Computability Theory

Check List For Final Exam, Xiaofeng Gao's Section, 2016 Spring

Description:

This checklist covers all the contents for the final exam. It includes Chapter 6, Chapter 7, and Chapter 9.
(Note: Multiple options are available to prepare for the final exam. Reading the textbook is a must for success. Slides, assignments, and answer keys can be good supplements for all topics. For the notations, please refer to the Notations in the text book, page 241-245.)

Chapter 6. Decidability, undecidability and partial decidability

1. Decidability:
   (a) **Definition**. A predicate $M(\mathbf{x})$ is decidable if its characteristic function $c_M(\mathbf{x})$ given by $c_M(\mathbf{x}) = \begin{cases} 1, & \text{if } M(\mathbf{x}) \text{ holds,} \\ 0, & \text{if } M(\mathbf{x}) \text{ does not hold.} \end{cases}$ is computable.
   (b) The predicate $M(\mathbf{x})$ is undecidable if it is not decidable.
   (c) In literature $M(\mathbf{x})$ is decidable can be described as $M(\mathbf{x})$ is recursively decidable, $M(\mathbf{x})$ has recursive decision problem, $M(\mathbf{x})$ is solvable, $M(\mathbf{x})$ is recursively solvable, or $M(\mathbf{x})$ is computable.
2. Undecidable problems in computability:
   (a) **Theorem**. The problem '$x \in W_x$' is undecidable.
   (b) **Corollary**. There is a computable function h such that both '$x \in Dom(h)$' and '$x \in Ran(h)$' are undecidable.
   (c) **Theorem**. (the Halting problem) The problem '$\phi_x(y)$ is defined' is undecidable.
   (d) **Theorem**. The problem '$\phi_x = \mathbf{0}$' is undecidable.
   (e) **Corollary**. The problem '$\phi_x = \phi_y$' is undecidable.
   (f) **Theorem**. Let c be any number. The followings are undecidable.
      i. Acceptance Problem: '$c \in W_x$',
      ii. Printing Problem: '$c \in E_x$'.
   (g) **Theorem**. (Rice's theorem) '$\phi_x \in \mathscr{B}$' is undecidable for $\emptyset \subsetneq \mathscr{B} \subsetneq \mathscr{C}_1$.
3. Partially decidable predicates:
   (a) **Definition**. A predicate $M(\mathbf{x})$ of natural numbers is partially decidable if the function given by $f(\mathbf{x}) = \begin{cases} 1, & \text{if } M(\mathbf{x}) \text{ holds,} \\ \text{undefined,} & \text{if } M(\mathbf{x}) \text{ does not hold,} \end{cases}$ is computable.
   The function is called the partial characteristic function for M.
   (b) In the literature the terms partially solvable, semi-computable, and recursively enumerable are used with the same meaning as partially decidable.
   (c) partially decidable predicates:
      i. The halting problem is partially decidable. Its partial characteristic function is given by $f(x, y) = \begin{cases} 1, & \text{if } P_x(y) \downarrow, \\ \text{undefined,} & \text{otherwise.} \end{cases}$
      ii. The problem '$x \notin W_x$' is not partially decidable. The domain of its partial characteristic function differs from the domain of every computable function.
   (d) **Theorem**. A predicate $M(\mathbf{x})$ is partially decidable iff there is a computable function $g(x)$ such that $M(\mathbf{x}) \Leftrightarrow \mathbf{x} \in Dom(g)$.
   (e) **Theorem**. A predicate $M(\mathbf{x})$ is partially decidable iff there is a decidable predicate $R(\mathbf{x}, y)$ such that $M(\mathbf{x}) \Leftrightarrow \exists y.R(\mathbf{x}, y)$.
   (f) **Theorem**. If $M(\mathbf{x}, y)$ is partially decidable, so is $\exists y.M(\mathbf{x}, y)$.

(g) **Corollary**. If $M(\mathbf{x}, \mathbf{y})$ is partially decidable, so is $\exists \mathbf{y}.M(\mathbf{x}, \mathbf{y})$.
(h) **Theorem**. $M(\mathbf{x})$ is decidable iff both $M(\mathbf{x})$ and $\neg M(\mathbf{x})$ are partially decidable.
(i) **Corollary**. The problem '$y \notin W_x$' is not partially decidable.
(j) **Theorem**. Let $f(\mathbf{x})$ be a partial function. Then f is computable iff the predicate '$f(\mathbf{x}) \simeq y$' is partially decidable.

Key Terms:

Decidability, Undecidability, the Halting problem, Rice's theorem, partial decidability.

Practice and Sources:

1. Slide08-Undecidability; 2. Textbook page 100-120; 3. Lab07-Undecidability

Chapter 7. Recursive And Recursively Enumerable Sets

1. Recursive Sets:
   (a) **Definition**. 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.
   (b) Examples of recursive sets:
   i. $\mathbb{N}$, $\mathbb{Z}$.
   ii. $\mathbb{E}$ (even numbers).
   iii. $\mathbb{O}$ (odd numbers).
   iv. $\mathbb{O}$ (prime numbers).
   v. Any finite set.
   (c) Examples of unsolvable problems:
   i. $K = \{x \mid x \in W_x\}$, $\overline{K} = \{x \mid x \notin W_x\}$
   ii. $Fin = \{x \mid W_x \text{ is finite}\}$, $Inf = \{x \mid W_x \text{ is infinite}\}$,
   iii. $Cof = \{x \mid W_x \text{ is cofinite}\}$, $Tot = \{x \mid \phi_x \text{ is total}\}$,
   iv. $Rec = \{x \mid W_x \text{ is recursive}\}$,
   v. $Ext = \{x \mid \phi_x \text{ is extensible to total recursive function}\}$.
   (d) **Fact**. Recursive Set $\Leftrightarrow$ Solvable Problem $\Leftrightarrow$ Decidable Predicate.
   (e) **Theorem**. If A, B are recursive sets, then so are the sets $\overline{A}$, $A \cap B$, $A \cup B$, $A \backslash B$.
2. Recursively Enumerable Sets (r.e. set):
   (a) **Definition**. Let A be a subset of $\mathbb{N}$. Then A is recursively enumerable if the function f given by $f(x) = \begin{cases} 1, & \text{if } x \in A, \\ \text{undefined}, & \text{if } x \notin A. \end{cases}$ 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 $\mathbb{N}$.
   (b) **Fact**. Partially Decidable Problem $\Leftrightarrow$ Partially Decidable Predicate $\Leftrightarrow$ R. E. Set
   (c) **Index Theorem**. A set is r.e. iff it is the domain of a unary computable function.
   (d) **Normal Form Theorem**. The set A is r.e. iff there is a primitive recursive predicate $R(\mathbf{x}, y)$ such that $\mathbf{x} \in A$ iff $\exists y.R(\mathbf{x}, y)$.
   (e) **Quantifier Contraction Theorem**. If $M(\mathbf{x}, \mathbf{y})$ is partially decidable, so is $\exists \mathbf{y}.M(\mathbf{x}, \mathbf{y})$ ($\{\mathbf{x} \mid \exists \mathbf{y}.M(\mathbf{x}, \mathbf{y})\}$ is r.e.).
   (f) **Uniformisation 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 $R(x, c(x))$.
   (g) **Complementation Theorem**. A is recursive iff A and $\overline{A}$ are r.e.
   (h) **Graph 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.
   (i) **Listing Theorem**. A is r.e. iff $A = \varnothing$ or $A = Ran(f)$ for a total function $f \in \mathscr{C}_1$.
   Equivalence Theorem. Let $A \subseteq \mathbb{N}$. Then the following are equivalent:
   i. A is r.e.
   ii. $A = \emptyset$ or A is the range of a unary total computable function.
   iii. A is the range of a (partial) computable function.
   Theorem. Every infinite r.e. set has an infinite recursive subset.
   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).
Theorem. The set $\{x \mid \phi_x \text{ is total}\}$ is not r.e.

   (j) **Closure Theorem**. The recursively enumerable sets are closed under union and intersection uniformly and effectively.
   (k) **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 $\theta \in \mathcal{A}$.
   Corollary. The sets $\{x \mid \phi_x \text{ is total}\}$ and $\{x \mid \phi_x \text{ is not total}\}$ are not r.e.
   (l) **Theorem**. If A and B are r.e., then so are $A \cap B$ and $A \cup B$.

3. Productive Sets:
   (a) **Definition**. A set A is productive if there is a total computable function g such that whenever $W_x \subseteq A$, then $g(x) \in A \setminus W_x$. g is called a productive function for A.
   Notation. A productive set is not r.e.
   (b) Examples of productive sets:
      i. $\{x \mid \phi_x \neq \mathbf{0}\}$ is productive.
      ii. $\{x \mid c \notin W_x\}$ is productive.
      iii. $\{x \mid c \notin E_x\}$ is productive.

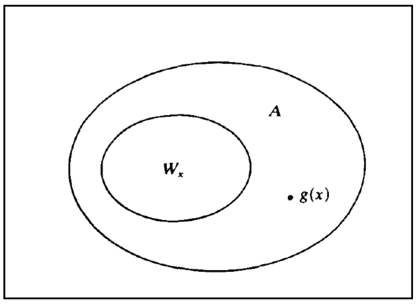

Fig. A productive set

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

4. Creative sets:
   (a) **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 $\Leftrightarrow$ 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.)
   (b) **Theorem**. Suppose that $\mathscr{A} \subseteq \mathscr{C}_1$ and let $A = \{x \mid \phi_x \in \mathscr{A}\}$. If A is r.e. and $A \neq \emptyset$, $\mathbb{N}$, then A is creative.
   (c) **Lemma**. Suppose that g is a total computable function. Then there is a total computable function k such that for all x, $W_{k(x)} = W_x \cup \{g(x)\}$.
   Subset Theorem. A productive set contains an infinite r.e. subset.
   Corollary. If A is creative, then $\overline{A}$ contains an infinite r.e. subset.

5. Simple Set:
   (a) **Definition**. A set A is simple if A is r.e.,$\overline{A}$ is infinite and contains no infinite r.e. subset.
   (b) **Theorem**. A simple set is neither recursive nor creative.
   (c) **Theorem**. There is a simple set.

Key Terms:

Recursive Set, Recursively Enumerable Set, Productive Set, Creative Set, Simple Set.

Practice and Sources:

1. Slide09-RESet
2. Textbook page 121-142;
3. Lab08-Lab10.

Table 1: Various Sets

Set	Definition	Theorem	Example	Counter Example
Recursive Set	$c_A(x) = \begin{cases} 1, & \text{if } x \in A, \\ 0, & \text{if } x \notin A. \end{cases}$ is computable.	① Recursive Function Theorems ② Closure: A,B are r. $\Rightarrow \overline{A}$, $A \cup B$, $A \cap B$ are r. ③ Rice Theorem: $\varnothing \subsetneq \mathscr{B} \subsetneq \mathscr{C}_1 \Rightarrow$ '$\phi_x \in \mathscr{B}$' is undecidable. ④ Any Theorems for Decidable Predicates.	$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{E}$, $\mathbb{O}$, $\mathbb{P}$ Any finite set	K, $\overline{K}$; Fin, Inf, Cof; Rec, Tot, Ext Any non-r.e. set
Recursively Enumerable Set (r.e. set)	$\chi_A(x) = \begin{cases} 1, & \text{if } x \in A, \\ \uparrow, & \text{if } x \notin A. \end{cases}$ is computable.	① Index $\leftrightarrow$ ② Listing $\begin{cases}$ ③ Equivlence $\exists$ infinite $r. \subseteq r.e.$ $r. \Leftrightarrow \exists f \in \mathscr{C}_1 \upuparrows, Ran(f)$ $\end{cases}$ ④ Normal Form $\begin{cases}$ ⑤ Uniformization ⑥ Graph ⑦ Quantifier Constraction $\end{cases}$ ⑧ Complementation (A is $r. \Leftrightarrow A, \overline{A}$ are r.e.) ⑨ Closure (A,B are r.e. $\Rightarrow A \cap B$, $A \cup B$ are r.e.) ⑩ Rice-Shapiro: $\mathscr{A} \subseteq \mathscr{C}_1$, $\{x \mid \phi_x \in \mathscr{A}\}$ is r.e., then $\forall f \in \mathscr{C}_1$, $f \in \mathscr{A} \Leftrightarrow \exists$ finite $\theta \subseteq f$ with $\theta \in \mathscr{A}$	all recursive set non-recursive r.e. set $\{x \mid x \in W_x\}$ $\{x \mid \phi_x(x) = 0\}$ $\{x \mid W_x \neq \varnothing\}$ $\{x \mid x\ 7's \text{ in } \pi\}$	$\overline{K}$; Fin, Inf, Cof; Tot, $\overline{Tot}$, Con; Rec, Ext
Productive Set	A is productive if $\exists$ total $g \in \mathscr{C}_1$ s.t. $\forall\, W_x \subseteq A$, $g(x) \in A \setminus W_x$	① Reduction Theorem A is productive and $A \leq_m B \Rightarrow B$ is productive ② Quasi-Rice Theorem $\mathscr{B} \subsetneq \mathscr{C}_1$, $f_\varnothing \in \mathscr{B} \Rightarrow \{x \mid \phi_x \in \mathscr{B}\}$ is productive ③ Quasi-Listing Theorem Productive set has r.e. subset	$\{x \mid \phi_x(x) \neq 0\}$ $\{x \mid c \notin W_x\}$ $\{x \mid c \notin E_x\}$ $\{x \mid \phi_x \text{ is not total}\}$	① r.e. set ② doesn't have r.e. subset
Creative Set	$\begin{cases}$ A is r.e.; $\overline{A}$ is productive. $\end{cases}$	① Quasi-Rice Theorem $\mathscr{A} \subseteq \mathscr{C}_1$, $A = \{x \mid \phi_x \in \mathscr{A}\}$. If A is r.e., $A \neq \varnothing$, $\mathbb{N}$, then A is creative	$\{x \mid \phi_x(x) = 0\}$ $\{x \mid c \in W_x\}$ $\{x \mid c \in E_x\}$	① non-r.e. set ② simple set
Simple Set	$\begin{cases}$ A is r.e.; $\overline{A}$ is infinite; $\overline{A}$ contains no infinite r.e. subset. $\end{cases}$	① Characteristic Theorem (A simple set is neither recursive nor creative) ② Existence Theorem (There is a simple set)	If A, B are simple: $A \oplus B$ is simple $A \otimes B$ is not simple $\overline{\overline{A} \otimes \overline{B}}$ is simple	Any recursive set Any creative set

Chapter 9. Reducibility And Degrees

1. Many-One Reducibility:
   (a) **Definition**. 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 $f : A \leq_m B$.
   Notation. If f is injective, then we are talking about one-one reducibility, denoted by $f : A \leq_1 B$.
   (b) **Theorem**. Let A, B, C be sets.
      i. $\leq_m$ is reflexive: $A \leq_m A$.
      ii. $\leq_m$ is transitive: $A \leq_m B$, $B \leq_m C \Rightarrow A \leq_m C$.
      iii. $A \leq_m B$ iff $\overline{A} \leq_m \overline{B}$.
      iv. If A is recursive and $B \leq_m A$, then B is recursive.
      v. If A is recursive and $B \neq \varnothing, \mathbb{N}$, then $A \leq_m B$.
      vi. If A is r.e. and $B \leq_m A$, then B is r.e.
      vii. (i). $A \leq_m \mathbb{N}$ iff $A = \mathbb{N}$; (ii). $A \leq_m \varnothing$ iff $A = \varnothing$.
      viii. (i). $\mathbb{N} \leq_m A$ iff $A \neq \varnothing$; (ii). $\varnothing \leq_m A$ iff $A \neq \mathbb{N}$.
   (c) **Corollary**. Neither $\{x \mid \phi_x \text{ is total}\}$ nor $\{x \mid \phi_x \text{ is not total}\}$ is m-reducible to K.
   Corollary. If A is r.e. and is not recursive, then $\overline{A} \not\leq_m A$ and $A \not\leq_m \overline{A}$.
   Notation. It contradicts to our intuition that A and $\overline{A}$ are equally difficult.
   (d) **Theorem**. A is r.e. iff $A \leq_m K$.
   Notation. K is the most difficult partially decidable problem.
2. m-Degrees:
   (a) **Definition**. Two sets A, B are many-one equivalent, notation $A \equiv_m B$ (abbreviated m-equivalent), if $A \leq_m B$ and $B \leq_m A$.
   (b) **Theorem**. The relation $\equiv_m$ is an equivalence relation.
   (c) **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.
   (d) **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.
      i. $\mathbf{a} \leq_m \mathbf{b}$ iff $A \leq_m B$ for some $A \in \mathbf{a}$ and $B \in \mathbf{b}$.
      ii. $\mathbf{a} <_m \mathbf{b}$ iff $\mathbf{a} \leq_m \mathbf{b}$ and $\mathbf{b} \not\leq_m \mathbf{a}$ ($\mathbf{a} \neq \mathbf{b}$).
   The relation $\leq_m$ is a partial order.
   Notation. From the definition of $\equiv_m$, $\mathbf{a} \leq_m \mathbf{b} \Leftrightarrow \forall A \in \mathbf{a}, B \in \mathbf{b}, A \leq_m B$.
   (e) **Theorem**. The relation $<_m$ is a partial ordering of m-degrees.
   (f) **Theorem**. Difficulty Class
      i. $\mathbf{o}$ and $\mathbf{n}$ are respectively the recursive m-degrees $\{\varnothing\}$ and $\{\mathbb{N}\}$.
      ii. 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 $\mathbf{a}$ other than $\mathbf{o}$, $\mathbf{n}$.
      iii. $\forall$ m-degree $\mathbf{a}$, $\mathbf{o} \leq_m \mathbf{a}$ provided $\mathbf{a} \neq \mathbf{n}$; $\mathbf{n} \leq_m \mathbf{a}$ provided $\mathbf{a} \neq \mathbf{o}$.
      iv. An r.e. m-degree consists of only r.e. sets.
      v. 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.
      vi. The maximum r.e. m-degree $d_m(K)$ is denoted by $\mathbf{0}'_m$.
   (g) Algebraic Structure
      i. **Theorem**. m-degrees form an upper semi-lattice.
      ii. Lattice: 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: $\forall a, b \in L$, $\{a, b\}$ has a join $a \vee b$. (the least upper bound)
meet-semilattice: $\forall a, b \in L$, $\{a, b\}$ has a meet $a \wedge b$. (the greatest lower bound)

iii. **Theorem**. Any pair of m-degrees $\mathbf{a}$, $\mathbf{b}$ have a least upper bound; i.e. there is an m-degree $\mathbf{c}$ such that
   A. $\mathbf{a} \leq_m \mathbf{c}$ and $\mathbf{b} \leq_m \mathbf{c}$ ($\mathbf{c}$ is an upper bound);
   B. $\mathbf{c} \leq_m$ any other upper bound of $\mathbf{a}$, $\mathbf{b}$.

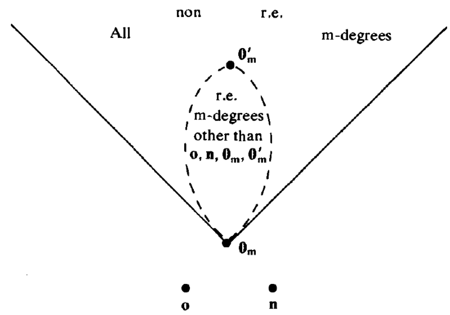

Fig. The m-degrees

3. m-complete r.e. sets:
   (a) **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).
   (b) **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. $0'_m$ consists exactly of all the m-complete sets.
   (c) **Myhill's Theorem**. A set is m-complete iff it is creative.
   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).
4. Relative Computability:
   (a) Unlimited Register Machine with Oracle (URMO):
      i. **Definition**. 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.
      ii. **Definition**. 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.
      iii. **Definition**. Let χ be a unary total function, and f 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.
      iv. **Theorem**.
         A. $\chi \in \mathscr{C}^\chi$.
         B. $\mathscr{C} \subseteq \mathscr{C}^\chi$.
         C. If χ is computable, then $\mathscr{C} = \mathscr{C}^\chi$.
         D. $\mathscr{C}^\chi$ is closed under substitution, recursion and minimalisation.
         E. If ψ is a total unary function that is χ-computable, then $\mathscr{C}^\psi \subseteq \mathscr{C}^\chi$.
   (b) χ-partial recursive function:

i. **Definition**. 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.

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

(c) Numbering URMO programs

i. 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.
ϕ_m^χ is $\phi_m^{\chi,1}$. $W_m^\chi = Dom(\phi_m^\chi)$ and $E_m^\chi = Ran(\phi_m^\chi)$.

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

(d) 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.

(e) χ-recursive and χ-r.e. sets :

i. **Definition**. Let A be a set

A. A is χ-recursive if c_A is χ-computable.

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

ii. **Theorem**. The following statements are valid.

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

B. For any set A, the following are equivalent.

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

C. $K^\chi \stackrel{\text{def}}{=} \{x \mid x \in W_x^\chi\}$ is χ-r.e. but not χ-recursive.

(f) Computability relative to set A means relative to characteristic function c_A.

5. Turing reducibility and Turing degrees:

(a) **Definition**. The set A is Turing reducible to B, notation $A \leq_T B$, if A has a B-computable characteristic function c_A.
Definition. A, B are Turing equivalent, notation $A \equiv_T B$, if $A \leq_T B$ and $B \leq_T A$.

(b) **Theorem**.

i. $\leq_T$ is reflexive and transitive.
ii. $\equiv_T$ is an equivalence relation.
iii. If $A \leq_m B$ then $A \leq_T B$.
iv. $A \equiv_T \overline{A}$ for all A.
v. If A is recursive, then $A \leq_T B$ for all B.
vi. If B is recursive and $A \leq_T B$, then A is recursive.
vii. If A is r.e. then $A \leq_T K$.

(c) **Definition**. A set A is T-complete if A is r.e. and $B \leq_T A$ for every r.e. set B.

(d) **Definition**. T-Degree

i. The equivalence class $d_T(A) = \{B \mid A \equiv_T B\}$ is the Turing degree (T-degree) of A.
ii. A T-degree containing a recursive set is called a recursive T-degree.
iii. A T-degree containing an r.e. set is called an r.e. T-degree.

(e) **Definition**. The set of degrees is ranged over by $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$.
   i. $\mathbf{a} \le \mathbf{b}$ iff $A \le_T B$ for all $A \in \mathbf{a}$ and $B \in \mathbf{b}$.
   ii. $\mathbf{a} < \mathbf{b}$ iff $\mathbf{a} \le \mathbf{b}$ and $\mathbf{a} \ne \mathbf{b}$.

   Notation. The relation $\le$ is a partial order.

(f) **Theorem**.
   i. There is precisely one recursive degree $\mathbf{0}$, which consists of all the recursive sets and is the unique minimal degree.
   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.
   iii. $d_m(A) \subseteq d_T(A)$; and if $d_m(A) \le_m d_m(B)$ then $d_T(A) \le d_T(B)$.

(g) **Theorem**. The jump operation:
   i. $K^A \stackrel{\text{def}}{=} \{x \mid x \in W_x^A\}$. 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'. $A <_T K^A$.
   ii. If B is A-r.e., then $B \le_T K^A$.
   iii. If A is recursive then $K^A \equiv_T K$.
   iv. If $A \le_T B$ then $K^A \le_T K^B$.
   v. If $A \equiv_T B$ then $K^A \equiv_T K^B$.

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

   Notation. By Relativization jump is a valid definition because the degree of K^A is the same for every $A \in \mathbf{a}$. The new definition of $\mathbf{0}'$ as the jump of $\mathbf{0}$ accords with our earlier definition of $0'$ as the degree of K.

(i) **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 $\mathbf{b} \le \mathbf{a}'$.

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

(k) **Theorem**. Any non-recursive r.e. degree contains a simple set.

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

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

(n) **Sack's Density Theorem**. For any r.e. degrees $\mathbf{a} < \mathbf{b}$, $\exists$ r.e. degree $\mathbf{c}$ with $\mathbf{a} < \mathbf{c} < \mathbf{b}$.

(o) **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 $\mathbf{b} \mid \mathbf{c}$).

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

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

(r) **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 $\mathbf{0} < \mathbf{a} < \mathbf{m}$).

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

Key Terms:

Many-one Reducibility, Many-one Equivalent, m-degrees, m-complete, Relative Computability, URMO, χ-computable, Turing Reducibility, Turing Degrees.

Practice and Sources:

1. Slide10-Reducibility;
2. Textbook page 157-181;
3. Lab11, Lab12

NP, NP-Complete and NP Reduction

1. **Decision Problem**: The "Yes" or "No" questions for any input instance.
   (a) For *maximization* problem: add a threshold k and determine whether there exists a solution with size/weight/measure $\geq k$.
   (b) For *minimization* problem: add a threshold k and determine whether there exists a solution with size/weight/measure $\leq k$.
2. **Polynomial Time Algorithm**: Algorithm A runs in poly-time if for every string s, $A(s)$ terminates in at most $p(|s|)$ "steps", where $p(.)$ is some polynomial.
3. **P** Problem: Decision problems for which there is a poly-time algorithm.
4. **NP** Problem: Decision problems for which there exists a poly-time certifier.
   (a) Certifier: a polynomial time algorithm to check whether a given string is a solution.
   (b) Certificate: a solution for a given instance.
5. **NP**-Completeness: a set of the hardest **NP** problems.
   (a) P is **NP**-Complete if i) $P \in \mathbf{NP}$; and ii) $\forall Q \in \mathbf{NP}$, $Q \leq_m^p P$.
   (b) P is **NP**-Hard if $\forall Q \in \mathbf{NP}$, $Q \leq_m^p P$.
6. **Polynomial Time Reduction**:
   (a) Cook Reduction: Problem X polynomial reduces (Cook) to problem Y if arbitrary instances of problem X can be solved using polynomial number of standard computational steps, plus polynomial number of calls to oracle that solves problem Y.
   (b) Karp Reduction: Problem X polynomial transforms (Karp) to problem Y if given any input $x \in X$, we can construct an input y such that x is a yes instance of X iff y is a yes instance of Y. Here we require $|y|$ to be of size polynomial in $|x|$. (Polynomial transformation is polynomial reduction with just one call to oracle for Y, exactly at the end of the algorithm for X.)
7. **co-NP** Problem: The decision problems whose complements are in **NP**.
   (a) Does **NP**=**co-NP**? Consensus opinion is "no". If **NP**≠**co-NP**, then **P**≠**NP**.
   (b) Does **P**=**NP**∩**co-NP**? Mixed opinions.
8. **Basic reduction strategies**
   (a) Reduction by simple equivalence. Example: VERTEX-COVER $\equiv_p$ INDEPENDENT-SET
   (b) Reduction from special case to general case. Example: VERTEX-COVER $\leq_p$ SET-COVER
   (c) Reduction by encoding with gadgets. Example: 3-SAT $\leq_p$ INDEPENDENT-SET.
9. **Sequencing Problems**:
   (a) HAM-CYCLE: given an undirected graph $G = (V, E)$, does there exists a simple cycle that contains every node in V? Proof: 3-SAT $\leq_p$ DIR-HAM-CYCLE $\leq_p$ HAM-CYCLE.
   (b) TSP: given a set of n cities and a pairwise distance function $d(u, v)$, is there a tour of length $\leq D$? Proof: HAM-CYCLE $\leq_p$ TSP.
10. **Partitioning Problems**:
   (a) 3D-MATCHING: given n instructors, n courses, and n times, and a list of the possible courses and times each instructor is willing to teach, is it possible to make an assignment so that all courses are taught at different times? Proof: 3-SAT $\leq_p$ 3D-MATCHING.
   (b) 3-COLOR: Given an undirected graph G does there exists a way to color the nodes red, green, and blue so that no adjacent nodes have the same color? Proof: 3-SAT $\leq_p$ 3-COLOR
   (c) Scheduling With Release Times: Given a set of n jobs with processing time t_i, release time r_i, and deadline d_i, is it possible to schedule all jobs on a single machine such

that job i is processed with a contiguous slot of t_i time units in the interval $[r_i, d_i]$? Proof:SUBSET-SUM $\leq_p$ SCHEDULE-RELEASE-TIMES.

11. **Numerical Problems**:
    (a) SUBSET-SUM: given natural numbers $w_1, \ldots, w_n$ and an integer W, is there a subset that adds up to exactly W? Proof: 3-SAT $\leq_p$ SUBSET-SUM

Key Terms:

Polynomial-time Reduction, P, NP, co-NP, NP-Complete, NP-Hard, Certificate, Certifier, Decision Problem

Practice and Sources:

1. Slide11-Reduction; Slide12-NPReduction
2. Lab-12, Lab13