Logics with Explicit Evidence

Melvin Fitting
City University of New York

WoLLIC July, 2005
Florianópolis, Brazil

The Pre-History

Give meaning to Intuitionistic Logic.

Do so in the spirit of the subject.

That is, keep it as constructive as you can.

Provability Semantics

BHK (Brouwer, Heyting, Kolmogorov)

What an intuitionist means by *truth* is what a classical mathematician means by *provable*.

In More Detail

1. A proof of $X \wedge Y$ is a proof of X and a proof of Y
2. A proof of $X \vee Y$ is a choice of one to prove, together with a proof of the one chosen
3. A proof of $X \supset Y$ is a construction converting any proof of X into a proof of Y
4. A proof of $\perp$ does not exist

Gödel Contributes

Provable has the following features, on top of classical logic:

- $\mathsf{BP} \supset \mathsf{P}$
- $\mathsf{B}(\mathsf{P} \supset \mathsf{Q}) \supset (\mathsf{BP} \supset \mathsf{BQ})$
- $\mathsf{BP} \supset \mathsf{BBP}$
- Conclude BP from P

Of course, this is S4.

Intuitionistic logic embeds into this.
Translate every subformula X by BX.

Unfortunately, this abstract notion of provable *cannot* be captured by a provability formula, *Bew*, of Peano arithmetic.

(Gödel 1933)

- Provability in arithmetic is captured by the modal logic GL

- **BUT**

- Intuitionistic logic embeds into S4

- S4 captures our intuitions about ‘mathematical provability’

Gödel Again

- Maybe S4 is not about provability in arithmetic
- Maybe S4 is about ***explicit*** proofs
- 1938, but not known until rediscovered independently

LP – Logic of Proofs

Introduce a family of proof polynomials,
terms that are 'modal like'
and are intended to represent
explicit proofs

(Artemov 1995)

Proof Polynomials

- variables, x, y, ..., intended to range over proofs

- constants, a, b, ..., for axioms
  (more generally, for obvious facts)

- application, if s proves $A \supset B$ and t proves A then $s \cdot t$ proves B

- proof checker, if t proves A then $!t$ proves that t proves A

- union, $+$ joins two proofs together

Formulas

- Propositional letters are formulas
- 'Falsehood' is a formula
- If *X* is a formula and *t* is a proof polynomial then *t*:*X* is a formula
- Build up using implication (with other connectives defined, say)

The Axiom System

Axioms:

- all classical tautologies
- $t{:}(X \supset Y) \supset (s{:}X \supset (t \cdot s){:}Y)$
- $t{:}X \supset X$
- $t{:}X \supset !t{:}t{:}X$
- $s{:}X \supset (s+t){:}X \qquad t{:}X \supset (s+t){:}X$

Rules

- Modus Ponens
- $c{:}X$ where X is an axiom and c is a proof constant

Internalization

Derived Inference Rule:

$$\frac{X}{p{:}X}$$

For some proof polynomial p

In fact, p 'reflects' the proof of X

This corresponds to the necessitation rule of S4

Sketch of Proof

By induction on length of axiomatic derivation.

- A (an axiom)
- $\dfrac{X \quad X \supset Y}{Y}$
- $\dfrac{}{c{:}A}$ (A an axiom)

- $c{:}A$
- $u{:}X$
  $t{:}(X \supset Y)$
  $t{:}(X \supset Y) \supset (u{:}X \supset (t \cdot u){:}Y)$
  $u{:}X \supset (t \cdot u){:}Y$
  $(t \cdot u){:}Y$
- $c{:}A$
  $c{:}A \supset !c{:}c{:}A$
  $!c{:}c{:}A$

A Very Simple Example

$c{:}((P \wedge Q) \supset P)$

$!c{:}c{:}((P \wedge Q) \supset P)$

$c{:}((P \wedge Q) \supset P) \supset$
$(x{:}(P \wedge Q) \supset (c \cdot x){:}P)$

$d{:}[c{:}((P \wedge Q) \supset P) \supset$
$(x{:}(P \wedge Q) \supset (c \cdot x){:}P)]$

$x{:}(P \wedge Q) \supset (c \cdot x){:}P$

$(d{\cdot}!c){:}(x{:}(P \wedge Q) \supset (c \cdot x){:}P)$

The Full Version

If

$$s_1{:}A_1, \ldots, s_n{:}A_n, B_1, \ldots, B_k \vdash X$$

then for some proof polynomial t,

$$s_1{:}A_1, \ldots, s_n{:}A_n, x_1{:}B_1, \ldots, x_k{:}B_k \vdash t(s_1, \ldots, s_n, x_1, \ldots, x_k){:}X$$

Realization Theorem

Every LP theorem is an 'analysis' of an S4 theorem
and
every S4 theorem has an LP 'analysis'.

LP Embeds in S4

Replace every proof polynomial by $\Box$.

Every LP axiom becomes an S4 axiom.
Every LP rule application becomes one in S4.

An LP proof becomes an S4 proof.

An LP theorem becomes an S4 theorem.

This is the easy direction.

S4 embeds in LP

1. $\Box(X \supset Y) \supset (\Box X \supset \Box Y)$

2. $\Box[\Box(X \supset Y) \supset (\Box X \supset \Box Y)]$

3. $\Box[\Box(X \supset Y) \supset (\Box X \supset \Box Y)] \supset [\Box\Box(X \supset Y) \supset \Box(\Box X \supset \Box Y)]$

4. $\Box\Box(X \supset Y) \supset \Box(\Box X \supset \Box Y)$

5. $\Box(X \supset Y) \supset \Box\Box(X \supset Y)$

6. $\Box(X \supset Y) \supset \Box(\Box X \supset \Box Y)$

1. $x{:}(X \supset Y) \supset (y{:}X \supset (x \cdot y){:}Y)$

2. $c{:}[x{:}(X \supset Y) \supset (y{:}X \supset (x \cdot y){:}Y)]$

3. $c{:}[x{:}(X \supset Y) \supset (y{:}X \supset (x \cdot y){:}Y)] \supset [!x{:}x{:}(X \supset Y) \supset (c{\cdot}!x){:}(y{:}X \supset (x \cdot y){:}Y)]$

4. $!x{:}x{:}(X \supset Y) \supset (c{\cdot}!x){:}(y{:}X \supset (x \cdot y){:}Y)$

5. $x{:}(X \supset Y) \supset !x{:}x{:}(X \supset Y)$

6. $x{:}(X \supset Y) \supset (c{\cdot}!x){:}(y{:}X \supset (x \cdot y){:}Y)$

An Example Using +

1. $\Box X \supset (\Box X \vee \Box Y)$
2. $\Box(\Box X \supset (\Box X \vee \Box Y))$
3. $\Box X \supset \Box\Box X$
4. $\Box\Box X \supset \Box(\Box X \vee \Box Y)$
5. $\Box X \supset \Box(\Box X \vee \Box Y)$
6. $\Box Y \supset (\Box X \vee \Box Y)$
7. $\Box(\Box Y \supset (\Box X \vee \Box Y))$
8. $\Box Y \supset \Box\Box Y$
9. $\Box\Box Y \supset \Box(\Box X \vee \Box Y)$
10. $\Box Y \supset \Box(\Box X \vee \Box Y)$
11. $(\Box X \vee \Box Y) \supset \Box(\Box X \vee \Box Y)$

1. $x{:}X \supset (x{:}X \vee y{:}Y)$
2. $a{:}(x{:}X \supset (x{:}X \vee y{:}Y))$
3. $x{:}X \supset !x{:}x{:}X$
4. $!x{:}x{:}X \supset (a{\cdot}!x){:}(x{:}X \vee y{:}Y)$
5. $x{:}X \supset (a{\cdot}!x){:}(x{:}X \vee y{:}Y)$
6. $y{:}Y \supset (x{:}X \vee y{:}Y)$
7. $b{:}(y{:}Y \supset (x{:}X \vee y{:}Y))$
8. $y{:}Y \supset !y{:}y{:}Y$
9. $!y{:}y{:}Y \supset (b{\cdot}!y){:}(x{:}X \vee y{:}Y)$
10. $y{:}Y \supset (b{\cdot}!y){:}(x{:}X \vee y{:}Y)$

    $x{:}X \supset (a{\cdot}!x + b{\cdot}!y){:}(x{:}X \vee y{:}Y)$

    $y{:}Y \supset (a{\cdot}!x + b{\cdot}!y){:}(x{:}X \vee y{:}Y)$
11. $(x{:}X \vee y{:}Y) \supset (a{\cdot}!x + b{\cdot}!y){:}(x{:}X \vee y{:}Y)$

Realization Theorem

(Artemov, 1995)

LP realizes S4, as you just saw.

Negative □ occurrences become variables.
Other occurrences are computed from them.

Original method of proof is constructive.
By induction on complexity of cut-free Gentzen proof,
use it to generate a realization.

A Simplified Example to illustrate the proof

I'll show where the last example came from.

I'll use tableaus instead of Gentzen systems.

Use *signed* formulas, *TX*, and *FX*

Branch closure if both *TA* and *FA*
or if *T*⊥

Classical Rules

$$\frac{T\ X \vee Y}{T\ X \mid T\ Y} \qquad \frac{F\ X \vee Y}{\begin{array}{c} F\ X \\ F\ Y \end{array}}$$

$$\frac{T\ X \supset Y}{F\ X \mid T\ Y} \qquad \frac{F\ X \supset Y}{\begin{array}{c} T\ X \\ F\ Y \end{array}}$$

etc.

Modal Rules

$$\frac{T\ \Box X}{T\ X} \qquad \frac{\begin{array}{c} S \\ F\ \Box X \end{array}}{\begin{array}{c} S^\sharp \\ F\ X \end{array}}$$

where $S^\sharp = \{T\ \Box Z \mid T\ \Box Z \in S\}$

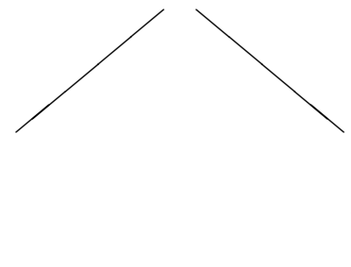
The $F\ \Box X$ rule is a *branch modification rule*.

S4 Proof

$F\,(\Box A \vee \Box B) \supset \Box(\Box A \vee \Box B)$ 1.

$T\,\Box A \vee \Box B$ 2.

$F\,\Box(\Box A \vee \Box B)$ 3.



$T\,\Box A$ 4.

$T\,\Box B$ 5.

Next Step

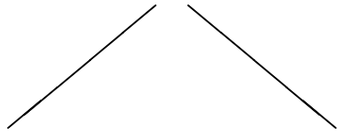
Label the $\Box$ operator involved in a $T\,\Box$ rule application with a variable, and follow this occurrence throughout the tableau.

Do this for each rule application.

$F\,(x{:}A \vee \Box y{:}B) \supset \Box((\Box A \vee \Box B))$ 1.

$T\,x{:}A \vee \Box y{:}B$ 2.

$F\,\Box(\Box A \vee \Box B)$ 3.



$T\,x{:}A$ 4.

$T\,x{:}A$ 6.

$F\,\Box A \vee \Box B$ 7.

$F\,\Box A$ 8.

$F\,\Box B$ 9.

$F\,A$ 10.

$T\,x{:}A$ 11.

$T\,A$ 12.

$T\,\Box y{:}B$ 5.

$T\,\Box y{:}B$ 13.

$F\,\Box A \vee \Box B$ 14.

$F\,\Box A$ 15.

$F\,\Box B$ 16.

$F\,B$ 17.

$T\,\Box y{:}B$ 18.

$T\,B$ 19.

Then

Pick a lowest application of $F\,\Box$.

We went from $\{T\,\Box A_1, \ldots T\,\Box A_n, \ldots, F\,\Box X\}$ to $\{T\,\Box A_1, \ldots T\,\Box A_n, F\,X\}$

Turn this over, and axiomatize. A proof of $(\Box A_1 \wedge \ldots \wedge \Box A_n) \supset X$ yields a proof of $(\Box A_1 \wedge \ldots \wedge \Box A_n) \supset \Box X$.

Since the rule application was lowest, each $\Box A_i$ has been realized, and there is an LP proof of $(s_1{:}A_1 \wedge \ldots \wedge s_n{:}A_n) \supset X$.

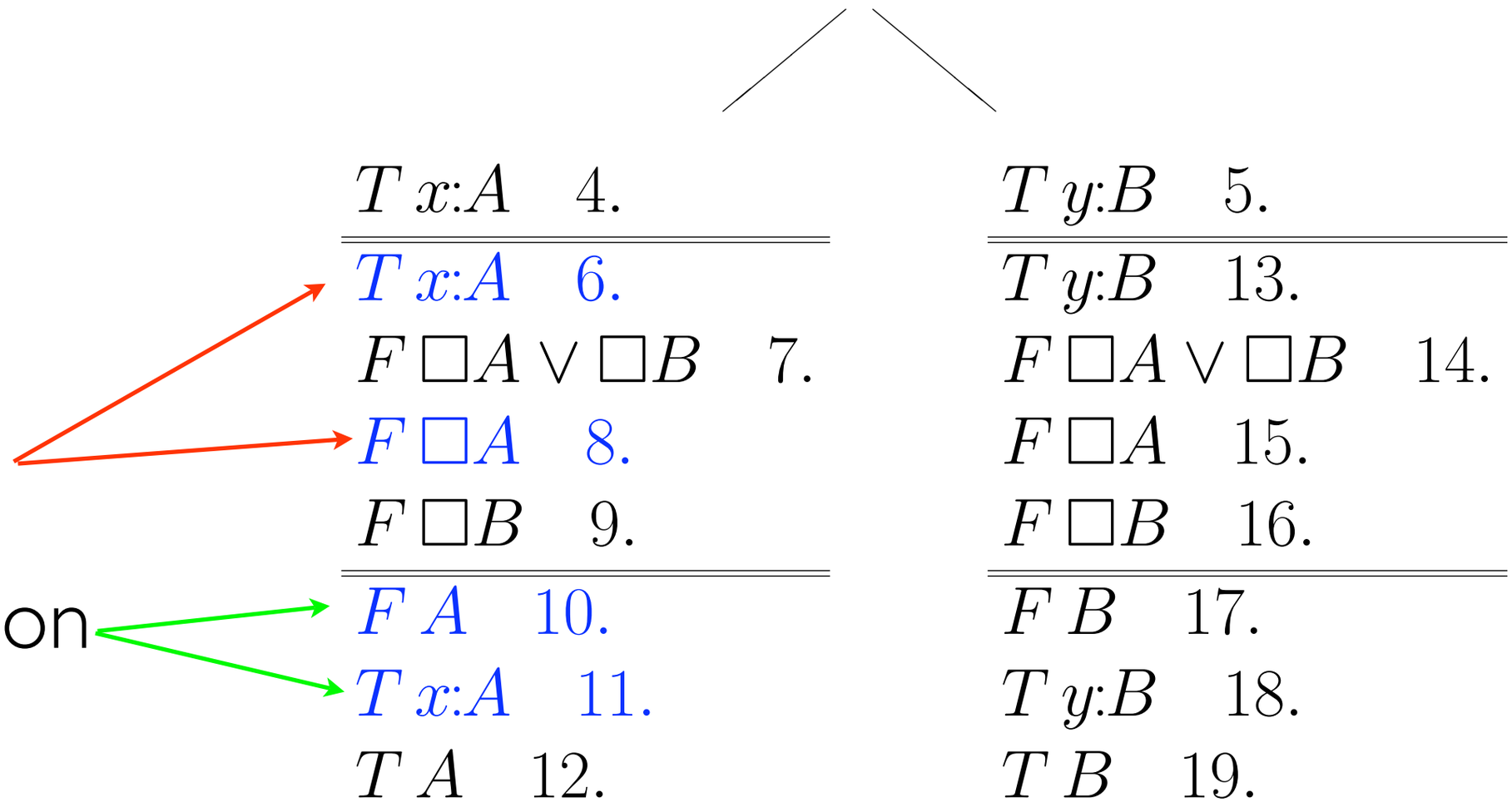
By Internalization, there is a t such that LP proves $(s_1{:}A_1 \wedge \ldots \wedge s_n{:}A_n) \supset t(s_1, \ldots, s_n){:}X$.

Use $t(s_1, \ldots, s_n)$ to realize $\Box$ in the $F\,\Box$ rule application.

$F\,(x{:}A \vee y{:}B) \supset \Box(\Box A \vee \Box B)$ 1.

$T\,x{:}A \vee y{:}B$ 2.

$F\,\Box(\Box A \vee \Box B)$ 3.



Turn

$$x{:}A \supset A$$

into

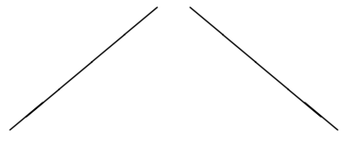
$$x{:}A \supset ?{:}A.$$

In this case, just use x. (The actual internalization algorithm produces something more complex.) Fill this in throughout the tableau.

$F\,(x{:}A \vee y{:}B) \supset \Box(x{:}A \vee \Box B)$ 1.

$T\,x{:}A \vee y{:}B$ 2.

$F\,\Box(x{:}A \vee \Box B)$ 3.



$T\,x{:}A$ 4.

$T\,x{:}A$ 6.

$F\,x{:}A \vee \Box B$ 7.

$F\,x{:}A$ 8.

$F\,\Box B$ 9.

$F\,A$ 10.

$T\,x{:}A$ 11.

$T\,A$ 12.

$T\,y{:}B$ 5.

$T\,y{:}B$ 13.

$F\,x{:}A \vee \Box B$ 14.

$F\,x{:}A$ 15.

$F\,\Box B$ 16.

$F\,B$ 17.

$T\,y{:}B$ 18.

$T\,B$ 19.

Similarly on the right branch.

$F\,(x{:}A \vee y{:}B) \supset \Box(x{:}A \vee y{:}B) \quad 1.$
$T\,x{:}A \vee y{:}B \quad 2.$
$F\,\Box(x{:}A \vee y{:}B) \quad 3.$

$T\,x{:}A \quad 4.$	$T\,y{:}B \quad 5.$
$T\,x{:}A \quad 6.$	$T\,y{:}B \quad 13.$
$F\,x{:}A \vee y{:}B \quad 7.$	$F\,x{:}A \vee y{:}B \quad 14.$
$F\,x{:}A \quad 8.$	$F\,x{:}A \quad 15.$
$F\,y{:}B \quad 9.$	$F\,y{:}B \quad 16.$
$F\,A \quad 10.$	$F\,B \quad 17.$
$T\,x{:}A \quad 11.$	$T\,y{:}B \quad 18.$
$T\,A \quad 12.$	$T\,B \quad 19.$

Now look at rule application to 3, on *left* branch.

Originally $F\,\Box(\Box A \vee \Box B)$ (3), $T\,\Box A(4)$
became $T\,\Box A$ (6), $F\,\Box A \vee \Box B$ (7).

With our replacements the problem is, solve for ? to derive

$$x{:}A \supset ?{:}(x{:}A \vee y{:}B)$$

from

$$x{:}A \supset (x{:}A \vee y{:}B)$$

The internalization algorithm gives

$$? = (c{\cdot}!x)$$

where c is a proof constant for $x{:}A \supset (x{:}A \vee y{:}B)$

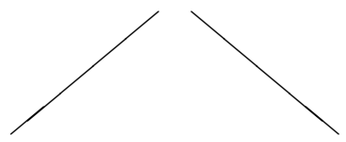
But doing the same thing on the right branch gives

$$(d{\cdot}!y)$$

where d is a proof constant for $y{:}B \supset (x{:}A \vee y{:}B)$

So we use $(c{\cdot}!x) + (d{\cdot}!y)$.

$F\,(x{:}A \vee y{:}B) \supset ((c{\cdot}!x) + (d{\cdot}!y)){:}(x{:}A \vee y{:}B) \quad 1.$
$T\,x{:}A \vee y{:}B \quad 2.$
$F\,((c{\cdot}!x) + (d{\cdot}!y)){:}(x{:}A \vee y{:}B) \quad 3.$



$T\,x{:}A \quad 4.$	$T\,y{:}B \quad 5.$
$T\,x{:}A \quad 6.$	$T\,y{:}B \quad 13.$
$F\,x{:}A \vee y{:}B \quad 7.$	$F\,x{:}A \vee y{:}B \quad 14.$
$F\,x{:}A \quad 8.$	$F\,x{:}A \quad 15.$
$F\,y{:}B \quad 9.$	$F\,y{:}B \quad 16.$
$F\,A \quad 10.$	$F\,B \quad 17.$
$T\,x{:}A \quad 11.$	$T\,y{:}B \quad 18.$
$T\,A \quad 12.$	$T\,B \quad 19.$

So, a realization for

$$(\Box A \vee \Box B) \supset \Box(\Box A \vee \Box B)$$

is

$$(x{:}A \vee y{:}B) \supset ((c{\cdot}!x) + (d{\cdot}!y)){:}(x{:}A \vee y{:}B)$$

Warning label:

This was a simplified example.
It gives the idea,
not the full proof.

Connection to Arithmetic

Translate into the language of arithmetic.
Propositional letters become arithmetic sentences.
Proof polynomials become numbers.
Operation symbols become the corresponding operations on Gödel numbers.
t:*X* becomes the arithmetic translate "*t* proves *X*"

X is a theorem of LP if and only if all translates are theorems of Peano Arithmetic

(Artemov 1995)

Gödel's Program Completed

- Intuitionistic logic embeds in S4.
- S4 embeds in LP, intuitionistic truth becomes informal provability.
- LP embeds in arithmetic, proof terms become real proofs.
- The BHK idea has become concrete.

LP Complexity

Let's begin with the familiar,
and work through to LP.

Classical

Logic	Derivability Problem is
Classical Propositional Logic	co-NP-complete

Intuitionistic

Logic	Derivability Problem is
Classical Propositional Logic	co-NP-complete
Intuitionistic Logic	PSPACE-complete

Modal

Logic	Derivability Problem is
Classical Propositional Logic	co-NP-complete
Intuitionistic Logic	PSPACE-complete
S4	PSPACE-complete

LP

Logic	Derivability Problem is
Classical Propositional Logic	co-NP-complete
Intuitionistic Logic	PSPACE-complete
S4	PSPACE-complete
LP	Π_2^p(-complete?)

The Constant Specification Matters

Constant Specification	Each Constant Proves	Derivability Problem for $\mathrm{LP}_{\mathcal{CS}}$ is	
Full	All axioms	Π_2^p	Kuznets, 2000
Schematically injective†	One scheme	Π_2^p-complete	Milnikel, 2005
Schematic	Finite number of schemes	Π_2^p	Kuznets, 2000
Empty	No axioms	co-NP-complete	folk result
Finite	Finite number of axioms	co-NP-complete	folk result
Neither finite nor schematic	decidable set of axioms	may be undecidable	Kuznets, 2004

† And axiomatically appropriate so that all axioms are evidenced.

And the LP family?

The solution of Gödel's problem
was the start of a project,
not a finish.

Just as there is a family of modal logics,
and a family of substructural logics,
there is a family of LP logics.

Common Features

All LP logics have some common features:

- Structured proof terms, proof polynomials, evidence terms.
- Internalization, so proofs using the logic have counterparts within the logic.
- Embedding in some (multi-) modal logic. Thus providing a detailed analysis of that logic's modal operator(s).

And Arithmetic?

The arithmetic completeness theorem
was an important motivation for LP.

A connection with arithmetic becomes
less important as we move further
from LP itself.

If we have it, fine, but if not, not.
Other motivations come in,
as you will see.

LP–
A Logic of Explicit Evidence

Think of LP as a logic of knowledge.
Evidence must be explicit.
A proof is one kind of evidence.
What if there are other kinds?

The Problem of Logical Omniscience

In standard logics of knowledge:

Axiom:

$$K(X \supset Y) \supset (KX \supset KY)$$

Rule:

$$\frac{X}{KX}$$

- The axiom implies knowledge is closed under consequence.

- The rule implies we know all logical truths, including huge tautologies.

- K does not represent *knowledg*e, but something more like *knowabl*e.

A Solution – Explicit Evidence

If all knowledge is knowledge for a reason,
the complexity of reasons provides
machinery to bound omniscience.

Let's use LP.

But, if LP is to be a logic,
and not just a tool,
we need a semantics.

LP Semantics

Kripke-style

Fitting (2004),

based on an earlier semantics

Mkrtychev (1997).

Frames

A *frame* is a structure $\langle \mathcal{G}, \mathcal{R} \rangle$, where $\mathcal{G}$ is a non-empty set of *states*, and $\mathcal{R}$ is a binary *accessibility* relation on $\mathcal{G}$.

Accessibility is transitive and reflexive, so we have S4 frames.

Possible Evidence

A proof polynomial counts as
possible evidence for some formulas,
but not for others, at states.

Possible evidence is not certain evidence.
Think of it as an expression of *relevance*.

Evidence Conditions

$\mathcal{E}$ is an evidence function on $\langle \mathcal{G}, \mathcal{R} \rangle$ if, for all proof polynomials s and t, for all formulas X and Y, and for all $\Gamma, \Delta \in \mathcal{G}$:

1. **Application** $(X \supset Y) \in \mathcal{E}(\Gamma, s)$ and $X \in \mathcal{E}(\Gamma, t)$ implies $Y \in \mathcal{E}(\Gamma, s \cdot t)$.
2. **Monotonicity** $\Gamma \mathcal{R} \Delta$ implies $\mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Delta, t)$.
3. **Proof Checker** $X \in \mathcal{E}(\Gamma, t)$ implies $t{:}X \in \mathcal{E}(\Gamma, !t)$.
4. **Sum** $\mathcal{E}(\Gamma, s) \cup \mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Gamma, s + t)$.

Weak Models

$\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$ is a *weak model*. ($\mathcal{V}$ maps propositional letters too sets of worlds.) Here are the truth conditions:

1. $\mathcal{M}, \Gamma \Vdash P$ for a propositional variable P provided $\Gamma \in \mathcal{V}(P)$.
2. $\mathcal{M}, \Gamma \Vdash \bot$ never holds—written $\Gamma \nVdash \bot$.
3. $\mathcal{M}, \Gamma \Vdash (X \supset Y)$ if and only if $\mathcal{M}, \Gamma \nVdash X$ or $\mathcal{M}, \Gamma \Vdash Y$.
4. $\mathcal{M}, \Gamma \Vdash (t{:}X)$ if and only if $X \in \mathcal{E}(\Gamma, t)$ and, for every $\Delta \in \mathcal{G}$ with $\Gamma \mathcal{R} \Delta$, $\mathcal{M}, \Delta \Vdash X$.

Key Condition

Condition 4 intuitively says that
t:*X* is true at a state provided
X is potentially known,
and *t* serves as possible evidence for *X*
at that state.

Constant Specifications

A *constant specification* $\mathcal{C}$ maps proof constants to sets of formulas. It is required that any formula having a proof constant with respect to $\mathcal{C}$ must be true at every possible world of every weak LP model.

$\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$ *meets the constant specification* $\mathcal{C}$ provided $\mathcal{C}(c) \subseteq \mathcal{E}(\Gamma, c)$, for each $\Gamma \in \mathcal{G}$.

Weak Completeness

Pick a constant specification, and use these constants in 'necessitation' rule applications.

X has an axiomatic proof
iff
X is valid in all weak models.

Fully Explanatory

A weak LP model $\mathcal{M}$ is *Fully Explanatory* provided that, whenever $\mathcal{M}, \Delta \Vdash X$ for every $\Delta \in \mathcal{G}$ such that $\Gamma \mathcal{R} \Delta$, then for some proof polynomial t we have $\mathcal{M}, \Gamma \Vdash (t{:}X)$.

Intuitively, if X is knowable at a state, there is a reason for X.

If $\mathcal{M}$ is a weak LP model, and if the Fully Explanatory condition is also met, then $\mathcal{M}$ is a *strong LP model*.

Strong Completeness

If the constant specification provides constants for exactly the axioms:

X is valid in all weak models
iff
X is valid in all strong models
iff
X has an axiomatic proof

The Semantics Applied

As usual in logic,
results may have both
semantic and proof-theoretic arguments.

Each provides its own insights.

The S4 Embedding

An S4 validity becomes an LP validity
on some replacement of □ occurrences
by proof polynomials.

This also has a semantic proof.
(Fitting 2004)

The proof using cut-elimination provides
an algorithm.

The semantic proof provides an
analysis of the role of +.

It turns out there is a version of
the S4 embedding result that
does not use +, but
the translation is more complex.

Embedding without +

The formula φ is fixed for now.

A is any assignment of a proof polynomial *variable* to each subformula of φ of the form $\Box X$ that is in a negative position (with different variables to different subformulas)

Define a map v_A on subformulas of φ.

If P is an atomic subformula of φ,
$v_A(P) = \{P\}$.

$v_A(X \supset Y) =$
$\{X' \supset Y' \mid X' \in v_A(X)$ and $Y' \in v_A(Y)\}$.

If $\Box X$ is a negative subformula of φ,
$v_A(\Box X) = \{x{:}X' \mid A(\Box X) = x$
and $X' \in v_A(X)\}$.

If $\Box X$ is a positive subformula of φ,
$v_A(\Box X) = \{t{:}(X_1 \vee \ldots \vee X_n) \mid$
$X_1, \ldots, X_n \in v_A(X)$
and t is any proof polynomial$\}$.

For the strong canonical model *without* $+$:

1. If ψ is a positive subformula of φ and $\mathcal{M}, \Gamma \Vdash_{\mathbf{LP}^-} \neg v_A(\psi)$ then $\mathcal{M}, \Gamma \nVdash_{\mathbf{S4}} \psi$.
2. If ψ is a negative subformula of φ and $\mathcal{M}, \Gamma \Vdash_{\mathbf{LP}^-} v_A(\psi)$ then $\mathcal{M}, \Gamma \Vdash_{\mathbf{S4}} \psi$.

Corollary If φ is a valid formula of $\mathbf{S4}$ then there are $\varphi_1, \ldots, \varphi_n \in v_A(\varphi)$ such that $\varphi_1 \vee \ldots \vee \varphi_n$ is strongly $\mathcal{C}$-LP$^-$ valid.

The converse is also true.

Define a map w_A (the Artemov embedding).

The definition is like v_A except:

If $\Box X$ is a positive subformula of φ,

$w_A(\Box X) = \{t{:}X' \mid X' \in w_A(X)$

and t is any proof polynomial$\}$

For every ψ that is a subformula of φ, and for each $\psi_1, \ldots, \psi_n \in v_A(\psi)$, there is a substitution σ and a formula $\psi' \in w_A(\psi)$ such that:

1. If ψ is a positive subformula of φ,
   $(\psi_1 \vee \ldots \vee \psi_n)\sigma \supset \psi'$ is strongly $\mathcal{C}$-LP valid.
2. If ψ is a negative subformula of φ,
   $\psi' \supset (\psi_1 \wedge \ldots \wedge \psi_n)\sigma$ is strongly $\mathcal{C}$-LP valid.

The Artemov embedding theorem
is now a corollary.

Cut Elimination

Cut Elimination itself has
a semantic proof.

Show soundness of a Gentzen
(or tableau) system with cut.

Show completeness without cut.
(Renne 2004)

Tableau Rules

A *tableau* for $F\,X$ is a tree, with $F\,X$ at the root, constructed using *branch extension rules*.

A tableau is *closed* if each branch is closed, and a branch is closed if it contains $T\,Z$ and $F\,Z$ for some formula Z, or $T\,\bot$.

A proof of X is a closed tableau for $F\,X$.

Non-Branching Rules

$$\frac{F\ X \supset Y}{\begin{array}{c} T\ X \\ F\ X \end{array}} \qquad \frac{T\ t{:}X}{T\ X} \qquad \frac{F\ !t{:}(t{:}X)}{F\ t{:}X}$$

$$\frac{F\ (s+t){:}X}{F\ s{:}X} \qquad \frac{F\ (s+t){:}X}{F\ t{:}X}$$

Branching Rules

$$\frac{T\ X \supset Y}{F\ X \mid T\ Y} \qquad \frac{F\ (t \cdot s){:}Y}{F\ t{:}X \supset Y \mid F\ s{:}X}$$

Constant Specification

Let $\mathcal{C}$ be a constant specification. I'll say X *has an LP tableau proof using* $\mathcal{C}$ if it has a proof using the machinery above together with the additional rule: a branch closes if it contains $F\, c{:}Z$ where $Z \in \mathcal{C}(c)$.

Satisfiable

Let $\mathcal{C}$ be a constant specification. A set S of signed formulas is weakly $\mathcal{C}$-LP satisfiable if there is a weak LP model $\mathcal{M}$ that meets $\mathcal{C}$, and a possible world Γ of it at which all T signed members of S are true and all F signed members are false.

$\mathcal{M}, \Gamma \Vdash X$ for all $T\,X \in S$

$\mathcal{M}, \Gamma \nVdash X$ for all $F\,X \in S$

Main Lemmas

If $\mathcal{T}$ is an LP tableau that is weakly $\mathcal{C}$-LP satisfiable, and a branch extension rule is applied to $\mathcal{T}$ to produce the tableau $\mathcal{T}'$, then $\mathcal{T}'$ is weakly $\mathcal{C}$-LP satisfiable.

If $\mathcal{T}$ is a closed tableau using $\mathcal{C}$, then $\mathcal{T}$ is not weakly $\mathcal{C}$-LP satisfiable.

Soundness

Is now by the usual argument.

Tableau Consistency

Let $\mathcal{C}$ be a constant specification. Call a set S *$\mathcal{C}$-tableau consistent* if there is no closed tableau for any finite subset of S, using $\mathcal{C}$.

Lindenbaum-type Lemma: Tableau consistent sets extend to maximal ones.

Model Construction

Construct a weak LP model candidate.

Let $\mathcal{G}$ be the collection of all maximal tableau consistent sets.

For $\Gamma, \Delta \in \mathcal{G}$, let $\Gamma \mathcal{R} \Delta$ provided $\{T\, t{:}X \mid T\, t{:}X \in \Gamma\} \subseteq \Delta$ and $\{F\, t{:}X \mid F\, t{:}X \in \Delta\} \subseteq \Gamma$, for all proof polynomials t and formulas X.

Let $X \in \mathcal{E}(\Gamma, t)$ provided $F\, t{:}X \notin \Gamma$.

Let $\Gamma \in \mathcal{V}(P)$ provided $T\, P \in \Gamma$, for a propositional letter P.

This gives us a structure $\mathcal{M} = \langle \mathcal{G}, \mathcal{R}, \mathcal{E}, \mathcal{V} \rangle$.

Basic Properties

$\mathcal{R}$ is reflexive and transitive.

And $\mathcal{E}$ is an evidence function.

Truth Lemma

For each $\Gamma \in \mathcal{G}$

$$T\,X \in \Gamma \Longrightarrow \mathcal{M}, \Gamma \Vdash X$$

$$F\,X \in \Gamma \Longrightarrow \mathcal{M}, \Gamma \nVdash X$$

Completeness

If X does not have a tableau proof, $\{F\, X\}$ is tableau consistent. A maximal extension of this set will be a member of $\mathcal{G}$ at which X is false.

Direct LP Variations

Instead of having S4 behind the scenes,
one might have
K, T, D, K4
These are straightforward.

One might have S5.
This is a bit more complicated.
The connection with arithmetic goes away.

Current work of Eric Pacuit.

Explicit Plus Implicit Knowledge

Combine a logic having explicit justifications with a conventional logic of knowledge.

S4LP

Think of LP as a logic of knowledge
with explicit justifications.

Think of S4 as a Hintikka-style
logic of knowledge.

Combine them.
But remember, we may not have reasons
for all the things we know.

(Artemov & Nogina 2004)

S4LP Axioms

- The axioms and rules of LP
- The axioms and rules of S4
- The schema $t{:}X \supset \Box X$

(Artemov & Nogina 2004)

S4LP Semantics

Use weak LP semantics.
Understand □ in the usual Hintikka way.
Sound and complete.

A cut-free Gentzen/tableau system
also exists.

(Artemov/Fitting 2004)

S4LPN

Add negative introspection to S4LP.
But how?

$\neg\Box X \supset \Box\neg\Box X$ Too cold

$\neg(x{:}X) \supset y{:}\neg(x{:}X)$ Too hot

$\neg(x{:}X) \supset \Box\neg(x{:}X)$ Just right

Soundness, completeness, internalization
(Artemov, Nogina 2005)

EBK-systems

(Evidence Based Knowledge)

Imagine a logic of knowledge
with multiple knowers (say *n*)
and also explicit knowledge.

What sort of introspection is allowed
to knowers?
Different logics result.

The Variety

Each knower has T knowledge
or S4 knowledge
or S5 knowledge

$t{:}X \supset K_i X$

Evidence is good for everybody

Logics are called
TnLP
S4nLP
S5nLP

Semantics

A genuine hybrid.

Use a Kripke/Hintikka style model,
accessibility relations for each knower,
an accessibility relation for explicit justifications,
and an evidence function.

Individual knowers part behaves Hintikka-ish.
Explicit justification part behaves as in LP models.

Thus individual accessibility relations are
reflexive, maybe also transitive, maybe also symmetric.

Additional requirement:
each knower's accessibility relation
is contained in the
explicit justification accessibility relation.

Soundness and completeness are provable.

Internalization is provable.

(Artemov 2005)

Common Knowledge

Well-known fixpoint formula:

$$CX \equiv E(X \wedge CX)$$

where

$$EX \equiv (K_1X \wedge K_2X \wedge \ldots \wedge K_nX)$$

Every explicit justification
acts like a common knowledge operator.

$$t{:}X \equiv E(X \wedge t{:}X)$$

This is current research,
explicit common knowledge
that can be resource-bounded
via complexity of
proof polynomials

The Forgetful Projection

Recall, LP embeds in S4 by forgetting details
each *t* becomes □

Introduce into multiple-knower logic
a "fool" operator J

Anything J knows, everybody knows.
(McCarthy 1979)

TnLP embeds in Tn + S4 (for J) + *any fool knows*

S4nLP embeds in S4n + S4 (for J) + *any fool knows*

S5nLP embeds in S5n + S4 (for J) + *any fool knows*

where *any fool knows* is

$$JX \supset (K_1X \wedge \ldots \wedge K_nX)$$

QLP

Quantified Logic of Proofs

If □ is something like
there is a proof of X

why not permit quantification
and make this explicit?

(Fitting, work in progress)

For Another Talk

This will be discussed in my talk on ~~Friday~~ Thursday.

Concluding

- Work is not ending, but beginning
- There are algorithms for extracting an LP embedding of an S4 theorem, and now efficient ones (Kuznetz)

- Concerning explicit reasons + multi-agents, what about communication?
- For example, is there a good analysis of the muddy children problem with explicit reasons?

- With multi-agents, what about private reasons as well as common reasons?

- For that matter, what about reasons for believing, instead of knowing?

- What about defeasable reasons?

- What about...
- What about...
- It's time to stop. Thank you.