

**CL 2002:**

*Computational Logic*

*(Lecture 11)*

Sergei N. Artemov

December 3, 2002

*Computer Science Program  
CUNY Graduate Center*

## This lecture plan

1. BHK problem revisited: modal logic route
2. Proof Polynomials and Proof Carrying Propositions
3. Logic of Proofs
4. Internalization property
5. Realization Theorem
6. Reflective Combinatory Logic, computational model.

*BHK problem:* find the intended provability semantics of intuitionistic logic satisfying *BHK* conditions:

- a proof of  $A \wedge B$  consists of a proof of  $A$  and a proof of  $B$ ,
- a proof of  $A \vee B$  is given by presenting either a proof of  $A$  or a proof of  $B$ ,
- a proof of  $A \rightarrow B$  is a construction which, given a proof of  $A$  returns a proof of  $B$ ,
- absurdity  $\perp$  is a proposition which has no proof,  $\neg A$  is  $A \rightarrow \perp$ .

Crucial for understanding connections between computations and derivations!

The first step toward solution was made by Gödel in 1933.

Gödel (1933) reduced intuitionistic propositional logic **Int** to the classical logic with built-in provability:  $\Box F \sim F$  *is provable*

*Gödel Provability Calculus, a.k.a. S4*

1. *Classical axioms and rules*

2.  $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$  *(implicit application)*

3.  $\Box F \rightarrow F$  *(reflexivity)*

4.  $\Box F \rightarrow \Box \Box F$  *(implicit proof checker)*

5. *Internalization rule:*

$$\frac{\vdash F}{\vdash \Box F}$$

*Reflects the basic intuition of Provability as a logic operator.*

Gödel's embedding of **Int** into **S4**:

1. translate **Int**-formula  $F$  into a classical language  $\square$ :

$$tr(F) = \text{“box each subformula of } F\text{”},$$

2. test the translation in **S4**:

$$\mathbf{Int} \text{ proves } F \Leftrightarrow \mathbf{S4} \text{ proves } tr(F)$$

(Gödel (1933), McKinsey & Tarski (1948))

The mission has not been accomplished though, since

***S4** itself was left without an exact provability model*

$$\mathbf{Int} \hookrightarrow \mathbf{S4} \hookrightarrow ? \hookrightarrow \text{REAL PROOFS}$$

FMU on Provability (Gödel, 1931):

$\text{Proof}_T(X, Y) \sim$  “ $X$  is a proof of  $Y$ ”

$\text{Provable}_T(Y) = \exists X \text{Proof}_T(X, Y) \sim Y$  is provable”

“ $T$  is consistent” =  $\text{Consis } T = \neg \text{Provable}_T(\text{false})$

Reflection scheme:  $\text{Provable}_T(\phi) \rightarrow \phi$

Consistency is a special case of reflection:

$$\neg \text{Provable}_T(\text{false}) = \text{Provable}_T(\text{false}) \rightarrow \text{false}$$

Incompleteness Theorem:

$T$  does not prove  $\text{Consis } T$

Reflection is not provable:

$T$  does not prove  $\text{Provable}_T(\varphi) \rightarrow \varphi$

Corollary Gödel (1933): **S4 modality**  $\neq$  *Provable*( $\cdot$ )

Indeed,  $\Box(\Box F \rightarrow F)$  is provable in **S4**:

$\Box F \rightarrow F$  - reflexivity axiom

$\Box(\Box F \rightarrow F)$  - by Internalization rule

However, under the interpretation of  $\Box$  as *Provable* this asserts that reflection is internally provable

$$\text{Provable}_T(\text{Provable}_T(F) \rightarrow F)$$

which is false by Gödel's Incompleteness Theorem.

**Gödel's problem:** *find an exact provability semantics of S4.*

*This loophole remained open for about 60 years.*

- Source of problem: nonconstructive  $\exists$ . The premise in  $\exists x Proof(x, F) \rightarrow F$  does not provide a specific proof of  $F$ , this “ $x$ ” may well be a nonstandard number which is not a code of a derivation.
- Cure: explicit representation of proofs. Gödel, 1938/95, Jäger & Artemov 1992 suggested considering format  $t:F$  (“ $t$  is a proof of  $F$ ”) with operations instead of quantifiers on proofs. **Explicit reflection**  $Proof(t, F) \rightarrow F$  for each specific  $t$  is internally provable. Indeed, if  $Proof(t, F)$  is true, then  $t$  indeed is a proof of  $F$  and hence  $Proof(t, F) \rightarrow F$  is provable then  $Proof(t, F)$  is false therefore  $\neg Proof(t, F)$  is true and provable hence  $Proof(t, F) \rightarrow F$  is provable. This allows us to circumvent the Incompleteness Theorem here. An appropriate class of BHK style operation on proofs is needed to capture the whole of **S4**.

## Principal difficulties:

- Finding the right format explicit provability. Gödel's suggestion of 1938 remained unpublished till 1995.
- The problem was pronounced unsolvable by Montague in 1963.
- Taming Skolem functions and self-referentiality  $t:F(t)$  turned out to be technically difficult. One has to give up many stereotypes coming from close areas, such as modal and combinatory logic and  $\lambda$ -calculus, etc.
- Big distraction and big help: Provability Logic with  $\Box F \sim \text{Provable}(F)$  (Solovay, Boolos, de Jongh, Visser, Magari, Sambin, Montagna, S.A., et al.). Modal logic incompatible with **S4**. Applications mostly limited to Proof Theory. Its mathematical methods, however, helped a lot.

**Gödel's provability problem:** reduces to finding a system of proof terms that corresponds to **S4**.

*If successful yields*

- *complete logical description of provability (Gödel's problem)*
- *formalization of constructive semantics for intuitionistic logic (**BHK**-problem)*
- *a new tool in modal logics,  $\lambda$ -calculi, and their applications*
- *existential semantics for modal logic:  $\Box F = \exists p$  such that... (at last!)*
- *quantitative logic of knowledge (logical omniscience problem)*
- *much richer type systems for programming languages (referential types, coding computations in types, etc.)*
- *etc.*

Complete solution has been found recently.

## Proof Polynomials

Basis for all invariant propositional operations on proofs

**variables**  $x, y, z, \dots$       *ranging over proofs*

**constants**  $a, b, c, \dots$       *proofs of instances of logical axioms*

“.” is **application**:      *applies  $s:(F \rightarrow G)$  to  $t:F$  and returns  $(s \cdot t):G$*

“!” is **proof checking**:      *computes ! $t$  a proof of  $t:F$*

“+” is **union**:      *takes union (concatenation) of two proofs*

**Proof Carrying Propositions:**  $F ::= \perp \mid \text{Var} \mid t:F \mid F \rightarrow F \mid F \wedge F \mid F \vee F \mid \neg F$

## Logic of Proof Carrying Propositions (a.k.a. Logic of Proofs)

**LP** = classical logic + additional atoms  $p:F$ ,  
( $p$  is a proof polynomial and  $F$  is a formula)

*A0. classical axioms and rules*

*A1.  $t:(F \rightarrow G) \rightarrow (s:F \rightarrow (t \cdot s):G)$*

*(application)*

*A2.  $t:F \rightarrow F$*

*(explicit reflexivity)*

*A3.  $t:F \rightarrow !t:(t:F)$*

*(proof checker)*

*A4.  $s:F \rightarrow (s \dagger t):F, \quad t:F \rightarrow (s \dagger t):F$*

*(union)*

*R1.  $A \vdash c:A$ , where  $A \in A0-A4$ ,  $c$  is a proof constant.*

*(axiom necessitation)*

A distant relative: typed combinatory logic **CL**.

Combinatory terms have dual meaning as typed terms and as derivations in a Hilbert style proof system. Constant combinators stand for proofs of axioms:

$$\mathbf{k}^{A,B} : (A \rightarrow (B \rightarrow A)), \quad \mathbf{s}^{A,B,C} : [(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))]$$

Variables in **CL** denote unspecified proofs, the operation of application “.” corresponds to the rule *modus ponens*

$$t:(F \rightarrow G) \rightarrow (s:F \rightarrow (t \cdot s):G)$$

The whole of **CL** corresponds to a fragment of **S4** consisting only of formulas of the sort

$$\Box A_1 \wedge \dots \wedge \Box A_n \rightarrow \Box B,$$

where  $A_1, \dots, A_n, B$  do not contain modalities.

## Examples of derivations

### Derivation in **S4**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ \Box & (A \rightarrow (B \rightarrow A \wedge B)) \\ \Box & A \rightarrow \Box(B \rightarrow A \wedge B) \\ \Box & A \rightarrow (\Box B \rightarrow \Box(A \wedge B)) \\ (\Box & A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{aligned}$$

### Derivation in **LP**

## Examples of derivations

### Derivation in **S4**

$$A \rightarrow (B \rightarrow A \wedge B)$$

$$\Box(A \rightarrow (B \rightarrow A \wedge B))$$

$$\Box A \rightarrow \Box(B \rightarrow A \wedge B)$$

$$\Box A \rightarrow (\Box B \rightarrow \Box(A \wedge B))$$

$$(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$$

### Derivation in **LP**

$$A \rightarrow (B \rightarrow A \wedge B)$$

## Examples of derivations

### Derivation in **S4**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ \Box(A \rightarrow (B \rightarrow A \wedge B)) \\ \Box A \rightarrow \Box(B \rightarrow A \wedge B) \\ \Box A \rightarrow (\Box B \rightarrow \Box(A \wedge B)) \\ (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{aligned}$$

### Derivation in **LP**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ c:(A \rightarrow (B \rightarrow A \wedge B)) \end{aligned}$$

## Examples of derivations

### Derivation in **S4**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ \Box & (A \rightarrow (B \rightarrow A \wedge B)) \\ \Box & A \rightarrow \Box(B \rightarrow A \wedge B) \\ \Box & A \rightarrow (\Box B \rightarrow \Box(A \wedge B)) \\ (\Box & A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{aligned}$$

### Derivation in **LP**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ c & : (A \rightarrow (B \rightarrow A \wedge B)) \\ x & : A \rightarrow (c \cdot x) : (B \rightarrow A \wedge B) \end{aligned}$$

## Examples of derivations

### Derivation in **S4**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ \Box & (A \rightarrow (B \rightarrow A \wedge B)) \\ \Box & A \rightarrow \Box(B \rightarrow A \wedge B) \\ \Box & A \rightarrow (\Box B \rightarrow \Box(A \wedge B)) \\ & (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{aligned}$$

### Derivation in **LP**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ c & : (A \rightarrow (B \rightarrow A \wedge B)) \\ x & : A \rightarrow (c \cdot x) : (B \rightarrow A \wedge B) \\ x & : A \rightarrow (y : B \rightarrow ((c \cdot x) \cdot y) : (A \wedge B)) \end{aligned}$$

## Examples of derivations

### Derivation in **S4**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ \Box(A \rightarrow (B \rightarrow A \wedge B)) \\ \Box A \rightarrow \Box(B \rightarrow A \wedge B) \\ \Box A \rightarrow (\Box B \rightarrow \Box(A \wedge B)) \\ (\Box A \wedge \Box B) \rightarrow \Box(A \wedge B) \end{aligned}$$

### Derivation in **LP**

$$\begin{aligned} & A \rightarrow (B \rightarrow A \wedge B) \\ c:(A \rightarrow (B \rightarrow A \wedge B)) \\ x:A \rightarrow (c \cdot x):(B \rightarrow A \wedge B) \\ x:A \rightarrow (y:B \rightarrow ((c \cdot x) \cdot y):(A \wedge B)) \\ (x:A \wedge y:B) \rightarrow ((c \cdot x) \cdot y):(A \wedge B) \end{aligned}$$

This was an easy ride, straightforward from **S4**.

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$A \rightarrow A \vee B$$

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

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

$$B \rightarrow A \vee B$$

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

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

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

Derivation in **LP**

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$\begin{aligned} &A \rightarrow A \vee B \\ &\Box(A \rightarrow A \vee B) \\ &\Box A \rightarrow \Box(A \vee B) \\ &B \rightarrow A \vee B \\ &\Box(B \rightarrow A \vee B) \\ &\Box B \rightarrow \Box(A \vee B) \\ &(\Box A \vee \Box B) \rightarrow \Box(A \vee B) \end{aligned}$$

Derivation in **LP**

$$\begin{aligned} &A \rightarrow A \vee B \\ &a:(A \rightarrow A \vee B) \\ &x:A \rightarrow (a \cdot x):(A \vee B) \end{aligned}$$

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$\begin{aligned} &A \rightarrow A \vee B \\ &\Box(A \rightarrow A \vee B) \\ &\Box A \rightarrow \Box(A \vee B) \\ &B \rightarrow A \vee B \\ &\Box(B \rightarrow A \vee B) \\ &\Box B \rightarrow \Box(A \vee B) \\ &(\Box A \vee \Box B) \rightarrow \Box(A \vee B) \end{aligned}$$

Derivation in **LP**

$$\begin{aligned} &A \rightarrow A \vee B \\ &a:(A \rightarrow A \vee B) \\ &x:A \rightarrow (a \cdot x):(A \vee B) \\ &B \rightarrow A \vee B \\ &b:(B \rightarrow A \vee B) \\ &y:B \rightarrow (b \cdot y):(A \vee B) \end{aligned}$$

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$\begin{aligned} &A \rightarrow A \vee B \\ &\Box(A \rightarrow A \vee B) \\ &\Box A \rightarrow \Box(A \vee B) \\ &B \rightarrow A \vee B \\ &\Box(B \rightarrow A \vee B) \\ &\Box B \rightarrow \Box(A \vee B) \\ &(\Box A \vee \Box B) \rightarrow \Box(A \vee B) \end{aligned}$$

Derivation in **LP**

$$\begin{aligned} &A \rightarrow A \vee B \\ &a:(A \rightarrow A \vee B) \\ &x:A \rightarrow (a \cdot x):(A \vee B) \\ &B \rightarrow A \vee B \\ &b:(B \rightarrow A \vee B) \\ &y:B \rightarrow (b \cdot y):(A \vee B) \end{aligned}$$

Orange parts are different, and we cannot just repeat the corresponding **S4** step. Operation  $\vdash$  is needed!

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$\begin{aligned} &A \rightarrow A \vee B \\ &\Box(A \rightarrow A \vee B) \\ &\Box A \rightarrow \Box(A \vee B) \\ &B \rightarrow A \vee B \\ &\Box(B \rightarrow A \vee B) \\ &\Box B \rightarrow \Box(A \vee B) \\ &(\Box A \vee \Box B) \rightarrow \Box(A \vee B) \end{aligned}$$

Derivation in **LP**

$$\begin{aligned} &A \rightarrow A \vee B \\ &a:(A \rightarrow A \vee B) \\ &x:A \rightarrow (a \cdot x):(A \vee B) \quad [\rightarrow (a \cdot x + b \cdot y):(A \vee B)] \\ &B \rightarrow A \vee B \\ &b:(B \rightarrow A \vee B) \\ &y:B \rightarrow (b \cdot y):(A \vee B) \quad [\rightarrow (a \cdot x + b \cdot y):(A \vee B)] \end{aligned}$$

Examples of derivations.

Some problems on the way from **S4**:

Derivation in **S4**

$$\begin{aligned} &A \rightarrow A \vee B \\ &\Box(A \rightarrow A \vee B) \\ &\Box A \rightarrow \Box(A \vee B) \\ &B \rightarrow A \vee B \\ &\Box(B \rightarrow A \vee B) \\ &\Box B \rightarrow \Box(A \vee B) \\ &(\Box A \vee \Box B) \rightarrow \Box(A \vee B) \end{aligned}$$

Derivation in **LP**

$$\begin{aligned} &A \rightarrow A \vee B \\ &a:(A \rightarrow A \vee B) \\ &x:A \rightarrow (a \cdot x):(A \vee B) [\rightarrow (a \cdot x + b \cdot y):(A \vee B)] \\ &B \rightarrow A \vee B \\ &b:(B \rightarrow A \vee B) \\ &y:B \rightarrow (b \cdot y):(A \vee B) [\rightarrow (a \cdot x + b \cdot y):(A \vee B)] \\ &(x:A \vee y:B) \rightarrow (a \cdot x + b \cdot y):(A \vee B) \end{aligned}$$

Examples of derivations. All three operations are needed

Derivation in **S4**

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

Derivation in **LP**

Examples of derivations. All three operations are needed

### Derivation in **S4**

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

### Derivation in **LP**

$x:A \rightarrow x:A \vee y:B$   
 $a:(x:A \rightarrow x:A \vee y:B)$   
 $x:A \rightarrow !x:x:A$   
 $!x:x:A \rightarrow (a \cdot !x):(x:A \vee y:B)$   
 $x:A \rightarrow (a \cdot !x):(x:A \vee y:B)$

Examples of derivations. All three operations are needed

### Derivation in **S4**

$$\begin{aligned}
&\Box A \rightarrow \Box A \vee \Box B \\
&\Box(\Box A \rightarrow \Box A \vee \Box B) \\
&\Box A \rightarrow \Box\Box A \\
&\Box\Box A \rightarrow \Box(\Box A \vee \Box B) \\
&\Box A \rightarrow \Box(\Box A \vee \Box B) \\
&\Box B \rightarrow \Box A \vee \Box B \\
&\Box(\Box B \rightarrow \Box A \vee \Box B) \\
&\Box B \rightarrow \Box\Box B \\
&\Box\Box B \rightarrow \Box(\Box A \vee \Box B) \\
&\Box B \rightarrow \Box(\Box A \vee \Box B) \\
&\Box A \vee \Box B \rightarrow \Box(\Box A \vee \Box B)
\end{aligned}$$

### Derivation in **LP**

$$\begin{aligned}
&x:A \rightarrow x:A \vee y:B \\
&a:(x:A \rightarrow x:A \vee y:B) \\
&x:A \rightarrow !x:x:A \\
&!x:x:A \rightarrow (a \cdot !x):(x:A \vee y:B) \\
&x:A \rightarrow (a \cdot !x):(x:A \vee y:B) \\
&y:B \rightarrow x:A \vee y:B \\
&b:(y:B \rightarrow x:A \vee y:B) \\
&y:B \rightarrow !y:y:B \\
&!y:y:B \rightarrow (b \cdot !y):(x:A \vee y:B) \\
&y:B \rightarrow (b \cdot !y):(x:A \vee y:B)
\end{aligned}$$

Examples of derivations. All three operations are needed

### Derivation in **S4**

$$\begin{aligned}
 &\Box A \rightarrow \Box A \vee \Box B \\
 &\Box(\Box A \rightarrow \Box A \vee \Box B) \\
 &\Box A \rightarrow \Box\Box A \\
 &\Box\Box A \rightarrow \Box(\Box A \vee \Box B) \\
 &\Box A \rightarrow \Box(\Box A \vee \Box B) \\
 &\Box B \rightarrow \Box A \vee \Box B \\
 &\Box(\Box B \rightarrow \Box A \vee \Box B) \\
 &\Box B \rightarrow \Box\Box B \\
 &\Box\Box B \rightarrow \Box(\Box A \vee \Box B) \\
 &\Box B \rightarrow \Box(\Box A \vee \Box B) \\
 &\Box A \vee \Box B \rightarrow \Box(\Box A \vee \Box B)
 \end{aligned}$$

### Derivation in **LP**

$$\begin{aligned}
 &x:A \rightarrow x:A \vee y:B \\
 &a:(x:A \rightarrow x:A \vee y:B) \\
 &x:A \rightarrow !x:x:A \\
 &!x:x:A \rightarrow (a \cdot !x):(x:A \vee y:B) \\
 &x:A \rightarrow (a \cdot !x):(x:A \vee y:B) [\rightarrow (a \cdot !x + b \cdot !y):(\dots)] \\
 &y:B \rightarrow x:A \vee y:B \\
 &b:(y:B \rightarrow x:A \vee y:B) \\
 &y:B \rightarrow !y:y:B \\
 &!y:y:B \rightarrow (b \cdot !y):(x:A \vee y:B) \\
 &y:B \rightarrow (b \cdot !y):(x:A \vee y:B) [\rightarrow (a \cdot !x + b \cdot !y):(\dots)] \\
 &x:A \vee y:B \rightarrow (a \cdot !x + b \cdot !y):(x:A \vee y:B)
 \end{aligned}$$

## Comparing formats

Type (logic) derivation  
*(plain types - propositions)*

$$A \rightarrow B, A \vdash B$$

$\lambda$ -derivation (Curry-Howard)

$$s:(A \rightarrow B), t:A \vdash (s \cdot t):B$$

*(plain typed  $\lambda$ -terms, explicit, but no proof iterations allowed)*

Modal derivation (in **S4**)

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

*(provability iterates, but is implicit)*

Proof polynomial derivation

$$x:A \vee y:B \vdash (a \cdot !x + b \cdot !y):(x:A \vee y:B)$$

*(provability is explicit*

$$a:(x:A \rightarrow x:A \vee y:B)$$

*and iterates freely)*

$$b:(y:B \rightarrow x:A \vee y:B)$$

Substitution Lemma.

**LP** enjoys substitution of proof polynomials for proof variables and propositions for propositional variables

Lifting Lemma. *If*  $A_1, \dots, A_n, y_1:B_1, \dots, y_m:B_m \vdash F$  *then*

$$x_1:A_1, \dots, x_n:A_n, y_1:B_1, \dots, y_m:B_m \vdash t:F$$

*for some proof polynomial*  $t(x_1, \dots, x_n, y_1, \dots, y_m)$ .

Constructive Necessication rule for **LP**:

$$\frac{\vdash F}{\vdash p:F}$$

Constructive Internalization:

$$\frac{A_1, \dots, A_n \vdash B}{x_1:A_1, \dots, x_n:A_n \vdash t(x_1, \dots, x_n):B}$$

The Curry-Howard isomorphism covers only a very simple special case of this rule when  $A_1, A_2, \dots, A_n, B$  are boolean formulas, i.e. do not contain modalities.

Polymorphism: multi-conclusion proofs

Operation “+” yields multiple types:

Imagine we have  $s:A$  and  $t:B$ . Then both holds:  $(s + t):A$  and  $(s + t):B$ , i.e. term  $s + t$  has types  $A$  and  $B$ .

Suppose, we want to restrict explicit modal considerations to single-conclusion proofs only. Then we will have some weird identities like  $x:\top \rightarrow \neg x:(\top \wedge \top)$ . This one, for example has a forgetful projection  $\Box\top \rightarrow \neg\Box(\top \wedge \top)$  which is provably false in any normal modal logic.

**Realization Theorem** (S.A, 1995): **S4** proves  $F$  iff there is an assignment  $r$  of proof polynomials to all  $\square$ 's in  $F$  such that the corresponding realization  $F^r$  is derivable in **LP**.

The part “if” is straightforward: given an **LP**-derivation replace proof polynomials by empty  $\square$ 's and get a derivation in **S4**. Part “only if” is not at all easy. Let us try the “naive” approach: induction on a given derivation in **S4**. Realization of **S4** axioms is trivial. Step: *modus ponens*

$$\frac{A \rightarrow B, \quad A}{B}$$

By I.H., the premises are realizable  $(A \rightarrow B)^r$  and  $A^r$ . Since  $r$  clearly commutes with  $\rightarrow$ , we have  $A^r \rightarrow B^r$  and  $A^r$ . Therefore,

$$\frac{A^r \rightarrow B^r, \quad A^r}{B^r}$$

What is wrong with this “proof”?

Yes, you are right. In

$$\frac{A^r \rightarrow B^r, \quad A^r}{B^r}$$

those  $r$ 's in  $A^r \rightarrow B^r$  and in  $A^r$  depend on derivations in **S4** of  $A \rightarrow B$  and of  $A$  respectively, and thus are *different*. In order to make this step one has to reconcile realizations of  $A \rightarrow B$  and  $A$ . In any case, such a realization step cannot be “local” and should depend on the whole derivation tree.

*True realization algorithm uses so-called normal (i.e. cut-free) derivations in **S4**.*

**Realization Theorem** (S.A, 1995): **S4** proves  $F$  iff there is an assignment  $r$  of proof polynomials to all  $\square$ 's in  $F$  such that the corresponding realization  $F^r$  is derivable in **LP**.

The proof uses a cut-free formulation of **LP**

Cut free proof system for **S4** contains only two modal rules:

$$\frac{A, \Gamma \Rightarrow \Delta}{\Box A, \Gamma \Rightarrow \Delta} (\Box \Rightarrow)$$

and

$$\frac{\Box \Gamma \Rightarrow A}{\Box \Gamma \Rightarrow \Box A} (\Rightarrow \Box)$$

where  $(\Box\{A_1, \dots, A_n\} = \{\Box A_1, \dots, \Box A_n\})$ .

The important thing here is that polarities of  $\Box$ 's do not mix. In particular, modalities introduced by the rule  $(\Rightarrow \Box)$  are positive ones, and remain such everywhere in the derivation.

All occurrences of  $\Box$ 's in a given derivation break into *families* of related ones, positive or negative.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{\Box A \Rightarrow A} \\
 \frac{\Box A \Rightarrow \Box A}{\Box A \Rightarrow \Box A, \Box B} \\
 \frac{\Box A \Rightarrow \Box A, \Box B}{\Box A \Rightarrow \Box A \vee \Box B} \\
 \frac{\Box A \Rightarrow \Box A \vee \Box B}{\Box A \Rightarrow \Box(\Box A \vee \Box B)} \\
 \hline
 \Box A \vee \Box B \Rightarrow \Box(\Box A \vee \Box B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{\Box B \Rightarrow B} \\
 \frac{\Box B \Rightarrow \Box B}{\Box B \Rightarrow \Box A, \Box B} \\
 \frac{\Box B \Rightarrow \Box A, \Box B}{\Box B \Rightarrow \Box A \vee \Box B} \\
 \frac{\Box B \Rightarrow \Box A \vee \Box B}{\Box B \Rightarrow \Box(\Box A \vee \Box B)} \\
 \hline
 \Box A \vee \Box B \Rightarrow \Box(\Box A \vee \Box B)
 \end{array}$$

Mark each family NOT containing the rule ( $\Rightarrow \Box$ ) by a fresh proof variable.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{\Box A \Rightarrow A} \\
 \frac{\Box A \Rightarrow \Box A}{\Box A \Rightarrow \Box A, \Box B} \\
 \frac{\Box A \Rightarrow \Box A, \Box B}{\Box A \Rightarrow \Box A \vee \Box B} \\
 \frac{\Box A \Rightarrow \Box A \vee \Box B}{\Box A \Rightarrow \Box(\Box A \vee \Box B)} \\
 \hline
 \Box A \vee \Box B \Rightarrow \Box(\Box A \vee \Box B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{\Box B \Rightarrow B} \\
 \frac{\Box B \Rightarrow \Box B}{\Box B \Rightarrow \Box A, \Box B} \\
 \frac{\Box B \Rightarrow \Box A, \Box B}{\Box B \Rightarrow \Box A \vee \Box B} \\
 \frac{\Box B \Rightarrow \Box A \vee \Box B}{\Box B \Rightarrow \Box(\Box A \vee \Box B)} \\
 \hline
 \Box B \Rightarrow \Box(\Box A \vee \Box B)
 \end{array}$$

Mark each family NOT containing the rule  $(\Rightarrow \Box)$  by a fresh proof variable. Each remaining family contains  $(\Rightarrow \Box)$ .

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow A}{x:A \Rightarrow \Box A} \\
 \frac{x:A \Rightarrow \Box A, \Box B}{x:A \Rightarrow \Box A \vee \Box B} \\
 \frac{x:A \Rightarrow \Box A \vee \Box B}{x:A \Rightarrow \Box(\Box A \vee \Box B)} \\
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow B}{y:B \Rightarrow \Box B} \\
 \frac{y:B \Rightarrow \Box A, \Box B}{y:B \Rightarrow \Box A \vee \Box B} \\
 \frac{y:B \Rightarrow \Box A \vee \Box B}{y:B \Rightarrow \Box(\Box A \vee \Box B)} \\
 \hline
 x:A \vee y:B \Rightarrow \Box(\Box A \vee \Box B)
 \end{array}$$

Mark each remaining family by a sum  $u_1 + \dots + u_n$  of *provisional variables* where  $n$  is the number of rules ( $\Rightarrow \square$ ) in a given family.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow u:A}{x:A \Rightarrow u:A, v:B} \\
 \frac{x:A \Rightarrow u:A \vee v:B}{x:A \Rightarrow [w_1 + w_2]:(u:A \vee v:B)} \\
 \hline
 x:A \Rightarrow [w_1 + w_2]:(u:A \vee v:B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow v:B}{y:B \Rightarrow u:A, v:B} \\
 \frac{y:B \Rightarrow u:A \vee v:B}{y:B \Rightarrow [w_1 + w_2]:(u:A \vee v:B)} \\
 \hline
 y:B \Rightarrow [w_1 + w_2]:(u:A \vee v:B)
 \end{array}$$


---


$$x:A \vee y:B \Rightarrow [w_1 + w_2]:(u:A \vee v:B)$$

Pick a  $(\Rightarrow \square)$ -node containing no provisional variables above it, e.g.  $u$ . Evaluate  $u$  by  $s(x)$  obtained by Internalization.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow s:A}{x:A \Rightarrow s:A, v:B} \\
 \frac{x:A \Rightarrow s:A \vee v:B}{x:A \Rightarrow [w_1 + w_2]:(s:A \vee v:B)} \\
 \hline
 x:A \Rightarrow [w_1 + w_2]:(s:A \vee v:B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow v:B}{y:B \Rightarrow s:A, v:B} \\
 \frac{y:B \Rightarrow s:A \vee v:B}{y:B \Rightarrow [w_1 + w_2]:(s:A \vee v:B)} \\
 \hline
 y:B \Rightarrow [w_1 + w_2]:(s:A \vee v:B)
 \end{array}$$


---


$$x:A \vee y:B \Rightarrow [w_1 + w_2]:(s:A \vee v:B)$$

Pick a  $(\Rightarrow \square)$ -node containing no provisional variables above it, e.g.  $v$ . Evaluate  $v$  by  $t(y)$  obtained by Internalization.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow s:A}{x:A \Rightarrow s:A, t:B} \\
 \frac{x:A \Rightarrow s:A \vee t:B}{x:A \Rightarrow [w_1 + w_2]:(s:A \vee t:B)} \\
 \hline
 x:A \Rightarrow [w_1 + w_2]:(s:A \vee t:B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow t:B}{y:B \Rightarrow s:A, t:B} \\
 \frac{y:B \Rightarrow s:A \vee t:B}{y:B \Rightarrow [w_1 + w_2]:(s:A \vee t:B)} \\
 \hline
 y:B \Rightarrow [w_1 + w_2]:(s:A \vee t:B)
 \end{array}$$


---


$$x:A \vee y:B \Rightarrow [w_1 + w_2]:(s:A \vee t:B)$$

Pick a  $(\Rightarrow \square)$ -node containing no provisional variables above it, e.g.  $w_1$ . Evaluate  $w_1$  by  $r_1(x)$  obtained by Internalization.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow s:A}{x:A \Rightarrow s:A, t:B} \\
 \frac{x:A \Rightarrow s:A, t:B}{x:A \Rightarrow s:A \vee t:B} \\
 \frac{x:A \Rightarrow [r_1 + w_2]:(s:A \vee t:B)}{x:A \vee y:B \Rightarrow [r_1 + w_2]:(s:A \vee t:B)}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow t:B}{y:B \Rightarrow s:A, t:B} \\
 \frac{y:B \Rightarrow s:A, t:B}{y:B \Rightarrow s:A \vee t:B} \\
 \frac{y:B \Rightarrow [r_1 + w_2]:(s:A \vee t:B)}{y:B \Rightarrow [r_1 + w_2]:(s:A \vee t:B)}
 \end{array}$$

Pick a  $(\Rightarrow \square)$ -node containing no provisional variables above it, here  $w_2$ . Evaluate  $w_2$  by  $r_2(x)$  obtained by Internalization.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow s:A}{x:A \Rightarrow s:A, t:B} \\
 \frac{x:A \Rightarrow s:A, t:B}{x:A \Rightarrow s:A \vee t:B} \\
 \frac{x:A \Rightarrow [r_1 + r_2]:(s:A \vee t:B)}{x:A \vee y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)} \\
 \\
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow t:B}{y:B \Rightarrow s:A, t:B} \\
 \frac{y:B \Rightarrow s:A, t:B}{y:B \Rightarrow s:A \vee t:B} \\
 \frac{y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)}{y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)} \\
 \\
 \frac{x:A \Rightarrow [r_1 + r_2]:(s:A \vee t:B) \quad y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)}{x:A \vee y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)}
 \end{array}$$

Substitution  $[u/s(\vec{x})]$  is always possible since  $s(\vec{x})$  contains no  $u$ 's. Hence convergence, since  $u$  is no longer in the picture.

$$\begin{array}{c}
 \frac{A \Rightarrow A}{x:A \Rightarrow A} \\
 \frac{x:A \Rightarrow s:A}{x:A \Rightarrow s:A, t:B} \\
 \frac{x:A \Rightarrow s:A \vee t:B}{x:A \Rightarrow [r_1 + r_2]:(s:A \vee t:B)} \\
 \hline
 x:A \Rightarrow [r_1 + r_2]:(s:A \vee t:B)
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{B \Rightarrow B}{y:B \Rightarrow B} \\
 \frac{y:B \Rightarrow t:B}{y:B \Rightarrow s:A, t:B} \\
 \frac{y:B \Rightarrow s:A \vee t:B}{y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)} \\
 \hline
 y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)
 \end{array}$$


---


$$x:A \vee y:B \Rightarrow [r_1 + r_2]:(s:A \vee t:B)$$

## Intended provability interpretation of **LP**.

A *proof predicate* is a provably  $\Delta_1$ -formula  $Proof(x, y)$  such that for every arithmetical sentence  $\varphi$

$$\mathbf{PA} \vdash \varphi \iff \text{for some } n \in \omega \quad Proof(n, \varphi) \text{ holds}$$

A proof predicate  $Proof(x, y)$  is *normal* if

- 1) (*finiteness of proofs*) For every proof  $k$  the set  $T(k) = \{l \mid Proof(k, l)\}$  is finite. The function from  $k$  to the code of  $T(k)$  as a finite set is computable.
2. (*conjoinability of proofs*) For any natural numbers  $k$  and  $l$  there is a natural number  $n$  such that

$$T(k) \cup T(l) \subseteq T(n).$$

The conjoinability indicates that normal proof predicates are multi-conclusion ones.

For every normal proof predicate  $Proof$  there are computable functions  $\mathbf{m}(x, y)$ ,  $\mathbf{a}(x, y)$ ,  $\mathbf{c}(x)$  such that for all arithmetical formulas  $\varphi, \psi$  and all natural numbers  $k, n$  the following formulas are valid:

$$Proof(k, \varphi \rightarrow \psi) \wedge Proof(n, \varphi) \rightarrow Proof(\mathbf{m}(k, n), \psi)$$

$$Proof(k, \varphi) \rightarrow Proof(\mathbf{a}(k, n), \varphi)$$

$$Proof(n, \varphi) \rightarrow Proof(\mathbf{a}(k, n), \varphi)$$

$$Proof(k, \varphi) \rightarrow Proof(\mathbf{c}(k), Proof(k, \varphi)).$$

An arithmetical *interpretation*  $*$  of the **LP** -language has the following parameters:

- a normal proof predicate *Proof* with the functions  $\mathbf{m}(x, y)$ ,  $\mathbf{a}(x, y)$ ,  $\mathbf{c}(x)$  as above,
- an evaluation of propositional letters by sentences of arithmetic, an evaluation of proof variables and proof constants by natural numbers
- commutation conditions

$$(t \cdot s)^* = \mathbf{m}(t^*, s^*), \quad (t + s)^* = \mathbf{a}(t^*, s^*), \quad (!t)^* = \mathbf{c}(t^*),$$

$$(t:F)^* = \mathit{Proof}(t^*, F^*).$$

Under an interpretation  $*$  a proof polynomial  $t$  becomes the natural number  $t^*$ , an **LP** -formula  $F$  becomes the arithmetical sentence  $F^*$ . A formula  $(t:F)^*$  is always provably  $\Delta_1$ .

## Completeness Theorem

*The following are equivalent*

1. **LP**  $\vdash F$  with constant specification **CS**
2.  $CS^* \models F^*$  for any interpretation  $*$
3. **PA**  $\vdash CS^* \rightarrow F^*$  for any interpretation  $*$

Other adequacy theorems for **LP** (S.A., 1994-97):

Functional completeness:

*Every propositionally definable invariant operation on proofs is a proof polynomial.*

Logical Completeness:

**LP** derives all valid identities in its own language.

Corollary: Gödel's, **BHK** problems

The foundational picture now looks like this:

**Int**  $\hookrightarrow$  **S4**  $\hookrightarrow$  **LP**  $\hookrightarrow$  *REAL PROOFS*

and all these embedding are exact.

Gödel's paper of 1933 left open two problems:

- the intended semantics of Gödel's provability calculus **S4**
- the modal logic of formal provability predicate.

The latter problem was solved in 1976 by R. Solovay who showed that the modal logic **GL** (also known under the names **G**, **L**, **K4.W**, **PRL**) axiomatized all propositional properties of the formal provability predicate. The former problem has found its solution via Proof Polynomials and explicit provability.

Those two mathematical models of provability complement each other and together cover all areas of applications.

Implicit provability model: logic **GL**.

1. *Classical axioms and rules*

2.  $\Box(F \rightarrow G) \rightarrow (\Box F \rightarrow \Box G)$  *(implicit application)*

3.  $\Box(\Box F \rightarrow F) \rightarrow \Box F$  *(Löb axiom)*

4.  $\Box F \rightarrow \Box \Box F$  *(implicit proof checker)*

5. *Internalization rule:*

$$\frac{\vdash F}{\vdash \Box F}$$

*Complete with respect to interpretation*  $\Box F$  *as*  $\text{Provable}(F)$

(Solovay, 1976). Represents Incompleteness Theorem. Applications in Proof Theory.

Explicit provability model: logic **S4/ LP**.

A long desired joint logic of propositions and proofs, gives BHK semantics to intuitionistic logic. Addresses uniformly issues in **CS/AI**: logics of knowledge, verification, typed languages and theories,  $\lambda$ 's. Generalizes the Curry-Howard Isomorphism, etc.

Is not capable of representing the second Incompleteness Theorem (thus not a replacement to the Implicit Model).

## Design comments

Operation “+” is needed. There is no +-free proof polynomial  $t(x, y)$  such that  $\mathbf{LP} \vdash (x:A \vee y:B) \rightarrow t(x, y):(A \vee B)$ .

Polymorphism: modal language of provability corresponds to multi-conclusion proofs. Indeed,

$$\mathbf{LP} \vdash s:A \wedge t:B \rightarrow (s + t):A \wedge (s + t):B.$$

The logic of single-conclusion proofs has been axiomatized by V.Krupski (Moscow).

## Exercises.

Find **S4** derivations and realizations by proof polynomials for

1.  $\Box A \rightarrow \Box(B \rightarrow \Box A)$
2.  $\Box A \wedge \Box B \rightarrow \Box(\Box A \wedge \Box B)$
3.  $(\Box A \vee \Box B) \rightarrow \Box(\Box A \vee B)$
4.  $(\Box A \vee \Box \neg B) \rightarrow \Box(B \rightarrow \Box A)$

## Reflective Combinatory Logic

Rigid typing, implicational fragment of **Int** on a background.

**k**: $[A \rightarrow (B \rightarrow A)]$

*old combinator k*

**s**: $[(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))]$

*old combinator s*

**d**: $[t:F \rightarrow F]$

*DENOTATE*

**o**: $[t:(F \rightarrow G) \rightarrow (s:F \rightarrow (t \cdot s):G)]$

*OPERATING SYSTEM*

**c**: $[t:F \rightarrow !t:(t:F)]$

*CODING*

## Computational semantics.

Standard set theoretical semantics of types, e.g. functional types are interpreted as sets of total functions. Some of the objects have constructive counterparts *names*, e.g. functions - programs that compute them.  $t:F$  is interpreted as a name (program) of type  $F$ . A more pedantic eye should already figure out that  $t:F$  is rather a singleton, i.e. a single element set containing the name (program) above.

**d:** $[t:F \rightarrow F]$  - realizes a fundamental denotational correspondence *name - object*, in particular, *program - function*.

**o:** $[t:(F \rightarrow G) \rightarrow (s:F \rightarrow (t \cdot s):G)]$  represents an operating system, which maps a program  $t$  and an input  $s$  to the result  $t \cdot s$

**c:** $[t:F \rightarrow !t:(t:F)]$  maps a program into its code (alias, name, etc.).  
Examples:  $t$  is a bytecode of a function,  $!t$  - its LISP code,  $!!t$  its higher level code with an interpreter to LISP,  $!!!t$  - its file name (something like *deepblue7-12.exe*),  $!!!!t$  its coding binary number in the library of programs, etc.

The computational semantics above can be easily made formal by picking a specific system of computable functions (functionals) and their codings.