Computer Science 513

More about Reductions and Reducibilities

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #15

Goals for Today

Goals for Today:

- Introduction of two more useful reduciibilities
- Discussion of additional properties of reducibilities
- Introduction and discussion of “sets that are computably enumerable, relative to an oracle”

Reductions and Reducibilities: Three Examples

- Recall that a ***reducibility*** is any binary relation $\preceq_Q$ on subsets of $\mathbb{N}$ such that, for all $A, B, C \subseteq \mathbb{N}$,
  - (a) $A \preceq_Q A$, and
  - (b) if $A \preceq_Q B$ and $B \preceq_Q C$ then $A \preceq_Q C$.
- As noted in Lecture #11, the set of ***many-one reductions*** forms a reducibility.

Reductions and Reducibilities: Three Examples

- ***Definition:*** A total function $f : \mathbb{N} \to \mathbb{N}$ is ***one-one*** if, for all $x, y \in \mathbb{N}$, if $x \neq y$ then $f(x) \neq f(y)$. That is, there is *at most* one number $x \in \mathbb{N}$ for any given number $z \in \mathbb{N}$ such that $f(x) = z$.
- ***Definition:*** Let $A, B \subseteq \mathbb{N}$. Then A is ***one-one reducible*** to B,

$$A \preceq_1 B,$$

if there exists a one-one computable total function $f : \mathbb{N} \to \mathbb{N}$ such that

$$A = \{x \in \mathbb{N} \mid f(x) \in B\}.$$

That is, $x \in A$ ***if and only if*** $f(x) \in B$, for ***all*** $x \in \mathbb{N}$.

Reductions and Reducibilities: Three Examples

Theorem #1: $\preceq_1$ is a reducibility.

How This is Proved:

- A straightforward application of the definition of a "reducibility" along with properties of total functions $f : \mathbb{N} \to \mathbb{N}$.

Details for this and other proofs are included in the supplemental document for this lecture.

Reductions and Reducibilities: Three Examples

- Recall that if $S \subseteq \mathbb{N}$ then the ***characteristic function*** is the total predicate $p_S : \mathbb{N} \to \mathbb{N}$ such that, for all $n \in \mathbb{N}$,

$$p_S(n) = \begin{cases} 1 & \text{if } n \in S \\ 0 & \text{if } n \notin S. \end{cases}$$

  Since p_S is a ***total unary*** function, ***p_S-computable functions*** can be defined as in the previous lecture.

- ***Definition:*** If $A, B \in S$ then A is ***Turing-reducible*** to B,

$$A \preceq_T B,$$

  if the characteristic function p_A of A is p_B-computable, where p_B is the characteristic function of B.

Reductions and Reducibilities: Three Examples

Theorem #2: $\preceq_T$ is a reducibility.

Sketch of Proof: This is easily proved using the definition of a reducibility and results established in the previous set of lecture notes (and given as Theorem #3 and #5). $\square$

Reductions and Reducibilities: Three Examples

These three reducibilities are related, as follows.

Theorem #3: Let $A, B \subseteq \mathbb{N}$.

(a) If $A \preceq_1 B$ then $A \preceq_m B$.

(b) If $A \preceq_m B$ then $A \preceq_T B$.

See the supplemental document for a proof of this claim.

Reductions and Reducibilities: General Properties

- ***Note:*** Students who have completed CPSC 413 have seen one more kind of reducibility — "polynomial-time (many-one) reducibility" — except that this was defined as a relationship between *languages* instead of as a relationship between subsets of $\mathbb{N}$.
- For the rest of this lecture, "$\preceq_Q$" will be used to denote some reducibility, as defined above.
- ***Notation:*** For $A, B \subseteq \mathbb{N}$, we will write $A \not\preceq_Q B$ to denote the fact that A is ***not*** reducible to B (for this kind of reducibility).

Reductions and Reducibilities: General Properties

Recall that if $A \subseteq \mathbb{N}$ then the ***complement*** of A is the set

$$\overline{A} = \mathbb{N} \setminus A = \{x \in \mathbb{N} \mid x \notin A\}.$$

Definition: If $\mathcal{W}$ is a collection of subsets of $\mathbb{N}$ then ***co-$\mathcal{W}$*** is the collection

$$\text{co-}\mathcal{W} = \{\overline{A} \mid A \in \mathcal{W}\}$$

— another (possibly different) collection of subsets of $\mathbb{N}$.

Reductions and Reducibilities: General Properties

Theorem #4: Let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$ and let $A \in \mathcal{W}$. Suppose that the following properties are satisfied.

(a) co-$\mathcal{W}$ is Q-closed.

(b) A is Q-complete for $\mathcal{W}$.

(c) $A \in$ co-$\mathcal{W}$.

Then $\mathcal{W} =$ co-$\mathcal{W}$.

How This is Proved: A straightforward application of the definitions that have now been supplied. $\square$

Reductions and Reducibilities: More About the Example Reducibilities

Theorem #5: Let $A, B \subseteq \mathbb{N}$.

(a) If $A \preceq_m B$ then $\overline{A} \preceq_m \overline{B}$.

(b) If $A \preceq_1 B$ then $\overline{A} \preceq_1 \overline{B}$.

This is also a reasonably straightforward consequence of the definitions of these reducibilities.

Exercise: Prove that if $A, B \subseteq \mathbb{N}$ and $A \preceq_T B$ then $\overline{A} \preceq_T \overline{B}$ as well.

Reductions and Reducibilities: More About the Example Reducibilities

Corollary #6: Let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$.

(a) If $\mathcal{W}$ is m-closed then so is co-$\mathcal{W}$.
(b) If $\mathcal{W}$ is 1-closed then so is co-$\mathcal{W}$.
(c) If $\mathcal{W}$ is T-closed then so is co-$\mathcal{W}$.

These are also reasonably straightforward consequences of previous results and applications of the definitions of these reducibilities and of closure. The supplemental document includes a few more details.

Reductions and Reducibilities: More About the Example Reducibilities

Theorem #7: The set of computable subsets of $\mathbb{N}$ is 1-closed, m-closed, and T-closed.

Sketch of Proof: Theorem #6, from the previous lecture, can be used to establish that the set of computable subsets of $\mathbb{N}$ is T-closed. Theorem #3, above, can then be applied to establish that this set must be m-closed and 1-closed, as well. $\square$

Reductions and Reducibilities: More About the Example Reducibilities

Theorem #8: The set of computably enumerable subsets of $\mathbb{N}$ is m-closed and 1-closed, but not T-closed.

Proof: The definitions of a "computably enumerable set" and many-one reducibility can be applied to establish that the set of computably enumerable subsets of $\mathbb{N}$ is m-closed. The relationship between many-one reducibility and one-one reducibility, established above, can then be used to establish that this set is 1-closed as well.

Properties of the sets K and $\overline{K}$, already established, can be used to show that the set of computably enumerable subsets of $\mathbb{N}$ is not T-closed too. $\square$

Once again, the supplemental document gives this argument in more detail.

Reductions and Reducibilities: Complete Problems and Consequences of This

Theorem #9: K is 1-complete for the class of computably enumerable sets.

The proof is included in the supplemental document for this lecture. It makes use of a clever (and not, at all, obvious definition of one from another) along with an application of the Universality Theorem and the Strengthened Parameter Theorem, introduced in recent lectures.

Reductions and Reducibilities: Complete Problems and Consequences of This

Theorem #10: Let $A \subseteq \mathbb{N}$ such that A is m-complete for the class of computably enumerable sets. Then $\overline{A}$ is not computably enumerable, so that A is not computable.

How This is Proved: A proof by contradiction can be obtained by assuming that the claim is false and applying results from earlier in the lecture along with properties of the set K that have now been identified. Once again, details are in the supplemental document for this course. $\square$

Reductions and Reducibilities

The following construction will be needed later on, when proving results about the "Arithmetic Hierarchy."

For $A, B \subseteq \mathbb{N}$, let

$$A \oplus B = \{2n \mid n \in A\} \cup \{2n + 1 \mid n \in B\}.$$

Theorem #11: Let $A, B \subseteq \mathbb{N}$.

(a) $A \preceq_T A \oplus B$.
(b) $B \preceq_T A \oplus B$.
(c) If $C \subseteq \mathbb{N}$ such that $A \preceq_T C$ and $B \preceq_T C$ then $A \oplus B \preceq_T C$.

Once again, a proof can be found in the supplemental document for this lecture.

Sets That are Computably Enumerable Relative to an Oracle

Definition: If $G : \mathbb{N}^k \to \mathbb{N}$ is a ***total function***, for some integer $k \geq 1$, then a set $B \subseteq \mathbb{N}$ is ***G-computably enumerable*** (abbreviated G-c.e.) if there is a G-computable (partial or total) function $g : \mathbb{N} \to \mathbb{N}$ such that

$$B = \{n \in \mathbb{N} \mid g(n) \text{ is defined}\}.$$

Recall, as well, that a set $B \subseteq \mathbb{N}$ is ***G-computable*** if the characteristic function $p_B : \mathbb{N} \to \mathbb{N}$ — the function such that, for all $n \in \mathbb{N}$,

$$p_B(n) = \begin{cases} 1 & \text{if } n \in B \\ 0 & \text{if } n \notin B \end{cases}$$

— is a G-computable *total* function.

Sets That are Computably Enumerable Relative to an Oracle

A variety of the proofs included in Lecture #10 (on computable and computably enumerable sets) relativize. These can be used to establish the following.

Theorem #12: Let $B \subseteq \mathbb{N}$ and let $G : \mathbb{N}^k \to \mathbb{N}$ be a total function, for some integer $k \geq 1$, as above. If B is G-computable then B is G-computably enumerable.

Theorem #13: Let B and G be as in the statement of Theorem #12. Then B is G-computable if and only if B and $\overline{B}$ are both G-computably enumerable.

Theorem #14: Let $B, C \subseteq \mathbb{N}$ and let G be as in the statement of Theorem #12. If B and C are both G-computably enumerable sets then so are $B \cup C$ and $B \cap C$.

Sets That are Computably Enumerable Relative to an Oracle

Theorem #15: Let $A \subseteq \mathbb{N}$ and let G be as in the statement of Theorem #12. Then A is G-computably enumerable if and only if there is a G-computable total predicate $Q : \mathbb{N}^2 \to \{0, 1\}$ such that

$$A = \{x \in \mathbb{N} \mid (\exists t) Q(x, t) = 1\}.$$

The proof — included in the supplemental document — makes use of a predicate that was described and studied when computing the “Relativized Step-Counter Theorem” in the previous lecture.

Sets That are Computably Enumerable Relative to an Oracle

Corollary #16: Let $A \subseteq \mathbb{N}$ and let G be as in the statement of Theorem #12. Then A is G-computably enumerable if and only if there is a G-computable set $B \subseteq \mathbb{N}$ such that

$$A = \{x \in \mathbb{N} \mid (\exists y)(\langle x, y \rangle \in B)\}.$$

Sets That are Computably Enumerable Relative to an Oracle

Next let $G : \mathbb{N} \to \mathbb{N}$ be a ***unary total function***.

Let

$$W_n^G = \{x \in \mathbb{N} \mid \Phi_G^{(1)}(x, n) \text{ is defined}\}.$$

That is, W_n^G is the domain of the partial or total function $f : \mathbb{N} \to \mathbb{N}$ that is G-computed by the $\mathcal{S}$-program with a macro whose number is n.

Recall that

$$W_n = \{x \in \mathbb{N} \mid \Phi^{(1)}(x, n) \text{ is defined}\}.$$

That is W_n is the domain of the partial or total function $g : \mathbb{N} \to \mathbb{N}$ that is computed by the $\mathcal{S}$-program whose number is n.

Sets That are Computably Enumerable Relative to an Oracle

- Recall that, when encodings of $\mathcal{S}$-programs with oracles were defined, the value set to identify instructions of type

$$V \leftarrow O(V)$$

  was 0 — the same as the value set to identify instructions of type

$$V \leftarrow V$$

  when encodings of regular $\mathcal{S}$-programs were defined.

- This can be used to show that if $I : \mathbb{N} \to \mathbb{N}$ is the identity function — so that $I(n) = n$ for all $n \in \mathbb{N}$ — then $W_n^I = W_n$ for every number $n \in \mathbb{N}$.

Sets That are Computably Enumerable Relative to an Oracle

The following is a straightforward consequence of the definitions that have been given.

Theorem #17 (Relativized Enumeration Theorem): Let $G : \mathbb{N} \to \mathbb{N}$ be a unary total function and let $B \subseteq \mathbb{N}$. Then B is G-computably enumerable if and only if there is a number $n \in \mathbb{N}$ such that $B = W_n^G$.

Sets That are Computably Enumerable Relative to an Oracle

Finally, suppose that $S \subseteq \mathbb{N}$ — that is, S is a ***set*** instead of a total function. Then

$$W_n^S = \{x \in \mathbb{N} \mid \Phi_{p_S}^{(1)}(x, n) \text{ is defined}\}$$

where, as usual, $p_S : \mathbb{N} \to \{0, 1\}$ is the unary total predicate such that for all $n \in \mathbb{N}$,

$$p_S(n) = \begin{cases} 1 & \text{if } n \in S \\ 0 & \text{if } n \notin S \end{cases}$$

In other words, $W_n^S = W_n^G$ where G is the above total function $G = p_S$.