More about Reductions and Reducibilities

Proofs of Claims

This document includes proofs of claims that are stated, but not proved, in the notes for Lecture #15.

Theorem 1. $\preceq_1$ *is a reducibility.*

Proof. Since "$\preceq_1$" is a binary relation between subsets of $\mathbb{N}$ it is necessary and sufficient, to show that this relation satisfies the conditions given in the definition of a "reducibility" — that is, that $A \preceq_1 A$ for every set $S \subseteq \mathbb{N}$ and that if $A \preceq_1 B$ and $B \preceq_1 C$ for sets $A, B, C \subseteq \mathbb{N}$ then $A \preceq_1 C$ as well.

(a) The identity function $I : \mathbb{N} \to \mathbb{N}$ such that $I(n) = n$ for all $n \in \mathbb{N}$ is certainly a one-one computable total function. Since $x \in A$ if and only if $f(x) = x \in A$, for every number $x \in \mathbb{N}$, it follows that $A \preceq_1 A$, for every set $A \subseteq \mathbb{N}$, as required.

(b) Let $A, B, C \subseteq \mathbb{N}$ such that $A \preceq_1 B$ and $B \preceq_1 C$. Then there exist total, computable, one-one functions $f, g : \mathbb{N} \to \mathbb{N}$ such that

$$A = \{x \in \mathbb{N} \mid f(x) \in B\} \quad \text{and} \quad B = \{y \in \mathbb{N} \mid g(y) \in C\}.$$

Let $h : \mathbb{N} \to \mathbb{N}$ such that $h = g \circ f$ — that is, for all $n \in \mathbb{N}$,

$$h = g \circ f(n) = g(f(n)).$$

Then h is a total computable, one-one function such that

$$A = \{x \in \mathbb{N} \mid h(x) \in C\},$$

so that $A \preceq_1 C$, as claimed.

Since A, B and C were arbitrarily chosen subsets of $\mathbb{N}$ such that $A \preceq_1 B$ and $B \preceq_1 C$, it follows that "$\preceq_1$" also satisfies the second condition in the definition of a "reducibility", as needed to establish the claim.

□

Turing-reducibility was also defined, using the oracle computations introduced in the previous lecture.

Theorem 2. *$\preceq_T$ is a reducibility.*

Proof. Since "$\preceq_T$ is a binary relation between subsets of $\mathbb{N}$, it is necessary and sufficient to prove that this satisfies the conditions in the definition of a "reducibility": For all $A, B, C \subseteq \mathbb{N}$, $A \preceq_T A$, and if $A \preceq_T B$ and $B \preceq_T C$ then $A \preceq_T C$.

(a) The first property follows from Theorem #3 in the notes for Lecture #14, which stated that G is G-computable, for every total function $G : \mathbb{N} \to \mathbb{N}$.

(b) The second property follows from Theorem #5 in the notes for Lecture #14, which stated that if a function $f : \mathbb{N} \to \mathbb{N}$ is G-computable, for a total function $G : \mathbb{N} \to \mathbb{N}$, and if G is H-computable, for a total function $H : \mathbb{N} \to \mathbb{N}$, then f is H-computable. □

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 $A \preceq_T B$.

Proof. (a) Part (a) follows directly from the definitions of "many-one reduction" and "one-one reduction," because any one-one computable total function $f : \mathbb{N} \to \mathbb{N}$ is a computable total function.

(b) Suppose that $A \preceq_m B$, so that there exists a computable total function $f : \mathbb{N} \to \mathbb{N}$ such that

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

Consider the $\mathcal{S}$-program with an oracle that can be obtained from the following one by macro expansion, where p_A (respectively, p_B) is the characteristic function of A (respectively, of B). It should not be hard to see that this p_B-computes p_A.

$$\begin{aligned}&X_1 \leftarrow f(X_1)\\&X_1 \leftarrow O(X_1)\\&Y \leftarrow X_1\end{aligned}$$

It follows by the definition of a "Turing reduction" that $A \preceq_T B$, as needed to complete the proof. □

Recall that if $A \subseteq \mathbb{N}$ then

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

and that if $\mathcal{W}$ is a collection of subsets of $\mathbb{N}$ then ***co-$\mathcal{W}$*** is the collection

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

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-$\mathcal{W}$.

Proof. Let $B \in \mathcal{W}$. Then $B \preceq_Q A$, since A is Q-complete for $\mathcal{W}$. Since $A \in$ co-$\mathcal{W}$ and co-$\mathcal{W}$ is Q-closed, it follows that $B \in$ co-$\mathcal{W}$. Now, since B was arbitrarily chosen from $\mathcal{W}$ it follows that $\mathcal{W} \subseteq$ co-$\mathcal{W}$.

Next let $B \in$ co-$\mathcal{W}$. Then $\overline{B} \in \mathcal{W}$ (since the complement of the complement of any set $S \subseteq \mathbb{N}$ is the same set S). It follows by the above that $\overline{B} \in$ co-$\mathcal{W}$. *This* implies that $B \in \mathcal{W}$ and, since B was arbitrarily chosen from co-$\mathcal{W}$, it now follows that co-$\mathcal{W} \subseteq \mathcal{W}$ — and that $\mathcal{W} =$ co-$\mathcal{W}$, as claimed. □

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 $\overline{A} \preceq_1 \overline{B}$.

Proof. □

Proof. Let $A, B \subseteq \mathbb{N}$ such that $A \preceq_m B$. Then there exists a total computable function $f : \mathbb{N} \to \mathbb{N}$ such that

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

However, this implies that

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

so that $\overline{A} \preceq_m \overline{B}$, as claimed in part (a).

The proof of (b) is virtually identical: Here, f is a one-one computable total function instead of an arbitrary computable total function. □

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-$\mathcal{W}$.

Proof. Suppose that $\mathcal{W}$ is m-closed. Let $B \in$ co-$\mathcal{W}$ and let $A \subseteq \mathbb{N}$ such that $A \preceq_m B$. It is necessary and sufficient to prove that $A \in$ co-$\mathcal{W}$ as well, in order to establish part (a) of the claim.

By Theorem 5, $\overline{A} \preceq_m \overline{B}$. Since $\overline{B} \in \mathcal{W}$ and $\mathcal{W}$ is m-closed, $\overline{A} \in \mathcal{W}$. It follows that $A \in$ co-$\mathcal{W}$, as required.

The proofs of parts (b) and (c) are virtually identical the proof of part (a): $\preceq_m$ should simply be replaced by $\preceq_1$ and $\preceq_T$, respectively, throughout the argument. ☐

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

Proof. Let $A, B \subseteq \mathbb{N}$ such that $A \preceq_T B$ and the set B is computable. Then (by the definition of Turing reducibility) if p_A is the characteristic function of A, and p_B is the characteristic function of B, then p_A is p_B-computable. Since p_B is a computable total function, it follows by Theorem #6 from the previous lecture that p_A is computable, that is, the set A is computable. Since A and B were arbitrarily chosen sets such that $A \preceq_T B$ and B is computable, it follows that the set of computable subsets of $\mathbb{N}$ is T-closed.

Suppose next that either $A \preceq_m B$, or $A \preceq_1 B$, and B is computable. Then it follows by Theorem 3 that $A \preceq_T B$, and it follows by the above argument that A is computable too. Thus the set of computable subsets of $\mathbb{N}$ is both m-closed and 1-closed as well, as claimed. ☐

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

Proof. Let $A, B \subseteq \mathbb{N}$ such that $A \preceq_m B$ and B is computably enumerable. Since $A \preceq_m B$, there exists a computable total function $f : \mathbb{N} \to \mathbb{N}$ such that $x \in A$ if and only if $f(x) \in B$ for all $x \in \mathbb{N}$. Since B is computably enumerable, there exists a computable partial (or total) function $g : \mathbb{N} \to \mathbb{N}$ such that $B = \{x \in \mathbb{N} \mid g(x) \text{ is defined}\}$.

Let $h = g \circ f$. Then h is certainly a computable partial (or total) function, since f and g are, and it follows from the above that $A = \{x \in \mathbb{N} \mid h(x) \text{ is defined}\}$. Thus A is computably enumerable.

Since $A, B \subseteq \mathbb{N}$ were arbitrarily chosen such that $A \preceq_m B$ and B is computably enumerable, t follows that the set of computably enumerable subsets of $\mathbb{N}$ is m-closed.

If $A, B \subseteq \mathbb{N}$ and $A \preceq_1 B$ then $A \preceq_m B$. This can be used to argue that the set of computably enumerable subsets of $\mathbb{N}$ is 1 closed as well.

Finally, recall from Lecture #10, the set

$$K = \{n \in \mathbb{N} \mid \text{the } \mathcal{L}\text{-program whose encoding is } n \text{ halts when executed on input } n\}.$$

It is easy to establish that $\overline{K} \preceq_T K$ (Indeed, $\overline{S} \preceq_T S$ for every subset S of $\mathbb{N}$.) Since K is computably enumerable and $\overline{K}$ is *not* computably enumerable (as shown in Lecture #10), it follows that the set of computably enumerable subsets of $\mathbb{N}$ is *not* T-closed. $\square$

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

Proof. Let $A \subseteq \mathbb{N}$ be an arbitrarily chosen computably enumerable set. Since K is computably enumerable, it is necessary and sufficient to prove that $A \preceq_1 K$ in order to prove the claim.

Since A is computably enumerable,

$$A = \{n \in \mathbb{N} \mid f(n) \text{ is defined}\}$$

for some computable partial (or total) function $f : \mathbb{N} \to \mathbb{N}$. Let $g : \mathbb{N}^2 \to \mathbb{N}$ be a partial function such that $g(t, x) = f(x)$ for all $t, x \in \mathbb{N}$, so that the value of g does not depend on its first input: g is computable, since f is.

It follows by the Universality and Strengthened Parameter Theorems that there exists a number $e \in \mathbb{N}$ such that, for all $t, x \in \mathbb{N}$,

$$g(t, x) = \Phi^{(2)}(t, x, e) = \Phi(t, S_1^1(x, e)) \tag{1}$$

and that the function $S_1^1 : \mathbb{N}^2 \to \mathbb{N}$ is primitive recursive. Now

$$\begin{aligned} A &= \{n \in \mathbb{N} \mid f(n) \text{ is defined}\} \\ &= \{n \in \mathbb{N} \mid g(S_1^1(n, e), n) \text{ is defined}\} \quad \text{(recalling that } g \text{ does not depend on its first input)} \\ &= \{n \in \mathbb{N} \mid \Phi(S_1^1(n, e), S_1^1(n, e)) \text{ is defined}\} \qquad \text{(by the equation at line (1))} \\ &= \{n \in \mathbb{N} \mid S_1^1(n, e) \in K\}. \end{aligned}$$

Since the function $h : \mathbb{N} \to \mathbb{N}$ such that $h(n) = S_1^1(n, e)$ is a total computable function (indeed, it is primitive recursive), $A \preceq_m K$.

Furthermore, the Strengthened Parameter Theorem implies that if $S_1^1(n, e) = S_1^1(m, e)$ for $n, m, e \in \mathbb{N}$ then $n = m$. Thus h is also one-one — so that $A \preceq_1 K$, as required. $\square$

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.*

Proof. It follows by Theorem 8 that the class of recursively enumerable subsets on $\mathbb{N}$ is m-closed. It now follows by Corollary 6 that the class

$$\mathcal{W} = \{\overline{A} \subseteq \mathbb{N} \mid A \text{ is computably enumerable}\}$$

is m-closed as well. Now — in order to obtain a contradiction (and establish the claim) — suppose that $A \subseteq \mathbb{N}$ such that A is m-complete for the class of computably enumerable sets, and that $\overline{A}$ is computably enumerable.

Then (since A is also the complement of $\overline{A}$) $A \in \mathcal{W}$ and it follows by Theorem 4 that $\mathcal{W}$ is equal to the class of computably enumerable sets. We now have the desired ***contradiction*** — because the above set K is an example of a set that is computably enumerable but (provably) does not belong to $\mathcal{W}$.

It follows that if $A \subseteq \mathbb{N}$ and A is m-complete for the class of computably enumerable sets then $\overline{A}$ is not computably enumerable — which also implies that A is not computable, as needed to complete the proof. $\square$

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 $A \oplus B \preceq_T C$.

Proof. It is immediate from the above definition of $A \oplus B$ that the following $\mathcal{S}$-program with an oracle $(A \oplus B)$-computes the characteristic function of the set A:

$$\begin{aligned} &X_1 \leftarrow 2 \times X_1 \\ &X_1 \leftarrow O(X_1) \\ &Y \leftarrow X \end{aligned}$$

It now follows by the definition of a "Turing-reduction" that $A \preceq_T A \oplus B$, as claimed.

If the first statement in the above $\mathcal{S}$-program is replaced with the statement

$$X_1 \leftarrow 2 \times X_1 + 1$$

then this program computes the characteristic function of the set B, instead — implying that $B \preceq_T A \oplus B$ as well.

Suppose that $C \subseteq \mathbb{N}$ and that $A \preceq_T C$ and $B \preceq_T C$ — so that the characteristic functions p_A and p_B of A and B, respectively, are both C-computable. Consider the following — which includes pseudo-instructions for the computations of each of the C-computable predicates p_A and p_B.

$$
\begin{array}{ll}
 & \text{IF } (X_1 \bmod 2) = 0) \text{ GOTO } D_1 \\
 & X_1 \leftarrow \lfloor (X_1 - 1)/2 \rfloor \\
 & Y \leftarrow p_B(X_1) \\
 & \text{GOTO } E_1 \\
[D_1] & X_1 \leftarrow \lfloor X_1/2 \rfloor \\
 & Y \leftarrow p_A(X_1)
\end{array}
$$

This program computes $A \oplus B$. It follows that $(A \oplus B) \preceq_T C$ as well, as required to complete the proof. $\square$

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 (from the lecture notes). 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*

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

Proof. Suppose, first, that A is G-computably enumerable. Then there is a G-computable partial or total function $f : \mathbb{N} \to \mathbb{N}$ such that

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

Let z_0 be the number of an $\mathcal{S}$-program with an oracle that G-computes f — at least one such number must exist. Then

$$A = n \in \mathbb{N} \mid (\exists t)(\text{halted}_G^{(1)}(x, z_0, t) = 1)\}$$

where $\text{halted}_G^{(1)} : \mathbb{N}^3 \to \{0, 1\}$ is the G-computable total predicate whose properties are considered in Theorem #9 of the precious set of notes (the "Relativized Step-Counter Theorem").

It now suffices to set $Q : \mathbb{N}^2 \to \{0, 1\}$ to be the total predicate such that, for all $x, t \in \mathbb{N}$,

$$Q(x, t) = \text{halted}_G^{(1)}(x, z_0, t)$$

— which is also G-computable — in order to establish the existence of the predicate Q mentioned in the claim.

Now suppose, instead, that there exists a G-computable total predicate $Q : \mathbb{N}^2 \to \mathbb{N}$ such that

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

It is necessary and sufficient to use this to show that A is G-computably enumerable. With that noted, consider the following $\mathcal{S}$-program with an oracle — which includes a pseudo-instruction to compute the predicate Q:

$$\begin{array}{ll} [B_1] & Z_1 \leftarrow Q(X_1, Y) \\ & Y \leftarrow Y + 1 \\ & \text{IF } Z_1 = 0 \text{ GOTO } B_1 \end{array}$$

Let $h : \mathbb{N} \to \mathbb{N}$ be the partial or total function that is G-computed by this program. Then

$$A = \{x \in \mathbb{N} \mid h(x) \text{ is defined}\}$$

as needed to establish that A is G-computably enumerable and to complete the proof. □

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

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

Proof. Suppose first that A is G-computably enumerable. Then it follows by Theorem 15 that there exists a G-computable total predicate $Q : \mathbb{N}^2 \to \mathbb{N}$ such that

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

Let $B = \{z \in \mathbb{N} \mid Q(\ell(z), r(z)) = 1\}$. Then (since the class of G-computable total functions includes the set of primitive recursive functions, and is closed under composition) B is a G-recursive set. It now suffices to note that

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

in order to establish one part of the claim.

Suppose, next, that $A \subseteq \mathbb{N}$ and that there exists a G-computable set $B \subseteq \mathbb{N}$ such that

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

Since B is G-computable, the total predicate $Q : \mathbb{N}^2 \to \{0,1\}$ such that, for all $x, y \in \mathbb{N}$, $Q(x,y) = 1$ if and only if $\langle x, y\rangle \in B$ is a G-computable total predicate. Since

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

it now follows by Theorem 15 that A is G-computably enumerable — as needed to complete the proof of the claim. □