Monotonic Learning, Interactive Realizers and Monads

Ugo de'Liguoro

Dipartimento di Informatica, Università di Torino

joint work with **Stefano Berardi**

TYPES'09, Aussois, May 13th 2009

Constructive interpretations of **PA**

- Known constructive interpretations of classical arithmetic ($\neg\neg$-translation, no counterxample, continuations, $\lambda\mu$-calculus, etc.) either substantially alter the statements and their proofs, or keep the witness construction implicit.

Constructive interpretations of **PA**

- Known constructive interpretations of classical arithmetic ($\neg\neg$-translation, no counterxample, continuations, $\lambda\mu$-calculus, etc.) either substantially alter the statements and their proofs, or keep the witness construction implicit.
- Coquand's game theoretic semantics of classical arithmetic provides a first example of a direct interpretation of proofs as winning strategies that really "follow" the proof.

Constructive interpretations of **PA**

- Known constructive interpretations of classical arithmetic ($\neg\neg$-translation, no counterxample, continuations, $\lambda\mu$-calculus, etc.) either substantially alter the statements and their proofs, or keep the witness construction implicit.
- Coquand's game theoretic semantics of classical arithmetic provides a first example of a direct interpretation of proofs as winning strategies that really "follow" the proof.
- The idea of "learning" strategies is similar but more flexible and concrete: the Learner (Eloisa, or Player) proceeds by testing her guesses against particular examples getting answers from the Nature (Abelard, Opponent or, prosaically, the standard classical model of arithmetic): the proof is her learning method, that eventually succeeds.

Monotonic learning

- The concept of learning is far too general: it allows for retracting and then resuming retracted guesses, in a form called “unbounded backtracking”.

Monotonic learning

- The concept of learning is far too general: it allows for retracting and then resuming retracted guesses, in a form called "unbounded backtracking".
- If we limit ourself to learning strategies that can only abandon a branch definitely, then we are about 1-backtracking and monotonic learning.

Monotonic learning

- The concept of learning is far too general: it allows for retracting and then resuming retracted guesses, in a form called "unbounded backtracking".
- If we limit ourself to learning strategies that can only abandon a branch definitely, then we are about 1-backtracking and monotonic learning.
- 1-backtracking is enough to model learning strategies which come from proofs using instances of *excluded middle* only if they are Σ_1^0 formulas (with parameters).

PRA, Primitive Recursive Arithmetic

Let **PRA** be the quantifier free fragment of **HA** with equality, and all equational axioms defining primitive recursive functions.

Any **PRA** formula $A(\vec{x})$ defines a primitive recursive predicate $[\![A]\!] : \mathbb{N}^k \rightarrow 2$ s.t.

$$\textbf{PRA} \vdash A \Rightarrow \forall \vec{n} \in \mathbb{N}^k.\ [\![A]\!](\vec{n}) = 1.$$

Note that, although **PRA** $\subseteq$ **HA**, for any A

$$\textbf{PRA} \vdash A \vee \neg A$$

even if **PRA** itself is not a decidable theory.

$\mathbf{EM}_1$, Excluded Middle over Σ^0_1 formulas

With A in the language of **PRA** consider:

$$(\mathbf{EM}_1) \qquad \forall \vec{x}.\ \exists y\ A(\vec{x}, y) \vee \forall y\ \neg A(\vec{x}, y)$$

which is classically equivalent to to the skolemized version

$$\forall \vec{x}, y.\ A(\vec{x}, \varphi(\vec{x})) \vee \neg A(\vec{x}, y),$$

and to

$$\forall \vec{x}, y.\ A(\vec{x}, y) \rightarrow A(\vec{x}, \varphi(\vec{x})).$$

Then **PRA** + $\mathbf{EM}_1$ is obtained form **PRA** by adding:

$$(\chi) \quad P(\vec{x}, y) \rightarrow \chi_P(\vec{x})$$
$$(\varphi) \quad \chi_P(\vec{x}) \rightarrow P(\vec{x}, \varphi_P(\vec{x}))$$

for each (definition in **PRA** of) primitive recursive predicate P.

How computability of **PRA** + $\mathbf{EM}_1$ predicates is recoverable?

A process that uses $\mathbf{EM}_1$ while learning say the value of a function or testing a predicate might proceed as follows:

1. first assume that $\forall y. \neg A(y)$,
2. if a counterxample is met, choose definitely $\exists y. A(y)$.

Since the positive (and assessed knowledge) might only grow, we call this process "monotonic".

Observe that the learner will change her mind at most once w.r.t. each instance of $\mathbf{EM}_1$ used in the proof (a finite object), hence her guessing activity will be eventually constant: but in some cases she will be never aware that a stable point has be reached.

$\mathbb{S}$, the poset of states of knowledge

A *state of knowledge* is some finite set

$$s = \{\langle P_1, \vec{m}_1, n_1\rangle, \ldots, \langle P_l, \vec{m}_l, n_l\rangle\}$$

with $P_1, \ldots, P_l$ (definitions of) primitive recursive predicates and:

- if $\langle P, \vec{m}, n\rangle \in s$ then $P(\vec{m}, n)$ holds;
- if $\langle P, \vec{m}, n\rangle, \langle P, \vec{m}, n'\rangle \in s$ then $n = n'$.

$\mathbb{S}$ is decidable, and $(\mathbb{S}, \sqsubseteq, \sqcup, \bot)$ is a poset with $\sqsubseteq = \subseteq$, which is closed under compatible join $\sqcup$ (just union of bounded elements) and bottom $\bot = \emptyset$.

Objects indexed over $\mathbb{S}$

Let P be a $k+1$-ary primitive recursive predicate, and $s \in \mathbb{S}$:

$$[\![\chi_P]\!](\vec{m}, s) = \begin{cases} 1 & \text{if } \langle P, \vec{m}, n\rangle \in s \text{ for some } n \\ 0 & \text{otherwise} \end{cases}$$

$$[\![\varphi_P]\!](\vec{m}, s) = \begin{cases} n & \text{if } \langle P, \vec{m}, n\rangle \in s \text{ for some } n \\ 0 & \text{otherwise} \end{cases}$$

Then both $[\![\chi_P]\!]$ and $[\![\varphi_P]\!]$ are computable; moreover for any fixed $\vec{m}$ and weakly increasing infinite sequence:

$$s_0 \sqsubseteq s_1 \sqsubseteq s_2 \sqsubseteq \cdots$$

$\lambda s.\ [\![\chi_P]\!](\vec{m}, s)$ and $\lambda s.\ [\![\varphi_P]\!](\vec{m}, s)$ are eventually constant.

We say they are *convergent* or recursive in the limit, in Gold's sense.

The interpretation of **PRA** + $\mathbf{EM}_1$ formulas

Let $\xi : Var \to (\mathbb{S} \to \mathbb{N})$ then $[\![\_]\!]^{\mathbb{S}}_{\xi}$ is inductively defined:

$$
\begin{array}{rcl}
[\![x]\!]^{\mathbb{S}}_{\xi} & = & \xi(x) \\
[\![n]\!]^{\mathbb{S}}_{\xi} & = & \lambda s.\ n \\
[\![f(t_1,\ldots,t_n)]\!]^{\mathbb{S}}_{\xi} & = & \lambda s.\ f([\![t_1]\!]^{\mathbb{S}}_{\xi}(s),\ldots,[\![t_n]\!]^{\mathbb{S}}_{\xi}(s)) \\
[\![\varphi_P(t_1,\ldots,t_n)]\!]^{\mathbb{S}}_{\xi} & = & \lambda s.\ [\![\varphi_P]\!]([\![t_1]\!]^{\mathbb{S}}_{\xi}(s),\ldots,[\![t_n]\!]^{\mathbb{S}}_{\xi}(s),\ s) \\
[\![P(t_1,\ldots,t_n)]\!]^{\mathbb{S}}_{\xi} & = & \lambda s.\ P([\![t_1]\!]^{\mathbb{S}}_{\xi}(s),\ldots,[\![t_n]\!]^{\mathbb{S}}_{\xi}(s)) \\
[\![\chi_P(t_1,\ldots,t_n)]\!]^{\mathbb{S}}_{\xi} & = & \lambda s.\ [\![\chi_P]\!](([\![t_1]\!]^{\mathbb{S}}_{\xi}(s),\ldots,[\![t_n]\!]^{\mathbb{S}}_{\xi}(s),\ s)
\end{array}
$$

confusing the symbols f and P with the primitive recursive function and predicate they refer to, respectively. This extends to a classical definition of $[\![A]\!]^{\mathbb{S}}_{\xi}$ which is however computable (and primitive recursive).

The states monad $\mathcal{S}$

$$\begin{aligned} \mathcal{S}X &= \mathbb{S} \to X \\ \eta^{\mathcal{S}}_X(x) &= \lambda s.\, x \ \ (\text{written } \lambda\_.\, x) \\ f^{*^{\mathcal{S}}}(\alpha) &= \lambda s.\ f(\alpha(s), s) \end{aligned}$$

where $f : X \to \mathcal{S}Y$, $\alpha : \mathbb{S} \to X$ and $f^{*^{\mathcal{S}}} : \mathcal{S}X \to \mathcal{S}Y$.

$(\mathcal{S}, \eta^{\mathcal{S}}, \_^{*^{\mathcal{S}}})$ is a Kleisli triple, hence a monad, with tensorial strength:

$$t^{\mathcal{S}}_{X,Y} : X \times \mathcal{S}Y \to \mathcal{S}(X \times Y)$$

defined as

$$t^{\mathcal{S}}_{X,Y}(x, \alpha) = \lambda s.\ (x, \alpha(s))$$

by means of which the definition of $[\![\_]\!]^{\mathbb{S}}_{\xi}$ can be rephrased.

Remark: differences w.r.t. Kripke models

It is not true that if $\mathbf{PRA} + \mathbf{EM}_1 \vdash A$ then $[\![A]\!]^{\mathcal{S}}_{\xi}(s) = 1$ for all $s \in \mathbb{S}$.

Worse, even if

$$[\![\chi_P]\!](\vec{m}, s) = 1 \;\;\&\;\; s \sqsubseteq s' \Rightarrow [\![\chi_P]\!](\vec{m}, s') = 1$$

this is not true in general

$$[\![A]\!]^{\mathcal{S}}_{\xi}(s) = 1 \;\;\&\;\; s \sqsubseteq s' \not\Rightarrow [\![A]\!]^{\mathcal{S}}_{\xi}(s') = 1.$$

Take $A := \chi_P(x) \to x = \mathrm{succ}(x)$, where $P(x, y) \Leftrightarrow x < y$
$s = \{\langle P, 1, 2\rangle\}$ and $s' = \{\langle P, 1, 2\rangle, \langle P, 0, 1\rangle\}$.

Interactive Realizers

An *interactive realizer* should strictly depend on the state, by telling what is missing in the finite knowledge we have of the standard model to reach a certain goal, e.g. to asses the truth of a statement.

As such it should be a convergent mapping $r : \mathbb{S} \to \mathbb{S}$ always allowing for "enlarging" the given states, which is only possible if $r(s)$ is consistent with s for all state s.

Interactive Realizers

Consider pairs $(\alpha, r) \in (\mathbb{S} \to X) \times (\mathbb{S} \to \mathbb{S})$ to be interpreted:

- α is a convergent map, representing the family of individuals in X of the form $\lim(\alpha \circ \sigma)$, for any w.i. sequence $\sigma : \mathbb{N} \to \mathbb{S}$;
- r is a convergent map that is able to "force" α to converge into some subset $Y \subseteq X$.

Interactive Realizers

Let $r : \mathbb{S} \to \mathbb{S}$ be a convergent map:

- r is a *realizer* if it is *compatible*: $r(s) \sqcup s$ exists for all $s \in \mathbb{S}$;
- the set of *prefixed points* of r is $Pref(r) = \{s \in \mathbb{S} \mid r(s) \sqsubseteq s\}$.

Lemma. If r is a realizer then $Pref(r)$ is a cofinal subset of $\mathbb{S}$ (hence $\neq \emptyset$).

Proof. Given any $s \in \mathbb{S}$ consider the sequence:

$$\sigma(0) = s, \quad \sigma(i+1) = r(\sigma(i)) \sqcup \sigma(i).$$

Then by convergence of r, $r \circ \sigma$ is eventually constant in a state in $Pref(r)$ which is over s.

Interactive Forcing

A realizer r forces $\alpha : \mathbb{S} \rightarrow X$ into some subset $Y \subseteq X$ if

$$\alpha(Pref(r)) \subseteq Y$$

We need a slightly more complex concept:

$$r \Vdash \alpha : \{Y_s\}_s \Leftrightarrow \forall s \in Pref(r).\ \alpha(s) \in Y_s$$

where $\bigcup_s Y_s \subseteq X$. Then define:

$$ext(A)_s = \{\vec{m} \mid [\![A]\!]^{\mathbb{S}}_{[\overrightarrow{\lambda\_.m}/\vec{x}]}(s) = 1\}$$

where A is a $\mathbf{PRA} + \mathbf{EM}_1$ formula, and write $ext(A) = \{ext(A)_s\}_s$.

Interactive realizer of the (χ)-axiom

If $P(\vec{\alpha}(s), n)$ is true but $\langle P, \vec{\alpha}(s), n' \rangle \notin s$ for any n', then

$$[\![P(\vec{x}, y) \to \chi_P(\vec{x})]\!]^{\mathcal{S}}_{[\vec{\alpha}, \lambda\_.n/\vec{x}, y]}(s) \neq 1$$

The following function always forces the formula above to be true:

$$r_P(\vec{m}, n, s) = \begin{cases} \{\langle P, \vec{m}, n \rangle\} & \text{if } P(\vec{m}, n) \text{ and } \forall n'.\langle P, \vec{m}, n' \rangle \notin s \\ \bot & \text{else} \end{cases}$$

Indeed $\lambda s.\ r_P(\vec{m}, n, s)$ is a realizer and

$$r_P^{*\mathcal{S}}(\vec{\alpha}, \beta) = \lambda s. r_P(\vec{\alpha}(s), \beta(s), s) \Vdash \vec{\alpha}, \beta : P(\vec{x}, y) \to \chi_P(\vec{x})$$

Realizability Theorem

Interactive Realizability Theorem. If $\mathbf{PRA} + \mathbf{EM}_1 \vdash A$ then for any convergent $\vec{\alpha}$ there exists a realizer r, depending uniformly on $\vec{\alpha}$, such that $r \Vdash \vec{\alpha} : ext(A)$ (where $\vec{\alpha}$ reads as $\langle \vec{\alpha} \rangle$).

The realizer is built along the proof itself, and reflects its structure.

Corollary. If $\mathbf{PRA} + \mathbf{EM}_1 \vdash A$, then

$$\forall \text{ convergent } \alpha_1 \ldots \alpha_k \in \mathcal{S}\mathbb{N}\ \forall s \in \mathbb{S}\ \exists s' \sqsupseteq s.\ [\![A]\!]^{\mathcal{S}}_{[\vec{\alpha}/\vec{x}]}(s') = 1.$$

The realizability monad

$$\begin{array}{rcl} \mathcal{R}X & = & (\mathbb{S} \to X) \times (\mathbb{S} \to \mathbb{S}) \\ \eta_X^{\mathcal{R}}(x) & = & (\lambda\_.x, \lambda\_.\bot) = (\eta_X^{\mathcal{S}}(x), \eta_{\mathbb{S}}^{\mathcal{S}}(\bot)) \\ f^{*^{\mathcal{R}}}(\alpha, r) & = & (f_1^{*^{\mathcal{S}}}(\alpha), m(r, f_2^{*^{\mathcal{S}}}(\alpha)) \end{array}$$

where $f : X \to (\mathbb{S} \to Y) \times (\mathbb{S} \to \mathbb{S})$ is identified with the pair $\langle f_1, f_2 \rangle$ where $f_i = \pi_i \circ f$.

We introduce a "merging" of realizers $m : (\mathbb{S} \to \mathbb{S})^2 \to (\mathbb{S} \to \mathbb{S})$ defined by:

$$m(r, r')(s) = \begin{cases} r'(s) & \text{if } r(s) \leq s \\ r(s) & \text{else.} \end{cases}$$

It sends realizers into realizers and is such that:

$$Pref(m(r, r'))) = Pref(r) \cap Pref(r').$$

The realizability monad vs the side-effect monad

This is similar to the Side-Effect monad (remark by Coquand):

$$\mathcal{E}X = \mathbb{S} \to (X \times \mathbb{S})$$

as

$$(\mathbb{S} \to X) \times (\mathbb{S} \to \mathbb{S}) \simeq \mathbb{S} \to (X \times \mathbb{S})$$

and the isomorphism sends convergent mapping into convergent ones; but

$$f^{*^{\mathcal{E}}}(\gamma) = \lambda s.\ f((\pi_1 \circ \gamma)(s), (\pi_2 \circ \gamma)(s))$$

has a quite restrictive (sequential) meaning, while our monad is more general, and also describe parallel computations.

Conclusions and perspectives

- Interactive realizers are computable functionals, representable e.g. in Gödel system **T**.

Conclusions and perspectives

- Interactive realizers are computable functionals, representable e.g. in Gödel system **T**.
- By a cleaver combination with Kleene realizability the interactive realizers can be extended to $\mathbf{HA} + \mathbf{EM}_1$ (Aschieri-Berardi, TLCA'09 to appear).

Conclusions and perspectives

- Interactive realizers are computable functionals, representable e.g. in Gödel system **T**.
- By a cleaver combination with Kleene realizability the interactive realizers can be extended to $\mathbf{HA} + \mathbf{EM}_1$ (Aschieri-Berardi, TLCA'09 to appear).
- The use of monads and of category theory should be a good structuring principle and a comparison tool.

Conclusions and perspectives

- Interactive realizers are computable functionals, representable e.g. in Gödel system **T**.
- By a cleaver combination with Kleene realizability the interactive realizers can be extended to **HA** + $\mathbf{EM}_1$ (Aschieri-Berardi, TLCA'09 to appear).
- The use of monads and of category theory should be a good structuring principle and a comparison tool.
- We should also revisit known constructive interpretations of classical proofs to see whether the learning approach end the interactive realizers can throw a new light.