Lecture 10: Proof and Computation

Michael Beeson

In this lecture, we turn back to logic, after our study of computability, and study the relations between provability and computability.

We will show that provability is in essence, just another kind of computation. It is that insight that underlies the Gödel incompleteness theorem.

Numerals

A *numeral* is a term in the sequence, $0, 0', 0'', \ldots$.

More formally, for each integer n we define the numeral for n, which is written $\bar{n}$, by

$$\bar{0} = 0$$

$$\overline{n+1} = \bar{n}'$$

In the first equation, $\bar{0} = 0$, the zero on the left is the natural number 0, and the 0 on the right is the constant symbol 0 of $\mathbf{PA}$. Using the same typeface for the two is perhaps misleading, but context actually permits only one interpretation.

Kleene uses boldface $\mathbf{x}$ instead of $\bar{x}$, but this has a serious disadvantage that it is difficult for use on a blackboard, whiteboard, or paper. The bar notation has become standard since Kleene's time. Also, we are using $\mathbf{x}$ for $x_1, \ldots, x_n$. We use $\bar{\mathbf{x}}$ to abbreviate $\bar{x}_1, \ldots, \bar{x}_n$.

Numerals are terms, not numbers

It is very important to realize the difference.

- A numeral is a term, that is, a syntactic expression, a sequence of symbols.
- It *denotes* a natural number.
- Example: when we write $\bar{0} = 0$ we mean that the term $\bar{0}$ is the constant symbol 0. The equal sign is not the $=$ symbol of $\mathbf{PA}$, it just means "is".
- The interpretation of $\bar{3}$ in the standard model of $\mathbf{PA}$ is the natural number three.

Value of a term

A closed term is one that has no variables. For example, numerals, or for a second example, $0'' + 0''$.

The value of a closed term t is the element of $\mathbb{N}$ that t denotes in the standard model. That is defined as follows:

$$\begin{array}{rcll}
Val(0) & is & 0 & \text{on the left, the constant 0;} \\
 & & & \text{on the right, the number 0} \\
Val(t+s) & is & Val(t) + Val(s) & \text{on the left, } + \text{ is a symbol;} \\
 & & & \text{on the right, addition} \\
Val(t \cdot s) & is & Val(t) \times Val(s) & \\
Val(t') & is & Val(t) + 1 &
\end{array}$$

Provable evaluation of closed terms

If t is a closed term, and $k = Val(t)$, then

$$\vdash t = \bar{k}$$

You will prove this in Exercise 10.3.

For example, with $2 = 0''$ and $4 = 0''''$, we have

$$\vdash 2 + 2 = 4$$

This is another one of the esoteric mysteries of mathematical logic, cleverly disguised as a commonplace fact. In an exercise, you will exhibit a complete proof.

Some useful lemmas

$$\vdash x + 0 = 0 + x$$

$$\vdash x + y' = x' + y$$

The first is proved in **PA** by induction on x, and then the second is proved (using the first) by induction on y.

Incidentally, using the second lemma one easily proves $x + y = y + x$. The diligent student will also want to check that $\mathbf{PA}$ proves the commutativity of multiplication and the distributive law, but we do not need these results at present.

Provability of true closed inequalities

We will prove (informally) that for each k,

$$\vdash \bar{k} + \bar{n}' \neq \bar{k}.$$

The base case $0 + \bar{n}' \neq 0$ follows from $0 + x = x$ and $x' \neq 0$. For the induction step, we need to show that

$$\vdash \bar{k}' + \bar{n}' \neq \bar{k}'.$$

We have

$$\begin{array}{rl} \vdash \bar{k} + \bar{n}' \neq \bar{k} & \text{induction hypothesis} \\ \vdash \bar{k} + \bar{n}' = \bar{k}' + \bar{n} & \text{using } \vdash x' + y = x + y' \\ \text{Therefore} & \vdash \bar{k}' + \bar{n} \neq \bar{k} \end{array}$$

By the axiom $x' = y' \supset x = y$, we have

$$\vdash (\bar{k}' + \bar{n})' \neq \bar{k}'$$

Then using $(x + y)' = x + y'$, we have

$$\vdash k' + n' \neq k'.$$

That was what we had to show.

Provability of true closed inequalities

We showed that for all k and n,

$$\vdash \bar{k} + \bar{n}' \neq \bar{k}.$$

Now we claim that if $k \neq m$, then $\vdash \bar{m} \neq \bar{k}$. It suffices, since $\vdash x = y \supset y = x$, to assume $m = k + (n + 1)$ for some n. Then $Val(\bar{k} + \bar{n}') = m$ and hence $\vdash \bar{k} + \bar{n}' = \bar{m}$. But $\vdash \bar{k} + \bar{n}' \neq \bar{k}$. Hence $\vdash \bar{m} \neq \bar{k}$, because $\vdash x = y \wedge x \neq z \supset x \neq z$.

Provability of true closed inequalities

Let t and s be closed terms. If $Val(t)$ and $Val(s)$ are not the same integer, then $\vdash t \neq s$.

Proof. Let m and k be the values of t and s respectively. Then $\vdash t = \bar{m}$ and$\vdash s = \bar{k}$. Since m and k are different, we have $\vdash \bar{m} \neq \bar{k}$. Hence $\vdash t \neq s$.

This works for closed terms only

If we allow variables, then the inequation

$$t(\mathbf{x}) \neq s(\mathbf{x})$$

expresses that a Diophantine equation has no solutions.

(To call an equation Diophantine means that it is a polynomial equation for which we want integer solutions.)

There are certainly lots of Diophantine equations of which we don't know whether they do or don't have solutions. More about this after we prove the incompleteness theorem. For now, the point is to emphasize that **closed** terms can be provably evaluated.

Free and bound variables

- I am assuming you know what it means: a given occurrence of a variable x is **free** in a formula A, or is **bound**.
- If a variable is bound, then it is bound by a certain quantifier.
- The **scope** of a quantifier: the scope of the quantifier at the beginning of a quantified subformula is the rest of the subformula.

Example. In

$$\forall x\,(x + 0' = x') \wedge x * x = y$$

what is the scope of the $\forall x$? Which occurrences of x are free and which are bound?

Notation: $A(x)$ versus just A

We say, "let A be a formula." Here the letter "A" is an informal variable ranging over formulas. It may stand for a particular formula or for some generic unspecific formulae. There is no implication about what variables are free in A and what variables are not free in A.

We also say, "let $A(x)$ be a formula."

- Here we imply that x *might* occur free in A.
- But, it might not actually occur at all in A.
- And of course, there is no implication that it does not occur bound.
- In short, the notation does not really commit us to anything more than "let A be a formula".
- It just calls our attention to a particular variable.

Substitution

An important operation is substituting a term t for a particular free variable x in a formula A. We establish now a precise notation for that:

$$A[x := t]$$

which is read aloud as "A with t for x", or "A with x gets t."

We use a similar notation for **simultaneous substitution**:

$$A[x, y := t, s] \qquad \text{or} \qquad A[x_1, \ldots, x_n := t_1, \ldots, t_n]$$

Note that simultaneous substitution is not the same as sequential substitution. We do not have $A[x, y := t, s] = A[x := t][y := s]$. For example, if t is y and s is a constant, the former contains y but the latter does not.

Substitution variations

- Kleene's (and my) version ignores bound occurrences of x, substituting only for the free occurrences.
- It's also possible to first rename all bound occurrences of x and then substitute for all occurrences.
- It's even possible to use a completely different stock of variables for bound and for free variables.
- These things have to be carefully considered when doing computer implementations of logic, but for a theory course, Kleene's version is fine.

Substitution notations

- In Kleene, you don't see $A[x := t]$. Instead, Kleene first calls attention to the variable x by using the notation $A(x)$. Then instead of writing $A[x := t]$, he writes $A(t)$.
- In my opinion, this breaks substitution into a two-step process, and is not ideal because the variable x is not visible in the expression $A(t)$ when the substitution is made.
- Kleene's notation is usually adequate, but occasionally more precision is needed.
- In other books, you may see $A[x/t]$ or $A[t/x]$ or with backslashes. It's confusing, but both notations are in use!

Notation $\exists ! x\, A(x)$

This is read "there exists a unique x such that $A(x)$."

It abbreviates the formula

$$\exists x\, (A(x) \wedge \forall y\, (A(y) \supset x = y)).$$

That is equivalent to

$$\exists x\, A(x) \wedge \forall x, y\, (A(x) \wedge A(y) \supset x = y).$$

Kleene (p. 199) chooses the former, so we follow him in taking that to be the official definition.

Representability

By the phrase "number-theoretic function", we mean a function from $\mathbb{N}^n$ to $\mathbb{N}$, for some $n \geq 1$; that is, a total function of n natural-number arguments. As usual we use $\mathbf{x}$ for $x_1, \ldots, x_n$. From now on, provability refers to provability in $\mathbf{PA}$ unless otherwise stated.

Definition

A number-theoretic function f is **representable**, or **representable in $\mathbf{PA}$**, if there is a formula $A(\mathbf{x}, y)$ such that

(i) for all $\mathbf{x}$, $\vdash A(\bar{\mathbf{x}}, \bar{y})$ if $y = f(\mathbf{x})$, and
(ii) for all $\mathbf{x}$, $\vdash \exists! y\, A(\bar{\mathbf{x}}, y)$.

Similarly, a predicate $P(\mathbf{x})$ is representable if and only if there is a formula $A(\mathbf{x})$ such that

(iii) $P(\bar{\mathbf{x}})$ implies $\vdash A(\bar{\mathbf{x}})$, and
(iv) $P(\bar{\mathbf{x}})$ implies $\vdash \neg A(\bar{\mathbf{x}})$.

Something to note

In the definition

(iii) $P(\bar{\mathbf{x}})$ implies $\vdash A(\bar{\mathbf{x}})$, and
(iv) $P(\bar{\mathbf{x}})$ implies $\vdash \neg A(\bar{\mathbf{x}})$.

Note the word is **implies**, not **if and only if**. We don't want to say that something is not provable, so that something being representable won't contain a hidden claim that $\mathbf{PA}$ is consistent.

Terminology

- What we called “representable” is called “numeralwise representable” in Kleene (page 200 for functions, p. 195 for predicates).
- The adjective “numeralwise” is not necessary, as there is no notion in Kleene without the adjective, and other textbooks nowadays just say “representable.”
- The reason Kleene used the adjective is explained in the remark after the definition, and further amplified in Exercise 10.4.

Discussion of representability

(i) for all x, $\vdash A(\bar{\mathbf{x}}, \bar{y})$ if $y = f(x)$, and
(ii) for all $\mathbf{x}$, $\vdash \exists! y\, A(\bar{\mathbf{x}}, y)$.

- condition (ii) is much weaker than requiring $\vdash \forall x \exists! y\, A(\mathbf{x}, y)$.
- It is only required to find one proof for each *particular* $\bar{x}$.
- These proofs can just encode direct calculations of $f(x)$.
- To prove $\forall \mathbf{x} \exists! y A(\mathbf{x}, y)$, we would need to give a reason why $f(x)$ is *always* defined.
- We might have no idea at all why $f(\mathbf{x})$ should always be defined, yet be able to verify it for every particular $\mathbf{x}$.

Representable predicates

(iii) $P(x)$ is true implies $\vdash A(\bar{\mathbf{x}})$, and
(iv) $P(x)$ is false implies $\vdash \neg A(\bar{\mathbf{x}})$.

- In Exercise 10.8, you will prove that a predicate is representable, if and only if its representing function is representable.

Predicates defined by an atomic formula are representable

An **atomic** formula has no logical symbols. A synonym (used by Kleene) is **prime** formula. In $\mathbf{PA}$ the only atomic formulas are equalities between two terms.

Let t and s be two terms possibly containing $\mathbf{x}$. Consider the predicate $P(x)$ defined by

$$\langle \mathbb{N}, +, \cdot, succ, 0 \rangle \models t(\bar{\mathbf{x}}) = s(\bar{\mathbf{x}}).$$

To show that P is representable we must show

(iii) $t(\bar{\mathbf{x}}) = s(\bar{\mathbf{x}})$ is true implies $\vdash t(\bar{\mathbf{x}}) = s(\bar{\mathbf{x}})$

(iv) $t(\bar{\mathbf{x}}) = s(\bar{\mathbf{x}})$ is false implies $\vdash t(\bar{\mathbf{x}}) \neq s(\bar{\mathbf{x}})$

Let m and k be the values of $t(\bar{\mathbf{x}})$ and $s(\bar{\mathbf{x}})$, respectively. Then $\vdash t(\bar{\mathbf{x}}) = \bar{m}$ and $\vdash s(\bar{\mathbf{x}}) = \bar{k}$. Then, by what we proved about inequalities of closed terms, either $\vdash t(\bar{\mathbf{x}}) = s(\bar{\mathbf{x}})$ (if $m = k$) or $\vdash t(\bar{\mathbf{x}}) \neq s(\bar{\mathbf{x}})$ (if $m \neq k$).

Bounded arithmetic predicates are representable

Indeed, every arithmetic predicate is represented by any bounded arithmetic formula that defines it.

Just to clarify:

- An arithmetic formula $A(\mathbf{x})$ is a string of symbols.
- It defines a predicate by

$$P(\mathbf{x}) := \langle \mathbb{N}, +, \cdot, succ, 0 \rangle \models A(\bar{\mathbf{x}})$$

- So we are claiming that in that case, P is also represented by A, as well as defined by A.

Overview of the proof that bounded arithmetic predicates are representable

- We prove it by induction on the length of the formula defining the bounded arithmetic predicate.
- We did the base case already, when the formula is atomic.
- For the induction step, we prove that the representable predicates are closed under the logical connectives and under bounded quantification.
- Pages 201–202 of Kleene contains this material, but we will do some of the details here.

One case of the proof

Consider conjunction. Suppose A represents P and B represents Q. That is, $P(x)$ is true implies $A(\bar{x})$ is true in $\langle \mathbb{N}, +, \cdot, succ, 0 \rangle$, and if and only if $\vdash A(\bar{x})$, and $P(x)$ is false implies $\vdash \neg\, A(\bar{x})$. Then we claim $A \wedge B$ represents the predicate "P and Q". So we must prove

(iii) "$P(x)$ and $Q(x)$" is true implies $\vdash A(\bar{\mathbf{x}}) \wedge B(\bar{\mathbf{x}})$, and
(iv) "$P(x)$ and $Q(x)$" is false implies $\vdash \neg(A(\bar{\mathbf{x}}) \wedge B(\bar{x}))$.

If "$P(x)$ and $Q(x)$" is true, then $P(x)$ and $Q(x)$ are both true, so by induction hypothesis $\vdash A(\bar{x})$ and $\vdash B(\bar{x})$. Hence $\vdash A(\bar{x}) \wedge B(\bar{x})$, which is of course $(A \wedge B)[x := \bar{x}]$. Moreover the converse is also immediate.
If "$P(x)$ and $Q(x)$" is false, then either $P(x)$ is false or $Q(x)$ is false. Then either $\vdash \neg A(\bar{x})$ or $\vdash \neg\, B(\bar{x})$. In either case, we have

$$\vdash \neg\,(A(\bar{x}) \wedge B(\bar{})).$$

in view of the tautologies

$$\neg A \supset \neg\,(A \wedge B)$$

The case of bounded quantification

There are two ways to define $x < n$. Kleene (p. 196 and p. 229) uses $\exists m\,(m' + x = n)$. The alternative would be $\exists m\,(x + m' = n)$. We follow Kleene here.

A basic lemma is that, for example,

$$\vdash x < \bar{3} \equiv x = 0 \vee x = 0' \vee x = 0''$$

or more generally

$$\vdash x < \bar{n}' \equiv x = 0 \vee x = 1 \ldots \vee x = \bar{n}$$

- You will prove this lemma and related results in the exercises.
- Thus a predicate defined by a quantifier "up to" a numeral is provably equivalent to a conjunction or disjunction. But since every closed term is provably equivalent to a numeral, this covers the general case.

Bounded arithmetic predicates are representable

We have outlined all the important steps of the proof. Compare the lecture slides to Kleene's treatment around page 200.