

**Formal proof theory  $T$**  – a theory in which the human arguments about proofs and provability should be formalized.

**Requirements:**

- encodings for formulas, proofs and programs
- $Provable(x)$  – “ $x$  is provable”
- $Proof(x, y)$  – “ $x$  is a proof of  $y$ ”

**Suitable candidates:**  $T = PA, ZF, \dots$

But all of them are **VERY UNFRIENDLY** in this role:  
axioms and rules say nothing about proofs and provability.

**Improvements** – proof theoretical interfaces for  $T$ :

$Provable$  — modal provability logics (**GL/S4**)

$Proof$  — logics of proofs (**FPL/LP**)

## Verification of decision procedures.

$Decide(\ulcorner \varphi \urcorner)$       yes ( $\varphi$  is valid)  
fail

“Private” verification (for oneself):

establish     $Decide(\ulcorner \varphi \urcorner) = \text{yes} \Rightarrow Provable(\ulcorner \varphi \urcorner)$ .

“Public” verification:

construct  $\boxed{t}$  s.t.     $Decide(\ulcorner \varphi \urcorner) = \text{yes} \Rightarrow Proof(t, \ulcorner \varphi \urcorner)$ ,

$\boxed{\text{distribute } t \text{ + trusted } ProofChecker().}$

## Core proof logic language:

$p_0, p_1, \dots$  – proof variables  
 $!^1, \times^2$  – operations on proofs

$$\left. \vphantom{\begin{array}{l} p_0, p_1, \dots \\ !^1, \times^2 \end{array}} \right\} \mapsto \mathbf{Tm}$$

$S_0, S_1, \dots$  – sentence variables  
 $\neg, \vee, \wedge, \rightarrow, (- : -)$

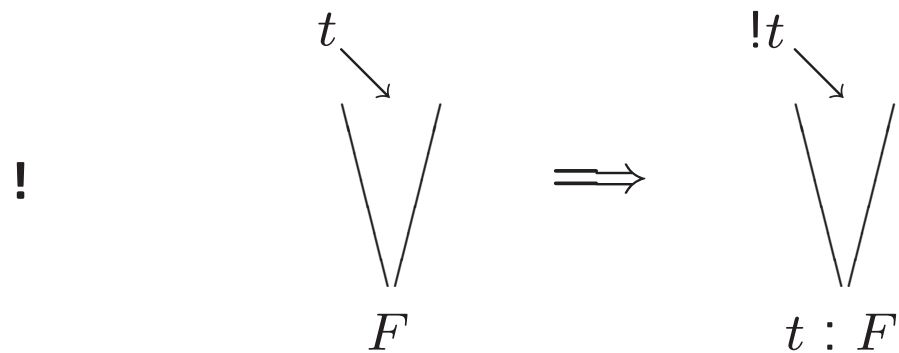
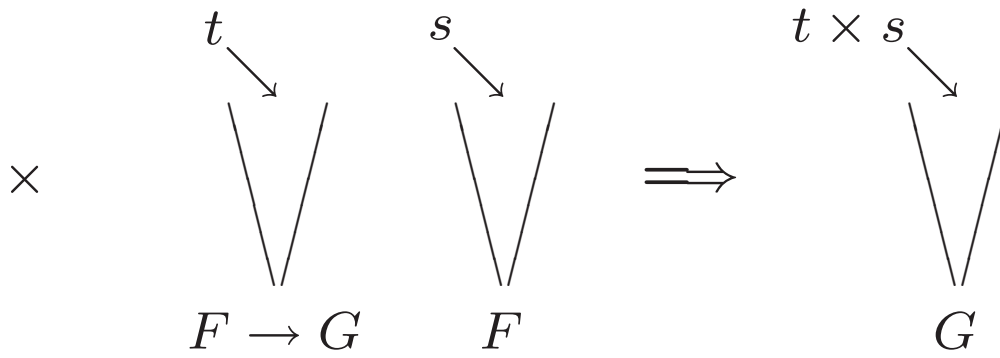
$$\left. \vphantom{\begin{array}{l} S_0, S_1, \dots \\ \neg, \vee, \wedge, \rightarrow, (- : -) \end{array}} \right\} \mapsto \mathbf{Fm}$$

$$\frac{t \in \mathbf{Tm}, \quad F \in \mathbf{Fm}}{(t : F) \in \mathbf{Fm}}$$

## Informal semantics:

$t : F$  – the arithmetical statement “ $t$  proves  $F$ ”,

$\times, !$  – act on proof codes:



Single-valued proof predicates – reflect the external derivations:

“ $x$  is a code of a derivation and  
 $y$  is the code of its last formula”

$p:F \wedge p:G \Rightarrow F = G$       How to formalize this without “=” ?

$t_1:F_1 \wedge \dots \wedge t_n:F_n \longmapsto S := \{ t_i = t_j \Rightarrow F_i = F_j \mid 1 \leq i, j \leq n \}$

Def: A unifier  $\sigma$  of  $S$  is a substitution s.t.  $t_i\sigma \neq t_j\sigma$  or  $F_i\sigma \equiv F_j\sigma$  holds for every  $i, j$ .

Def:  $A = B \pmod{S}$  iff  $A\sigma \equiv B\sigma$  for every unifier  $\sigma$  of  $S$ .

Lemma: *The relation  $A = B \pmod{S}$  is decidable.*

Unification axioms:

$t_1:F_1 \wedge \dots \wedge t_n:F_n \rightarrow (A \leftrightarrow B)$  when  $A = B \pmod{S}$ .

**System FLP** (Single-conclusion proof logic)

A0. Propositional axioms and rules

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

A2.  $t:F \rightarrow F$

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

A4. Unification axioms

**Theorem 1:** **FLP** is sound and complete w.r. to arithmetical provability interpretations based on single-valued proof predicates.

**Theorem 2:** **FLP** is decidable.

**Theorem 3:** The rule with a scheme  $\frac{F_1, \dots, F_n}{F}$  is **PA**-admissible

iff **FLP**  $\vdash F_1 \wedge \dots \wedge F_n \rightarrow F$ .

Moreover, all the operations on **PA**-derivations induced by admissible rules of this kind can be represented by proof terms (Lifting Lemma).

## Language extension by references

A **pattern** is a formula of the form  $F = F_0 \wedge p_1 : F_1 \wedge \dots \wedge p_n : F_n$  where  $p_1, \dots, p_n$  are proof letters and  $F$  does not contain any other proof letter.

Let  $x$  be a proof or sentence letter which occurs in a pattern  $F = F(x)$  and  $t$  be a proof term. A **reference**  $(x.F)_t$  denotes:

```
let  $G = \text{goal}(t)$  in
if  $t : G$  then match  $G$  with
     $F(x)$  return  $x$ 
    |  $\_$  return  $\_$ 
else  $\_$ 
```

$\text{goal}(t)$  – the last formula of the (m.b. incomplete) proof  $t$ ;  
 $\_$  denotes “any”.

Calculation of the value of  $(x.F)_t$  when  $t$  denotes a correct proof:

$t \mapsto G := \text{goal}(t) \mapsto \text{match } G \text{ with } F(x); \text{return } x$

Ex:

$\text{goal}(t) := (S. S)_t$  denotes  $G$  when  $t : G$ ;

$\text{is\_proof}(t) := t : \text{goal}(t)$  means “ $t$  is a complete proof”;

$\text{refl}(t) := (p. (p : S))_t$  denotes  $s$  when  $t : (s : F)$ ;

$\exists \bar{x}_{t:F(\bar{x})} G(\bar{x}) := F((\bar{x}. F)_t) \wedge G((\bar{x}. F)_t)$ ;

$\forall \bar{x}_{t:F(\bar{x})} G(\bar{x}) := F((\bar{x}. F)_t) \rightarrow G((\bar{x}. F)_t)$ .

$$\frac{\text{is\_proof}(p)}{\text{goal}(p)} \quad \frac{\text{is\_proof}(p)}{\text{refl}(!p) : \text{goal}(p)} \quad \frac{p : \neg \text{goal}(p)}{\perp}$$

$$\frac{\exists S_0, S_1_{p_0:(S_0 \rightarrow S_1)} p_1 : S_0}{\text{is\_proof}(p_0 p_1)} \quad \frac{\exists S_0, S_1_{p:(S_0 \rightarrow S_1)} S_0}{(S_1.S_0 \rightarrow S_1)_p}$$

System  $\mathbf{FLP}_{ref} = \mathbf{FLP} +$

A5.  $t:F(\bar{e}) \rightarrow t:F((\bar{x}.F)_t)$  where  $F = F(\bar{x})$  is a pattern,  
 $\bar{x} = (x_1, \dots, x_n)$  is a list of all proof and sentence letters from  $F$ ,  
 $\bar{e} = (e_1, \dots, e_n)$  is a list of expressions s.t.  $F(\bar{e}) \in Fm$ ,  
 $(\bar{x}.F)_t = ((x_1.F)_t, \dots, (x_n.F)_t)$ .

The scope of Unification axioms (A4) now includes references. The semantics of  $A = B \text{ (mod } S)$  relation involves **Second Order unification**, but in restricted form which still remains decidable.

Theorems 1',2',3'.  $\mathbf{FLP}_{ref}$  is decidable, sound and complete w.r. to arithmetical single-conclusion proof interpretations. It provides the same *admissibility test* for arithmetical inference rules specified by schemes in  $\mathbf{FLP}_{ref}$ -language.