CSC363 Tutorial #6

More reductions, Arithmetic Hierarchy

March 1, 2023

Things covered in this tutorial

- What is a *m*-reduction?
- What is the *arithmetic hierarchy*?
- Why are we learning all of this?
- Why did I enroll in this course?
- Can I get a hint for A4?

You know why you enrolled in this course.

Reductions, once again...

Recall: Given two languages A and B, we say A **Turing reduces to** B ($A \leq_T B$) if given an oracle for B, you can build a decider for A.

The HP-oracle.

Reductions, once again...

Task: Let

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

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

Prove that $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$.

Reductions, once again...

Task: Let

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

Prove that $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$.
Answer: Assuming we have a decider `in_HP` for HP, we can build the following decider for $\overline{\mathrm{HP}}$:

```
in_HPbar(x):
  if in_HP(x):
    reject
  accept
```

Reductions, once again...

Task: Let

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

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

Prove that $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$.
Answer: Assuming we have a decider `in_HP` for HP, we can build the following decider for $\overline{\mathrm{HP}}$:

```
in_HPbar(x):
  if in_HP(x):
    reject
  accept
```

But $\overline{\mathrm{HP}}$ is not c.e., yet HP is c.e.. Why is $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$?

m-reductions address this issue with Turing reductions.

m-reductions

Also known as *many-one reductions*. We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

The m-reduction is a *stronger* version of the Turing reduction.

m-reductions

Scenario: The aliens come back, and see that we have been abusing their HP-oracle to illegally decide unrecognizable languages like $\overline{\mathrm{HP}}$.

m-reductions

Scenario: The aliens come back, and see that we have been abusing their HP-oracle to illegally decide unrecognizable languages like $\overline{\mathrm{HP}}$.

They confiscate the HP-oracle from humans.

m-reductions

However, the aliens are still willing to solve the halting problem for you.

m-reductions

However, the aliens are still willing to solve the halting problem for you.

Now say you wanted to decide whether something is in A_{TM}. Recall

$$A_{\mathrm{TM}} = \{(x, w) : M_x(e) \text{ accepts}\}.$$

You just have to do the following:

m-reductions

However, the aliens are still willing to solve the halting problem for you.

Now say you wanted to decide whether something is in A_{TM}. Recall

$$A_{\mathrm{TM}} = \{(x, w) : M_x(e) \text{ accepts}\}.$$

You just have to do the following:

- ⋆ Given an instance (x, w) of A_{TM}, construct the following Turing machine T:

  ```
  T(z):
    ignore z
    run M_x(w) # might loop!
    if M_x(w) rejects: loop
    else: halt
  ```

  This machine T has a number; call this machine's number $f(x, w)$.

m-reductions

However, the aliens are still willing to solve the halting problem for you.

Now say you wanted to decide whether something is in A_{TM}. Recall

$$A_{\mathrm{TM}} = \{(x, w) : M_x(e) \text{ accepts}\}.$$

You just have to do the following:

- ⋆ Given an instance (x, w) of A_{TM}, construct the following Turing machine T:

  ```
  T(z):
    ignore z
    run M_x(w) # might loop!
    if M_x(w) rejects: loop
    else: halt
  ```

  This machine T has a number; call this machine's number $f(x, w)$.
- ⋆ Ask the aliens whether $f(x, w) \in \mathrm{HP}$, with a bribe of [REDACTED].

m-reductions

We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

m-reductions

We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

Ok... how is this different from Turing reductions?

Answer: To show $A \leq_T B$, you assume that you have a B-oracle, and build a decider for A.

m-reductions

We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

Ok... how is this different from Turing reductions?

Answer: To show $A \leq_T B$, you assume that you have a B-oracle, and build a decider for A.

To show $A \leq_m B$, you assume that you have a B-oracle, and build a decider for A, *with the following restrictions*:

- ⋆ You must call the B-oracle exactly once – no more, no less.
- ⋆ **You must return whatever the B-oracle returns**; negating the return value of the B-oracle (or performing any modification to the return value) is illegal.
  In other words, the last line of your decider for A must be `return in_B(...)`.

m-reductions

Task: We've shown $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$ using the following decider for $\overline{HP}$:

```
in_HPbar(x):
  if in_HP(x):
    accept
  reject
```

Why isn't the above proof acceptable for showing that $\overline{\mathrm{HP}} \leq_m \mathrm{HP}$?

m-reductions

Task: We've shown $\overline{\mathrm{HP}} \leq_T \mathrm{HP}$ using the following decider for $\overline{HP}$:

```
in_HPbar(x):
  if in_HP(x):
    accept
  reject
```

Why isn't the above proof acceptable for showing that $\overline{\mathrm{HP}} \leq_m \mathrm{HP}$?

Answer: Remember; in a *m*-reduction proof of $A \leq_m B$, you must return whatever the B-oracle returns. You can't make any modifications (such as negation) to what the B-oracle returns.
The last line of your decider for A must be `return in_B(...)`.

m-reductions

Example: show that $\{\text{even numbers}\} \leq_m \{\text{odd numbers}\}$.

m-reductions

Example: show that $\{\text{even numbers}\} \leq_m \{\text{odd numbers}\}$.

Proof. I want to use the following procedure, using an oracle for the odd numbers:

```
is_even(x):
  if is_odd(x):
    reject
  accept
```

Unfortunately, this is not allowed...

m-reductions

Example: show that $\{\text{even numbers}\} \leq_m \{\text{odd numbers}\}$.

Proof.

```
is_even(x):
  t = x + 1
  return is_odd(t)
```

m-reductions

Example: show that $\{\text{even numbers}\} \leq_m \{\text{odd numbers}\}$.

Proof.

```
is_even(x):
  t = x + 1
  return is_odd(t)
```

This is acceptable!

m-reductions

We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

The above function f is called the *reduction function*.[1]

[1] This function does not have to be injective.

m-reductions

We say A m-**reduces to** B ($A \leq_m B$) if there is a computable function $f : \mathbb{N} \to \mathbb{N}$ such that:

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

The above function f is called the *reduction function*.[1]

Task: Show that:

- $\star$ $A \leq_m \{0,1\}$, where A is a computable set.
- $\star$ $A \not\leq_m \emptyset$, where A is any nonempty set.
- $\star$ $A \leq_m \mathrm{HP} = \{x : M_x(x) \text{ halts}\}$, where A is a c.e. set.

[1] This function does not have to be injective.

m-reductions

Note that if A is c.e., then $A \leq_m \mathrm{HP}$.

Is the converse true? If A is any set with $A \leq_m \mathrm{HP}$, does it follow that A is c.e.?

m-reductions

Note that if A is c.e., then $A \leq_m \mathrm{HP}$.

Is the converse true? If A is any set with $A \leq_m \mathrm{HP}$, does it follow that A is c.e.?

Yes. In fact, if $A \leq_m B$ and B is c.e., then so is A. (Think about how you would recognize membership in A!)

m-reductions

Note that if A is c.e., then $A \leq_m \mathrm{HP}$.

Is the converse true? If A is any set with $A \leq_m \mathrm{HP}$, does it follow that A is c.e.?

Yes. In fact, if $A \leq_m B$ and B is c.e., then so is A. (Think about how you would recognize membership in A!)

Consequently, $\overline{\mathrm{HP}} \not\leq_m \mathrm{HP}$.

Arithmetic Hierarchy

(I included this because the assignment's explanation might not be clear enough)

In assignment 3 questions 1 and 4, you prove that any set A is c.e. if and only if there is a computable binary relation R such that

$$A = \{x : \exists y\ R(x, y)\}.$$

Arithmetic Hierarchy

(I included this because the assignment's explanation might not be clear enough)

In assignment 3 questions 1 and 4, you prove that any set A is c.e. if and only if there is a computable binary relation R such that

$$A = \{x : \exists y \; R(x, y)\}.$$

For example:

- ⋆ The set of even numbers E is c.e., since

$$E = \{x : \exists y \; R(x, y)\}$$

where $R(x, y)$ is true iff

Arithmetic Hierarchy

(I included this because the assignment's explanation might not be clear enough)

In assignment 3 questions 1 and 4, you prove that any set A is c.e. if and only if there is a computable binary relation R such that

$$A = \{x : \exists y \; R(x, y)\}.$$

For example:

- ⋆ The set of even numbers E is c.e., since

  $$E = \{x : \exists y \; R(x, y)\}$$

  where $R(x, y)$ is true iff x is even.
- ⋆ The halting set is c.e., since

  $$\mathrm{HP} = \{x : \exists s \; \phi(x, s)\}$$

  where $\phi(x, s)$ is true iff

Arithmetic Hierarchy

(I included this because the assignment's explanation might not be clear enough)

In assignment 3 questions 1 and 4, you prove that any set A is c.e. if and only if there is a computable binary relation R such that

$$A = \{x : \exists y \; R(x, y)\}.$$

For example:

- ⋆ The set of even numbers E is c.e., since

  $$E = \{x : \exists y \; R(x, y)\}$$

  where $R(x, y)$ is true iff x is even.
- ⋆ The halting set is c.e., since

  $$\mathrm{HP} = \{x : \exists s \; \phi(x, s)\}$$

  where $\phi(x, s)$ is true iff $M_x(x)$ halts in s steps or less.

Arithmetic Hierarchy

What about Tot?

$$\mathrm{Tot} = \{x : M_x \text{ halts on any input}\}$$

Arithmetic Hierarchy

What about Tot?

$$\mathrm{Tot} = \{x : M_x \text{ halts on any input}\}$$

Task: find a computable 3-ary relation R such that

$$\mathrm{Tot} = \{x : \forall y \exists s \ R(x, y, s)\}.$$

Arithmetic Hierarchy

What about Tot?

$$\mathrm{Tot} = \{x : M_x \text{ halts on any input}\}$$

Task: find a computable 3-ary relation R such that

$$\mathrm{Tot} = \{x : \forall y \exists s \; R(x, y, s)\}.$$

Answer: $R(x, y, s)$ is true iff $M_x(y)$ halts in s steps or less.

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\ R(x, y, s)\}.$$

So far we have:

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\, R(x, y, s)\}.$$

So far we have:

- ⋆ Any computable set C can be written in the form $C = \{x : R(x)\}$, where R is some computable relation.

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\ R(x, y, s)\}.$$

So far we have:

- ⋆ Any computable set C can be written in the form $C = \{x : R(x)\}$, where R is some computable relation.
- ⋆ Any c.e. set A can be written in the form $A = \{x : \exists y\ R(x, y)\}$.

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\ R(x, y, s)\}.$$

So far we have:

- ⋆ Any computable set C can be written in the form $C = \{x : R(x)\}$, where R is some computable relation.
- ⋆ Any c.e. set A can be written in the form $A = \{x : \exists y\ R(x, y)\}$.
- ⋆ Tot can be written in the form $\{x : \forall y \exists z\ R(x, y, z)\}$.

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\ R(x, y, s)\}.$$

So far we have:

- ⋆ Any computable set C can be written in the form $C = \{x : R(x)\}$, where R is some computable relation.
- ⋆ Any c.e. set A can be written in the form $A = \{x : \exists y\ R(x, y)\}$.
- ⋆ Tot can be written in the form $\{x : \forall y \exists z\ R(x, y, z)\}$.
- ⋆ ??? can be written in the form $\{x : \exists y_1 \forall y_2 \exists y_3\ R(x, y_1, y_2, y_3)\}$.

Arithmetic Hierarchy

$$\mathrm{Tot} = \{x : \forall y \exists s\ R(x, y, s)\}.$$

So far we have:

- ⋆ Any computable set C can be written in the form $C = \{x : R(x)\}$, where R is some computable relation.
- ⋆ Any c.e. set A can be written in the form $A = \{x : \exists y\ R(x, y)\}$.
- ⋆ Tot can be written in the form $\{x : \forall y \exists z\ R(x, y, z)\}$.
- ⋆ Cof can be written in the form $\{x : \exists y_1 \forall y_2 \exists y_3\ R(x, y_1, y_2, y_3)\}$.

$\mathrm{Cof} = \{x :$ There are finitely many inputs y for which $M_x(y)$ loops$\}$.

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

- ⋆ A set A is Σ^0_n if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \ldots y_n \; R(x, y_1, y_2, \ldots, y_n)\}.$$

- ⋆ A set A is Π^0_n if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \ldots y_n \; R(x, y_1, y_2, \ldots, y_n)\}.$$

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

- ⋆ A set A is Σ^0_n if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

- ⋆ A set A is Π^0_n if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

Task: Show that

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

- ⋆ A set A is Σ^0_n if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

- ⋆ A set A is Π^0_n if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

Task: Show that

- ⋆ Any computable set C is Σ^0_0 and Π^0_0.

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ_n^0-Π_n^0 *Hierarchy*.

- ⋆ A set A is Σ_n^0 if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

- ⋆ A set A is Π_n^0 if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

Task: Show that

- ⋆ Any computable set C is Σ_0^0 and Π_0^0.
- ⋆ Any c.e. set A is Σ_1^0. The complement of any c.e. set is Π_1^0.

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

- ⋆ A set A is Σ^0_n if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

- ⋆ A set A is Π^0_n if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

Task: Show that

- ⋆ Any computable set C is Σ^0_0 and Π^0_0.
- ⋆ Any c.e. set A is Σ^0_1. The complement of any c.e. set is Π^0_1.
- ⋆ Tot is Π^0_2.

Arithmetic Hierarchy

This alternating syntactical combination is known as the *Arithmetic Hierarchy* or the Σ^0_n-Π^0_n *Hierarchy*.

- ⋆ A set A is Σ^0_n if there is a n-ary relation R such that

$$A = \{x : \exists y_1 \forall y_2 \exists y_3 \forall y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

- ⋆ A set A is Π^0_n if there is a n-ary relation R such that

$$A = \{x : \forall y_1 \exists y_2 \forall y_3 \exists y_4 \dots y_n \; R(x, y_1, y_2, \dots, y_n)\}.$$

Task: Show that

- ⋆ Any computable set C is Σ^0_0 and Π^0_0.
- ⋆ Any c.e. set A is Σ^0_1. The complement of any c.e. set is Π^0_1.
- ⋆ Tot is Π^0_2.
- ⋆ Cof is Σ^0_3.