Estonian Winter School in Computer Science, 2004

Proof Polynomilas

"For it is far better to know something about everything than to know all about one thing. This universality is the best."

Blaise Pascal, *Penses*

(Lectures 5-6)

Sergei Artemov
Graduate Center of the City University of New York

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 $A\rightarrow\perp$.

Kolmogorov 1985: "The paper [1932] was written with a hope that the logic of problem solutions [SA: i.e. the logic of proofs] will eventually become a permanent chapter in a logic course."

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. $\Box(F \to G) \to (\Box F \to \Box G)$ *(implicit application)*
3. $\Box F \to F$ *(reflexivity)*
4. $\Box F \to \Box\Box F$ *(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 $\Box$:

   $tr(F)$ = *"box each subformula of* F*"*,

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

   Int *proves* $F \quad \Leftrightarrow \quad$ **S4** *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*

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

Five Minute University (FMU) on Provability (Gödel, 1931):

Proof $_T(X,Y)$ $\sim$ "X *is a proof of* Y"

Provable $_T(Y) = \exists X$ *Proof* $_T(X,Y)$ $\sim$ Y *is provable*"

"T *is consistent*" $=$ *Consis* $T = \neg$*Provable* $_T(\bot)$

Reflection scheme: *Provable* $_T(\varphi) \to \varphi$

Consistency is a special case of reflection:

$$\neg\text{Provable}_T(\bot) = \text{Provable}_T(\bot) \to \bot$$

Incompleteness Theorem:

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

Reflection is not provable:

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

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

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

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 $\exists x \mathit{Proof}(x, F) \rightarrow F$ 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, Strassen & Artemov 1992 suggested considering format $t{:}F$ ("t is a proof of F") with operations instead of quantifiers on proofs. Explicit reflection $\mathit{Proof}(t, F) \rightarrow F$ for each specific t is internally provable. Indeed, if $\mathit{Proof}(t, F)$ is true, then t indeed is a proof of F and hence $\mathit{Proof}(t, F) \rightarrow F$ is provable then $\mathit{Proof}(t, F)$ is false therefore $\neg \mathit{Proof}(t, F)$ is true and provable hence $\mathit{Proof}(t, F) \rightarrow F$ 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 λ-calculus, etc.
- A big help: Provability Logic with $\Box F \;\sim\;$ *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 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, λ-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 $x, y, z, \ldots$	*ranging over proofs*
constants $a, b, c, \ldots$	*proofs of instances of logical axioms*
"$\cdot$" is **application**:	*applies* $s{:}(F \to G)$ *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*

Logic of Propositions and 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.	$s{:}F \rightarrow (s{+}t){:}F, \quad t{:}F \rightarrow (s{+}t){:}F$	*(union)*
R1.	$\vdash c{:}A$, *where* $A \in$*A0-A4*, c *is a proof constant.*	*(axiom necessitation)*

A close 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:

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

Variables in **CL** denote unspecified proofs, the operation of application "·" corresponds to the rule *modus ponens*

$$t{:}(F \to G) \ \to (s{:}F \to (t \cdot s){:}G)$$

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

$$\Box A_1 \wedge \ldots \wedge \Box A_n \to \Box B,$$

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

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**

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**

$A \to (B \to A \wedge B)$

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

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

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

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

Derivation in **LP**

$A \to (B \to A \wedge B)$

$c{:}(A \to (B \to A \wedge B))$

Examples of derivations

Derivation in **S4**

$A \to (B \to A \wedge B)$

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

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

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

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

Derivation in **LP**

$A \to (B \to A \wedge B)$

$c{:}(A \to (B \to A \wedge B))$

$x{:}A \to (c \cdot x){:}(B \to A \wedge B)$

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**

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

$c{:}(A \rightarrow (B \rightarrow A \wedge B))$

$x{:}A \rightarrow (c \cdot x){:}(B \rightarrow A \wedge B)$

$x{:}A \rightarrow (y{:}B \rightarrow ((c \cdot x) \cdot y){:}(A \wedge B))$

Examples of derivations

Derivation in **S4**	Derivation in **LP**
$A \rightarrow (B \rightarrow A \wedge B)$	$A \rightarrow (B \rightarrow A \wedge B)$
$\Box(A \rightarrow (B \rightarrow A \wedge B))$	$c{:}(A \rightarrow (B \rightarrow A \wedge B))$
$\Box A \rightarrow \Box(B \rightarrow A \wedge B)$	$x{:}A \rightarrow (c\cdot x){:}(B \rightarrow A \wedge B)$
$\Box A \rightarrow (\Box B \rightarrow \Box(A \wedge B))$	$x{:}A \rightarrow (y{:}B \rightarrow ((c\cdot x)\cdot y){:}(A \wedge B))$
$(\Box A \wedge \Box B) \rightarrow \Box(A \wedge B)$	$(x{:}A \wedge y{:}B) \rightarrow ((c\cdot x)\cdot y){:}(A \wedge B)$

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

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \rightarrow A \vee B$	
$\Box(A \rightarrow A \vee B)$	
$\Box A \rightarrow \Box(A \vee B)$	
$B \rightarrow A \vee B$	
$\Box(B \rightarrow A \vee B)$	
$\Box B \rightarrow \Box(A \vee B)$	
$(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$	

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \to A \vee B$	$A \to A \vee B$
$\Box(A \to A \vee B)$	$a{:}(A \to A \vee B)$
$\Box A \to \Box(A \vee B)$	$x{:}A \to (a \cdot x){:}(A \vee B)$
$B \to A \vee B$	
$\Box(B \to A \vee B)$	
$\Box B \to \Box(A \vee B)$	
$(\Box A \vee \Box B) \to \Box(A \vee B)$	

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \rightarrow A \vee B$	$A \rightarrow A \vee B$
$\Box(A \rightarrow A \vee B)$	$a{:}(A \rightarrow A \vee B)$
$\Box A \rightarrow \Box(A \vee B)$	$x{:}A \rightarrow (a \cdot x){:}(A \vee B)$
$B \rightarrow A \vee B$	$B \rightarrow A \vee B$
$\Box(B \rightarrow A \vee B)$	$b{:}(B \rightarrow A \vee B)$
$\Box B \rightarrow \Box(A \vee B)$	$y{:}B \rightarrow (b \cdot y){:}(A \vee B)$
$(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$	

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \rightarrow A \vee B$	$A \rightarrow A \vee B$
$\Box(A \rightarrow A \vee B)$	$a{:}(A \rightarrow A \vee B)$
$\Box A \rightarrow \Box(A \vee B)$	$x{:}A \rightarrow (a{\cdot}x){:}(A \vee B)$
$B \rightarrow A \vee B$	$B \rightarrow A \vee B$
$\Box(B \rightarrow A \vee B)$	$b{:}(B \rightarrow A \vee B)$
$\Box B \rightarrow \Box(A \vee B)$	$y{:}B \rightarrow (b{\cdot}y){:}(A \vee B)$
$(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$	

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

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \rightarrow A \vee B$	$A \rightarrow A \vee B$
$\Box(A \rightarrow A \vee B)$	$a{:}(A \rightarrow A \vee B)$
$\Box A \rightarrow \Box(A \vee B)$	$x{:}A \rightarrow (a\cdot x){:}(A \vee B)[\rightarrow (a\cdot x + b\cdot y){:}(A \vee B)]$
$B \rightarrow A \vee B$	$B \rightarrow A \vee B$
$\Box(B \rightarrow A \vee B)$	$b{:}(B \rightarrow A \vee B)$
$\Box B \rightarrow \Box(A \vee B)$	$y{:}B \rightarrow (b\cdot y){:}(A \vee B)[\rightarrow (a\cdot x + b\cdot y){:}(A \vee B)]$
$(\Box A \vee \Box B) \rightarrow \Box(A \vee B)$	

Examples of derivations.

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

Derivation in **S4**	Derivation in **LP**
$A \to A \vee B$	$A \to A \vee B$
$\Box(A \to A \vee B)$	$a{:}(A \to A \vee B)$
$\Box A \to \Box(A \vee B)$	$x{:}A \to (a\cdot x){:}(A \vee B)[\to (a\cdot x + b\cdot y){:}(A \vee B)]$
$B \to A \vee B$	$B \to A \vee B$
$\Box(B \to A \vee B)$	$b{:}(B \to A \vee B)$
$\Box B \to \Box(A \vee B)$	$y{:}B \to (b\cdot y){:}(A \vee B)[\to (a\cdot x + b\cdot y){:}(A \vee B)]$
$(\Box A \vee \Box B) \to \Box(A \vee B)$	$(x{:}A \vee y{:}B) \to (a\cdot x + b\cdot y){:}(A \vee B)$

Examples of derivations. All three operations are needed

Derivation in **S4** Derivation in **LP**

$\Box A \to \Box A \vee \Box B$

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

$\Box A \to \Box\Box A$

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

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

$\Box B \to \Box A \vee \Box B$

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

$\Box B \to \Box\Box B$

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

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

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

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**	Derivation in **LP**
$\Box A \rightarrow \Box A \vee \Box B$	$x{:}A \rightarrow x{:}A \vee y{:}B$
$\Box(\Box A \rightarrow \Box A \vee \Box B)$	$a{:}(x{:}A \rightarrow x{:}A \vee y{:}B)$
$\Box A \rightarrow \Box\Box A$	$x{:}A \rightarrow !x{:}x{:}A$
$\Box\Box A \rightarrow \Box(\Box A \vee \Box B)$	$!x{:}x{:}A \rightarrow (a \cdot !x){:}(x{:}A \vee y{:}B)$
$\Box A \rightarrow \Box(\Box A \vee \Box B)$	$x{:}A \rightarrow (a \cdot !x){:}(x{:}A \vee y{:}B)$
$\Box B \rightarrow \Box A \vee \Box B$	$y{:}B \rightarrow x{:}A \vee y{:}B$
$\Box(\Box B \rightarrow \Box A \vee \Box B)$	$b{:}(y{:}B \rightarrow x{:}A \vee y{:}B)$
$\Box B \rightarrow \Box\Box B$	$y{:}B \rightarrow !y{:}y{:}B$
$\Box\Box B \rightarrow \Box(\Box A \vee \Box B)$	$!y{:}y{:}B \rightarrow (b \cdot !y){:}(x{:}A \vee y{:}B)$
$\Box B \rightarrow \Box(\Box A \vee \Box B)$	$y{:}B \rightarrow (b \cdot !y){:}(x{:}A \vee y{:}B)$
$\Box A \vee \Box B \rightarrow \Box(\Box A \vee \Box B)$	

Examples of derivations. All three operations are needed

Derivation in **S4**	Derivation in **LP**
$\Box A \rightarrow \Box A \vee \Box B$	$x{:}A \rightarrow x{:}A \vee y{:}B$
$\Box(\Box A \rightarrow \Box A \vee \Box B)$	$a{:}(x{:}A \rightarrow x{:}A \vee y{:}B)$
$\Box A \rightarrow \Box\Box A$	$x{:}A \rightarrow !x{:}x{:}A$
$\Box\Box A \rightarrow \Box(\Box A \vee \Box B)$	$!x{:}x{:}A \rightarrow (a\cdot !x){:}(x{:}A \vee y{:}B)$
$\Box A \rightarrow \Box(\Box A \vee \Box B)$	$x{:}A \rightarrow (a\cdot !x){:}(x{:}A \vee y{:}B)[\rightarrow (a\cdot !x + b\cdot !y){:}(\ldots)]$
$\Box B \rightarrow \Box A \vee \Box B$	$y{:}B \rightarrow x{:}A \vee y{:}B$
$\Box(\Box B \rightarrow \Box A \vee \Box B)$	$b{:}(y{:}B \rightarrow x{:}A \vee y{:}B)$
$\Box B \rightarrow \Box\Box B$	$y{:}B \rightarrow !y{:}y{:}B$
$\Box\Box B \rightarrow \Box(\Box A \vee \Box B)$	$!y{:}y{:}B \rightarrow (b\cdot !y){:}(x{:}A \vee y{:}B)$
$\Box B \rightarrow \Box(\Box A \vee \Box B)$	$y{:}B \rightarrow (b\cdot !y){:}(x{:}A \vee y{:}B)[\rightarrow (a\cdot !x + b\cdot !y){:}(\ldots)]$
$\Box A \vee \Box B \rightarrow \Box(\Box A \vee \Box B)$	$x{:}A \vee y{:}B \rightarrow (a\cdot !x + b\cdot !y){:}(x{:}A \vee y{:}B)$

Comparing formats

Type (logic) derivation $A \rightarrow B,\ A \vdash B$

(plain types - propositions)

λ-derivation (Curry-Howard) $s{:}(A \rightarrow B),\ t{:}A \vdash (s \cdot t){:}B$

(plain typed λ-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) $b{:}(y{:}B \rightarrow x{:}A \vee y{:}B)$

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: ***S4** proves F iff there is an assignment r of proof polynomials to all $\Box$'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 $\Box$'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

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

those r's in $A^r \to B^r$ and in A^r depend on derivations in **S4** of $A \to B$ and of A respectively, and thus are *different*. In order to make this step one has to reconcile realizations of $A \to B$ 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****.*

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 $t(x_1, \dots, x_n, y_1, \dots, y_n)$.

Proof. Induction on the proof of F.

Base: If F is an axiom of **LP**, then $\vdash c{:}F$ for some constant c. If F is A_i, then $x_i{:}A_i \vdash x_i{:}F$. If F is $y_j{:}B_j$, then $y_j{:}B_j \vdash !y_j{:}F$.

Induction Step. If F is obtained by *Modus Ponens* from $C \to F$ and C, then, by I.H., both $s_1{:}(C \to F)$ and $s_2{:}C$ are provable from $x_1{:}A_1, \dots, x_n{:}A_n, y_1{:}B_1, \dots, y_m{:}B_m$. Using application axiom we get $s_1 s_2{:}F$. Suppose F is obtained by Axiom Necessitation, i.e. F is $c{:}A$ for some axiom A. Using proof checking axiom get $!c{:}c{:}A$.

Example. Consider an **LP** derivation from hypotheses:

1. $A, y{:}B \vdash A$, a hypothesis
2. $A, y{:}B \vdash y{:}B$, a hypothesis
3. $A, y{:}B \vdash A \to (y{:}B \to (A \wedge y{:}B))$, an axiom
4. $A, y{:}B \vdash y{:}B \to (A \wedge y{:}B)$, by MP, from 1,3
5. $A, y{:}B \vdash A \wedge y{:}B$, by MP, from 2,4

and its step by step lifting

1. $x{:}A, y{:}B \vdash x{:}A$
2. $x{:}A, y{:}B \vdash !y{:}y{:}B$
3. $x{:}A, y{:}B \vdash c{:}(A \to (y{:}B \to (A \wedge y{:}B)))$
4. $x{:}A, y{:}B \vdash (c{\cdot}x){:}(y{:}B \to (A \wedge y{:}B))$, by axiom $c{:}(A \to \ldots) \to (x{:}A \to (c{\cdot}x){:}\ldots)$
5. $x{:}A, y{:}B \vdash (c{\cdot}x{\cdot}!y){:}(A \wedge y{:}B)$, by axiom $(c{\cdot}x){:}(y{:}B \to \ldots) \to (!y{:}y{:}B \to (c{\cdot}x{\cdot}!y){:}\ldots)$

More examples

Usually there are multiple ways of lifting. Consider a “baby example”: lifting of $x : A \vdash A$. An easy guess gives lifting $x : A \vdash x : A$. However, the general lifting algorithm above brings somewhat different result. We have to take a derivation $x{:}A \vdash A$:

1. $x{:}A \vdash x{:}A \rightarrow A$
2. $x{:}A \vdash x{:}A$
3. $x{:}A \vdash A$

and internalize it:

1. $x{:}A \vdash c{:}(x{:}A \rightarrow A)$
2. $x{:}A \vdash !x{:}x{:}A$
3. $x{:}A \vdash c{:}(x{:}A \rightarrow A) \rightarrow (!x{:}x{:}A \rightarrow (c{\cdot}!x){:}A)$
4. $x{:}A \vdash !x{:}x{:}A \rightarrow (c{\cdot}!x){:}A$
5. $x{:}A \vdash (c{\cdot}!x){:}A$

Corollaries of Lifting Lemma.

Constructive Necessication rule for **LP**:

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

Constructive Internalization:

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

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

The proof uses a cut-free formulation of **S4**, which is based on the Gentzen's system for the classical propositional logic. Here sequents are allowed to have any finite number of formulas to the left and to the right of $\Rightarrow$. Without loss of generality we also assume that sequent axioms are of the form $\perp \Rightarrow$ and $p \Rightarrow p$ for any propositional letter p. It is an easy exercise to show that an axiom $A \Rightarrow A$ for any A can be derived from atomic axioms above without using the Cut rule.

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

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 break into *families* of related ones.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{\Box A \Rightarrow A}}{\Box A \Rightarrow \Box A}}{\Box A \Rightarrow \Box A \vee \Box B}}{\Box A \Rightarrow \Box(\Box A \vee \Box B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{\Box B \Rightarrow B}}{\Box B \Rightarrow \Box B}}{\Box B \Rightarrow \Box A \vee \Box B}}{\Box B \Rightarrow \Box(\Box A \vee \Box B)}}{\Box A \vee \Box B \Rightarrow \Box(\Box A \vee \Box B)}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow \Box A}}{x{:}A \Rightarrow \Box A \vee \Box B}}{x{:}A \Rightarrow \Box(\Box A \vee \Box B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow \Box B}}{y{:}B \Rightarrow \Box A \vee \Box B}}{y{:}B \Rightarrow \Box(\Box A \vee \Box B)}}{x{:}A \vee y{:}B \Rightarrow \Box(\Box A \vee \Box B)}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow v_1{:}A}}{x{:}A \Rightarrow v_1{:}A \vee w_1{:}B}}{x{:}A \Rightarrow (u_1+u_2){:}(v_1{:}A \vee w_1{:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow w_1{:}B}}{y{:}B \Rightarrow v_1{:}A \vee w_1{:}B}}{y{:}B \Rightarrow (u_1+u_2){:}(v_1{:}A \vee w_1{:}B)}}{x{:}A \vee y{:}B \Rightarrow (u_1+u_2){:}(v_1{:}A \vee w_1{:}B)}$$

Pick a $(\Rightarrow \Box)$-node containing no undeveloped $(\Rightarrow \Box)$-nodes above it. Read $\Rightarrow$ as $\vdash$, evaluate u_i (here v) by $s(\vec{x})$ obtained by Internalization. Deposit all constant specification to *CS*. Perform a substitution $[u_i/s(\vec{x})]$ in the tree.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow v_1{:}A}}{x{:}A \Rightarrow v_1{:}A\vee w_1{:}B}}{x{:}A \Rightarrow (u_1+u_2){:}(v_1{:}A\vee w_1{:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow w_1{:}B}}{y{:}B \Rightarrow v_1{:}A\vee w_1{:}B}}{y{:}B \Rightarrow (u_1+u_2){:}(v_1{:}A\vee w_1{:}B)}}{x{:}A\vee y{:}B \Rightarrow (u_1+u_2){:}(v_1{:}A\vee w_1{:}B)}$$

Here $s(\vec{x})$ may be taken $c{\cdot}!x$ (cf. example above), where constant c satisfies specification $c{:}(x{:}A \rightarrow A)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A \vee w_1{:}B}}{x{:}A \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow w_1{:}B}}{y{:}B \Rightarrow (c{\cdot}!x){:}A \vee w_1{:}B}}{y{:}B \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)}}{x{:}A \vee y{:}B \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)}$$

Lemma: a) Substitution $[u_i/s(\vec{x})]$ is always possible, since $s(\vec{x})$ does not contain provisional variables. b) The resulting node sequent becomes derivable in **LP** (with $\Rightarrow$ read as $\vdash$).

Pick another $(\Rightarrow \Box)$-node containing no undeveloped $(\Rightarrow \Box)$-nodes above it and repeat the above procedure of replacing a provisional variable by a term containing no provisional variables.

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A \vee w_1{:}B}}{x{:}A \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{\color{red}{y{:}B \Rightarrow w_1{:}B}}}{y{:}B \Rightarrow (c{\cdot}!x){:}A \vee w_1{:}B}}{y{:}B \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)}
}{x{:}A \vee y{:}B \Rightarrow (u_1{+}u_2){:}((c{\cdot}!x){:}A \vee w_1{:}B)}
$$

Here $w_1 := d{\cdot}!y$, where $d{:}(y{:}B \rightarrow B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A}}{x{:}A \Rightarrow (c{\cdot}!x){:}A \vee (d{\cdot}!y){:}B}}{x{:}A \Rightarrow (u_1 + u_2){:}((c{\cdot}!x){:}A \vee (d{\cdot}!y){:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow (d{\cdot}!y){:}B}}{y{:}B \Rightarrow (c{\cdot}!x){:}A \vee (d{\cdot}!y){:}B}}{y{:}B \Rightarrow (u_1 + u_2){:}((c{\cdot}!x){:}A \vee (d{\cdot}!y){:}B)}}{x{:}A \vee y{:}B \Rightarrow (u_1 + u_2){:}((c{\cdot}!x){:}A \vee (d{\cdot}!y){:}B)}$$

Two more provisional variables to go. Convergence: w_1 is no longer in the picture.

Now we have to handle a provisional variable from a sum of those. Again we use lifting to evaluate $u_1 := s(x)$.

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c\cdot!x){:}A}}{x{:}A \Rightarrow (c\cdot!x){:}A \vee (d\cdot!y){:}B}}{x{:}A \Rightarrow (u_1 + u_2){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)}
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow (d\cdot!y){:}B}}{y{:}B \Rightarrow (c\cdot!x){:}A \vee (d\cdot!y){:}B}}{y{:}B \Rightarrow (u_1 + u_2){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)}
}{x{:}A \vee y{:}B \Rightarrow (u_1 + u_2){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)}
$$

Here $s(x) := e\cdot!(c\cdot!x)$, where $e{:}[(c\cdot!x){:}A\rightarrow(c\cdot!x){:}A\vee(d\cdot!y){:}B]$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c\cdot!x){:}A}}{x{:}A \Rightarrow (c\cdot!x){:}A\vee(d\cdot!y){:}B}}{x{:}A \Rightarrow (s(x)+u_2){:}((c\cdot!x){:}A\vee(d\cdot!y){:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow (d\cdot!y){:}B}}{y{:}B \Rightarrow (c\cdot!x){:}A\vee(d\cdot!y){:}B}}{y{:}B \Rightarrow (s(x)+u_2){:}((c\cdot!x){:}A\vee(d\cdot!y){:}B)}}{x{:}A\vee y{:}B \Rightarrow (s(x)+u_2){:}((c\cdot!x){:}A\vee(d\cdot!y){:}B)}$$

Note that the node just developed became provable in **LP** despite the fact that it still contains a provisional variable.

The last step of realization - variable $u_2 := t(y)$, where $t(y) = f\cdot!(d\cdot!y)$, and $f{:}[(d\cdot!y){:}B \rightarrow (c\cdot!x){:}A \vee (d\cdot!y){:}B]$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow (c\cdot!x){:}A}}{x{:}A \Rightarrow (c\cdot!x){:}A \vee (d\cdot!y){:}B}}{x{:}A \Rightarrow (s(x)+t(y)){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow (d\cdot!y){:}B}}{y{:}B \Rightarrow (c\cdot!x){:}A \vee (d\cdot!y){:}B}}{y{:}B \Rightarrow (s(x)+t(y)){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)}}{x{:}A \vee y{:}B \Rightarrow (s(x)+t(y)){:}((c\cdot!x){:}A \vee (d\cdot!y){:}B)}$$

A more economical evaluation brings a shorter realization below with constant specification $a{:}(x{:}A \rightarrow x{:}A \vee y{:}B)$, $b{:}(y{:}B \rightarrow x{:}A \vee y{:}B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{x{:}A \Rightarrow A}}{x{:}A \Rightarrow x{:}A}}{x{:}A \Rightarrow x{:}A \vee y{:}B}}{x{:}A \Rightarrow [a\cdot !x + b\cdot !y]{:}(x{:}A \vee y{:}B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \Rightarrow B}{y{:}B \Rightarrow B}}{y{:}B \Rightarrow y{:}B}}{y{:}B \Rightarrow x{:}A \vee y{:}B}}{y{:}B \Rightarrow [a!x + b\cdot !y]{:}(x{:}A \vee y{:}B)}}{x{:}A \vee y{:}B \Rightarrow [a\cdot !x + b\cdot !y]{:}(x{:}A \vee y{:}B)}$$

Intended provability interpretation of **LP**.

A *proof predicate* is a provably Δ_1-formula *Proof*(x,y) such that for every arithmetical sentence φ

$$\textbf{PA} \vdash \varphi \quad \Leftrightarrow \quad \text{for some } n \in \omega \qquad \textit{Proof}(n,\varphi) \text{ holds}$$

A proof predicate *Proof*(x,y) is *normal* if
1) (*finiteness of proofs*) For every proof k the set $T(k)=\{l \mid \textit{Proof}(k,l)\}$ 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 φ, ψ and all natural numbers k, n the following formulas are valid:

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

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

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

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

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 Δ_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:

$$\mathbf{Int} \quad \hookrightarrow \quad \mathbf{S4} \quad \hookrightarrow \quad \mathbf{LP} \quad \hookrightarrow \quad \textit{REAL PROOFS}$$

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 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, λ's. Generalizes the Curry-Howard Isomorphism, etc.

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

Other developments

0. Basic Logic of Proofs = operation free versions for major classes of proof predicates, pre-**LP** studies. Artemov, T. Strassen;

1. Functional Logic of Proofs **FLP**. V. Krupski;

2. Logical models of referential data structures. V. Krupski, Artemov;

3. Joint Logic of Proofs and Provability **LPP**. Tanya Sidon-Yavorskaya;

4. Models for **LP**. Mkrtychev;

5. Realizability of **Int** in the $+$-free fragment of **LP**. Kopylov, Fitting;

6. Explicit counterpart of modal logic **S5**. Artemov, Kazakov, Shapiro;

7. Explicit counterparts of major normal modal logics **K**, **K4**, **T**. Brezhnev;

8. Complexity questions in Logic of Proofs. Kuznets;

9. First order logic of proofs. Artemov, Sidon-Yavorskaya, Yavorsky;

10. Logic with quantifiers over proofs. Yavorsky;

11. Disjunctive Property, complexity of the reflexive fragment. N. Krupski;

12. Reflexive λ-calculus (natural deduction system for **LP**). Artemov, Alt, Bonelli;

13. Explicit provability in verification theory. Artemov;

14. Reflection in automated deduction systems. Alen, Artemov, Barzilay, Constable, Hickey, Nogin;

15. Kripke semantics for **LP**. Fitting;

16. Tableau proof system for **LP**. Renne;

17. Reflexive Combinatory Logic. Artemov;

18. Game semantics for **LP**. Paquit, Renne;

19. Epistemic Logic with justifications. Fitting, Paquit;

20. Intuitionistic Logic of Proofs. Artemov, Kanazawa, Iemhoff.

Today we consider some of those developments:
Joint logic of provability and proofs **LPP** by Sidon-Yavorskaya;
Reflexive Combinatory Logic **RCL** by S.A..

Joint Logic of Proofs and Provability **LPP.**

Two different models of probability originated in Gödel's work of 1933: explicit **S4/LP** and implicit **GL** are compatible since they are both based on the same arithmetical proof predicates. The task of merging these two models into one comprehensive system was accomplished by Tanya Sidon-Yavorskaya. The language of logic of proofs and provability **LPP** (which has a nickname **GL+LP**) is the extension of **LP**-language by the modality $\Box$ and two monadic function symbols $\Downarrow^{\Box}$ and $\Uparrow_{\Box}$ for generating proof polynomials

Axioms for Logic of Proofs and Provability **LPP** naturally split into two groups. The first group *General* describes relations between provability and proof predicates. The second group *Operation Specification* contains axioms specifying the operations on proofs.

System **LPP**$_\emptyset$.

Axioms: I. *General*

1. axioms of **GL**
2. $t{:}A \rightarrow A$
3. $t{:}A \rightarrow \Box(t{:}A)$
4. $\neg(t{:}A) \rightarrow \Box\neg(t{:}A)$

II. *Operation Specification*

5. axioms of **LP**
6. $t{:}\Box A \rightarrow (\Downarrow^{\Box} t){:}A$
7. $t{:}A \rightarrow (\Uparrow_{\Box} t){:}\Box A$

Rules: *modus ponens* and two modal rules

$$\frac{A,\ \ A \rightarrow B}{B} \qquad\qquad \frac{A}{\Box A} \qquad\qquad \frac{\Box A}{A}$$

The system **LPP** is the closure of **LPP**$_\emptyset$ by *axiom necessitation*:
c:A, *where c is a proof constant,* A is one of the axioms 1.–7.

Main results concerning **LPP**: (Tanya Sidon-Yavorskaya,1997).

Decidability, arithmetical soundness and completeness, functional completeness, internalization, etc.

Completeness Theorem

The following are equivalent
1. **LPP** $\vdash F$ *with constant specification* CS
2. **PA** $+ \mathit{CS}^* \vdash F^*$ *for any interpretation* $*$

Reflexive Combinators (S.A.).

Characteristic features of the Reflexive Combinatory Logic **RCL** are the Combinatory Logic format, a Church style rigid typing, the implicational intuitionistic logic on level 0, and the Internalization Property, which immediately captures the usual **CL** and much more.

k:$[A\to(B\to A)]$	*old combinator* **k**
s:$[(A\to(B\to C))\to((A\to B)\to(A\to C))]$	*old combinator* **s**
d:$[t{:}F \to F]$	*DENOTATE*
o:$[t{:}(F \to G) \ \to (s{:}F \to (t\cdot s){:}G)]$	*INTERPRETER*
c:$[t{:}F \ \to\ !t{:}(t{:}F)]$	*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:$[t{:}F \to F]$ - realizes a fundamental denotational correspondence *name - object*, in particular, *program - function*.

o:$[t{:}(F \to G) \ \to (s{:}F \to (t{\cdot}s){:}G)]$ represents an interpreter, which maps a program t and an input s to the result $t \cdot s$

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

Reflexive Combinatory Logic **RCL**

(Official formulation.)

Language of **RCL**, well-defined formulas of **RCL**, and derivable formulas of **RCL** are all defined by a joint induction as follows

1. Language of **RCL** contains propositional variables $p_0, p_1, \ldots$ and a connective $\rightarrow$. Each propositional formula is also an atomic formula.

2. If S and T are formulas then $S\rightarrow T$ is also a formula.

3. For each formula F we fix it own set of proof variables $x_0, x_1, \ldots$ "of type F". If x is a variable of type F, then $x{:}F$ is a formula. Here and below "$t{:}F$ is a formula" and "t is a term of type F" are identical.

4. Each axiom A1-A6 below is a formula

A1. For each formulas A and $t{:}A$ there is an axiom

$$t{:}A \rightarrow A.$$

A2. For each formula $A \rightarrow (B \rightarrow A)$ there is a constant **k** and an axiom

$$\mathbf{k}{:}(A \rightarrow (B \rightarrow A)).$$

A3. For each formula $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ there is a constant **s** and the axiom

$$\mathbf{s}{:}[(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))].$$

A4. For each formula $t{:}A \rightarrow A$ there is a constant **d** and an axiom

$$\mathbf{d}{:}(t{:}A \rightarrow A).$$

A5. If $A, B, u{:}(A \rightarrow B), v{:}A$ are formulas then $(u \cdot v){:}B$ is again a formula. For formula $u{:}(A \rightarrow B) \rightarrow (v{:}A \rightarrow (u \cdot v){:}B)$ there is a constant **o** and an axiom

$$\mathbf{o}{:}[u{:}(A \rightarrow B) \rightarrow (v{:}A \rightarrow (u \cdot v){:}B)].$$

A6. If A and $t{:}A$ are formulas then $!t{:}t{:}A$ is a formula. For formula $t{:}A \rightarrow\, !t{:}t{:}A$ there is a constant **c** and an axiom

$$\mathbf{c}{:}(t{:}A \rightarrow\, !t{:}t{:}A).$$

5. Every formula proven from a given hypotheses is a well-defined formula. *Hypotheses* — arbitrary finite multiset Γ of formulas. A derivation from Γ is defined in the usual way with the only rule *modus ponens*:

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

An obvious semantics for **RCL** is the provability one inherited from **LP**. Combinatory terms are understood as proofs in (for example) **PA** or in **HA**. A formula $t{:}F$ is interpreted as *Proof*(t, F), conbimators **k**,**s**,**d**,**o**,**c** are terms corresponding to proofs of arithmetic instances of A2-A6.

RCL can be recast in the usual typed style. Well-defined formulas become *types*, objects t formulas $t : F$ — *combinatory terms of type* F, constants – *constant combinators* of a given type, hypohteses — *context*, formulas derivable from Γ became *nonempty types in a context* Γ, *and so on*

RCL contains the implicative intuitionistic logic, the usual combinatory logic.

Example: **RCL** emulates application:

$$\frac{u{:}(A\to B),\qquad v{:}A}{(u\cdot v){:}B}.$$

1.	$u{:}(A\to B)$	*a hypothesis*
2.	$v{:}A$	*a hypothesis*
3.	$\mathbf{o}{:}[u{:}(A\to B)\to(v{:}A\to(u\cdot v){:}B)]$	5
4.	$\mathbf{o}{:}[u{:}(A\to B)\to(v{:}A\to(u\cdot v){:}B)]\to[u{:}(A\to B)\to(v{:}A\to(u\cdot v){:}B)]$	4
5.	$u{:}(A\to B)\to(v{:}A\to(u\cdot v){:}B)$	*from 3,4*
6.	$v{:}A\to(u\cdot v){:}B$	*from 1,5*
7.	$(u\cdot v){:}B$	*from 2,6*

Example of a derivation in **RCL** without hypotheses.

Let $f{:}A$ be an axiom 2-6.

1.	$f{:}A$	*an axiom*
2.	$\mathbf{c}{:}(f{:}A \rightarrow\, !f{:}f{:}A)$	6
3.	$\mathbf{c}{:}(f{:}A \rightarrow\, !f{:}f{:}A) \rightarrow (f{:}A \rightarrow\, !f{:}f{:}A)$	4
4.	$f{:}A \rightarrow\, !f{:}f{:}A$	*2,3*
5.	$!f{:}f{:}A$	*1,4*

Theorem. **RCL** enjoys Internalization: *if* $A_1, \dots, A_n \vdash B$ *then for any variables* $x_1, \dots, x_n$ *of corresponding types there is a term* $t(x_1, \dots, x_n)$ *such that* $x_1{:}A_1, \dots, x_n{:}A_n \vdash t(x_1, \dots, x_n){:}B$.

Proof. Induction on the derivation $A_1, \dots, A_n \vdash B$. Given hypotheses Γ' $= \{x_1{:}A_1, \dots, x_n{:}A_n\}$. If B is A_i then put $t := x_i$. If B is an axiom 1 then put t equal to the constant $\mathbf{d}$ of type $t{:}A \to A$ and use 4. If B is an axiom 2-6 of sort $f{:}A$ then put t equal $!f$ and then follow the last example. Since $!f{:}f{:}A$ is derivable without hypotheses it is also derivable from Γ'. Finally, let B be obtained by *modus ponens* from $A \to B$ and A. By the induction hypothesis, $\Gamma' \vdash p(\vec{x}){:}(A \to B)$ and $\Gamma' \vdash q(\vec{x}){:}A$. Put t equal to $p(\vec{x}) \cdot q(\vec{x})$ and use the first example above to show that $\Gamma' \vdash t{:}B$.

Exercise 50. Given an **S4**-derivation $x{:}A, B \vdash A \wedge B$:
1. $x{:}A, B \vdash x{:}A$, a hypothesis
2. $x{:}A, B \vdash x{:}A \rightarrow A$, an axiom
3. $x{:}A, B \vdash A$, by MP, from 1,2,
4. $x{:}A, B \vdash B$ a hypothesis
5. $x{:}A, B \vdash A \rightarrow (B \rightarrow A \wedge B)$, an axiom
6. $x{:}A, B \vdash B \rightarrow A \wedge B$, by MP, from 3,5
7. $x{:}A, B \vdash A \wedge B$, by MP, from 4,6,

find a proof polynomial $t(x, y)$ and a derivation in **LP** $x{:}A, y{:}B \vdash t(x, y){:}(A \wedge B)$.

Exercise 51. 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. $(\Box A \vee \Box\neg B) \rightarrow \Box(B \rightarrow \Box A)$

Exercise 52. Show that the following subsystem of **LP** also enjoys Internalization: Language: $\rightarrow$, $\cdot$, $!$, proof constants and variables, propositional variables.
A0. axioms and rules of implicational intuitionistic logic

A1. $t{:}(F \to G) \ \to (s{:}F \to (t{\cdot}s){:}G)$ *A3.* $t{:}F \ \to\ !t{:}(t{:}F)$
R1. $\vdash c{:}A$, *where* $A \in$*A0,A1,A3,* c *is a proof constant.*

Exercise 53. Show that **LP** is not complete with respect to the standard proof predicate "form a textbook" based on the gödel numbering. Hint: note that in the usual gödel numbering a code of a proof is always longer than codes of each of its formulas.

Exercise 54. Consider the $+$-free fragment of **LP**, the system **LP**$_{-}$. It is obviously sound w.r.t. the class $\mathcal{F}$ of single conclusion proof system (i.e. the ones where each proof proves a single theorem). Show that **LP**$_{-}$ is not complete w.r.t. $\mathcal{F}$. It suffices to find a formula P in the language of **LP**$_{-}$ such that P holds for each single conclusion proof system and fails for some multi conclusion ones.

Exercise 55. Does the Deduction Theorem hold for **LP**? for **LP**$_{-}$? for a subsystem of **LP** from **Exercise 52**?

Exercise 56. Show that no formula of sort $t{:}F$ is derivable in **LP** without using rule *R1*.

Exercise 57. Establish the Internalization for the Sidon-Yavorskaya logic of proof and provability **LPP**.

Exercise 58. Consider the Löb axiom $\Box(\Box F \rightarrow F) \rightarrow \Box F$ from **GL** and **LPP**. Find a proof polynomial $t(x)$ of **LPP** such that **LPP** $\vdash x{:}(\Box F \rightarrow F) \rightarrow t(x){:}F$.

Exercise 59. Show that the explicit version of the Löb axiom $x{:}(y{:}P \rightarrow P) \rightarrow t{:}P$ is not provable in **LPP** for any proof term t (here x, y are proof variables, P a propositional variable). Hint: Plug $\perp$ for P.

Exercise 60. It is not the case that each formula F can be lifted internally, i.e. that for each formula F there is a proof polynomial p such that $F \rightarrow p{:}F$ holds in **LP**. Show that if F is a $\wedge, \vee$ combination of formulas of sort $t{:}G$, then F can be lifted internally. Hint: induction on F.

Exercise 61. Show that a formula $\Box P$ where P is a propositional variable cannot be lifted internally in **LPP**, i.e. for no proof term t a formula $F = \Box P \rightarrow t : \Box P$ is derivable in **LPP**. Hint: Consider an interpretation $*$ which thus maps P to $0 = 1$. Then $(\Box P)^*$ is an arithmetical formula $\neg$*Consis(PA)*. Suppose the arithmetical translation of the whole F (which is $\neg$*Consis(PA)*$\rightarrow$ *Proof*$(t^*, \neg$*Consis(PA)*$))$ is derivable in **PA**. Consider a theory

T=**PA**+¬*Consis(PA)*. In this theory the formula *Proof*$(t^*, \neg$*Consis(PA)*$)$ is provable. This formula is a ground Δ_1 formula hence it is provable in **PA** itself (otherwise both **PA** and T prove its negation). Now, get a contradiction with the Second Incompleteness Theorem.

Exercise 62. Prove in **LPP** the following constructive versions of the proof checking principle: $x{:}F \rightarrow t(x){:}\Box F$ and $x{:}F \rightarrow \Box s(x){:}F$ for appropriate proof terms $t(x)$ and $s(x)$.

Exercise 63. Is $\Box F \rightarrow \Box s(x){:}F$ provable in **LPP** for an appropriate proof term $t(x)$?

Exercise 64. Derive in **LPP** a principle $t{:}F \rightarrow \Box F$. Show that axiom 3 of **LPP** is redundant.

Exercise 65. Show that **LPP** proves $x{:}\Box P \rightarrow \Box t(x){:}P$ for some proof term $t(x)$. Does **LPP** prove $\Box x{:}P \rightarrow u(x){:}\Box P$ for some proof term $u(x)$?

Exercise 66. Derive in **RCL** all axioms of the implicational intuitionistic logic.

Exercise 67. Verify soundness of **RCL** with respect to the standard provability interpretation in the intuitionistic arithmetic **HA**. Prove that if **RCL** proves F then **HA** proves F^* for each arithmetical interpretation $*$.

Exercise 68. Establish conservativity of **RCL** with respect to **Int**: if a purely propositional formula is derivable in **RCL**, then it is derivable in **Int** itself. Hint: use de Jongh's arithmetical completeness theorem: if F is not derivable in **Int** then **HA** does not prove F^* for some arithmetical interpretation $*$.

Exercise 69. Establish the Deduction Theorem for the Reflexive Combinatory Logic **RCL**.

Exercise 70. Show that $P \rightarrow t{:}P$ where P is a propositional variable is not derivable in **RCL** for any term t.

Exercise 71. Is Internalization Rule for **RCL** invertable? Let $x{:}A \vdash t(x){:}B$ hold in **RCL** and x do not occur in A, B. Could we conclude that $A \vdash B$ also holds in **RCL**?