Reducibility and Degree*

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

Outline

General Remark

A problem is a set of numbers.

General Remark

A problem is a set of numbers.

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

General Remark

A problem is a set of numbers.

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

There are several inequivalent ways of reducing a problem to another problem.

General Remark

A problem is a set of numbers.

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

There are several inequivalent ways of reducing a problem to another problem.

The differences between different reductions 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 (m-reducible) to the set B if there is a total computable function f such that $x \in A$ iff $f(x) \in B$ for all x.

Many-One Reduction

The set A is many-one reducible (m-reducible) to the set B if there is a total computable function f such that $x \in A$ iff $f(x) \in B$ for all x.

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

Many-One Reduction

The set A is many-one reducible (m-reducible) to the set B if there is a total computable function f such that $x \in A$ 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 we are talking about one-one reducibility.

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

$$f_{\mathbf{0}}(x,y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases}$$

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

$$f_{\mathbf{0}}(x,y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases}$$

2. Rice Theorem is proved by showing that $K \leq_m \{x \mid \phi_x \in \mathscr{B}\}$.

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

$$f_{\mathbf{0}}(x,y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases}$$

2. Rice Theorem is proved by showing that $K \leq_m \{x \mid \phi_x \in \mathscr{B}\}$.

$$f_g(x,y) = \begin{cases} g(y) & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad \begin{array}{l} x \in W_x \Rightarrow \phi_k(x) = g \in \mathscr{B} \\ x \notin W_x \Rightarrow \phi_k(x) = f_\varnothing \notin \mathscr{B} \end{array}$$

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

$$f_{\mathbf{0}}(x,y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases}$$

2. Rice Theorem is proved by showing that $K \leq_m \{x \mid \phi_x \in \mathscr{B}\}$.

$$f_g(x,y) = \begin{cases} g(y) & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad \begin{array}{l} x \in W_x \Rightarrow \phi_k(x) = g \in \mathscr{B} \\ x \notin W_x \Rightarrow \phi_k(x) = f_\varnothing \notin \mathscr{B} \end{array}$$

3. $\{x \mid \phi_x \text{ is total}\} \leq_m \{x \mid \phi_x = \mathbf{0}\}$.

Examples

1. K is m-reducible to $\{x \mid \phi_x = \mathbf{0}\}$, $\{x \mid c \in W_x\}$ and $\{x \mid \phi_x \text{ is total}\}$.

$$f_{\mathbf{0}}(x,y) = \begin{cases} 0 & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases}$$

2. Rice Theorem is proved by showing that $K \leq_m \{x \mid \phi_x \in \mathscr{B}\}$.

$$f_g(x,y) = \begin{cases} g(y) & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \qquad \begin{array}{l} x \in W_x \Rightarrow \phi_k(x) = g \in \mathscr{B} \\ x \notin W_x \Rightarrow \phi_k(x) = f_\emptyset \notin \mathscr{B} \end{array}$$

3. $\{x \mid \phi_x \text{ is total}\} \leq_m \{x \mid \phi_x = \mathbf{0}\}$.

$$\phi_{k(x)} = \mathbf{0} \circ \phi_x$$

Elementary Properties

Let A, B, C be sets.

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

   $f : A \leq_m A$ is the identity function.

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

   $f : A \leq_m A$ is the identity function.

2. $\leq_m$ transitive: $A \leq_m B, B \leq_m C \Rightarrow A \leq_m C$.

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

   $f : A \leq_m A$ is the identity function.

2. $\leq_m$ transitive: $A \leq_m B$, $B \leq_m C \Rightarrow A \leq_m C$.

   Let $f : A \leq_m B$, $g : B \leq_m C$, then $g \circ f : A \leq_m C$.

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

   $f : A \leq_m A$ is the identity function.

2. $\leq_m$ transitive: $A \leq_m B$, $B \leq_m C \Rightarrow A \leq_m C$.

   Let $f : A \leq_m B$, $g : B \leq_m C$, then $g \circ f : A \leq_m C$.

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

Elementary Properties

Let A, B, C be sets.

1. $\leq_m$ is reflexive: $A \leq_m A$.

   $f : A \leq_m A$ is the identity function.

2. $\leq_m$ transitive: $A \leq_m B$, $B \leq_m C \Rightarrow A \leq_m C$.

   Let $f : A \leq_m B$, $g : B \leq_m C$, then $g \circ f : A \leq_m C$.

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

   If $g : A \leq_m B$, then $x \in A \Leftrightarrow f(x) \in B$; so $x \in \overline{A} \Leftrightarrow g(x) \in \overline{B}$.
   Hence $g : \overline{A} \leq_m \overline{B}$.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

   $g : B \leq_m A$; then $c_B(x) = c_A(g(x))$. So c_B is computable.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

   $g : B \leq_m A$; then $c_B(x) = c_A(g(x))$. So c_B is computable.

5. If A is recursive and $B \neq \varnothing, \mathbb{N}$, then $A \leq_m B$.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

   $g : B \leq_m A$; then $c_B(x) = c_A(g(x))$. So c_B is computable.

5. If A is recursive and $B \neq \varnothing, \mathbb{N}$, then $A \leq_m B$.

   Let $b \in B$, $c \notin B$, $f(x) = \begin{cases} b & \text{if } x \in A; \\ c & \text{if } x \notin A. \end{cases}$; then f is computable.

   $x \in A \Leftrightarrow f(x) \in B$.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

   $g : B \leq_m A$; then $c_B(x) = c_A(g(x))$. So c_B is computable.

5. If A is recursive and $B \neq \varnothing, \mathbb{N}$, then $A \leq_m B$.

   Let $b \in B$, $c \notin B$, $f(x) = \begin{cases} b & \text{if } x \in A; \\ c & \text{if } x \notin A. \end{cases}$; then f is computable.

   $x \in A \Leftrightarrow f(x) \in B$.

6. If A is r.e. and $B \leq_m A$, then B is r.e.

Elementary Properties (2)

4. If A is recursive and $B \leq_m A$, then B is recursive.

   $g : B \leq_m A$; then $c_B(x) = c_A(g(x))$. So c_B is computable.

5. If A is recursive and $B \neq \varnothing, \mathbb{N}$, then $A \leq_m B$.

   Let $b \in B$, $c \notin B$, $f(x) = \begin{cases} b & \text{if } x \in A; \\ c & \text{if } x \notin A. \end{cases}$; then f is computable.

   $x \in A \Leftrightarrow f(x) \in B$.

6. If A is r.e. and $B \leq_m A$, then B is r.e.

   Let $g : B \leq_m A$, $A = Dom(h)$, $(h \in \mathscr{C}_1)$; then $B = Dom(h \circ g)$ (B is r.e.)

Elementary Properties (3)

7. (i). $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; (ii). $A \leq_m \varnothing$ iff $A = \varnothing$.

Elementary Properties (3)

7. (i). $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; (ii). $A \leq_m \varnothing$ iff $A = \varnothing$.

   (i)."$\Leftarrow$": By reflexivity, $\mathbb{N} \leq_m \mathbb{N}$.
   (i)."$\Rightarrow$": Let $f : A \leq_m \mathbb{N}$, then $x \in A \Leftrightarrow f(x) \in \mathbb{N}$. Thus $A = \mathbb{N}$.
   (ii). $A \leq_m \varnothing \Leftrightarrow \overline{A} \leq_m \mathbb{N} \Leftrightarrow \overline{A} = \mathbb{N} \Leftrightarrow A = \varnothing$.

Elementary Properties (3)

7. (i). $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; (ii). $A \leq_m \varnothing$ iff $A = \varnothing$.

   (i).“$\Leftarrow$": By reflexivity, $\mathbb{N} \leq_m \mathbb{N}$.
   (i).“$\Rightarrow$": Let $f : A \leq_m \mathbb{N}$, then $x \in A \Leftrightarrow f(x) \in \mathbb{N}$. Thus $A = \mathbb{N}$.
   (ii). $A \leq_m \varnothing \Leftrightarrow \overline{A} \leq_m \mathbb{N} \Leftrightarrow \overline{A} = \mathbb{N} \Leftrightarrow A = \varnothing$.

8. (i). $\mathbb{N} \leq_m A$ iff $A \neq \varnothing$; (ii). $\varnothing \leq_m A$ iff $A \neq \mathbb{N}$.

Elementary Properties (3)

7. (i). $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; (ii). $A \leq_m \varnothing$ iff $A = \varnothing$.

   (i).“$\Leftarrow$": By reflexivity, $\mathbb{N} \leq_m \mathbb{N}$.
   (i).“$\Rightarrow$": Let $f : A \leq_m \mathbb{N}$, then $x \in A \Leftrightarrow f(x) \in \mathbb{N}$. Thus $A = \mathbb{N}$.
   (ii). $A \leq_m \varnothing \Leftrightarrow \overline{A} \leq_m \mathbb{N} \Leftrightarrow \overline{A} = \mathbb{N} \Leftrightarrow A = \varnothing$.

8. (i). $\mathbb{N} \leq_m A$ iff $A \neq \varnothing$; (ii). $\varnothing \leq_m A$ iff $A \neq \mathbb{N}$.

   (i). “$\Rightarrow$": Let $f : \mathbb{N} \leq_m A$, then $A = Ran(f)$, so $A \neq \varnothing$ (f is total).
   (i). “$\Leftarrow$": If $A \neq \varnothing$, choose $c \in A$. If $g(x) = c$, we have $g : \mathbb{N} \leq_m A$.
   (ii). $\varnothing \leq_m A \Leftrightarrow \mathbb{N} \leq_m \overline{A} \Leftrightarrow \overline{A} \neq \varnothing \Leftrightarrow A = \mathbb{N}$.

Corollary

Corollary. Neither $\{x \mid \phi_x \text{ is total}\}$ nor $\{x \mid \phi_x \text{ is not total}\}$ is m-reducible to K.

Corollary

Corollary. Neither $\{x \mid \phi_x \text{ is total}\}$ nor $\{x \mid \phi_x \text{ is not total}\}$ is m-reducible to K.

Proof. By contradiction, if $\{x \mid \phi_x \text{ is total}\} \leq_m K$, and K is r.e., then $\{x \mid \phi_x \text{ is total}\}$ is r.e. (same as $\{x \mid \phi_x \text{ is not total}\}$).

However, by Rice-Shapiro Theorem, Neither $\{x \mid \phi_x \text{ is total}\}$ nor $\{x \mid \phi_x \text{ is not total}\}$ is r.e.

Corollary (2)

Fact. If A is r.e. and is not recursive, then $\overline{A} \not\leq_m A$ and $A \not\leq_m \overline{A}$.

Corollary (2)

Fact. If A is r.e. and is not recursive, then $\overline{A} \not\leq_m A$ and $A \not\leq_m \overline{A}$.

Proof. "$\overline{A} \not\leq_m A$": By contradiction, if $\overline{A} \leq_m A$, then $\overline{A}$ is r.e., then A is recursive!

"$A \not\leq_m \overline{A}$": By contradiction, if $A \leq_m \overline{A}$, then $\overline{A} \leq_m A$, then A is recursive!

Corollary (2)

Fact. If A is r.e. and is not recursive, then $\overline{A} \not\leq_m A$ and $A \not\leq_m \overline{A}$.

Proof. "$\overline{A} \not\leq_m A$": By contradiction, if $\overline{A} \leq_m A$, then $\overline{A}$ is r.e., then A is recursive!

"$A \not\leq_m \overline{A}$": By contradiction, if $A \leq_m \overline{A}$, then $\overline{A} \leq_m A$, then A is recursive!

Notation: It contradicts to our intuition that A and $\overline{A}$ are equally difficult.

Theorem

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

Theorem

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

Proof. "$\Leftarrow$". Since $A \leq_m K$, and K is r.e., then A is r.e.

Suppose A is r.e. Let $f(x,y)$ be $f(x,y) = \begin{cases} 1, & \text{if } x \in A, \\ \uparrow, & \text{if } x \notin A. \end{cases}$

By s-m-n Theorem $\exists s(x) : \mathbb{N} \to \mathbb{N}$ such that $f(x,y) = \phi_{s(x)}(y)$.

It is clear that $x \in A$ iff $\phi_{s(x)}(s(x))$ is defined iff $s(x) \in K$. Hence $A \leq_m K$.

Theorem

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

Proof. "$\Leftarrow$". Since $A \leq_m K$, and K is r.e., then A is r.e.

Suppose A is r.e. Let $f(x,y)$ be $f(x,y) = \begin{cases} 1, & \text{if } x \in A, \\ \uparrow, & \text{if } x \notin A. \end{cases}$

By s-m-n Theorem $\exists s(x) : \mathbb{N} \to \mathbb{N}$ such that $f(x,y) = \phi_{s(x)}(y)$.

It is clear that $x \in A$ iff $\phi_{s(x)}(s(x))$ is defined iff $s(x) \in K$. Hence $A \leq_m K$.

Notation. K is the most difficult partially decidable problem.

Many-One Equivalence

Definition. Two sets A, B are many-one equivalent, notation $A \equiv_m B$ (abbreviated m-equivalent), if $A \leq_m B$ and $B \leq_m A$.

Many-One Equivalence

Definition. Two sets A, B are many-one equivalent, notation $A \equiv_m B$ (abbreviated m-equivalent), if $A \leq_m B$ and $B \leq_m A$.

Theorem. $\equiv_m$ is an equivalence relation.

Many-One Equivalence

Definition. Two sets A, B are many-one equivalent, notation $A \equiv_m B$ (abbreviated m-equivalent), if $A \leq_m B$ and $B \leq_m A$.

Theorem. $\equiv_m$ is an equivalence relation.

Proof.
(1). Reflexivity: $A \leq_m A \Rightarrow A \equiv_m A$.
(2). Symmetry: $A \equiv_m B \Rightarrow B \leq_m A,\ A \leq_m B \Rightarrow B \equiv_m A$.
(3). Transitivity: $A \equiv_m B,\ B \equiv_m C \Rightarrow A \leq_m C,\ C \leq_m A \Rightarrow A \equiv_m C$.

Examples

1. $\{x \mid c \in W_x\} \equiv_m K$.

Examples

1. $\{x \mid c \in W_x\} \equiv_m K$.

"$\Leftarrow$": $f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \Rightarrow K \leq_m \{x \mid c \in W_x\}$

"$\Rightarrow$": $\{x \mid c \in W_x\}$ is r.e., so $\{x \mid c \in W_x\} \leq_m K$.

Thus $\{x \mid c \in W_x\} \equiv_m K$.

Examples

1. $\{x \mid c \in W_x\} \equiv_m K$.

   "$\Leftarrow$": $f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \Rightarrow K \leq_m \{x \mid c \in W_x\}$

   "$\Rightarrow$": $\{x \mid c \in W_x\}$ is r.e., so $\{x \mid c \in W_x\} \leq_m K$.

   Thus $\{x \mid c \in W_x\} \equiv_m K$.

2. If A is recursive, $A \neq \varnothing, \mathbb{N}$, then $A \equiv_m \overline{A}$.

Examples

1. $\{x \mid c \in W_x\} \equiv_m K$.

   "$\Leftarrow$": $f_{\mathbb{N}}(x,y) = \begin{cases} y & \text{if } x \in W_x \\ \uparrow & \text{if } x \notin W_x \end{cases} \Rightarrow K \leq_m \{x \mid c \in W_x\}$

   "$\Rightarrow$": $\{x \mid c \in W_x\}$ is r.e., so $\{x \mid c \in W_x\} \leq_m K$.

   Thus $\{x \mid c \in W_x\} \equiv_m K$.

2. If A is recursive, $A \neq \varnothing, \mathbb{N}$, then $A \equiv_m \overline{A}$.

   $A \neq \varnothing, \mathbb{N} \Rightarrow \overline{A} \neq \varnothing, \mathbb{N}$.

   A is recursive, by previous theorem $A \leq_m \overline{A}$. Similarly, $\overline{A} \leq_m A$.

Example (2)

3. If A is r.e. but not recursive, then $A \not\equiv_m \overline{A}$.

Example (2)

3. If A is r.e. but not recursive, then $A \not\equiv_m \overline{A}$.

A is r.e. but not recursive $\Rightarrow A \not\leq_m \overline{A}, \overline{A} \not\leq_m A$.

Example (2)

3. If A is r.e. but not recursive, then $A \not\equiv_m \overline{A}$.

   A is r.e. but not recursive $\Rightarrow A \not\leq_m \overline{A}, \overline{A} \not\leq_m A$.

4. $\{x \mid \phi_x = \mathbf{0}\} \equiv_m \{x \mid \phi_x \text{ is total}\}$.

Example (2)

3. If A is r.e. but not recursive, then $A \not\equiv_m \overline{A}$.

   A is r.e. but not recursive $\Rightarrow A \not\leq_m \overline{A}, \overline{A} \not\leq_m A$.

4. $\{x \mid \phi_x = \mathbf{0}\} \equiv_m \{x \mid \phi_x \text{ is total}\}$.

   "$\Leftarrow$": $\phi_{k(x)} = \mathbf{0} \circ \phi_x \Rightarrow \{x \mid \phi_x \text{ is total}\} \leq_m \{x \mid \phi_x = \mathbf{0}\}$.

   "$\Rightarrow$": Let $\phi_{k(x)}(y) = \begin{cases} 0 & \text{if } \phi_x(y) = 0; \\ \uparrow & \text{if } \phi_x(y) \neq 0. \end{cases}$.

   Then $\phi_x = \mathbf{0} \Leftrightarrow \phi_{k(x)}$ is total $\Rightarrow \{x \mid \phi_x = \mathbf{0}\} \leq_m \{x \mid \phi_x \text{ is total}\}$.

m-Degree

Definition. Let $d_m(A)$ be $\{B \mid A \equiv_m B\}$.

m-Degree

Definition. Let $d_m(A)$ be $\{B \mid A \equiv_m B\}$.

Definition. An m-degree is an equivalence class of sets under the relation $\equiv_m$. It is any class of sets of the form $d_m(A)$ for some set A.

m-Degree

Definition. Let $d_m(A)$ be $\{B \mid A \equiv_m B\}$.

Definition. An m-degree is an equivalence class of sets under the relation $\equiv_m$. It is any class of sets of the form $d_m(A)$ for some set A.

A recursive m-degree is an m-degree that contains a recursive set.
An r.e. m-degree is an m-degree that contains an r.e. set.

Expression

Definition. The set of *m*-degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

Expression

Definition. The set of m-degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

Definition (Partial Order on m-Degree). Let $\mathbf{a}$, $\mathbf{b}$ be m-degrees.

(1). $\mathbf{a} \leq_m \mathbf{b}$ iff $A \leq_m B$ for some $A \in \mathbf{a}$ and $B \in \mathbf{b}$.

(2). $\mathbf{a} <_m \mathbf{b}$ iff $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b} \not\leq_m \mathbf{a}$ ($\mathbf{a} \neq \mathbf{b}$).

The relation $<_m$ is a partial order.

Expression

Definition. The set of m-degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

Definition (Partial Order on m-Degree). Let $\mathbf{a}$, $\mathbf{b}$ be m-degrees.

(1). $\mathbf{a} \leq_m \mathbf{b}$ iff $A \leq_m B$ for some $A \in \mathbf{a}$ and $B \in \mathbf{b}$.

(2). $\mathbf{a} <_m \mathbf{b}$ iff $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b} \not\leq_m \mathbf{a}$ ($\mathbf{a} \neq \mathbf{b}$).

The relation $<_m$ is a partial order.

Notation. From the definition of $\equiv_m$,
$\mathbf{a} \leq_m \mathbf{b} \Leftrightarrow \forall A \in \mathbf{a}, B \in \mathbf{b}, A \leq_m B$.

Theorem

Theorem. The relation $<_m$ is a partial ordering of m-degrees.

Theorem

Theorem. The relation $<_m$ is a partial ordering of m-degrees.

Proof.

(1) By transitivity $\mathbf{a} \leq_m \mathbf{b}$, $\mathbf{b} \leq_m \mathbf{c}$ implies $\mathbf{a} \leq_m \mathbf{c}$.

If $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b} \leq_m \mathbf{a}$, we have to prove that $\mathbf{a} = \mathbf{b}$.

(2) Irreflexivity: Let $A \in \mathbf{a}$ and $B \in \mathbf{b}$, then we have $A \leq_m B$ and $B \leq_m A$, so $A \equiv_m B$. Hence $\mathbf{a} = \mathbf{b}$.

Consequently, $<_m$ is partial ordering.

Some Facts

1. **o** and **n** are respectively the recursive m-degrees $\{\emptyset\}$ and $\{\mathbb{N}\}$.
   $A \leq_m \mathbf{N} \Leftrightarrow A = \mathbf{N}$; $A \leq_m \emptyset \Leftrightarrow A = \emptyset$.

Some Facts

1. **o** and **n** are respectively the recursive m-degrees $\{\varnothing\}$ and $\{\mathbb{N}\}$.

   $A \leq_m \mathbf{N} \Leftrightarrow A = \mathbf{N}$; $A \leq_m \varnothing \Leftrightarrow A = \varnothing$.

2. The recursive m-degree $\mathbf{0}_m$ consists of all the recursive sets except $\varnothing, \mathbb{N}$.

   $\mathbf{0}_m \leq_m \mathbf{a}$ for any m-degree **a** other than **o** and **n**.

   A is recursive, $B \leq_m A \Rightarrow B$ is recursive;

   A is recursive and $B \neq \varnothing, \mathbb{N} \Rightarrow A \leq_m B$.

Some Facts

1. **o** and **n** are respectively the recursive m-degrees $\{\emptyset\}$ and $\{\mathbb{N}\}$.

   $A \leq_m \mathbf{N} \Leftrightarrow A = \mathbf{N}$; $A \leq_m \emptyset \Leftrightarrow A = \emptyset$.

2. The recursive m-degree $\mathbf{0}_m$ consists of all the recursive sets except $\emptyset, \mathbb{N}$.

   $\mathbf{0}_m \leq_m \mathbf{a}$ for any m-degree $\mathbf{a}$ other than **o** and **n**.

   A is recursive, $B \leq_m A \Rightarrow B$ is recursive;

   A is recursive and $B \neq \emptyset, \mathbb{N} \Rightarrow A \leq_m B$.

3. $\forall$ m-degree **a**, $\mathbf{o} \leq_m \mathbf{a}$ provided $\mathbf{a} \neq \mathbf{n}$; $\mathbf{n} \leq_m \mathbf{a}$ provided $\mathbf{a} \neq \mathbf{o}$.

   $\mathbb{N} \leq_m A \Leftrightarrow A \neq \emptyset$; $\emptyset \leq_m A \Leftrightarrow A \neq \mathbb{N}$.

Facts (2)

4. An r.e. m-degree consists of only r.e. sets.

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

Facts (2)

4. An r.e. m-degree consists of only r.e. sets.

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

5. If $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b}$ is an r.e. m-degree, then $\mathbf{a}$ is also an r.e. m-degree.

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

Facts (2)

4. An r.e. m-degree consists of only r.e. sets.

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

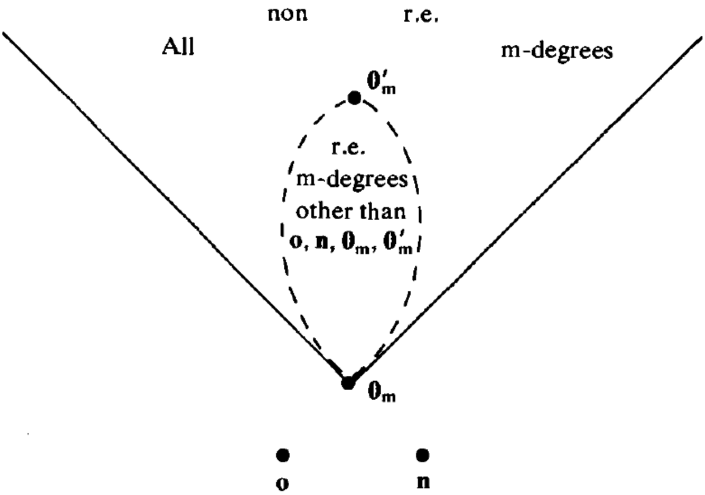
5. If $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b}$ is an r.e. m-degree, then $\mathbf{a}$ is also an r.e. m-degree.

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

6. The maximum r.e. m-degree $d_m(K)$ is denoted by $\mathbf{0}'_m$.

   A set A is r.e. iff $A \leq_m K$.

Illumination



Facts about r.e. *m*-Degrees

1. Excluding **o** and **n**, there is a minimum r.e. m-degree $\mathbf{0}_m$ (in fact $\mathbf{0}_m$ is minimum among all m-degrees).

Facts about r.e. m-Degrees

1. Excluding $\mathbf{o}$ and $\mathbf{n}$, there is a minimum r.e. m-degree $\mathbf{0}_m$ (in fact $\mathbf{0}_m$ is minimum among all m-degrees).
2. The r.e. m-degrees form an initial segment of the m-degrees; i.e., anything below an r.e. m-degree is also r.e.

Facts about r.e. m-Degrees

1. Excluding **o** and **n**, there is a minimum r.e. m-degree $\mathbf{0}_m$ (in fact $\mathbf{0}_m$ is minimum among all m-degrees).

2. The r.e. m-degrees form an initial segment of the m-degrees; i.e., anything below an r.e. m-degree is also r.e.

3. There is a maximum r.e. m-degree $\mathbf{0}'_m$.

Facts about r.e. m-Degrees

1. Excluding $\mathbf{o}$ and $\mathbf{n}$, there is a minimum r.e. m-degree $\mathbf{0}_m$ (in fact $\mathbf{0}_m$ is minimum among all m-degrees).
2. The r.e. m-degrees form an initial segment of the m-degrees; i.e., anything below an r.e. m-degree is also r.e.
3. There is a maximum r.e. m-degree $\mathbf{0}'_m$.
4. While there are uncountably many m-degrees, only countably many of these are r.e.

Algebraic Structure

Theorem. The *m*-degrees form an upper semi-lattice.

Group

In mathematics, a group is an algebraic structure consisting of a set together with an operation $(G, \bullet)$ that combines any two of its elements to form a third element.

Group

In mathematics, a group is an algebraic structure consisting of a set together with an operation $(G, \bullet)$ that combines any two of its elements to form a third element.

To qualify as a group, the set and the operation must satisfy four conditions (group axioms), namely closure, associativity, identity and invertibility.

Group

In mathematics, a group is an algebraic structure consisting of a set together with an operation $(G, \bullet)$ that combines any two of its elements to form a third element.

To qualify as a group, the set and the operation must satisfy four conditions (group axioms), namely closure, associativity, identity and invertibility.

closure: $a, b \in G \Rightarrow a \bullet b \in G$.

associativity: $(a \bullet b) \bullet c = a \bullet (b \bullet c)$.

identity: $\forall a \in G$, $\exists$ identity element $e \in G$, s.t. $e \bullet a = a \bullet e = a$.

invertibility: $\forall a \in G$, $\exists$inverse $b \in G$ s.t. $a \bullet b = b \bullet a = e$ $(b = a^{-1})$.

Lattice

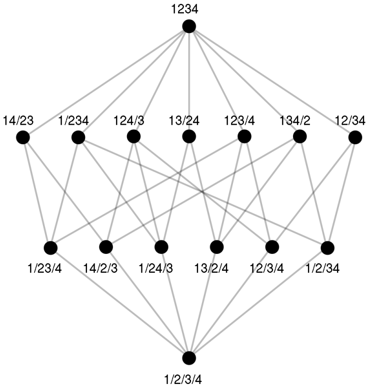
In mathematics, a lattice is a partially ordered set (poset) $(L, \leq)$ in which any two elements have a unique supremum (also called a least upper bound or join) and a unique infimum (also called a greatest lower bound or meet).

To qualify as a lattice, the set and the operation must satisfy tow conditions: join-semilattice, meet-semilattice.

join-semilattice:	$\forall a, b \in L$, the set $\{a, b\}$ has a join $a \vee b$. (the least upper bound)
meet-semilattice:	$\forall a, b \in L$, the set $\{a, b\}$ has a meet $a \wedge b$. (the greatest lower bound)

The Name “Lattice"

The name "lattice" is suggested by the form of the Hasse diagram depicting it. I.e., the right picture is the lattice of partitions of a four-element set $\{1, 2, 3, 4\}$, ordered by the relation "is a refinement of".



Upper Semi-lattice

Theorem. Any pair of m-degrees **a**, **b** have a least upper bound; i.e. there is an m-degree **c** such that

(i). $\mathbf{a} \leq_m \mathbf{c}$ and $\mathbf{b} \leq_m \mathbf{c}$ (**c** is an upper bound);

(ii). $\mathbf{c} \leq_m$ any other upper bound of **a**, **b**.

Proof

(i). Pick $A \in \mathbf{a}$, $B \in \mathbf{b}$, and let $C = A \oplus B$, i.e.,

$$C = \{2x \mid x \in A\} \cup \{2x+1 \mid x \in B\}.$$

Then

$$x \in A \Leftrightarrow 2x \in C \Longrightarrow A \leq_m C;$$

$$x \in B \Leftrightarrow 2x+1 \in C \Longrightarrow B \leq_m C;$$

Thus $\mathbf{c}$ is an upper bound of $\mathbf{a}$, $\mathbf{b}$.

Proof (2)

(ii). Let $\mathbf{d}$ is an m-degree such that $\mathbf{a} \leq_m \mathbf{d}$, and $\mathbf{b} \leq_m \mathbf{d}$.

$\forall D \in \mathbf{d}$, suppose $f : A \leq_m D$ and $g : B \leq_m D$. Then

$$\begin{aligned} x \in C &\Leftrightarrow (x \text{ is even } \& \frac{x}{2} \in A) \vee (x \text{ is odd } \& \frac{x-1}{2} \in B) \\ &\Leftrightarrow (x \text{ is even } \& f(\frac{x}{2}) \in D) \vee (x \text{ is odd } \& g(\frac{x-1}{2}) \in D) \end{aligned}$$

Thus we have $h : C \leq_m D$ if we define $h = \begin{cases} f(\frac{x}{2}) & \text{if } x \text{ is even;} \\ g(\frac{x-1}{2}) & \text{if } x \text{ is odd.} \end{cases}$

Hence $\mathbf{c} \leq_m \mathbf{d}$.

Definition

Definition. An r.e. set is m-complete if every r.e. set is m-reducible to it.

Definition

Definition. An r.e. set is m-complete if every r.e. set is m-reducible to it.

Notation. $\mathbf{0}'_m$, the m-degree of K is maximum among all r.e. m-degrees, and thus K is m-complete r.e. set (or just called m-complete set).

Theorem

Theorem. The following statements are valid.

(i) K is m-complete.

(ii) A is m-complete iff $A \equiv_m K$ iff A is r.e. and $K \leq_m A$.

(iii) $\mathbf{0}'_m$ consists exactly of all the m-complete sets.

Examples

The following sets are m-complete.

Examples

The following sets are m-complete.

(i) $\{x \mid c \in W_x\}$.

Examples

The following sets are m-complete.

(i) $\{x \mid c \in W_x\}$.

(ii) Every non-trivial r.e. set of the form $\{x \mid \phi_x \in \mathscr{B}\}$.

Examples

The following sets are m-complete.

(i) $\{x \mid c \in W_x\}$.

(ii) Every non-trivial r.e. set of the form $\{x \mid \phi_x \in \mathscr{B}\}$.

(iii) $\{x \mid \phi_x(x) = 0\}$.

Examples

The following sets are m-complete.

(i) $\{x \mid c \in W_x\}$.

(ii) Every non-trivial r.e. set of the form $\{x \mid \phi_x \in \mathscr{B}\}$.

(iii) $\{x \mid \phi_x(x) = 0\}$.

(iv). $\{x \mid x \in E_x\}$.

Creative Set

Theorem. Any m-complete set is creative.

Creative Set

Theorem. Any m-complete set is creative.

Proof. If A is m-complete, A is r.e. set.

Also, $K \leq_m A$, so $\overline{K} \leq_m \overline{A}$. Thus $\overline{A}$ is productive.

Myhill's Theorem

Myhill's Theorem. A set is m-complete iff it is creative.

m-Complete r.e. Sets

Corollary. If $\mathbf{a}$ is the m-degree of any simple set, then $\mathbf{0}_m <_m \mathbf{a} <_m \mathbf{0}'_m$ (Simple sets are not m-complete).

m-Complete r.e. Sets

Corollary. If $\mathbf{a}$ is the m-degree of any simple set, then $\mathbf{0}_m <_m \mathbf{a} <_m \mathbf{0}'_m$ (Simple sets are not m-complete).

Proof. Simple sets are designed to be neither recursive nor creative.

Outline

Comparison

m-reducibility has two unsatisfactory features:

Comparison

m-reducibility has two unsatisfactory features:

(i) The exceptional behavior of $\varnothing$ and $\mathbb{N}$.

Comparison

m-reducibility has two unsatisfactory features:

(i) The exceptional behavior of $\varnothing$ and $\mathbb{N}$.

(ii) The invalidity of $A \equiv_m \overline{A}$ in general.

Comparison

m-reducibility has two unsatisfactory features:

(i) The exceptional behavior of $\varnothing$ and $\mathbb{N}$.

(ii) The invalidity of $A \not\equiv_m \overline{A}$ in general.

The problem is due to the restricted use of oracles.

E.g. $x \in \overline{A}$ iff $x \notin A$

Relative Computability

Suppose χ is a total unary function.

Relative Computability

Suppose χ is a total unary function.

Informally a function f is computable relative to χ, or χ-computable, if f can be computed by an algorithm that is effective in the usual sense, except from time to time during computations f is allowed to consult the oracle function χ.

Relative Computability

Suppose χ is a total unary function.

Informally a function f is computable relative to χ, or χ-computable, if f can be computed by an algorithm that is effective in the usual sense, except from time to time during computations f is allowed to consult the oracle function χ.

Such an algorithm is called a χ-algorithm.

URMO - Unlimited Register Machine with Oracle

A URM with oracle, URMO for short, can recognize a fifth kind of instruction, $O(n)$, for every $n \geq 1$.

URMO - Unlimited Register Machine with Oracle

A URM with oracle, URMO for short, can recognize a fifth kind of instruction, $O(n)$, for every $n \geq 1$.

If χ is the oracle, then the effect of $O(n)$ is to replace the content r_n of R_n by $\chi(r_n)$.

URMO - Unlimited Register Machine with Oracle

A URM with oracle, URMO for short, can recognize a fifth kind of instruction, $O(n)$, for every $n \geq 1$.

If χ is the oracle, then the effect of $O(n)$ is to replace the content r_n of R_n by $\chi(r_n)$.

P^{χ} denote the program P when used with the function χ in the oracle.

URMO - Unlimited Register Machine with Oracle

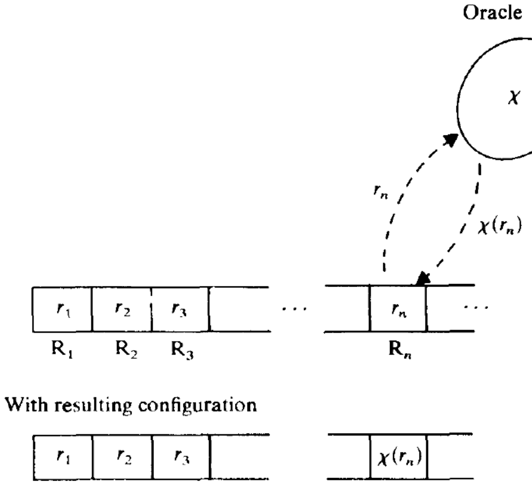
A URM with oracle, URMO for short, can recognize a fifth kind of instruction, $O(n)$, for every $n \geq 1$.

If χ is the oracle, then the effect of $O(n)$ is to replace the content r_n of R_n by $\chi(r_n)$.

P^{χ} denote the program P when used with the function χ in the oracle.

$P^{\chi}(\mathbf{a}) \downarrow b$ means the computation $P^{\chi}(\mathbf{a})$ with initial configuration $a_1, a_2, \cdots, a_n, 0, 0, \cdots$ stops with the number b is register R_1.

Illumination



URMO-Computable

Let χ be a unary total function, and suppose the f is a partial function from $\mathbb{N}^n$ to $\mathbb{N}$.

(a) Let P be a URMO program, then P URMO-computes f relative to χ (or f is χ-computed by P) if, for every $\mathbf{a} \in \mathbb{N}^n$ and $b \in \mathbb{N}$, $P^\chi(\mathbf{a}) \downarrow b$ iff $f(\mathbf{a}) \simeq b$.

(b) The function f is URMO-computable relative to χ (or χ-computable) if there is a URMO program that URMO-computes it relative to χ.

URMO-Computable

Let χ be a unary total function, and suppose the f is a partial function from $\mathbb{N}^n$ to $\mathbb{N}$.

(a) Let P be a URMO program, then P URMO-computes f relative to χ (or f is χ-computed by P) if, for every $\mathbf{a} \in \mathbb{N}^n$ and $b \in \mathbb{N}$, $P^{\chi}(\mathbf{a}) \downarrow b$ iff $f(\mathbf{a}) \simeq b$.

(b) The function f is URMO-computable relative to χ (or χ-computable) if there is a URMO program that URMO-computes it relative to χ.

$\mathscr{C}^{\chi}$ is the set of all χ-computable functions.

Facts

(i) $\chi \in \mathscr{C}^{\chi}$.

Use URMO program $O(1)$.

Facts

(i) $\chi \in \mathscr{C}^{\chi}$.

Use URMO program $O(1)$.

(ii) $\mathscr{C} \subseteq \mathscr{C}^{\chi}$.

Any URM program is a URMO program.

Facts

(i) $\chi \in \mathscr{C}^{\chi}$.

Use URMO program $O(1)$.

(ii) $\mathscr{C} \subseteq \mathscr{C}^{\chi}$.

Any URM program is a URMO program.

(iii) If χ is computable, then $\mathscr{C} = \mathscr{C}^{\chi}$.

Since $\mathscr{C} \subseteq \mathscr{C}^{\chi}$, we need to prove $\mathscr{C}^{\chi} \subseteq \mathscr{C}$. χ is computable, then whenever a value of χ is requested simply compute it by the algorithm for χ. By Church's thesis, f is computable.

Facts

(i) $\chi \in \mathscr{C}^{\chi}$.

Use URMO program $O(1)$.

(ii) $\mathscr{C} \subseteq \mathscr{C}^{\chi}$.

Any URM program is a URMO program.

(iii) If χ is computable, then $\mathscr{C} = \mathscr{C}^{\chi}$.

Since $\mathscr{C} \subseteq \mathscr{C}^{\chi}$, we need to prove $\mathscr{C}^{\chi} \subseteq \mathscr{C}$. χ is computable, then whenever a value of χ is requested simply compute it by the algorithm for χ. By Church's thesis, f is computable.

(iv) $\mathscr{C}^{\chi}$ is closed under substitution, recursion and minimalisation.

Construct corresponding URMO programs.

Facts

(i) $\chi \in \mathscr{C}^{\chi}$.

Use URMO program $O(1)$.

(ii) $\mathscr{C} \subseteq \mathscr{C}^{\chi}$.

Any URM program is a URMO program.

(iii) If χ is computable, then $\mathscr{C} = \mathscr{C}^{\chi}$.

Since $\mathscr{C} \subseteq \mathscr{C}^{\chi}$, we need to prove $\mathscr{C}^{\chi} \subseteq \mathscr{C}$. χ is computable, then whenever a value of χ is requested simply compute it by the algorithm for χ. By Church's thesis, f is computable.

(iv) $\mathscr{C}^{\chi}$ is closed under substitution, recursion and minimalisation.

Construct corresponding URMO programs.

(v) If ψ is a total unary function that is χ-computable, then $\mathscr{C}^{\psi} \subseteq \mathscr{C}^{\chi}$.

By Church's thesis.

Partial Recursive Function

The class $\mathscr{R}^\chi$ of χ-partial recursive functions is the smallest class of functions such that

(a) the basic functions are in $\mathscr{R}^\chi$.

(b) $\chi \in \mathscr{R}^\chi$.

(c) $\mathscr{R}^\chi$ is closed under substitution, recursion, and minimalisation.

Partial Recursive Function

The class $\mathscr{R}^{\chi}$ of χ-partial recursive functions is the smallest class of functions such that

(a) the basic functions are in $\mathscr{R}^{\chi}$.

(b) $\chi \in \mathscr{R}^{\chi}$.

(c) $\mathscr{R}^{\chi}$ is closed under substitution, recursion, and minimalisation.

χ-recursive, χ-primitive recursive are defined in the obvious way.

Partial Recursive Function

The class $\mathscr{R}^\chi$ of χ-partial recursive functions is the smallest class of functions such that

(a) the basic functions are in $\mathscr{R}^\chi$.

(b) $\chi \in \mathscr{R}^\chi$.

(c) $\mathscr{R}^\chi$ is closed under substitution, recursion, and minimalisation.

χ-recursive, χ-primitive recursive are defined in the obvious way.

Theorem. For any χ, $\mathscr{R}^\chi = \mathscr{C}^\chi$.

Numbering URMO programs

Let's fix an effective enumeration of all URMO programs

$$Q_0, Q_1, Q_2, \ldots.$$

Numbering URMO programs

Let's fix an effective enumeration of all URMO programs

$$Q_0, Q_1, Q_2, \ldots.$$

Let $\phi_m^{\chi,n}$ be the n-ary function χ-computed by Q_m.

Let ϕ_m^{χ} be $\phi_m^{\chi,1}$.

Numbering URMO programs

Let's fix an effective enumeration of all URMO programs

$$Q_0, Q_1, Q_2, \ldots.$$

Let $\phi_m^{\chi,n}$ be the n-ary function χ-computed by Q_m.

Let ϕ_m^{χ} be $\phi_m^{\chi,1}$.

W_m^{χ} is $Dom(\phi_m^{\chi})$ and E_m^{χ} is $Ran(\phi_m^{\chi})$.

Numbering URMO programs

S-m-n Theorem. For each $m, n \geq 1$ there is a total computable $(m+1)$-ary function $s_n^m(e, \mathbf{x})$ such that for any χ

$$\phi_e^{\chi, m+n}(\mathbf{x}, \mathbf{y}) \simeq \phi_{s_n^m(e,\mathbf{x})}^{\chi, n}(\mathbf{y}).$$

Numbering URMO programs

S-m-n Theorem. For each $m, n \geq 1$ there is a total computable $(m+1)$-ary function $s_n^m(e, \mathbf{x})$ such that for any χ

$$\phi_e^{\chi, m+n}(\mathbf{x}, \mathbf{y}) \simeq \phi_{s_n^m(e,\mathbf{x})}^{\chi, n}(\mathbf{y}).$$

Notice that $s_n^m(e, \mathbf{x})$ does not refer to χ.

Universal Programs for Relative Computability

Universal Function Theorem. For each n, the universal function $\psi_U^{\chi,n}$ for n-ary χ-computable functions given by

$$\psi_U^{\chi,n}(e,\mathbf{x}) \simeq \phi_e^{\chi,n}(\mathbf{x})$$

is χ-computable.

Relativization

Once we have the S-m-n Theorem and the Universal Function Theorem, we can do the recursion theory relative to an oracle.

χ-Recursive and χ-r.e. Sets

Let A be a set

(a) A is χ-recursive if c_A is χ-computable.

(b) A is χ-r.e. if the partial characteristic function
$$f(x) = \begin{cases} 1 & \text{if } x \in A, \\ \uparrow & \text{if } x \notin A \end{cases}$$ is χ-computable.

χ-Recursive and χ-r.e. Sets

Theorem. The following statements are valid.

(i) For any set A, A is χ-recursive iff A and $\overline{A}$ are χ-r.e.

χ-Recursive and χ-r.e. Sets

Theorem. The following statements are valid.

(i) For any set A, A is χ-recursive iff A and $\overline{A}$ are χ-r.e.

(ii) For any set A, the following are equivalent.

- A is χ-r.e.
- $A = W_m^\chi$ for some m.
- $A = E_m^\chi$ for some m.
- $A = \emptyset$ or A is the range of a total χ-computable function.
- For some χ-decidable predicate $R(x, y)$, $x \in A$ iff $\exists y.R(x, y)$.

χ-Recursive and χ-r.e. Sets

Theorem. The following statements are valid.

(i) For any set A, A is χ-recursive iff A and $\overline{A}$ are χ-r.e.

(ii) For any set A, the following are equivalent.

- A is χ-r.e.
- $A = W_m^\chi$ for some m.
- $A = E_m^\chi$ for some m.
- $A = \emptyset$ or A is the range of a total χ-computable function.
- For some χ-decidable predicate $R(x, y)$, $x \in A$ iff $\exists y.R(x, y)$.

(iii) $K^\chi \overset{\text{def}}{=} \{x \mid x \in W_x^\chi\}$ is χ-r.e. but not χ-recursive.

Computability Relative to a Set

Computability relative to a set A means computability relative to its characteristic function c_A.

Computability Relative to a Set

Computability relative to a set A means computability relative to its characteristic function c_A.

For example:

P^A for P^{c_A} (if P is a URMO program),

$\mathscr{C}^A$ for $\mathscr{C}^{c_A}$,

ϕ_m^A for $\phi_m^{c_A}$.

W_m^A for $W_m^{c_A}$,

E_m^A for $E_m^{c_A}$,

K^A for K^{c_A},

A-recursive for c_A-recursive

A-r.e. for c_A-r.e.

...

Outline

Turing Reducibility and Turing Degrees

The set A is Turing reducible to B, notation $A \leq_T B$, if A has a B-computable characteristic function c_A.

Turing Reducibility and Turing Degrees

The set A is Turing reducible to B, notation $A \leq_T B$, if A has a B-computable characteristic function c_A.

The sets A, B are Turing equivalent, notation $A \equiv_T B$, if $A \leq_T B$ and $B \leq_T A$.

Notation

Suppose $A \leq_T B$ and P is the URMO program that computes c_A relative to B. Then $\forall x$, $P^B(x)$ converges and

$$P^B(x) \downarrow 1 \text{ if } x \in A$$
$$P^B(x) \downarrow 0 \text{ if } x \notin A$$

When calculating $P^B(x)$ there will be a finite number of requests to the oracle for a value $c_B(n)$ of c_B. These requests amount to a finite number of questions of the form '$n \in B$?'.

So for any x, '$x \in A$?' is settled in a mechanical way by answering a finite number of questions about B.

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.

Facts.

(i) $\leq_T$ is reflexive and transitive.
$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.
$A \equiv_T B$ iff $\mathscr{C}^A = \mathscr{C}^B$;

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.

$A \equiv_T B$ iff $\mathscr{C}^A = \mathscr{C}^B$;

(iii) If $A \leq_m B$ then $A \leq_T B$.

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.

$A \equiv_T B$ iff $\mathscr{C}^A = \mathscr{C}^B$;

(iii) If $A \leq_m B$ then $A \leq_T B$.

If $f : A \leq_m B$ and P is URM program to compute f, then the URMO program P, $O(1)$ is B-compute c_A.

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.

$A \equiv_T B$ iff $\mathscr{C}^A = \mathscr{C}^B$;

(iii) If $A \leq_m B$ then $A \leq_T B$.

If $f : A \leq_m B$ and P is URM program to compute f, then the URMO program P, $O(1)$ is B-compute c_A.

(iv) $A \equiv_T \overline{A}$ for all A.

Facts.

(i) $\leq_T$ is reflexive and transitive.

$A \leq_T B$ iff $\mathscr{C}^A \subseteq \mathscr{C}^B$;

(ii) $\equiv_T$ is an equivalence relation.

$A \equiv_T B$ iff $\mathscr{C}^A = \mathscr{C}^B$;

(iii) If $A \leq_m B$ then $A \leq_T B$.

If $f : A \leq_m B$ and P is URM program to compute f, then the URMO program $P, O(1)$ is B-compute c_A.

(iv) $A \equiv_T \overline{A}$ for all A.

$c_{\overline{A}} = \overline{sg} \circ c_A$, $\overline{A}$ is A-recursive $\Longrightarrow \overline{A} \leq_T A$. (Similarly $A \leq_T \overline{A}$.)

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Since $\mathscr{C} \subseteq \mathscr{C}^X$.

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Since $\mathscr{C} \subseteq \mathscr{C}^\chi$.

(vi) If B is recursive and $A \leq_T B$, then A is recursive.

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Since $\mathscr{C} \subseteq \mathscr{C}^{\chi}$.

(vi) If B is recursive and $A \leq_T B$, then A is recursive.

If χ is computable, then $\mathscr{C} = \mathscr{C}^{\chi}$.

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Since $\mathscr{C} \subseteq \mathscr{C}^\chi$.

(vi) If B is recursive and $A \leq_T B$, then A is recursive.

If χ is computable, then $\mathscr{C} = \mathscr{C}^\chi$.

(vii) If A is r.e. then $A \leq_T K$.

Facts. (2)

(v) If A is recursive, then $A \leq_T B$ for all B.

Since $\mathscr{C} \subseteq \mathscr{C}^\chi$.

(vi) If B is recursive and $A \leq_T B$, then A is recursive.

If χ is computable, then $\mathscr{C} = \mathscr{C}^\chi$.

(vii) If A is r.e. then $A \leq_T K$.

If $A \leq_m B$ then $A \leq_T B$; A set A is r.e. iff $A \leq_m K$.

Turing Degrees

A set A is T-complete if A is r.e. and $B \leq_T A$ for every r.e. set B.

Turing Degrees

A set A is T-complete if A is r.e. and $B \leq_T A$ for every r.e. set B.

The equivalence class $d_T(A) = \{B \mid A \equiv_T A\}$ is called Turing degree of A, or T-degree of A.

Turing Degrees

A set A is T-complete if A is r.e. and $B \leq_T A$ for every r.e. set B.

The equivalence class $d_T(A) = \{B \mid A \equiv_T A\}$ is called Turing degree of A, or T-degree of A.

A T-degree containing a recursive set is called a recursive T-degree.

Turing Degrees

A set A is T-complete if A is r.e. and $B \leq_T A$ for every r.e. set B.

The equivalence class $d_T(A) = \{B \mid A \equiv_T A\}$ is called Turing degree of A, or T-degree of A.

A T-degree containing a recursive set is called a recursive T-degree.

A T-degree containing an r.e. set is called an r.e. T-degree.

Turing Reducibility and Turing Degrees

The set of degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

Turing Reducibility and Turing Degrees

The set of degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

$\mathbf{a} \leq \mathbf{b}$ iff $A \leq_T B$ for all $A \in \mathbf{a}$ and $B \in \mathbf{b}$.

Turing Reducibility and Turing Degrees

The set of degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

$\mathbf{a} \leq \mathbf{b}$ iff $A \leq_T B$ for all $A \in \mathbf{a}$ and $B \in \mathbf{b}$.

$\mathbf{a} < \mathbf{b}$ iff $\mathbf{a} \leq \mathbf{b}$ and $\mathbf{a} \neq \mathbf{b}$.

Turing Reducibility and Turing Degrees

The set of degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.

$\mathbf{a} \leq \mathbf{b}$ iff $A \leq_T B$ for all $A \in \mathbf{a}$ and $B \in \mathbf{b}$.

$\mathbf{a} < \mathbf{b}$ iff $\mathbf{a} \leq \mathbf{b}$ and $\mathbf{a} \neq \mathbf{b}$.

The relation $\leq$ is a partial order.

Theorem

(i) There is precisely one recursive degree $\mathbf{0}$, which consists of all the recursive sets and is the unique minimal degree.

If A is recursive, then $A \leq_T B$ for all B; If B is recursive and $A \leq_T B$, then A is recursive.

Theorem

(i) There is precisely one recursive degree $\mathbf{0}$, which consists of all the recursive sets and is the unique minimal degree.

If A is recursive, then $A \leq_T B$ for all B; If B is recursive and $A \leq_T B$, then A is recursive.

(ii) Let $\mathbf{0}'$ be the degree of K. Then $\mathbf{0} < \mathbf{0}'$ and $\mathbf{0}'$ is a maximum among all r.e. degrees.

From (i), $\mathbf{0} \leq \mathbf{0}'$; $\mathbf{0} \neq \mathbf{0}'$ since K is not recursive. Since A is r.e. $\Rightarrow$ $A \leq_T K$, we have if $\mathbf{a}$ is any r.e. degree, $\mathbf{a} \leq \mathbf{0}'$.

Theorem

(i) There is precisely one recursive degree $\mathbf{0}$, which consists of all the recursive sets and is the unique minimal degree.

If A is recursive, then $A \leq_T B$ for all B; If B is recursive and $A \leq_T B$, then A is recursive.

(ii) Let $\mathbf{0}'$ be the degree of K. Then $\mathbf{0} < \mathbf{0}'$ and $\mathbf{0}'$ is a maximum among all r.e. degrees.

From (i), $\mathbf{0} \leq \mathbf{0}'$; $\mathbf{0} \neq \mathbf{0}'$ since K is not recursive. Since A is r.e. $\Rightarrow$ $A \leq_T K$, we have if $\mathbf{a}$ is any r.e. degree, $\mathbf{a} \leq \mathbf{0}'$.

(iii) $d_m(A) \subseteq d_T(A)$; and if $d_m(A) \leq_m d_m(B)$ then $d_T(A) \leq d_T(B)$.

If $A \leq_m B$ then $A \leq_T B$.

Jump Operation

Theorem. The following statements are valid.

(i) $K^A \stackrel{\text{def}}{=} \{x \mid x \in W_x^A\}$ is A-r.e.

Since K^χ is χ-r.e.

Jump Operation

Theorem. The following statements are valid.

(i) $K^A \stackrel{\text{def}}{=} \{x \mid x \in W_x^A\}$ is A-r.e.

Since K^χ is χ-r.e.

(ii) If B is A-r.e., then $B \leq_T K^A$.

By relativised s-m-n theorem, if B is A-r.e., then $B \leq_m K^A$.

Jump Operation

Theorem. The following statements are valid.

(i) $K^A \stackrel{\text{def}}{=} \{x \mid x \in W_x^A\}$ is A-r.e.

Since K^χ is χ-r.e.

(ii) If B is A-r.e., then $B \leq_T K^A$.

By relativised s-m-n theorem, if B is A-r.e., then $B \leq_m K^A$.

(iii) If A is recursive then $K^A \equiv_T K$.

"$\Leftarrow$" $K \leq_T K^A$ since K is A-r.e. for any A;

"$\Rightarrow$" If A is recursive then A-computable partial characteristic function of K^A is actually computable (if χ is computable, then $\mathscr{C} = \mathscr{C}^\chi$). Hence K^A is r.e., and $K^A \leq_T K$.

Jump Operation

Theorem. The following statements are valid.

(i) $K^A \stackrel{\text{def}}{=} \{x \mid x \in W_x^A\}$ is A-r.e.

Since K^χ is χ-r.e.

(ii) If B is A-r.e., then $B \leq_T K^A$.

By relativised s-m-n theorem, if B is A-r.e., then $B \leq_m K^A$.

(iii) If A is recursive then $K^A \equiv_T K$.

"$\Leftarrow$" $K \leq_T K^A$ since K is A-r.e. for any A;

"$\Rightarrow$" If A is recursive then A-computable partial characteristic function of K^A is actually computable (if χ is computable, then $\mathscr{C} = \mathscr{C}^\chi$). Hence K^A is r.e., and $K^A \leq_T K$.

(iv) $A <_T K^A$.

"$A \leq_T K^A$" is given by (ii). "$A \not\equiv_T K^A$" is given by "K^χ is χ-r.e. but not χ-recursive."

Relativization

(v) If $A \leq_T B$ then $K^A \leq_T K^B$.

If $A \leq_T B$, then since K^A is A-r.e. it is also B-r.e., so $K^A \leq_T K^B$.

Relativization

(v) If $A \leq_T B$ then $K^A \leq_T K^B$.

If $A \leq_T B$, then since K^A is A-r.e. it is also B-r.e., so $K^A \leq_T K^B$.

(vi) If $A \equiv_T B$ then $K^A \equiv_T K^B$.

Follows immediately from (v).

Jump Operation

K^A is a T-complete A-r.e. set. Also called the completion of A, or the jump of A, and denoted as A'.

Jump Operation

K^A is a T-complete A-r.e. set. Also called the completion of A, or the jump of A, and denoted as A'.

Definition. The jump of $\mathbf{a}$, denoted $\mathbf{a}'$, is the degree of K^A for any $A \in \mathbf{a}$.

Jump Operation

K^A is a T-complete A-r.e. set. Also called the completion of A, or the jump of A, and denoted as A'.

Definition. The jump of $\mathbf{a}$, denoted $\mathbf{a}'$, is the degree of K^A for any $A \in \mathbf{a}$.

Notation (1). By Relativization jump is a valid definition because the degree of K^A is the same for every $A \in \mathbf{a}$.

Notation (2). The new definition of $\mathbf{0}'$ as the jump of $\mathbf{0}$ accords with our earlier definition of $0'$ as the degree of K.

Basic Properties

Theorem. For any degree $\mathbf{a}$ and $\mathbf{b}$, the following statements are valid.

(i) $\mathbf{a} < \mathbf{a}'$.

(ii) If $\mathbf{a} < \mathbf{b}$ then $\mathbf{a}' < \mathbf{b}'$

(iii) If $B \in \mathbf{b}$, $A \in \mathbf{a}$ and B is A-r.e. then $\mathbf{b} \leq \mathbf{a}'$.

Important Results

Theorem. Any degrees $\mathbf{a}, \mathbf{b}$ have a unique least upper bound.

Theorem. Any non-recursive r.e. degree contains a simple set.

Theorem. There are r.e. sets A, B s.t. $A \not\leq_T B$ and $B \not\leq_T A$. Hence, if $\mathbf{a}$, $\mathbf{b}$ are $d_T(A)$, $d_T(B)$ respectively, $\mathbf{a} \not\leq \mathbf{b}$ and $\mathbf{b} \not\leq \mathbf{a}$, and thus $\mathbf{0} < \mathbf{a} < \mathbf{0}'$ and $\mathbf{0} < \mathbf{b} < \mathbf{0}'$.

Degrees $\mathbf{a}$, $\mathbf{b}$ such that $\mathbf{a} \not\leq \mathbf{b}$ and $\mathbf{b} \not\leq \mathbf{a}$ are called incomparable degrees, denoted as $\mathbf{a} \mid \mathbf{b}$.

Theorem. For any r.e. degree $\mathbf{a} > \mathbf{0}$, there is an r.e. degree $\mathbf{b}$ such that $\mathbf{b} \mid \mathbf{a}$.

Important Results (2)

Sack's Density Theorem. For any r.e. degrees $\mathbf{a} < \mathbf{b}$ there is an r.e. degree $\mathbf{c}$ with $\mathbf{a} < \mathbf{c} < \mathbf{b}$.

Sack's Splitting Theorem. For any r.e. degrees $\mathbf{a} > \mathbf{0}$ there are r.e. degrees $\mathbf{b}$, $\mathbf{c}$ such that $\mathbf{b} < \mathbf{a}$ $\mathbf{c} < \mathbf{a}$ and $\mathbf{a} = \mathbf{b} \cup \mathbf{c}$ (hence $\mathbf{b} \mid \mathbf{c}$).

Lachlan, Yates Theorem.
(a). $\exists$ r.e. degrees $\mathbf{a}$, $\mathbf{b} > \mathbf{0}$ such that $\mathbf{0}$ is the greatest lower bound of $\mathbf{a}$ and $\mathbf{b}$.
(b). $\exists$ r.e. degrees $\mathbf{a}$, $\mathbf{b}$ having no greatest lower bound (either among all degrees or among r.e. degrees).

Shoenfield Theorem. There is a non-r.e. degree $\mathbf{a} < \mathbf{0}'$.

Spector Theorem. There is a minimal degree. (A minimal degree is a degree $\mathbf{m} > \mathbf{0}$ such that there is no degree $\mathbf{a}$ with $\mathbf{0} < \mathbf{a} < \mathbf{m}$).

Theorem. For any r.e. m-degree $\mathbf{a} >_m \mathbf{0_m}$, $\exists$ an r.e. m-degree $\mathbf{b}$ s.t. $\mathbf{b} \mid \mathbf{a}$.