

CS21 Decidability and Tractability

Lecture 14
February 8, 2008

February 8, 2008

CS21 Lecture 14

1

Outline

- Gödel Incompleteness Theorem

February 8, 2008

CS21 Lecture 14

2

Background

- Hilbert's program (1920's):
 - formalize mathematics in axiomatic form
 - derive all true statements "mechanically" from initial axioms
 - would put mathematicians out of business!
 - very influential proposal
- to start: try for all true statements about the natural numbers ("number theory")

February 8, 2008

CS21 Lecture 14

3

Background:

- Kurt Gödel (1931): it is not possible!
- no formalization of number theory can prove all true statements
- stunning result
- considered one of greatest 20th century achievements in math.

February 8, 2008

CS21 Lecture 14

4

Background

- We will prove using:
 - RE languages and non-RE languages
 - reductions
- Idea:
 - set of all theorems is RE
 - set of all true statements is not RE
- This kind of proof of Gödel's result attributed to Turing (1937).

February 8, 2008

CS21 Lecture 14

5

Number Theory

- formal language to express properties of $\mathbf{N} = \{0, 1, 2, 3, \dots\}$
- allowable symbols: parentheses, and
 - variables x, y, z, \dots ranging over \mathbf{N}
 - operators $+$ (addition) and $*$ (multiplication)
 - constants 0 (additive id) and 1 (mult. identity)
 - relation $=$ (equality)
 - quantifiers \forall (for all) and \exists (exists)
 - propositional operators \wedge (and) \vee (or) \neg (not) \Rightarrow (implies) \Leftrightarrow (iff)

February 8, 2008

CS21 Lecture 14

6

Peano Arithmetic

- Peano Arithmetic: proof system for number theory. Axioms:

– 0 is not a successor

$$\forall x \neg (0 = x + 1)$$

– the successor function is one-to-one

$$\forall x \forall y (x+1 = y+1 \Rightarrow x = y)$$

– 0 is an identity for +

$$\forall x x + 0 = x$$

February 8, 2008

CS21 Lecture 14

13

Peano Arithmetic

– + is associative

$$\forall x \forall y x + (y + 1) = (x + y) + 1$$

– multiplying by zero gives 0

$$\forall x x * 0 = 0$$

– * distributes over +

$$\forall x \forall y x * (y + 1) = (x * y) + x$$

– induction axiom

$$(\varphi(0) \wedge \forall x (\varphi(x) \Rightarrow \varphi(x+1))) \Rightarrow \forall x \varphi(x)$$

February 8, 2008

CS21 Lecture 14

14

Peano Arithmetic

- rules of inference:

modus ponens $\frac{\varphi \quad \varphi \Rightarrow \psi}{\psi}$

generalization $\frac{\varphi}{\forall x \varphi}$

February 8, 2008

CS21 Lecture 14

15

Proof systems

- a proof is a sequence of formulas

$$\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n$$

such that each φ_i is either

– an axiom, or

– follows from formulas earlier in list from rules of inference

- A sentence is a theorem of the proof system if it has a proof

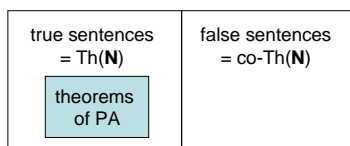
February 8, 2008

CS21 Lecture 14

16

Proof systems

- A proof system is **sound** if all theorems in that proof system are true (better have this)
- Peano Arithmetic (PA) is sound.



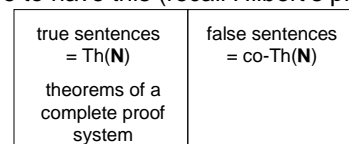
February 8, 2008

CS21 Lecture 14

17

Proof systems

- A proof system is **complete** if all true sentences are theorems in that proof system
- hope to have this (recall Hilbert's program)



February 8, 2008

CS21 Lecture 14

18

Incompleteness Theorem

Theorem: Peano Arithmetic is not complete.

(same holds for any reasonable proof system for number theory)

Proof outline:

- the set of theorems of PA is RE
- the set of true sentences (= Th(**N**)) is not RE

February 8, 2008

CS21 Lecture 14

19

Incompleteness Theorem

- Lemma: the set of theorems of PA is RE.
- Proof:
 - TM that recognizes the set of theorems of PA:
 - systematically try all possible ways of writing down sequences of formulas
 - accept if encounter a proof of input sentence (note: true for any reasonable proof system)

February 8, 2008

CS21 Lecture 14

20

Incompleteness Theorem

- Lemma: Th(**N**) is not RE
- Proof:
 - reduce from co-HALT (show $\text{co-HALT} \leq_m \text{Th}(\mathbf{N})$)
 - recall co-HALT is not RE
 - what should $f(\langle M, w \rangle)$ produce?
 - construct γ such that M loops on $w \Leftrightarrow \gamma$ is true

February 8, 2008

CS21 Lecture 14

21

Incompleteness Theorem

- we will define $\text{VALCOMP}_{M,w}(y) \equiv \dots$ (details to come) so that it is true iff y is a (halting) computation history of M on input w
- then define $f(\langle M, w \rangle)$ to be:
 - $\gamma \equiv \neg \exists y \text{VALCOMP}_{M,w}(y)$
- YES maps YES?
 - $\langle M, w \rangle \in \text{co-HALT} \Rightarrow \gamma$ is true $\Rightarrow \gamma \in \text{Th}(\mathbf{N})$
- NO maps to NO?
 - $\langle M, w \rangle \notin \text{co-HALT} \Rightarrow \gamma$ is false $\Rightarrow \gamma \notin \text{Th}(\mathbf{N})$

February 8, 2008

CS21 Lecture 14

22

Expressing computation in the language of number theory

- we'll write configurations over an alphabet of size p , where p is a prime that depends on M
- y is a power of p :
 - $\text{POWER}_p(y) \equiv \exists z (\text{DIV}(z, y) \wedge \text{PRIME}(z)) \Rightarrow z = p$
- $d = p^k$ and length of v as a p -ary string is k
 - $\text{LENGTH}(v, d) \equiv \text{POWER}_p(d) \wedge v < d$

February 8, 2008

CS21 Lecture 14

23

Expressing computation in the language of number theory

- the p -ary digit of v at position y is b (assuming y is a power of p):
 - $\text{DIGIT}(v, y, b) \equiv \exists u \exists a (v = a + by + upy \wedge a < y \wedge b < p)$
- the three p -ary digits of v at position y are b, c , and d (assuming y is a power of p):
 - $\text{3DIGIT}(v, y, b, c, d) \equiv \exists u \exists a (v = a + by + cpy + dppy + uppy \wedge a < y \wedge b < p \wedge c < p \wedge d < p)$

February 8, 2008

CS21 Lecture 14

24

Expressing computation in the language of number theory

- the three p-ary digits of v at position y “match” the three p-ary digits of v at position z according to M’s transition function (assuming y and z are powers of p):

$$\text{MATCH}(v, y, z) \equiv$$

$$\forall (a,b,c,d,e,f) \in C \text{ 3DIGIT}(v, y, a, b, c) \wedge \text{3DIGIT}(v, z, d, e, f)$$

where $C = \{(a,b,c,d,e,f) : abc \text{ in config. } C_i \text{ can legally change to } def \text{ in config. } C_{i+1}\}$

Expressing computation in the language of number theory

- all pairs of 3-digit sequences in v up to d that are exactly c apart “match” according to M’s transition function (assuming c, d powers of p)

$$\text{MOVE}(v, c, d) \equiv$$

$$\forall y (\text{POWER}_p(y) \wedge y + pc < d) \Rightarrow \text{MATCH}(v, y, y + pc)$$

Expressing computation in the language of number theory

- the string v starts with the start configuration of M on input $w = w_1 \dots w_n$ padded with blanks out to length c (assuming c is a power of p):

$$\text{START}(v, c) \equiv$$

$$\bigwedge_{i=0,1,2,\dots,n} \text{DIGIT}(v, p^i, k_i) \wedge p^n < c \wedge \forall y (\text{POWER}_p(y) \wedge p^n < y < c \Rightarrow \text{DIGIT}(v, y, k))$$

where $k_0 k_1 k_2 k_3 \dots k_n$ is the p-ary encoding of the start configuration, and k is the p-ary encoding of a blank symbol.

Expressing computation in the language of number theory

- string v has a halt state in it somewhere before position d (assuming d is power of p):

$$\text{HALT}(v, d) \equiv$$

$$\exists y (\text{POWER}_p(y) \wedge y < d \wedge \forall a \in H \text{ DIGIT}(v, y, a))$$

where H is the pair of p-ary digits corresponding to states q_{accept} and q_{reject} .

Expressing computation in the language of number theory

- string v is a valid (halting) computation history of machine M on string w:

$$\text{VALCOMP}_{M,w}(v) \equiv$$

$$\exists c \exists d (\text{POWER}_p(c) \wedge c < d \wedge \text{LENGTH}(v, d) \wedge \text{START}(v, c) \wedge \text{MOVE}(v, c, d) \wedge \text{HALT}(v, d))$$

- M does not halt on input w:

$$\neg \exists v \text{ VALCOMP}_{M,w}(v)$$

Incompleteness Theorem

$$v = 136531362313603131031420314253$$

$$\text{VALCOMP}_{M,w}(v) \equiv$$

$$\exists c \exists d (\text{POWER}_p(c) \wedge c < d \wedge \text{LENGTH}(v, d) \wedge \text{START}(v, c) \wedge \text{MOVE}(v, c, d) \wedge \text{HALT}(v, d))$$

Incompleteness Theorem

- Lemma: $\text{Th}(\mathbf{N})$ is not RE
- Proof:
 - reduce from co-HALT (show $\text{co-HALT} \leq_m \text{Th}(\mathbf{N})$)
 - recall co-HALT is not RE

 - constructed γ such that
M loops on $w \Leftrightarrow \gamma$ is true