Reducibilities

And other cool stuff

Co-hosted by Paul

Alien-Computability

- We saw *A*-computable, *A*-c.e. (given any set *A*: *Alien*)

- $P_e^A, \boldsymbol{\phi}_e^A, W_e^A$ (everything can be relativized)

- We can have: *A*-Σ_n and *A*-Π_n (written as Σ_n^A , Π_n^A)

- A function f is A-p.c. iff for some $e \in \mathbb{N}$, $f = \boldsymbol{\phi}_e^A$.

We can say f is A-p.c. via $\boldsymbol{\phi}_e$

- A function f is A-computable iff for some $e \in \mathbb{N}$, $f = \boldsymbol{\phi}_e^A$ and $\boldsymbol{\phi}_e^A$ is total. We also write $f \leq_T A$.

- A set B is A-c.e. iff for some $e \in \mathbb{N}$, $B = W_e^A$

- A set B is A-computable iff I_B is A-computable. We write $B \leq_T A$

- We can also write $f \leq_T g$ for functions *f,g*

Turing Degrees $\boldsymbol{\mathcal{D}}$

- If $S \leq_T B$ and $S \geq_T B$, then we write $S \equiv_T B$ and say they are Turing equivalent

- $\equiv_T$ is an equivalence relation

- The equivalence classes are called Turing degrees

- Also called degrees of **unsolvability**

Partial Order

- Let S be a set and R be a binary relation on S (i.e. $R \subseteq S \times S$)

R is said to be a partial order (non-strict) on S if:

1. $(\forall a \in S)[R(a,a)]$
2. $(\forall a \in S)(\forall b \in S)[R(a,b)\&R(b,a) \rightarrow a = b]$
3. $(\forall a \in S)(\forall b \in S)(\forall c \in S)[R(a,b)\ \&R(b,c) \rightarrow R(a,c)]$

Total Order

4. $(\forall a \in S)(\forall b \in S)[R(a,b) \text{ or } R(b,a)]$

Every two elements are comparable

Every total order is a partial order, but not the converse

Examples

- Partial order: $P(\mathbb{N})$ and the relation $\subseteq$

- Total order: $\mathbb{N}$ and $\leq$

Structures

- A set equipped with relations and functions

- $(\mathbb{N}, \leq)$ is a partial order structure

- We know also it is a total order structure

$(\boldsymbol{\mathcal{D}}, \leq)$

- The set of Turing degrees can be equipped with a partial order

- This partial order is obtained by defining Turing reducibility on $\boldsymbol{\mathcal{D}}$

- Note that, so far $\leq_T$ is defined on $P(\mathbb{N})$

- Recall that, an element from $\boldsymbol{\mathcal{D}}$ is an equivalence class (set of sets)

This makes $\boldsymbol{\mathcal{D}} \subseteq P(P(\mathbb{N}))$

Lifting $\leq_T$ to $\boldsymbol{\mathcal{D}}$

- For $\mathbf{a}, \mathbf{b} \in \boldsymbol{\mathcal{D}}$, we write $\mathbf{a} \leq \mathbf{b}$ if:

  for some $A \in \mathbf{a}$ and $B \in \mathbf{b}$ we have: $A \leq_T B$

- Is this well-defined?

In other words, if $A \leq_T B$ for **some** $A \in \mathbf{a}$ and $B \in \mathbf{b}$, does this mean that $A \leq_T B$ **for all** $A \in \mathbf{a}$ and $B \in \mathbf{b}$?

- For the definition to make sense, you want the behavior of a degree to be the same as any of its sets

- One can show that $(\boldsymbol{\mathcal{D}}, \leq)$ is a partial order structure

- One can also show that it is NOT total order

- Note: I made a mistake last lecture when I said that $(P(\mathbb{N}), \leq_T)$ is a partial order. Why?

- $\leq_T$ is a partial order on degrees, not on sets.

- $(P(\mathbb{N}), \leq_T)$ is just a preorder, also called quasiorder (reflexive and transitive binary relation)

Sad thing about Turing Reducibility

- It does not distinguish between C.e. sets and Co-c.e. sets

- This is because for any set A, A and its complement $\bar{A}$ are both of the same Turing degree

- It is possible to have $A \leq_T B$ where we can computably enumerate B but can't enumerate A

m-reducibility: A stronger reducibility

- $A \leq_m B$, A is many-one reducible to B if there is a computable function f such that:

$$\text{For all } x \in \mathbb{N},\ x \in A \textit{ iff } f(x) \in B$$

- Again, $\leq_m$ is a preorder on $\mathcal{P}(\mathbb{N})$, which can induce an equivalence relation with equivalence classes called m-degrees

- If f is injective, we write $A \leq_1 B$ and say A is 1-reducible to B

- $\leq_1$ implies $\leq_m$ implies $\leq_T$

- Exercise: Find examples that the converse implications fail

- If $C \leq_m B$ and B is A-c.e. , then C is also A-c.e.

- If $B \in \Sigma_n^A$ (or Π_n^A), and $C \leq_m B$, then $C \in \Sigma_n^A$ (or Π_n^A)

Break

How many elements in $\mathcal{D}$?

Example 1

- $K_0 = \{\langle e, x\rangle : \varphi_e(x)\downarrow\}$ is in Σ_1

- For every A in Σ_1, $A \leq_m K_0$

Indeed, we know that $A = W_e$ for some $e \in \mathbb{N}$.

Consider now the function f given by $f(x) = \langle e, x\rangle$.
Clearly f is computable, and $x \in A \Leftrightarrow f(x) \in K_0$

- Note that f is also injective, and so $A \leq_1 K_0$

C-complete

- The example we gave shows that the set K_0 is Σ_1-complete

- More generally, given a reducibility $\leq_r$ and a class of sets **C**,

we say that a set *B* is **C**-*complete* w.r.t. $\leq_r$ if:

1. $B \in \mathbf{C}$

2. $C \leq_r B$ for every $C \in \mathbf{C}$

- If 1. isn't happening, we say *B* is **C**-*hard*

- When we don't specify the reducibility, we mean it is m-reducibility

Σ_n -completeness (and Π_n -completeness)

- When we say Σ_n -complete, without a reducibility specified, we mean with respect to 1-reducibility

- Equivalently in this case, m-reducibility

- $\emptyset^{(n)}$ is Σ_n -complete

- $\overline{\emptyset^{(n)}}$ is Π_n -complete

Examples 2

- Consider the set $\mathbf{Tot} = \{e : \varphi_e \text{ is total}\}$

- $\mathbf{Tot}$ is in Π_2
- For every A in Π_2 , $A \leq_m \mathbf{Tot}$
- This means that $\mathbf{Tot}$ is Π_2 *-complete*

Proof:

- A in Π_2 means that there exists a computable relation R such that

$$x \in A \Leftrightarrow (\forall y)(\exists z)R(x, y, z)$$

- Consider the following function:

$$\gamma(x, u) = \begin{cases} 0 & \text{if } (\forall y \leq u)(\exists z)R(x, y, z) \\ \uparrow & o.w. \end{cases}$$

- $\gamma(x, u)$ is clearly p.c.

- There exists computable f such that $\gamma(x, u) = \varphi_{f(x)}(u)$

- This follows from the s-m-n theorem

- Now observe the following:

$$x \in A \Longrightarrow \varphi_{f(x)} \text{ is total}$$

$$x \in \bar{A} \Longrightarrow \varphi_{f(x)} \text{ is NOT total}$$

- This means that:

$$x \in A \Longleftrightarrow f(x) \in \mathbf{Tot}$$

Q.E.D

- Remark: f could be chosen injective

Example 3

- Consider the set $\mathbf{Fin} = \{e : W_e \text{ is finite}\}$

- $\mathbf{Fin}$ is $\Sigma_?$
- Actually, $\mathbf{Fin}$ is $\Sigma_?$ -complete
- Because in the proof of Example 2, we have that when $x \in \bar{A}$, the domain of $\varphi_{f(x)}$ is finite

So, we have

- Let A be an arbitrary set from Σ_2

- Then $\bar{A} \in \Pi_2$, and so by the proof of Example 2, there is a computable (can be chosen injective) f such that:

$$x \in \bar{A} \Longrightarrow \varphi_{f(x)} \text{ is total} \Leftrightarrow W_{f(x)} = \mathbb{N} \text{ which is infinite}$$

$$x \in A \Longrightarrow W_{f(x)} \text{ is finite}$$

- In other words, $x \in A \Leftrightarrow f(x) \in \mathbf{Fin}$

Facts:

- B is c.e. in A iff $B \leq_1 A'$
- If $B \leq_T A$ then $B' \leq_1 A'$
- A' is c.e. in A
- If B is c.e. in A then B is c.e. in $\bar{A}$
- $\Sigma_n^{\emptyset^{(m)}} = \Sigma_{m+n}$

Break

Some cool stuff: Kolmogorov Complexity

- Consider the following function: $K(x) = \mu e(\varphi_e(0) = x)$

- In some sense, this function gives the shortest program that can output x

- This output can be regarded as the shortest description of the string $gn^{-1}(x)$

- We say a string s is **random**, if $K(gn(s)) \geq gn(s)$

Useful stuff

- Let *A*, *B* be two sets (very general)
- We denote the set of functions from *A* to *B* by B^A
- This notation is a cool connection with combinatorics. What is $|B^A|$?
- $P(A)$ can be identified with $\{0,1\}^A$ (the set of characteristic functions of subsets of *A*)
- $|P(A)| = |\{0,1\}|^{|A|}$

Computability and real numbers

- A real number $r \in \mathbb{R}$ is computable if when given any $n \in \mathbb{N}$ one can compute a rational number $q \in \mathbb{Q}$ such that $|r - q| \leq 2^{-n}$

- $\mathbb{R}$ can be viewed as $\{0,1\}^{\mathbb{N}}$

- $\{0,1\}^{\mathbb{N}}$ this is known as the Cantor space

- The word space is related to topology

H10

After some experience

Remember H10 ?

- A set A is Diophantine if there exists a polynomial $P_A(x, y_1, \dots, y_n)$ such that
$$a \in A \Leftrightarrow (\exists y_1) \dots (\exists y_n) P_A(x, y_1, \dots, y_n) = 0$$

- A is clearly Σ_1, i.e. C.E.

- Every set from Σ_1 is Diophantine

- One can show that a set of **positive** integers is Diophantine iff it is the range of a polynomial function

Simple examples of Diophantine sets

- $\leq = \{(x, y) : (\exists z)\ x + z - y = 0\}$

- The set of prime numbers is the range of a polynomial function

- The record for the lowest degree of such a polynomial is 5 (with 42 variables)

- The record for fewest variables is 10 with degree about 1.6×10^4

The key result for H10

- The exponential function $h(x, y) = x^y$ is Diophantine.

We mean by that

$$\{(x, y, z) : x^y = z\}$$

is Diophantine

Open Problem

- Hilbert 10^{th} over $\mathbb{Q}$

- Lots of number theory, rings and fields stuff

Logic

Theories and Axioms

- You saw the partial order definition
- They form a set of sentences (logical formulas without free variables)

- Such a collection of sentences is called a *theory*

- A set of *axioms* is just a theory. Usually it is picked so they describe the basic facts about the theory without redundancy
- By describing basic facts I mean one can deduce the whole theory from the axioms by a *proof*

Proof system

- A list of formulas such that each formula is either an axiom, or comes from previous formulas by a rule of inference

- Example of a rule of inference: Modus ponens

$$
\begin{array}{c}
P \\
P \rightarrow Q \\
----- \\
Q
\end{array}
$$

Logic: Theorems

- A *theorem* is a sentence that can be the end of a proof

- A theorem is also called a *consequence*

- Example: Let PO denote the set of partial order axioms.

We have

$$\mathrm{PO} \vdash \quad (\forall x)(\forall y)(\forall z)(\forall w)[\, x \leq y \& y \leq z \& z \leq w \rightarrow x \leq w]$$

($\vdash$ is the verb "proves")

Theories and Computability

- A set Ax *axiomatizes* a theory T if every sentence in T is provable from Ax

- It is of interest sometimes to look for Ax which is computable, or c.e.

- Fact: The set of consequences (theorems) of a c.e. set of axioms is c.e.

- Craig's Theorem: A c.e. theory has a computable set of axioms (primitive recursive actually)

Consistency

- A theory is consistent if it has a *model*

- Examples: The structure $(\mathbb{N}, \leq) \vDash$ PO ($\vDash$ is the verb "models")

$(\boldsymbol{\mathcal{D}}, \leq_T) \vDash$ PO

- A theory T is inconsistent if it can prove a sentence and its negations

$$\mathrm{T} \vdash \varphi \& \neg\varphi$$

- This also means that for **any** sentence φ, $\mathrm{T} \vdash \varphi$

Soundness

- Suppose you have a theory T and a sentence φ such that T $\vdash \varphi$

- Soundness of the proof system means that for every model *M*,

$$M \vDash \mathrm{T} \Longrightarrow M \vDash \varphi$$

- The last line is usually abbreviated as T $\vDash \varphi$ (semantic implication)

- So basically, soundness of a proof system is: If T $\vdash \varphi$ then T $\vDash \varphi$

Completeness

- Completeness of a proof system is: If T $\vDash \varphi$ then T $\vdash \varphi$

- Gödel completeness theorem: For any first order theory T, and any sentence φ (in the language of the theory): If T $\vDash \varphi$ then T $\vdash \varphi$

- A theory T is *complete* if for every sentence φ its language,

either T $\vdash \varphi$ or T $\vdash \neg\varphi$

Axiom Independence

- Suppose you have a consistent list of axioms A1,A2,A3,A4

- What does it mean that, say, A2 is independent from the rest?

- This means $\{A1,A3,A4\} \nvDash A2$

- This also means that: There is a model $M1 \vDash \{A1,A2,A3,A4\}$ and there is also a model $M2 \vDash \{A1,\neg A2,A3,A4\}$

Example

A1: $(\forall a)[R(a,a)]$
A2: $(\forall a)(\forall b)[R(a,b)\&R(b,a) \rightarrow a = b]$
A3: $(\forall a)(\forall b)(\forall c)[R(a,b)\&R(b,c) \rightarrow R(a,c)]$

- PO = {A1,A2,A3}, Pre = {A1,A3}

- A2 is independent of A1, A3 because

$$(\mathcal{D}, \leq_T) \vDash \{A1,A2,A3\} \text{ and } (P(\mathbb{N}), \leq_T) \vDash \{A1,\neg A2,A3\}$$

- Pre is clearly an example of an incomplete theory since

$$\text{Pre} \nvdash \text{A2 and Pre} \nvdash \neg\text{A2}$$

Theory of Arithmetic

- The theory Th(ℕ) of all the facts about the structure of natural numbers is LIFE

- Naturally there is a desire to capture it through a manageable set of axioms

- By manageable I mean finite, or just computable
- By capture I mean axiomatize
- Sadly, this isn't possible (Gödel's Incompleteness Theorem)

Gödel's First Incompleteness

- Within the language of PA, Gödel used his numbering tricks to make sentences speak about themselves (self reference)

- The idea is to create a formula $P(x, y)$ using 0,+,x,(,),s,→,¬, ... such that y is the Gödel number of a proof in PA of the sentence whose Gödel number is x

- Look now at this sentence: $\neg\exists y P(e, y)$ where $e = gn(\neg\exists y P(e, y))$

- **It** says e (myself), not provable

- **We** see (as outsiders) that it is true in the model $(\mathbb{N}, 0, +, \times, s)$

Gödel's Second Incompleteness

- Gödel decided to play more with his numbering trick and created a sentence that speaks about PA (about the system from within the system)

- The sentence said: PA is consistent.

- Consis(PA): $\neg\exists y P(gn(0 \neq 0), y)$ (there is no proof of $0 \neq 0$)

- Then Gödel showed that: PA $\nvdash$ Consis(PA)

- In other words, PA cannot prove its own consistency

Generalizability of the Incompleteness Theorems

- All those proofs of Gödel just required that the system is powerful enough to express arithmetic

- So, he was able to prove similar facts about, e.g., set theory

- $\emptyset = 0, \{\emptyset\} = 1, \{\emptyset, \{\emptyset\}\} = 2, \ldots, n = \{0,1, \ldots, n-1\}$

In philosophical terms

- A system which is powerful (powerful enough to describe arithmetic) does not have a computable list of axioms from which every fact could follow

- Imagine yourself creating a manageable (finite or computable) list of rules (laws) from which everything in your system of interest should follow.

- Unless your system is very weak, you can't

Factory Analogy

- Imagine you have a factory that creates machines

- You want to create a machine which can test **every** machine in the factory

- It can test everything except **itself**

- It might be able to test certain aspects of itself, but not all of itself without **external** interference

Camera analogy

- A camera can't take a picture of itself

- Maybe with the aid of an **external** system of mirrors

Peano Arithmetic (example of axiomatization)

- The structure of natural numbers could be described (axiomatized) by the following set of axioms PA:

1. Natural numbers not empty
2. They can be built from a special number, call it 0, and a special function *s* (call it successor)

3. So, for every x, if x is a natural number, then s(x) is also a natural

4. For every x, s(x) is not 0
5. m=n iff s(m)=s(n)
6. If a = b, and a is natural, then b is natural
7. If 0 has a property P, and for every n, if n has P then s(n) has P, then P applies to all natural numbers

Structure of arithmetic

We have a structure $\mathcal{N} = (\mathbb{N}, 0, +, \times, s)$ which satisfies:

1. $\forall x\ 0 \neq s(x)$
2. $\forall x \forall y\ (s(x) = s(y) \rightarrow x = y)$
3. $\forall x\ 0 \neq s(x)$
4. For each formula $\varphi(x, \bar{y})$ in the language of Peano Arithmetic:

$$\forall \bar{y}\ [\varphi(0, \bar{y})\ \&\ \forall x\big(\varphi(x, \bar{y}) \rightarrow \varphi(s(x), \bar{y})\big) \rightarrow \forall x\ \varphi(x, \bar{y})]$$

That last axiom is actually an axiom schema. It unfolds into an infinite set of axioms

+, X

- $\forall x\ x + 0 = x$
- $\forall x \forall y\ (x + s(y) \rightarrow s(x + y))$