

Graph Rewriting for Model Construction in Modal Logic

Bilal SAID

Institut de Recherche en Informatique de Toulouse - IRIT
Université de Toulouse

29 January 2010

Locating my work

	Automated reasoning	
Theory	x	Practice
	Graph rewriting	

Panorama

- 1 Modal logics
- 2 Model Construction
- 3 Event-driven pattern matching

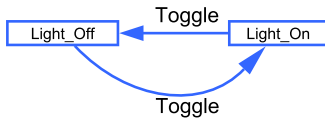
Outline

- 1 Modal logics
 - Modelling with graphs
 - Talking about graphs
 - Reasoning about graphs
- 2 Model Construction
- 3 Event-driven pattern matching

Example: Switch-Bulb system



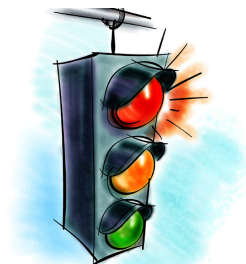
Example: Switch-Bulb system



Various systems, situations. . .



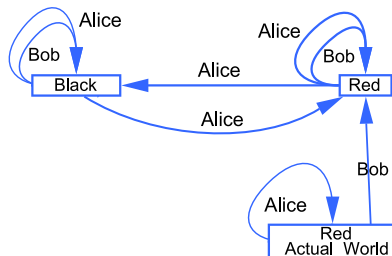
Guess-the-card game



Traffic-light

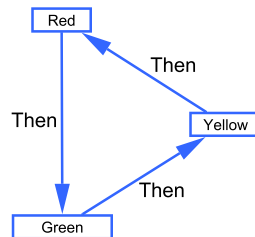
. . .

Various systems, situations...



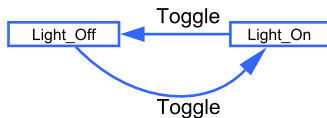
Knowledge

...



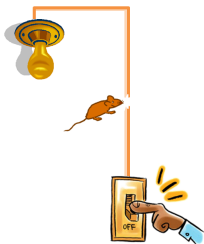
Time

Switch-Bulb system

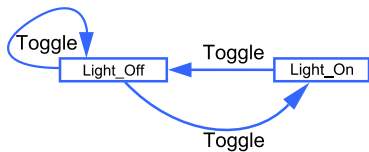
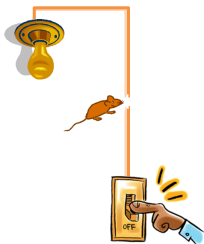


at the state “Light_Off”, after “Toggle” necessarily “Light_On”

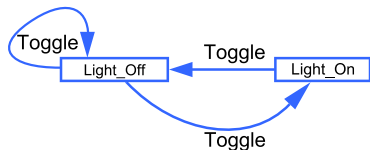
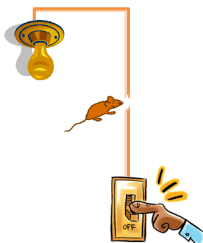
Switch-Bulb-Mouse



Switch-Bulb-Mouse

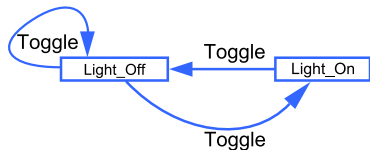
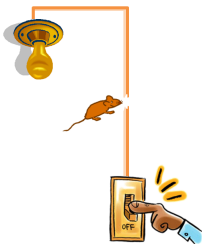


Switch-Bulb-Mouse



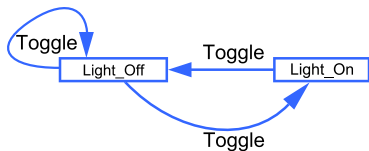
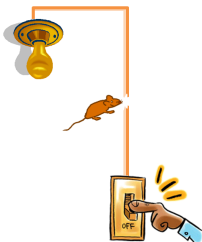
at state “Light_Off”, after “Toggle” not necessarily “Light_On”

Switch-Bulb-Mouse



at state "Light_Off", after "Toggle" not necessarily "Light_On"
i.e. possibly not "Light_On"

Switch-Bulb-Mouse



at state "Light_Off", after "Toggle" not necessarily "Light_On"
i.e. possibly not "Light_On"
i.e. possibly "Light_Off"

Formal language: Formulas

- “Light_Off” and after “Toggle” necessarily “Light_On”
 $Light_Off \wedge [Toggle] Light_On$
- “Light_Off” and after “Toggle” possibly “Light_Off”
 $Light_Off \wedge \langle Toggle \rangle Light_Off$

Generic sentences:

Necessarily A $\Box A$

Possibly A $\Diamond A$

How to evaluate these formulas?

Formal language: Formulas

- “Light_Off” and after “Toggle” necessarily “Light_On”

$$Light_Off \wedge [Toggle] Light_On$$
- “Light_Off” and after “Toggle” possibly “Light_Off”

$$Light_Off \wedge \langle Toggle \rangle Light_Off$$

Generic sentences:

Necessarily A $\Box A$

Possibly A $\Diamond A$

How to evaluate these formulas?

Formal language: Formulas

- “Light_Off” and after “Toggle” necessarily “Light_On”

$$Light_Off \wedge [Toggle] Light_On$$
- “Light_Off” and after “Toggle” possibly “Light_Off”

$$Light_Off \wedge \langle Toggle \rangle Light_Off$$

Generic sentences:

Necessarily A $\Box A$

Possibly A $\Diamond A$

How to evaluate these formulas?

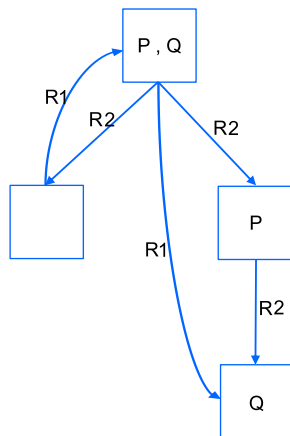
Kripke models

Model

= transition system

- possible **W**orlds
= states
- accessibility **R**elation
= transitions
- **V**aluation
= labeling function

$$M = (W, R, V)$$



Semantics: Truth conditions

■ Atoms

- $M, w \Vdash P$ iff $P \in V(w)$

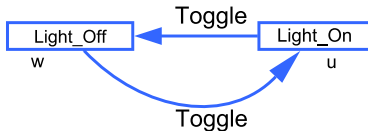
■ Classical operators

- $M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$
- $M, w \Vdash A \vee B$ iff $M, w \Vdash A$ or $M, w \Vdash B$
- ...

■ Modal operators

- $M, w \Vdash \Box A$ iff for all u , if wRu then $M, u \Vdash A$
- $M, w \Vdash \Diamond A$ iff exists u s.t. wRu and $M, u \Vdash A$

Example



- $M, u \models \text{Light_On}$
- $M, w \models \text{Light_Off} \wedge [\text{Toggle}]\text{Light_On}$
- ...

Various systems, situations. . .

Examples

- The card is red, *Alice knows* it and *knows* that Bob *does not*
 $Card_Red \wedge K_{Alice} Card_Red \wedge K_{Alice} \neg K_{Bob} Card_Red$
- *Always*, if it is Red then *next* it turns out Green
 $G(Red \rightarrow XGreen)$
- . . .

How to take these into account?

- Change truth conditions
- Constraints on R
- Constraints on V

Various systems, situations. . .

Examples

- The card is red, *Alice knows* it and *knows* that Bob *does not*
 $Card_Red \wedge K_{Alice} Card_Red \wedge K_{Alice} \neg K_{Bob} Card_Red$
- *Always*, if it is Red then *next* it turns out Green
 $G(Red \rightarrow XGreen)$
- . . .

How to take these into account?

- Change truth conditions
- Constraints on R
- Constraints on V

Constraints on R

One relation:

- Transitive
future of future is future
- Reflexive
I know s.th. hence it is true
- Serial
there is always a future
- Symmetric
- Equivalence (*universal*)
- Confluent (*Church-Rosser*)
- ...

Two or more:

- R_I included in R_J
- $R_I = R_J \cup R_K$
- $R_J = (R_I)^{-1}$
- $R_J = (R_I)^*$
(transitive closure)
- $R_I \circ R_J = R_J \circ R_I$
- Confluent
- ...

Constraints on R

One relation:

- Transitive
future of future is future
- Reflexive
I know s.th. hence it is true
- Serial
there is always a future
- Symmetric
- Equivalence (*universal*)
- Confluent (*Church-Rosser*)
- ...

Two or more:

- R_I included in R_J
- $R_I = R_J \cup R_K$
- $R_J = (R_I)^{-1}$
- $R_J = (R_I)^*$
(transitive closure)
- $R_I \circ R_J = R_J \circ R_I$
- Confluent
- ...

Constraints on R

One relation:

- Transitive
future of future is future
- Reflexive
I know s.th. hence it is true
- Serial
there is always a future
- Symmetric
- Equivalence (*universal*)
- Confluent (*Church-Rosser*)
- ...

Two or more:

- R_I included in R_J
- $R_I = R_J \cup R_K$
- $R_J = (R_I)^{-1}$
- $R_J = (R_I)^*$
(transitive closure)
- $R_I \circ R_J = R_J \circ R_I$
- Confluent
- ...

Constraints on R

One relation:

- Transitive
future of future is future
- Reflexive
I know s.th. hence it is true
- Serial
there is always a future
- Symmetric
- Equivalence (*universal*)
- Confluent (*Church-Rosser*)
- ...

Two or more:

- R_I included in R_J
- $R_I = R_J \cup R_K$
- $R_J = (R_I)^{-1}$
- $R_J = (R_I)^*$
(transitive closure)
- $R_I \circ R_J = R_J \circ R_I$
- Confluent
- ...

Constraints on V

- HL(@): a nominal is true at a **unique** world
- Intuitionistic: atoms **persist** along paths
- PAL: literals **persist** throughout (updated) models
- ...

Interests

Given		Question
Property P	System S	does S have P?
Property P		is there a system S having P?
Property P		which system S may have P?

Problems

	Input		Output
	Formula	Model	
Model Checking	x	x	Yes/No
Satisfiability / Validity	x		Yes/No
Model Construction	x		Model/Counter-Model

Methods

Since 1950's...

- Sequent calculi [Beth, Gentzen, ...]
 - Proof search
- Tableau calculi [Smullyan, Fitting, Massacci, ...]
 - Model existence check
- Tableau by graph rewriting [à la toulousaine]
 - Model construction

Example

a node with the input formula

$$\Box P \ \& \ \leftrightarrow \ Q \ \& \ \leftrightarrow \ (R \vee \sim P)$$

Example

$M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$

A

B

$\Box P \ \& \ \leftrightarrow \ Q \ \& \ \leftrightarrow \ (R \vee \sim P)$

Example

$M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$

$$\begin{array}{c} \Box P \ \& \ \Box Q \ \& \ \Box (R \vee \sim P) \\ \Box P \\ \Box Q \ \& \ \Box (R \vee \sim P) \end{array}$$

Example

$M, w \Vdash A \wedge B$ iff $M, w \Vdash A$ and $M, w \Vdash B$

$\Box P \ \& \ \langle \rangle Q \ \& \ \langle \rangle (R \vee \sim P)$

$\Box P$

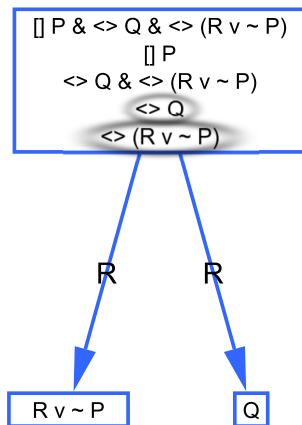
$\langle \rangle Q \ \& \ \langle \rangle (R \vee \sim P)$

$\langle \rangle Q$

$\langle \rangle (R \vee \sim P)$

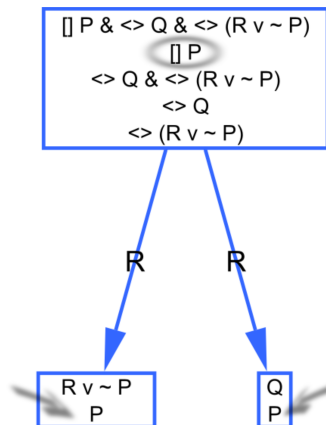
Example

$M, w \models \Diamond A$ iff $\exists u \mid wRu$ and $M, u \models A$



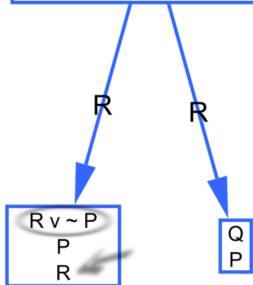
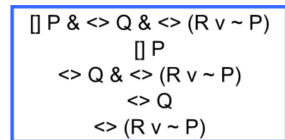
Example

$M, w \models \Box A$ iff $\forall u : wRu$ then $M, u \models A$

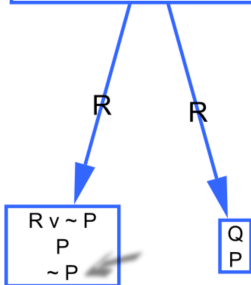
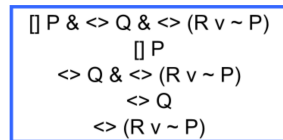


Example

$M, w \models A \vee B$ iff $M, w \models A$ or $M, w \models B$

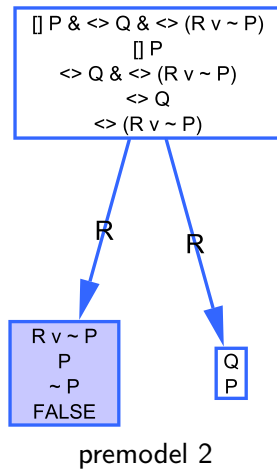
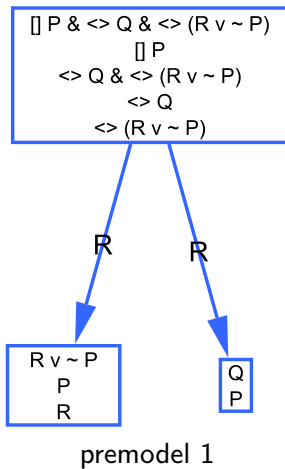


premodel 1

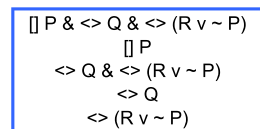


premodel 2

Example



Example



R

R



premodel 1



Model

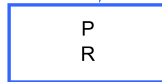
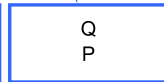


extraction


 $\sim P$ [or P]
 $\sim Q$ [or Q]
 $\sim R$ [or R]

R

R

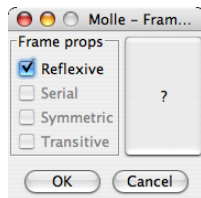
 $\sim Q$ [or Q] $\sim R$ [or R]
 $M, w \Vdash P$ then $P \in V(w)$

Reasoning Tools

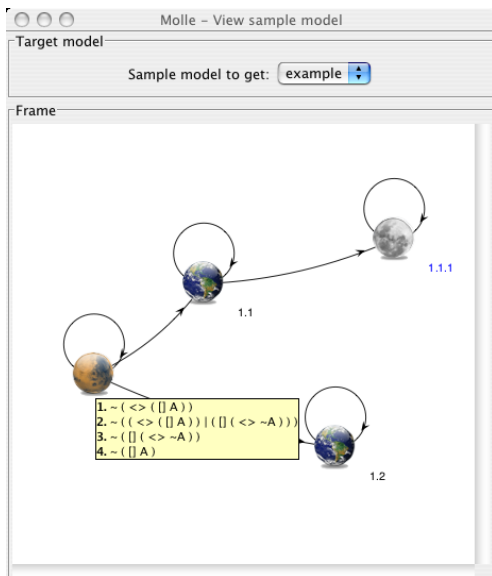
- Fast (but: geek!, not generic)
 - FaCT [Horrocks]
 - LWB [Heuerding]
 - K-SAT [Giunchiglia & Sebastiani]
- Generic (but: limited, requires coding in OCaml)
 - TWB [Abate & Goré]
- Educative (but: not generic!)
 - OOPS [Valkenhoef, Vaart & Verbrugge]
 - Molle [Mazzucchi & Mocci]

Reasoning Tools

Molle



Reflexivity only!



Reasoning Tools

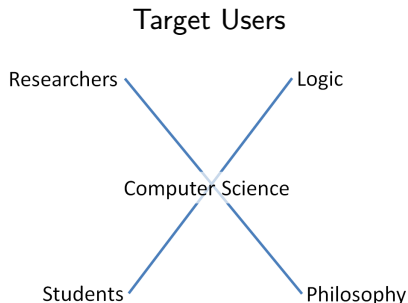
OOPS

- Model update
 - High-level language
 - Graph viewer
- $S5_n$ only!



Desiderata

- Generic: users' own new methods
- Educative: user-friendly
- Performance: reasonable time



Outline

- 1 Modal logics
- 2 Model Construction
 - Graph rewriting rules
 - On paper
 - In LoTREC
 - In Demo
- 3 Event-driven pattern matching

Uniform methodology

Semantics

via

Graph rewriting rules

How?

What are graph rewriting rules?

Uniform methodology

Semantics

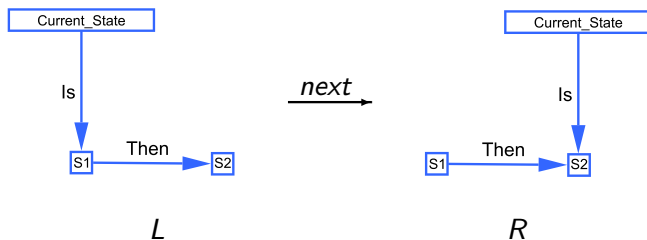
via

Graph rewriting rules

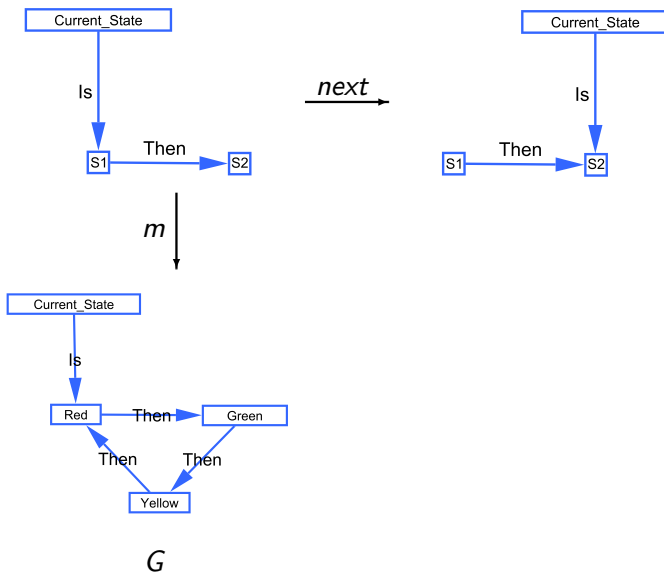
How?

What are graph rewriting rules?

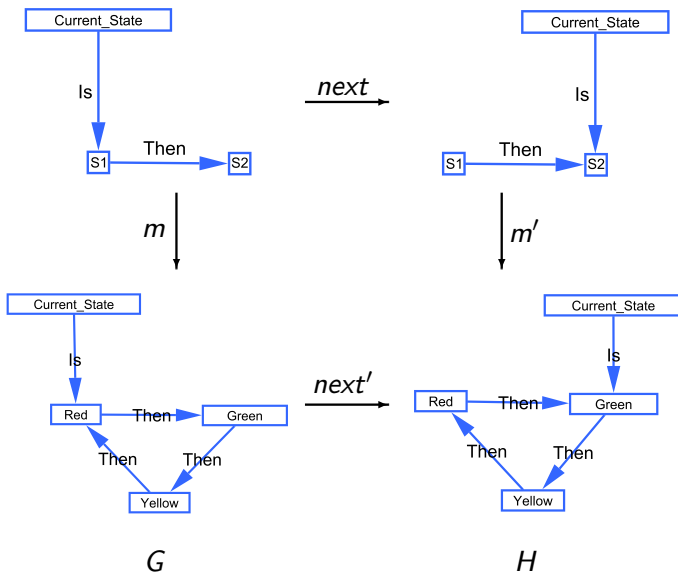
Definition



Matching



Application



Uniform methodology

Semantics

via

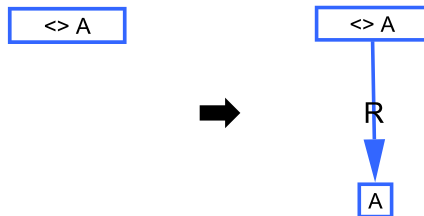
Graph rewriting rules

How?

~~What are graph rewriting rules?~~

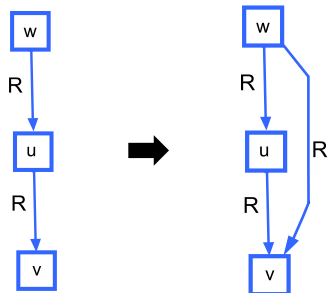
Truth conditions

$M, w \Vdash \Diamond A$ iff $\exists u \mid wRu$ and $M, u \Vdash A$



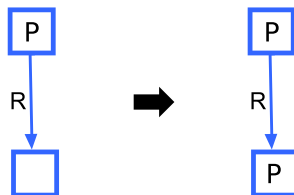
R -Constraints

Transitivity



V-Constraints

Persistence



Certifying the method

- Termination: does it halt?
- Soundness: does it consider ALL the semantics?
- Completeness: does it consider s.th. ELSE?
- Complexity: what is its time / space costs?
- ...

How to experiment with it?

User-defined language

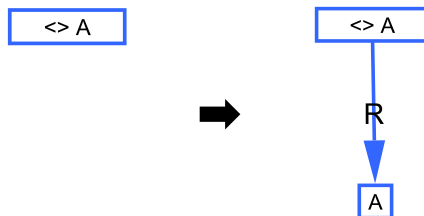
Example (definition)

name	arity	display
not	1	$\sim -$
and	2	$- \ \& \ -$
...		
pos	1	$\langle \rangle \ -$
...		

Example (usage)

- pos P
 $\langle \rangle \ P$
- and not Q not P
 $\sim Q \ \& \ \sim P$

User-defined rules



Rule Pos

hasElement w pos variable A

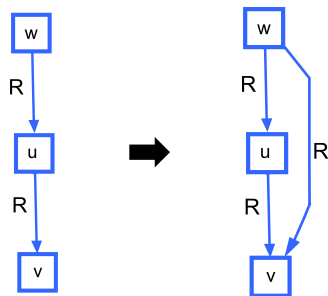
createNewNode u

link w u R

add u variable A

End

User-defined rules



Rule Pos

isLinked w u R

isLinked u v R

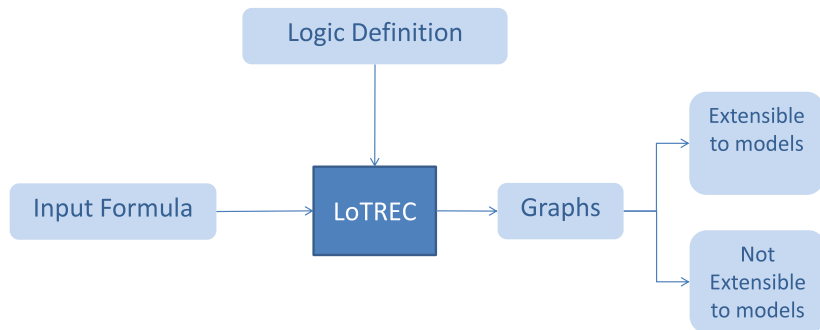
link w u R

End

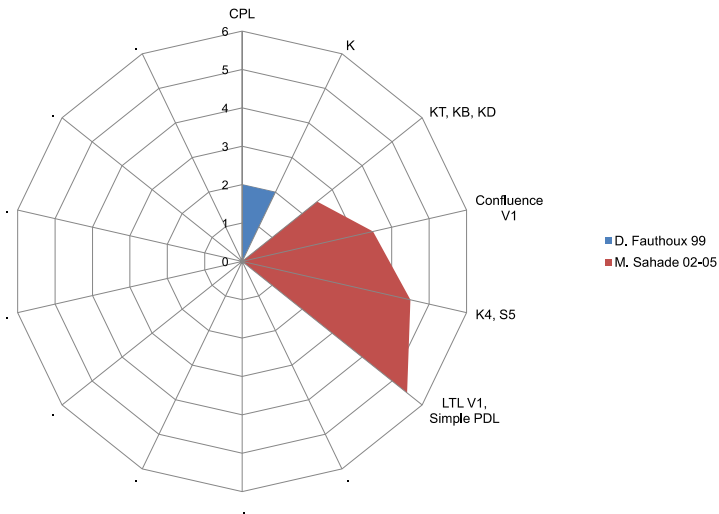
User-defined strategies

- Ordering on the rules
- Saturation (repeat...end)
- Priority (firstRule...end)

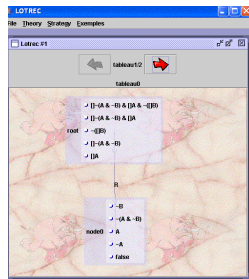
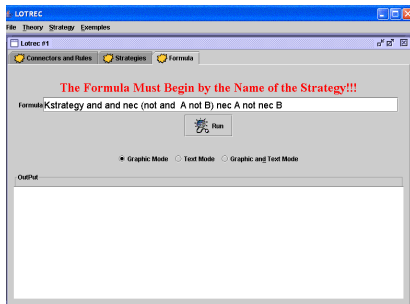
The black box



What was done?



What was done?



Some of what I did

LoTREC 7.0 - Tableaux Theorem Prover - S4Optimal.xml

Logic View Premodels Help

Loaded Logics

S4Optimal.xml

Connectors Rules Strategies Predefined Formulas

Rules List

- Stop
- NotNot
- And
- NotOr
- NotImp
- NotAnd
- NotEquiv
- Imp
- Equiv
- CutOrLeft
- CutOrRight
- CutOrNotLeft
- CutOrNotRight
- Or
- propagateNec2ActualWorld
- propagateNotPos2ActualWorld
- copyNec
- copyNotPos

Selected Rule

Name Stop

Conditions

hasElement node0 variable a
hasElement node0 not variable a

Actions

add node0 FALSE
stop node0

Comments

Premodels Construction Settings

Select a Formula $\Box (\Box (P \vee Q)) \wedge \Box (\Box (\neg P)) \wedge \Box (\Box (\neg R))$

Or Compose your own formula

and and nec pos nec or P Q nec pos not P pos nec not R

[infix formula editor...](#)

Build Premodels Step By Step... Satisfiability Check...

Controls

Next Rule:

Next Pause Stop

Premodels List

Tableaux Tree

- premodel.1
- premodel.2.1
- premodel.3.1
- premodel.4.1
- premodel.5
- premodel.2.2.1
- premodel.2.3.1
- premodel.2.4
- premodel.3.2.1
- premodel.3.3
- premodel.4.2
- premodel.2.2.2.1
- premodel.2.2.3
- premodel.3.2.2
- premodel.2.2.2.2

Premodels Views

premodel.1

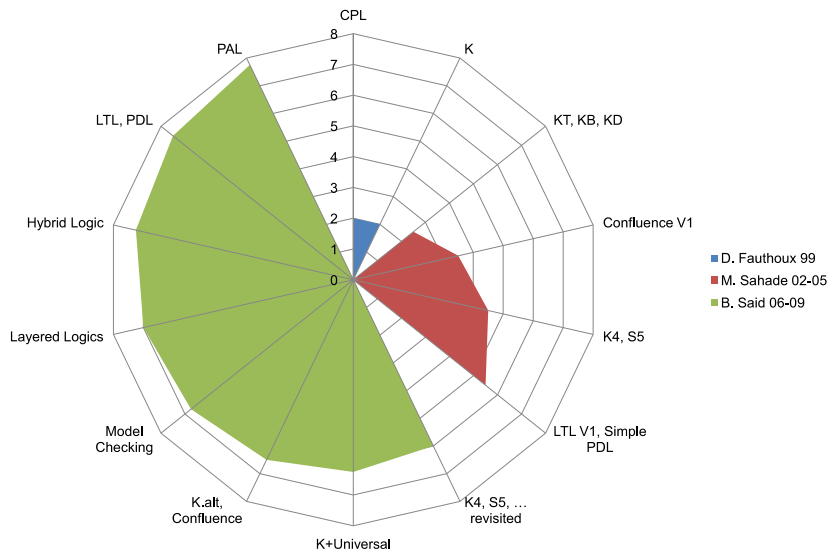
The diagram shows a complex tableau tree structure with nodes containing logical formulas and branching rules like Loop, Contradiction, and Cut.

Some of what I did

Added techniques:

- One occurrence rules application (K.alt, LTL,...)
- Defining non-injective patterns (Confluence)
- Extend the language (Model Checking,...)
- Model checking (LTL, PDL,...)
- Labeled formulas (PAL)
- Nodes as memory cells (S4 + histories)
- Run in step-by-step mode (debugging)
- Code new extensions
- ...

Some of what I did



Running live

<http://www.irit.fr/Lotrec>

The screenshot shows a web browser window displaying the LoTREC Generic Tableau Prover website. The browser's address bar shows the URL <http://www.irit.fr/ACTIVITES/LILaC/Lotrec/>. The website has a blue header with navigation links: Home, Run LoTREC, Download, Documentation, news, contact, login, links, and about. A search bar is located on the right side of the header. The main content area features a large banner with the text "Proving by Tableau becomes easier..." and a section titled "LoTREC 2.0 Release" dated Mon, 04/14/2008 - 15:16 by Bilal Said. Below this, it states "LoTREC 2.0 is released:" and lists two items: "Launch the LoTREC Web Start (1)" with a "Webstart" button, and "Download executable package (2)". A "What's New" section follows, detailing updates like "Tableau builder controls" and "Step-by-Step run mode". The footer includes "Add new comment", "Read more", and "S4 with histories".

LoTREC Generic Tableau Prover

Proving by Tableau becomes easier...

LoTREC 2.0 Release
Mon, 04/14/2008 - 15:16 — Bilal Said

LoTREC 2.0 is released:

- Launch the LoTREC Web Start (1)

Webstart

- Download executable package (2)

What's New

- Tableau builder controls:** stop, pause/resume mode were introduced to give the user more control during the tableaux construction. Thus, endless loops, that may be raised from unpredicted behavior of some rules combinations, can be stopped manually by the user.
- Step-by-Step run mode:** users can make break points on any sub set of their rules,

[Add new comment](#) [Read more](#)

S4 with histories

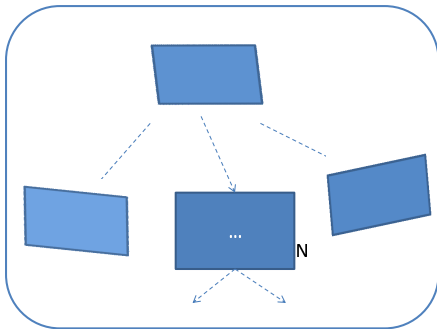
Outline

- 1 Modal logics
- 2 Model Construction
- 3 Event-driven pattern matching
 - Mechanism
 - Semantics
 - Evaluation

Naive pattern matching

C1

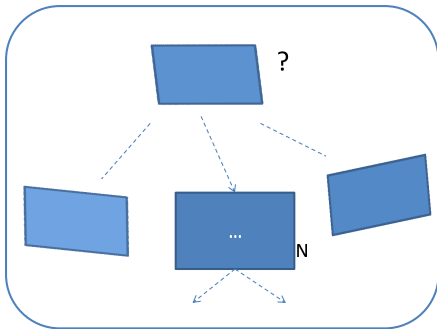
• hasElement n ◇ A



Naive pattern matching

C1

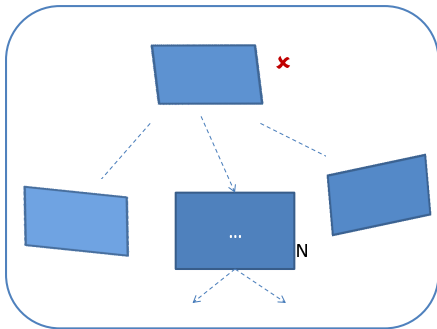
• hasElement n ◇ A



Naive pattern matching

C1

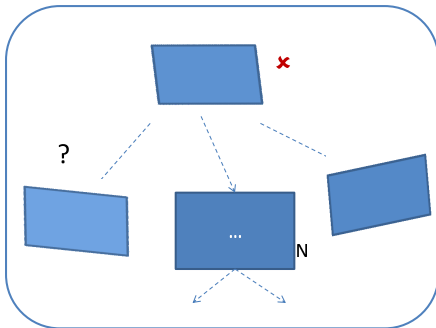
• hasElement n ◇ A



Naive pattern matching

C1

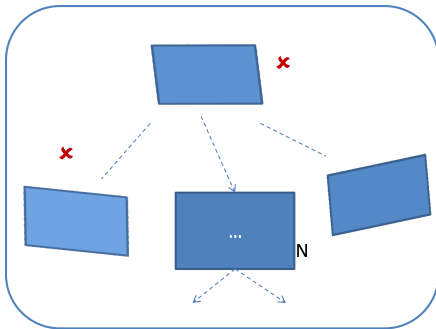
• hasElement n ◇ A



Naive pattern matching

C1

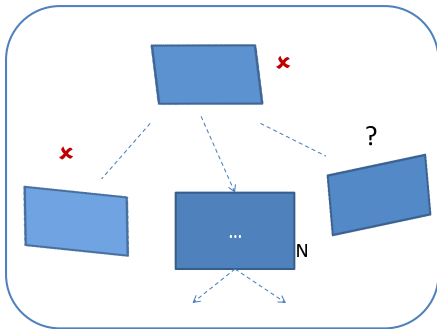
• hasElement n ◇ A



Naive pattern matching

C1

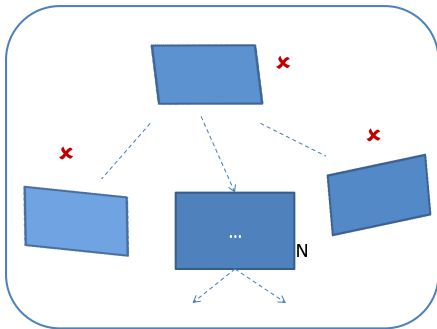
• hasElement n ◇ A



Naive pattern matching

C1

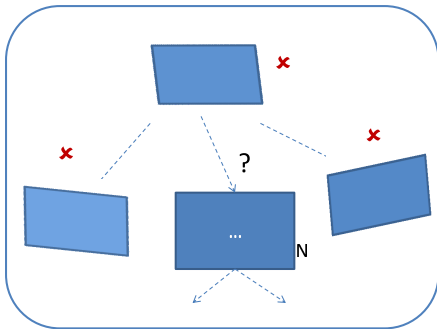
• hasElement n ◇ A



Naive pattern matching

C1

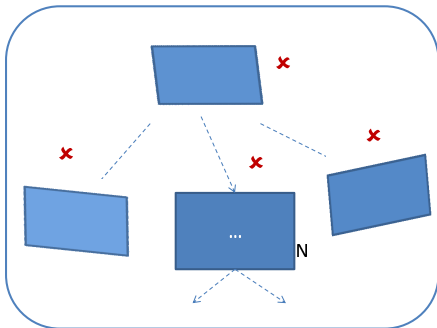
• hasElement n ◇ A



Naive pattern matching

C1

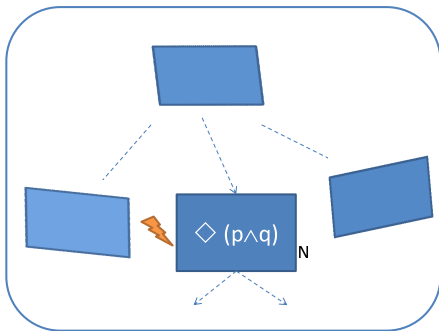
• hasElement n ◇ A



$$O(|G|^{|L|})$$

Naive pattern matching

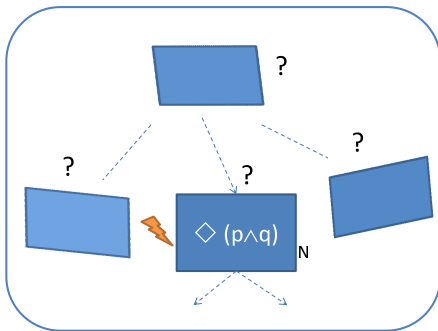
C1

• hasElement n $\diamond A$ 

$$O(|G|^{|L|})$$

Naive pattern matching

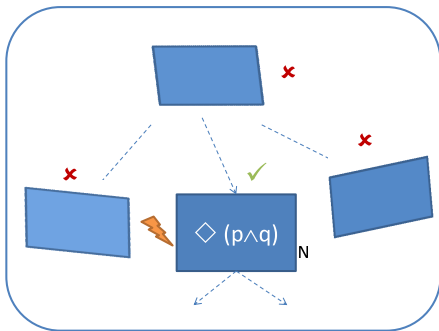
C1

• hasElement n $\Diamond$ A

$$O(|G|^{|L|})$$

Naive pattern matching

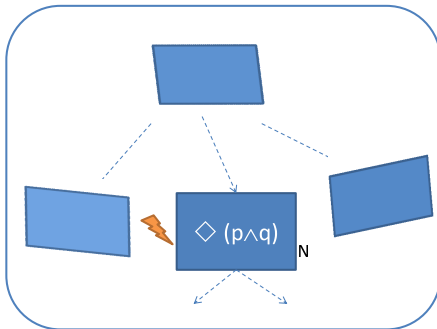
C1

• hasElement n $\Diamond A$ 

$$O(|G|^{|L|})$$

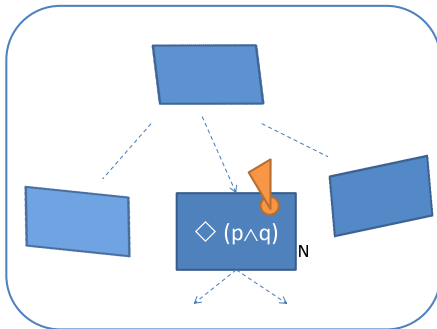
Event-based matching in LoTREC

C1

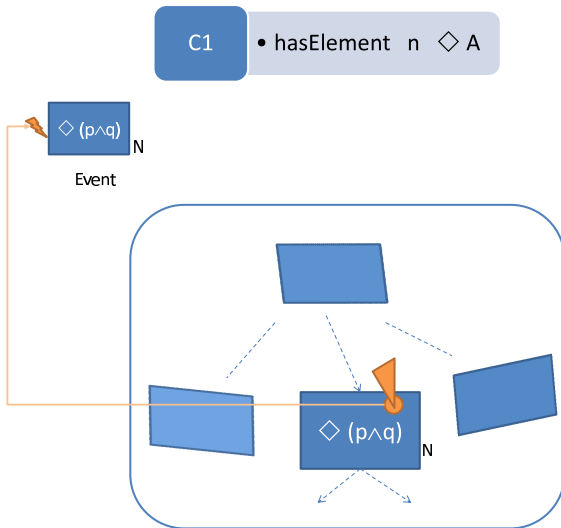
• hasElement n $\Diamond A$ 

Event-based matching in LoTREC

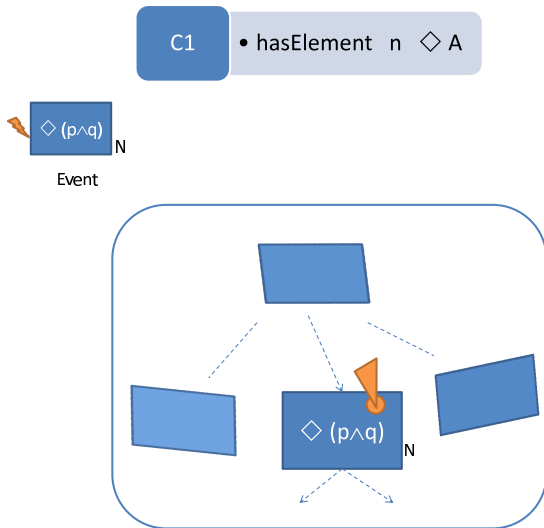
C1

• hasElement n $\diamond A$ 

Event-based matching in LoTREC



Event-based matching in LoTREC

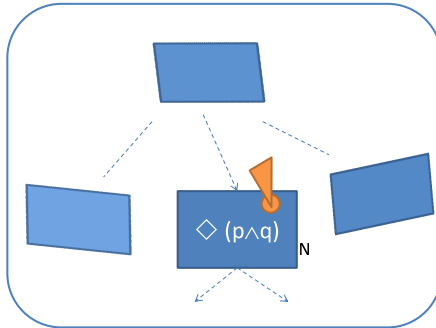


Event-based matching in LoTREC

C1

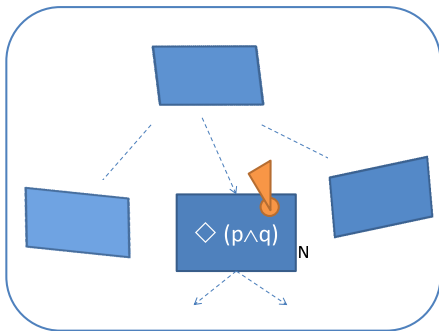
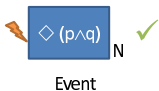
• hasElement n $\Diamond A$ 

Event



Event-based matching in LoTREC

C1

• hasElement $n \diamond A$ 

$$O(k^{|L|}) \text{ where } k \ll |G|$$

Equivalence to usual semantics

[Gasquet, Said & Schwarzentruher 09]

Rewriting **with** event-based matching

=

Rewriting **without** event-based matching

Reasons:

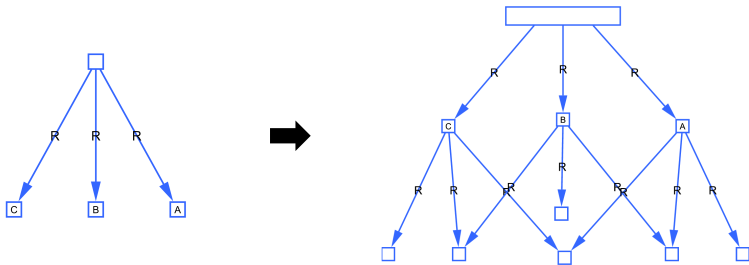
- Every successful pattern is considered
- Only unfruitful events are deleted

Related works

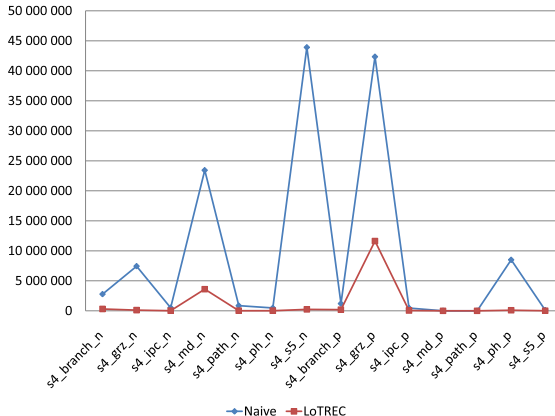
- PROGRES [Zündorf 99]
 - chooses an optimal plan over $|L|!$ local search plans
 - tracks invalid patterns
- Incremental Update [Varró & Varró 04]
 - tracks successful patterns in DB
 - stores & updates are space & time consuming

VS. other rewriting tools

- Comparison is not fair: formula matching
- Benchmark is hard to setup
- General purpose tools are not competent:
3 levels confluent graph takes:
 - ≈ 6 sec in AGG
 - < 0.5 sec in LoTREC



VS. naive pattern matching



Nb tentatives of pattern matching
Hardest S4 formulas in LWB benchmark

Conclusion

Research

Implementation of

- Time Sub-Intervals Logic
[Goranko et al. 08]
- Public Announcement Logic (PAL)
[De Lima et al. 09]

Not possible using another platform!

Academic

Accessed through logic courses:

- *Automated Reasoning*
Prof. C. Pêcheur
Université Catholique de Louvain, Belgium
- *FGI 3 - Logik*
Dr. C. Eschenbach
University of Hamburg, Germany
- *Logique, informatique et sciences cognitives*
Prof. R. Villemaire
University of Quebec at Montreal, Canada

Recap

Recap on the contributions of my thesis:

- Develop & maintain the LoTREC platform
- Study & implement new logics
- Promote the software in research & academic fields
- Establish the links with graph rewriting theory
- Study the event-based pattern matching
- Clarify the semantics of our rewriting system

2010: The Odyssey continues...

Currently:

- Book: “Kripke’s World”
Authors: [Gasquet, Herzig, Said, Schwarzenruber]

Next events:

- Universal Logic 2010, April - Lisbon (tutorial)
- ESSLLI 2010, August - Copenhagen, (1 week course)

What about?

- Language extension: SQL-queries?
- Performance: backtracking?
- Generic interface with other tools: will be user-friendly?
- Converse: what about CPDL?
- New methods for new logics. . .

Merci!