Math 29: Reducibilities

May 4th, 2022

1 Solving Problems Using Other Problems

As we have already seen, every c.e. problem can be solved if we can solve the halting problem. The halting problem can be solved if we can solve any productive problem. This is an example of **reducing** some problem to another one. If we can solve the latter, then we can solve the former. A notion of **reducibility** is a a specific way that we think about reducing a problem to another one. As we shall see, some reducibilities are stronger than others.

We will discuss three main notions of reducibility in this course: m-reducibility, 1-reducibility, and Turing reducibility. We have already seen some examples of 1-reducibility when we discussed creative and simple sets. Turing reducibility will be the most important, which "correctly" captures our intuitive idea of what it means to be able to solve one problem if we can solve another.

2 1-reducibility

A set A is **one-one reducible** (1-reducible) to a set B if there is a total, computable, injective function f such that $n \in A$ if and only if $f(n) \in B$. In this case, we write $A \leq_1 B$, and if $B \leq_1 A$ as well, then we write $A \equiv_1 B$ and say A is 1-equivalent to B. The 1-degree of A is $\{B : A \equiv_1 B\}$.

We can rephrase some previous results using this terminology. We proved that, for any c.e. set C, $C \leq_1 K$. We proved that, for any productive set P, $K^c \leq_1 P$. Furthermore, for any creative set C, $K \leq_1 C$, and by combining with the previous, in fact $K \equiv_1 C$.

We also saw that, for any simple set S, $K \not\leq_1 S$. This final fact justifies the assertion that 1-reducibility is not the "correct" notion of reducibility for solving one problem through solving another: We mentioned that any effectively simple set can solve the halting problem, yet none of them are 1-equivalent to K. Thus we need to come up with a different definition.

3 m-reducibility

A set A is **many-one reducible** (m-reducible) to a set B if there is a total computable function G such that $n \in A$ if and only if $f(m) \in B$. As before, $A \equiv_m B$ when $A \leq_m B$ and $B \leq_m A$. Notice that if $A \leq_1 B$, then $A \leq_m B$ because 1-reduction adds an additional requirement to the function. Maybe it is the case that, while a priori different, these reductions in fact coincide. It turns out this is not true.

Given a set A, define B to be the set whose characteristic function is given by

$$\chi_B(n) = \begin{cases} 1 & \text{if } k! \leq n < (k+1)! \text{ and } \chi_A(k) = 1 \\ 0 & \text{otherwise} \end{cases}$$

That is, take A and fatten it up by duplicating each element into a finite block, with each block growing exponentially. Then $B \leq_m A$: simply send n to the k which defines its block. However, if A is not computable, then we need not have $B \leq_1 A$.

Recall that f dominates g if it is larger than g on all but finitely many inputs.

Lemma 1. *Let A be a set whose principle function dominates all computable functions, and let B be defined from A as above. Then $B \not\leq_1 A$.*

Proof: Consider the principle function of B. Notice that if $k \in A$ and $k > 0$, then $p_B(k) < (k+1)!$. Now suppose for the sake of contradiction that $B \leq_1 A$ with witness f. But then consider $g(n) = max_{k \leq n}(f((n+1)!)) + 1$. Clearly g is a computable function. However, if $n \in A$, then $p_B(n) < (n+1)!$, so $\{f(k) : k \leq (n+1)!\}$ contains at least $n+1$ elements of A. (Those coming from $f(p_B(0))$, $f(P_B(1))$, $\ldots$, $f(p_B(n))$.

Therefore $p_A(n+1) < g(n+1)$. But this is a contradiction of the fact that p_A dominates every computable function, as p_A does not dominate g. Therefore no such f can exist, and $B \not\leq_1 A$.

Thus m-reducibility is not the same as 1-reducibility. However, it falls prey to the same problem we had with 1-reducibility: when proving that K being 1-reducible to a c.e. set C implies that C is creative, we did not use injectivity of f in proving that $s \circ f$ was a productive function. So even if f is simply total computable, $s \circ f$ is still a productive function for C^c. Therefore, $K \leq_m C$ for c.e. C if and only if C is creative. Therefore, m-reducibility still does not capture what we are looking for.

4 Computable Isomorphism

A **computable permutation** is a total, computable bijection $\pi : \omega \to \omega$. Two sets A and B are computably isomorphic if there is a computable bijection π such that

$$\pi(A) = \{n : k \in A \text{ and } \pi(k) = n\} = B$$

In such an instance, we write $A \equiv B$.

The following theorem, known as **Myhill's Isomorphism Theorem**, is essentially a computable analog of the Cantor-Schröder-Bernstein theorem, which says that two sets have the same cardinality if and only if there are injections from one to the other in both directions.

Theorem 2. *$A \equiv_1 B$ if and only if $A \equiv B$.*

Proof: The right-to-left direction is immediate, as π and π^{-1} are both total computable injections.

Suppose $A \equiv_1 B$ as witnessed by f and g, i.e. f reduces A to B and g reduces B to A. We define a computable permutation π in stages. That is, we will build finite sets $X_s \subset A$ and $Y_s \subset B$ such that π_s will provide a permutation between X_s and Y_s. As always, we start with $\pi_0 = X_0 = Y_0 = \emptyset$.

Given π_{2s}, X_{2s}, and Y_{2s}, complete stage $2s+1$ as follows. If $\pi(s)$ has already been defined at a previous stage, i.e. if $s \in X_{2s}$ then continue to stage $2s+2$ without doing anything. If it has not, compute $f(s)$. If $f(s) \notin Y_{2s}$, then set $X_{2s+1} = X_{2s} \cup \{s\}$, $Y_{2s+1} = Y_{2s} \cup \{f(s)\}$, and $\pi_{2s+1} = \pi_{2s} \cup \{\langle s, f(s)\rangle\}$. If $f(s) \in Y_{2s}$, then apply π_{2s} back and forth to find $x \in X_{2s}$ such that $f(x) \notin Y_{2s}$. Then set $X_{2s+1} = X_{2s} \cup \{x\}$, $Y_{2s+1} = Y_{2s} \cup \{f(x)\}$, and $\pi_{2s+1} = \pi_{2s} \cup \{\langle x, f(x)\rangle\}$.

For stage $2s+2$ given stage $2s+1$, do the same thing but define $\pi^{-1}_{2s+2}(s)$ in terms of g using Y_{2s+1} rather than f using X_{2s}.

Let $\pi = \bigcup_{s\in\omega} \pi_s$. π is computable because f and g are. It is a total bijection because we define $\pi(s)$ and $\pi^{-1}(s)$ by stage $2s+2$. Finally, suppose $n \in A$. If $\pi(n)$ was defined using the first case, then $\pi(n)$ was the end of a chain of the form $f(g(\dots g(f(n))\dots)$. Then by the properties of f, $n \in A$ if and only if $\pi(n) \in B$ because f preserves "A-ness" to "B-ness" and g preserves "B-ness" to "A-ness." A similar argument ensures the same if $\pi(n)$ is defined via the second case.