Labelled Modal Tableaux

Guido Governatori[1]

National ICT Australia (NICTA)

Advances in Modal Logic, Nancy, 10 September 2008

[1] This work was supported by the Australia Research Council under Discovery Project No. DP0452628 on "Combining modal logic for dynamic and multi-agents systems".

Overview

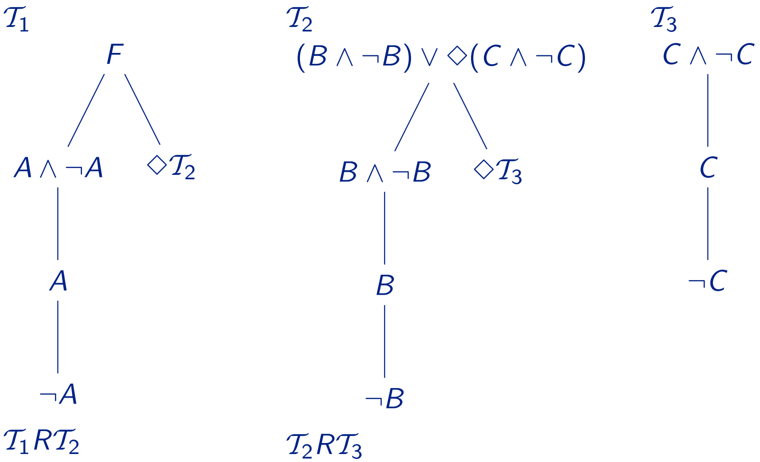
- Labelled Tableaux
- Relative Complexity

Tableaux

- Semantic tableaux method is a refutation proof method
- We start from the negation of the formula we want to prove, and systematically we try to build a model/countermodel according to inference rules that correspond to the semantic evaluation clauses
- A tableaux is a binary tree
- When we discover a contradiction we close the branch
- A proof of A is a closed tree for $\neg A$

Modal Tableaux

$$F = (A \land \neg A) \lor \Diamond((B \land \neg B) \lor \Diamond(C \land \neg C))$$



Tableaux vs Modal Tableaux

Labelled Modal Tableaux

- forest of classical trees
- labels to simulate the possible world structure
  Labelled Modal Tableaux
  - $A : i$
  - ground labels:
    (w_3, w_2, w_1);
  - free variable labels:
    (W_3, w_2, W_1, w_1).
    Free Variable Labelled Modal Tableaux
    - propagation based (Single Step Tableaux)
    - unification based (KEM)

How to chose your tableaux system?

1. Relative complexity

How to chose your tableaux system?

1. Relative complexity
2. Expressive power, logic coverage, extensibility

How to chose your tableaux system?

1. Relative complexity
2. Expressive power, logic coverage, extensibility
3. It is your pet system

How to chose your tableaux system?

1. Relative complexity
2. Expressive power, logic coverage, extensibility
3. It is your pet system
4. You want to spend the next two years to develop a yet another labelled tableaux

How to chose your tableaux system?

1. Relative complexity
2. Expressive power, logic coverage, extensibility
3. It is your pet system
4. You want to spend the next two years to develop a yet another labelled tableaux
5. You don't understand the other systems

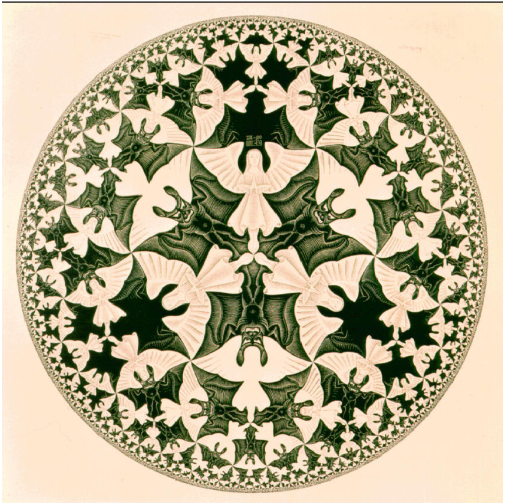
Comparing Two Tableaux Systems

- Experimental comparison
  - optimisation of the implementation
  - programming skills
- Theoretical comparison
  - identify the appropriate logic
  - identify the appropriate class of formulas
  - identify the appropriate comparison measure

Relative Complexity: Methodology

- Logic: modular, and no “optimisations” can be/have been devised;
- Formulas: they should challenge only the modal features and not the propositional ones;
- Comparison Measure: it should be appropriate for the problem at hand.

The Beauty of Symmetry



The Quest for Beauty

- Logic: modular, and no "optimisations" can be/have been devised: DB.
- Formulas: they should challenge only the modal features and not the propositional ones: $p \rightarrow (\Box\Diamond)^n p$.

p-simulation

Definition

A proof system $\mathcal{A}$ p-simulates a proof system $\mathcal{B}$ iff there is a function g, computable in polynomial time, which maps proofs in $\mathcal{B}$ for any given formula ϕ, to proofs in $\mathcal{A}$ for ϕ. (Cook & Reckhow 1979)

p-simulation

Definition

A proof system $\mathcal{A}$ p-simulates a proof system $\mathcal{B}$ iff there is a function g, computable in polynomial time, which maps proofs in $\mathcal{B}$ for any given formula ϕ, to proofs in $\mathcal{A}$ for ϕ. (Cook & Reckhow 1979)

- Semi-decidable systems;
- can compare only derivations of theorems;
- Non-deterministic notion;
- Not exhaustive searches.

p-search-simulation

Definition

A system $\mathcal{A}$ p-search-simulates a system $\mathcal{B}$ iff there is a polynomial function g such that for any formula ϕ, g maps derivations (trees) from ϕ in $\mathcal{A}$ to derivations (trees) from ϕ in $\mathcal{B}$ (de Nivelle, Schmidt & Hustadt 2000).

p-search-simulation

Definition

A system $\mathcal{A}$ p-search-simulates a system $\mathcal{B}$ iff there is a polynomial function g such that for any formula ϕ, g maps derivations (trees) from ϕ in $\mathcal{A}$ to derivations (trees) from ϕ in $\mathcal{B}$ (de Nivelle, Schmidt & Hustadt 2000).

- We can compare derivations for generic formulas;
- It requires exhaustive proof search procedures.

Label Formalism

$$i = (\overbrace{W_2}^{h(i)}, \overbrace{(\underbrace{w_3}_{h(b(i))}, \underbrace{(w_2, (W_1, w_1))}_{b(b(i))})}^{b(i)})$$

each $b(\ldots b(i)) = s(i)$
$s^3(i) = (w_2, (W_1, w_1))$
$h^3(i) = w_2$
$c^3(i) = (W_2, (w_3, w_0))$
where $w_0 = (w_2, (W_1, w_1))$

Single Step Tableaux (SST)

Massacci (1994,2000), Beckert and Gorè (2001)

$$\frac{\pi : i}{\pi_0 : w_n, i}\pi \qquad \frac{\Diamond p : W_2, w_0}{p : w_3, W_2, w_0}$$

$$\frac{\nu : i}{\nu_0 : W_n, i}\nu_{\mathrm{D}} \qquad \frac{\Box p : w_1, w_0}{p : W_1, w_1, w_0}$$

$$\frac{\nu : i}{\nu_0 : b(i)}\nu_{\mathrm{B}} \qquad \frac{\Box p : w_1, w_0}{p : w_0}$$

$$\frac{\begin{array}{c} X : i \\ \neg X : j \end{array}}{\times}\; i, j \text{ unify}$$

KEM

$$\frac{\pi : i}{\pi_0 : w_n, i} \qquad \frac{\nu : i}{\nu_0 : W_n, i}$$

$$\frac{\begin{array}{c} X : i \\ \neg X : j \end{array}}{\times} \, i, j \ \sigma_{DB}\text{-unify} \qquad \frac{}{A : i \qquad \neg A : i}$$

- world-unification: two world symbols;
- basic-unification (σ): two labels stepwise;
- axiom-unification (σ^{D}, σ^{B}): two labels axiom-wise;
- logic-unification σ_{DB}: two labels recursively

Basic Unification

- Two atomic labels unify iff
  - they are the same constant
  - one of them is a variable
- Two labels i and k unify iff
  - they have the same length
  - for each n, $h^n(i)$ and $h^n(k)$ unify

$$W_2, w_2, w_1$$
$$w_3, W_1, w_1$$

Proposition

The basic unification of two labels can be computed in linear time.

B-morphism

$$\Phi_C^{i,n} = \{s^m(i) : m > n \text{ and } h^m(i) \in \Phi_C\}$$
$$\Phi_V^{i,n} = \{s^m(i) : m > n \text{ and } h^m(i) \in \Phi_V\}$$

Definition

Given a label i and an integer n, we will say that i has the *bmorphism property for n* iff there is a morphism $\theta : \Phi_C^{i,n} \mapsto \Phi_V^{i,n}$ such that

1. θ is injective, and
2. if $\theta(s^k(i)) = s^l(i)$, then $k < l$.

The meaning of bmorphism is that every constant can be consumed by a variable coming after it.

B-unification

Two labels σ^B-unify iff

- their distance is $2n$
- the segments of the same length σ-unify
- the longest label has the bmorphism property for m, where m is the length of the shortest segment.

$$W_2, w_2, w_1$$
$$w_1$$

$$W_4, W_3, w_3, W_2, w_2, w_1$$
$$W_1, w_1$$

Bmorphism Algorithm

Bmorphism(i, n)

$$
\begin{array}{l}
Bcount := 0 \\
\quad \textbf{for } x \textbf{ from } \ell(i) \textbf{ to } n+1 \\
\qquad \textbf{if } Bcount \geq 0 \\
\qquad\quad \textbf{then if } h^x(i) \in \Phi_V \\
\qquad\qquad \textbf{then } Bcount := Bcount + 1 \\
\qquad\qquad \textbf{else } Bcount := Bcount - 1 \\
\quad \textbf{return } Bcount
\end{array}
$$

$$i = (W_3, (w_4, (W_2, (W_1, (w_3, (w_2, W_1))))))$$

The *Bmorphism* function returns the following values for n from 7 to 1.

$i, 7 \mapsto 0,\ i, 6 \mapsto 1,\ i, 5 \mapsto 0,\ i, 4 \mapsto 1,\ i, 3 \mapsto 2,\ i, 2 \mapsto 1,\ i, 1 \mapsto 0$

Complexity of B-unification

Proposition

Let i be a label and n a positive integer such that $n < \ell(i)$. Bmorphism$(i, n) \geq 0$ iff i has the bmorphism property for n.

Proposition

The σ^B-unification of two labels can be computed in linear time.

DB Unification

$$[i;j]\sigma_{DB} = \begin{cases} [i;j]\sigma^{DB} \\ [c^n(i);c^m(j)]\sigma^{DB}, [s^n(i);s^m(j)]\sigma_{DB} \end{cases}$$

$$\frac{W_1, w_2, w_1}{W_2, w_3, w_1}$$

since

$$\frac{W_1, w_2, w_0}{w_0} \qquad w_0 = \frac{w_1}{W_2, w_3, w_1}$$

Complexity of DB-Unification

Proposition

The σ_{DB} unification of an atomic label and a label can be computed in polynomial (quadratic) time.

$$
\begin{aligned}
c^n(j) &= w_0 & s^n(j) &= j\\
c^{n-1}(j) &= (h^n(j), w_0) & s^{n-1}(j) &= b(j)\\
c^{n-2}(j) &= (h^n(j), (h^{n-1}(j), w_0)) & s^{n-2}(j) &= b(b(j))\\
&\vdots & &\vdots
\end{aligned}
$$

and

$$
\begin{aligned}
c^{n-1}(c^{n-1}(j)) &= w_0 & s^{n-1}(s^{n-1}(j)) &= b(j)\\
c^{n-2}(c^{n-1}(j)) &= (h^{n-1}(j), w_0) & s^{n-2}(s^{n-1}(j)) &= b(b(j))\\
&\vdots & &\vdots
\end{aligned}
$$

KEM vs SST

Proposition

The length of the proof of $p \longrightarrow (\Box\Diamond)^n p$ in KEM is $O(n^2)$.

Proposition

The length of the proof of $p \longrightarrow (\Box\Diamond)^n p$ in SST is $O(2^{n+1})$.

1.	$\neg(p \longrightarrow (\Box\Diamond)^n p) : w_1$
2.	$p : w_1$
3.	$\neg(\Box\Diamond)^n p : w_1$
4.	$\neg\Diamond(\Box\Diamond)^{n-1} p : w_2, w_1$
5.	$\neg(\Box\Diamond)^{n-1} p : W_1, w_2, w_1$
	$\vdots$
$2n+3$	$\neg p : W_n, w_{n+1}, \ldots, W_1, w_2, w_1$

1.	$\neg(p \longrightarrow (\Box\Diamond)^n p) : w_1$
2.	$p : w_1$
3.	$\neg(\Box\Diamond)^n p : w_1$
4.	$\neg\Diamond(\Box\Diamond)^{n-1} p : w_2, w_1$
5.	$\neg(\Box\Diamond)^{n-1} p : W_1, w_2, w_1$
6.	$\neg(\Box\Diamond)^{n-1} p : w_1$
7.	$\neg\Diamond(\Box\Diamond)^{n-2} p : w_3, W_1, w_2, w_1$
8.	$\neg\Diamond(\Box\Diamond)^{n-2} p : w_3, w_1$
9.	$\neg(\Box\Diamond)^{n-2} p : W_2, w_3, W_1, w_2, w_1$
10.	$\neg(\Box\Diamond)^{n-2} p : w_2, w_1$
11.	$\neg(\Box\Diamond)^{n-2} p : W_2, w_3, w_1$
12.	$\neg(\Box\Diamond)^{n-2} p : w_1$

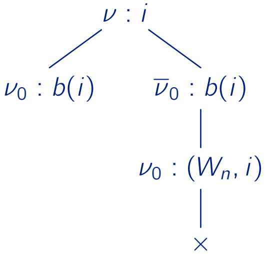
and the winner is ...

Theorem

SST cannot p-search-simulate KEM.

Lemma

The rule ν_B is a derived rule in KEM, and it can be derived in polynomial time.



Theorem

KEM p-search-simulates SST.

The longest journey begins with a single step

Does it matter?

Experimental results

- KEM faster than SST from $p \longrightarrow (\Box\Diamond)^3 p$.

Does it matter?

Experimental results

- KEM faster than SST from $p \rightarrow (\Box\Diamond)^3 p$.
- One order of magnitude faster than SST for $n = 5$.

Does it matter?

Experimental results

- KEM faster than SST from $p \rightarrow (\Box\Diamond)^3 p$.
- One order of magnitude faster than SST for $n = 5$.
- Two orders of magnitude for $n = 7$.

Does it matter?

Experimental results

- KEM faster than SST from $p \to (\Box\Diamond)^3 p$.
- One order of magnitude faster than SST for $n = 5$.
- Two orders of magnitude for $n = 7$.
- SST faster for formulas like $\Diamond\Diamond\Diamond\Box\Box\Box\Diamond p \wedge \Diamond\Diamond\Box\Diamond\Box\neg p$.

Does it matter?

Experimental results

- KEM faster than SST from $p \to (\Box\Diamond)^3 p$.
- One order of magnitude faster than SST for $n = 5$.
- Two orders of magnitude for $n = 7$.
- SST faster for formulas like $\Diamond\Diamond\Diamond\Box\Box\Box\Diamond p \wedge \Diamond\Diamond\Box\Diamond\Box\neg p$.

Does it matter?

Experimental results

- KEM faster than SST from $p \to (\Box\Diamond)^3 p$.
- One order of magnitude faster than SST for $n = 5$.
- Two orders of magnitude for $n = 7$.
- SST faster for formulas like $\Diamond\Diamond\Diamond\Box\Box\Box\Diamond p \land \Diamond\Diamond\Box\Diamond\Box\neg p$.

Challenge

Free beer to the first 5 who can give plausible and natural readings of $p \to (\Box\Diamond)^7 p$ *and* $\Diamond\Diamond\Diamond\Box\Box\Box\Diamond p \land \Diamond\Diamond\Box\Diamond\Box\neg p$

Beyond Standard Modal Logic I

- KEM covers all 15 basic modal logics (Artosi, Benassi, Governatori, and Rotolo 1998).

Beyond Standard Modal Logic I

- KEM covers all 15 basic modal logics (Artosi, Benassi, Governatori, and Rotolo 1998).
- Easily extended to multi-modal logics (Governatori 1995, Gabbay and Governatori 2000).

Beyond Standard Modal Logic I

- KEM covers all 15 basic modal logics (Artosi, Benassi, Governatori, and Rotolo 1998).
- Easily extended to multi-modal logics (Governatori 1995, Gabbay and Governatori 2000).
  - Immediate without interaction axioms

Beyond Standard Modal Logic I

- KEM covers all 15 basic modal logics (Artosi, Benassi, Governatori, and Rotolo 1998).
- Easily extended to multi-modal logics (Governatori 1995, Gabbay and Governatori 2000).
  - Immediate without interaction axioms
  - Straighforward with inclusion axioms: subsitution for variables of appropriate types ranges over few types of constants.

Beyond Standard Modal Logic I

- KEM covers all 15 basic modal logics (Artosi, Benassi, Governatori, and Rotolo 1998).
- Easily extended to multi-modal logics (Governatori 1995, Gabbay and Governatori 2000).
  - Immediate without interaction axioms
  - Straighforward with inclusion axioms: subsitution for variables of appropriate types ranges over few types of constants.
- Non normal modal logic: adding constraints on substitution of variables: substitution injective on a tree for regular logics, injective on a branch for monotonic logics (does not hold for classical logic, loose interpreation of the labels). Constants are mapped to themselves only if they are the results of a unification (Governatori and Luppi 2000).

Beyond Standard Modal Logic II

- conditional logics: $A > B$ undesrstood as $[A]B$. Labels indexed by formulas (conditions on formulas on top on strutural conditions on labels for unifications). Thus the label $W_1^{A \wedge B}, w_1$ represent the selection function of w_1 for the proposition $A \wedge B$. (Artosi and Governatori 1998, Artosi, Governatori and Rotolo 2002).

Beyond Standard Modal Logic II

- conditional logics: $A > B$ undesrstood as $[A]B$. Labels indexed by formulas (conditions on formulas on top on strutural conditions on labels for unifications). Thus the label $W_1^{A\wedge B}, w_1$ represent the selection function of w_1 for the proposition $A \wedge B$. (Artosi and Governatori 1998, Artosi, Governatori and Rotolo 2002).
- Exotic logics (Jones Pörn 1985): labels are first class citizens. Inference conditions for them (Artosi, Governatori and Sartor 1996, Governatori 1996, 1997).

$$\text{Exc}\frac{A : (W^s, i) \quad A : (W^d, j)}{A : (W, [i, j]\sigma_{\mathrm{JP}})} \qquad \text{LPNC}\frac{A : i^s \quad B : i^d}{\times}$$

$$\text{LPB}\frac{}{A : i^s \mid B : i^d}\; i \text{ restricted} \qquad \frac{\Box A : i \quad \neg A : j}{\Box A : k^s \quad \neg A : k^s}[i, j]\sigma_{\mathrm{JP}} = k$$

So how do you choose your tableaux system?

1. Relative complexity
2. Expressive power, logic coverage, extensibility
3. It is your pet system
4. You want to spend the next two years to develop a yet another labelled tableaux
5. You don't understand the other systems

Questions?

References

A. Artosi, P. Benassi, G. Governatori, and A. Rotolo (1998).
Shakespearian modal logic: A labelled treatment of modal identity.
In *Advances in Modal Logic. Volume 1*, 1–21. CSLI.

A. Artosi and G. Governatori (1998).
A tableaux methodology for deontic conditional logics.
In *Proc. ΔEON'98*, 65–81.

A. Artosi, G. Governatori, and A. Rotolo (2002).
Labelled tableaux for non-monotonic reasoning: Cumulative consequence relations.
Journal of Logic and Computation, 12(6):1027–1060.

A. Artosi, G. Governatori, and G. Sartor (1996).
Towards a computational treatment of deontic defeasibility.
In *Deontic Logic Agency and Normative Systems*, 27–46. Springer.

B. Beckert and R. Goré (2001).
Free variable tableaux for propositional modal logics.
Studia Logica, 69(1):59–96.

S.A. Cook and R.A. Reckhow (1979).
The relative efficiency of propositional proof systems.
Journal of Symbolic Logic, 44:36–50.

H. de Nivelle, R. Schmidt, and U. Hustadt (2000).
Resolution-based methods for modal logics.
Logic Journal of IGPL, 8:265–292.

D.M. Gabbay and G. Governatori (2000).
Fibred modal tableaux.
In *Labelled Deduction*, 163–194. Kluwer.

G. Governatori (1995).
Labelled tableaux for multi-modal logics.
In *Proc. TABLEAUX'95*, LNAI 918, 79–94, Springer.

G. Governatori (1996).
Labelling ideality and subideality.
In *Proc. Practical Reasoning*, LNAI 1085, 291–304, Springer.

G. Governatori (1997).
Un modello formale per il ragionamento giuridico.
PhD thesis, CIRFID, University of Bologna.

G. Governatori and A. Luppi (2000).
Labelled tableaux for non-normal modal logics.
In *Proc. AI*IA 99*, LNAI 1792, 119–130, Springer-Verlag.

U. Hustadt and R. Schmidt (1997).
On evaluating decision procedures for modal logic.
In *Proc. IJCAI'97*, 202–207.

A.J.I. Jones and I. Pörn (1985).
Ideality, sub-ideality and deontic logic.
Synthese, 65:275–290.

F. Massacci (1994).
Strongly analytic tableaux for normal modal logic.
In *Proc. CADE-12*, LNAI 814, 723–737, Springer.

F. Massacci (2000).
Single step tableaux for modal logic.
Journal of Automated Reasoning, 24(3):319–364.