

Computer Science 513

Reducibilities — and Many-One Reductions

Instructor: Wayne Eberly

Department of Computer Science
University of Calgary

Lecture #11

Goals for Today

Goals for Today:

- Introduction to ***reducibilities*** and their properties.
- Introduction to ***many-one reductions*** — reductions between *sets* that are closely related to the kind of ***mapping reductions*** (possibly also called ***many-one reductions***) between *languages* that you — probably — learned about in CPSC 351 (or 313).
- Something, in a way, new: The introduction of the notion of ***completeness*** of a set, with respect to a class. This is something you have seen as a part of ***complexity theory*** (in CPSC 413) but probably not, before this, as part of ***computability theory***.

Reducibilities

Definition: 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 $A \preceq_Q C$.
- Various kinds of reducibilities — that are useful for proving that sets not computable, or not computably enumerable — will be introduced in CPSC 513.

Many-One Reductions

Definition: Let $A, B \subseteq \mathbb{N}$. A is **many-one reducible** to B , written as

$$A \preceq_m B,$$

if there exists a **Turing-computable total function** $f : \mathbb{N} \rightarrow \mathbb{N}$ such that

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

In other words: For all $x \in \mathbb{N}$, $x \in A$ if and only if $f(x) \in B$.

- You (probably) saw a similar kind of “many-one reduction” (possibly called a *mapping reduction*) between *languages* in CPSC 351 (or 313).

Many-One Reductions

Note that, “ $\preceq_m$ ” is a binary relation on subsets of $\mathbb{N}$.

Theorem #1: The binary relation “ $\preceq_m$ ” is a reducibility.

Proof: Exercise.



Many-One Reductions: An Example

Recall the set

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

- It was shown, in the previous lecture, that the set K is computably enumerable, but not computable.

Let

Total = $\{n \in \mathbb{N} \mid \text{the program whose encoding is } n$
computes a total function $f_n : \mathbb{N} \rightarrow \mathbb{N}$ (of one variable)}.

Many-One Reductions: An Example

Claim: $K \preceq_m \text{Total}$.

Proof:

- Let $\mathcal{P}$ be the $\mathcal{S}$ -program obtained from the following after macro expansion:

$$Y \leftarrow \Phi^{(1)}(X_2, X_2)$$

where the computable partial function $\Phi^{(1)} : \mathbb{N}^2 \rightarrow \mathbb{N}$ is as described in the previous lecture.

- Let p be the encoding of this program — so that $\#(\mathcal{P}) = p$.

Many-One Reductions: An Example

- Let $g : \mathbb{N}^2 \rightarrow \mathbb{N}$ be the function computed by $\mathcal{P}$. Since $\mathcal{P}$ ignores its first input, $g(x, z)$ is defined for *all* $x \in \mathbb{N}$ if and only if $g(z, z)$ is defined — for every $z \in \mathbb{N}$.
- Recall that, by the Parameter Theorem, there exists a primitive recursive function $S_1^1 : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that, for all $x_1, x_2 \in \mathbb{N}$,

$$g(x_1, x_2) = \Phi^{(2)}(x_1, x_2, p) = \Phi^{(1)}(x_1, S_1^1(x_2, p)).$$

- Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be the function such that, for all $x \in \mathbb{N}$, $f(x) = S_1^1(x, p)$. Then f is a primitive recursive function, so it is certainly a computable total function.

Many-One Reductions: An Example

It remains only to notice that, for all $z \in \mathbb{N}$,

$$\begin{aligned} z \in K &\iff \Phi^{(1)}(z, z) \text{ is defined} \\ &\iff \mathcal{P} \text{ halts on inputs } x \text{ and } z, \text{ for all } x \in \mathbb{N} \\ &\iff \Phi^{(1)}(x, S_1^1(z, p)) \text{ is defined for all } x \in \mathbb{N} \\ &\iff \Phi^{(1)}(x, f(z)) \text{ is defined for all } x \in \mathbb{N} \\ &\iff f(z) \in \text{Total}, \end{aligned}$$

so that $K \preceq_m \text{Total}$, as claimed.



Many-One Reductions: Another Example

Let

$\text{Empty} = \{n \in \mathbb{N} \mid \text{the program whose encoding on } n$
does not halt on any (single) input}\}.

Exercise: Use the function $f : \mathbb{N} \rightarrow \mathbb{N}$, used in the above proof, to prove that $\overline{K} \preceq_m \text{Empty}$.

Closure under a Reducibility

Definition: Let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$, and let $\preceq_Q$ be a reducibility. Then $\mathcal{W}$ is **closed under $\preceq_Q$** — or “**Q-closed**” — if, for all $A, B \subseteq \mathbb{N}$,

if $A \in \mathcal{W}$ and $B \preceq_Q A$ then $B \in \mathcal{W}$, for all $A, B \subseteq \mathbb{N}$.

- In particular, a collection $\mathcal{W}$ of subsets of $\mathbb{N}$ is **m-closed** if the following property is satisfied: For all sets $A, B \subseteq \mathbb{N}$, if $A \preceq_m B$ and $B \in \mathcal{W}$, then $A \in \mathcal{W}$.

Closure under a Reducibility

Theorem #2:

- (a) The collection of all ***computable*** subsets of $\mathbb{N}$ is m-closed.
- (b) The collection of all ***computably enumerable*** subsets of $\mathbb{N}$ is m-closed.

Closure under a Reducibility

Proof: Let $A \subseteq \mathbb{N}$ and let $B \subseteq \mathbb{N}$.

- (a) Suppose that $A \preceq_m B$ and B is computable — so that the characteristic function $p_B : \mathbb{N} \rightarrow \{0, 1\}$ of B is a computable total function.

Since $x \in A$ if and only if $f(x) \in B$, for all $x \in \mathbb{N}$, the characteristic function $p_A : \mathbb{N} \rightarrow \{0, 1\}$ is just the composition of functions $p_A = p_B \circ f$, that is, the function such that $p_A(x) = p_B(f(x))$ for all $x \in \mathbb{N}$.

Since p_B and f are both computable, p_A is certainly computable as well.

Since A and B are arbitrarily chosen subsets of $\mathbb{N}$, it follows that the collection of computable subsets of $\mathbb{N}$ is m-closed, as claimed in part (a).

Closure under a Reducibility

- (b) Now suppose that $A \preceq_m B$ and B is computably enumerable — so that B is the domain of some computable partial or total function $g : \mathbb{N} \rightarrow \mathbb{N}$.

Let $h = g \circ f : \mathbb{N} \rightarrow \mathbb{N}$, so that $h(x) = g(f(x))$ for all $x \in \mathbb{N}$. Note that h is a computable partial or total function as well. Now, for all $x \in \mathbb{N}$,

$$x \in A \Rightarrow f(x) \in B$$

$$\Rightarrow h(x) = g(f(x)) \text{ is defined,}$$

and

$$x \notin A \Rightarrow f(x) \notin B$$

$$\Rightarrow h(x) = g(f(x)) \text{ is undefined,}$$

so that A is the domain of h — and A is, therefore, computably enumerable.

Since A and B are arbitrarily chosen subsets of $\mathbb{N}$, it follows that the collection of computably enumerable subsets of $\mathbb{N}$ is m-closed, as claimed in part (b) □

Closure under a Reducibility

Corollary #3: Let $A \subseteq \mathbb{N}$ and let $B \subseteq \mathbb{N}$.

- (a) If $A \preceq_m B$ and A is **not** computable then B is **not** computable, either.
 - (b) If $A \preceq_m B$ and A is **not** computably enumerable then B is **not** computably enumerable, either.
- This follows immediately from the definition of “closure under a reducibility” and Theorem #2 — and its proof is left as an easy **exercise**.
 - Since we now know of sets that are provably not computable, and provably not computably enumerable, this provides a method — involving establishing a many-one reduction — to prove that other sets are not computable, and not computably enumerable, as well.

Closure under a Reducibility

Example Applications:

- It is proved. above, that $K \preceq_m \text{Total}$. Since it was proved, in the Lecture #10, that the set K is not computable, it follows by part (a) of Corollary #3 that the set “Total” is not computable, either.
- It is also noted, above, that $\overline{K} \preceq_m \text{Empty}$. Since it was proved, in Lecture #10, that the set $\overline{K}$ is not computably enumerable, it follows by part (b) of Corollary #3 that the set “Empty” is not computably enumerable, either.

Another Example of a Many-One Reduction

Let

$$\begin{aligned} K_0 &= \{x \in \mathbb{N} \mid \Phi^{(1)}(\ell(x), r(x)) \text{ is defined}\} \\ &= \{\langle x, y \rangle \mid \Phi_y^{(1)}(x) = \Phi^{(a)}(x, y) \text{ is defined}\}. \end{aligned}$$

- K_0 is computably enumerable — the program which executes our “universal S -program” on inputs $\ell(x)$ and $r(x)$ (for a given input $x \in \mathbb{N}$) computes a partial function that is defined on input x if and only if $x \in K_0$.

Another Example of a Many-One Reduction

Claim: Let $S \subseteq \mathbb{N}$ such that S is computably enumerable. Then $S \preceq_m K_0$.

Proof:

- Since S is computably enumerable, there exists some computable partial (or total) function $g : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $x \in \mathbb{N}$, $x \in S$ if and only if $g(x)$ is defined.
- Since g is computable, there exists an integer n such that $\mathcal{P}$ computes g , where $\mathcal{P}$ is the S -program such that $\#(\mathcal{P}) = n$.
- Let $f : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $x \in \mathbb{N}$,

$$f(x) = \langle x, n \rangle.$$

Note that f is a computable total function.

Another Example of a Many-One Reduction

- It follows that, for all $x \in \mathbb{N}$,

$$x \in S \iff g(x) \text{ is defined}$$

$$\iff \Phi^{(1)}(x, n) \text{ is defined} \quad (\text{by the choice of } n)$$

$$\iff \langle x, n \rangle \in K_0 \quad (\text{by the definition of } K_0)$$

$$\iff f(x) \in K_0 \quad (\text{by the definition of } f).$$

- Since f is a computable total function it follows that $S \preceq_m K_0$, as claimed. □

Completeness

Once again, let “ $\preceq_Q$ ” be a reducibility. Let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$.

Definition: A set $A \subseteq \mathbb{N}$ is ***Q-complete for $\mathcal{W}$*** if

- (a) $A \in \mathcal{W}$, and
- (b) $B \preceq_Q A$ for every subset $B \in \mathcal{W}$.

Note: This resembles, and (arguably) generalizes, the notion of “ $\mathcal{NP}$ -completeness” introduced in CPSC 413.

- It follows, by the results established above, that the set K_0 is m-complete for the class of computably enumerable subsets of $\mathbb{N}$.

Completeness

Theorem #4: Let “ $\preceq_Q$ ” be a reducibility, and let $\mathcal{W}_1$ and $\mathcal{W}_2$ be collections of subsets of $\mathbb{N}$ such that the following properties are satisfied:

- (a) $\mathcal{W}_1$ is Q-closed.
- (b) $\mathcal{W}_1 \subsetneq \mathcal{W}_2$.

If a set $A \subseteq \mathbb{N}$ such that A is Q-complete for $\mathcal{W}_2$, then $A \notin \mathcal{W}_1$.

Completeness

Proof: Suppose that “ $\preceq_Q$ ” is a reducibility and that $\mathcal{W}_1$ and $\mathcal{W}_2$ are collections of subsets of $\mathbb{N}$ satisfying properties (a) and (b) in the statement of the claim. Let $A \subseteq \mathbb{N}$ such that A is Q-complete for $\mathcal{W}_2$.

- Suppose, to obtain a contradiction, that $A \in \mathcal{W}_1$.
- Since $\mathcal{W}_1 \subsetneq \mathcal{W}_2$, there exists a subset B of $\mathbb{N}$ such that $B \in \mathcal{W}_2$ but $B \notin \mathcal{W}_1$.
- Since $B \in \mathcal{W}_2$, and A is Q-complete for $\mathcal{W}_2$, $B \preceq_Q A$.
- Since $A \in \mathcal{W}_1$, and $\mathcal{W}_1$ is Q-closed, this implies that $B \in \mathcal{W}_1$.
- However, $B \notin \mathcal{W}_1$, and this **contradiction** establishes that our only assumption must be false: $A \notin \mathcal{W}_1$. □

Completeness

Application: Let $\mathcal{W}_1$ be the collection of computable subsets of $\mathbb{N}$ and let $\mathcal{W}_2$ be the collection of computably enumerable subsets of $\mathbb{N}$. Consider the reducibility “ $\preceq_m$ ”.

- It follows by part (a) of Theorem #2 that $\mathcal{W}_1$ is m-closed.
- It was shown, in Lecture #10, that $\mathcal{W}_1 \subsetneq \mathcal{W}_2$.
- As noted above, K_0 is m-complete for $\mathcal{W}_2$.
- It now follows by Theorem #4 that $K_0 \notin \mathcal{W}_1$ — that is the set K_0 is not computable.

Completeness

Theorem #5: Let " $\preceq_Q$ " be a reducibility and let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$ such that $\mathcal{W}$ is Q-closed. Suppose that $A \subseteq \mathbb{N}$, $B \subseteq \mathbb{N}$, and the following properties are satisfied:

- (a) $A \preceq_Q B$.
- (b) A is Q-complete for $\mathcal{W}$.
- (c) $B \in \mathcal{W}$.

Then B is also Q-complete for $\mathcal{W}$.

Completeness

Proof: Let “ $\preceq_Q$ ” be a reducibility, let $\mathcal{W}$ be a collection of subsets of $\mathbb{N}$ such that $\mathcal{W}$ is Q-closed, and suppose that A and B are subsets of $\mathbb{N}$ that satisfy properties (a), (b) and (c) in the statement of the claim.

- Let C be a subset of $\mathbb{N}$ such that $C \in \mathcal{W}$. Then $C \preceq_Q A$, since A is Q-complete for $\mathcal{W}$.
- Since $A \preceq_Q B$ as well, and “ $\preceq_Q$ ” is a reducibility, it now follows that $C \preceq_Q B$.
- Since C was arbitrarily chosen from $\mathcal{W}$ it follows that $C \preceq_Q B$ for **every** subset C of $\mathbb{N}$ such that $C \in \mathcal{W}$.
- Since $B \in \mathcal{W}$, by property (c), it now follows that B is Q-complete for $\mathcal{W}$, as claimed.



Another Many-One Reduction

Once again, recall the set

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

Claim: $K_0 \preceq_m K$.

Proof:

- It is necessary, and sufficient, to describe a computable total function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $z \in \mathbb{N}$

$$z \in K_0 \iff f(z) \in K$$

— that is, such that, for all $x, y \in \mathbb{N}$ (with $z = \langle x, y \rangle$),

$$\begin{aligned} \Phi^{(1)}(x, y) \text{ is defined} &\iff \\ &\Phi^{(1)}(f(\langle x, y \rangle), f(\langle x, y \rangle)) \text{ is defined.} \end{aligned}$$

Another Many-One Reduction

- Let $\mathcal{P}$ be the $\mathcal{S}$ -program that is obtained, by macro expansion, from the following,

$$Y \leftarrow \Phi^{(1)}(\ell(X_2), r(X_2))$$

and let $p = \#(\mathcal{P})$. Then — by the definition of $\Phi^{(2)}$ — the value computed by $\mathcal{P}$, on inputs $x_1, x_2 \in \mathbb{N}$, is

$$\Phi^{(2)}(x_1, x_2, p).$$

Another Many-One Reduction

- Recall the ***Parameter Theorem***: For all positive integers n and m there exists a primitive recursive function

$$S_m^n : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$$

such that, for all $x_1, x_2, \dots, x_m, u_1, u_2, \dots, u_n, y \in \mathbb{N}$,

$$\begin{aligned} \Phi^{(m+n)}(x_1, x_2, \dots, x_m, u_1, u_2, \dots, u_n, y) = \\ \Phi^{(m)}(x_1, x_2, \dots, x_m, S_m^n(u_1, u_2, \dots, u_n, y)). \end{aligned}$$

- In particular, there exists a primitive recursive function $S_1^1 : \mathbb{N}^2 \rightarrow \mathbb{N}$ such that, for all $x_1, u_1, y \in \mathbb{N}$,

$$\Phi^{(2)}(x_1, u_1, y) = \Phi^{(1)}(x_1, S_1^1(u_1, y)).$$

Another Many-One Reduction

- Now let $f : \mathbb{N} \rightarrow \mathbb{N}$ such that, for all $z \in \mathbb{N}$,

$$f(z) = S_1^1(z, p).$$

Since S_1^1 is primitive recursive and p is a constant, f is a computable total function.

Another Many-One Reduction

- It now suffices to note that, for all $z \in \mathbb{N}$,

$$f(z) \in K \iff \Phi^{(1)}(f(z), f(z)) \text{ is defined} \\ \text{(by the definition of } K)$$

$$\iff \Phi^{(1)}(S_1^1(z, p), S_1^1(z, p)) \text{ is defined} \\ \text{(by the definition of } f)$$

$$\iff \Phi^{(2)}(S_1^1(z, p), z, p) \text{ is defined} \\ \text{(by the definition of } S_1^1)$$

$$\iff \Phi^{(1)}(\ell(z), r(z)) \text{ is defined} \\ \text{(by the definition of the program } \mathcal{P})$$

$$\iff z \in K_0 \quad \text{(by the definition of } K_0).$$

- It now follows that $K_0 \preceq_m K$, as claimed.



Completeness: Application of This Result

Now consider the reducibility " $\preceq_m$ " and the collection, $\mathcal{W}$, of computably enumerable subsets of $\mathbb{N}$. Consider, as well, the sets K_0 and K .

- As noted above, $K_0 \preceq_m K$.
- It has also been established, above, that K_0 is m-complete for $\mathcal{W}$.
- It was established in Lecture #10 that the set K is computably enumerable, that is, $K \in \mathcal{W}$.
- It now follows by Theorem #5 that K is m-complete for the collection, $\mathcal{W}$, of computably enumerable subsets of $\mathbb{N}$, as well.

Equivalence

Once again, let “ $\preceq_Q$ ” be a reducibility.

Definition: If $A, B \subseteq \mathbb{N}$ then A is ***Q-equivalent*** to B ,

$$A \equiv_Q B,$$

if $A \preceq_Q B$ and $B \preceq_Q A$.

Example: Consider the reducibility “ $\preceq_m$ ” and the sets K and K_0 .

- Since K is computably enumerable, and K_0 is m-complete for the collection of computably enumerable subsets of $\mathbb{N}$, $K \preceq_m K_0$.
- As shown above, $K_0 \preceq_m K$ as well.
- Therefore, $K \equiv_m K_0$.

Concluding Remarks

- Theorems #4 and #5 might remind you of results established in CPSC 413 concerning “polynomial-time many-one reductions” and the complexity classes $\mathcal{P}$ and $\mathcal{NP}$.
- Indeed, we will see that these theorems will allow use to use reducibilities, to prove things about the computability of sets, just as polynomial-time many-one reductions were used to establish results about the computational complexity of languages, in CPSC 413.

Concluding Remarks

- Note that, while we have shown that $K \preceq_m \text{Total}$, so that $K_0 \preceq_m \text{Total}$ as well, we *have not* shown that the set “Total” is m-complete for the set of computably enumerable sets, even though K_0 is — because we have not shown that “Total” is computably enumerable!
- We will see later that the set “Total” is **not** computably enumerable — so that it is also not m-complete for the collection of computably enumerable subsets of $\mathbb{N}$.
- Understanding relationships between sets that are not computable will be one of the objectives of this course in the weeks to follow.