A Hierarchy of Results Linking Logic and Computation

Propositional Calculus (Two-letter formulae)	Decidable in Polynomial time
Propositional Calculus (General formulae)	Decidable but NP-complete
Predicate Calculus	Semi-decidable
First-order Arithmetic	Undecidable

Models, Satisfaction and Validity

An interpretation under which a formula F comes out true is said to **satisfy** the formula.

An interpretation which satisfies every formula in some set Σ is a **model** for Σ.

An inference is **valid** if every model for its premisses satisfies its conclusion. (This means that the truth of the premisses is sufficient to guarantee the truth of the conclusion.) We write $\Sigma \models C$.

The conclusion of a valid inference is a **logical consequence** of the premisses. If the set of premisses is empty, the conclusion is said to be **logically true** or **valid**.

Monotonicity

If $\Sigma \models C$ and $\Sigma \subseteq \Sigma'$, then $\Sigma' \models C$.

Proof. 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 $\Sigma \models C$.

Proof. If $\Sigma \models A$, any model for Σ satisfies A, so it's a model for $\Sigma \cup \{A\}$. If $\Sigma \cup \{A\} \models C$, any model for $\Sigma \cup \{A\}$ satisfies C. Hence if both inferences are valid, any model for Σ satisfies C, so $\Sigma \models C$.

Conditional Proof

If $\Sigma \cup \{A\} \models C$ then $\Sigma \models A \rightarrow C$.
(To prove a conditional, assume the antecedent and derive the consequent.)

Proof.
Any model for Σ satisfies either A or $\neg A$.

If it satisfies A, 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 $A \rightarrow C$, so $\Sigma \models A \rightarrow C$.

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.)

Proof.
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 $\neg A$, so $\Sigma \models \neg A$.

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.)

Proof.
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 it satisfies A, it is a model for $\Sigma \cup \{A\}$ and hence satisfies C.

If it satisfies B, it is a model for $\Sigma \cup \{B\}$ and so again satisfies C.

In either case C is satisfied, so $\Sigma \models C$.

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.

This property does not hold for second-order logic.

Application of Compactness

Finiteness is not a first-order property.

Suppose a first-order formula ϕ, containing predicate P, is put forward as expressing that "infinitely many objects are P".

Consider the formulae

$$\phi(n): \quad \exists x_1, \ldots, x_n (P(x_1) \wedge \cdots \wedge P(x_n) \wedge x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \cdots \wedge x_{n-1} \neq x_n)$$

Let $\Sigma = \{\phi_1, \phi_2, \phi_3, \ldots\}$. Any model for Σ must contain infinitely many objects satisfying P; hence $\Sigma \models \phi$.

By compactness $\Sigma_0 \models \phi$ for some finite subset $\Sigma_0 \subset \Sigma$. Let n be the largest number for which $\phi_n \in \Sigma_0$. Then $\{\phi_n\} \models \phi$. Hence any interpretation with exactly n objects satisfying P must satisfy ϕ.

So ϕ does not express the proposition "There are infinitely many Ps".

Expressing infinity in second-order logic

An infinite set is one which can be put into one-to-one correspondence with a proper subset of itself, i.e., there is a bijection between the set and one of its proper subsets.

If we write $\mathcal{B}(P,Q)$ to mean there is a bijection between the set of Ps and the set of Qs, then the formula

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

says there are infinitely many Ps.

The predicate second-order predicate $\mathcal{B}$ can be defined as follows:

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

Proof Systems

A **proof** is a demonstration, completed in a finite number of steps, that some inference is valid or invalid.

A **proof system** is a finitely-specifiable set of rules for constructing proofs.

Given any inference, a proof system might
(1) certify it as valid (by allowing the construction of a proof of validity),
(2) certify it as invalid (by allowing the construction of a proof of invalidity), or
(3) neither.

(We assume that the proof system is **consistent**, i.e., does not certify any inference as both valid and invalid.)

Properties of Proof Systems

- 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.

Truth Tables as a Proof System

In the Propositional Calculus, it is possible to construct a truth table for any inference.

You can read off from the truth table whether or not the inference is valid, and the answer is correct in every case.

Hence the method of truth tables is both sound and complete for the Propositional Calculus, and since you can guarantee to reach an answer in finitely-many steps it is also a decision procedure.

Natural Deduction as a Proof System

Natural Deduction can be used for both the Propositional Calculus and the Predicate Calculus.

In either logic, it is possible to derive the conclusion of an inference from the premisses using Natural Deduction if and only if the inference is valid. This means that Natural Deduction is both sound and complete for both logics.

However, there is no general method for either constructing the derivation which must exist if the inference is valid or for showing that no such derivation exists in the case of an invalid inference. Hence Natural Deduction is not a decision procedure.

Sound and Complete implies Compact

Assume that we have a sound and complete proof system for some logic, and let $\Sigma \models C$ be a valid inference in that logic, where Σ may be an infinite set.

By completeness, there is a proof of C in the system using only premisses from Σ.

Since this proof is of finite length, it 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.

By soundness, this means that $\Sigma_0 \models C$.

Hence the logic is compact.

Truth trees are not a decision procedure for predicate calculus

We look for a model for

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

The truth tree never ends:

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

First-order Theories

A first-order theory is a set Θ 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.

For a given domain and a suitable first-order language interpreted over this domain, *the theory of the domain* is the set of all sentences true under this interpretation.

Axiomatisation

A theory Θ is **finitely axiomatisable** if there exists a finite set A of axioms and/or axiom schemas such that the set of logical consequences of A is precisely Θ.

An **axiom** is any formula.

An **axiom schema** is a template into which predicates or formulae can be inserted to generate new axioms.

The logical consequences of an axiom set are called the **theorems** of the first-order theory specified by the set.

The First-order Theory of Identity

The identity relation, denoted "=", is axiomatised by the axiom:

$$\forall x(x = x)$$

and the axiom schema

$$\forall x \forall y(x = y \rightarrow (\Phi[x] \rightarrow \Phi[y])).$$

Some theorems of the theory of identity:

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

The first-order theory of identity is not categorical. To define identity uniquely we need the *second-order* formula

$$\forall x \forall y(x = y \leftrightarrow \forall P(P(x) \leftrightarrow P(y))).$$

First-order logic with identity

We avoid the non-categoricity problem by introducing identity as a logical constant rather than a reinterpretable symbol.

An interpretation satisfies the formula $t_1 = t_2$ if and only if the terms t_1 and t_2 denote the same domain element under the interpretation.

First-order theory of "less than" on the real numbers

This 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)

Some theorems of this theory:

$\forall x, y(x < y \rightarrow \neg(y < x))$
$\forall x, y, z, w((x < y \wedge z < w) \rightarrow (x < w \vee z < y))$

"Less than" is asymmetric

Theorem
$\forall x \forall y(x < y \rightarrow \neg(y < x))$

Proof
Suppose $x < y$.

If also $y < x$ then we have $x < y \wedge y < x$, so by *Transitive* we have $x < x$.

But by *Irreflexive* we have $\neg(x < x)$.

Thus we have $x < x \wedge \neg(x < x)$, which is a contradiction.

Therefore $\neg(y < x)$ (by *Not-intro*).

Hence $x < y \rightarrow \neg(y < x)$ (by *If-intro*).

Hence $\forall x \forall y(x < y \rightarrow \neg(y < x))$ (by *All-intro*).

The Theory of "Less Than" is not Categorical

Both $(\mathbb{R}, <)$ and $(\mathbb{Q}, <)$ have the same first-order theory. But they are not isomorphic structures: so the theory is not categorical.

To distinguish them we need a second-order statement:

S: *There is a non-empty proper subset L of the domain 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.*

This is true of $(\mathbb{Q}, <)$ but not of $(\mathbb{R}, <)$.
(E.g., let $L = \{x \in \mathbb{Q} \mid x^2 < 2\}$.)

In symbols, S is:

$$\begin{array}{l}\exists P(\exists x P(x) \wedge \\ \quad \forall x \forall y(P(x) \wedge \neg P(y) \rightarrow x < y) \wedge \\ \quad \forall x(P(x) \rightarrow \exists y(x < y \wedge P(y))) \wedge \\ \quad \forall x(\neg P(x) \rightarrow \exists y(y < x \wedge \neg P(y))))\end{array}$$

First-Order Arithmetic

The non-logical vocabulary consists of the following symbols:

- A constant symbol "0" (intended interpretation: the natural number zero).
- A one-place function symbol "s" (intended interpretation: the successor function).
- A two-place function symbol "$+$" (intended interpretation: addition).
- A two-place function symbol "$*$" (intended interpretation: multiplication).

The logical vocabulary is the standard first-order predicate calculus with identity.

Peano's Axioms

Zero is not the successor of any number:

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

Distinct numbers have distinct successors:

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

Addition of zero:

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

Addition of a successor:

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

Multiplication by zero:

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

Multiplication by a successor:

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

Induction schema:

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

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

Proof. We use an instance of **Ind**:

(1) $0 + 0 = 0$
$\wedge\ \forall x(0 + x = x \rightarrow 0 + sx = sx)$
$\rightarrow\ \forall x(0 + x = x)$

Base case (first conjunct of antecedent)

(2) $0 + 0 = 0$

follows directly from **A1** by All-elim.

General case (second conjunct):
Assume $0 + a = a$.
From **A2**, by All-elim, $0 + sa = s(0 + a)$.
Substituting a for $0 + a$ gives $0 + sa = sa$.
By If-intro we now have

(3) $0 + a = a \rightarrow 0 + sa = sa$,

and All-intro give us the general case.

The theorem follows from (1), (2) and (3) by And-intro and If-elim.