Logics for Computation

Lecture #5: About Trees, and How to Cut Them

Carlos Areces and Patrick Blackburn

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

INRIA Nancy Grand Est
Nancy, France

ESSLLI 2008 - Hamburg - Germany

The Story so Far

- We have introduced the $\langle R\rangle$ operator to talk about complex relational structures.
  Nothing fancy (yet), just a simple extension of PL.
- We have used it to describe some properties over models.

  E.g., the following formulas are valid: $\langle R\rangle(p \wedge q) \rightarrow \langle R\rangle p$
  $\langle R\rangle\top \wedge \neg\langle R\rangle\neg p \rightarrow \langle R\rangle\neg p$
- We have discussed when two models are the same.
- We have seen an algorithm to check whether a formula is true in a given model.

What do we do Today

- We will define a tableax algorithm for satisfiability of formulas containing $\langle R\rangle$.
  - We already know how to check, given a model, if the formula holds in the model (model checking).
  - Today, we will see how do we check whether a formula has a model.
- We will also go back to the question When are two models the same?
  - and talk about trees ...
  - ... and how to cut them.

Counting models

- The proof that the satisfiability problem for PL is decidable is very simple:
  - Suppose that you are given a formula φ and you are looking for a model of φ.
  - First note that propositional symbols that do not appear in φ are irrelevant.
  - We know that our models has only one point.
  - Hence, we only need to list all possible ways of labelling that single node with propositional symbols in φ.
- What about the $\langle R\rangle$ language?

The Tableaux Method for Relational Structures

- We want to devise a tableau method for the language we introduced to talk about complex relational structures.
- Let's review the tableaux method that we introduced for propositional logic:

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}\ (\wedge)$$

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad s{:}\neg\psi}\ (\neg\wedge)$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}\ (\neg\neg)$$

- Pretty neat: 3 rules for an NP-complete problem!
- But now we want to deal with more than a single point.
- The solution is: labels!
- They will help us keep track of what is going on in each point in our model.

Now Lines!

- We have dealt in the previous slide with multiple points. What about lines?
- Remember that the operator we introduced to talk about lines in our language was $\langle R\rangle\varphi$ and we said that

  $\mathcal{M}, w \models \langle R\rangle\varphi$ iff there is w' s.t. wRw' and $\mathcal{M}, w' \models \varphi$.

- Start with the labelled formula $s{:}\langle R\rangle\varphi$. If this formula is satisfiable, it is because there is an R-sucessor t where φ holds.

$$\frac{s{:}\langle R\rangle\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}\ (\langle R\rangle)$$

for t a new label

- Start with the labelled formula $s{:}\neg\langle R\rangle\varphi$. If there is an R-successor t, then φ should not hold at t.

$$\frac{\begin{array}{c} s{:}\neg\langle R\rangle\varphi \\ sRt \end{array}}{t{:}\neg\varphi}\ (\neg\langle R\rangle)$$

The Complete Cast, plus an Example

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\langle R\rangle\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\langle R\rangle\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

$$\begin{array}{c} s{:}(\neg\langle R\rangle p \wedge (\langle R\rangle q \wedge \langle R\rangle p)) \\ s{:}\neg\langle R\rangle p \\ s{:}(\langle R\rangle q \wedge \langle R\rangle p) \\ s{:}\langle R\rangle q \\ s{:}\langle R\rangle p \\ sRt \\ t{:}q \\ sRu \\ u{:}p \\ t{:}\neg p \\ u{:}\neg p \end{array}$$

contradiction!!!

- Which are the similarities/differences with tableaux for PL?
- How do we know that we got it right?
- What can we learn from the calculus?

A Closer Look

- Which similarities / differences with tableaux for PL?
  - Does the calculus terminate?
  - What are labels? What are they doing? Can we use them?
  - Is this an algorithm?
  - Is it a good algorithm?

$$\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}} \qquad \frac{s{:}\langle R\rangle\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}}$$

for t a new label

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad s{:}\neg\psi} \qquad \frac{\begin{array}{c} s{:}\neg\langle R\rangle\varphi \\ sRt \end{array}}{t{:}\neg\varphi}$$

$$\frac{s{:}\neg\neg\varphi}{s{:}\varphi}$$

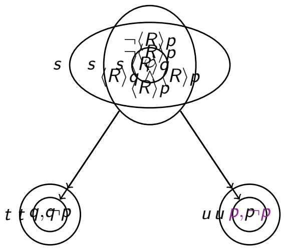
- Did we get it right?
  - Did we get it right in the PL case, to start with?! Consider the rule:

$$\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\quad \begin{array}{c} s{:}\varphi \\ s{:}\neg\psi \end{array}}$$

- What can we learn from the calculus?
  - Something about models!

Tree Models

- Let us see the tableux proof we did before again, for the formula
  $\varphi = \neg\langle R\rangle p \wedge (\langle R\rangle q \wedge \langle R\rangle p)$



$$
\begin{array}{c}
s{:}(\neg\langle R\rangle p \wedge (\langle R\rangle q \wedge \langle R\rangle p))\\
s{:}\neg\langle R\rangle p\\
s{:}(\langle R\rangle q \wedge \langle R\rangle p)\\
s{:}\langle R\rangle q\\
s{:}\langle R\rangle p\\
sRt\\
t{:}q\\
sRu\\
u{:}p\\
t{:}\neg p\\
u{:}\neg p
\end{array}
$$

Tree and Finite Model Properties

- Using the rules of the tableaux calculus we only explore finite, tree models.
- Let's assume that the calculus is correct (you will have to believe me).
- Then the $\langle R\rangle$-language
  - cannot say infinite,
  - cannot say non-tree.

$$
\frac{s{:}(\varphi \wedge \psi)}{\begin{array}{c} s{:}\varphi \\ s{:}\psi \end{array}}
\qquad
\frac{s{:}\langle R\rangle\varphi}{\begin{array}{c} sRt \\ t{:}\varphi \end{array}} \text{ for } t \text{ a new label}
$$

$$
\frac{s{:}\neg(\varphi \wedge \psi)}{s{:}\neg\varphi \quad\mid\quad s{:}\neg\psi}
\qquad
\frac{s{:}\neg\neg\varphi}{s{:}\varphi}
\qquad
\frac{\begin{array}{c} s{:}\neg\langle R\rangle\varphi \\ sRt \end{array}}{t{:}\neg\varphi}
$$

Theorem: A formula in the $\langle R\rangle$-language is satisfialle if and only if it is satisfiable in a finite, tree relational structure.

What we Covered in this Lecture

- We introduce a tableaux method to check satisfiability for the language with $\langle R\rangle$.
- We saw that we can use labels to describe what is going on in each point of a relational structure.
- More importantly: we saw that tableaux are a way to sistematically explore relational structures.
- Actually, from the tableaux algorithm we could learn some model properties: we only need to consider finite tree models.

Relevant Bibliography I

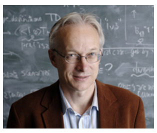
Tableaux Algorithms

- The Tableaux Method is the core algorithm of most current theorem provers for relational languages.
- You might have heard about description logics. Racer, FaCT++, Pellet are all based on a calculus similar to the one we studied today.
- Fitting's Web page: `http://comet.lehman.cuny.edu/fitting/`

Fitting, Melvin (1983). *Proof Methods for Modal and Intuitionistic Logics*. D. Reidel Publishing Co., Dordrecht.

Relevant Bibliography II

Tree and Finite Model Properties



- Unraveling, the procedure to turn arbitrary models into trees, was introduced by Segerberg.
- Segerberg's Web page: `http://www.phil.ucalgary.ca/philosophy/people/segerberg.html`
- A more general result about turning things into other things can be proved using bisimulations.
- van Benthem's Web page: `http://staff.science.uva.nl/~johan/`

Segerberg, Krister (1971). *An Essay in Classical Modal Logic*, Department of Philosophy Uppsala University, Sweden. Uppsala Philosophical Studies.

van Benthem, Johan (1985). *Modal Logic and Classical Logic*, Bibliopolis.

Interesting Links

- Some provers for the $\langle R\rangle$ language based on the tableaux algorithm:
  - Racer `http://www.racer-systems.com/`
  - FaCT++ `http://owl.man.ac.uk/factplusplus/`
  - Pellet `http://pellet.owldl.com/`
  - HTab `http://trac.loria.fr/projects/htab/wiki`
- Some based on other algorithms:
  - MSpass (translation based) `http://www.cs.man.ac.uk/~schmidt/mspass/`
  - HyLoRes (resolution based) `http://trac.loria.fr/projects/hylores`

The Next Lecture

No Way to Say Warm in French