CSC363 Tutorial #5

Oracles and Reductions

February 15, 2023

Things covered in this tutorial

- ⋆ What is an oracle?
- ⋆ In an alternate universe, what are some impossible problems?
- ⋆ What is a (Turing) reduction?
- ⋆ How do you show that one problem is harder than another problem?
- ⋆ How can I show that something is uncomputable without a proof by contradiction?

Oracles: The Scenario

Task: Show that $\mathrm{HP} = \{x : M_x(x) \text{ halts}\}$ is computable.

Oracles: The Scenario

~~**Task:** Show that $\mathrm{HP} = \{x : M_x(x) \text{ halts}\}$ is computable.~~

HP is not computable!

Oracles: The Scenario

~~**Task:** Show that $\mathrm{HP} = \{x : M_x(x) \text{ halts}\}$ is computable.~~

HP is not computable! *But what if it were computable...*

Oracles: The Scenario

An advanced alien civilization discovers a way to decide whether $x \in \mathrm{HP}$ or not. They build a decider for HP called the oracle machine.

Oracles: The Scenario

An advanced alien civilization discovers a way to decide whether $x \in \mathrm{HP}$ or not. They build a decider for HP called the oracle machine.

Not sponsored

Oracles: The Scenario

An advanced alien civilization discovers a way to decide whether $x \in \mathrm{HP}$ or not. They build a decider for HP called the oracle machine.

Not sponsored

After feeding any x into the oracle machine, the oracle machine will tell you whether $x \in \mathrm{HP}$ or not, in $O(1)$ time!

Oracles: The Scenario

As a special offer, the aliens give you a copy of their oracle machine. (In return, you give the aliens permission to use humans as live biological test subjects).

Oracles: The Scenario

As a special offer, the aliens give you a copy of their oracle machine. (In return, you give the aliens permission to use humans as live biological test subjects).

Oracles: The Scenario

As a special offer, the aliens give you a copy of their oracle machine. (In return, you give the aliens permission to use humans as live biological test subjects).

We can now decide HP in $O(1)$ time! A very good deal.

However, you do not have access to the internals of the oracle machine. The oracle explodes upon any attempt to breach its case.

Oracles: The Scenario

$$\mathrm{HP} = \{x : M_x(x) \text{ halts}\}.$$

We can now decide HP in $O(1)$ time! A very good deal.

Question: Now that we have the oracle machine, which previously uncomputable languages become computable?

Oracles: The Scenario

$$\mathrm{HP} = \{x : M_x(x) \text{ halts}\}.$$

With an oracle to HP,

$$\mathrm{HP}_2 = \{(x, y) : M_x(y) \text{ halts}\}$$

becomes decidable:

Oracles: The Scenario

$$\mathrm{HP} = \{x : M_x(x) \text{ halts}\}.$$

With an oracle to HP,

$$\mathrm{HP}_2 = \{(x, y) : M_x(y) \text{ halts}\}$$

becomes decidable:

```
HP2_DECIDER(x, y):
  def M(z):
    ignore z
    run M_x(y)  # x, y are constants inside of M
    halt

  e = turing_machine_number(M)
  if e in HP: # call the oracle
    accept
  reject
```

Oracles: The Scenario

```
HP2_DECIDER(x, y):
  def M(z):
    ignore z
    run M_x(y) # x, y are constants inside of M
    halt

  e = turing_machine_number(M)
  if e in HP: # call the oracle
    accept
  reject
```

Question: Why does `HP2_DECIDER` always halt? Doesn't `HP2_DECIDER` call `run M_x(y)`, which might loop?

Oracles: The Scenario

```
HP2_DECIDER(x, y):
  def M(z):
    ignore z
    run M_x(y) # x, y are constants inside of M
    halt

  e = turing_machine_number(M)
  if e in HP: # call the oracle
    accept
  reject
```

Question: Why does `HP2_DECIDER` always halt? Doesn't `HP2_DECIDER` call `run M_x(y)`, which might loop?

Answer: Look carefully! `HP2_DECIDER` never calls M, so it doesn't call `run M_x(y)`.

Oracles: The Scenario

```
HP2_DECIDER(x, y):
  def M(z):
    ...

  ...
```

Useful Trick! You can construct M without running it.

Oracles: The Scenario

$$\mathrm{HP} = \{x : M_x(x) \text{ halts}\}.$$

Task: Given access to an oracle machine for HP, show that the following sets are decidable.

- ⋆ $S = \{p : p \text{ is prime}\}$.
- ⋆ $\overline{\mathrm{HP}} = \{x : M_x(x) \text{ loops}\}$.
- ⋆ $\mathrm{A}_{\mathrm{TM}} = \{(M, w) : M \text{ is a TM that accepts } w\}$.

Even harder languages than HP?

Suppose we have access to the oracle machine for HP. Are there any undecidable languages?

Even harder languages than HP?

Suppose we have access to the oracle machine for HP. Are there any undecidable languages?

Yes. Just like how every TM is assigned a number, each "HP-oracle TM" (M^{HP}) can still be assigned a natural number.

$$\mathrm{HP}' = \{x : M_x^{\mathrm{HP}}(x) \text{ halts}\}$$

is undecidable.

Even harder languages than HP?

Proof: Suppose HP' had a decider D that may access the oracle machine. Build the following HP-oracle TM T:

```
T(x):
  run D(x) # D may consult HP oracle
  if D accepts:
    loop
  else:
    accept
```

Since T consults the HP-oracle, $T = M_e^{\mathrm{HP}}$ for some $e \in \mathbb{N}$.

Question: Does $T(e)$ halt?

Even harder languages than HP?

Question: Suppose a more advanced alien civiliaztion built an oracle for HP'. Are there any undecidable languages left?

Even harder languages than HP?

Question: Suppose a more advanced alien civiliaztion built an oracle for HP'. Are there any undecidable languages left?

Answer: Yes.

$$\mathrm{HP}'' = \{x : M_x^{\mathrm{HP}'}(x) \text{ halts}\}$$

is undecidable.

Even harder languages than HP?

Question: Suppose a more advanced alien civiliaztion built an oracle for HP'. Are there any undecidable languages left?

Answer: Yes.

$$\mathrm{HP}'' = \{x : M_x^{\mathrm{HP}'}(x) \text{ halts}\}$$

is undecidable.

The $'$ operator is called the Turing jump: given any language L, L' is "even more undecidable" than L.

Reductions

Let A, B be languages. We say that A **Turing reduces to** B (denoted $A \leq_T B$) when:

Given an oracle for B, one can build a decider for A.

"$A \leq_T B$" should be interpreted as: A is "**less difficult**" than B. If you can decide B, then you can decide A as well.

Reductions

Let A, B be languages. We say that A **Turing reduces to** B (denoted $A \leq_T B$) when:

Given an oracle for B, one can build a decider for A.

"$A \leq_T B$" should be interpreted as: A is "**less difficult**" than B. If you can decide B, then you can decide A as well.

We've shown:

- $S = \{p : p \text{ is prime}\} \leq_T \mathrm{HP}$.
- $\overline{\mathrm{HP}} = \{x : M_x(x) \text{ loops}\} \leq_T \mathrm{HP}$.
- $\mathrm{A_{TM}} = \{(M, w) : M \text{ is a TM that accepts } w\} \leq_T \mathrm{HP}$.

Reductions

Let A, B be languages. We say that **A Turing reduces to B** (denoted $A \leq_T B$) when:

Given an oracle for B, one can build a decider for A.

Task: Show that

$$\mathrm{HP} \leq_T \mathrm{A_{TM}} = \{(M, w) : M \text{ is a TM that accepts } w\}.$$

Conclude that $\mathrm{A_{TM}}$ is undecidable.

Reductions

Let A, B be languages. We say that A **Turing reduces to** B (denoted $A \leq_T B$) when:

Given an oracle for B, one can build a decider for A.

Task: Show that

$$\mathrm{HP} \leq_T \mathrm{Total} = \{M : M \text{ is a TM that halts on any input}\}.$$

Conclude that Total is undecidable.