Essential Incompleteness of Arithmetic Verified by Coq

Russell O'Connor

Radboud University Nijmegen

University of California at Berkeley

TPHOLS 2005

August 23, 2005

Essential Incompleteness of Arithmetic Verified by Coq

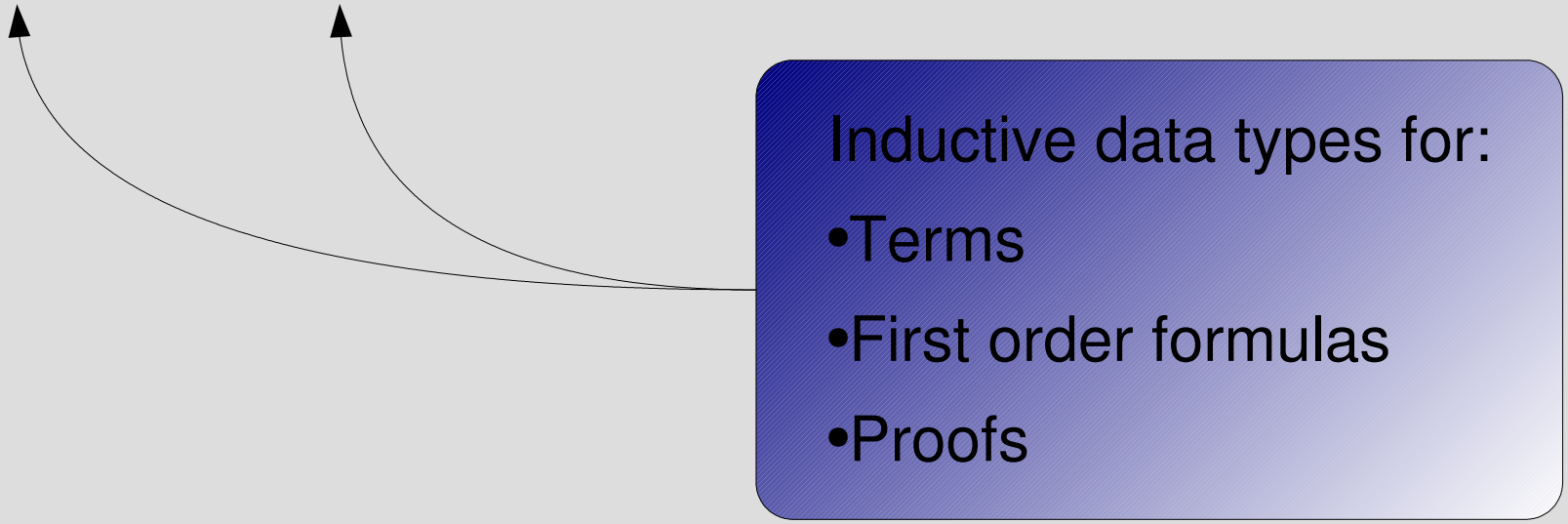
1. Statement of incompleteness theorem
2. Coding by natural numbers
3. Primitive recursive functions
4. Corollaries of incompleteness theorem
5. Discussion of formalization
6. Future Work: Second incompleteness theorem

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.



Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

Recursive functions for:

- Free variables
- Substitution

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

A formula with no free variables

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

Formula $\Rightarrow$ Prop

(think Formula $\Rightarrow$ type)

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

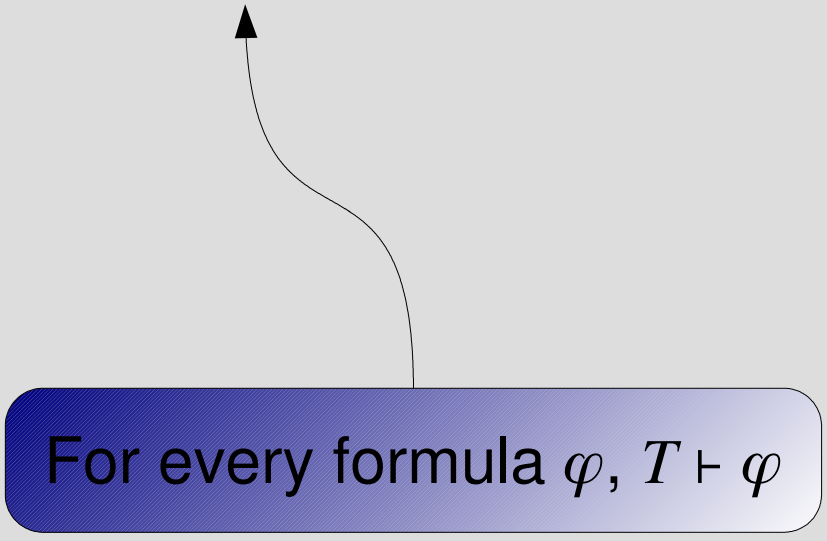
A finite axiomatization of:

0, S, +, ×, <

without induction

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.



Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

- For every formula φ, $\varphi \in T$ or $\varphi \notin T$
- Classically this is trivially true
- This is not the recursive requirement.

Gödel-Rosser Incompleteness Theorem

For any decidable axiom system T extending NN such that T can express its own axioms, there exists a sentence G such that if either $T \vdash G$ or $T \vdash \neg G$ then T is inconsistent.

This replaces the requirement that T is recursive

This is weaker that T being recursive

T Can Express Its Own Axioms

There exists a formula with one free variable $\varphi(x)$ such that

- If $\psi \in T$ then $T \vdash \varphi(\ulcorner \psi \urcorner)$
- If $\psi \notin T$ then $T \vdash \neg\varphi(\ulcorner \psi \urcorner)$

T Can Express Its Own Axioms

There exists a formula with one free variable $\varphi(x)$ such that

- If $\psi \in T$ then $T \vdash \varphi(\ulcorner\psi\urcorner)$
- If $\psi \notin T$ then $T \vdash \neg\varphi(\ulcorner\psi\urcorner)$

$\ulcorner\psi\urcorner$ is ψ coded by a number written as a closed term.

Coding

- Extensive use of Cantor Pairing Function

$$cPair(a,b) = a + \sum_{i=1}^{a+b} i$$

- Formulas are coded by giving each symbol a number and pairing it with its recursively coded arguments.
- No prime number decomposition theorem is needed.

Primitive Recursive Functions

- Inductive data type for primitive recursive expressions
- Evaluation function for primitive recursive expressions
- Every primitive recursive function is representable in NN
  - Requires Chinese remainder theorem and Gödel's beta function

Primitive Recursive Functions

- Inductive
- Evaluatio

expressions

- Every primitive recursive function is representable in NN
  - Requires Chinese remainder theorem and Gödel's beta function

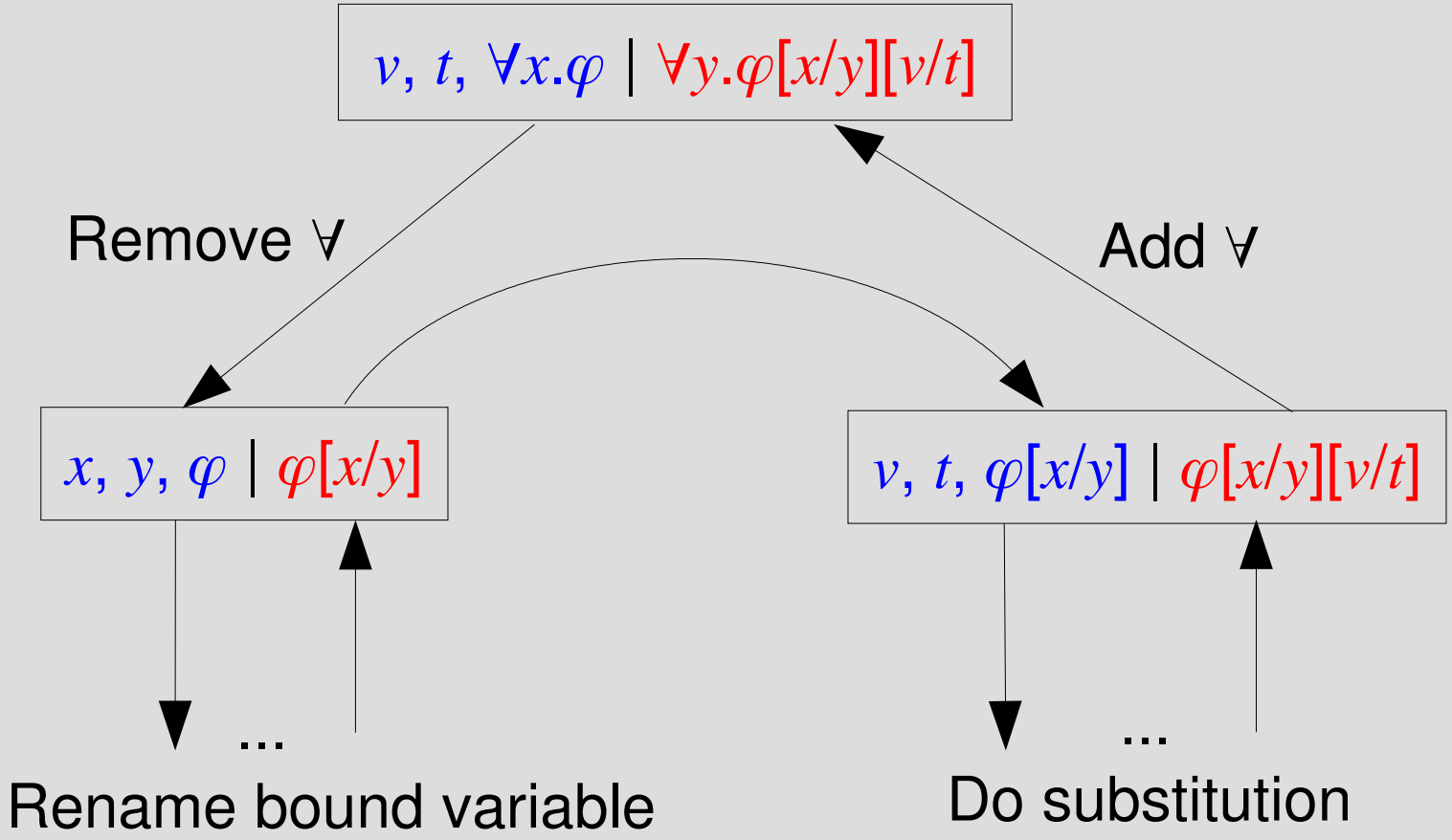
An n-ary function, f, is representable by φ in T if

$$T \vdash \varphi(x_0, \ulcorner a_1 \urcorner, \ldots, \ulcorner a_n \urcorner) \Leftrightarrow x_0 = \ulcorner f(a_1, \ldots, a_n) \urcorner$$

Substitution Is Primitive Recursive

- Recursion on depth does not work.
- Course-of-values recursion does not work.
  - Recursive call's input may have a larger code than the original input.
- Instead consider the trace of the computation
  - A primitive recursive function can check if a trace is correct
  - Do a bounded search for the correct trace

Substitution Is Primitive Recursive



Corollaries

- Gödel Sentence with ω-consistency
- If PA is consistent then PA is essentially incomplete.

Corollaries

- Gödel Sentence with ω-consistency
- If PA is consistent then PA is essentially incomplete.

Proved that PA is consistent

Corollaries

- Gödel Sentence with ω-consistency
- PA is essentially incomplete.

Corollaries

- Gödel Sentence with ω-consistency
- PA is incomplete.

Machine Verified Proofs

- 1986 – Shankar – Boyer-Moore
  - Incompleteness for any finite extension of Z2 (hereditarily finite set theory)
- 2003 – O'Connor – Coq
  - Incompleteness for any "nice" extension of NN
- 2004 – Harrison – HOL-Light
  - Incompleteness for any $\sum_1$-complete system with $\sum_1$-definable axioms

The Good

- No changes in Coq were made.
- Dependent types ensures that all formulas are well-formed.
- A determined novice Coq user can prove difficult theorems.

The Bad

- Dependent types are hard to work with.
- Program Extraction will not produce the Gödel sentence.
  - Neither possible nor practical
- Used a lot of cut and paste
  - This is a problem in other programming languages as well.

The Future

- Gödel's second incompleteness theorem
  - Formalize the Hilbert-Bernays-Löb derivability conditions:
    1. If PA $\vdash \varphi$ then PA $\vdash \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\varphi\urcorner)$
    2. PA $\vdash \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\varphi\urcorner) \Rightarrow \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\mathrm{Pr}_{\mathrm{PA}}(\ulcorner\varphi\urcorner)\urcorner)$
    3. PA $\vdash \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\varphi \Rightarrow \psi\urcorner) \Rightarrow \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\varphi\urcorner) \Rightarrow \mathrm{Pr}_{\mathrm{PA}}(\ulcorner\psi\urcorner)$

The Future

- Gödel's second incompleteness theorem
  - Formalize the Hilbert-Bernays-Löb derivability conditions:
    1. If $PA \vdash \varphi$ then $PA \vdash Pr_{PA}(\ulcorner \varphi \urcorner)$
    2. $PA \vdash Pr_{PA}(\ulcorner \varphi \urcorner) \Rightarrow Pr_{PA}(\ulcorner Pr_{PA}(\ulcorner \varphi \urcorner) \urcorner)$
    3. $PA \vdash Pr_{PA}(\ulcorner \varphi \Rightarrow \psi \urcorner) \Rightarrow Pr_{PA}(\ulcorner \varphi \urcorner) \Rightarrow Pr_{PA}(\ulcorner \psi \urcorner)$

The 2nd condition is the hardest to prove

Statistics

- Took me approximately 16 months
  - Shankar took about 18 months
- Proof Size
  - 46 files
  - 7 036 lines of specifications
  - 37 906 lines of proof
  - 1 267 747 total characters.
- Information Content
  - 146 008 bytes (`gzip -9`)