A Calculus of Primitive Recursive Constructions

Ludovic PATEY
PPS, Paris 7

Joint work with Hugo HERBELIN

May 12, 2014

Summary

Primitive Recursive Arithmewhat ?

- Quantifier-free formalization of natural numbers
- Finitist reasonning (Tait, 1981)
- Good metatheory for relative consistency proofs

PRIMITIVE RECURSIVE ARITHMETIC

- Talks about natural numbers
- Logic-free version with judgements of the form

$$u_0 = v_0, \ldots, u_n = v_n \vdash_{\mathrm{PRA}} u = v$$

where u_i, v_i are arithmetical expressions with free variables.

FORMAL DEFINITION

Functions:

$$
\begin{array}{rcl}
f^0, g^0 & ::= & 0^0 \\
f^1, g^1 & ::= & \sigma^1 \mid \xi^1 \mid \kappa^1_1 \mid \texttt{comp}\ f^m\ g^1_1 \ldots g^1_m \mid \texttt{rec}\ f^0\ g^2 \\
f^{n+2}, g^{n+2} & ::= & \kappa^{n+2}_m \mid \texttt{comp}\ f^m\ g^{n+2}_1 \ldots g^{n+2}_m \mid \texttt{rec}\ f^{n+1}\ g^{n+3}
\end{array}
$$

Terms:

$$u, v_i \quad ::= \quad x \mid f^n\ (v_1, \ldots, v_n)$$

Context:

$$\Gamma \quad ::= \quad \epsilon \mid \Gamma, u = v$$

RULES: BASIC AXIOMS

$$\frac{}{\Gamma \vdash_{\mathrm{PRA}} t = t} \qquad \frac{}{\Gamma \vdash_{\mathrm{PRA}} \xi^1(t) = 0^0}$$

$$\frac{1 \leq m \leq n}{\Gamma \vdash_{\mathrm{PRA}} \kappa^n_m(t_1, \ldots, t_n) = t_m} \qquad \frac{u = v \in \Gamma}{\Gamma \vdash_{\mathrm{PRA}} u = v}$$

Rules: Composition and recursion scheme

$$\frac{}{\Gamma \vdash_{\text{PRA}} (\texttt{comp}\ f^m\ g_1^n \ldots g_m^n)(t_1, \ldots, t_n) = f^m(g_1^n(t_1, \ldots, t_n), \ldots, g_m^n(t_1, \ldots, t_n))}$$

$$\frac{}{\Gamma \vdash_{\text{PRA}} (\texttt{rec}\ f^n\ g^{n+2})(0^0, t_1, \ldots, t_n) = f^n(t_1, \ldots, t_n)}$$

$$\frac{}{\Gamma \vdash_{\text{PRA}} (\texttt{rec}\ f^n\ g^{n+2})(\sigma^1(m), t_1, \ldots, t_n) = g^{n+2}(m, (\texttt{rec}\ f^n\ g^{n+2})(m, t_1, \ldots, t_n), t_1, \ldots, t_n)}$$

RULES: EQUALITY

$$\frac{\Gamma \vdash_{\mathrm{PRA}} u = v}{\Gamma \vdash_{\mathrm{PRA}} v = u} \qquad \frac{\Gamma \vdash_{\mathrm{PRA}} u = v \qquad \Gamma \vdash_{\mathrm{PRA}} v = w}{\Gamma \vdash_{\mathrm{PRA}} u = w}$$

$$\frac{\Gamma \vdash_{\mathrm{PRA}} u_1 = v_1 \qquad \dots \qquad \Gamma \vdash_{\mathrm{PRA}} u_n = v_n}{\Gamma \vdash_{\mathrm{PRA}} f^n(u_1, \dots, u_n) = f^n(v_1, \dots, v_n)}$$

RULES: INDUCTION SCHEME

$$\frac{\begin{array}{c}\Gamma \vdash_{\text{PRA}} f^{n+1}(0^0, u_1, \ldots, u_n) = g^{n+1}(0^0, u_1, \ldots, u_n) \\ \Gamma, f^{n+1}(x, u_1, \ldots, u_n) = g^{n+1}(x, u_1, \ldots, u_n) \vdash_{\text{PRA}} \\ f^{n+1}(\sigma^1(x), u_1, \ldots, u_n) = g^{n+1}(\sigma^1(x), u_1, \ldots, u_n)\end{array} \quad x \text{ fresh}}{\Gamma \vdash_{\text{PRA}} f^{n+1}(t, u_1, \ldots, u_n) = g^{n+1}(t, u_1, \ldots, u_n)}$$

Coding within PRA

- A large collection of basic structures can be coded
  - Booleans
  - Tuples
  - ...
- Easier to manipulate as primitive objects

CODING WITHIN PRA

- We need a formalism which distinguishes object's kinds
- And it is, guess what ? ....

Type theory

SUMMARY

CALCULUS OF INDUCTIVE CONSTRUCTIONS

- Distinguishes between object's kinds
- Enables to define naturally simple and complex structures

RESTRICTING CIC (PARCE QUE LE MONDE BOUGE)

Idea: restricting to objects of type

$$A_0 \to A_1 \to \cdots \to A_n$$

where A_i has no arrow.

FORMAL DEFINITION

Types:

$$A, B ::= X\ u \mid 0 \mid 1 \mid A + B \mid (\Sigma_{x:A}.B) \mid u =_A v \mid (\mu X^{A \to \texttt{Type}}\ x^A.B)\ u$$

$$C ::= A \mid \Pi_{x:A}.C$$

Terms:

$$\begin{aligned} u, v, w ::= {} & x \mid \texttt{eq}_{\text{refl}}\ A\ u \mid \texttt{match}\ u\ \texttt{return}\ A\ \texttt{end} \\ & \mid () \mid \texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ A\ \texttt{with}\ () \Rightarrow N\ \texttt{end} \\ & \mid \texttt{inl}\ u \mid \texttt{inr}\ u \mid \texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ A\ \texttt{with} \\ & \quad \texttt{inl}\ y \Rightarrow v; \texttt{inr}\ z \Rightarrow w\ \texttt{end} \\ & \mid (u, v) \mid \texttt{match}\ u\ \texttt{as}\ y\ \texttt{in}\ z\ \texttt{return}\ P \\ & \texttt{with}\ (i, w)_{x:A.T} \Rightarrow v\ \texttt{end} \mid (\texttt{fix}_k\ f\ x_1 \ldots x_n := u)\ u_1 \ldots u_n \end{aligned}$$

$$q ::= u \mid \lambda x^A.q$$

FORMAL DEFINITION

Context:

$$\Gamma \quad ::= \epsilon \mid \Gamma, x : A \mid \Gamma, X : \texttt{Type}$$

Typing rules: sum type

$$\textsc{Sum-C}\ \frac{\Gamma \vdash A : \texttt{Type} \qquad \Gamma \vdash B : \texttt{Type}}{\Gamma \vdash A + B : \texttt{Type}}$$

$$\textsc{Sum-L}\ \frac{\Gamma \vdash A + B : \texttt{Type} \qquad \Gamma \vdash t : A}{\Gamma \vdash \texttt{inl}\ t : A + B}$$

$$\textsc{Sum-R}\ \frac{\Gamma \vdash A + B : \texttt{Type} \qquad \Gamma \vdash u : B}{\Gamma \vdash \texttt{inr}\ u : A + B}$$

TYPING RULES: SUM TYPE

$$\text{Sum-E}\ \frac{\Gamma \vdash u : A + B \qquad \Gamma, y : A \vdash v : P[x \backslash \texttt{inl}\ y] \qquad \Gamma, x : A + B \vdash P : \texttt{Type} \qquad \Gamma, z : A \vdash w : P[x \backslash \texttt{inr}\ z]}{\Gamma \vdash \begin{array}{l} \texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ P\ \texttt{with} \\ \texttt{inl}\ y \Rightarrow v \\ \texttt{inr}\ z \Rightarrow w\,\texttt{end} \end{array} : P[x \backslash u]}$$

TYPING RULES: UNIT TYPE

$$\text{Unit-T}\;\frac{}{\Gamma \vdash 1 : \texttt{Type}} \qquad \text{Unit-O}\;\frac{}{\Gamma \vdash () : 1}$$

$$\text{Unit-E}\;\frac{\Gamma \vdash u : 1 \qquad \Gamma, x : 1 \vdash P : \texttt{Type} \qquad \Gamma \vdash v : P[x\backslash()]}{\Gamma \vdash \texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ P\ \texttt{with}\ () \Rightarrow v\ \texttt{end} : P[x\backslash u]}$$

TYPING RULES: EMPTY TYPE & INDUCTIVE TYPE

$$\text{Empty-T}\ \frac{}{\Gamma \vdash 0 : \texttt{Type}}$$

$$\text{Empty-E}\ \frac{\Gamma \vdash u : 0 \qquad \Gamma \vdash A : \texttt{Type}}{\Gamma \vdash \texttt{match}\ u\ \texttt{return}\ A\ \texttt{end} : A}$$

$$\text{Ind-T}\ \frac{\Gamma \vdash u : A \qquad \Gamma, X : A \to \texttt{Type}, x : A \vdash F : \texttt{Type}}{\Gamma \vdash (\mu X^{A \to \texttt{Type}}\ x^A.F)\ u : \texttt{Type}}$$

Typing rules: equality

$$\text{Eq-R}\ \frac{\Gamma \vdash u : A}{\Gamma \vdash \texttt{eq}_{\text{refl}}\ A\, u : u = u}$$

$$\text{Eq-E}\ \frac{\begin{array}{ccc} \Gamma \vdash u : A & & \Gamma, z : A \vdash P : \texttt{Type} \\ \Gamma \vdash v : A & \Gamma \vdash w : P[z \backslash u] & \Gamma \vdash p : u = v \end{array}}{\begin{array}{l} \Gamma \vdash \texttt{match}\ p\ \texttt{in}\ \_ = z\ \texttt{return}\ P\ \texttt{with} \\ \texttt{eq}_{\text{refl}} \Rightarrow w\, \texttt{end} : P[z \backslash v] \end{array}}$$

Typing rules: proper indices

$$\text{Pair-T}\ \frac{\Gamma \vdash A : \texttt{Type} \qquad \Gamma, x : A \vdash T : \texttt{Type}}{\Gamma \vdash (\Sigma_{x:A} T) : \texttt{Type}}$$

$$\text{Pair-C}\ \frac{\Gamma \vdash \Sigma_{x:A} T : \texttt{Type} \qquad \Gamma \vdash u : A \qquad \Gamma \vdash v : T[x \backslash u]}{\Gamma \vdash (u, v)_{x:A.T} : \Sigma_{x:A} T}$$

$$\text{Pair-E}\ \frac{\begin{array}{c}\Gamma \vdash \Sigma_{x:A} T \qquad \Gamma, y : \Sigma_{x:A} T \vdash P : \texttt{Type} \\ \Gamma, i : A, w : T[x \backslash i] \vdash v : P[y \backslash (i, w)_{x:A.T}]\end{array}}{\Gamma \vdash \begin{array}{l}\texttt{match}\ u\ \texttt{as}\ y\ \texttt{return}\ P\ \texttt{with} \\ (i, w)_{x:A.T} \Rightarrow v\end{array} : P[y \backslash u]}$$

SUMMARY

TYPING TRANSLATION

- There exists a natural embedding of natural integers into

$$Nat = \mu X.1 + X$$

- PRA rules are simulated by typing rules

TYPING TRANSLATION : FUNCTIONS

$$
\begin{array}{lcl}
[\![0^0]\!]() & = & \texttt{inl}\ () \\
[\![\sigma^1]\!](t) & = & \texttt{inr}\ t \\
[\![\xi^1]\!](t) & = & \texttt{inl}\ () \\
[\![\kappa^n_m]\!](t_1, \ldots, t_n) & = & t_m
\end{array}
$$

where t is a term of TT-PRA.

TYPING TRANSLATION : FUNCTIONS

$$[\![\texttt{comp}\ f^m\ g_1^n \ldots g_m^n]\!](t_1, \ldots, t_n) = \\ [\![f]\!]([\![g_1^n]\!](t_1, \ldots, t_n) \ldots [\![g_m^n]\!](t_1, \ldots, t_n))$$

$$[\![\texttt{rec}\ f^n\ g^{n+2}]\!](u, t_1, \ldots, t_n) = \\ (\texttt{fix}_1\ h\, x\, y_1 \ldots y_n := \\ \texttt{match}\ x\ \texttt{as}\ \_\ \texttt{return}\ \mathsf{Nat}\ \texttt{with} \\ \texttt{inl}\ \_ \Rightarrow [\![f^n]\!](y_1, \ldots, y_n) \\ \texttt{inr}\ z \Rightarrow [\![g^{n+2}]\!](z, (h\, z\, y_1\ \ldots\ y_n), y_1, \ldots, y_n))\ u\ t_1 \ldots t_n$$

TYPING TRANSLATION : TERMS

$$\begin{array}{rcl} [\![x]\!] & = & x \\ [\![f^n\ (v_1, \ldots, v_n)]\!] & = & [\![f^n]\!]([\![v_1]\!]\ \ldots\ [\![v_n]\!]) \end{array}$$

SOUNDNESS AND COMPLETENESS

Theorem

$$u_1 = v_1, \ldots, u_n = v_n \vdash_{PRA} u = v$$

is a valid PRA judgement iff for some proof term p,

$$x_1 : [\![u_1]\!] =_{Nat} [\![v_1]\!], \ldots, x_n : [\![u_n]\!] =_{Nat} [\![v_n]\!] \vdash_{TT} p : [\![u]\!] =_{Nat} [\![v]\!]$$

is a valid judgement.

SUMMARY

UNTYPING TRANSLATION

- Inhabitants of types in TT-PRA can be mapped to natural integers
- A type is transformed into its characteristic function
- A judgement $\vdash_{\mathrm{TT}} p : A$ becomes $\vdash_{\mathrm{PRA}} [\![A]\!]\ [\![p]\!] = 0$

SOUNDNESS

Theorem

If

$$x_1 : A_1 \ldots, x_n : A_n \vdash_{TT} p : A$$

is a valid TT-PRA judgement, then

$$[\![A_1]\!]\, x_1 = 0, \ldots, [\![A_n]\!]\, x_n = 0 \vdash_{PRA} [\![A]\!]\, [\![p]\!] = 0$$

is a valid PRA judgement.

Untyping translation : terms

$$[\![\texttt{match}\ u\ \texttt{return}\ A\ \texttt{end}]\!] = <0>$$

$$[\![()]\!] = <1>$$

$$[\![\texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ P\ \texttt{with}\ () \Rightarrow v\ \texttt{end}]\!] = [\![v]\!]$$

Untyping translation : terms

$$[\![\texttt{inl}\ u]\!] = <2, [\![u]\!]> \qquad [\![\texttt{inr}\ v]\!] = <3, [\![v]\!]>$$

$$\begin{array}{l} [\![\texttt{match}\ u\ \texttt{as}\ x\ \texttt{return}\ P\ \texttt{with} \\ \texttt{inl}\ y \Rightarrow v; \texttt{inr}\ y \Rightarrow P\,\texttt{end}]\!] = \\ \quad (\lambda z.(1 \dot{-} ((\pi_1\ z) == 2) \cdot [\![v]\!][y \backslash (\pi_2\ z)] \\ \quad +(1 \dot{-} ((\pi_1\ z) == 3) \cdot [\![P]\!][y \backslash (\pi_2\ z)])\ [\![u]\!] \end{array}$$

UNTYPING TRANSLATION : TERMS

$$[\![(u, v)]\!] = < 4, [\![u]\!], [\![v]\!] >$$

$$\begin{array}{l}[\![\texttt{match}\ u\ \texttt{as}\ z\ \texttt{in}\ y\ \texttt{return}\ P \\ \texttt{with}\ (i, w)_{x:A.T} \Rightarrow v\ \texttt{end}]\!] = \\ \quad (\lambda z.[\![v]\!][i\backslash(\pi_2\ z)][w\backslash(\pi_3\ z)])\ [\![u]\!]\end{array}$$

$$[\![\ \texttt{eq}_{\mathbf{refl}}\ A]\!] = < 5 >$$

$$\begin{array}{l}[\![\texttt{match}\ u\ \texttt{as}\ x\ \texttt{in}\ z\ \texttt{return}\ P \\ \qquad \texttt{with}\ \texttt{eq}_{\mathbf{refl}} \Rightarrow v\ \texttt{end}]\!] \quad = \quad [\![v]\!]\end{array}$$

$$[\![x]\!] = x$$

UNTYPING TRANSLATION : TYPES

$$
\begin{aligned}
[\![X\ u]\!] &= x\ [\![u]\!] \\
[\![0]\!] &= \lambda x.1 \\
[\![1]\!] &= \lambda x.(\pi_1\ x) == 1 \\
[\![A+B]\!] &= \lambda x.(((\pi_1\ x) == 2) + ([\![A]\!]\ (\pi_2\ x))) \\
&\quad \cdot(((\pi_1\ x) == 3) + ([\![B]\!]\ (\pi_2\ x))) \\
[\![(\Sigma_{x:A}.B)\ u]\!] &= \lambda y.((\pi_1\ y) == 4) + ([\![A]\!](\pi_2\ y)) \\
&\quad +([\![B]\!][x\backslash(\pi_2\ y)]\ (\pi_3\ y)) \\
[\![u = v]\!] &= \lambda x.((\pi_1\ x) == 5) + ([\![u]\!] == [\![v]\!]) \\
[\![(\mu X^{A\to \mathtt{Type}}.B)\ u]\!] &= (\mathtt{fix}_A\ x\ [\![B]\!])\ [\![u]\!]
\end{aligned}
$$

CONCLUSION

- There exists a “natural” restriction of CIC capturing PRA.
  - Collection of primitive types
  - Proof relevance

- Implementation in Coq ?

```
coq -pra MyPraProof.v
```

References

Haskell B Curry.
A formalization of recursive arithmetic.
American Journal of Mathematics, pages 263–282, 1941.

Hugo Herbelin and Arnaud Spiwack.
The rooster and the syntactic bracket.
arXiv preprint arXiv:1309.5767, 2013.

Daniel Leivant.
Global semantic typing for inductive and coinductive computing.
arXiv preprint arXiv:1312.6323, 2013.

QUESTIONS

Thank you for listening !