COMP340-08B Reasoning About Programs

37. Limitations of Predicate Logic

Robi Malik

THE UNIVERSITY OF WAIKATO

DEPARTMENT OF COMPUTER SCIENCE
TARI ROROHIKO

Characterising The Natural Numbers

Goal:

Define a predicate symbol nat using a set of axioms in predicate logic such that the only possible models are such that $\mathrm{nat}(x)$ is true only for the natural numbers (or for some isomorphic structure).

First Attempt

Axioms

1. $\mathrm{nat}(0)$
2. $\forall x\,(\mathrm{nat}(x) \rightarrow \mathrm{nat}(\mathrm{s}(x)))$

Non-standard model

- Domain $\mathbf{D} = \{0\}$
- Interpret 0 as 0
- Interpret s as $\mathrm{s}\colon \mathbf{D} \to \mathbf{D};\ 0 \mapsto 0$
- $\mathrm{nat}(x)$ true for all $x \in \mathbf{D}$

Second Attempt

Axioms

1. $\mathrm{nat}(0)$
2. $\forall x\,(\mathrm{nat}(x) \rightarrow \mathrm{nat}(\mathrm{s}(x)))$
3. $\forall x\,(\mathrm{nat}(x) \rightarrow x \neq \mathrm{s}(x))$

Non-standard model

- Domain $\mathbf{D} = \{0, 1\}$
- Interpret 0 as 0
- Interpret s as $\mathrm{s}\colon \mathbf{D} \to \mathbf{D};\ x \mapsto x + 1 \bmod 2$
- $\mathrm{nat}(x)$ true for all $x \in \mathbf{D}$

Third Attempt

Axioms

1. $\mathrm{nat}(0)$
2. $\forall x\,(\mathrm{nat}(x) \rightarrow \mathrm{nat}(\mathrm{s}(x)))$
3. $\forall x\,(\mathrm{nat}(x) \rightarrow \mathrm{s}(x) \neq 0)$
4. $\forall x\,\forall y\,(\mathrm{nat}(x) \wedge \mathrm{nat}(y) \wedge \mathrm{s}(x) = \mathrm{s}(y) \rightarrow x = y)$

*Successor is an **injective** function.*

Non-standard model

- Domain $\mathbf{D} = \mathbb{N} \cup \{\infty\}$
- Interpret 0 as 0
- Interpret s as $\mathrm{s}\colon \mathbf{D} \to \mathbf{D};\ x \mapsto \begin{cases} x + 1, & \text{if } x \in \mathbb{N}; \\ \infty, & \text{otherwise.} \end{cases}$
- $\mathrm{nat}(x)$ true for all $x \in \mathbf{D}$

A Simpler Characterisation Problem

Given a unary predicate symbol p, construct predicate logic formulas:

$\mathrm{ATLEAST}_{2,\mathrm{p}}$ — which is true if and only if p holds for at least two distinct elements.

$\mathrm{ATLEAST}_{k,\mathrm{p}}$ — which is true if and only if p holds for at least k distinct elements, for a given number k.

$\mathrm{FINITE}_{\mathrm{p}}$ — which is true if and only if p holds for some finite number of distinct elements.

Cardinality Formulas

(ATLEAST$_{2,p}$) $\exists x_1 \exists x_2 (p(x_1) \wedge p(x_2) \wedge x_1 \neq x_2)$

(ATLEAST$_{k,p}$) $\exists x_1 \exists x_2 \ldots \exists x_k$
$(p(x_1) \wedge p(x_2) \ldots \wedge p(x_k) \wedge$
$x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge \ldots \wedge x_{k-1} \neq x_k)$

(ATMOST$_{k,p}$) $\exists x_1 \exists x_2 \ldots \exists x_k$
$(p(x_1) \wedge p(x_2) \ldots \wedge p(x_k) \wedge$
$\forall y\, (p(y) \rightarrow y = x_1 \vee \ldots \vee y = x_k)))$

(EXACTLY$_{k,p}$) ATLEAST$_{k,p}$ $\wedge$ ATMOST$_{k,p}$

(FINITE$_p$) ???

We cannot Characterise the Integers!

If we can characterise the natural numbers, we can also characterise finiteness:

New function symbol f_p represents an injective map from the domain of p into the set $\{0,\ldots, n-1\}$.

(FINITE$_p$) $\exists n\, (nat(n) \wedge$
$\forall x\, (p(x) \rightarrow nat(f_p(x)) \wedge f_p(x) < n) \wedge$
$\forall x_1 \forall x_2\, (f_p(x_1) = f_p(x_2) \rightarrow x_1 = x_2))$

Since this is not possible, we cannot characterise the natural numbers.

The Compactness Theorem

Let Φ be an **unsatisfiable** set of predicate logic formulas.
Then there exists a **finite subset**
$\Phi' \subseteq \Phi$
such that Φ' is also unsatisfiable.

Note: This result holds even if Φ contains an infinite number of formulas.

First Order Logic

- Predicate logic as we have used so far is also known as **First Order Logic**.
- It is limited in its expressive power:
  - *The natural numbers cannot be characterised accurately.*
- To overcome these problems, we have to extend the logic language, resulting in ... **Second Order Logic**.

We cannot Characterise Finiteness!

Proof.

Assume there exists a predicate logic formula FINITE$_p$ that is true if and only if p holds for some finite number of distinct elements.

Then consider the following set of formulas

$\Phi = \{$FINITE$_p$, ATLEAST$_{1,p}$, ATLEAST$_{2,p}$, $\ldots\}$

Clearly Φ is unsatisfiable, but every finite subset of Φ is satisfiable.

This contradicts compactness, therefore the formula FINITE$_p$ cannot exist. □

Second Order Logic

- Second Order Logic introduces the ability to quantify over predicate and function symbols.

$$\exists \boldsymbol{f}\ (\forall x\, (p(x) \rightarrow q(\boldsymbol{f}(x))) \wedge \forall x_1 \forall x_2\, (\boldsymbol{f}(x_1) = \boldsymbol{f}(x_2) \rightarrow x_1 = x_2))$$

- This removes the restrictions of First Order Logic and makes it possible to characterise the natural numbers.

The Peano* Axioms

1. nat(0)
2. $\forall x\ (\text{nat}(x) \to \text{nat}(\text{s}(x)))$
3. $\forall x\ (\text{nat}(x) \to \text{s}(x) \neq 0)$
4. $\forall x\ \forall y\ (\text{nat}(x) \wedge \text{nat}(y) \wedge \text{s}(x) = \text{s}(y) \to x = y)$
5. $\forall P\ (P(0) \wedge \forall x\ (P(x) \to P(\text{s}(x)) \to \forall x\ (\text{nat}(x) \to P(x)))$

** Guiseppe Peano (1858-1932)*

The Axiom of Induction

The fifth Peano Axiom

$$\forall P\ (P(0) \wedge \forall x\ (P(x) \to P(\text{s}(x)) \to \forall x\ (\text{nat}(x) \to P(x)))$$

uses second order logic to fully capture the concept of induction.
It can be replaced by a weaker First Order Logic **axiom scheme**

$$A(0) \wedge \forall x\ (A(x) \to A(\text{s}(x)) \to \forall x\ (\text{nat}(x) \to A(x))$$

but this only covers those predicates that can be "explicitly named" in the logic.

About Second Order Logic

- In second order logic we can construct all the formulas in the set
  $\Phi = \{\text{FINITE}_p, \text{ATLEAST}_{1,p}, \text{ATLEAST}_{2,p}, \ldots\}$
- Φ is unsatisfiable, but every finite subset of Φ is satisfiable. Therefore second order logic does not have the compactness property.
- There are unsatisfiable sets of formulas that cannot be proven to be unsatisfiable using only a finite number of formulas.
- There can be no sound and complete deductive system for this logic.

Gödel's* Incompleteness Theorem

Let **D** be a sound deductive system for second order logic. Then there exists a valid formula of second order logic that is not provable in **D**.

**Kurt Gödel (1906-1978)*

Conclusions

Logic	Expressive Power	Decidability Status
Propositional	Only finite structures	Decidable / recursive (NP)
First Order	Some infinite structures	Semi-decidable / recursively enumerable
Second Order	All properties of infinite structures	Not semi-decidable / not recursively enumerable

Reading

Huth & Ryan:

Section 2.5
Undecidability of Predicate Logic

Section 2.6
Expressiveness of Predicate Logic

pp. 131–141