Lecture 38: Gödel Incompleteness 2

CSCI 81
Spring, 2015

Kim Bruce

Gödel Incompleteness

- Gödel 1: Let T be a decidable set of axioms true of the natural numbers & that implies the axioms of Peano Arithmetic. Then there is a sentence γ which is true of N but is not provable in T.
  - Proof only depended on ability to encode computation.
  - Set of statements provable from a decidable T is semi-decidable, but Th(N) is not.
  - T consistent $\Rightarrow$ Provable(T) = $\{\phi \mid T \vdash \phi\} \subseteq$ Th(N).

Gödel's Proof

- More interesting in that used fixed point to construct sentence that asserts its own unprovability. $N \models \phi$ iff not $\vdash_{PA} \phi$.
- Suppose had such sentence. Claim $N \models \phi$.
  - Suppose not: If $N \not\models \phi$ then $\vdash_{PA} \phi$. But PA consistent, so can't prove false things. Thus $N \models \phi$.
- But then not $\vdash_{PA} \phi$, so ϕ is true but not provable.

Construct ϕ as Fixed-Point

- For any formula ψ, let 'ψ' be a numeric encoding of ψ. Write ϕ_n for the formula with numeric code n. Thus $\phi_{'\psi'} = \psi$
- Thm: For any $\psi(x)$ with one free variable x, there exists a sentence τ s.t. $\vdash_{PA} \tau \leftrightarrow \psi('\tau')$
  - *Done last time*

Proving Incompleteness

- Can define formula Proof(x,y) true iff sequence encoded by x is a valid proof in PA of ϕ_y.
  - i.e. Proof('π', 'ϕ') iff π is a proof in PA of ϕ
- Define Provable(y) iff $\exists$x Proof(x,y)
  - Thus $\vdash_{PA} \phi$ iff $N \vDash$ Provable('ϕ') (*)
  - In fact $\vdash_{PA} \phi$ iff $\vdash_{PA}$ Provable('ϕ') because argument can be encoded in PA.
- Use fixed point to get τ s.t
  $\vdash_{PA} \tau \leftrightarrow \neg$Provable('$\tau$')

Incompleteness

- $\vdash_{PA} \tau \leftrightarrow \neg$Provable('$\tau$')
- As above τ must be true and therefore not provable:
  - $N \vDash \tau \Rightarrow N \vDash \neg$Provable('$\tau$') $\Rightarrow N \nvDash$ Provable('τ') $\Rightarrow \nvdash_{PA} \tau$ *by * on previous page.*
- Proved there is τ s.t. τ is true of N, but not provable in PA.

Defining Truth?

- Can we define predicate True(x) s.t. for all ϕ, $\vDash \phi$ iff $\vDash$ True('ϕ') ?
- By fixed point theorem, exists σ s.t.
  - $\vDash \sigma$ iff $\vDash \neg$True('σ')
  - Therefore can't define formula True.
- Can reason about consistency and provability, but not truth.

Consistency is Contradictory!

- Define Consis = $\neg$ Provable('$\perp$')
- Gödel's second incompleteness theorem: No sufficiently powerful deductive system can prove its own consistency, unless it is inconsistent (*and hence can prove anything*).
  - We'll do proof with PA, but stronger works too.

Proof of Gödel 2

- Let τ be fixed point: $\vdash_{PA} \tau \leftrightarrow \neg\text{Provable}(`\tau')$
  - If $\vdash_{PA} \tau$, then $\vdash_{PA} \text{Provable}(`\tau')$, but by above also get $\vdash_{PA} \neg\text{Provable}(`\tau')$, so inconsistent. (*)
  - Formalizing in PA, get
    $\vdash_{PA} \text{Provable}(`\tau') \rightarrow \neg\text{Consis}$
    or equivalently,
    $\vdash_{PA} \text{Consis} \rightarrow \neg\text{Provable}(`\tau')$
  - Suppose $\vdash_{PA}$ Consis. Then $\vdash_{PA} \neg\text{Provable}(`\tau')$ and hence, $\vdash_{PA} \tau$, and by (*), PA is inconsistent.
  - Thus PA consistent $\Rightarrow$ $\nvdash_{PA}$ Consis

Computability

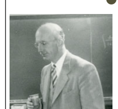
- Leibniz, Hilbert
  - Human reasoning reduced to calculations
  - Formalist philosophy of mathematics
- Gödel, Church, Kleene, Rosser, Turing
- Post (like grammars), Labeled Markov Algorithms (rewriting with goto's), ...

Lambda Calculus

- λ-calculus invented in 1928 by Church in Princeton & first published in 1932.
- Goal to provide a foundation for logic
- First to state explicit conversion rules.
- Original version inconsistent, but corrected
  - "If this sentence is true then 1 = 2" *problematic!!*
- 1933, definition of natural numbers

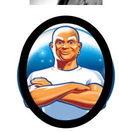
Curry's paradox

Collaborators



- 1931-1934: Grad students:
  - J. Barkley Rosser and Stephen Kleene
  - Church-Rosser confluence theorem ensured consistency (earlier version inconsistent)

  - Kleene showed λ-definable functions very rich
    - Equivalent to Herbrand-Gödel recursive functions
    - Equivalent to Turing-computable functions.
    - Founder of recursion theory, invented regular expressions



- Church(-Turing) thesis:
  - λ-definability ≡ effectively computable

Undecidability

- Convertibility problem for λ-calculus undecidable.
- Validity in first-order predicate logic undecidable.
- Proved independently year later by Turing.
  - First showed halting problem undecidable

Alan Turing

- Turing
  - 1936, in Cambridge, England, definition of Turing machine
  - 1936-38, in Princeton to get Ph.D. under Church.
  - 1937, first published fixed point combinator
    - (λx. λy. (y (x x y))) (λx. λy. (y (x x y)))
  - *Kleene did not use fixed-point operator in defining functions on natural numbers!*
  - Broke German enigma code in WW2, Turing test AI
  - Persecuted as homosexual, committed suicide in 1954

Breaking the Code
The imitation game

Hilbert's Program

- Response to foundational crisis in math
  - Foundations of math threatened by paradoxes.
    - Set of all sets that do not contain themselves.
  - Reduce math to finite complete set of axioms & prove they are consistent

Goals

- Provide secure foundation for math:
  - Formalization of all math (axioms & rules)
  - Completeness: All true statements provable
  - Consistency: Proof no contradictions can be obtained
    - Use only finitistic reasoning.
  - Conservation: Proofs about "real" objects using ideal objects (infinite sets) can be obtained with ideal objects.
  - Decidability: Algorithm for deciding truth.

Implications of Gödel Incompleteness

- Can't formalize even number theory w/o leaving out some true statements.
- No complete consistent extension of Peano Arithmetic w/ semidecidable set of axioms
- No extension of PA can prove its own consistency
- No algorithm to determine truth of statements in consistent extension of PA (Church/Turing)

Blow to Hilbert!

- Gödel 2nd Incompleteness holds of number theory, set theory, etc.
- Destroyed Hilbert's program.
- Stronger system can prove consistency of weaker systems
  - Can prove consistency of PA within set theory.

Impact

- Very little impact on practice of Math
  - Exception: Paris-Harrington Theorem in combinatorics is true but not provable in PA.
  - Implies consistency of PA
  - Math lives on "dangerous" foundations, but any problems likely easily repairable.
- Independence of Axiom of Choice, Continuum Hypothesis (Gödel plus others) from ZF set theory had more impact.

Undecidability

- Shocking at first
  - Pervasive
- Reduction proofs led to work on Complexity Hierarchy.
- NP-Completeness