A Sequent Calculus for a Modal Logic on Finite Data Trees

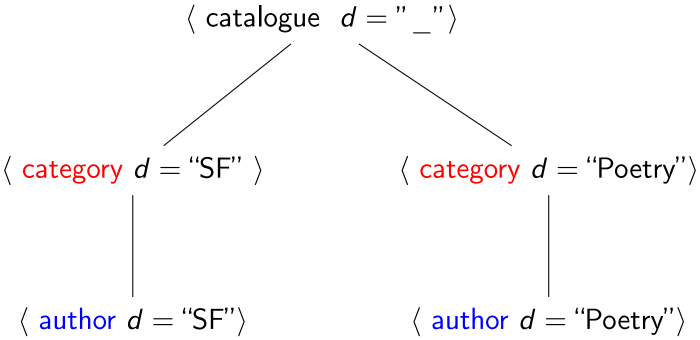
David Baelde, Simon Lunel and Sylvain Schmitz

LSV - ENS Cachan
Inria Rennes, Mitsubishi Electric R&D Centre Europe

CSL 2016, 1er septembre 2016

XML and XPath

- An XML tree :



- A request : $\Box($ category $\supset \Diamond_{=}$ author $)$

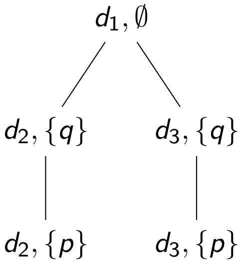
Validity problem for modal data logic

- Modal data logic : addressed by model theory and combinatorial algorithms (e.g. Figueira, 2010)
- Want to study it with proof theory
  - $\rightarrow$ witness for validity of a formula (proof tree) and a counter-example otherwise
  - $\rightarrow$ amenable to reasonable implementations avoiding worst-case complexity
  - $\rightarrow$ same complexity as combinatorial algorithms

Data tree

A *data tree* $\mathcal{A} = (A, \delta, \mathcal{V})$ is a tree $A = (T, <)$ where :

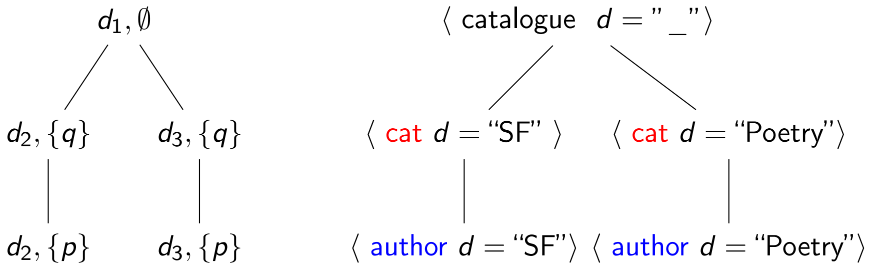
- T the set of nodes of A,
- $<$ the usual descendant relation on trees,
- $\delta\colon T \longrightarrow \mathbb{D}$ a function which associates a datum from a countable set $\mathbb{D}$ to a node of A,
- $\mathcal{V}\colon T \longrightarrow \mathcal{P}(\mathbb{A})$ a valuation which associates a set of atomic propositions to every node of A.



Data tree

A *data tree* $\mathcal{A} = (A, \delta, \mathcal{V})$ is a tree $A = (T, <)$ where :

- T the set of nodes of A,
- $<$ the usual descendant relation on trees,
- $\delta\colon T \longrightarrow \mathbb{D}$ a function which associates a datum from a countable set $\mathbb{D}$ to a node of A,
- $\mathcal{V}\colon T \longrightarrow \mathcal{P}(\mathbb{A})$ a valuation which associates a set of atomic propositions to every node of A.



Modal Data Logic (DataGL)

Syntax

$\phi ::= a \mid \bot \mid \phi \supset \phi \mid \Box_{=}\phi \mid \Box_{\neq}\phi$,
where $a \in \mathbb{A}$ the set of atomic propositions.

Modal Data Logic (DataGL)

Syntax

$\phi ::= a \mid \bot \mid \phi \supset \phi \mid \Box_{=}\phi \mid \Box_{\neq}\phi$,
where $a \in \mathbb{A}$ the set of atomic propositions.

Semantic

Let $\mathcal{A} = (A, <, \delta, \mathcal{V})$ be a *finite data tree*.

- $t \models a$ iff $a \in \mathcal{V}(w)$.
- $t \not\models \bot$
- $t \models \phi \supset \psi$ iff if $t \models \phi$, then $t \models \psi$.
- $t \models \Box_{=}\phi$ iff $\forall t_1 < t$ s.t. $\delta(t_1) = \delta(t)$, then $t_1 \models \phi$.
- $t \models \Box_{\neq}\psi$ iff $\forall t_1 < t$ s.t. $\delta(t_1) \neq \delta(t)$, then $t_1 \models \psi$.

DataGL : a bimodal logic

Definition

Two relations $R^{=}$ and $R^{\neq}$:

$$\begin{cases} tR^{=}t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) = \delta(t), \\ tR^{\neq}t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) \neq \delta(t). \end{cases}$$

DataGL : a bimodal logic

Definition

Two relations $R^=$ and $R^{\neq}$:

$$\begin{cases} tR^= t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) = \delta(t), \\ tR^{\neq} t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) \neq \delta(t). \end{cases}$$

- $R^=$ transitive and irreflexive,

DataGL : a bimodal logic

Definition

Two relations $R^{=}$ and $R^{\neq}$:

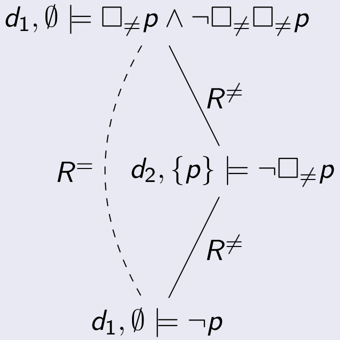
$$\begin{cases} tR^{=}t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) = \delta(t), \\ tR^{\neq}t_1 & \textit{iff} \quad t_1 < t \textit{ and } \delta(t_1) \neq \delta(t). \end{cases}$$

- $R^{=}$ transitive and irreflexive,
- $R^{\neq}$ irreflexive,
  and if $t_1 R^{\neq} t_2$ and $t_2 R^{\neq} t_3$, then $t_1 R^{\neq} t_3$ or $t_1 R^{=} t_3$.

DataGL : an example

Example

- $\Box_= p \supset \Box_= \Box_= p$ *is valid,*
- *but* $\Box_{\neq} p \supset \Box_{\neq} \Box_{\neq} p$ *is not.*



DataGL without Data = GL

- GL = complete logic of finite trees,

DataGL without Data = GL

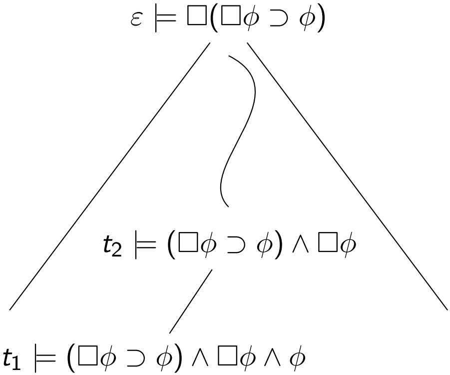
- GL = complete logic of finite trees,
- $\Box\phi := \Box_{=}\phi \wedge \Box_{\neq}\phi$,

DataGL without Data = GL

- GL = complete logic of finite trees,
- $\Box\phi := \Box_{=}\phi \wedge \Box_{\neq}\phi$,
- Loeb axiom (L) : $\Box(\Box\phi \supset \phi) \supset \Box\phi$.

DataGL without Data = GL

- GL = complete logic of finite trees,
- $\Box\phi := \Box_{=}\phi \wedge \Box_{\neq}\phi$,
- Loeb axiom (L) : $\Box(\Box\phi \supset \phi) \supset \Box\phi$.



Avron's sequent calculus (1984)

Complete sequent calculus for GL
extending a standard propositional calculus with :

$$\frac{\Box\Gamma, \Gamma, \Box\phi \Rightarrow \phi}{\Box\Gamma \Rightarrow \Box\phi}\,\Box$$

Avron's sequent calculus (1984)

Complete sequent calculus for GL
extending a standard propositional calculus with :

$$\frac{\Box\Gamma, \Gamma, \Box\phi \Rightarrow \phi}{\Box\Gamma \Rightarrow \Box\phi}\ \Box$$

$$\varepsilon \models \Box\Gamma$$

$$t \models \Box\Gamma \wedge \Gamma \wedge \Box\phi$$

Avron's sequent calculus (1984)

Complete sequent calculus for GL
extending a standard propositional calculus with :

$$\frac{\Box\Gamma, \Gamma, \Box\phi \Rightarrow \phi}{\Box\Gamma \Rightarrow \Box\phi}\ \Box$$

$$\varepsilon \models \Box\Gamma$$

$$t \models \Box\Gamma \wedge \Gamma \wedge \Box\phi$$

- Extraction of the information of $\Box\Gamma$ thanks to transitivity and irreflexivity,

Avron's sequent calculus (1984)

Complete sequent calculus for GL
extending a standard propositional calculus with :

$$\frac{\Box\Gamma, \Gamma, \Box\phi \Rightarrow \phi}{\Box\Gamma \Rightarrow \Box\phi}\ \Box$$

$\varepsilon \models \Box\Gamma$

$t \models \Box\Gamma \wedge \Gamma \wedge \Box\phi$

- Extraction of the information of $\Box\Gamma$ thanks to transitivity and irreflexivity,
- $\Box\phi$ is here to integrate axiom (L).

Toward a first proof system for DataGL

- $\Box_=$ has the same behavior as $\Box$,

Toward a first proof system for DataGL

- $\Box_=$ has the same behavior as $\Box$,
- thus a sound rule for $\Box_=$ will be :

$$\frac{\Box_= \Gamma^=, \Gamma^=, \Box_{\neq} \Gamma^{\neq}, \Box_= \phi \Rightarrow \phi}{\Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq} \Rightarrow \Box_= \phi}\ \Box_=$$

Toward a first proof system for DataGL

- $\Box_=$ has the same behavior as $\Box$,
- thus a sound rule for $\Box_=$ will be :

$$\dfrac{\Box_=\Gamma^=, \Gamma^=, \Box_{\neq}\Gamma^{\neq}, \Box_=\phi \Rightarrow \phi}{\Box_=\Gamma^=, \Box_{\neq}\Gamma^{\neq} \Rightarrow \Box_=\phi}\ \Box_=$$

$$\varepsilon \models \Box_=\Gamma^= \wedge \Box_{\neq}\Gamma^{\neq}$$

$$\Big|\ R^=$$

$$t \models \Box_=\Gamma^= \wedge \Gamma^= \wedge \Box_{\neq}\Gamma^{\neq} \wedge \Box_=\phi$$

Sequent with history

- Issue : $R^{\neq}$ not transitive.

Sequent with history

- Issue : $R^{\neq}$ not transitive.
  - $\rightarrow$ Follow $R^{\neq}$ twice $\Rightarrow$ need to consider two cases depending on the newly encountered datum.

Sequent with history

- Issue : $R^{\neq}$ not transitive.
  - $\rightarrow$ Follow $R^{\neq}$ twice $\Rightarrow$ need to consider two cases depending on the newly encountered datum.
  - $\rightarrow$ Follow $R^{\neq}$ n times $\Rightarrow$ n cases to consider.

Sequent with history

- Issue : $R^{\neq}$ not transitive.
  - $\rightarrow$ Follow $R^{\neq}$ twice $\Rightarrow$ need to consider two cases depending on the newly encountered datum.
  - $\rightarrow$ Follow $R^{\neq}$ n times $\Rightarrow$ n cases to consider.
  - $\rightarrow$ Furthermore, need to remember the previous hypotheses.

Sequent with history

- Issue : $R^{\neq}$ not transitive.
  - $\rightarrow$ Follow $R^{\neq}$ twice $\Rightarrow$ need to consider two cases depending on the newly encountered datum.
  - $\rightarrow$ Follow $R^{\neq}$ n times $\Rightarrow$ n cases to consider.
  - $\rightarrow$ Furthermore, need to remember the previous hypotheses.
- Solution : Use an history $\mathcal{H}$ which remembers the hypotheses previously encountered.

History

The history $\mathcal{H}$ is a finite multi-set of sets of formulæ.
We will denote it $\mathcal{H} = H_0; \ldots; H_n$.

Rule for the modality $\Box_{\neq}$

- The two-premises case :

$$\dfrac{\begin{array}{l}\ldots \\ \ldots\end{array}}{\Box_{=} H_0^{=}, \Box_{\neq} H_0^{\neq}; \Box_{=} \Gamma^{=}, \Box_{\neq} \Gamma^{\neq} \Rightarrow \Box_{\neq} \psi} \; \Box_{\neq}$$

Rule for the modality $\Box_{\neq}$

- The two-premises case :

$$\dfrac{\begin{array}{c}\ldots \\ \ldots\end{array}}{\Box_{=} H_0^{=}, \Box_{\neq} H_0^{\neq} ; \Box_{=} \Gamma^{=}, \Box_{\neq} \Gamma^{\neq} \Rightarrow \Box_{\neq} \psi}\ \Box_{\neq}$$

$$\varepsilon \models \Box_{=} H_0^{=} \wedge \Box_{\neq} H_0^{\neq}$$

$$\Big|\ R^{\neq}$$

$$t \models \Box_{=} \Gamma^{=} \wedge \Box_{\neq} \Gamma^{\neq}$$

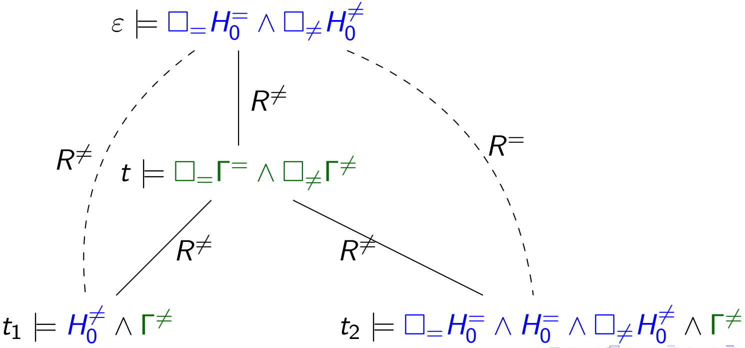
- The two-premises case :

$$\frac{\begin{array}{c}\Box_= H_0^=, \Box_{\neq} H_0^{\neq}; \Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq}, \Box_{\neq}\psi; H_0^{\neq}, \Gamma^{\neq} \Rightarrow \psi \\ \cdots \end{array}}{\Box_= H_0^=, \Box_{\neq} H_0^{\neq}; \Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq} \Rightarrow \Box_{\neq}\psi}\;\Box_{\neq}$$

$$\begin{tikzcd}
& \varepsilon \models \Box_= H_0^= \wedge \Box_{\neq} H_0^{\neq} \arrow[d, dash, "R^{\neq}"] \arrow[ddl, dash, dashed, bend right, "R^{\neq}"'] \\
& t \models \Box_= \Gamma^= \wedge \Box_{\neq} \Gamma^{\neq} \arrow[dl, dash, "R^{\neq}"] \\
t_1 \models H_0^{\neq} \wedge \Gamma^{\neq} &
\end{tikzcd}$$

- The two-premises case :

$$\dfrac{\begin{array}{c}\Box_= H_0^=, \Box_{\neq} H_0^{\neq}; \Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq}, \Box_{\neq}\psi; H_0^{\neq}, \Gamma^{\neq} \Rightarrow \psi \\ \Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq}, \Box_{\neq}\psi; \Box_= H_0^=, H_0^=, \Box_{\neq} H_0^{\neq}, \Gamma^{\neq} \Rightarrow \psi\end{array}}{\Box_= H_0^=, \Box_{\neq} H_0^{\neq}; \Box_= \Gamma^=, \Box_{\neq} \Gamma^{\neq} \Rightarrow \Box_{\neq}\psi}\ \Box_{\neq}$$



Example of a proof search

$$\dfrac{\dfrac{\dfrac{\dfrac{\qquad\qquad\qquad \qquad\qquad\qquad}{\qquad\qquad}}{\qquad}}{\qquad}}{;\ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}$$

Example of a proof search

$$\cfrac{\cfrac{}{;\Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}}{;\ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\ R\supset$$

Example of a proof search

$$\dfrac{\dfrac{\color{blue}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi}; \psi \Rightarrow \color{red}{\Box_{\neq}\psi}}{; \Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}\,\Box_{\neq}}{;\ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\,R\supset$$

Example of a proof search

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \Box_{\neq}\psi; \psi, \Box_{\neq}\psi \Rightarrow \psi} \qquad \dfrac{}{}}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \psi \Rightarrow \Box_{\neq}\psi}\,\Box}{; \Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}\,\Box_{\neq}}{;\ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\,R\supset
$$

Example of a proof search

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \Box_{\neq}\psi; \psi, \Box_{\neq}\psi \Rightarrow \psi}\ ax \qquad \dfrac{}{}}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \psi \Rightarrow \Box_{\neq}\psi}\ \Box}{; \Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}\ \Box_{\neq}}{;\ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\ R\supset
$$

Example of a proof search

$$\dfrac{\dfrac{\dfrac{}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \Box_{\neq}\psi; \psi, \Box_{\neq}\psi \Rightarrow \psi}\ ax \qquad \dfrac{}{\Box_{\neq}\psi; \Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi \Rightarrow \psi}}{\dfrac{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \psi \Rightarrow \Box_{\neq}\psi}{; \Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}\ \Box_{\neq}}\ \Box}{; \ \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\ R\supset$$

Example of a proof search

$$
\dfrac{
  \dfrac{}{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \Box_{\neq}\psi; \psi, \Box_{\neq}\psi \Rightarrow \psi}\, ax
  \qquad
  \dfrac{\text{Failure}}{\Box_{\neq}\psi; \Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi \Rightarrow \psi}
}{
  \dfrac{\Box_{\neq}\psi, \Box_{\neq}\Box_{\neq}\psi; \psi \Rightarrow \Box_{\neq}\psi}{
    \dfrac{; \Box_{\neq}\psi \Rightarrow \Box_{\neq}\Box_{\neq}\psi}{; \Rightarrow \Box_{\neq}\psi \supset \Box_{\neq}\Box_{\neq}\psi}\, R\supset
  }\, \Box_{\neq}
}\, \Box
$$

Rule for $\Box_{\neq}$

$$\dfrac{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; H_i^{\neq}, \Gamma^{\neq} \Rightarrow \psi \qquad (\mathcal{H} \setminus H_i; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; \Box_{=}H_i^{=}, H_i^{=}, \Box_{\neq}H_i^{\neq}, H_{j_{j\neq i}}^{\neq}, \Gamma^{\neq} \Rightarrow \psi)_i}{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq} \Rightarrow \Box_{\neq}\psi} \Box_{\neq}$$

Rule for $\Box_{\neq}$

$$\dfrac{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; H_i^{\neq}, \Gamma^{\neq} \Rightarrow \psi \qquad (\mathcal{H} \setminus H_i; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; \Box_{=}H_i^{=}, H_i^{=}, \Box_{\neq}H_i^{\neq}, H_{j_{j\neq i}}^{\neq}, \Gamma^{\neq} \Rightarrow \psi)_i}{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq} \Rightarrow \Box_{\neq}\psi} \Box_{\neq}$$

- n premises $= n$ cases to consider, but 2 kind of cases :
- Extraction of the information depends on the case.

Rule for $\Box_{\neq}$

$$\dfrac{\mathcal{H}; \Box_= \Gamma^=, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; H_i^{\neq}, \Gamma^{\neq} \Rightarrow \psi \qquad (\mathcal{H} \setminus H_i; \Box_= \Gamma^=, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; \Box_= H_i^=, H_i^=, \Box_{\neq} H_i^{\neq}, H_{j_{j \neq i}}^{\neq}, \Gamma^{\neq} \Rightarrow \psi)_i}{\mathcal{H}; \Box_= \Gamma^=, \Box_{\neq}\Gamma^{\neq} \Rightarrow \Box_{\neq}\psi} \, \Box_{\neq}$$

- n premises $= n$ cases to consider, but 2 kind of cases :
  - $\rightarrow$ First premise : datum of the arrival node different from the one of all other nodes.
- Extraction of the information depends on the case.

Rule for $\Box_{\neq}$

$$\frac{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; H_i^{\neq}, \Gamma^{\neq} \Rightarrow \psi \qquad (\mathcal{H} \setminus H_i; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq}, \Box_{\neq}\psi; \Box_{=}H_i^{=}, H_i^{=}, \Box_{\neq}H_i^{\neq}, H_{j_{j\neq i}}^{\neq}, \Gamma^{\neq} \Rightarrow \psi)_i}{\mathcal{H}; \Box_{=}\Gamma^{=}, \Box_{\neq}\Gamma^{\neq} \Rightarrow \Box_{\neq}\psi} \Box_{\neq}$$

- n premises $=$ n cases to consider, but 2 kind of cases :
  - $\rightarrow$ First premise : datum of the arrival node different from the one of all other nodes.
  - $\rightarrow$ Other premises : datum of the arrival node is the same than the one of a previous node.
- Extraction of the information depends on the case.

Adequation of the calculus

Soundness theorem

The sequent calculus is sound.

Completeness theorem

The sequent calculus is complete.

Properties of our calculus

Small proof property

Let ϕ a provable formula.
We can construct a proof tree of polynomial depth.

Properties of our calculus

Small proof property

Let ϕ a provable formula.
We can construct a proof tree of polynomial depth.

Upper complexity bound

The validity problem is PSPACE-easy.

Properties of our calculus

Small proof property

Let ϕ a provable formula.
We can construct a proof tree of polynomial depth.

Upper complexity bound

The validity problem is PSPACE-easy.

Small model property

Let ϕ be a non-provable formula.
We can build a counter-model of polynomial depth.

Lower complexity bound

Chagrov & Rybakov, 2002

The validity problem in GL is PSPACE-hard.

Lower complexity bound

Chagrov & Rybakov, 2002

The validity problem in GL is PSPACE-hard.

Theorem

The validity problem in DataGL is PSPACE-hard.

Lower complexity bound

Chagrov & Rybakov, 2002

The validity problem in GL is PSPACE-hard.

Theorem

The validity problem in DataGL is PSPACE-hard.

Corollary

The validity problem in DataGL is PSPACE-complete.

Conclusion and prospects

- Contributions :

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,

Conclusion and prospects

- Contributions :
    - Sound and complete sequent calculus,
    - Optimal complexity results,

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,
  - Optimal complexity results,
  - Small model property.

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,
  - Optimal complexity results,
  - Small model property.
- Prospects :

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,
  - Optimal complexity results,
  - Small model property.
- Prospects :
  - Implementation,

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,
  - Optimal complexity results,
  - Small model property.
- Prospects :
  - Implementation,
  - Cut-elimination procedure,

Conclusion and prospects

- Contributions :
  - Sound and complete sequent calculus,
  - Optimal complexity results,
  - Small model property.
- Prospects :
  - Implementation,
  - Cut-elimination procedure,
  - Extension of the calculus to larger fragments of XPath.

Questions

Questions ?