Logic: What next?

Raffaella Bernardi and Rosella Gennari

bernardi@inf.unibz.it
P.zza Domenicani 3

Faculty of Computer Science, Free University of Bolzano-Bozen

http://www.inf.unibz.it/~bernardi/Courses/Logic06

Contents

1. Why Logic for Computer Science?

Problem: let f be a function on $\mathbb{Z}$. You know that there exists a zero of f. Write a concurrent program $ZERO$ that finds such a zero.

You can solve the problem by writing two programs, S_1 computing the positive zeros of f, and S_2 computing the non-positive zeros of f:

```
found := false; x := 0;
while ¬found do
   x := x + 1;
   found := f(x) = 0;
od
```

```
found := false; y := 1;
while ¬found do
   y := y + 1;
   found := f(y) = 0;
od
```

Now take as $ZERO$ the parallel composition of S_1 and S_2: $[S_1 \mid S_2]$. This terminates when *both* S_1 and S_2 terminate.

So, the program $ZERO$ seems to solve the problem...

But what happens if f has only one zero, a positive one?

Taken from Verification of Sequential and Concurrent Programs, Apt, K.R., and Olderog, E., 1997.

1.1. Logic for Programmers

- to verify the correctness of operational descriptions,
- to check whether algorithms terminate, are sound or complete and the meaning of this,
- to provide rigorous support for techniques usable by programmers.

1.2. Constraint Logic Programming (CLP)

Consider the first-order formula

$$P(f(a)).$$

What does it *mean*? Well, it has several meanings depending on the chosen interpretation (as you all know by now).

A *constraint program* is made of 'first-order formulas' for which you can specify the intended interpretation. E.g.,

$$\begin{aligned} &1.odd(1). \\ &2.odd(X+2) \leftarrow odd(X). \end{aligned}$$

has its intended semantics in the interpretation $\langle \mathbb{N}; +; 1, 2 \rangle$ where $+$, 1 and 2 have their standard interpretations over $\mathbb{N}$. If you take $\mathbb{R}$ as domain then the interpretation and evaluation of the formula are different.

This feature of CLP helps you in writing correct declarative programs. E.g., possible project/thesis: work within LODE

http://www.inf.unibz.it/~gennari/index.php?page=lode

2. Only PL and FOL?

We have looked at PL and FOL. But there are many. Which one to choose depends on what one has to do with it, what needs to be modeled. Interesting issues are:

Expressivity Can we express all we need to say?

Decidability Given a logic $\mathcal{L}$, are its reasoning problems **decidable**?

- The problem of deciding whether a formula φ is logically implied by a theory Γ is undecidable in full FOL.
- Logical implication is decidable if we restrict to FOL using only at most *two* variable names; such language is called $\mathcal{L}_2$.

Recall: The property of (un)decidability is a general property of the problem and not of a particular algorithm solving it.

2.1. Modal Logic like Temporal Logic

Modal logics are used in computer science, e.g., for knowledge representation and in formal methods.

Think of *temporal modal logics* which are used to specify properties of reactive systems such as concurrent programs, network protocols or embedded systems.

> The most important reason for the successful applications of these logics is that they provide a good balance between expressive power and computational complexity (Grädel, 1999).

In the course of Nonclassical Logic we see modal logic(s) through these glasses. For instance, this is not the case of (full) first-order logic.

3. Logic and other disciplines

3.1. Logic and Linguistics

Syntax "Parsing as deduction"

Semantics FOL formulas as meaning representations of sentences. Advantages:

1. Logics have a precise semantics in terms of *models* —so if we can translate/represent a natural language sentence S into a logical formula ϕ, then we have a precise grasp on at least part of the meaning of S.
2. Important *inference problems* have been studied for the best known logics, and often good *computational implementations* exists. So translating into a logic gives us a handle on inference.

3.2. Logic and DB, KR, Ontologies, Semantic Web

Logic gives a way to formally and precisely represent background knowledge (upper ontology: $\texttt{Human beings} \subset \texttt{Animate}$) and domain knowledge (domain ontology: $\forall x.\texttt{Teach}(\texttt{x},\texttt{y}) \wedge \texttt{Course}(y) \rightarrow \texttt{Teacher}(x)$), as well as data (facts: $\texttt{Student}(\texttt{r})$).

Therefore, it gives a way to querying DB via ontologies:

$$\mathcal{KB} \cup DB \models \phi$$

Similarly, it gives ways of adding semantic information into the web (*Semantic Web*) and hence, making intelligent searching within web pages.

4. Related Courses at FUB

- European Masters in Computational Logics
- Streams: “Information and Knowledge Management” (IKM).
- Courses:
  - Theory of Computing
  - Computational Logic
  - Programming Languages
  - Artificial Intelligence
  - Knowledge Representation
  - Seminars in Knowledge Representation
  - Knowledge Base and Database
  - Formal Methods
  - Nonclassical Logics
  - Computational Linguistics

So, you have many ways for enjoying Logic! :)

but for now . . . good luck for the exam.