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
1 Motivation

2 The theory \textbf{PRA} + \textbf{EM}_1

3 The state monad

4 Interactive realizers

5 Conclusions
Constructive interpretations of PA

- Known constructive interpretations of classical arithmetic (\(\neg\neg\)-translation, no counterexample, 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 counterexample, 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.
Known constructive interpretations of classical arithmetic (¬¬-translation, no counterxample, continuations, λμ-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 $\Sigma^0_1$ formulas (with parameters).
PRA, Primitive Recursive Arithmetic

Let \( \text{PRA} \) be the quantifier free fragment of \( \text{HA} \) with equality, and all equational axioms defining primitive recursive functions.

Any \( \text{PRA} \) formula \( A(\vec{x}) \) defines a primitive recursive predicate \( \llbracket A \rrbracket : \mathbb{N}^k \rightarrow 2 \) s.t.

\[
\text{PRA} \vdash A \Rightarrow \forall \vec{n} \in \mathbb{N}^k. \llbracket A \rrbracket (\vec{n}) = 1.
\]

Note that, although \( \text{PRA} \subseteq \text{HA} \), for any \( A \)

\[
\text{PRA} \vdash A \lor \neg A
\]

even if \( \text{PRA} \) itself is not a decidable theory.
EM₁, Excluded Middle over $\Sigma^0_1$ formulas

With $A$ in the language of PRA consider:

$$(\text{EM}_1) \quad \forall \vec{x}. \exists y \ A(\vec{x}, y) \lor \forall y \neg A(\vec{x}, y)$$

which is classically equivalent to the skolemized version

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

and to

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

Then PRA + EM₁ is obtained from 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 $P$. 
A process that uses $EM_1$ 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 counterexample 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 $EM_1$ 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 been reached.
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 \( n = n' \).

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

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

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

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

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

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

$\lambda s. \llbracket \chi_P \rrbracket (\vec{m}, s)$ and $\lambda s. \llbracket \varphi_P \rrbracket (\vec{m}, s)$ are eventually constant.

We say they are convergent or recursive in the limit, in Gold’s sense.
The interpretation of \(\text{PRA} + \text{EM}_1\) formulas

Let \(\xi : \text{Var} \to (\mathbb{S} \to \mathbb{N})\) then \(\llbracket \_ \rrbracket^S_\xi\) is inductively defined:

\[
\begin{align*}
\llbracket x \rrbracket^S_\xi &= \xi(x) \\
\llbracket n \rrbracket^S_\xi &= \lambda s. n \\
\llbracket f(t_1, \ldots, t_n) \rrbracket^S_\xi &= \lambda s. f(\llbracket t_1 \rrbracket^S_\xi(s), \ldots, \llbracket t_n \rrbracket^S_\xi(s)) \\
\llbracket \varphi_P(t_1, \ldots, t_n) \rrbracket^S_\xi &= \lambda s. \llbracket \varphi_P \rrbracket(\llbracket t_1 \rrbracket^S_\xi(s), \ldots, \llbracket t_n \rrbracket^S_\xi(s), s) \\
\llbracket P(t_1, \ldots, t_n) \rrbracket^S_\xi &= \lambda s. P(\llbracket t_1 \rrbracket^S_\xi(s), \ldots, \llbracket t_n \rrbracket^S_\xi(s)) \\
\llbracket \chi_P(t_1, \ldots, t_n) \rrbracket^S_\xi &= \lambda s. \llbracket \chi_P \rrbracket((\llbracket t_1 \rrbracket^S_\xi(s), \ldots, \llbracket t_n \rrbracket^S_\xi(s), s)
\end{align*}
\]

confusing the symbols \(f\) and \(P\) with the primitive recursive function and predicate they refer to, respectively. This extends to a classical definition of \(\llbracket A \rrbracket^S_\xi\) which is however computable (and primitive recursive).
The states monad $S$

\[
SX = \mathbb{S} \rightarrow X
\]

\[
\eta^S_X(x) = \lambda s. x \quad \text{(written } \lambda s. x)\]

\[
f^*_S(\alpha) = \lambda s. f(\alpha(s), s)
\]

where $f : X \rightarrow SY$, $\alpha : \mathbb{S} \rightarrow X$ and $f^*_S : SX \rightarrow SY$.

$(S, \eta^S, f^*_S)$ is a Kleisli triple, hence a monad, with tensorial strength:

\[
t^S_{X, Y} : X \times SY \rightarrow S(X \times Y)
\]

defined as

\[
t^S_{X, Y}(x, \alpha) = \lambda s. (x, \alpha(s))
\]

by means of which the definition of $\llbracket \_ \rrbracket^S_\xi$ can be rephrased.
Remark: differences w.r.t. Kripke models

It is not true that if \( \text{PRA} + \text{EM}_1 \vdash A \) then \( \llbracket A \rrbracket^S_\xi(s) = 1 \) for all \( s \in S \).

Worse, even if

\[
\llbracket \chi_P \rrbracket(\vec{m}, s) = 1 \ \& \ s \sqsubseteq s' \Rightarrow \llbracket \chi_P \rrbracket(\vec{m}, s') = 1
\]

this is not true in general

\[
\llbracket A \rrbracket^S_\xi(s) = 1 \ \& \ s \sqsubseteq s' \nRightarrow \llbracket A \rrbracket^S_\xi(s') = 1.
\]

Take \( A := \chi_P(x) \rightarrow x = \text{succ}(x) \), 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 assess the truth of a statement.

As such it should be a convergent mapping $r : \mathcal{S} \to \mathcal{S}$ 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:

- \(\alpha\) 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” \(\alpha\) to converge into some subset \(Y \subseteq X\).
Interactive Realizers

Let \( r : S \to S \) be a convergent map:

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

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

**Proof.** Given any \( s \in S \) 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 \( \text{Pref}(r) \) which is over \( s \).
A realizer $r$ forces $\alpha : S \rightarrow X$ into some subset $Y \subseteq X$ if

$$\alpha(\text{Pref}(r)) \subseteq Y$$

We need a slightly more complex concept:

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

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

$$\text{ext}(A)_s = \{ \vec{m} \mid [A]^{S}_{[\lambda \_m/x]}(s) = 1\}$$

where $A$ is a $\text{PRA} + \text{EM}_1$ formula, and write $\text{ext}(A) = \{\text{ext}(A)_s\}_s$. 
Interactive realizer of the \((\chi)\)-axiom

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

\[
\llbracket P(\bar{x}, y) \to \chi P(\bar{x}) \rrbracket^S_{[\bar{\alpha}, \lambda \cdot n/\bar{x}, y]}(s) \neq 1
\]

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

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

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

\[
r_P^* (\bar{\alpha}, \beta) = \lambda s. r_P(\bar{\alpha}(s), \beta(s), s) \vdash \bar{\alpha}, \beta : P(\bar{x}, y) \to \chi P(\bar{x})
\]
Realizability Theorem

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

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

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

$$\forall \text{ convergent } \alpha_1 \ldots \alpha_k \in S\mathbb{N} \forall s \in S \exists s' \sqsubset s. \left[ A \right]^{S}_{[\vec{\alpha}/\vec{x}]}(s') = 1.$$
The realizability monad

\[ R_X = (S \to X) \times (S \to S) \]

\[ \eta^R_X(x) = (\lambda_.x, \lambda_.\bot) = (\eta^S_X(x), \eta^S_S(\bot)) \]

\[ f^*^R(\alpha, r) = (f^*_1(\alpha), m(r, f^*_2(\alpha)) \]

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

We introduce a “merging” of realizers \( m : (S \to S)^2 \to (S \to S) \) 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:

\[ \text{Pref}(m(r, r')) = \text{Pref}(r) \cap \text{Pref}(r') \]
The realizability monad vs the side-effect monad

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

\[ E \mathcal{X} = \mathbb{S} \rightarrow (\mathcal{X} \times \mathbb{S}) \]

as

\[ (\mathbb{S} \rightarrow \mathcal{X}) \times (\mathbb{S} \rightarrow \mathbb{S}) \cong \mathbb{S} \rightarrow (\mathcal{X} \times \mathbb{S}) \]

and the isomorphism sends convergent mapping into convergent ones; but

\[ f^* (\gamma) = \lambda s. f((\pi_1 \circ \gamma)(s), (\pi_2 \circ \gamma)(s)) \]

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 $\mathbf{T}$.

- By a clever 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 $\mathbf{T}$.

- By a clever combination with Kleene realizability the interactive realizers can be extended to $\mathbf{HA + EM}_1$ (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 $\mathbb{T}$.
- By a cleaver combination with Kleene realizability the interactive realizers can be extended to $\text{HA} + \text{EM}_1$ (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.