

Foundations of Proof Systems

PRFSYS

Benjamin Werner — Dominik Kirst

LIX, Ecole polytechnique

Team Partout, Inria Saclay

Sept. 25th 2026

Simply typed λ -calculus

$T ::= At \mid T \rightarrow T \quad t ::= x^T \mid \lambda x^T. t \mid (t t)$

$$\frac{}{\Gamma \vdash x^A : A} \text{Var} \qquad \frac{\vdash t : B}{\Gamma \vdash \lambda x^A. t : A \rightarrow B} \text{Lam}$$
$$\frac{\vdash t : A \rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash (t u) : B} \text{App}$$

Adding sum and product types to $\lambda \rightarrow$

$$T ::= a \mid T \rightarrow T \mid T \times T \mid T + T$$

$$t ::= x \mid \lambda x.t \mid t t \mid (t, t) \mid \pi_1(t) \mid \pi_2(t) \mid i(t) \mid j(t) \mid \delta(t, x.t, x.t)$$

$$\lambda x.t \ u \triangleright_{\beta} t[x \setminus u]$$

$$\pi_1(t, u) \triangleright_{\beta} t$$

$$\pi_2(t, u) \triangleright_{\beta} u$$

$$\frac{\vdash t : A \quad \vdash u : B}{\vdash (t, u) : A \times B}$$

$$\frac{\vdash t : A \times B}{\vdash \pi_1(t) : A}$$

$$\frac{\vdash t : A \times B}{\vdash \pi_2(t) : B}$$

Adding sum and product types to $\lambda \rightarrow$ (2)

$$T ::= a \mid T \rightarrow T \mid T \times T \mid T + T$$

$$t ::= x \mid \lambda x.t \mid tt \mid (t, t) \mid \pi_1(t) \mid \pi_2(t) \mid i(t) \mid j(t) \mid \delta(t, x.t, x.t)$$

$$\lambda x.t \ u \triangleright_{\beta} t[x \setminus u]$$

$$\pi_1(t, u) \triangleright_{\beta} t$$

$$\pi_2(t, u) \triangleright_{\beta} u$$

$$\frac{\vdash t : A}{\vdash i(t) : A + B}$$

$$\frac{\vdash u : B}{\vdash j(u) : A + B}$$

$$\delta(i(t), x^A.u, y^B.v) \triangleright_{\beta} u[x^A \setminus t]$$

$$\delta(j(t), x^A.u, y^B.v) \triangleright_{\beta} v[y^B \setminus t]$$

$$\frac{\vdash t : A + B \quad \vdash u : C \quad \vdash v : C}{\vdash \delta(t, x^A.u, y^B.v) : C}$$

Normalization of $\lambda \rightarrow x +$

$$| A \times B | \equiv \{ t \in SN \mid t \triangleright^* (u, v) \Rightarrow u \in |A| \wedge v \in |B| \}$$

$$| A + B | \equiv \{ t \in SN \mid (t \triangleright^* i(u) \Rightarrow u \in |A|) \wedge (t \triangleright^* j(v) \Rightarrow v \in |B|) \}$$

$\mathcal{N}$ = terms which are not of the form: $\lambda x.t$, (u,v) , $i(u)$ or $j(v)$

In general, the metatheory follows the same schemes: inversion lemmas, type inference, subject reduction, **strong normalization**...

Heyting's semantics

To make the point of *constructivity*

- ▶ a proof of $n=n$ is 0 (some trivial object)
- ▶ a proof of $A \wedge B$ is (can be reduced to) (a,b) with $a:A$ and $b:B$
- ▶ a canonical proof of $A \vee B$ is (ε, c) with $\varepsilon=0$ and $c:A$ or $\varepsilon=1$ and $c:B$
- ▶ a proof of $A \Rightarrow B$ is a computational function f , s.t. if $a:A$, then $f(a) : B$
- ▶ a canonical proof of $\exists x.A$ is a pair (t,a) s.t. $a: A[x \setminus t]$
- ▶ a proof of $\forall x.A$ is a computational function f , s.t. for all n , $f(n) : A[x \setminus n]$

A type for every proposition

$$|P(t_1, \dots, t_n)| \equiv N$$

$$|A \wedge B| \equiv |A| \times |B|$$

$$|A \vee B| \equiv |A| + |B|$$

$$|A \Rightarrow B| \equiv |A| \rightarrow |B|$$

$$|\forall x.A| \equiv N \rightarrow |A|$$

$$|\exists x.A| \equiv N \times |A|$$

$$\Gamma = [A_1; \dots; A_n]$$

We give ourselves $x^{A_1}_1 \dots x^{A_n}_n$

To a derivation of $\Gamma \vdash A$

we associate a term $t : |A|$

with free variables $x^{A_1}_1 \dots x^{A_n}_n$

(and the free variables of the derivation)

Proofs as terms (1)

$$\Gamma = [A_1; \dots; A_n]$$

We give ourselves $x^{A_1}_1 \dots x^{A_n}_n$

To a derivation σ of $\Gamma \vdash A$ we associate a term $\|\sigma\| : |A|$

$$\text{if } A \in \Gamma \frac{}{\Gamma \vdash A} \text{Ax}$$

$$\|Ax(A)\| = x^A_k$$

$$\frac{\frac{\sigma}{\Gamma, A \vdash B}}{\Gamma \vdash A \Rightarrow B} \rightarrow\text{-I}$$

$$\|\rightarrow\text{-I}(\sigma)\| = \lambda x^A. \|\sigma\|$$

$$\frac{\frac{\sigma}{\Gamma \vdash A \rightarrow B} \quad \frac{\tau}{\Gamma \vdash A}}{\Gamma \vdash B} \rightarrow\text{-E}$$

$$\|\rightarrow\text{-E}(\sigma, \tau)\| = (\|\sigma\| \|\tau\|)$$

A logical cut

$$\begin{array}{c}
 \frac{\sigma}{\Gamma, A \vdash B} \\
 \hline
 \Gamma \vdash A \Rightarrow B
 \end{array}
 \xrightarrow{-I}
 \frac{\tau}{\Gamma \vdash A}
 \xrightarrow{-E}
 \frac{}{\Gamma \vdash B}$$

$(\lambda x^A. \|\sigma\| \|\tau\|)$

β -redex !

Cut elimination :

$$\frac{\sigma[A \searrow \tau]}{\Gamma \vdash B}$$

$(\lambda x^A. \|\sigma\| \|\tau\|) \triangleright_{\beta} \|\sigma\| [x^A \searrow \|\tau\|]$

Cut elimination and β -reduction commute

Proofs as terms (2)

$$\frac{\frac{\sigma}{\Gamma \vdash A} \quad \frac{\tau}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B} \wedge\text{-I}$$

$$||\wedge\text{-I}(\sigma, \tau)|| = (||\sigma||, ||\tau||)$$

$$\frac{\frac{\sigma}{\Gamma \vdash A \wedge B}}{\Gamma \vdash A} \wedge\text{-E (g)}$$

$$||\wedge\text{-E1}(\sigma)|| = \pi_1(||\sigma||)$$

$$\frac{\frac{\sigma}{\Gamma \vdash A \wedge B}}{\Gamma \vdash B} \wedge\text{-E (d)}$$

$$||\wedge\text{-E2}(\sigma)|| = \pi_2(||\sigma||)$$

Again, a cut is a redex $\pi_i(t, u)$

Proofs as terms (3)

$$\frac{\frac{\sigma}{\Gamma \vdash A}}{\Gamma \vdash A \vee B} \quad \vee\text{-I (g)} \quad i(||\sigma||)$$

$$\frac{\frac{\sigma}{\Gamma \vdash B}}{\Gamma \vdash A \vee B} \quad \vee\text{-I (d)} \quad j(||\sigma||)$$

$$\frac{\frac{\sigma}{\Gamma \vdash A \vee B} \quad \frac{x^A. \tau_1}{\Gamma, A \vdash C} \quad \frac{y^B. \tau_2}{\Gamma, B \vdash C}}{\Gamma \vdash C} \quad \vee\text{-E} \quad \delta(||\sigma||, x^A. ||\tau_1||, y^B. ||\tau_2||)$$

$$\frac{\Gamma \vdash A}{\Gamma \vdash \forall x. A} \quad \forall\text{-I}$$

$$\|\forall\text{-I}(\sigma)\| = \lambda x^N. \|\sigma\|$$

$$\frac{\Gamma \vdash \forall x. A}{\Gamma \vdash A[x \setminus t]} \quad \forall\text{-E}$$

$$\|\forall\text{-E}(\sigma, t)\| = (\|\sigma\| \ t)$$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x. A} \quad \exists\text{-I}$$

$$\|\exists\text{-I}(\sigma, \tau)\| = (\|\sigma\|, \|\tau\|)$$

$$\frac{\Gamma \vdash \exists x. A \quad \Gamma, A \vdash B}{\Gamma \vdash B} \quad \exists\text{-E}$$

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash A} \text{ } \perp\text{-E}$$

Intuitionistic logic

$$\frac{}{\Gamma \vdash A \vee \neg A} \text{ } \text{EM}$$

classical logic

First-Order Theories : Arithmetic

Function Symbols: 0, S, +, ×

Predicate Symbols: =

Axioms

- ▶ $\forall x, 0 + x = x$
- ▶ $\forall x \forall y, S(x) + y = S(x + y)$
- ▶ $\forall x, 0 \neq S(x)$
- ▶ $\forall x \forall y, S(x) = S(y) \Rightarrow x = y$
- ▶ $P(0) \wedge (\forall y, P(y) \Rightarrow P(S(y))) \Rightarrow \forall x, P(x)$



Giuseppe Peano
1889

- ▶ $\forall x, 0 \times x = 0$
- ▶ $\forall x \forall y, S(x) \times y = y + (x \times y)$
- ▶ $\forall x, x = x$
- ▶ $\forall x \forall y, P(x) \wedge x = y \Rightarrow P(y)$

Deduction Modulo

We add an equivalence relation $=_R$ over the propositions of FOL

We identify propositions *modulo* $=_R$ for provability

$$\text{Conv} \quad \frac{\Gamma \vdash A}{\Gamma \vdash A'} \quad \text{if } A =_R A'$$

Generally, $=_R$ will be defined through a rewrite relation R

Rewrite rules

$$0 + x \triangleright x$$

$$S(x) + y \triangleright S(x+y)$$

subsume the axioms

$$0 + x = x$$

$$S(x) + y = S(x+y)$$

A presentation of Heyting Arithmetic

Axioms

$$\forall x. x=x$$

$$\forall x. \forall y. x=y \wedge P(x) \Rightarrow P(y)$$

$$P(0) \wedge (\forall x. P(x) \Rightarrow P(S(x))) \Rightarrow \forall y. P(y)$$

closed normal object:

$0, S(0), S(S(0)), \dots$

closed normal atomic proposition

$n=m$ ($\top$ and $\perp$ are not atomic)

Rewrite rules

$$0 + x \triangleright x$$

$$S(x) + y \triangleright S(x+y)$$

$$0 \times x \triangleright 0$$

$$S(x) \times y \triangleright x \times y + y$$

$$\text{pred}(S(x)) \triangleright x \quad \text{pred}(0) \triangleright 0$$

$$\text{EQZ}(S(x)) \triangleright \perp \quad \text{EQZ}(0) \triangleright \top$$

Cuts in deduction modulo

Previous presentation: new additional rule

$$(\text{conv}) \frac{\Gamma \vdash A}{\Gamma \vdash B} \quad \text{if } A =_R B$$

we do not want it to interfere with cuts.

$$\begin{array}{c} \wedge\text{-i} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \\ \text{should be a cut} \quad (\text{conv}) \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A' \wedge B} \\ \wedge\text{-e} \frac{\Gamma \vdash A' \wedge B}{\Gamma \vdash A'} \end{array}$$

We can rather reformulate the rules:

$$\wedge\text{-i} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash C} \quad \text{if } C =_R A \wedge B$$

is now a cut

$$\wedge\text{-i} \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A' \wedge B} \\ \wedge\text{-e} \frac{\Gamma \vdash A' \wedge B}{\Gamma \vdash A'}$$

(we do the same for all rules)

Well-behaved relations

If $C =_R A \wedge B$ then either:

- C is atomic
- or $C = A' \wedge B'$ with $A =_R A'$ and $B =_R B'$

Same for $\vee, \Rightarrow, \forall, \exists$

This is the case for our presentation of Arithmetic

Then, a cut-free intuitionistic proof of $[] \vdash A$ always ends with an introduction rule

Can we generalize this for proofs in Arithmetic ?

Axiomatic Cuts

Equality Cut

"introduction"

$$\frac{}{\forall X. X=X}$$

σ_P

$$\frac{}{t=t}$$

$$\frac{}{P(t)}$$

$$\frac{}{t=t \wedge P(t)}$$

$$\frac{}{\forall x y. x=y \wedge P(x) \Rightarrow P(y)}$$

$$\frac{}{t=t \wedge P(t) \Rightarrow P(t)}$$

$$\frac{}{P(t)}$$

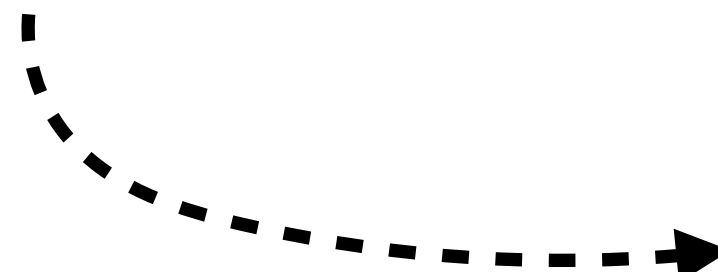
"elimination"

σ_P

$$\frac{}{P(t)}$$

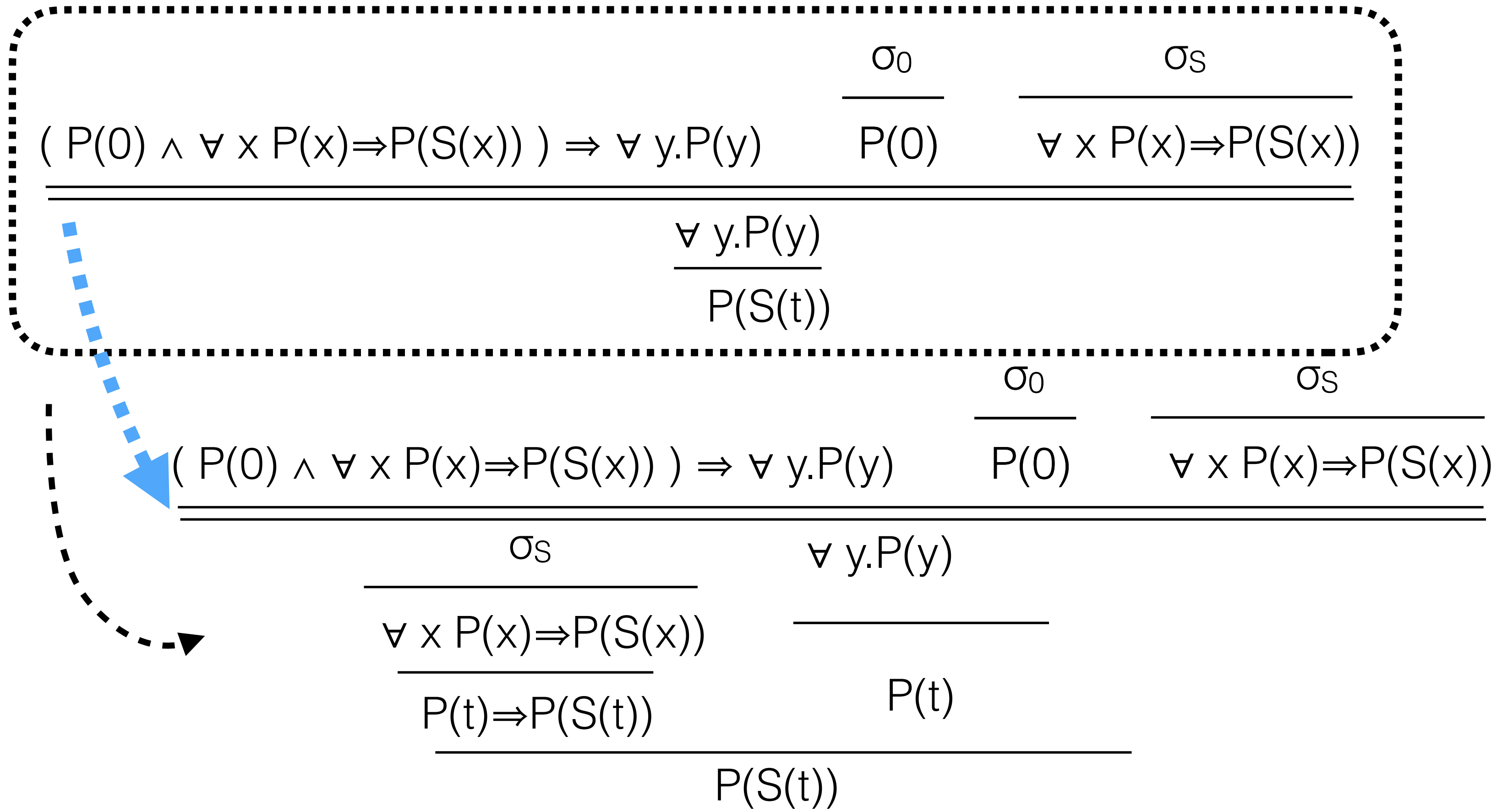
Induction Cut (1)

$$\begin{array}{c}
 \frac{
 \begin{array}{cc}
 \frac{\sigma_0}{P(0)} & \frac{\sigma_S}{\forall x P(x) \Rightarrow P(S(x))} \\
 (P(0) \wedge \forall x P(x) \Rightarrow P(S(x))) \Rightarrow \forall y.P(y) &
 \end{array}
 }{
 \frac{\forall y.P(y)}{P(0)}
 }
 \end{array}$$



$$\frac{\sigma_0}{P(0)}$$

Induction cut (2)



Properties

easy:

If t is a term without free variables, then $t \triangleright^* S^n(0)$

Cut free proofs:

Take A without free variables. Any cut-free proof of A in HA either :

- ends with an introduction
- is refl or $t=t$ (from refl)
- is Leibniz or partial application of $L : \forall y. t=y \wedge P(t) \Rightarrow P(y), u=t \wedge P(t) \Rightarrow P(u)$
- Is Induction or a partial application of it: $\forall y. P(y)$

by induction over the structure of the proof (somewhat tedious)

A without free variables. A cut-free proof of A in HA is either :

- ends with an introduction
- is refl or $t=t$ (from refl)
- is Leibniz or partial application of L : $\forall y. t=y \wedge P(t) \Rightarrow P(y), u=y \wedge P(t) \Rightarrow P(u)$
- Is Induction of proof partial application: $\forall y. P(y)$

Constructivity :

- If $\vdash_{HA} A \vee B$, then either $\vdash_{HA} A$ or $\vdash_{HA} B$
- if $\vdash_{HA} \exists x. A(x)$ then we can extract n and a proof of $\vdash_{HA} A(n)$

Consider : $\forall x. \exists y. x=y+y \vee x = S(y+y)$

Adding primitive natural numbers and a recursor

In Rocq:

```
Inductive nat : Type :=
| 0 : nat
| S : nat -> nat.
```

Additional atomic type: N

Additional constants: $0 : N, S : N \rightarrow N$

Additional operators: $R_T : N \rightarrow T \rightarrow (N \rightarrow T \rightarrow T) \rightarrow T$

$R\ 0\ t_0\ t_s \triangleright t_0$

$R\ (S\ n)\ t_0\ t_s \triangleright t_s\ n\ (R\ n\ t_0\ t_s)$

A terminating programming language

- ▶ Incomplete (no chance to have all algorithms)
- ▶ But quite expressive
- ▶ Can be made even more expressive (general inductive types, System F...)

$$\text{add} = \lambda n. \lambda m. (R_N \ n \ m \ \lambda p. \lambda r. S(r))$$

$$\text{mul} = \lambda n. \lambda m. (R_N \ n \ 0 \ \lambda p. \lambda r. (\text{add } m \ r))$$

quite straightforward for all primitive recursive functions

A non-primitive recursive function

(of course, Ackermann)

$$\text{Ack}(0, b) = S(b)$$

$$\text{Ack}(S(a), 0) = \text{Ack}(a, 1)$$

$$\text{Ack}(S(a), S(b)) = \text{Ack}(a, \text{Ack}(S(a), b))$$

Ack =

Normalization for System T

Blackboard ?

Cut Elimination for Heyting Arithmetic

Induction Cut (1)

interpret by $R_{|P|}$

$$\begin{array}{c}
 \frac{P(0) \Rightarrow (\forall x P(x) \Rightarrow P(S(x))) \Rightarrow \forall y.P(y)}{\frac{\forall y.P(y)}{P(0)}} \quad \frac{\sigma_0}{P(0)} \quad \frac{\sigma_S}{\forall x P(x) \Rightarrow P(S(x))}
 \end{array}$$

$(R \parallel \sigma_0 \parallel \parallel \sigma_S \parallel 0)$

$$\frac{\sigma_0}{P(0)}$$

Induction cut (2)

$$\begin{array}{c}
 \frac{\frac{\frac{\sigma_0}{P(0)} \quad \frac{\sigma_S}{\forall x P(x) \Rightarrow P(S(x))}}{(\ P(0) \wedge \forall x P(x) \Rightarrow P(S(x)) \) \Rightarrow \forall y.P(y)} \\
 \hline
 \frac{\forall y.P(y)}{P(S(t))}
 \end{array}$$

$$\begin{array}{c}
 \frac{\frac{\sigma_0}{P(0)} \quad \frac{\sigma_S}{\forall x P(x) \Rightarrow P(S(x))}}{(\ P(0) \wedge \forall x P(x) \Rightarrow P(S(x)) \) \Rightarrow \forall y.P(y)} \\
 \hline
 \frac{\sigma_S \quad \forall y.P(y)}{P(t) \Rightarrow P(S(t))}
 \end{array}$$

$$\begin{array}{c}
 \frac{\frac{(R \parallel \sigma_0 \parallel \parallel \sigma_S \parallel S(t)) \quad \frac{\forall x P(x) \Rightarrow P(S(x))}{P(t) \Rightarrow P(S(t))}}{\parallel \sigma_S \parallel t \ (R \parallel \sigma_0 \parallel \parallel \sigma_S \parallel t)} \quad \frac{\forall y.P(y)}{P(t)} \\
 \hline
 P(S(t))
 \end{array}$$