On the Relative Complexity of 2 Labelled Modal Tableaux

Guido Governatori
School of Information Technology & Electrical Engineering
The University of Queensland
guido@itee.uq.edu.au

Overview

- Motivation
- Modal Logic
- Labelled Modal Tableaux
- Relative Complexity
- KEM vs SST
- Conclusion

Motivation

- 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

Modal Logic

$$\langle W, R, v \rangle$$

- $w \models \Box A$ iff $\forall x : wRx, x \models A$
- $w \models \Diamond A$ iff $\exists x : wRx, x \models A$
- $\Box A \equiv \neg\Diamond\neg A$

Axiom	**Condition on** R
K: $\Box(A \to B) \to (\Box A \to \Box B)$	no conditions
D: $\Box A \to \Diamond A$	serial, $\forall x \exists y (xRy)$
T: $\Box A \to A$	reflexive, $\forall x (xRx)$
4: $\Box A \to \Box\Box A$	transitive, $\forall x, y, z (xRy \wedge yRz \to xRz)$
B: $A \to \Box\Diamond A$	symmetric, $\forall x, y (xRy \to yRx)$
5: $\Diamond A \to \Box\Diamond A$	euclidean, $\forall x, y, z (xRy \wedge xRz \to yRz)$

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

- 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)

Methodology

- Logic: modular, and no “optimisations” can be/have been devised; DB.
- Formulas: they should challenge only the modal features and not the propositional ones; $p \rightarrow (\Box\Diamond)^n p$.
- Comparison Measure: it should be appropriate for the problem at hand.

p-simulation

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

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

p-search-simulation

A system $\mathscr{A}$ p-search-simulates a system $\mathscr{B}$ iff there is a polynomial function g such that for any formula ϕ, g maps derivations (trees) from ϕ in $\mathscr{A}$ to derivations (trees) from ϕ in $\mathscr{B}$ (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 Goré (2001)

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

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

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

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

KEM

$$\frac{\pi : i}{\pi_0 : w_n, i}$$

$$\frac{\nu : i}{\nu_0 : W_n, i}$$

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

- 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

$$\begin{array}{c} W_2, w_2, w_1 \\ w_3, W_1, w_1 \end{array}$$

Proposition 1 *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 1 *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.

$$\begin{array}{r} W_2, w_2, w_1 \\ w_1 \end{array}$$

$$\begin{array}{r} W_4, W_3, w_3, W_2, w_2, w_1 \\ W_1, w_1 \end{array}$$

Bmorphism Algorithm

```
$Bmorphism(i, n)$
        $Bcount := 0$
          for $x$ from $\ell(i)$ to $n + 1$
              if $Bcount \geq 0$
                 then if $h^x(i) \in \Phi_V$
                        then $Bcount := Bcount + 1$
                        else $Bcount := Bcount - 1$
          return $Bcount$
```

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

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 2 *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 3 *The σ^B-unification of two labels can be computed in linear time.*

DB Unification

$$[i;j]\sigma_{DB} = \left\{ \begin{array}{l} [i;j]\sigma^{DB} \\ {[c^n(i);c^m(j)]\sigma^{DB}, [s^n(i);s^m(j)]\sigma_{DB}} \end{array} \right.$$

$$\begin{array}{l} W_1, w_2, w_1 \\ W_2, w_3, w_1 \end{array}$$

since

$$\begin{array}{r} W_1, w_2, w_0 \\ w_0 \end{array} \qquad w_0 = \begin{array}{r} w_1 \\ W_2, w_3, w_1 \end{array}$$

Complexity of DB-Unification

Proposition 4 *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 5 *The length of the proof of $p \rightarrow (\Box\Diamond)^n p$ in KEM is $O(n^2)$.*

$$\begin{array}{rl}
1. & \neg(p \rightarrow (\Box\Diamond)^n p) : w_1 \\
2. & p : w_1 \\
3. & \neg(\Box\Diamond)^n p : w_1 \\
4. & \neg\Diamond(\Box\Diamond)^{n-1} p : w_2, w_1 \\
5. & \neg(\Box\Diamond)^{n-1} p : W_1, w_2, w_1 \\
 & \vdots \\
2n+3 & \neg p : W_n, w_{n+1}, \dots, W_1, w_2, w_1
\end{array}$$

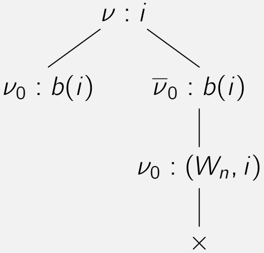
Proposition 6 *The length of the proof of $p \rightarrow (\Box\Diamond)^n p$ in SST is $O(2^{n+1})$.*

$$\begin{array}{rll}
1. & \neg(p \rightarrow (\Box\Diamond)^n p) : w_1 & \\
2. & p : w_1 & \\
3. & \neg(\Box\Diamond)^n p : w_1 & \\
4. & \neg\Diamond(\Box\Diamond)^{n-1} p : w_2, w_1 & \\
5. & \neg(\Box\Diamond)^{n-1} p : W_1, w_2, w_1 & 4\nu_D \\
6. & \neg(\Box\Diamond)^{n-1} p : w_1 & 4\nu_B \\
7. & \neg\Diamond(\Box\Diamond)^{n-2} p : w_3, W_1, w_2, w_1 & \\
8. & \neg\Diamond(\Box\Diamond)^{n-2} p : w_3, w_1 & \\
9. & \neg(\Box\Diamond)^{n-2} p : W_2, w_3, W_1, w_2, w_1 & 7\nu_D \\
10. & \neg(\Box\Diamond)^{n-2} p : w_2, w_1 & 7\nu_B \\
11. & \neg(\Box\Diamond)^{n-2} p : W_2, w_3, w_1 & 8\nu_D \\
12. & \neg(\Box\Diamond)^{n-2} p : w_1 & 8\nu_B \\
 & \vdots &
\end{array}$$

and the winner is . . .

Theorem 7 *SST cannot p-search-simulate KEM.*

Lemma 8 *The rule ν_B is a derived rule in KEM, and it can be derived in polynomial time.*



Theorem 9 *KEM p-search-simulates SST.*