Logic as a Formal System

Lecture notes for COM3412 *Logic and Computation*

16th February 2009

1 Introduction

Historically, logic and computation have been deeply intertwined, with major developments in each field inspired by those in the other. This interdependence continues to the present day. In this module we shall look in detail at some of the major points of interaction (though inevitably there will be much that we don't cover).

A theme which runs through all the things we'll be looking at is the correlation between logical and computational difficulty. Roughly speaking, tasks that can be specified with limited logical resources are more amenable to computational handling that those which require more complex resources to specify. The main particular results which bear this out are:

- The decision problem for Propositional Calculus inferences whose constituent formulae each contain at most two schematic letters is *tractable*, i.e., can be solved by an algorithm running in polynomial time.
- The decision problem for arbitrary Propositional Calculus inferences is *NP-complete*: which means that although it is solvable, it belongs to a class of algorithms for which the best known algorithms run in exponential time.
- The decision problem for (first-order) Predicate Calculus inferences is *semi-decidable*, in that there exists a procedure for testing validity which is correct in the sense that if it terminates it will deliver the correct answer, but is only guaranteed to terminate in the case of valid inferences.
- The decision problem for first-order arithmetic (or equivalently, for pure second-order logic) is not even semi-decidable: there is (provably) no algorithm that is guaranteed to give the correct answer even in cases where it terminates.

We shall start by looking at some properties of first-order logic, with an emphasis on the computational or algorithmic aspects, and introduce the idea of a logical theory. We then embark on a survey of some of the major landmarks of the field: the Halting Problem for Turing Machines and its relation to the decision problem for first-order logic, Gödel's completeness and incompleteness theorems, and Cook's Theorem, which shows that the decision problem for the Propositional Calculus is NP-complete.

2 Properties of valid inference

We have defined validity by the rule that an inference is valid iff any model for the premisses satisfies the conclusion. We say that the conclusion is a **logical consequence** of the premisses, written $\Sigma \models C$, where Σ is the set of premisses, and C is the conclusion. A number of properties follow directly from this definition:

- **Monotonicity.** If $\Sigma \models C$, and $\Sigma \subseteq \Sigma'$, then $\Sigma' \models C$. In other words, logical consequence is so robust that the addition of extra premisses can never force you to retract it. (Compare this with what often passes for consequence in everyday reasoning.)

  The proof of monotonicity is trivial: if $\Sigma \subseteq \Sigma'$ then any model for Σ' must be a model for Σ, so if every model for Σ satisfies C then so does every model for Σ'.

- **Cut.** If $\Sigma \models A$ and $\Sigma \cup \{A\} \models C$ then $\Sigma \models C$. (This is helpful for doing proofs: it says that you can make use of intermediate results or "lemmas", here represented by A.)

  If $\Sigma \models A$ then any model for Σ satisfies A and is therefore a model for $\Sigma \cup \{A\}$. If $\Sigma \cup \{A\} \models C$ then any model for $\Sigma \cup \{A\}$ satisfies C. Hence if both inferences are valid, then any model for Σ satisfies C.

If we bring in the truth tables then more properties follow, corresponding to some familiar methods of proof:

- **Conditional Proof.** If $\Sigma \cup \{A\} \models C$ then $\Sigma \models A \rightarrow C$. Thus in order to prove a conditional, assume the antecedent and derive the consequent. (This corresponds to the "if-introduction" rule.)

  Any model for Σ satisfies either A or $\neg A$. If it satisfies A then it is a model for $\Sigma \cup \{A\}$ and hence, since $\Sigma \cup \{A\} \models C$, satisfies C. By the truth table, $A \rightarrow C$ is true whenever C is true, so the model satisfies $A \rightarrow C$ in this case. If on the other hand it satisfies $\neg A$ then by the truth table it also satisfies $A \rightarrow C$. Hence every model for Σ satisfies $A \rightarrow C$.

- **Proof by Contradiction.** If $\Sigma \cup \{A\} \models B \wedge \neg B$ then $\Sigma \models \neg A$. If assuming A leads to a contradiction, then A must be false.

  Any model for $\Sigma \cup \{A\}$ would satisfy $B \wedge \neg B$, which is impossible from the truth tables. Hence there are no such models. This means that no model for Σ can satisfy A, and hence any model for Σ must satisfy $\neg A$.

- **Proof by Cases.** If $\Sigma \models A \vee B$, $\Sigma \cup \{A\} \models C$, and $\Sigma \cup \{B\} \models C$, then $\Sigma \models C$. (This corresponds to the "or-elimination" rule.)

  Since $\Sigma \models \{A \vee B\}$, any model for Σ satisfies $A \vee B$; by the truth table it must satisfy either A or B. If the former, it is a model for $\Sigma \cup \{A\}$ and hence satisfies C; if the latter, it is a model for $\Sigma \cup \{B\}$ and so again satisfies C. In either case C is satisfied.

An additional property which does not follow from the general definition of validity, but which can be proved to hold for the First-Order Predicate Calculus, is:

- **Compactness.** If $\Sigma \models C$ then there is a finite subset $\Sigma_0 \subseteq \Sigma$ such that $\Sigma_0 \models C$. This means that no valid inference in first-order logic can require infinitely many premisses: if infinitely many are given, all but a finite number of them can be discarded without detracting from the validity of the inference. (Note that this property does not hold for second-order logic.)

Compactness is such an important property that we now devote a whole section to it.

3 Compactness

First-order logic can be proved to be compact. (This follows from the fact that there exist sound and complete proof systems for it—see the section on proof systems for an explanation of this.) This has important implications for the expressive power of the logic. For example, we can use compactness to show that we can't define the notions of "finite" and "infinite" in first-order logic.

To see why, suppose we have a formula ϕ, involving the predicate P, which someone claims expresses the proposition "Infinitely many objects are P". For this claim to be correct, it must be the case that an interpretation satisfies ϕ if and only if it contains infinitely many objects for which the predicate P holds. Now consider the following sequence of formulae:

$$\begin{aligned}
\phi_1 &: \exists x P(x) \\
\phi_2 &: \exists x, y(P(x) \wedge P(y) \wedge x \neq y) \\
\phi_3 &: \exists x, y, z(P(x) \wedge P(y) \wedge P(z) x \neq y \wedge x \neq z \wedge y \neq z) \\
&\ldots
\end{aligned}$$

In this sequence, ϕ_n says that there are at least n objects with the property P. Let $\Sigma = \{\phi_1, \phi_2, \phi_3, \ldots\}$. Then in any model for Σ, there must be at least 1 P, at least 2 Ps, at least 3 Ps, and so on. For any $n \in \mathbb{N}$, there must be at least n Ps. The only way of satisfying these requirements is by having a model with infinitely many Ps. Hence $\Sigma \models \phi$.

Now we invoke compactness. This says that there must be some *finite* subset $\Sigma_0 \subset \Sigma$ such that $\Sigma_0 \models \phi$. Since Σ_0 is finite, there is a largest value of n for which $\phi_n \in \Sigma_0$. In that case $\Sigma_0 \subseteq \{\phi_1, \phi_2, \ldots, \phi_n\}$, so we have $\{\phi_1, \phi_2, \ldots, \phi_n\} \models \phi$. Note further that ϕ_n ("there are at least n Ps") implies ϕ_{n-1} ("there are at least $n-1$ Ps"), and all the other formulae in Σ_0. Hence we can say that $\{\phi_n\} \models \phi$.

What does this imply? It means that any model for ϕ_n must satisfy ϕ. But any interpretation in which there are at least n Ps is a model for ϕ_n. In particular any interpretation in which there are exactly n Ps is a model for ϕ_n. Hence any such interpretation satisfies ϕ. This has only finitely many Ps and hence contradicts the claim that an interpretation satisfies ϕ if and only if it contains infinitely many Ps.

It follows that ϕ cannot, after all, be a correct expression in first-order logic of the proposition "There are infinitely many Ps", and hence that first-order logic cannot express this proposition.

Do not confuse this fact with the following, different, fact: it is possible to write down a first-order formula involving P which only has models containing infinitely many Ps. An example of such a formula would be the $A \wedge B \wedge C \wedge D$, where

$$\begin{aligned}
A &\equiv \forall x \neg R(x, x) \\
B &\equiv \forall x \forall y \forall z(R(x, y) \wedge R(y, z) \rightarrow R(x, z)) \\
C &\equiv \exists x P(x) \\
D &\equiv \forall x(P(x) \rightarrow \exists y(P(y) \wedge R(x, y))
\end{aligned}$$

Here A and B say that R is an irreflexive, transitive relation; C says that there's at least one P, and D says that any P is related by R to another P. This generates an infinite sequence of Ps each of which is R-related to the next. Since R is transitive and irreflexive, this means that no two members of the sequence are equal, and hence that there are infinitely many Ps.

Thus we can, in first-order logic, write down a formula which only has models containing infinitely many Ps; but that is not the same as writing down a formula which is satisfied by an interpretation *if and only if* that interpretation has infinitely many Ps, which is what the compactness argument shows to be impossible.

What kind of logic *can* we use to say there are infinitely many Ps? We need **Second-order Logic** for this. In first-order logic, predicates (and indeed functions) can only apply to terms, i.e., to expressions denoting individuals (e.g., individual constants and variables). Thus predicates express properties and relations of individuals. These are first-order properties and relations. In second-order logic we can have predicates expressing properties of first-order properties and relations, i.e., second-order properties. Likewise, whereas in first-order logic we can only quantify over individuals, in second-order logic we can quantify over first-order properties. Thus we can write formulae such as

$$\forall P((P(a) \wedge \mathcal{G}(P) \rightarrow P(b)) \wedge (P(a) \wedge \neg \mathcal{G}(P) \rightarrow \neg P(b)))$$

which may be interpreted as saying "Bob has all Alan's good qualities and none of his bad ones". (Note that I'm using "calligraphic" script for second-order predicates.)

Now, how do we use second-order logic to say that there are infinitely many Ps? First we need to know what "infinite" really means; it turns out that the neatest way of characterising it is as follows: *a set is infinite if and only if it can be put into one-to-one correspondence with a proper subset of itself.* That is, there is a bijection between the set and one of its proper subsets. An example is the following bijection between the natural numbers and the square numbers:

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & 5 & \cdots \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\ 0 & 1 & 4 & 9 & 16 & 25 & \cdots \end{array}$$

For each natural number there is a unique square number, and for each square number there is a unique natural number, and yet the set of square numbers is a proper subset of the set of square numbers. You can't do this with a finite set: a finite set always has *more* elements than any of its proper subsets, so you can't set up an appropriate bijection.

In second-order logic, we can talk about sets by talking about the properties which characterise them. Thus to talk about a set S we set up a predicate $P(x)$ meaning "x is a member of S". To say that there is a bijection between the sets represented by predicates P and Q, we must say there is a relation R such that every P is R-related to a unique Q (so R acts as an injective function mapping Ps to Qs) and that for every Q there is a P which is R related to it (so that R is surjective). In second-order logic:

$$\begin{aligned} \exists R(&\forall x(P(x) \rightarrow \exists y(Q(y) \wedge R(x,y))) \wedge \\ &\forall x \forall y \forall z(R(x,y) \wedge R(x,z) \rightarrow y = z) \wedge \\ &\forall x \forall y \forall z(R(x,z) \wedge R(y,z) \rightarrow x = y) \wedge \\ &\forall x(Q(x) \rightarrow \exists y(P(y) \wedge R(y,x)))) \end{aligned}$$

The first line says that every P is R-related to at least one Q, the second says that R is an function (no x is R-related to more than one Q), the third says that R is injective (same output implies same input), and the fourth says that R is surjective (every Q has at least one P related to it). Let's abbreviate this big formula to $\mathcal{B}(P,Q)$ (i.e., $\mathcal{B}$ for bijection—it's in calligraphic script because it's a second-order predicate). Then to say there are infinitely many Ps we must say that there is a bijection between P and one of its proper subsets, i.e.,

$$\exists Q(\forall x(Q(x) \rightarrow P(x)) \wedge \exists x(P(x) \wedge \neg Q(x)) \wedge \mathcal{B}(P,Q)).$$

The first conjunct says that the set of Qs is a subset of the set of Ps, and the second conjunct ensures that it is a proper subset; the third says there's a bijection between them.

This formula says that there are infinitely many Ps. It is satisfied by all and only those interpretations in which there are indeed infinitely many Ps. It is a second-order formula both because it quantifies over first-order predicates (this is shown by the "$\exists Q$") and because it contains a second-order predicate ("$\mathcal{B}$"). We know already that no first-order formula can do the same job.

4 Proof Systems

A proof system for a logic is a system for constructing proofs. A proof is a demonstration that some inference in the logic is valid or invalid. The demonstration must be complete in a finite number of steps and must use only some finitely-specifiable set of rules laid down in advance. A proof system, in effect, separates all inferences into three classes, those which it certifies as valid, those which it certifies as invalid, and those for which it does not come to either conclusion (we assume that the system produces at most one answer for each inference, so it doesn't certify any inferences as both valid and invalid—in other words that it is *consistent*).

4.1 Soundness, Completeness, and Decidability

The terms used for this are defined as follows (throughout, when we talk about a proof system, without further qualification, we mean a proof system for first-order predicate calculus):

- A proof system is **sound** if every inference it certifies as valid is in fact valid.
- A proof system is **complete** if it certifies as valid every inference which is in fact valid.
- A proof system is a **decision procedure** if it provides an algorithm which can determine, for any inference, whether or not it is valid.

A familiar example of a decision procedure is the method of truth tables applied to the Propositional Calculus.

The system of Natural Deduction outlined in the Logical Preliminaries can be shown to be both sound and complete. It is sound because whenever you use it to derive a conclusion from some premisses, the conclusion does in fact logically follow from the premisses; and it is complete because for any valid inference there is a derivation of the conclusion from the premisses. (Note that validity and provability are different concepts; the point of the proof system is to set up provability as a *criterion* for validity.) On the other hand Natural Deduction is not a decision procedure: an inference is invalid if its conclusion can't be derived from its premisses, but Natural Deduction doesn't provide any means for determining this.

When we turn to the Predicate Calculus, we find that the system of Natural Deduction is still both sound and complete. This was first established (for an equivalent system) by Gödel in 1930; this is **Gödel's Completeness Theorem**—very important, but it tends to be overshadowed by the more charismatic Incompleteness Theorem. As with the Propositional Calculus, though, it is not a decision procedure. The problem of finding a decision procedure for first-order logic exercised the best logical minds for the first third of the 20th century, having been posed as a key problem (often known by its German name: *Entscheidungsproblem*) in the year 1900.

Note that the Completeness Theorem implies compactness (as defined at the end of §2). If $\Sigma \models C$ then by completeness there is a proof of C using only premisses from Σ. Any such proof is of finite length, and therefore uses only some finite subset $\Sigma_0 \subseteq \Sigma$ of the premisses. It is therefore a proof of C using only premisses from Σ_0. Hence $\Sigma_0 \models C$, as required.

The Truth Trees method you studied in the second year is also a sound and complete proof system, but it is not a decision procedure.

It *is* a decision procedure for the Propositional Calculus. For any finite set of formulae of the Propositional Calculus, the truth tree must eventually be completed; if every branch is then closed, the set is unsatisfiable, otherwise it is satisfiable.

In the Predicate Calculus, it is not hard to find a set for which the truth tree can never be completed. The problem comes from the rules for dealing with existential and negated universal sentences, which require one to introduce new constants. This leads to an open-endedness which in some cases prevents the tree from being completed.

Example. We shall try to find a model for the set $\{\neg P(a,a), \forall x \exists y P(x,y)\}$.

$$
\begin{array}{ll}
 & 1.\quad \neg P(a,a) \\
 & 2.\quad \forall x \exists y P(x,y) \\
 & \text{---------------} \; 2[x/a] \\
\times & 3.\quad \exists y P(a,y) \\
 & \text{---------------} \; 3[y/b] \\
 & 4.\quad P(a,b) \\
 & \text{---------------} \; 2[x/b] \\
\times & 5.\quad \exists y P(b,y) \\
 & \text{---------------} \; 5[y/c] \\
 & 6.\quad P(b,c) \\
 & \text{---------------} \; 2[x/c] \\
\times & 7.\quad \exists y P(c,y) \\
 & \text{---------------} \; 7[y/d] \\
 & 8.\quad P(c,d) \\
 & \quad \vdots
\end{array}
$$

We have set in train a never-ending process. Each time we use the $\exists$-rule to generate a new constant we can then use that constant to generate yet another instance of (1) using the $\forall$-rule. At no stage do we generate a contradiction (this could only come from $P(a,a)$), and the tree never completes.

And yet the original set (1,2) does have a model. For example, choose any domain with more than one element, and let $P(x,y)$ denote the relation "x is different from y" (i.e., the negation of the identity relation. Then (1) is satisfied since it says that a is not different from itself, and (2) is satisfied since it says that given any element we can find an element different from it, which is certainly true if the domain has at least two elements. Thus the inference

$$\frac{\forall x \exists y P(x,y)}{P(a,a)}$$

is invalid, but its invalidity cannot be demonstrated by means of a truth tree. That shows that truth trees do not provide a decision procedure for the predicate calculus.

5 First-order Theories

A first-order theory is a set Θ ("theta") of sentences of FOPC with the following properties:

1. Θ is satisfiable, i.e., Θ has a model.
2. Θ is closed under logical consequence, i.e., whenever $\Theta \models C$, we have $C \in \Theta$.

A theory is **categorical** if all its models are isomorphic, i.e., they differ only in respect of the labelling of the domain elements.

If you have a particular domain and fix on an interpretation of a suitable first-order language over this domain, you can talk about *the* theory of the domain, i.e., the set of all sentences true under this interpretation.

Example. The first order theory of *identity* includes the sentence $\forall x(x = x)$ (everything is identical to itself—i.e., identity is reflexive), $\forall x \forall y(x = y \rightarrow y = x)$ (symmetric), $\forall x \forall y \forall z(x = y \wedge y = z \rightarrow x = z)$ (transitive). It also includes infinitely many sentences of the form $\forall x \forall y(x = y \rightarrow (\Phi[x] \rightarrow \Phi[y]))$ which says that if x is identical to y then any true formula remains true when one or more occurrences

of x are replaced by y. (Here Φ represents a template standing for any formula P into which can be substituted a term, e.g., x or y; note that we get different theories of identity as we vary the stock of non-logical vocabulary available for constructing the formula P.)

It turns out that we can derive the entire first-order theory of identity from (1) the identity axiom $\forall x(x = x)$, together with (2) the infinitely many instances of the **axiom schema** $\forall x \forall y(x = y \rightarrow (\Phi[x] \rightarrow \Phi[y]))$ (collectively called Leibniz's Law). Thus (1) and (2) together constitute a **complete axiomatisation** of the theory. Note that although this set of axioms is infinite, it is **finitely-specifiable**, since the infinitely many axioms in (2) are covered by a single axiom schema.

To illustrate, we shall derive transitivity, leaving the (somewhat easier) case of symmetry as an *Exercise*.

Let Φ be the template $\forall z(\cdots = z \rightarrow x = z)$. With this instance of Φ, Leibniz's Law says

$$\forall x \forall y(x = y \rightarrow (\forall z(x = z \rightarrow x = z) \rightarrow \forall z(y = z \rightarrow x = z))).$$

Since $\forall z(x = z \rightarrow x = z)$ is a tautology, this simplifies to

$$\forall x \forall y(x = y \rightarrow \forall z(y = z \rightarrow x = z)),$$

which is easily seen to be equivalent to

$$\forall x \forall y \forall z(x = y \wedge y = z \rightarrow x = z).$$

[This uses an equivalence with which you should be familiar: $A \rightarrow (B \rightarrow C) \cong (A \wedge B) \rightarrow C$, and another with which you may not be: $A \rightarrow \forall x B \cong \forall x(A \rightarrow B)$, where x does not occur in A.]

The first-order theory of identity is not, in fact, categorical. It is satisfied by any equivalence relation for which equivalent elements cannot be distinguished using only the predicates expressible in the language. Thus we cannot define identity uniquely in first-order logic, and for this reason it is customary to extend first-order logic by the addition of "=" as a logical constant (i.e., not part of the non-logical vocabulary, and therefore not reinterpretable) which denotes the identity relation under every interpretation. This is called **first-order predicate calculus with identity**, and is the system we shall largely be using from now on.

Example. The first-order theory of the "less than" relation on the real numbers (written "$x < y$") is completely axiomatised by the formulae:

1.	$\forall x \neg(x < x)$	(*Irreflexive)*
2.	$\forall x \forall y \forall z(x < y \wedge y < z \rightarrow x < z)$	(*Transitive)*
3.	$\forall x \forall y(x = y \vee x < y \vee y < x)$	(*Linear)*
4.	$\forall x \exists y(y < x)$	(*Unbounded below)*
5.	$\forall x \exists y(x < y)$	(*Unbounded above)*
6.	$\forall x \forall y(x < y \rightarrow \exists z(x < z \wedge z < y))$	*(Dense)*

Take any formula containing "$<$" as its only non-logical symbol. Either it or its negation is true when interpreted as a statement about the real numbers. The true member of the pair is a logical consequence of the above axioms (i.e., is satisfied by any model for those axioms).

Consider, for example, the formula $\exists x \exists y(x < y \wedge y < x)$. Since "less than" is asymmetric, this formula is false under the real numbers interpretation. Hence its negation must be a logical consequence of the axioms. We can prove this as follows: if the unnegated formula is true then there exist domain elements x and y such that $x < y \wedge y < x$. By Axiom 2, this means that $x < x$, contradicting Axiom 1. Hence the unnegated formula must be false, so the negated formula is true.

Although 1–6 completely axiomatise the first-order theory of "less than" for the real numbers, this is not in fact a categorical theory. That is, the real numbers with "less than" provide only one model for the

theory, but there are others which are not isomorphic to it, for example the *rational* numbers with "less than". The first-order theory of "less than" for the rationals is the same as the first-order theory of "less than" for the reals, but these two mathematical structures are not isomorphic. For example, consider the following statement concerning a domain Δ:

S: There is a non-empty proper subset L of Δ such that every element of L is less than every element of $\Delta \setminus L$, L has no greatest element, and $\Delta \setminus L$ has no least element.

If Δ is the set of real numbers, then S is false: no such set L can exist (this follows from a fundamental property of the real numbers: if a set of real numbers has an upper bound, then it has a least upper bound). But if Δ is the rational numbers, we can easily exhibit a suitable set L, for example the set consisting of all rational numbers less than π. (Here $\Delta \setminus L$ has no least element since π itself is not a rational number.) Thus for the rational numbers, S is true, but for the real numbers S is false. Since the rationals and the reals have the same first-order theory for the "less than" relation, it follows that S cannot be expressed by any first-order formula. It is in fact a *second-order* statement, and that is because it quantifies over *sets* of domain elements (it has the form "There exists a set of domain elements L such that ..."), whereas in first-order logic all quantification is over the domain elements themselves.

6 First-order Arithmetic

The real prize would be a complete axiomatisation of the first order theory of addition and multiplication over the natural numbers, the theory known as "first-order arithmetic". This is an essential step in the enterprise of evolving a system for proving the truth or falsity of every mathematical statement: in other words, reducing mathematics to a mechanical system. As you probably know, Gödel showed that this can't be done, but before we discuss his work it is important to have an understand of just what it is he showed to be impossible[1].

The language of first-order arithmetic contains the following non-logical vocabulary:

- The constant symbol "0";
- A 1-place function symbol "s" (which we shall write as a prefix operator without brackets), and two two-place function symbols "$+$" and "$*$" (written as infix operators).

That is all! The logical symbols are all the usual ones (connectives and quantifiers), including identity "=". Note that there are no predicates in the non-logical vocabulary; the only predicate is the logical constant "=".

The **standard interpretation** of the non-logical vocabulary is just what you expect:

- "0" denotes the number zero.
- "s" denotes the successor function.
- "+" denotes the addition function.
- "$*$" denotes the multiplication function.

First-order arithmetic consists of all sentences of this language which are true under the standard interpretation. Here are some examples:

- $\forall x(x + x = ss0 * x)$.
  This says that for any natural number x, $x + x = 2x$.

[1] A lot of hot air on the subject of Gödel's theorem would be avoided if people understood more exactly what it says; but even people who understand it pretty well have been led to deduce some very shaky conclusions from it.

- $\forall x \exists y \exists z \exists w \exists v (x = y * y + z * z + w * w + v * v)$.
  This says that every number is the sum of four squares.

A sentence in this language whose truth value is still unknown is *Goldbach's Conjecture*:

$$\forall x \exists y \exists z (Prime(y) \wedge Prime(z) \wedge ssx + ssx = y + z),$$

where the predicate $Prime$ is defined by

$$Prime(x) \equiv \forall u \forall v (u * v = x \rightarrow u = x \oplus v = x).$$

(Here "$\oplus$" is the *exclusive "or"* connective, defined by $A \oplus B \equiv (A \wedge \neg B) \vee (\neg A \wedge B)$.) Goldbach's Conjecture says that every even number greater than 2 can be expressed as the sum of two primes. Try it: 4=2+2, 6=3+3, 8=3+5, 10=3+7, 12=5+7, 14=3+11, ... Your place in mathematical history will be assured if you can *either* find an even number which can't be expressed as the sum of two primes, *or* prove that there isn't one.

Our target is a finitely-specifiable set of axioms whose logical consequences comprise all and only the true sentences of first-order arithmetic.

The most plausible candidate that has been suggested contains the following axioms[2], known as the *Peano axioms*, after the Italian mathematician Giuseppe Peano (1858–1932):

S1 Zero is not the successor of any natural number:

$$\forall x \neg (sx = 0)$$

S2 Distinct numbers have distinct successors:

$$\forall x \forall y (sx = sy \rightarrow x = y)$$

A1 Addition of zero:

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

A2 Addition of a successor:

$$\forall x \forall y (x + sy = s(x + y))$$

M1 Multiplication by zero:

$$\forall x (x * 0 = 0)$$

M2 Multiplication by a successor:

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

Ind Induction schema:

$$\Phi(0) \wedge \forall x (\Phi(x) \rightarrow \Phi(sx)) \rightarrow \forall x (\Phi(x))$$

[2]Peano's axioms included two not given here, namely "Zero is a natural number" and "Every natural number has a unique successor". For us, these are implicit in the facts that (1) "0" is a constant in our non-logical vocabulary, and the domain of interpretation is the natural numbers, and (2) "s" is a function in our non-logical vocabulary.

The set of all logical consequences of these axioms are the theorems of **Peano arithmetic** (PA)—we'll look at some examples in one of the Group Meetings.

What is the relationship between PA and the true first-order arithmetic? We say that PA is **sound** if all its theorems are in fact true under the standard interpretation; and that it is **complete** if every sentence (in the vocabulary of PA) that is true under the standard interpretation is a theorem of PA.

It would be nice if PA were both sound and complete.

What Gödel showed is that *no finitely-specifiable axiom system for first-order arithmetic can be both sound and complete*. Thus we know that if PA is sound, then it isn't complete, and if it is complete then it isn't sound. We don't even know which of these two cases holds, though in practice we always assume the former. Moreover, if we replace PA by any other such system, the same conclusion follows.

We will discuss how Gödel proved his theorem below. Setting Gödel aside for the moment, suppose in fact PA were both sound and complete. Then if Goldbach's Conjecture were true, it would be a logical consequence of the axioms, and it it were false, its negation would be a logical consequence of the axioms. Since we have a complete proof system for first-order logic (e.g., Natural Deduction, or Truth Trees), there would exist a proof in such a system of either Goldbach's Conjecture or its negation (but not both!). What we would need then is an effective procedure for finding this proof, i.e., a decision procedure for first-order predicate calculus.