $A$ can be enumerated as $A = \{h(0), h(1), \ldots, h(n), \ldots\}$, where $h$ is a primitive recursive function.

$\{E_0, E_1, \ldots, E_n, \ldots\}$ is another enumeration of all r.e. sets.

R.e. set are **effectively generated** sets, which is a list compiled by an informal effective procedure (may go on ad infinitum).
An Example

The set \( \{ x \mid \text{if there is a run of exactly } x \text{ consecutive 7’s in the decimal expansion of } \pi \} \) is r.e.
An Example

The set \( \{ x \mid \text{if there is a run of exactly } x \text{ consecutive 7's in the decimal expansion of } \pi \} \) is r.e.

Proof. Run an algorithm that computes successive digits in the decimal expansion of \( \pi \). Each time a run of 7s appears, count the number of consecutive 7s in the run and add this number to the list.
A set is r.e. iff it is the range of a computable function.
Applying Listing Theorem

A set is r.e. iff it is the range of a computable function.

**Equivalence Theorem.** Let $A \subseteq \mathbb{N}$. Then the following are equivalent:

(a). $A$ is r.e.

(b). $A = \emptyset$ or $A$ is the range of a unary total computable function.

(c). $A$ is the range of a (partial) computable function.
Applying Listing Theorem

**Theorem.** Every infinite r.e. set has an infinite recursive subset.
Applying Listing Theorem

**Theorem.** Every infinite r.e. set has an infinite recursive subset.

**Proof.** Suppose $A = \text{Ran}(f)$ where $f$ is a total computable function. An infinite recursive subset is enumerated by the total increasing computable function $g$ given by

$$
\begin{align*}
g(0) &= f(0), \\
g(n + 1) &= f(\mu y (f(y) > g(n))).
\end{align*}
$$

($g$ is total since $A = \text{Ran}(f)$ is infinite. $g$ is computable by minimalisation and recursion). $\text{Ran}(g)$ is an infinite recursive subset of $A$. 
Applying Listing Theorem

**Theorem.** An infinite set is recursive iff it is the range of a total increasing computable function (if it can be recursively enumerated in increasing order).
Applying Listing Theorem

**Theorem.** An infinite set is recursive iff it is the range of a total increasing computable function (if it can be recursively enumerated in increasing order).

**Proof.** "⇒" Suppose $A$ is recursive and infinite. Then $A$ is enumerated by the increasing function $f$ given by

\[
    f(0) = \mu y (y \in A), \\
    f(n+1) = \mu y (y \in A \land y > f(n)).
\]

$f$ is total since $A$ is infinite. $f$ is computable by minimalisation and recursion. $\text{Ran}(g)$ is an infinite recursive subset of $A$. 

Applying Listing Theorem

**Theorem.** An infinite set is recursive iff it is the range of a total increasing computable function (if it can be recursively enumerated in increasing order).

**Proof.** “⇒” Suppose $A$ is recursive and infinite. Then $A$ is enumerated by the increasing function $f$ given by

\[
\begin{align*}
    f(0) &= \mu y (y \in A), \\
    f(n+1) &= \mu y (y \in A \land y > f(n)).
\end{align*}
\]

$f$ is total since $A$ is infinite. $f$ is computable by minimalisation and recursion. $\text{Ran}(g)$ is an infinite recursive subset of $A$.

“⇐”: Suppose $A$ is the range of the computable total increasing function $f$; i.e., $f(0) < f(1) < f(2) < \cdots$ It is clear that if $y = f(n)$ then $n \leq y$. Hence

\[
y \in A \iff y \in \text{Ran}(f) \iff \exists n \leq y (f(n) = y)
\]
Applying Listing Theorem

**Theorem.** The set \( \{ x \mid \phi_x \text{ is total} \} \) is not r.e.
Applying Listing Theorem

**Theorem.** The set \( \{ x \mid \phi_x \text{ is total} \} \) is not r.e.

*Proof.* If \( \{ x \mid \phi_x \text{ is total} \} \) were a r.e. set, then there would be a total computable function \( f \) whose range is the r.e. set.

The function \( g(x) \) given by \( g(x) = \phi_{f(x)}(x) + 1 \) would be total and computable.
Let \( f(x, y) = \begin{cases} 1 & \text{if } P_x(x) \text{ does not converge in } y \text{ or fewer steps}, \\ \text{undefined} & \text{otherwise}. \end{cases} \)

Since \( f(x, y) \) is computable by Church’s Thesis, from s-m-n theorem, there is a total computable function \( k(x) \), such that \( \phi_{k(x)}(y) \simeq f(x, y) \).

From the definition of \( f \), we have

\[
\begin{align*}
& x \in W_x \Rightarrow (\exists y)(P_x(x) \text{ converges in } y \text{ steps}) \Rightarrow \phi_{k(x)} \text{ is not total} \\
& x \notin W_x \Rightarrow (\forall y)(P_x(x) \text{ does not converge in } y \text{ steps}) \Rightarrow \phi_{k(x)} \text{ is total}
\end{align*}
\]

Therefore, ‘\( x \notin W_x \)’ iff. ‘\( \phi_{k(x)} \) is total’. We have ‘\( \phi_x \) is total’ is not partially computable.
**Theorem.** The recursively enumerable sets are closed under union and intersection uniformly and effectively.
Theorem. The recursively enumerable sets are closed under union and intersection uniformly and effectively.

Proof. According to S-m-n Theorem there are primitive recursive functions $r(x, y), s(x, y)$ such that

\[ W_{r(x,y)} = W_x \cup W_y, \]
\[ W_{s(x,y)} = W_x \cap W_y. \]
**Theorem.** The recursively enumerable sets are closed under union and intersection uniformly and effectively.

**Proof.** According to S-m-n Theorem there are primitive recursive functions \( r(x, y), s(x, y) \) such that

\[
W_r(x, y) = W_x \cup W_y, \\
W_s(x, y) = W_x \cap W_y.
\]
Rice-Shapiro Theorem. Suppose that $\mathcal{A}$ is a set of unary computable functions such that the set $\{x \mid \phi_x \in \mathcal{A}\}$ is r.e. Then for any unary computable function $f$, $f \in \mathcal{A}$ iff there is a finite function $\theta \subseteq f$ with $\theta \in \mathcal{A}$. 
Proof of Rice-Shapiro Theorem

Suppose $A = \{ x \mid \phi_x \in \mathcal{A} \}$ is r.e.
Proof of Rice-Shapiro Theorem

Suppose \( A = \{ x \mid \phi_x \in \mathcal{A} \} \) is r.e.

Suppose \( f \in \mathcal{A} \) but \( \forall \) finite \( \theta \subseteq f. \theta \notin \mathcal{A} \).
Proof of Rice-Shapiro Theorem

Suppose \( A = \{ x \mid \phi_x \in \mathcal{A} \} \) is r.e.

Suppose \( f \in \mathcal{A} \) but \( \forall \) finite \( \theta \subseteq f. \theta \notin \mathcal{A} \).

Let \( P \) be a partial characteristic function of \( K \).
Define the computable function \( g(z, t) \) by

\[
g(z, t) \simeq \begin{cases} f(t), & \text{if } P(z) \downarrow \text{ in } t \text{ steps,} \\ \uparrow, & \text{otherwise.} \end{cases}
\]

According to S-m-n Theorem, there is a primitive recursive function \( s(z) \) such that \( g(z, t) \simeq \phi_{s(z)}(t) \).
Proof of Rice-Shapiro Theorem

Suppose $A = \{ x \mid \phi_x \in A \}$ is r.e.

Suppose $f \in A$ but $\forall$ finite $\theta \subseteq f. \theta \notin A$.

Let $P$ be a partial characteristic function of $K$
Define the computable function $g(z, t)$ by

$$
 g(z, t) \simeq \begin{cases} 
 f(t), & \text{if } P(z) \downarrow \text{ in } t \text{ steps,} \\
 \uparrow, & \text{otherwise.}
\end{cases}
$$

According to S-m-n Theorem, there is a primitive recursive function $s(z)$ such that $g(z, t) \simeq \phi_{s(z)}(t)$.

By construction $\phi_{s(z)} \subseteq f$ for all $z$.
Proof of Rice-Shapiro Theorem

Suppose \( A = \{x \mid \phi_x \in \mathcal{A}\} \) is r.e.

Suppose \( f \in \mathcal{A} \) but \( \forall \) finite \( \theta \subseteq f. \theta \notin \mathcal{A} \).

Let \( P \) be a partial characteristic function of \( K \).

Define the computable function \( g(z, t) \) by

\[
g(z, t) \sim \begin{cases} f(t), & \text{if } P(z) \downarrow \text{ in } t \text{ steps}, \\ \uparrow, & \text{otherwise.} \end{cases}
\]

According to S-m-n Theorem, there is a primitive recursive function \( s(z) \) such that \( g(z, t) \sim \phi_{s(z)}(t) \).

By construction \( \phi_{s(z)} \subseteq f \) for all \( z \).

\( z \in K \Rightarrow \phi_{s(z)} \) is finite \( \Rightarrow s(z) \notin A; \) 
\( z \notin K \Rightarrow \phi_{s(z)} = f \Rightarrow s(z) \in A. \)
Proof of Rice-Shapiro Theorem

Suppose $f$ is a computable function and there is a finite $\theta \in \mathcal{A}$ such that $\theta \subseteq f$ and $f \notin \mathcal{A}$.
Proof of Rice-Shapiro Theorem

Suppose $f$ is a computable function and there is a finite $\theta \in \mathcal{A}$ such that $\theta \subseteq f$ and $f \notin \mathcal{A}$.

Define the computable function $g(z, t)$ by

$$g(z, t) \simeq \begin{cases} f(t), & \text{if } t \in \text{Dom}(\theta) \lor z \in K, \\ \uparrow, & \text{otherwise.} \end{cases}$$

According to S-m-n Theorem, there is a primitive recursive function $s(z)$ such that $g(z, t) \simeq \phi_{s(z)}(t)$. 

Suppose $f$ is a computable function and there is a finite $\theta \in \mathcal{A}$ such that $\theta \subseteq f$ and $f \notin \mathcal{A}$.

Define the computable function $g(z, t)$ by

$$g(z, t) \simeq \begin{cases} f(t), & \text{if } t \in Dom(\theta) \lor z \in K, \\ \uparrow, & \text{otherwise}. \end{cases}$$

According to S-m-n Theorem, there is a primitive recursive function $s(z)$ such that $g(z, t) \simeq \phi_{s(z)}(t)$.

$$z \in K \Rightarrow \phi_{s(z)} = f \Rightarrow s(z) \notin A;$$
$$z \notin K \Rightarrow \phi_{s(z)} = \theta \Rightarrow s(z) \in A.$$
Reversing Rice-Shapiro Theorem

\[ \{ x \mid \phi_x \in \mathcal{A} \} \text{ is r.e. if the following hold:} \]

1. \( \Theta = \{ g(\theta) \mid \theta \in \mathcal{A} \text{ and } \theta \text{ is finite} \} \) is r.e., where \( g \) is a canonical encoding of the finite functions.

2. \( \forall f \in \mathcal{A}, \exists \text{ finite } \theta \in \mathcal{A}, \theta \subseteq f. \)
Corollary

The sets \( \{ x \mid \phi_x \text{ is total} \} \) and \( \{ x \mid \phi_x \text{ is not total} \} \) are not r.e.
Corollary

The sets \( \{ x \mid \phi_x \text{ is total} \} \) and \( \{ x \mid \phi_x \text{ is not total} \} \) are not r.e.

*Proof.* Consider the set \( \mathcal{A} = \{ f \mid f \in C_1 \land f \text{ is total} \} \). For no \( f \in \mathcal{A} \) is there a finite \( \theta \subseteq f \) with \( \theta \in \mathcal{A} \). Hence \( \{ x \mid \phi_x \text{ is total} \} \) is not r.e.
The sets \( \{x \mid \phi_x \text{ is total}\} \) and \( \{x \mid \phi_x \text{ is not total}\} \) are not r.e.

**Proof.** Consider the set \( \mathcal{A} = \{f \mid f \in \mathcal{C}_1 \land f \text{ is total}\} \). For no \( f \in \mathcal{A} \) is there a finite \( \theta \subseteq f \) with \( \theta \in \mathcal{A} \). Hence \( \{x \mid \phi_x \text{ is total}\} \) is not r.e.

Consider the set \( \mathcal{B} = \{f \mid f \in \mathcal{C}_1 \land f \text{ is not total}\} \). Then if \( f \) is any total computable function, \( f \not\in \mathcal{B} \); but every finite function \( \theta \subseteq f \) is in \( \mathcal{B} \). Hence \( \{x \mid \phi_x \text{ is not total}\} \) is not r.e. by Rice-Shapiro theorem.
The following sets are not recursively enumerable:

\[\begin{align*}
\text{Fin} & = \{x \mid W_x \text{ is finite}\}, \\
\text{Inf} & = \{x \mid W_x \text{ is infinite}\}, \\
\text{Cof} & = \{x \mid W_x \text{ is cofinite}\}, \\
\text{Rec} & = \{x \mid W_x \text{ is recursive}\}, \\
\text{Tot} & = \{x \mid \phi_x \text{ is total}\}, \\
\text{Con} & = \{x \mid \phi_x \text{ is total and constant}\}, \\
\text{Ext} & = \{x \mid \phi_x \text{ is extensible to a total recursive function}\}.
\end{align*}\]
Outline

1. Recursive Sets
   - Decidable Predicate
   - Reduction
   - Rice Theorem

2. Recursively Enumerable Set
   - Partial Decidable Predicates
   - Theorems

3. Special Sets
   - Productive Sets
   - Creative Set
   - Simple Sets
Non-r.e. Sets

**Target.** We consider non-r.e. sets to form *creative sets*. Suppose $A$ is any non-r.e. set, then if $W_x$ is an r.e. set contained in $A$, there must be a number $y \in A \setminus W_x$. This number $y$ is a witness of $A \neq W_x$. 
Non-r.e. Sets

**Target.** We consider non-r.e. sets to form *creative sets*. Suppose $A$ is any non-r.e. set, then if $W_x$ is an r.e. set contained in $A$, there must be a number $y \in A \setminus W_x$. This number $y$ is a witness of $A \neq W_x$.

**Example.** Consider $\overline{K} = \{x \mid x \not\in W_x\}$
**Non-r.e. Sets**

**Target.** We consider non-r.e. sets to form creative sets. Suppose $A$ is any non-r.e. set, then if $W_x$ is an r.e. set contained in $A$, there must be a number $y \in A \setminus W_x$. This number $y$ is a witness of $A \neq W_x$.

**Example.** Consider $\overline{K} = \{x \mid x \not\in W_x\}$

Suppose $W_x \subseteq \overline{K}$. Then $x \in \overline{K} \setminus W_x$. So $x$ is a witness that the inclusion $W_x \subseteq \overline{K}$ is strict.
Non-r.e. Sets

Target. We consider non-r.e. sets to form creative sets. Suppose $A$ is any non-r.e. set, then if $W_x$ is an r.e. set contained in $A$, there must be a number $y \in A \setminus W_x$. This number $y$ is a witness of $A \neq W_x$.

Example. Consider $\overline{K} = \{ x \mid x \not\in W_x \}$

Suppose $W_x \subseteq \overline{K}$. Then $x \in \overline{K} \setminus W_x$. So $x$ is a witness that the inclusion $W_x \subseteq \overline{K}$ is strict.

We call $\overline{K}$ productive.
**Definition.** A set $A$ is **productive** if there is a total computable function $g$ such that whenever $W_x \subseteq A$, then $g(x) \in A \setminus W_x$.

The function is called a **productive function** for $A$. 
**Definition.** A set $A$ is **productive** if there is a total computable function $g$ such that whenever $W_x \subseteq A$, then $g(x) \in A \setminus W_x$.

The function is called a **productive function** for $A$.

**Notation.** A productive set is not r.e.
**Definition.** A set $A$ is **productive** if there is a total computable function $g$ such that whenever $W_x \subseteq A$, then $g(x) \in A \setminus W_x$.

The function is called a **productive function** for $A$.

**Notation.** A productive set is not r.e.

**Example.** $\overline{K}$ is productive with productive function $g(x) = x$. 

![Diagram](image_url) **Fig. A productive set**
**Reduction Theorem**

**Theorem.** Suppose that $A$ and $B$ are sets such that $A$ is productive, and there is a total computable function such that $x \in A$ iff $f(x) \in B$. Then $B$ is productive.
**Reduction Theorem**

**Theorem.** Suppose that $A$ and $B$ are sets such that $A$ is productive, and there is a total computable function such that $x \in A$ iff $f(x) \in B$. Then $B$ is productive.

**Proof.** Suppose $W_x \subseteq B$. Then $W_z = f^{-1}(W_x) \subseteq f^{-1}(B) = A$ for some $z$. 
**Theorem.** Suppose that $A$ and $B$ are sets such that $A$ is productive, and there is a total computable function such that $x \in A$ iff $f(x) \in B$. Then $B$ is productive.

**Proof.** Suppose $W_x \subseteq B$. Then $W_z = f^{-1}(W_x) \subseteq f^{-1}(B) = A$ for some $z$.

Moreover, $f^{-1}(W_x)$ is r.e. (by substitution), so there is a $z$ such that $f^{-1}(W_x) = W_z$. Now $W_z \subseteq A$, and $g(z) \in A \setminus W_z$. Hence $f(g(z)) \in B \setminus W_x$. 
**Theorem.** Suppose that $A$ and $B$ are sets such that $A$ is productive, and there is a total computable function such that $x \in A$ iff $f(x) \in B$. Then $B$ is productive.

**Proof.** Suppose $W_x \subseteq B$. Then $W_z = f^{-1}(W_x) \subseteq f^{-1}(B) = A$ for some $z$.

Moreover, $f^{-1}(W_x)$ is r.e. (by substitution), so there is a $z$ such that $f^{-1}(W_x) = W_z$. Now $W_z \subseteq A$, and $g(z) \in A \setminus W_z$. Hence $f(g(z)) \in B \setminus W_x$.

$f(g(z))$ is a witness to the fact that $W_x \neq B$. 
**Reduction Theorem**

**Theorem.** Suppose that $A$ and $B$ are sets such that $A$ is productive, and there is a total computable function such that $x \in A$ iff $f(x) \in B$. Then $B$ is productive.

**Proof.** Suppose $W_x \subseteq B$. Then $W_z = f^{-1}(W_x) \subseteq f^{-1}(B) = A$ for some $z$.

Moreover, $f^{-1}(W_x)$ is r.e. (by substitution), so there is a $z$ such that $f^{-1}(W_x) = W_z$. Now $W_z \subseteq A$, and $g(z) \in A \setminus W_z$. Hence $f(g(z)) \in B \setminus W_x$.

$f(g(z))$ is a witness to the fact that $W_x \neq B$.

We now need to obtain the witness $f(g(z))$ effectively from $x$. Apply the s-m-n theorem to $\phi_x(f(y))$, one gets a total computable function $k(x)$ such that $\phi_{k(x)}(y) = \phi_x(f(y))$. Then $W_{k(x)} = f^{-1}(W_x)$. It follows that $f(g(k(x))) \in B \setminus W_x$. 
Proof

\[ A = f^{-1}(B) \]
\[ W_z = f^{-1}(W_x) \]
\[ g(z) \]

\[ f \]
\[ B \]
\[ W_x \]
\[ f(g(z)) \]
Examples

1. \( \{ x \mid \phi_x \neq 0 \} \) is productive.
Examples

1. \( \{ x \mid \phi_x \neq 0 \} \) is productive.

Proof. \( f(x, y) = \begin{cases} 
0 & \text{if } x \in W_x \\
\uparrow & \text{if } x \not\in W_x 
\end{cases} \). Reduce from \( \overline{K} \).
Examples

1. \( \{ x \mid \phi_x \neq 0 \} \) is productive.

**Proof.** \( f(x, y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \not\in W_x \end{cases} \). Reduce from \( \overline{K} \).

2. \( \{ x \mid c \not\in W_x \} \) is productive.
Examples

1. \( \{ x \mid \phi_x \neq 0 \} \) is productive.

   **Proof.** \( f(x, y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \). Reduce from \( \overline{K} \).

2. \( \{ x \mid c \notin W_x \} \) is productive.

   **Proof.** \( f(x, y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \). Reduce from \( \overline{K} \).
Examples

1. \( \{ x \mid \phi_x \neq 0 \} \) is productive.

**Proof.** \( f(x, y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \). Reduce from \( \overline{K} \).

2. \( \{ x \mid c \notin W_x \} \) is productive.

**Proof.** \( f(x, y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \). Reduce from \( \overline{K} \).

3. \( \{ x \mid c \notin E_x \} \) is productive.
Application of Rich’s Theorem

**Theorem.** Suppose that $\mathcal{B}$ is a set of unary computable functions with $f_\emptyset \in \mathcal{B}$ and $\mathcal{B} \neq \mathcal{C}_1$. Then the set $B = \{ x \mid \phi_x \in \mathcal{B} \}$ is productive.
Theorem. Suppose that $\mathcal{B}$ is a set of unary computable functions with $f_\emptyset \in \mathcal{B}$ and $\mathcal{B} \neq \mathcal{C}_1$. Then the set $B = \{x \mid \phi_x \in \mathcal{B}\}$ is productive.

Proof. Choose a computable function $g \notin \mathcal{B}$. Consider function $f$ defined by

$$f(x, y) = \begin{cases} 
g(y), & \text{if } x \in W_x, \\
\uparrow, & \text{if } x \notin W_x. 
\end{cases}$$
Application of Rich’s Theorem

**Theorem.** Suppose that $\mathcal{B}$ is a set of unary computable functions with $f_\emptyset \in \mathcal{B}$ and $\mathcal{B} \neq \mathcal{C}_1$. Then the set $B = \{x \mid \phi_x \in \mathcal{B}\}$ is productive.

**Proof.** Choose a computable function $g \notin \mathcal{B}$. Consider function $f$ defined by

$$f(x, y) = \begin{cases} g(y), & \text{if } x \in W_x, \\ \uparrow, & \text{if } x \notin W_x. \end{cases}$$

By s-m-n theorem there is some total computable function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x, y)$.
Application of Rich’s Theorem

**Theorem.** Suppose that \( \mathcal{B} \) is a set of unary computable functions with \( f_\emptyset \in \mathcal{B} \) and \( \mathcal{B} \neq \mathcal{C}_1 \). Then the set \( B = \{ x \mid \phi_x \in \mathcal{B} \} \) is productive.

**Proof.** Choose a computable function \( g \notin \mathcal{B} \). Consider function \( f \) defined by

\[
f(x, y) = \begin{cases} 
g(y), & \text{if } x \in W_x, \\
\uparrow, & \text{if } x \notin W_x. 
\end{cases}
\]

By s-m-n theorem there is some total computable function \( k(x) \) such that \( \phi_{k(x)}(y) \simeq f(x, y) \).

It is clear that \( x \in W_x \) iff \( \phi_{k(x)} = g \) iff \( \phi_{k(x)} \notin \mathcal{B} \). Thus \( x \in \overline{K} \) iff \( k(x) \in B \).
Application of Rich’s Theorem

**Theorem.** Suppose that $\mathcal{B}$ is a set of unary computable functions with $f_{\emptyset} \in \mathcal{B}$ and $\mathcal{B} \neq C_1$. Then the set $B = \{x \mid \phi_x \in \mathcal{B}\}$ is productive.

**Proof.** Choose a computable function $g \notin \mathcal{B}$. Consider function $f$ defined by

$$f(x, y) = \begin{cases} g(y), & \text{if } x \in W_x, \\ \uparrow, & \text{if } x \notin W_x. \end{cases}$$

By s-m-n theorem there is some total computable function $k(x)$ such that $\phi_{k(x)}(y) \simeq f(x, y)$.

It is clear that $x \in W_x$ iff $\phi_{k(x)} = g$ iff $\phi_{k(x)} \notin \mathcal{B}$. Thus $x \in \overline{K}$ iff $k(x) \in B$.

**Example.** $\{x \mid \phi_x \text{ is not total}\}$ is productive.

($\mathcal{B} = \{f \mid f \in C_1 \land f \text{ is not total}\}$.)
**Creative Sets**

**Definition.** A set \( A \) is **creative** if it is r.e. and its complement \( \overline{A} \) is productive.
Creative Sets

**Definition.** A set $A$ is **creative** if it is r.e. and its complement $\overline{A}$ is productive.

**Example.** $K$ is creative. (The simplest example of a creative set).
Definition. A set $A$ is creative if it is r.e. and its complement $\overline{A}$ is productive.

Example. $K$ is creative. (The simplest example of a creative set).

Notation. From the theorem that $A$ is recursive $\iff A$ and $\overline{A}$ are r.e. we can say that a creative set is an r.e. set that fails to be recursive in a very strong way. (Creative sets are r.e. sets having the most difficult decision problem.)
Examples

1. \( \{ x \mid c \in W_x \} \) is creative.
Examples

1. \( \{ x \mid c \in W_x \} \) is creative.

2. \( \{ x \mid c \in E_x \} \) is creative.
Examples

1. \( \{ x \mid c \in W_x \} \) is creative.

2. \( \{ x \mid c \in E_x \} \) is creative.

3. \( A = \{ x \mid \phi_x(x) = 0 \} \) is creative.
Examples

1. \( \{ x \mid c \in W_x \} \) is creative.

2. \( \{ x \mid c \in E_x \} \) is creative.

3. \( A = \{ x \mid \phi_x(x) = 0 \} \) is creative.

Proof. \( A \) is r.e.
Examples

1. \( \{ x \mid c \in W_x \} \) is creative.

2. \( \{ x \mid c \in E_x \} \) is creative.

3. \( A = \{ x \mid \phi_x(x) = 0 \} \) is creative.

Proof. \( A \) is r.e.

To obtain a productive function for \( \overline{A} \), by s-m-n theorem one gets a total computable function \( g(x) \) such that \( \phi_{g(x)}(y) = 0 \iff \phi_x(y) \) is defined.

Then \( g(x) \in A \iff g(x) \in W_x \). So if \( W_x \subseteq \overline{A} \) we must have \( g(x) \in \overline{A} \setminus W_x \).

Thus \( g \) is a productive function for \( \overline{A} \).
Theorem. Suppose that $A \subseteq C_1$ and let $A = \{x \mid \phi_x \in A\}$. If $A$ is r.e. and $A \neq \emptyset, \mathbb{N}$, then $A$ is creative.
Theorem. Suppose that $\mathcal{A} \subseteq \mathcal{C}_1$ and let $A = \{x \mid \phi_x \in \mathcal{A}\}$. If $A$ is r.e. and $A \neq \emptyset, \mathbb{N}$, then $A$ is creative.

Proof. Suppose $A$ is r.e. and $A \neq \emptyset, \mathbb{N}$.

If $f_\emptyset \in \mathcal{A}$, then $A$ is productive by the previous theorem. This is a contradiction.

Thus $f_\emptyset \notin \mathcal{A}$. $\overline{A}$ is productive by the same theorem. Hence $A$ is creative.
Examples

1. \( A = \{ x \mid c \in W_x \} \) is creative. It corresponds to 
\[ \mathcal{A} = \{ f \in \mathcal{C}^1 \mid f(c) \downarrow \}. \]
Examples

1. \( A = \{x \mid c \in W_x\} \) is creative. It corresponds to \( A = \{f \in \mathcal{C}_1 \mid f(c) \downarrow\} \).

2. \( A = \{x \mid c \in E_x\} \) is creative. It corresponds to \( A = \{f \in \mathcal{C}_1 \mid \exists x(f(x) \downarrow c)\} \).
Examples

1. $A = \{x \mid c \in W_x\}$ is creative. It corresponds to $A = \{f \in C_1 \mid f(c) \downarrow\}$.

2. $A = \{x \mid c \in E_x\}$ is creative. It corresponds to $A = \{f \in C_1 \mid \exists x(f(x) \downarrow c)\}$.

3. $A = \{x \mid W_x \neq \emptyset\}$ is creative. It corresponds to $A = \{f \in C_1 \mid f \neq f_\emptyset\}$. 
Discussion

**Question.** Are all non-recursive r.e. sets creative?
**Question.** Are all non-recursive r.e. sets creative?

The answer is negative. By a special construction we can obtain r.e. sets that are neither recursive nor creative.
Lemma. Suppose that $g$ is a total computable function. Then there is a total computable function $k$ such that for all $x$, $W_{k(x)} = W_x \cup \{g(x)\}$. 
Lemma. Suppose that $g$ is a total computable function. Then there is a total computable function $k$ such that for all $x$, $W_{k(x)} = W_x \cup \{g(x)\}$.

Proof. Using the s-m-n theorem, take $k(x)$ to be a total computable function such that

$$\phi_{k(x)}(y) = \begin{cases} 
1, & \text{if } y \in W_x \lor y = g(x), \\
\uparrow, & \text{otherwise}
\end{cases}.$$
Subset Theorem

**Theorem.** A productive set contains an infinite r.e. subset.
Subset Theorem

**Theorem.** A productive set contains an infinite r.e. subset.

**Proof.** Let $A$ be a productive set with productive function $g$. The idea is to enumerate a non-repetitive infinite set $B = \{y_0, y_1, \cdots \} \subseteq A$. 
Theorem. A productive set contains an infinite r.e. subset.

Proof. Let $A$ be a productive set with productive function $g$. The idea is to enumerate a non-repetitive infinite set $B = \{y_0, y_1, \cdots \} \subseteq A$.

Take $e_0$ to be some index for $W_{e_0} = \emptyset$. Since $W_{e_0} \subseteq A$, $g(e_0) \in A$. Put $y_0 = g(e_0) \in A$. 
**Theorem.** A productive set contains an infinite r.e. subset.

**Proof.** Let $A$ be a productive set with productive function $g$. The idea is to enumerate a non-repetitive infinite set $B = \{y_0, y_1, \cdots \} \subseteq A$.

Take $e_0$ to be some index for $W_{e_0} = \emptyset$. Since $W_{e_0} \subseteq A$, $g(e_0) \in A$.

Put $y_0 = g(e_0) \in A$.

For $n \geq 0$, assume $\{y_0, \cdots, y_n\} \subseteq A$. Find an $e_{n+1}$ s.t. $\{y_0, \cdots, y_n\} = W_{e_{n+1}} \subseteq A$. Then $g(e_{n+1}) \in A \setminus W_{e_{n+1}}$. Thus if we put $y_{n+1} = g(e_{n+1})$, we have $y_{n+1} \in A$ and $y_{n+1} \neq y_0, \cdots, y_n$. 


Subset Theorem

**Theorem.** A productive set contains an infinite r.e. subset.

**Proof.** Let $A$ be a productive set with productive function $g$. The idea is to enumerate a non-repetitive infinite set $B = \{y_0, y_1, \cdots \} \subseteq A$.

Take $e_0$ to be some index for $W_{e_0} = \emptyset$. Since $W_{e_0} \subseteq A$, $g(e_0) \in A$. Put $y_0 = g(e_0) \in A$.

For $n \geq 0$, assume $\{y_0, \cdots, y_n\} \subseteq A$. Find an $e_{n+1}$ s.t. $\{y_0, \cdots, y_n\} = W_{e_{n+1}} \subseteq A$. Then $g(e_{n+1}) \in A \setminus W_{e_{n+1}}$. Thus if we put $y_{n+1} = g(e_{n+1})$, we have $y_{n+1} \in A$ and $y_{n+1} \neq y_0, \cdots, y_n$.

By the Lemma there is some total computable function $k$ such that for all $x$, $W_{k(x)} = W_x \cup \{g(x)\}$. So the infinite set $\{e_0, \ldots, k^n(e_0), \ldots\}$ is r.e.
Subset Theorem

**Theorem.** A productive set contains an infinite r.e. subset.

**Proof.** Let $A$ be a productive set with productive function $g$. The idea is to enumerate a non-repetitive infinite set $B = \{y_0, y_1, \cdots \} \subseteq A$. Take $e_0$ to be some index for $W_{e_0} = \emptyset$. Since $W_{e_0} \subseteq A$, $g(e_0) \in A$. Put $y_0 = g(e_0) \in A$.

For $n \geq 0$, assume $\{y_0, \cdots, y_n\} \subseteq A$. Find an $e_{n+1}$ s.t. $\{y_0, \cdots, y_n\} = W_{e_{n+1}} \subseteq A$. Then $g(e_{n+1}) \in A \setminus W_{e_{n+1}}$. Thus if we put $y_{n+1} = g(e_{n+1})$, we have $y_{n+1} \in A$ and $y_{n+1} \neq y_0, \cdots, y_n$.

By the Lemma there is some total computable function $k$ such that for all $x$, $W_k(x) = W_x \cup \{g(x)\}$. So the infinite set $\{e_0, \ldots, k^n(e_0), \ldots \}$ is r.e.

It follows that the infinite set $\{g(e_0), \ldots, g(k^n(e_0)), \ldots \}$ is a r.e. subset of $A$. 
Illumination

\[ y_{n+1} = g(e_{n+1}) \]
Corollary

If $A$ is creative, then $\overline{A}$ contains an infinite r.e. subset.
Simple Sets

**Definition.** A set $A$ is **simple** if

(i) $A$ is r.e.,

(ii) $\overline{A}$ is infinite,

(iii) $\overline{A}$ contains no infinite r.e. subset.
**Theorem.** A simple set is neither recursive nor creative.
Simple Sets

**Theorem.** A simple set is neither recursive nor creative.

*Proof.* Since \( \overline{A} \) can not be r.e., \( A \) can not be recursive.
**Theorem.** A simple set is neither recursive nor creative.

*Proof.* Since $\overline{A}$ can not be r.e., $A$ can not be recursive. (iii) implies that $A$ can not be creative.
Theorem. There is a simple set.
**Theorem.** There is a simple set.

**Proof.** Define $f(x) = \phi_x(\mu z (\phi_x(z) > 2x))$. Let $A$ be $\text{Ran}(f)$. 
Theorem. There is a simple set.

Proof. Define \( f(x) = \phi_x(\mu z (\phi_x(z) > 2x)) \). Let \( A \) be \( \text{Ran}(f) \).

(i) \( A \) is r.e.
Theorem. There is a simple set.

Proof. Define $f(x) = \phi_x(\mu z (\phi_x(z) > 2x))$. Let $A$ be $\text{Ran}(f)$.

(i) $A$ is r.e.

(ii) $\overline{A}$ is infinite. This is because $A \cap \{0, 1, \ldots, 2n\}$ contains at most the elements $\{f(0), f(1), \ldots, f(n - 1)\}$. 
Simple Sets

**Theorem.** There is a simple set.

**Proof.** Define \( f(x) = \phi_x(\mu z(\phi_x(z) > 2x)) \). Let \( A \) be \( \text{Ran}(f) \).

(i) \( A \) is r.e.

(ii) \( \overline{A} \) is infinite. This is because \( A \cap \{0, 1, \ldots, 2n\} \) contains at most the elements \( \{f(0), f(1), \ldots, f(n - 1)\} \).

(iii) Suppose \( B \) is an infinite r.e. set. Then there is a total computable function \( \phi_b \) such that \( B = E_b \). Since \( \phi_b \) is total, \( f(b) \) is defined and \( f(b) \in A \). Hence \( B \not\subseteq \overline{A} \).