

CDM

Degrees and Completeness

Klaus Sutner
Carnegie Mellon University
www.cs.cmu.edu/~sutner

Completeness

1

Battleplan

- Comparing Problems
- Tractability
- Classical vs. Feasible
- Polynomial Time Reductions
- Cook's Theorem

Completeness

2

Comparing Problems

It is well-known in the theory of algorithms that some computational problems can be reduced to others: an algorithm for problem A can be obtained (easily) given an algorithm for problem B .

Example 1. *The problem of finding a matching in a bipartite graph can be reduced to solving a network flow problem.*

More generally, we are interested in algorithms for problem A that may use certain subroutines providing solutions for problem B .

In general, the algorithm may make any number of calls to the subroutine for B to obtain solutions for a number of different instances. The special case where only one call is permissible will turn out to be particularly interesting.

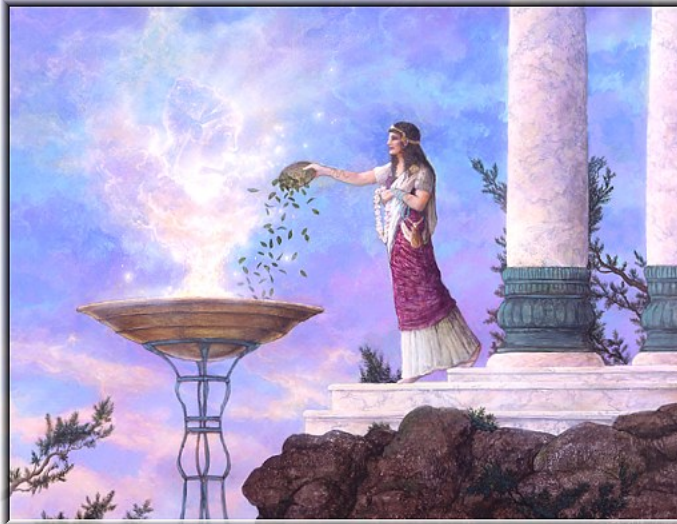
Completeness

3

Subroutines?

Completeness

4



Completeness

5

Subroutine sounds like Basic or Fortran, let's use oracles instead.

Oracle Algorithms

Suppose we have two computational problems A and B . We are given some input x for problem A .

- We compute away happily, producing some intermediate instance z for problem B .
- Instead of calling a subroutine for B , we ask the oracle for the solution $sol(z)$ for z .
- We are given the solution and continue with our computation, possibly asking for more help from the oracle, until finally we have the solution for x .

Note that we are not trying to describe any kind of practical algorithm at this point, we are only trying to compare the relative strength of problems A and B .

Intuitively, problem B is hard if many problems A can be solved using an algorithm with oracle B .

Turing Reductions

Here is the crucial definition that allows us to compare computational problems.

Definition 1. A is **Turing reducible** to B iff a solution for A can be computed using B as an oracle.

In symbols, $A \leq_T B$.

In other words, our "algorithm" gets solutions for instances for B for free.

The terminology goes back to Turing: he used oracle Turing machines.

Formally, we should think of A and B as languages here (or, if you prefer, sets of natural numbers). In particular for decision problems we use the Yes-instances to determine their difficulty.

A Quasi-Order

It is obvious from the definitions that \leq_T is reflexive. A little argument shows that this relation is also transitive.

Proposition 1. *Turing reducibility is a quasi-order (reflexive and transitive):*

- $A \leq_T A$.
- $A \leq_T B \leq_T C$ implies $A \leq_T C$.

Proof.

Every call to oracle B in the algorithm for A can be replaced by a computation which uses calls to oracle C .

The actual computation can be absorbed by the algorithm, so that all that remains are the calls to oracle C . □

Turing Degree

So we can lump together problems that are mutually reducible.

Definition 2. *Two sets A and B are Turing equivalent, whenever as*

$$A \leq_T B \wedge B \leq_T A$$

This defines an equivalence relation \equiv_T whose equivalence classes are called Turing degrees.

Notation: $\deg_T(A)$ is the Turing degree of A .

Turing degrees are also called degree of unsolvability since they measure the distance between a problem and solvability.

Note that we can extend the quasi-order \leq_T to a partial order on Turing degrees.

The order on these degrees is a measure of the information content of the set: the higher up in the order, the more information.

Degrees of Unsolvability

At the bottom level of this classification are the decidable sets. Note that a decidable oracle is useless in the sense that it could be replaced by an actual algorithm. More precisely, $A \leq_T B$ where B is decidable implies that A is also decidable. Hence

$$\deg_T(\emptyset) = \text{decidable sets.}$$

But the degree of the Halting problem is distinct

$$\deg_T(\emptyset) < \deg_T(K)$$

meaning that every set in the first degree is Turing reducible to every set in the second degree, but not conversely.

More Degrees

Yet higher degrees are obtained by looking at yet more complicated decision problems. A good source for such problems are questions related to c.e. sets.

Recall that Kleene's Normal Form theorem provides us with an enumeration

$$W_e = \text{domain of } \{e\}$$

of all c.e. sets.

Here are some interesting classes of c.e. sets.

$$\text{FIN} = \{ e \in \mathbb{N} \mid W_e \text{ is finite} \}$$

$$\text{INF} = \{ e \in \mathbb{N} \mid W_e \text{ is infinite} \}$$

$$\text{TOT} = \{ e \in \mathbb{N} \mid W_e = \mathbb{N} \}$$

$$\text{REC} = \{ e \in \mathbb{N} \mid W_e \text{ is decidable} \}$$

Higher Degrees of Unsolvability

Then

$$\deg_T(\emptyset) < \deg_T(K) < \deg_T(\text{FIN}) < \deg_T(\text{REC})$$

Also,

$$\deg_T(\text{FIN}) = \deg_T(\text{INF}) = \deg_T(\text{TOT})$$

Exercise 1. Prove the last assertion: FIN, INF and TOT are all Turing-equivalent.

Non-Closure

But note that the degree of K , a semi-decidable set, contains sets that fail to be semi-decidable. The problem is that by the very definition of Turing reducibility we have

$$\mathbb{N} - A \leq_T A$$

so that the complement of K is trivially in $\deg_T(K)$.

This indicates that Turing reducibility is a bit too powerful in the realm of semi-decidable sets. We will need to modify our notion of reducibility to deal with this issue.

Intermediate Sets

But first a peculiar phenomenon. Let's say that a degree is c.e. if it contains at least one c.e. set. So $\deg_T(\emptyset)$ and $\deg_T(K)$ are both c.e..

Experience has shown over the years that if any natural decision problem is semi-decidable it turns out to lie in one of those two degrees:

- It is either decidable, or
- as hard as the Halting Problem.

Note that this is not a theorem, just an observation.

Any decision problem invented by a mathematician (as opposed to a logician) falls into one of these two cases (provided it is not too complicated altogether).

Why?

Emil Post and Friedberg/Muchnik

The question whether there are any intermediate c.e. degrees was first posed by E. Post in 1944 in a seminal paper.

It turned out to be a very hard question, for a decade no one was able to come up with an answer.

Then, surprisingly, two people found the solution almost simultaneously: R. M. Friedberg (an undergraduate) in the US and A. A. Muchnik in Russia (they published their results in 1957 and 1956, respectively).

Theorem 1. *Friedberg, Muchnik*
There are intermediate c.e. degrees.

What is even more surprising, they used essentially the same method!

Alas, the construction is a bit too complicated for us, see the website for notes that give a complete proof of the Friedberg/Muchnik theorem.

Weaker Reductions

As already mentioned, when dealing with semi-decidable sets it is a nuisance that the Turing degree of semi-decidable set contains non semi-decidable sets (except for the trivial case $\deg_T(\emptyset)$).

Here are less powerful reductions that are better behaved.

Definition 3. A is **many-one reducible** to B if there exists a recursive function $f : \mathbb{N} \rightarrow \mathbb{N}$ such that

$$x \in A \iff f(x) \in B.$$

If function f is in addition injective then A is **one-one reducible** to B .

In symbols: $A \leq_m B$ and $A \leq_1 B$.

In other words: we can only ask a single question of the oracle, and whatever answer the oracle returns is also our answer. One-one reductions in addition are required to ask different questions to the oracle for different inputs.

Downward Closure

Proposition 2.

$$A \leq_1 B \text{ implies } A \leq_m B \text{ implies } A \leq_T B$$

The opposite implications are all false, but it requires a bit of effort to separate many-one and one-one reductions.

Proposition 3. Let \leq be \leq_m or \leq_1 .

- $A \leq_m B$ and B decidable implies that A is decidable.
- $A \leq_m B$ and B semi-decidable implies that A is semi-decidable.

Proof. This is easy to see from the Σ_1 and Δ_1 definitions of semi-decidable and decidable sets. \square

Lower Bounds

We can use reductions to provide lower bounds: $A \leq_m B$ certainly means that B is at least as complicated as A .

Lemma 1. $K \leq_m \text{INF}$ and $K \leq_m \text{TOT}$

Proof. There is a recursive function f such that

$$\{f(e)\}(x) = \begin{cases} \text{if } e \in K, \\ \text{otherwise.} \end{cases}$$

But then $K \leq_m \text{TOT}, \text{INF}$.

□

Hence neither INF nor TOT can be decidable. Of course, that's a bit lame, they are much worse than just undecidable.

Many-One and One-One Degrees

As before with Turing reductions one can collect mutually reducible sets into a degree.

To this end let $A \equiv_m B$ if $A \leq_m B \wedge B \leq_m A$ and let $A \equiv_1 B$ if $A \leq_1 B \wedge B \leq_1 A$.

Proposition 4. \equiv_m and \equiv_1 are equivalence relations.

The equivalence classes are correspondingly called *many-one degrees* and *one-one degrees*.

Many-one and one-one degrees provide a finer partition than Turing degrees.

Complicate Semi-Decidable Sets

Many-one and one-one degree give a partial order with the nice property that everything lying below a c.e. set is also c.e..

The most complicated c.e. set we know so far is K , the Halting Set.

What is its position in this partial order for many-one or one-one degrees?

One might suspect that K is fairly high up. In fact, it (more precisely: its degree) might be a largest element in the order.

This is captured in the next definition.

Completeness

Definition 4. A set $C \subseteq \mathbb{N}$ is (*many-one, one-one*) **complete** if C is c.e. and for all A c.e.: $A \leq_m C$ (or $A \leq_1 C$).

Note that it is entirely unclear whether a complete set exists: in a sense, a complete set has to contain information about all c.e. sets. It is not hard to construct such a set in terms of pure set theory, but we need one that is itself just c.e..

Lemma 2. The Halting set K is one-one complete.

Proof.

There is a primitive recursive function f such that

$$\forall z \{f(e, x)\}(z) \downarrow \iff \{e\}(x) \downarrow$$

But then $x \in W_e \iff f(e, x) \in K$. It is not hard to make sure the f is also injective.

□

Variants

One can define variants of K that are easily seen to lie in the same one-one degree – and are thus really the same as K from the point of view of information content.

$$K_0 = \{ \langle e, x \rangle \mid x \in W_e \}$$

$$K_1 = \{ e \mid W_e \neq \emptyset \}$$

Here $\langle e, x \rangle$ is understood to be one of the standard pairing functions.

Proposition 5. K, K_0 and K_1 are all one-one equivalent.

Proof.

The one-one equivalence of K and K_0 is taken care of in the last lemma.

For K and K_1 essentially the same function works.

□

Myhill's Isomorphism Theorem

One-one equivalent sets are very similar indeed as the next theorem shows.

Definition 5. Two sets $A, B \subseteq \mathbb{N}$ are **recursively isomorphic** if there is a recursive permutation p of \mathbb{N} such that $p(A) = B$.

In symbols: $A \equiv B$.

There is an analogue to the Schröder-Bernstein theorem that associates the existence of computable injections in both directions with the existence of a computable bijection.

Theorem 2.

$$A \equiv B \iff A \equiv_1 B.$$

In other words, the one-one degrees are simply obtained by applying a recursive permutation to the given set.

Thus K, K_0 and K_1 can all be obtained from each other by recursive permutations of \mathbb{N} .

Proof Sketch

Suppose $A \leq_1 B$ via f and $B \leq_1 A$ via g . Define a new functions h in stages using a zig-zag construction: $h = \bigcup h_\sigma$ where h_σ is finite.

$\sigma = 0$: $h_0 = \emptyset$

$\sigma > 0$, even:

Assume that h_σ is injective and $\forall x \in \text{dom } h_\sigma (x \in A \iff h_\sigma(x) \in B)$.

Define h on $x = (\sigma + 1)/2$.

If $h_\sigma(x) \downarrow$ do nothing, otherwise compute $f(x), f \circ h_\sigma^{-1} \circ f(x), f \circ (h_\sigma^{-1} \circ f)^2(x), \dots$. As f and h_σ are injective there can be no repetitions in this sequence.

Hence for some i : $y = f \circ (h_\sigma^{-1} \circ f)^i(x) \notin \text{rg } h_\sigma$. Set $h_{\sigma+1}(x) = y$.

$\sigma > 0$, odd:

Similar, exchange f, h_σ by g, h_σ^{-1} . □

The Lattice of CE Sets

As we have seen, algebra is often the great simplifier – is there an algebraic angle to semi-decidability? Sure, we can think of the collection of all c.e. sets as an algebraic structure

$$\mathcal{E} = \langle \text{c.e. sets}, \cup, \cap, \mathbf{0}, \mathbf{1} \rangle.$$

It is easy to see that \mathcal{E} is a distributive lattice with least and greatest element.

What properties of c.e. sets can be expressed in terms of this lattice?

Proposition 6. *Decidability is definable over \mathcal{E} .*

To see this recall that A is decidable iff both A and its complement are c.e.. So A is complemented in the lattice \mathcal{E} :

$$\mathcal{E} \models \exists X (X \cap A = \emptyset \wedge X \cup A = \mathbb{N}).$$

Indeed, the decidable sets form a Boolean subalgebra of \mathcal{E} .

Finiteness

Less obvious is that we can also define finiteness over \mathcal{E} :

A is finite iff

$$\mathcal{E} \models \forall X (X \subseteq A \implies X \text{ decidable}).$$

Overall, the structure of \mathcal{E} is rather complicated. While Π_2 sentences are decidable, in general there is no algorithm to test the validity of assertions about \mathcal{E} .

Theorem 3. *The theory of \mathcal{E} is undecidable.*

So the great simplifier doesn't quite work in this case.

How About Completeness?

Decidable sets are easy to recognize in \mathcal{E} . How about complete sets.

More narrowly, what is the role of K in this lattice?

Note that K, K_0 and K_1 are all images of each other under automorphisms of the lattice (the automorphisms induced by the recursive permutations of \mathbb{N} which exist by Myhill's theorem), so they are all essentially the same.

So we are looking for structural properties of K in terms of the lattice \mathcal{E} , not the fact that K has a definition that involves universal computation and the like.

This turns out to be a very difficult question leading to open problems, but we still can get a little mileage out it.

Productive Sets

K is not complemented in \mathcal{E} . So for any c.e. set W we have

$$W \subseteq \overline{K} \text{ implies } \exists x (x \in \overline{K} - W).$$

This property is definably over \mathcal{E} and motivates the next, stronger definition (which abandons \mathcal{E}).

Definition 6. *A set P is productive if there is a partial recursive function p such that $W_e \subseteq P \implies p(e) \in P - W_e$.*

So p computes a witness for the fact that $W_e \not\subseteq P$.

Note that \overline{K} is productive with trivial witness function $f(x) = x$.

Clearly, no c.e. set with a productive complement can be decidable.

Creative Minds

Definition 7. *E. Post, 1944*

A c.e. set C is creative if its complement is productive.

Since creative sets are undecidable, Post suggested with respect to membership questions for c.e. sets that clever thinking is indispensable.

The conclusion is inescapable that even for such a fixed, well-defined body of mathematical propositions, mathematical thinking is, and must remain, essentially creative.

Be that as it may, creative sets are crucial to the understanding of of one-one completeness.

Improving Productive Functions

Lemma 3. *Let P be productive. Then there is a productive function p for P that is total and injective.*

Proof. Let q be an arbitrary productive function for P . In the first step transform q into a total function as follows. Define h by

$$W_{h(x)} = \begin{cases} W_x & \text{if } q(x) \downarrow, \\ \emptyset & \text{otherwise.} \end{cases}$$

Then define q' by

$$q'(x) = \begin{cases} q(x) & \text{if } q(x) \downarrow \text{ before } q(h(x)), \\ q(h(x)) & \text{otherwise.} \end{cases}$$

Note that $q(x)$ or $q(h(x))$ must converge since $q(x) \uparrow \Rightarrow W_{h(x)} = \emptyset \subseteq P$. Hence q' is total.

Proof, contd.

Now let $W_{g(x)} = W_x \cup \{q'(x)\}$. Define $p(0) = q'(0)$.

For $x > 0$ determine $p(x)$ by enumerating $q'(x)$, $q'(g(x))$, $q'(g^2(x)) \dots$. If a repetition occurs, i.e., $q'(g^i(x)) = q'(g^j(x))$, then $W_x \not\subseteq P$ and we may set $p(x) = \mu(z \mid z \notin \{p(0), \dots, p(x-1)\})$.

But otherwise for some i we have $q'(g^i(x)) \notin \{p(0), \dots, p(x-1)\}$ and we may set $p(x) = q'(g^i(x))$ where i is minimal such.

□

So we may safely assume that the witness function of any productive set is total and injective.

This will be used in the next theorem.

Creative is Complete

Theorem 4. *If P is productive then $\mathbb{N} - K \leq_1 P$. Hence any creative set is \leq_1 -complete.*

Proof. Let p be a total injective productive function for P . Define f recursive by

$$W_{f(x,e)} = \begin{cases} \{p(x)\} & \text{if } e \in K, \\ \emptyset & \text{otherwise.} \end{cases}$$

By the recursion theorem there is a recursive injective function ω such that $W_{\omega(e)} = W_{f(\omega(e),e)}$.

To see this let $F(e, e_0, z) = \{f(e, e_0)\}(z)$. By the recursion theorem for any fixed e_0 there exists some e^* such that $F(e^*, e_0, -) = \{f(e^*, e_0)\} = \{e^*\}$. In fact, our proof shows that e^* can be computed from e_0 , i.e., $\omega(e_0) = e^*$ is a recursive function and furthermore injective.

Hence $W_{\omega(e)} = \{p(\omega(x))\}$ if $e \in K$ and $W_{\omega(e)} = \emptyset$ otherwise.

Proof, contd.

Now suppose $e \in K$. Then $W_{\omega(e)} \not\subseteq P$ because otherwise the witness constructed by p would be in $W_{\omega(e)}$. Hence $p(\omega(x)) \notin P$.

On the other hand if $e \notin K$ then $\emptyset = W_{\omega(e)} \subseteq P$, whence $p(\omega(x)) \in P$.

Therefore $e \in K \Leftrightarrow p(\omega(x)) \notin P$.

□

Theorem 5. *Let $C \subseteq \mathbb{N}$ be c.e. The following are equivalent:*

1. C is many-one-complete,
2. C is one-one-complete,
3. C is creative.

Proof. (3) \Rightarrow (2) by theorem ?? (2) \Rightarrow (1) is trivial. (1) \Rightarrow (3) since K is creative and for C m-o-complete we have $K \leq_m C$.

□

Summary

- We can compare problems using oracle Turing machines. This produces Turing degrees, or degrees of unsolvability.
- A classical result is the existence of intermediate c.e. degrees (solution to Post's Problem by Friedberg/Muchnik).
- Finer measures of complexity are given by many-one reductions and one-one reductions.
- One-one degrees are closely connected to computable permutations of \mathbb{N} .
- With respect to one-one reductions there are complete sets in the class of c.e. sets.
- There is a nice characterization of one-one-complete sets in terms of creative sets.