Variable free reasoning on finite trees

Maarten Marx

ILLC, Amsterdam, The Netherlands

M4M, Nancy, September 2003

Introduction

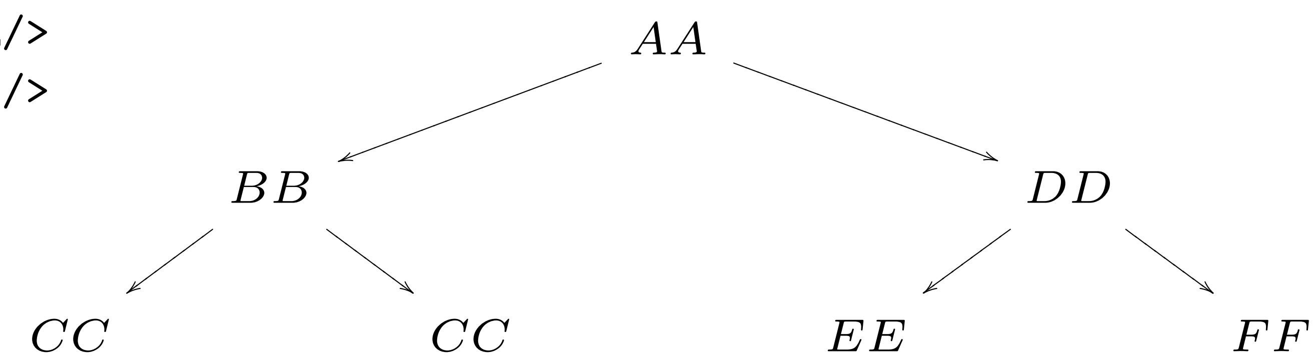
- Modal languages interpreted on finite trees.
- More precisely, node labelled, sibling ordered finite trees.
- Interested in *Completeness* questions:
  - Functional completeness
  - Completeness for definitions (Beth's property)
  - Complete optimal decison algorithms

Motivation

- New data storage format: XML.
- XML documents are finite node labeled ordered trees.
- Most succesful XML query language XPath is variable free, and very modal in flavour.
- XPath query containment can effectively be reduced to satisfiability in a corresponding modal language.
- XPath query evaluation can be reduced to model checking for the corresponding modal language.

XML documents are node labelled ordered trees

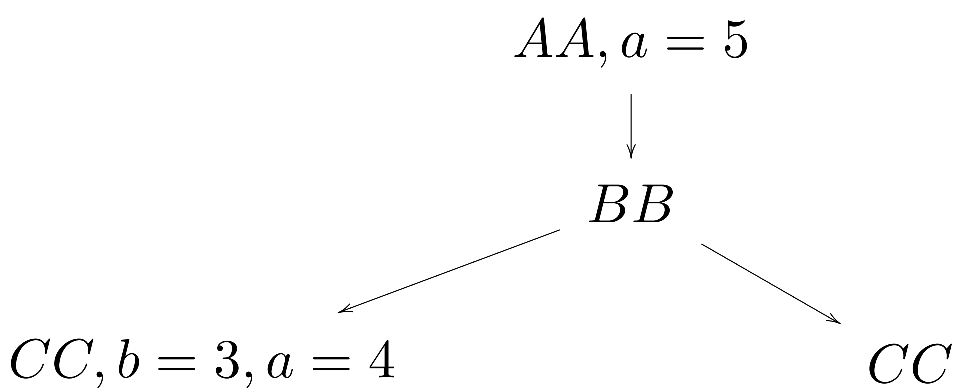
```
<AA>
  <BB>
    <CC/>
    <CC/>
  </BB>
  <DD>
    <EE/>
    <FF/>
  </DD>
</AA>
```



Attributes

XML Attributes can be modelled by multiple labels.

```
<AA a=5>
  <BB>
    <CC b=3,a=4/>
    <CC/>
  </BB>
</AA>
```

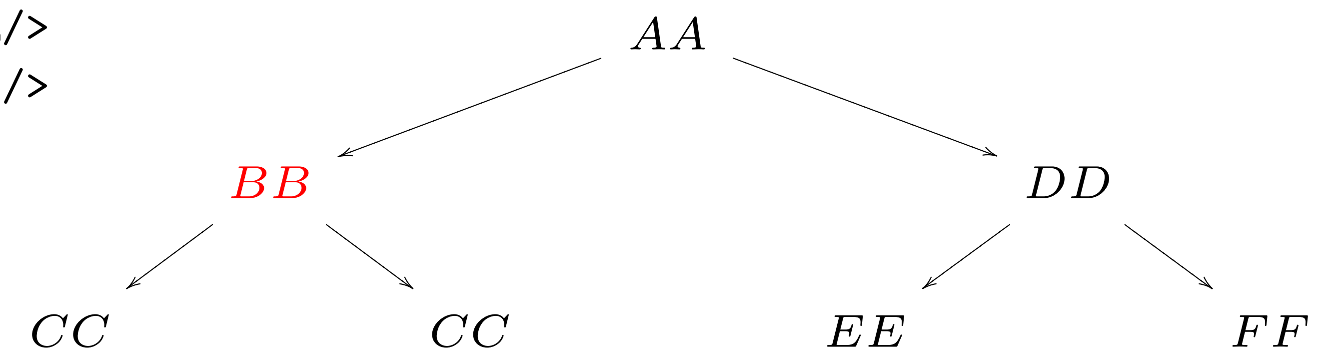


XPath, a query language for XML

- XPath is a W3C standard query language for XML documents.
- Given a XML document D, a node n in D, an XPath query Q selects all nodes from D which are reachable from n by the path described in Q.
- Examples:

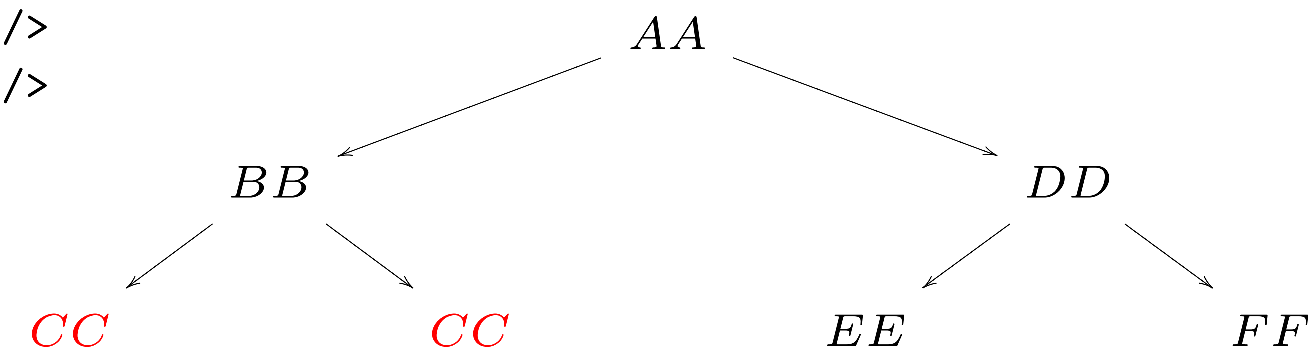
/AA/BB

```
<AA>
  <BB>
    <CC/>
   <CC/>
  </BB>
  <DD>
    <EE/>
    <FF/>
  </DD>
</AA>
```



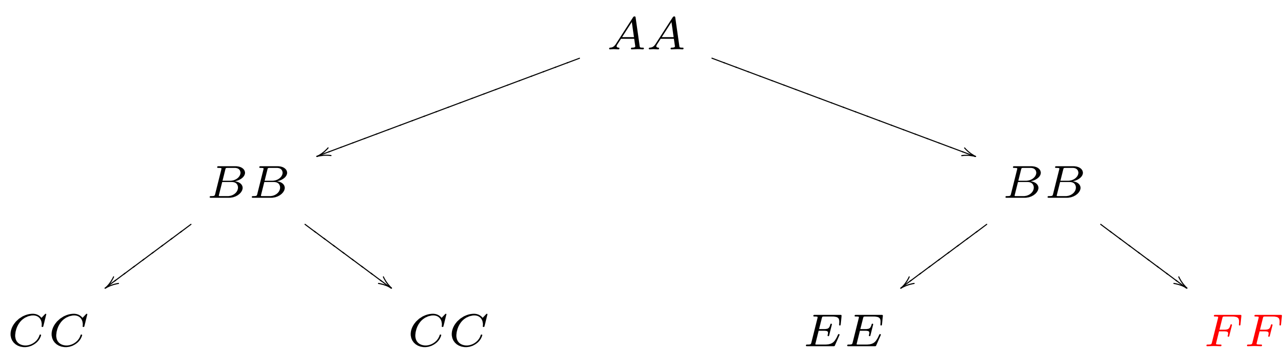
//CC

```
<AA>
  <BB>
   <CC/>
   <CC/>
  </BB>
  <DD>
    <EE/>
    <FF/>
  </DD>
</AA>
```



//BB[EE]/FF

```
<AA>
  <BB>
    <CC/>
    <FF/>
  </BB>
  <BB>
   <EE/>
   <FF/>
  </BB>
</AA>
```



Finite sibling ordered trees

- First order structures with two binary relations.
- Domain is a finite set.
- Dominance relation:

$$xR_{\downarrow}y \text{ iff } x \text{ is an ancestor of } y.$$

- Linear order on the children of each node

$$xR_{\rightarrow}y \text{ iff } x \text{ and } y \text{ are siblings and } x \text{ is strictly on the left of } y.$$

- First order language in signature $R_{\downarrow}$ and $R_{\rightarrow}$ and unary $P_1, P_2, \ldots$

Modal or variable-free approaches

The language is two-sorted with interactions:

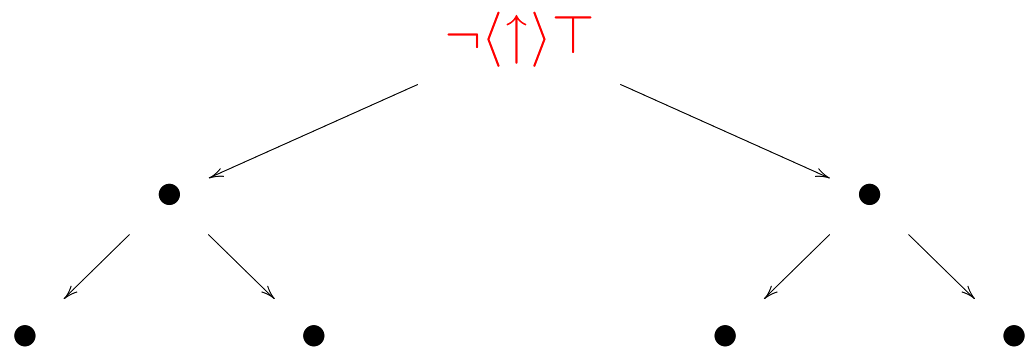
- Sort for paths in the tree: regular expressions over the four basic steps
  - $\downarrow$ (child),
  - $\uparrow$ (parent),
  - $\rightarrow$ (right sibling),
  - $\leftarrow$ (left sibling).
- Sort for nodes in the tree:
  - labels
  - closed under the Boolean operations

Interactions:

- if π is of the path sort, and ϕ of the node sort, then $\langle\pi\rangle\phi$ is of the node sort.
  - $\langle\pi\rangle\phi$ holds at nodes from which there is a π path to a ϕ node.
  - $V(\langle\pi\rangle\phi) = \{t \mid \exists t' : t \; \pi \; t' \wedge t' \in V(\phi)\}$.
- if ϕ is of the node sort, then $?\phi$ is of the path sort.
  - $?\phi$ is called a test.
  - $?\phi$ denotes the identity path from a ϕ node to itself.
  - $?\phi$ denotes $\{(t,t) \mid t \in V(\phi)\}$.

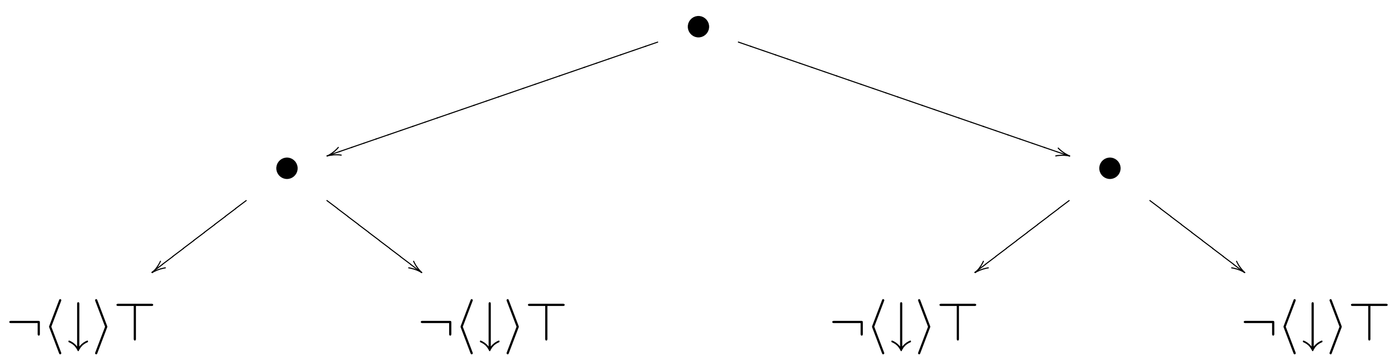
Example expressions

I am the root: $\neg\langle\uparrow\rangle\top$



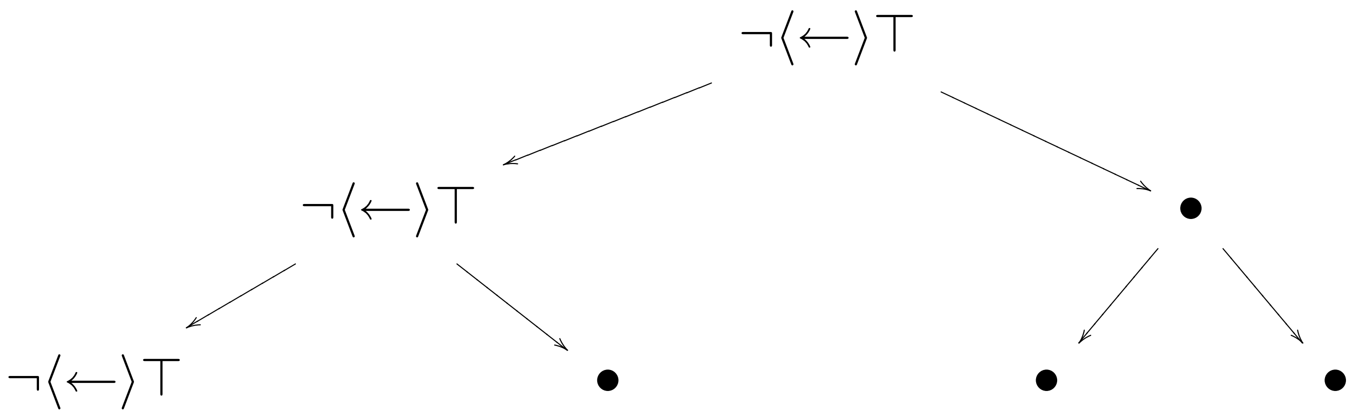
Example expressions

I am a leaf: $\neg\langle\downarrow\rangle\top$



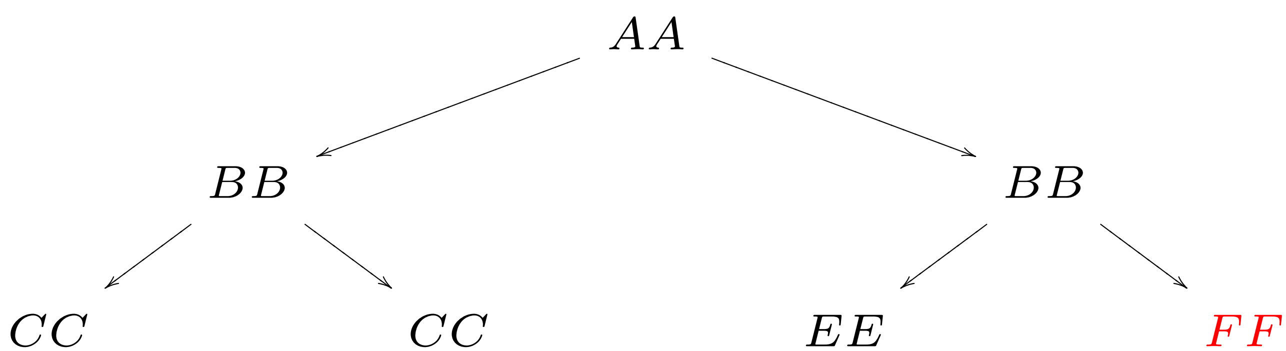
Example expressions

I am a first daugther: $\neg\langle\leftarrow\rangle\top$



//BB[EE]/FF

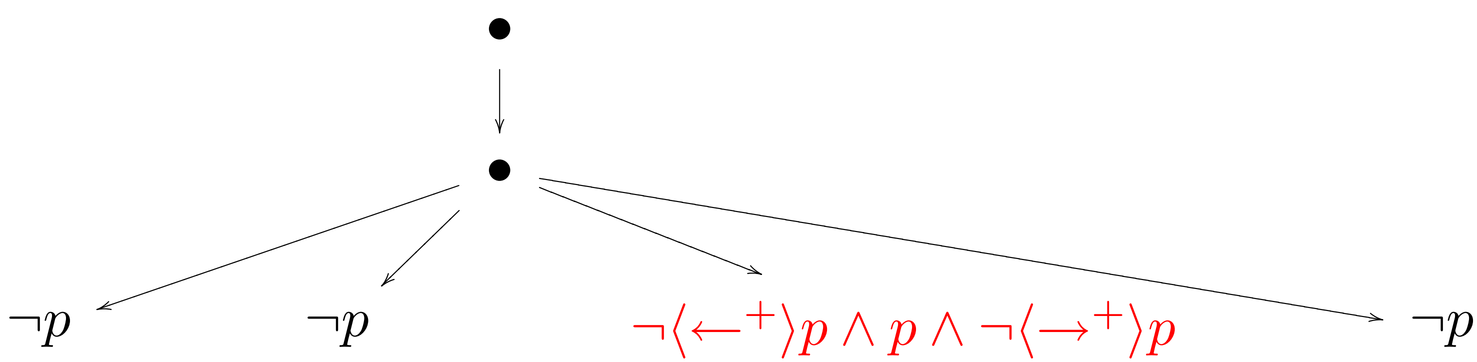
```
<AA>
  <BB>
    <CC/>
    <FF/>
  </BB>
  <BB>
   <EE/>
   <FF/>
  </BB>
</AA>
```



//BB[EE]/FF is equivalent to $FF \wedge \langle\uparrow\rangle(BB \wedge \langle\downarrow\rangle EE \wedge \langle\uparrow^*\rangle root)$.

Example expressions

I am the unique p among my siblings: $p \wedge \neg\langle\leftarrow^{+}\rangle p \wedge \neg\langle\rightarrow^{+}\rangle p$



Three languages

- Full PDL. Here called $\mathcal{X}_{Reg}$. [Kracht 95]

$$\begin{array}{lcl} \pi & ::= & \leftarrow \;|\; \rightarrow \;|\; \uparrow \;|\; \downarrow \;|\; \pi;\pi \;|\; \pi \cup \pi \;|\; \pi^* \;|\; ?\phi \\ \phi & ::= & p \;|\; \top \;|\; \neg\phi \;|\; \phi \wedge \phi \;|\; \langle\pi\rangle\phi. \end{array}$$

- Fragment of $\mathcal{X}_{Reg}$, keeping *conditional paths*. Here called $\mathcal{X}_{cp}$. [Palm 95]

$$\pi ::= \leftarrow \;|\; \rightarrow \;|\; \uparrow \;|\; \downarrow \;|\; ?\phi;\pi \;|\; \pi^*.$$

- Modal language corresponding to Core XPath: $\mathcal{X}_{Core}$.

$$\pi ::= \leftarrow \;|\; \rightarrow \;|\; \uparrow \;|\; \downarrow \;|\; \pi^*.$$

Overview

1. Comparing expressive power. (functional completeness)
2. Completeness for definitions (Beth's property)
3. Model checking
4. Complete decision algorithms

Comparing expressive power

- How are these three languages related?
- How do they relate to other formalisms used on ordered trees?

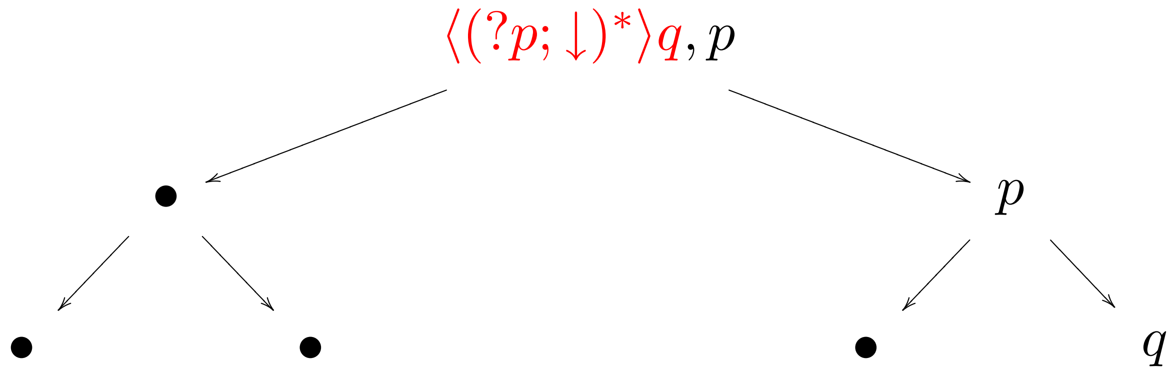
1. $\mathcal{X}_{Core} \subsetneq \mathcal{X}_{cp} \subsetneq \mathcal{X}_{Reg}$.
2. $\mathcal{X}_{cp}$ is equivalent to the language with just four until operators.
3. $\mathcal{X}_{cp}$ is first order logic and $\mathcal{X}_{Reg}$ is stronger.

$\mathcal{X}_{Core}$ and Core XPath

- We study the W3C standard XPath 1.0.
- Gottlob et al singled out the *logical core* of XPath 1.0, and called it Core XPath.
- Theorem Every Core XPath root expression is equivalent to an $\mathcal{X}_{Core}$ expression.
- E.g., `/AA//BB` is equivalent to $BB \wedge \langle\uparrow^*\rangle(AA \wedge root)$.

Adding conditional paths

- A conditional path is an expression of the form $?\phi; \pi$.
- **EXAMPLE** $\langle(?p; \downarrow)^*\rangle q$ is true on all nodes from which there is a path going down along p nodes ending in a q node.



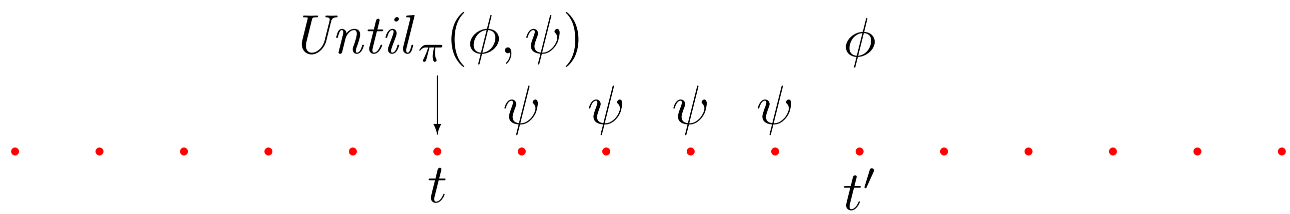
- $\langle(?p; \downarrow)^*\rangle q$ behaves like Until in temporal logic: Until q holds, p is true.

$\mathcal{X}_{cp}$ and until

- $\mathcal{X}_{until}$ is the modal language with four binary modal operators: $Until_\pi(\phi, \psi)$ for $\pi \in \{\leftarrow, \rightarrow, \uparrow, \downarrow\}$,

- $\mathfrak{M}, t \models Until_\pi(\phi, \psi)$ iff

$$\exists t'(t\ \pi^+\ t' \wedge \mathfrak{M}, t' \models \phi \wedge \forall t''(t\ \pi^+ t''\ \pi^+\ t' \rightarrow \mathfrak{M}, t'' \models \psi)).$$



- Theorem $\mathcal{X}_{cp}$ and $\mathcal{X}_{until}$ are equally expressive.

$\mathcal{X}_{cp}$ and first order logic

- By the meaning definition of $Until_{\pi}(\phi, \psi)$, $\mathcal{X}_{until}$ is contained in the first order logic of trees.

- By Kamp's Theorem, on linear trees, the reverse also holds.

- Gabbay strenghtened Kamp's theorem into the separation theorem: every $\mathcal{X}_{until}$ formula interpreted in linear trees is equivalent to a boolean combination of past, future and present formulas.

- We generalized Gabbay's result to all ordered trees.

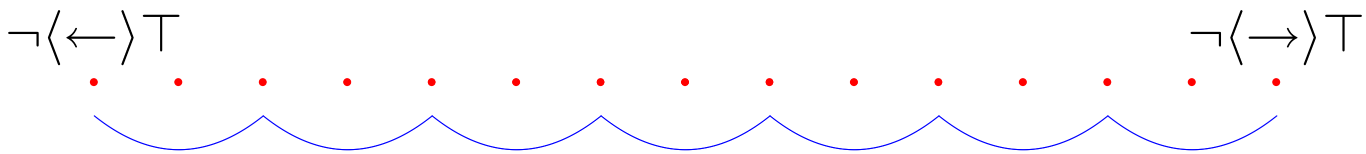
- Thus $\mathcal{X}_{cp}$, $\mathcal{X}_{until}$ and first order logic are equally expressive.

Functional Completeness

Theorem Every first order definable set of nodes in an ordered tree is definable by an $\mathcal{X}_{cp}$ (or equivalently, by an $\mathcal{X}_{until}$) formula.

But $\mathcal{X}_{Reg}$ is more expressive. It can express second order properties of nodes like having an odd number of daughters:

$$\langle\downarrow\rangle(\neg\langle\leftarrow\rangle\top \wedge \langle(\rightarrow;\rightarrow)^*\rangle\neg\langle\rightarrow\rangle\top).$$



Overview

1. Comparing expressive power. (functional completeness)

2. Completeness for definitions (Beth's property)

3. Model checking

4. Complete decision algorithms

Beth's property

- Does not hold for the first order (until) fragment.
- Let Γ be $root \rightarrow p \;\; \wedge \;\; p \rightarrow [\downarrow]\neg p \;\; \wedge \;\; \neg p \rightarrow [\downarrow]p$.
- For $\mathfrak{M}$ any tree, if $\mathfrak{M} \models \Gamma$ then $\mathfrak{M}, n \models p \iff n$ is even.
- But evenness is not first order expressible.
- The $\mathcal{X}_{Reg}$ definition of p is of course $\langle(\uparrow;\uparrow)^*\rangle root$.

Question 1 Can we find for all implicit $\mathcal{X}_{cp}$ definitions, the explicit definition in $\mathcal{X}_{Reg}$? Or better, for interpolants?

Question 2 Does $\mathcal{X}_{Reg}$ have interpolation or Beth's property? (for PDL this is still unknown).

Overview

1. Comparing expressive power. (functional completeness)

2. Completeness for definitions (Beth's property)

3. Model checking

4. Complete decision algorithms

Model checking

- XPath query evaluation can effectively be reduced to $\mathcal{X}_{Reg}$ model checking.

- From [Gottlob et al, PODS 2003] and the equivalence between $\mathcal{X}_{Core}$ and Core XPath it follows that $\mathcal{X}_{Core}$ model checking is hard for **PTIME** (combined complexity).

- The largest language $\mathcal{X}_{Reg}$ can be model checked in linear time, that is, $O(|D| \cdot |Q|)$.

- This follows from the same result for PDL.

Overview

1. Comparing expressive power. (functional completeness)
2. Completeness for definitions (Beth's property)
3. Model checking
4. Complete decision algorithms

Deciding $\mathcal{X}_{Reg}$

- We want to decide the question: given a set Γ of constraints on trees, is ϕ valid on these trees?
- Or $\Gamma \models \phi$, where $\models$ is the *global* consequence relation.
- Motivation
  - Query optimization
  - DTD's XSchema

Deciding $\mathcal{X}_{Reg}$, Complexity

- Decidability is easily obtained from Rabin's theorem, but the complexity is too high.

- Easy lower bound is EXPTIME (by an interpretation of ordinary PDL with one program and only the diamonds $\langle a \rangle$ and $\langle a^* \rangle$.

Decision algorithm by reduction

- Goal: Effectively reduce $\phi \models \psi$ to $\phi' \models \psi'$ in a simpler language.
- In fact to PDL without complex programs at all, but only with the paths $\downarrow$ and $\rightarrow$.
- Then to modal logic of finite binary branching trees.
- Consequence problem for this can easily be shown to be in EXPTIME.
  - reduce it to CTL plus counting, or
  - by bottom up Hintikka Set Elimination.

Reductions

- Idea 1 Delete union, composition and tests by the PDL equivalences:
- $\langle \pi_1 ; \pi_2 \rangle \phi \equiv \langle \pi_1 \rangle \langle \pi_2 \rangle \phi$.
- $\langle \pi_1 \cup \pi_2 \rangle \phi \equiv \langle \pi_1 \rangle \phi \vee \langle \pi_2 \rangle \phi$.
- $\langle ?\psi \rangle \phi \equiv \phi \wedge \psi$.

Reductions (for star)

- Idea 2 "Axiomatize" starred programs:
- for each subformula θ add a new propositional variable q_θ.
- Replace θ throughout by q_θ.
- Ensure that $\theta \equiv q_\theta$ by a constraint.

Reductions for star using constraints

- Example $\langle\downarrow^*\rangle p$ Add as a constraint:

$$q_{\langle\downarrow^*\rangle p} \leftrightarrow p \vee \langle\downarrow\rangle q_{\langle\downarrow^*\rangle p}. \tag{1}$$

- Then $(1) \models q_{\langle\downarrow^*\rangle p} \leftrightarrow \langle\downarrow^*\rangle p$.

- Proof is by induction on the (finite!) number of daughters.

  - t is a leaf:

$$\begin{array}{llr} t \models q_{\langle\downarrow^*\rangle p} & \Longleftrightarrow & \text{(by (1))} \\ t \models p \vee \langle\downarrow\rangle q_{\langle\downarrow^*\rangle p} & \Longleftrightarrow & \text{(as } t \text{ is a leaf)} \\ t \models p & \Longleftrightarrow & \text{(by meaning def and } t \text{ is a leaf)} \\ t \models p \vee \langle\downarrow\rangle\langle\downarrow^*\rangle p & \Longleftrightarrow & \text{(by meaning def)} \\ t \models \langle\downarrow^*\rangle p. & & \end{array}$$

  - t has $k+1$ daughters: By induction hypothesis.

Where this technique breaks

- It is crucial for the last proof that the path under the star "makes a step".

- This is not the case with formulas of the following form
  - $\langle (\downarrow^*)^* \rangle \phi$
  - $\langle (\downarrow ; \uparrow)^* \rangle \phi$

- In fact the reduction does not work in these cases.

Counterexample

- Consider the not satisfiable formula $\langle(\downarrow^*)^*\rangle\bot$.
- Then the axiomatization becomes

$$
\begin{array}{lcl}
q_{\bot} & \leftrightarrow & \bot \\
q_{\langle(\downarrow^*)^*\rangle\bot} & \leftrightarrow & q_{\bot} \vee q_{\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot} \\
q_{\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot} & \leftrightarrow & q_{\langle(\downarrow^*)^*\rangle\bot} \vee q_{\langle\downarrow\rangle\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot} \\
q_{\langle\downarrow\rangle\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot} & \leftrightarrow & \langle\downarrow\rangle q_{\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot}.
\end{array}
$$

- $q_{\langle(\downarrow^*)^*\rangle\bot}$ can be satisfied in the trivial tree with only a root by setting the valuation

$$V(root) = \{q_{\langle\downarrow^*\rangle\langle(\downarrow^*)^*\rangle\bot}, q_{\langle(\downarrow^*)^*\rangle\bot}\}$$

- This model makes the axioms true.
- Thus the reduction does not work for the formula $\langle(\downarrow^*)^*\rangle\bot$.

Reduction for until

- The reduction works for the until language. The constraint is

$$q_{Until_\pi(\phi,\psi)} \leftrightarrow \langle\pi\rangle q_\phi \vee \langle\pi\rangle q_{(\psi\wedge Until_\pi(\phi,\psi))}.$$

- Effective reduction of $\phi \models \psi$ to $q_\phi, \nabla(\phi,\psi) \models q_\psi$, with $\nabla(\phi,\psi)$

- Right hand only contains diamonds of the form $\langle\pi\rangle$ for $\pi \in \{\downarrow,\uparrow,\leftarrow,\rightarrow\}$.

- Reduce it further to $\phi \models \psi$ on binary branching trees for formulas containing only diamonds $\langle\downarrow_0\rangle$ and $\langle\downarrow_1\rangle$.

Modal logic of binary trees

- The consequence problem is different for finite and for infinite trees:
- For instance, $\langle \downarrow_0 \rangle \top \models \bot$ is true on finite trees, but not on all trees.
- For this reason we cannot use known results about deterministic PDL or ordinary modal logic.
- But we can embed the problem into CTL plus counting.

Mimicking finiteness in CTL

- Relativize all diamonds by a new variable d.
- Add as constraints:
  - d holds at the root.
  - for every path there is a point in the future where d is false.
  - Everywhere, if d is false it remains false forever.
- Then $\mathfrak{M}$ relativized to $[d]_{\mathfrak{M}}$ is a finite tree.

Compare with first order logic on trees

- Recall that the until language and the first order language are equally expressive on ordered trees.
- The until language is decidable in EXPTIME.
- The optimal decision procedure for the first order language is non-elementary!

Conclusions and further research

- Variable free tree formalisms are to be preferred over first order formalisms:
  - formulas are easy and intuitive
  - the computational complexity is much lower while having the same expressive power.

- Current work focuses on rewriting $\mathcal{X}_{Reg}$ formulas into a normal form which can be reduced.

- We conjecture that the EXPTIME result holds for the full orientation logic.

- Further research is needed on Beth's definability property.

- Also desirable: a natural extension of first order logic which is as expressive as $\mathcal{X}_{Reg}$.