Tableau Development for a Bi-Intuitionistic Tense Logic

John G. Stell   Renate A. Schmidt
David Rydeheard

RAMiCS 14, Marienstatt im Westerwald
30 April 2014

At RAMiCS 2012 I talked about relations on graphs.

Algebra of all relations on given graph is not a relation algebra.

No converse, but an adjoint pair: left and right converse.

Today: these relations as accessibility relations for a modal logic.

Gives a semantics for a bi-intuitionistic modal logic where

$$\Diamond\alpha \leftrightarrow \lrcorner\Box\neg\alpha$$

Provides case study for development of a tableau calculus

Semantics for classical tense logic

Language: Propositional variables, $\vee$, $\wedge$, $-$, $\rightarrow$, $\top$, $\bot$

$\Diamond$ (sometime in future), $\blacklozenge$ (sometime in past), $\Box$, $\blacksquare$.

Frame: Set U with relation $R \subseteq U \times U$

Valuation assigns a subset $[\![p]\!] \subseteq U$ to each variable,

extended to all formulas by

$$[\![\alpha \vee \beta]\!] = [\![\alpha]\!] \cup [\![\beta]\!]$$

$$[\![-\alpha]\!] = -[\![\alpha]\!]$$

etc.

The modalities are expressed in terms of $\ominus$ and $\oplus$, the erosion and dilation operations on subsets:

The **dilation**, $\oplus$, and the **erosion**, $\ominus$, are given by:

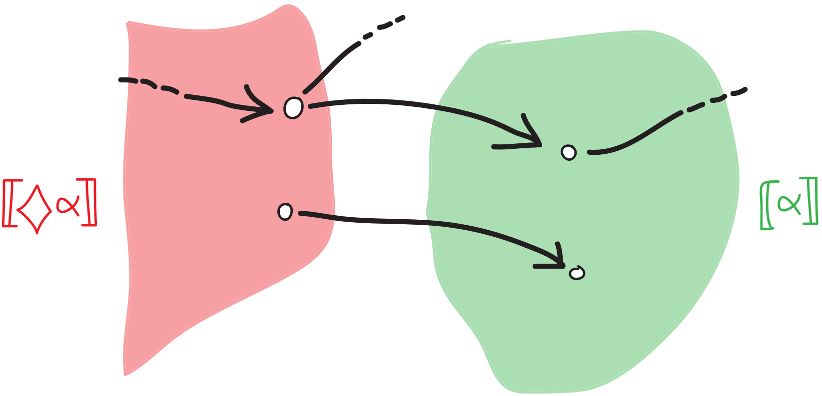
$$X \oplus R \quad = \quad \{u \in U : \exists x\,((x, u) \in R \wedge x \in X)\}$$

Places accessible from X

$$R \ominus X \quad = \quad \{u \in U : \forall x\,((u, x) \in R \rightarrow x \in X)\}$$

Places from where only X is accessible

Semantics for classical tense logic



$$[\![\Diamond\alpha]\!] = [\![\alpha]\!] \oplus \breve{R}$$

Semantics for classical tense logic

$$[\![\Box \alpha]\!] = R \ominus [\![\alpha]\!] \qquad [\![\Diamond \alpha]\!] = [\![\alpha]\!] \oplus \breve{R}$$

$$[\![\blacklozenge \alpha]\!] = [\![\alpha]\!] \oplus R \qquad [\![\blacksquare \alpha]\!] = \breve{R} \ominus [\![\alpha]\!]$$

Generalize from sets to graphs

Imagine a tense logic

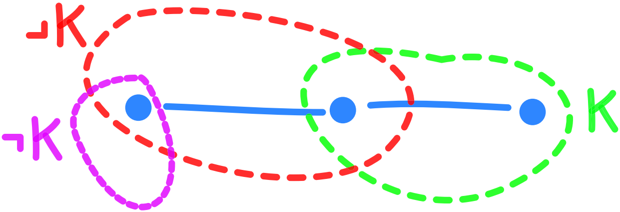
where U is a graph,

where $[\![\alpha]\!]$ is a subgraph,

and where R is a relation on the graph.

Why? ...

The subgraphs of a graph form a bi-Heyting algebra so this a natural semantics for a bi-intuitionistic logic.

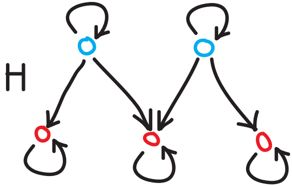


$$\top \to p \lor \neg p$$ ✗

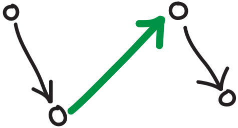
$$p \land \neg p \to \bot$$ ✗

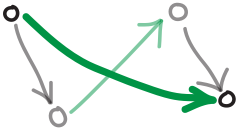
Generalize from graphs to pre-orders

Pre-order H describes incidence structure of graph

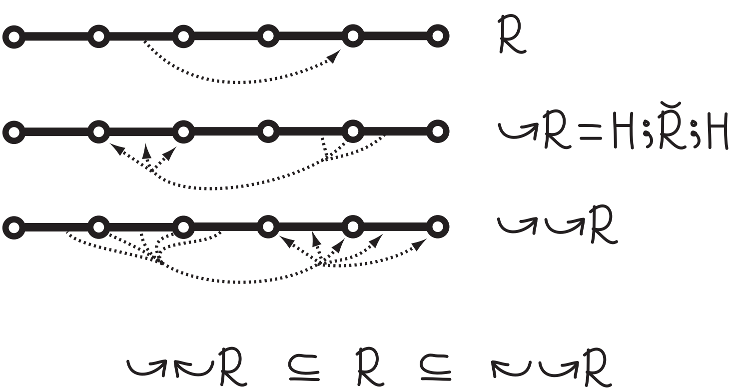


For any pre-order $H \subseteq U \times U$ a relation $R \subseteq U \times U$ is **stable** if $R = H \,;\, R \,;\, H$





Converses



Semantics for BISKT

$$
\begin{array}{rclrcl}
[\![\bot]\!] & = & \varnothing & [\![\top]\!] & = & U \\
[\![\alpha \vee \beta]\!] & = & [\![\alpha]\!] \cup [\![\beta]\!] & [\![\alpha \wedge \beta]\!] & = & [\![\alpha]\!] \cap [\![\beta]\!] \\
[\![\neg\alpha]\!] & = & \neg[\![\alpha]\!] & [\![\lrcorner\alpha]\!] & = & \lrcorner[\![\alpha]\!] \\
[\![\alpha \to \beta]\!] & = & [\![\alpha]\!] \to [\![\beta]\!] & [\![\alpha \succ \beta]\!] & = & [\![\alpha]\!] \succ [\![\beta]\!] \\
[\![\Box\alpha]\!] & = & R \ominus [\![\alpha]\!] & [\![\Diamond\alpha]\!] & = & [\![\alpha]\!] \oplus (\smile R) \\
[\![\blacklozenge\alpha]\!] & = & [\![\alpha]\!] \oplus R & [\![\blacksquare\alpha]\!] & = & (\smile R) \ominus [\![\alpha]\!]
\end{array}
$$

Relationship of boxes to diamonds

$$\Diamond \alpha \longleftrightarrow \lrcorner \Box \neg \alpha$$

$$\neg \blacklozenge \lrcorner \alpha \longleftrightarrow \blacksquare \alpha$$

BISKT and BIKT

Goré, Postneice and Tiu (2010) have a bi-intuitionistic tense logic BIKT where the frames have two independent accessibility relations, both stable.

They have no relationship between boxes and diamonds.

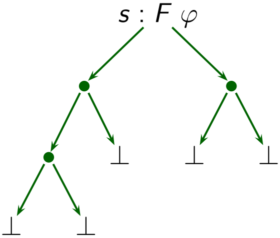
The left converse allows us to connect these two relations,

and allows us to express $\Diamond$ in terms of $\Box$ and $\blacksquare$ in terms of $\blacklozenge$.

For some applications it seems likely that time looking forwards and time looking backward should be connected.

Semantic labelled tableau

- Aim: construct a counter-model or prove given formula(e)
- Goal-directed, top-down
- Branching rules $\rightsquigarrow$ derivations are trees



$$\frac{s : T\ \alpha \wedge \beta}{s : T\ \alpha,\ s : T\ \beta} \qquad \frac{s : F\ \alpha \wedge \beta}{s : F\ \alpha \mid s : F\ \beta}$$

$$\frac{s : T\ \neg\alpha,\ H(s,t)}{t : F\ \alpha}$$

$$\frac{s : F\ \neg\alpha}{H(s,u),\ u : T\ \alpha}\ (u \text{ new})$$

etc

Tableau calculus for BISKT: Rules for bi-intuitionistic logic

$$\frac{s : T\ \alpha,\ s : F\ \alpha}{\bot} \qquad \frac{s : T\ \bot}{\bot}$$

$$\frac{s : T\ \alpha \wedge \beta}{s : T\ \alpha,\ s : T\ \beta} \qquad \frac{s : F\ \alpha \wedge \beta}{s : F\ \alpha \mid s : F\ \beta}$$

$$\frac{s : F\ \alpha \vee \beta}{s : F\ \alpha,\ s : F\ \beta} \qquad \frac{s : T\ \alpha \vee \beta}{s : T\ \alpha \mid s : T\ \beta}$$

$$\frac{s : T\ \neg\alpha,\ H(s,t)}{t : F\ \alpha} \qquad \frac{s : F\ \neg\alpha}{H(s, f_{\neg\alpha}(s)),\ f_{\neg\alpha}(s) : T\ \alpha}$$

$$\frac{s : F\ \lrcorner\alpha,\ H(t,s)}{t : T\ \alpha} \qquad \frac{s : T\ \lrcorner\alpha}{H(f_{\lrcorner\alpha}(s), s),\ f_{\lrcorner\alpha}(s) : F\ \alpha}$$

$$\frac{s : T\ \alpha \to \beta,\ H(s,t)}{t : F\ \alpha \mid t : T\ \beta} \qquad \frac{s : F\ \alpha \to \beta}{H(s, f_{\alpha\to\beta}(s)),\ f_{\alpha\to\beta}(s) : T\ \alpha,\ f_{\alpha\to\beta}(s) : F\ \beta}$$

$$\frac{s : F\ \alpha \succ \beta,\ H(t,s)}{t : F\ \alpha \mid t : T\ \beta} \qquad \frac{s : T\ \alpha \succ \beta}{H(f_{\alpha\succ\beta}(s), s),\ f_{\alpha\succ\beta}(s) : T\ \alpha,\ f_{\alpha\succ\beta}(s) : F\ \beta}$$

Rules for the tense operators

$$\frac{s : T\ \Box\alpha,\ R(s,t)}{t : T\ \alpha} \qquad \frac{s : F\ \Box\alpha}{R(s, f_{\Box\alpha}(s)),\ f_{\Box\alpha}(s) : F\ \alpha}$$

$$\frac{s : F\ \blacklozenge\alpha,\ R(t,s)}{t : F\ \alpha} \qquad \frac{s : T\ \blacklozenge\alpha}{R(f_{\blacklozenge\alpha}(s), s),\ f_{\blacklozenge\alpha}(s) : T\ \alpha}$$

$$\frac{s : F\ \Diamond\alpha,\ H(t,s),\ R(t,u),\ H(v,u)}{v : F\ \alpha}$$

$$\frac{s : T\ \Diamond\alpha}{H(g_{\Diamond\alpha}(s), s),\ R(g_{\Diamond\alpha}(s), g'_{\Diamond\alpha}(s)),\ H(f_{\Diamond\alpha}(s), g'_{\Diamond\alpha}(s)),\ f_{\Diamond\alpha}(s) : T\ \alpha}$$

$$\frac{s : T\ \blacksquare\alpha,\ H(s,t),\ R(u,t),\ H(u,v)}{v : T\ \alpha}$$

$$\frac{s : F\ \blacksquare\alpha}{H(s, g_{\blacksquare\alpha}(s)),\ R(g'_{\blacksquare\alpha}(s), g_{\blacksquare\alpha}(s)),\ H(g'_{\blacksquare\alpha}(s), f_{\blacksquare\alpha}(s)),\ f_{\blacksquare\alpha}(s) : F\ \alpha}$$

Rules for frame/model conditions and blocking

$$\text{refl: } \frac{}{H(s,s)} \qquad \text{tr: } \frac{H(s,t),\;\; H(t,u)}{H(s,u)}$$

$$\text{mon: } \frac{s : T\,\alpha,\;\; H(s,t)}{t : T\,\alpha} \qquad \text{stab: } \frac{H(s,t),\;\; R(t,u),\;\; H(u,v)}{R(s,v)}$$

$$\text{ub: } \frac{}{s \approx t \;\mid\; s \not\approx t}$$

$$\frac{s \not\approx s}{\bot} \qquad \frac{s \approx t}{t \approx s} \qquad \frac{s \approx t,\;\; G[s]_\lambda}{G[\lambda/t]}$$

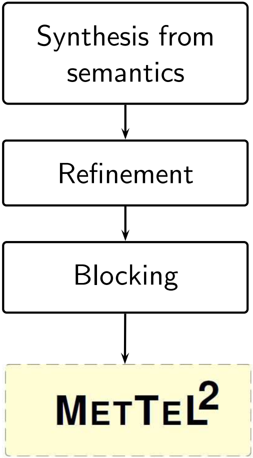
Tableau development process

Define a sound and complete calculus

Making tableau calculus effective

Ensure termination for decidable logics

Generate a prover



Joint work with Dmitry Tishkovsky and Mohammad Khodadadi, 2007–14

Synthesis from semantics and rule refinement

Idea

Transformation of definition of semantics into Skolemised conjunctive normal form + rule refinement

- $s \in [\![\neg\alpha]\!] \quad \text{iff} \quad \forall x\,((s,x) \in H \to x \notin [\![\alpha]\!])$

$$\rightsquigarrow \quad \frac{s : T\ \neg\alpha,\ \ H(s,t)}{t : F\ \alpha} \qquad \frac{s : F\ \neg\alpha}{H(s, f_{\neg\alpha}(s)),\ \ f_{\neg\alpha}(s) : T\ \alpha}$$

$f_{\neg\alpha}(s)$ is witness for successor

- $H\,;R\,;H \subseteq R \quad \rightsquigarrow \quad \dfrac{H(s,t),\ \ R(t,u),\ \ H(u,v)}{R(s,v)}$

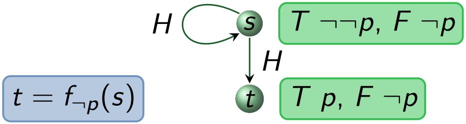
Blocking

General idea

Use the tableau calculus to find finite models by identifying labels

Unrestricted blocking

$$\text{(ub)} \quad \frac{}{s \approx t \;\mid\; s \not\approx t}$$



Assume $s \approx t$
Rewrite $t \rightarrow s$

$s : T\ \neg\neg p$
$H(s, s)$
$s : F\ \neg p$
$H(s, t)$
$t : T\ p$
$t : F\ \neg p$
$s \approx t$

Blocking (cont'd)

General idea

Use the tableau calculus to find finite models by identifying labels

Unrestricted blocking

$$\text{(ub)} \qquad \frac{}{s \approx t \mid s \not\approx t}$$

$T\ \neg\neg p,\ F\ \neg p,\ T\ p$

$t = f_{\neg p}(s)$

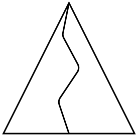
$s \approx t$

$s : T\ \neg\neg p$
$H(s, s)$
$s : F\ \neg p$
~~$H(s, t)$~~ ~~$H(s, s)$~~
~~$t : T\ p$~~ $s : T\ p$
~~$t : F\ \neg p$~~ ~~$s : F\ \neg p$~~
$s \approx t$

Soundness, completeness and termination

Theorem. Let φ be any BISKT formula. For $a : F\ \varphi$ as input:

1. If φ is valid, then *Tab* constructs a closed tableau.
   A counter-model of φ can be read off from any fully-expanded, open branch.
2. The same holds for $Tab + (ub)$.
3. Every fully-expanded, open tableau derivation has a finite, open branch, provided (ub) is applied eagerly (or often enough).



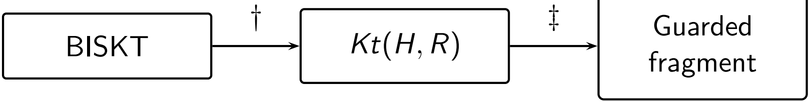
Proof: Consequence of

- General results in tableau synthesis framework
- BISKT has the finite model property
  (the only part which requires explicit and non-trivial proof)

Decidability, finite model property and complexity

Theorem

1. BISKT is decidable.
2. Let φ be any satisfiable BISKT-formula. Then there is a finite model for φ with a bounded number of domain elements.

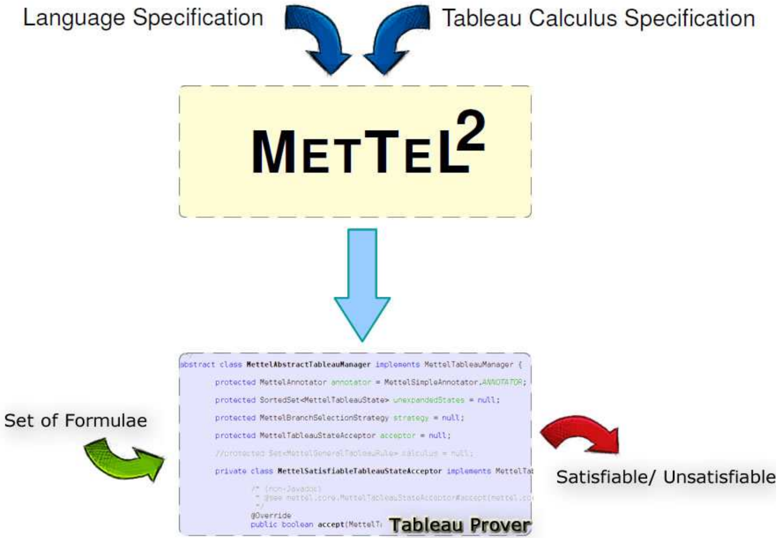


† = similar to embedding of IPL into S4
‡ = axiomatic translation principle [SH07]

Theorem

The complexity of testing BISKT-validity is PSPACE-complete.

Prover generation with MetTeL   www.mettel-prover.org



Using MetTeL to generate a prover for BISKT

```
// Input file for Mettel to generate prover for BISKT
//
// Reference: Stell, J. G., Schmidt, R. A., and Rydeheard, D. (2014),
// "Tableau Development for a Bi-Intuitionistic Tense Logic". In Proc.
// RAMiCS 2014. LNCS Vol. 8428, Springer, 412-428.

specification BISKT;

options {
    name.separator=
}

syntax BISKT {
   sort labelled, signed, formula, world;

   // Tableau formulae:  s:S formula  represented by  @ s (S formula)
   labelled at = '@' world signed;

   // Signed formulae
   signed trueValue = 'T' formula;
   signed falseValue = 'F' formula;

   // BISKT formulae
   formula false = 'false';
   // formula atom = '#' formula;
   formula negation          = '-' formula;
   formula dualNegation      = '~' formula;
   formula whiteBox          = '[]' formula;
   formula leftWhiteDiamond  = '<|>' formula;
   formula blackDiamond      = '<<>>' formula;
   formula leftBlackBox      = '[[|]]' formula;
```

Input file for MetTeL (cont'd): The tableau language

```
  formula conjunction       = formula '&' formula;
  formula disjunction       = formula '|' formula;
  formula implication       = formula '->' formula;
  formula dualImplication   = formula '>-' formula;

  // functions to create new witnesses
  world successorImp         = 'fi' '(' world ',' formula ',' formula ')';
  world successorDualImp     = 'fDi' '(' world ',' formula ',' formula ')';
  world successorNot         = 'fn' '(' world ',' formula ')';
  world successorDualNot     = 'fDn' '(' world ',' formula ')';
  world successorWhiteBox      = 'fwb' '(' world ',' formula ')';
  world successorLeftWhiteDia  = 'flwd' '(' world ',' formula ')';
  world successorBlackDia      = 'fbd' '(' world ',' formula ')';
  world successorLeftBlackBox  = 'flbb' '(' world ',' formula ')';
  world successorlg1         = 'lg1' '(' world ',' formula ')';
  world successorlg2         = 'lg2' '(' world ',' formula ')';
  world successorlh1         = 'lh1' '(' world ',' formula ')';
  world successorlh2         = 'lh2' '(' world ',' formula ')';

  // intuitionistic relation
  labelled greater_than = 'H' '(' world ',' world ')';

  // accessibility relation for white box
  labelled relation = 'R' '(' world ',' world ')';

  // for blocking
  labelled equality = '[' world '=' world ']';
  labelled not = 'not' labelled;

}
```

Input file for MetTeL (cont'd): The tableau calculus

```
tableau BISKT {
   @s(T P)  @s(F P)          // closure rule
       /                                        priority 0 $;
   @s(T false)               // rule for false
       /                                        priority 0 $;
   @s(T(P|Q))                // or rules
       / @s(T P) $| @s(T Q)                     priority 7 $;
   @s(F(P|Q))
       / @s(F P)  @s(F Q)                       priority 1 $;
   @s(T(P&Q))                // and rules
       / @s(T P)  @s(T Q)                       priority 1 $;
   @s(F(P&Q))
       / @s(F P) $| @s(F Q)                     priority 7 $;
   @s(T(-P))  H(s,t)         // intuitionistic negation
       / @t(F P)                                priority 2 $;
   @s(F(-P))
       / H(s,fn(s,P))  @fn(s,P)(T P)            priority 10 $;
   @s(T(~P))                 // dual negation
       / H(fDn(s,P),s)  @fDn(s,P)(F P)          priority 10 $;
   @s(F(~P))  H(t,s)
       / @t(T P)                                priority 2 $;
   @s(T(P->Q))  H(s,t)       // intuitionistic implication
       / @t(F P) $| @t(T Q)                     priority 2 $;
   @s(F(P->Q))
       / H(s,fi(s,P,Q))  @fi(s,P,Q)(T P)  @fi(s,P,Q)(F Q)
                                                priority 10 $;
   @s(T(P>-Q))               // dual implication
       / H(fDi(s,P,Q),s)  @fDi(s,P,Q)(T P)  @fDi(s,P,Q)(F Q)
                                                priority 10 $;
   @s(F(P>-Q))  H(t,s)
       / @t(F P) $| @t(T Q)                     priority 2 $;
```

Input file for MetTeL (cont'd): The tableau calculus

```
@s(T([]P))  R(s,t)        // white box
    / @t(T P)                          priority 2 $;
@s(F([]P))
    / R(s,fwb(s,P))  @fwb(s,P)(F P)    priority 10 $;
@s(T(<l>P))               // left white diamond operator
    / H(lg1(s,P),s)  R(lg1(s,P),lg2(s,P))  H(flwd(s,P),lg2(s,P))  @flwd(s,P)(T P)
                                       priority 10 $;
@s(F(<l>P))  H(t,s)  R(t,u)  H(v,u)
    / @v(F P)                          priority 4 $;
@s(T(<<>>P))              // black diamond operator
    / R(fbd(s,P),s)  @fbd(s,P)(T P)    priority 10 $;
@s(F(<<>>P))  R(t,s)
    / @t (F P)                         priority 2 $;
@s(T([[l]]P))  H(s,t)  R(u,t)  H(u,v) // left black box
    / @v (T P)                         priority 4 $;
@s(F([[l]]P))
    / H(s,lh1(s,P))  R(lh2(s,P),lh1(s,P))  H(lh2(s,P),flbb(s,P))  @flbb(s,P)(F P)
                                       priority 10 $;

// Frame and model conditions
@s P                      // H is a preorder: reflexivity
    / H(s,s)                           priority 3 $;
H(s,t)  H(t,u)            // transitivity
    / H(s,u)                           priority 2 $;

@s(T P)  H(s,t)           // monotonicity: sets form downsets
    / @t(T P)                          priority 2 $;

H(s,t)  R(t,u)  H(u,v)    // H;R;H subset R
    / R(s,v)                           priority 4 $;
```

Input file for MetTeL (cont'd): The tableau calculus

```
   // for blocking
   @s P  @t Q              // ub rule
       / [s=t] $| (not([s=t]))            priority 9 $;
   // H(s,t)                  // predecessor blocking
   //     / [s=t] $| (not([s=t]))            priority 9 $;
   // R(s,t)
   //     / [s=t] $| (not([s=t]))            priority 9 $;

   // property of equality
   not([s=s])
       /                                  priority 0 $;

}
```

Scope

- Any set of rules specifiable in a quantifier-free first-order language with multiple sorts and equality
- Function and relation symbols have fixed finite arity

Creating the prover with MetTeL

```
> java -jar mettel2.jar -i BISKT_specification
```

creates the prover.

Running the generated prover:

```
> java -jar BISKT.jar
```

with input

```
@a (T (- (- p)))
```

produces

```
Satisfiable.
Model: [( @ a ( T p ) ), ( @ a ( T ( - ( - p ) ) ) ),
( H ( a , a ) ), ( [ a = a ] )]
```

Running MetTeL (cont'd)

Running the prover with:

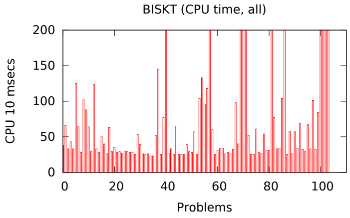
```
@a (F ((- (- (- p))) -> (- p)))
```

produces

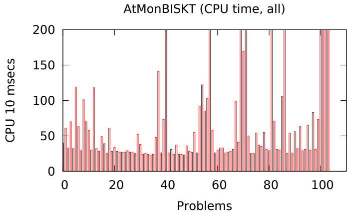
```
Unsatisfiable.
```

Experiments: Atomic monotonicity or not

$$\frac{s : T\ \alpha,\ H(s,t)}{t : T\ \alpha}$$



$$\frac{s : T\ p,\ H(s,t)}{t : T\ p}$$



http://staff.cs.manchester.ac.uk/~schmidt/publications/biskt13/

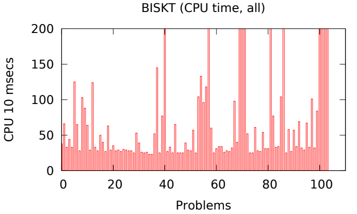
Ongoing work

Develop tableau calculi for $Kt(H, R)$, the modal interpretation of BISKT

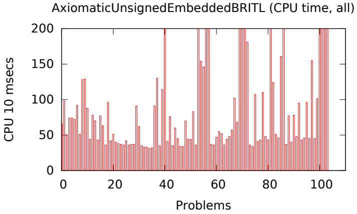
Have analysed numerous ways of defining tableau calculi

BISKT tableau vs modal tableau via embedding

BISKT tableau



Modal tableau



Future work

Extend BISKT with modalities based on right converse

Extend and improve the tableau generation tool support