Mining Human Proofs from Machine Proofs

Rob Arthan & Paulo Oliva

Queen Mary University of London

http://www.lemma-one.com/papers/papers.html

DREAM Seminar

4th May 2016

Overview

- Studying logics in or near the space between Intuitionistic Affine Logic $\mathbf{AL_i}$ and Classical Łukasiewicz logic $\mathbf{\L L_c}$:

$$\begin{array}{ccc} \mathbf{AL_c} & \longrightarrow & \mathbf{\L L_c} \\ \Big\uparrow & & \Big\uparrow \; \text{and beyond, e.g., } \mathbf{CL} \\ \mathbf{AL_i} & \longrightarrow & \mathbf{\L L_i}. \end{array}$$

- Using Mace4 to find semantic data (finite models)
- Using Prover9 to find proof-theoretic data (i.e. proofs!), e.g.,
  - To study translations of classical systems into intuitionistic ones.
- Making proofs readable and meaningful by an iterative process:
  - Use human insight to look for abstractions and decompose proofs into smaller steps.
  - If the smaller steps are still too complex, use Prover9 to prove them, and re-analyse the results.

Outline

Affine Logic: Language

- Intuitionistic affine logic $\mathbf{AL_i}$ has formulas built using:
  - Variables: $P, Q, R \ldots$
  - Falsehood: $\bot$
  - Conjunction: $A \otimes B$
  - Implication: $A \multimap B$.
- Negation: $A^\bot$ abbreviates $A \multimap \bot$.
- Sequents: $\Gamma \vdash A$ where Γ is a *multiset* of formulas.
- No disjunction:
  - For simplicity ...
  - ... and it is definable in $\mathbf{ŁL_c}$.

Affine Logic: Deductive System

- Axiom schemata:

$$\frac{}{\Gamma, A \vdash A}\,[\text{ASM}] \qquad \frac{}{\Gamma, \bot \vdash A}\,[\text{EFQ}].$$

- Introduction and elimination for $\multimap$ and $\otimes$:

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B}\,[\multimap\text{I}] \qquad \frac{\Gamma \vdash A \quad \Delta \vdash A \multimap B}{\Gamma, \Delta \vdash B}\,[\multimap\text{E}]$$

$$\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B}\,[\otimes\text{I}] \quad \frac{\Gamma \vdash A \otimes B \quad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C}\,[\otimes\text{E}].$$

- Weakening is admissible: $\dfrac{\Gamma \vdash B}{\Gamma, A \vdash B}\,[\text{WK}]$.
- Contraction is not admissible: $P \vdash P \otimes P$ is unprovable.

The Other Three Logics

- Classical Affine Logic $\mathbf{AL_c} = \mathbf{AL_i} + [\text{DNE}]$:

$$\Gamma, A^{\perp\perp} \vdash A. \qquad [\text{DNE}]$$

- Intuitionistic Łukasiewicz Logic $\mathbf{ŁL_i} = \mathbf{AL_i} + [\text{CWC}]$:

$$\Gamma, A \otimes (A \multimap B) \vdash B \otimes (B \multimap A). \qquad [\text{CWC}]$$

- Classical Łukasiewicz Logic $\mathbf{ŁL_c} = \mathbf{AL_c} + [\text{CWC}] = \mathbf{ŁL_i} + [\text{DNE}]$:

$$\begin{array}{ccc} \mathbf{AL_c} & \longrightarrow & \mathbf{ŁL_c} \\ \uparrow & & \uparrow \\ \mathbf{AL_i} & \longrightarrow & \mathbf{ŁL_i}. \end{array}$$

Semantics

- Algebraic semantics over algebras with signature $(0, 1, +, \rightarrow)$ called *(dual) pocrims*:
  - Order by: $x \geq y \equiv x \rightarrow y = 0$;
  - $(0, +, \geq)$ becomes an ordered commutative monoid;
  - 0 is least element: $x \geq 0$, i.e., $x \rightarrow x = 0$;
  - 1 is an annihilator: $1 + x = 1$;
  - implication is residuated: $x + y \rightarrow z = x \rightarrow y \rightarrow z$.
- The logics are sound and complete for subclasses of pocrims as follows:

$$
\begin{array}{ccc}
\mathbf{AL_c} : \text{involutive pocrims} & \longrightarrow & \mathbf{ŁL_c} : \text{MV-algebras} \\
\uparrow & & \uparrow \\
\mathbf{AL_i} : \text{pocrims} & \longrightarrow & \mathbf{ŁL_i} : \text{hoops.}
\end{array}
$$

where:

- involutive pocrims satisfy $\neg\neg x = x$, where $\neg x = x \rightarrow 1$;
- hoops satisfy $x + (x \rightarrow y) = y + (y \rightarrow x)$;
- MV-algebra $=$ involutive hoop.

Remarks on Hoops and **ŁL$_\mathbf{i}$**

- Hoops and **ŁL$_\mathbf{i}$** have been studied from various points of view.
- Hoops were first studied by Büchi and Owens.
- Bosbach gave an equational axiomatisation of hoops.
- Ferreirim studied hoops from the point of view of universal algebra.
- Can usefully view [CWC] as a weak form of contraction.
- **ŁL$_\mathbf{i}$** may be viewed as Hajek's Basic Logic **without** the intuitionistically unacceptable axiom of *arrow prelinearity*:

$$(A \multimap B) \multimap C, (B \multimap A) \multimap C \vdash C. \qquad \text{[PREL]}$$

Exploiting the Semantics

- Semantics give a powerful handle on the theories.
- Finite counter-examples are important for many results.
- Checking associativity is tedious and error-prone.
- Had a lot of success with Mace4.
  - E.g., classify pocrims with 4 elements:
    - 2 MV-algebras
    - 5 hoops
    - 3 involutive pocrims
    - 7 pocrims
  - Mace4 finds the examples in a few minutes.
  - Proving the classification is a short "homework" exercise.
- Combined with some POFM, get some nice general results.
- We found a heuristic test which allows us to test many simple formulas.
- But no data to satisfy a hungry proof-theorist.

Decision Problem in $\mathbf{AL_i}$, $\mathbf{AL_c}$, $\mathbf{ŁL_i}$ and $\mathbf{ŁL_c}$

- $\mathbf{AL_i}$ and $\mathbf{AL_c}$ admit cut-elimination.
  - Resulting decision procedure is EXPTIME
  - NP-complete?
- $\mathbf{ŁL_c}$ is equivalent to the equational theory of MV-algebras:
  - Known to be decidable by a reduction to (linear) real arithmetic.
  - NP-complete
- $\mathbf{ŁL_i}$ was shown to be decidable by Ferreirm & Blok.
  - Bova & Montagna's work on GBL-algebras imply decision problem is PSPACE-complete.
    - Their algorithm is justified by abstract algebra, and
    - doesn't produce proofs
  - We have a heuristic which works well on simple formulas.
    - Our heuristic is justified by abstract algebra, and
    - doesn't produce proofs

Aside: Continuous Logic

- The continuous logic **CL** of Ben Yaacov et al. is classical Łukasiewicz logic $\mathbf{ŁL_c}$ extended with
  - a unary logical connective called *halving* and written $A/2$.
  - the axiom schemata:

$$A/2 \multimap A \vdash A/2$$
$$A/2 \vdash A/2 \multimap A.$$

  - Algebraic semantics: *continuous hoops*, aka *coops* (see `http://arxiv.org/abs/1212.2887`)
- We had already decomposed $\mathbf{ŁL_c}$ as the combination of:
  - the classical principle axiomatizing $\mathbf{AL_c}$:

$$A^{\perp\perp} \vdash A$$

  - and the intuitionistic principle axiomatizing $\mathbf{ŁL_i}$:

$$A, A \otimes (A \multimap B) \vdash B \otimes (B \multimap A).$$

- Wanted to understand halving in $\mathbf{ŁL_i}$ alone

Putting Prover9 to work in ŁL_{i}

- This led us to ask if the following rule is admissible in ŁL_{i}:

$$\frac{H \multimap A \vdash H \quad H \vdash H \multimap A \quad H' \multimap A \vdash H' \quad H' \vdash H' \multimap A}{H \vdash H'}$$

- I.e., does every model of ŁL_{i} admit at most one halving operator?
- Answer: Yes!
  - Proof found (in about 3 minutes) using Prover9.
  - Subsequently massaged into human-readable form by us.
  - Less than 2 pages, but involves subtle applications of [CWC].
  - The de Bruijn factor is 1.09 after compression . . .
    and about the same on paper . . .

The Machine Proof

Appendix

Formal proof of Theorem 3 as output by Prover9:

```
1 x >= y & y >= z -> x >= z # label(non_clause).  [assumption].
2 x >= y & y >= x -> x = y # label(non_clause).  [assumption].
3 x + z >= y <-> z >= x ==> y # label(non_clause).  [assumption].
4 x >= y -> x + z >= y + z # label(non_clause).  [assumption].
5 x >= y -> y ==> z >= x ==> z # label(non_clause).  [assumption].
6 x >= y -> z ==> x >= z ==> y # label(non_clause).  [assumption].
7 y = y ==> x & z = z ==> x -> y = z
                             # label(non_clause) # label(goal).  [goal].
8 (x + y) + z = x + (y + z).  [assumption].
9 x + y = y + x.  [assumption].
10 x + 0 = x.  [assumption].
11 x >= x.  [assumption].
12 -(x >= y) | -(y >= z) | x >= z.  [clausify(1)].
13 -(x >= y) | -(y >= x) | y = x.  [clausify(2)].
14 -(x + y >= z) | y >= x ==> z.  [clausify(3)].
15 x + y >= z | -(y >= x ==> z).  [clausify(3)].
16 x >= 0.  [assumption].
17 -(x >= y) | x + z >= y + z.  [clausify(4)].
18 -(x >= y) | y ==> z >= x ==> z.  [clausify(5)].
19 -(x >= y) | z ==> x >= z ==> y.  [clausify(6)].
20 x + (x ==> y) = y + (y ==> x).  [assumption].
21 c1 ==> c2 = c1.  [deny(7)].
22 c3 ==> c2 = c3.  [deny(7)].
23 c3 != c1.  [deny(7)].
24 x + (y + z) = y + (x + z).  [para(9(a,1),8(a,1,1)),rewrite([8(2)])].
27 0 + x = x.  [para(10(a,1),9(a,1)),flip(a)].
28 x >= y ==> (y + x).  [hyper(14,a,11,a)].
30 -(x + y >= z) | x >= y ==> z.  [para(9(a,1),14(a,1))].
31 -(x >= y) | 0 >= x ==> y.  [para(10(a,1),14(a,1))].
32 x + (x ==> y) >= y.  [hyper(15,b,11,a)].
33 x >= y ==> 0.  [hyper(14,a,16,a)].
34 x + y >= y.  [hyper(17,a,16,a),rewrite([27(3)])].
35 0 ==> x >= y ==> x.  [hyper(18,a,16,a)].
36 x + ((x ==> y) + z) = y + ((y ==> x) + z).
                             [para(20(a,1),8(a,1,1)),rewrite([8(3)])].
41 c3 + x >= c2 | -(x >= c3).  [para(22(a,1),15(b,2))].
43 -(x + (y + z) >= u) | x + z >= y ==> u.  [para(24(a,1),14(a,1))].
46 0 ==> x = x + (x ==> 0).  [para(27(a,1),20(a,1))].
52 x ==> 0 = 0.  [hyper(13,a,16,a,b,33,a),flip(a)].
53 0 ==> x = x.  [back_rewrite(46),rewrite([52(4),10(4)])].
54 x >= y ==> x.  [back_rewrite(35),rewrite([53(2)])].
55 x ==> (y + z) >= x ==> z.  [hyper(19,a,34,a)].
70 x >= y ==> (x + y).  [para(9(a,1),28(a,2,2))].
81 c2 >= c1.  [para(21(a,1),54(a,2))].
82 c2 >= c3.  [para(22(a,1),54(a,2))].
86 x + c2 >= c1.  [hyper(12,a,34,a,b,81,a)].
```



```
89 x ==> c2 >= x ==> c3.  [hyper(19,a,82,a)].
127 x >= c2 ==> c1.  [hyper(30,a,86,a)].
171 c2 ==> c1 = 0.  [hyper(13,a,16,a,b,127,a),flip(a)].
180 c1 + c1 = c2.
        [para(171(a,1),20(a,1,2)),rewrite([9(3),27(3),21(5)]),flip(a)].
205 c1 + (x + c1) = x + c2.  [para(180(a,1),8(a,2,2)),rewrite([9(4)])].
271 x + ((x ==> y) + ((y ==> x) ==> z)) = y + (z + (z ==> (y ==> x))).
                                          [para(20(a,1),36(a,1,2)),flip(a)].
275 (x ==> y) + z >= x ==> (y + ((y ==> x) + z)).  [para(36(a,1),28(a,2,2))].
418 c1 >= c1 ==> c3.  [para(21(a,1),89(a,1))].
419 0 >= c1 ==> (c1 ==> c3).  [hyper(31,a,418,a)].
609 c3 + (x + (x ==> c3)) >= c2.  [hyper(41,b,32,a)].
895 c3 ==> (x + c2) >= c3.  [para(22(a,1),55(a,2))].
996 c1 ==> (c1 ==> c3) = 0.  [hyper(13,a,16,a,b,419,a),flip(a)].
5220 c3 ==> (c1 + (x + c1)) >= c3.  [para(205(a,2),895(a,1,2))].
10398 c3 + (x ==> c3) >= x ==> c2.  [hyper(43,a,609,a)].
16713 c3 + ((c3 ==> c1) + ((c1 ==> c3) ==> c1)) = c2.
          [para(996(a,1),271(a,2,2,2)),rewrite([9(15),27(15),180(14)])].
20059 c1 + (c3 ==> c1) >= c3.  [hyper(12,a,275,a,b,5220,a),rewrite([9(5)])].
20066 c3 ==> c1 >= c1 ==> c3.  [hyper(14,a,20059,a)].
20564 c3 + (c1 ==> c3) >= c1.  [para(21(a,1),10398(a,2))].
20570 c1 ==> c3 >= c3 ==> c1.  [hyper(14,a,20564,a)].
20614 c3 ==> c1 = c1 ==> c3.  [hyper(13,a,20066,a,b,20570,a),flip(a)].
20625 c1 + c3 = c2.
   [back_rewrite(16713),rewrite([20614(4),20(10),996(7),9(4),27(4),9(3)])].
20634 c3 >= c1.  [para(20625(a,1),28(a,2,2)),rewrite([21(4)])].
20637 c1 >= c3.  [para(20625(a,1),70(a,2,2)),rewrite([22(4)])].
20793 -(c1 >= c3).  [ur(13,b,20634,a,c,23,a)].
20794 $F.  [resolve(20793,a,20637,a)].
```

The Human Proof

```
   a ==> b = a.                    % assumption 1
   c ==> b = c.                    % assumption 2
end_of_list.
formulas(goals).
   a = c.
end_of_list.
```

To our surprise Prover9 took just a few seconds to produce the proof shown in the appendix. The proof that Prover9 found seems perplexingly intricate at first glance, but after studying it for a little while, we found we could edit it into a form fit for human consumption. From a human perspective, the proof involves the 9 intermediate claims given in the following lemma. Once these are proved, we will see that the desired result is an easy consequence of claim (9),

Lemma 2 *Let* $\mathbf{M} = (M, 0, +, \to; \geq)$ *be a hoop and let* $a, b, c, x, y \in M$*. Assume that, (i),* $a \to b = a$ *and, (ii),* $c \to b = c$*. Then the following hold:*

(1) $b \geq a$ *and* $b \geq c$,
(2) $a + a = b$,
(3) $a \to (a \to c) = 0$,
(4) $(x \to y) + z \geq x \to (y + (y \to x) + z)$,
(5) $c \to (a + a + x) \geq c$,
(6) $c \to a \geq a \to c$,
(7) $c \to a = a \to c$,
(8) $c + (c \to a) + ((a \to c) \to a) = b$,
(9) $a + c = b$.

Proof: In the proof below (in)equalities which are not labelled as following from one of the assumptions (i) and (ii) or an earlier part of the lemma follow immediately from the axioms of a pocrim.

(1) We have $b \geq a \to b$ and, by (i), $a \to b = a$). So $b \geq a$ and similarly $b \geq c$ using (ii).
(2) By (1) we have $b \to a = 0$. Therefore

$$\begin{aligned} a + a &= a + (a \to b) && (i) \\ &= b + (b \to a) && [\textsf{cwc}] \\ &= b. \end{aligned}$$

(3) By (i) and (1) we have $a = a \to b \geq a \to c$ and hence $0 \geq a \to (a \to c)$, which implies (3).
(4) By [cwc] $x + (x \to y) + z = y + (y \to x) + z$, whence (4) follows.
(5) We have

$$\begin{aligned} c \to (b + x) &\geq c \to b \\ &= c && (ii) \end{aligned}$$

and then using (2) we obtain (5).
(6) By (5), as $(c \to a) + a \geq c \to (a + a)$, we have $(c \to a) + a \geq c$ and hence (6).
(7) Our assumptions are symmetric in a and c. Hence, (6) holds with a and c interchanged, i.e., $a \to c \geq c \to a$, which taken with (6) gives (7).
(8) We have

$$\begin{aligned} c + (c \to a) + ((a \to c) \to a) &= a + (a \to c) + ((a \to c) \to a) && [\textsf{cwc}] \\ &= a + a + (a \to (a \to c)) && [\textsf{cwc}] \\ &= b + (a \to (a \to c)) && (2) \\ &= b. && (3) \end{aligned}$$

(9) We have

$$\begin{aligned} b &= c + (c \to a) + ((a \to c) \to a) && (8) \\ &= c + (a \to c) + ((a \to c) \to a) && (7) \\ &= c + a + (a \to (a \to c)) && [\textsf{cwc}] \\ &= c + a. && (3) \end{aligned}$$

This completes the proof of the lemma. ■

It is interesting to note the complexity of the proof in terms of uses of [cwc] (used 6 times!) and the important sub-lemma (2) (used twice) as depicted in the outline proof tree shown in Figure 5.

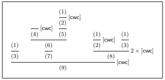

Fig. 5: Outline of the Proof of Lemma 2

Finally, from part (9) of Lemma 2 we have the theorem that the equation $a \to b = a$ uniquely determines a in terms of b:

Theorem 3 *In any hoop, if* $a \to b = a$ *and* $c \to b = c$ *then* $a = c$.

Proof: Since the assumptions are symmetric in a and c it is enough to show $c \geq a$, from which we can immediately conclude $a \geq c$ and hence $a = c$. By Lemma 2 (9) we have $c \geq a \to b$ and hence $c \geq a$. ■

We already have the part of Theorem 1 that gives soundness and completeness of $\mathbf{LL_i}$ for bounded hoops. Theorem 3 now gives us that the continuous logic

Fun with Prover9 and $\mathbf{Ł L_i}$

- Tried the Prover9 approach on range of conjectures
- Still had conjectures that we could prove with the heuristic but not with Prover9
  - More on this later in the talk
- With the kind assistance of Geoff Sutcliffe contributed a batch of problems to TPTP
  - LCL882+1.p to LCL903+1.p
  - Mix of counter-example generation and proof problems

See *(Dual) Hoops have Unique Halving* Essays in Memory of Bill McCune. LNCS 7788 or `http://arxiv.org/abs/1203.0436`

Prover9 Performance on TPTP Proof Problems

TPTP Name	Problem Statement	Seconds
LCL888+1.p	Halving is unique: rule for $a = b/2$	3.38
LCL889+1.p	Halving is unique: rule for $a \geq b/2$	229.13
LCL890+1.p	Halving is unique: rule for $a \leq b/2$ (i)	1,216.69
LCL891+1.p	Halving is unique: rule for $a \leq b/2$ (ii)	12,724.08
LCL892+1.p	Halving is unique: rule for $a \leq b/2$ (iii)	51,876.82
LCL893+1.p	$x/2 = x$ implies $x = 0$	0.01
LCL894+1.p	Weak conjunction is l.u.b. in a hoop (Horn)	1.90
LCL895+1.p	Weak conjunction is l.u.b. in a hoop (Equational)	14.41
LCL896+1.p	Associativity of weak conjunction implies CWC	5.95
LCL897+1.p	Weak conjunction is associative in a hoop	0.10
LCL898+1.p	An involutive hoop has CSD	66.30
LCL899+1.p	A bounded pocrim with CSD is involutive	0.01
LCL900+1.p	A bounded pocrim with CSD is a hoop	7.21
LCL901+1.p	An idempotent pocrim with CSD is boolean	0.74
LCL902+1.p	A boolean pocrim is involutive	0.02
LCL903+1.p	A boolean pocrim is idempotent	1.42

Double Negation Translations

- Schemes for encoding $\mathbf{L} + [\mathrm{DNE}]$ in $\mathbf{L}$ for some extension $\mathbf{L}$ of $\mathbf{AL_i}$.
- **Kolmogorov:** double negate every subformula. E.g.,

$$P \otimes (P \multimap Q) \longmapsto (P^{\perp\perp} \otimes (P^{\perp\perp} \multimap Q^{\perp\perp})^{\perp\perp})^{\perp\perp}.$$

- **Gentzen:** double negate variables. E.g.,

$$P \otimes (P \multimap Q) \longmapsto P^{\perp\perp} \otimes (P^{\perp\perp} \multimap Q^{\perp\perp}).$$

- **Gödel:** rewrite implication using conjunction and negation. E.g.,

$$P \otimes (P \multimap Q) \longmapsto P \otimes (P \otimes Q^{\perp})^{\perp}.$$

- **Glivenko:** double negate outermost formula only. E.g.,

$$P \otimes (P \multimap Q) \longmapsto (P \otimes (P \multimap Q))^{\perp\perp}.$$

Double Negation Translations (continued)

- $\mathbf{L}$ will typically be an intuitionistic extension of $\mathbf{AL_i}$.
- Want encoding to reflect in $\mathbf{L}$ the proof theory of "classical $\mathbf{L}$", i.e., $\mathbf{L_c} = \mathbf{L} + [\mathrm{DNE}]$.
- Following Troelstra, we say an encoding $A \mapsto A^\dagger$ is a *correct double negation translation* if:

  (DNS1) $\mathbf{L_c}$ proves $A^\dagger \vdash A$ and $A \vdash A^\dagger$,

  (DNS2) if $\mathbf{L_c}$ proves $\vdash A$ then $\mathbf{L}$ proves $\vdash A^\dagger$,

  (DNS3) $\mathbf{L}$ proves $(A^\dagger)^{\perp\perp} \vdash A^\dagger$.
- For short, we just say the encoding *"works"* in $\mathbf{L}$.
- E.g., Kolmogorov and Gödel encodings work in $\mathbf{AL_i}$ and in $\mathbf{ŁL_i}$. (The proofs are fairly routine exercises in induction over derivations.)

Double Negation is a Homomorphism in $\mathbf{ŁL_i}$

- We had proved using semantic methods (with some help from Mace4) that there are extensions of $\mathbf{AL_i}$ where Gentzen works and Glivenko doesn't and vice versa.
- Do Gentzen and Glivenko encodings work in $\mathbf{ŁL_i}$?
- We conjectured two homomorphism properties:

$$(A \otimes B)^{\perp\perp} \simeq A^{\perp\perp} \otimes B^{\perp\perp}$$
$$(A \multimap B)^{\perp\perp} \simeq A^{\perp\perp} \multimap B^{\perp\perp}$$

  (where $X \simeq Y$ means $\mathbf{ŁL_i}$ proves $X \vdash Y$ and $Y \vdash X$).
- Follows easily from these that all four translations are equivalent in $\mathbf{ŁL_i}$ and hence that Gentzen and Glivenko encodings work in $\mathbf{ŁL_i}$.
- A key intermediate result for the homomorphism properties is:

$$(A^{\perp} \multimap B)^{\perp} \simeq A^{\perp} \otimes B^{\perp} \qquad (*)$$

- After a two hour search, Prover9 found a proof of (*).

Understanding the Proofs

- The first proofs of the homomorphism properties were long.
- But patterns emerge. E.g., certain derived connectives keep appearing:

$A \wedge B \equiv A \otimes (A \multimap B)$	weak conjunction
$A \vee B \equiv (B \multimap A) \multimap A$	strong disjunction
$A \Rightarrow B \equiv A \multimap A \otimes B$	strong implication
$A \downarrow B \equiv A^{\perp} \otimes (B \multimap A).$	NOR, Peirce's *ampheck*

- Analysed the Prover9 proofs by identifying key lemmas and feeding them back in first as conjectures and then as axioms.
- Resulting account for humans is about 7 pages containing 17 lemmas and theorems.

On Affine Logic and Łukasiewicz Logic `http://arXiv.org/abs/1404.0570`

Demo

- Let's get Prover9 to prove that $\mathbf{ŁL_i}$ proves:

$$\vdash (A^{\perp\perp} \multimap A)^{\perp\perp}.$$

Improving a Prover9 Proof

	Theorem	Length	Depth	Time
(1)	$(A^{\perp\perp} \multimap A)^{\perp\perp}$	109 steps	9	1 min
(2)	$(A^{\perp} \multimap B)^{\perp} \simeq A^{\perp} \otimes B^{\perp}$	412 steps	22	133 min
(3)	$(A \wedge B)^{\perp} \simeq A \Rightarrow B^{\perp}$	147 steps	13	86 min
(2)	$(A^{\perp} \multimap B)^{\perp} \simeq A^{\perp} \otimes B^{\perp}$	140 steps*	10	43 sec

($*$) using (3)

Further Properties of the Connectives

- Have De Morgan properties for the various connectives:

$$
\begin{aligned}
(A \otimes B)^\perp &\simeq A \multimap B^\perp \\
(A \multimap B)^\perp &\simeq A^{\perp\perp} \otimes B^\perp \\
(A \wedge B)^\perp &\simeq A \Rightarrow B^\perp \\
(A \Rightarrow B)^\perp &\simeq A^{\perp\perp} \wedge B^\perp \\
(A \wedge B)^\perp &\simeq A^\perp \vee B^\perp \\
(A \vee B)^\perp &\simeq A^\perp \wedge B^\perp .
\end{aligned}
$$

- The conjunctions $\otimes$ and $\wedge$ and the implications $\multimap$ and $\Rightarrow$ become the usual intuitionistic connectives given contraction ($A \vdash A \otimes A$).
- N.b., $\vee$ is not the usual intuitionistic disjunction.
- $\downarrow$ is definable in terms of $\wedge$ and $^\perp$:

$$A \downarrow B \simeq A^\perp \wedge B^\perp .$$

A Challenge Problem

- A conjecture about the weak conjunction and the strong implication:

$$A \wedge B \Rightarrow C \simeq A \Rightarrow B \Rightarrow C$$

- Prover9 proofs:
  - Bob Veroff (242 lines with Horn aziomatization)
    Found using Veroff's method of proof sketches
  - Michael Kinyon (214 lines with equational axiomatization)
    Found using Waldmeister first
- Have not yet teased out a human readable proof

See `http://www.cs.unm.edu/~veroff/HOOPS/` for Prover9 proofs

Concluding Remarks

- Successfully mined human-readable proofs from machine proofs.
- Results have informed research outside ATP.
- Human input is identifying the “right” abstractions;
  - Find useful derived concepts;
  - Recover an intuitive proof plan.
- Useful to have interactive support for proof factoring.
- Interesting AI challenge to automate human aspects.
- Engineering applications?
  - *“OK, your system has proved it works . . . ”*
  - *“But what does the proof mean?”*
- The late Bill McCune was the real star:
  - Mace4 provided quick returns and good value for a very low entry cost, and then
  - Prover9 found constructive proofs that we would never have found without it.

Thank you!