Logics for Computation

Carlos Areces and Patrick Blackburn

{carlos.areces,patrick.blackburn}@loria.fr

INRIA Nancy Grand Est
Nancy, France

ESSLLI 2008 - Hamburg - Germany

What is Logic, and why should I care?

- Probably all of you have heard about 'Logic' before.
- But what is logic for you? Perhaps it's the science that studies strange symbols like

$$(p \wedge q) \rightarrow (p \vee q)$$

$$\forall x(\mathrm{Human}(x) \rightarrow \mathrm{Mortal}(x))$$

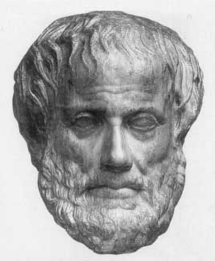
that are (allegedly) important in natural language semantics, computer science, computational linguistics, for (somewhat mysterious) reasons.

- And perhaps you've encountered what logicians called 'theorems', expressions like:

$$p \vee \neg p \quad \text{or} \quad p \rightarrow p.$$

Back to Aristotle

Or perhaps you have met logic in a philosophical setting? You're aware of the work of Aristotle (384 BC - 322 BC), and in particular his discussion of syllogisms.



All vampires are demons.
Angel is a vampire.
Therefore Angel is a demon.

Tomorrow it will rain or it won't...

Either way, logic may not have struck you as particularly exciting or relevant to your work.

- Sentences like "John loves Mary, or not" or "It will rain or it won't, tomorrow" sound a bit silly. They don't seem to be very informative.
- Nor do simple syllogisms seem to have much to with with reasoning in natural language (though, to be fair, they do seem similar to the types of arguments found when reasoning about simple ontologies or when working with WordNet).
- And they certainly seem far removed from the type of arguments found in computer science and mathematics. And the mathematicians notion of 'theorem' seems very different (and much richer) than the logicians notion.

Don't despair!

- This foundational course was designed for people with little or no knowledge of logic, or for those unconvinced that it really can help them in their work.
- It explains a number of topics and technical skills in logic (for example, tableaux, the Davis Putnam Method, the Compactness Theorem).
- But more importantly, it offers a framework form thinking about logic.
- That is, we try to fill in the big picture that is so often missing.

Logic are languages

- We want you to think of logics as languages.
- In particular we want you to think of logics as ways of talking about relational structures or models.
- That is there are two key components in the way we will approach logic
  - The logic: fairly simple, precisely defined, formal languages. (This is where the funny symbols like $\wedge$ and $\exists$ live).
  - The model or relational structure: A simple 'world' (or 'database') that the logic talks about.

Semantic perspective

That is, our perspective on logic is fundamentally semantic. It is due to Alfred Tarski (1902–1983).

The semantic perspective is also known as the model-theoretic perspective, or even the Tarskian perspective.

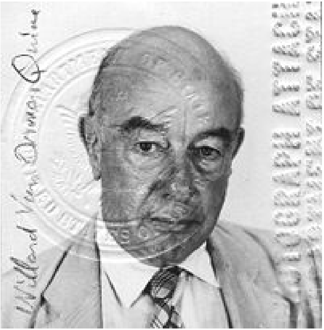
Logic or logics?

The semantic perspective gives us a good way to think about the following question: How many logics are there?
There are (at least) two ways to thing about Logic.

- Option 1, The Monotheistic Approach: Choosing one of all possible logical languages and saying ‘This is THE Logic’, or
- Option 2, The Polytheistic Approach: As a discipline that investigates different logical languages.

Monotheism in the 20th century

Logical monotheism was a powerful force for much of the twentieth century.



Perhaps the most influential monotheist was Willard van Orman Quine, who championed first-order classical logic s the one-true-logic with vigor.

Though (disturbingly for the monotheists) there were always those who worshiped at other temples (such as the intuitionistic logicians and Arthur Prior).

Polytheism in the 21st century

- But polytheism gradually became the dominant thread as time went by.
- Why? Because logic spread everywhere. Computer scientists used it. Early artificial intelligence relied on it. It cropped up in economic and cognitive science. And it became a corner stone of natural language semantics.

Fighting for polytheism

Nowadays logical polytheism is pretty philosophically respectable too. Graham Priest is one of it's most interesting and original proponents.

Polytheism and semantics

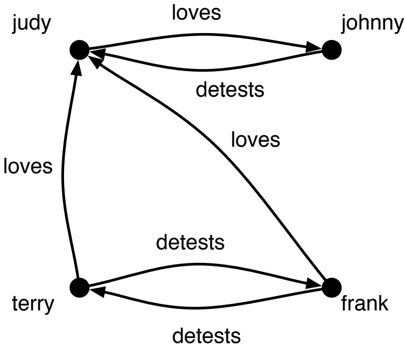
- However the most important point for this course, is that polytheism as regards logic is very natural from a semantic perspective.
- Once we have fixed the model or relational structures we wish to work with (that is, once we have fixed our 'world') it becomes natural to play with different ways of talking about it.
- Indeed today we are going to talk about relational structures without bothering too much about the logics at all.

Relational Structures (informal)

A relational structure (or model) consists of the following

- A non-empty set (often called D, for domain) of the model; think of these as the objects of interest.
- A collection of relations R on the objects in D; think of these as the relations of interest (we shall only work with binary relations (that is, two place relations like "loves", "¡", or "to-the-right-of" in this course) to keep the notation simple.
- A collection of properties on the objects in D; think of these as the properties of interests (perhaps "is red", "is activated", or "is an even number").
- A collection of designated individuals, that is, elements of D that we find really special (maybe "Buffy", "0", or "1")

Our first relational structure



Reminder

A small mathematical reminder:

- Properties are thought of as subsets. That is, given any set D, a property on D is simply a subset P of D; that is $P \subseteq D$.
- Binary relations are though of as sets of ordered pairs. That is, given any set D, a binary relation R is a subset of $D \times D$; that is, $R \subseteq D \times D$.

Relational Structures (more formally)

A relational structure (or model) is a tuple of the form:

$$\langle D, \{R_m\}, \{P_n\}, \{C_l\} \rangle$$

Relational Structures (more formally)

A relational structure (or model) is a tuple of the form:

$$\langle D, \{R_m\}, \{P_n\}, \{C_l\} \rangle$$

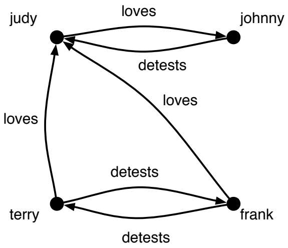
Sometimes we work with simpler forms. For example the following

$$\langle D, R, \emptyset, \emptyset \rangle$$

we would usually write as:

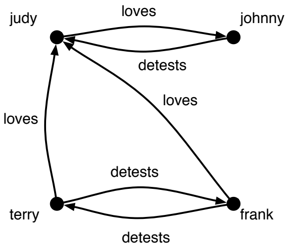
$$\langle D, R \rangle$$

Another look at our first relational structure



$$\langle D, \{L, D\}, \emptyset, C\rangle \quad \text{or} \quad \langle D, \{L, D\}, \emptyset, C\rangle$$

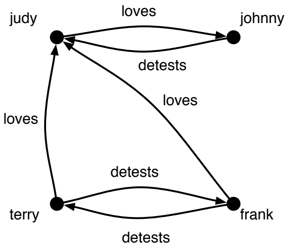
Another look at our first relational structure



$$\langle D, \{L, D\}, \emptyset, C\rangle \quad \text{or} \quad \langle D, \{L, D\}, \emptyset, C\rangle$$

$D = \{judy, johnny, terry, frank\}$

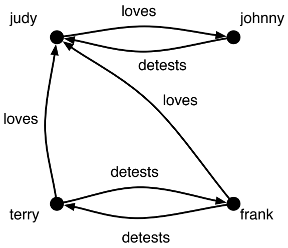
Another look at our first relational structure



$$\langle D, \{L, D\}, \emptyset, C\rangle \quad \text{or} \quad \langle D, \{L, D\}, \emptyset, C\rangle$$

$D = \{judy, johnny, terry, frank\}$
$L = \{(judy, johnny), (terry, judy), (frank, judy), \}$

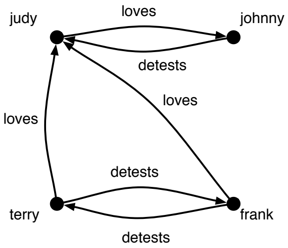
Another look at our first relational structure



$$\langle D, \{L, D\}, \emptyset, C\rangle \quad \text{or} \quad \langle D, \{L, D\}, \emptyset, C\rangle$$

$$D = \{judy, johnny, terry, frank\}$$
$$L = \{(judy, johnny), (terry, judy), (frank, judy), \}$$
$$D = \{(johnny, judy), (terry, frank), (frank, terry), \}$$

Another look at our first relational structure



$$\langle D, \{L, D\}, \emptyset, C\rangle \quad \text{or} \quad \langle D, \{L, D\}, \emptyset, C\rangle$$

$D = \{judy, johnny, terry, frank\}$
$L = \{(judy, johnny), (terry, judy), (frank, judy), \}$

$D = \{(johnny, judy), (terry, frank), (frank, terry), \}$

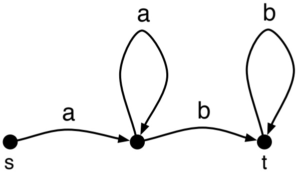
$C = \{judy, johnny, terry, frank\}$

What can be thought of as a relational structure. . . ?

That's the wrong question — the real question is, what can't be thought of as a relational structure?

In fact, it is very hard to think of anything (barring some rather extreme mathematical examples) that can't be viewed as a relational structure.

Example



This shows a finite state automaton for the formal language $a^n b^m$ $(n, m > 0)$, that is, for the set of all strings consisting of a non-empty block of *a*s followed by a non-empty block of *b*s.

A general and important modelling tool

- All common mathematical structures can be though of as relational structures. For a start, functions are simply special kinds of relations.
- Moreover, groups, rings, field, vector spaces, . . . can all be viewed as relational structures.
- Thus Tarski's idea had substantial mathematical impact, and this was decisive in establishing model-theory as an academic discipline.
- Now it is the turn of other disciplines — and in particular, those disciplines related to the LLI theme of this summer school — to draw on the Tarskian insights.

The three big themes

In this course we use the model-theoretic perspective to provide a window on the following three issues:

- Inference: roughly speaking, what methods are there for gaining new information by working with this logic?
- Expressivity: roughly speaking, what can I describe (and what can't I describe) using this logic?
- Computation: too many to list — can computers help with this logic? If so how, and how much?

The three big themes

In this course we use the model-theoretic perspective to provide a window on the following three issues:

- Inference: roughly speaking, what methods are there for gaining new information by working with this logic?
- Expressivity: roughly speaking, what can I describe (and what can't I describe) using this logic?
- Computation: too many to list — can computers help with this logic? If so how, and how much?
- Interestingness: Hard to quantify — but perhaps equally fundamental.

Inference tasks

The semantic perspective give us a good way to think about inference tasks. We shall concentrate on the following three:

- Given a certain logical description ϕ, and a model $\mathcal{M}$, does the formula correctly describe (some aspect of) the model? More simply: is the formulas true (or satisfied) in the model? This task is called the model checking task.
- Given a certain logical description ϕ, does there exist a model where that formula is true? This task is called the satisfiability checking task (or the model building task).
- Given a logical description φ, is it true (or satisfied) in all models? This task is called the validity checking task.

The Model Checking task

- As we shall learn, this is the simplest of the three tasks.
- However, it has also proved to be one of the most useful.
- A classic application is hardware verification. The model $\mathcal{M}$ is a mathematical picture of (say) a chip. The logical description ϕ describes some desirable feature of of the chip. If the $\mathcal{M}$ makes ϕ true, then the chip will have that property.
- Incidentally, this example already suggests the need for "designing logics for their application". After all, there is not reason to think that an off-the-shelf logic will provide exactly what is needed to talk usefully about chips and their properties.

Satisfiability checking (or model building)

- A nice way to think of this problem is in terms of constraints. we have some description. Is there anything that matches this description? That is, does a model making this description actually exist, and can we build it?
- Very useful. The description might be almost anything: for example, a description of a parse tree (if you're doing computational linguistics).

Validity checking

- A great deal of attention has been devoted to this task — essentially, when people talk about "writing a theorem prover", they are talking about creating a computational tool for solving this task.
- The real question is: why? After all, we've already mentions that $p \vee \neg p$ and $p \rightarrow p$ are not going to set too many pulses racing...

Logic is a tool for working with theories

- Let's turn to mathematics. The intuitive idea is that we write down a set Σ of all our axioms. These are the properties that we assume are fundamental and indisputable; what we take for granted. Σ is our theory.
- For example Peano axioms are a theory for the natural numbers.
- Checking if the Goldbach theory is true in the natural numbers boils down to verifying that

$$(\bigwedge \mathsf{PEANO}) \rightarrow \mathsf{GOLDBACH}$$

  is a 'trivial" formula, that is, a validity.

Axiomatics is an ancient idea

The idea goes back to Euclid's celebrated book "The Elements". This is rightly considered one of the foundational blocks of mathematics. It is certainly that, but it is also one of the foundations of modern logic.

Expressivity

- The theme of expressivity is fundamental to this course — and to a model-theoretically inclined logician, the theme is absolutely fundamental — though this early in the course is difficult to say very much about it.
- But the fundamental point is this. Once we have said which relational structures we are interested in, there are many logics suitable for talking about them. Each offers a different (often a fascinatingly different) perspective on the same "world".
- Linguists may like to recall the Sapir-Whorf hypothesis: loosely speaking, the limits of our language are the limits of the world. This analogy should not be taken too literally, but it may be suggestive.

Computation

However, we can already say quite a bit about computation and how it enters the course. In fact it does so at a number of levels.

- First, ideas from theoretical computer science (such as computational complexity) are fundamental tools for analyzing logics.
- Second, more and more computer science is setting the agenda in logic.
- Third, at a practical level we simply need computers when working with logic.

Let's consider these points in turn...

How easy is it? Is it even possible?

- They say that there are some things that cannot be bought for all the money in the world. (True Love?).
- There are problems that cannot be algorithmically solved even with unlimited computing resources.
- The Halting Problem: Given a program P, decide whether P ends or not.
- Some logics are algorithmically unsolvable in this sense (or to be more precise, the inference problems they give rise to are algorithmically unsolvable).
- Even when an inference problem can be algorithmically solvable, the question arises: how hard is it?

Computational Logics: Logic in Action!

- Logic was born as part of philosophy, and achieved greatness as a branch of mathematics.
  - Originally meant to model human reasoning processes
  - and to help making correct inferences.
  - Mathematicians then turned it into a new tool for mathematics.
- With the advent of computer science, things changed
  - Logic played a fundamental part in the development of computers (logic circuits)
  - but nowadays computer science fuels logic.
- In this course a computational view on logical systems will never be far away.

Why do we Need Computers?

- Why do we need computers?
  - well, after all, if we are lazy and don't want to do the work, it would be nice if somebody else could do it for us!
  - even if we could overcome our laziness, we wouldn't be able to do the task ourselves.
- Some of the inference tasks we want to tackle are simply too difficult to perform without the help of computers
  - sometimes billions of possibilities need to be checked to verify that a system satisfies a certain property we want to enforce
  - and even using computers we need to be clever, or all the time till the end of the universe won't be enough. that is, computational logic is not (just) about clever engineering.

Where to from here?

- There are nine more lectures — we're viewing the first and second parts as separate lectures (so we want to keep the break in the middle short).
- Mixture of technical and conceptual — with a little bit of history mixed in.
- But above all, we want to tell a story, and we want you to listen (hence we're only putting the slides online after the course).
- And the story, of course, is the story of logic (or at least, our version of that story).

What we Covered Today

- Logic as language: the model theoretic (or semantic, or Tarskian) perspective
- Relational structures: the heart of the semantic approach.
- The themes of inference, expressivity, and computation.
- Three inference tasks: the model checking task, the satisfiability checking (or model building task), and the validity checking task.
- Briefly mentioned expressivity and raised a number of issues concerning computation.

Relevant Bibliography

There are many good introductions to logic out there. Two interesting ones, written from radically different perspectives are:

- *Philosophy of Logic* by Willard Van Orman Quine, Harvard University Press; New edition, 1980). Still in stock at amazon.
- *An Introduction to Non-Classical Logic*, by Graham Priest, Cambridge University Press; 2nd edition, 2008.

The first is a monotheistic bible. The second raises polytheism to levels worthy of Terry Pratchett's novel "Small Gods".

Relevant Bibliography

- *An introduction to good old fashioned model theory*, by Harold Simmons, http://www.cs.man.ac.uk/~hsimmons/BOOKS/books.html
- *The Logic of Time: A Model-Theoretic Investigation into the Varieties of Temporal Ontology and Temporal Discourse*, by Johan van Benthem, Synthese Library, 2nd revised edition, 1991.
- The life of “the greatest sane logician” and inventor of Model Theory. *Alfred Tarski: Life and Logic*, by Anita Burdman Feferman and Solomon Feferman, Cambridge University Press, 1 paperback edition, 2008

Relevant Bibliography

Finally, earlier I warned you not to take the Sapir-Whorf hypotheses too seriously. Unfortunately, many people have done this, with appalling results. For a brief, elegant, end very very funny critique of such "work", read the following article:

- The great Eskimo vocabulary hoax, by Geoffrey Pullum, in his "Topic ... Comment" Column, *Natural Language and Linguistic Theory*, 7, 275–281, 1989.