Knowledge Spaces and Interactive Realizers

Stefano Berardi, Ugo de'Liguoro

CSL'12, Fontainebleau, September 2012

Introduction

Introduction

- *Classical logic and mathematics* are about the truth and ideal, not necessarily computable, entities.

Introduction

- *Classical logic and mathematics* are about the truth and ideal, not necessarily computable, entities.
- *Intuitionistic logic and constructive mathematics* put the mathematician activity on epistemic grounds, emphasizing mental activity and computability

Introduction

- *Classical logic and mathematics* are about the truth and ideal, not necessarily computable, entities.
- *Intuitionistic logic and constructive mathematics* put the mathematician activity on epistemic grounds, emphasizing mental activity and computability
- *Proof mining* is the apparently paradoxical enterprise to extract a constructive content out of non constructive proofs

Introduction

- *Classical logic and mathematics* are about the truth and ideal, not necessarily computable, entities.
- *Intuitionistic logic and constructive mathematics* put the mathematician activity on epistemic grounds, emphasizing mental activity and computability
- *Proof mining* is the apparently paradoxical enterprise to extract a constructive content out of non constructive proofs
- *Interactive realizability* aims at a semantical foundation of proof mining, by interpreting classical proofs as **learning strategies**

Example (1)

For $f : \mathbb{N} \to \mathbb{N}$ recursive, the following statement holds:

$$\forall\, k\ \exists\, x_1 < \cdots < x_k.\ f(x_1) \leq \cdots \leq f(x_k)$$

Proof. By induction over k: if we have $x_1 < \cdots < x_{k-1}$ s.t. $f(x_1) \leq \cdots \leq f(x_{k-1})$ take:

$$x_k = \min\{m \mid x_{k-1} < m \ \wedge\ \neg\exists p > m.\ f(m) > f(p)\}$$

Example (1)

For $f : \mathbb{N} \to \mathbb{N}$ recursive, the following statement holds:

$$\forall k \, \exists x_1 < \cdots < x_k . \, f(x_1) \leq \cdots \leq f(x_k)$$

Proof. By induction over k: if we have $x_1 < \cdots < x_{k-1}$ s.t. $f(x_1) \leq \cdots \leq f(x_{k-1})$ take:

$$x_k = \min\{m \mid x_{k-1} < m \ \wedge \ \neg\exists p > m. \, f(m) > f(p)\}$$

The proof can be red as the pseudo-algorithm:

$S := \{0\}$ or any other singleton $\subseteq \mathbb{N}$
while $|S| < k$ do
    $n := \min\{m \mid \max(S) < m \ \wedge \ \neg\exists p > m. \, f(m) > f(p)\}$
    $S := S \cup \{n\}$
return S

Example (2)

By relativizing to finite approximations of f, the pseudo-algorithm becomes an effective procedure:

$X := \emptyset$ or any (finite) $\subseteq \{`f(p) > f(q)' \mid p < q \wedge f(p) > f(q)\}$
$S := \{0\}$ or any other singleton $\subseteq \mathbb{N}$
while $|S| < k$ do
$\quad n := \min\{m \mid \max(S) < m \ \wedge \ \neg\exists p > m.\ `f(m) > f(p)' \in X\}$
$\quad X := X \cup \{`f(m) > f(n)' \mid S \ni m < n \wedge f(m) > f(n)\}$
$\quad S := \{m \in S \mid `f(m) > f(n)' \notin X\} \cup \{n\}$
return S

Example (3)

Let us define:

$$\begin{aligned}\alpha(X) &= \text{the tuple } x_1 < \cdots < x_h \text{ of maximal length s.t.}\\ &\quad \forall i < j.`f(x_i) > f(x_j)' \notin X \wedge\\ &\quad (f(x_1), \ldots, f(x_h)) \text{ is lexicographically minimal}\\ r(X) &= \{`f(m) > f(n)' \mid |\alpha(X)| < k \wedge\\ &\qquad \alpha(X) \ni m < n \wedge f(m) > f(n)\}\end{aligned}$$

Example (3)

Let us define:

$$\begin{aligned}
\alpha(X) &= \text{the tuple } x_1 < \cdots < x_h \text{ of maximal length s.t.} \\
&\quad \forall i < j. `f(x_i) > f(x_j)' \notin X \wedge \\
&\quad (f(x_1), \ldots, f(x_h)) \text{ is lexicographically minimal} \\
r(X) &= \{`f(m) > f(n)' \mid |\alpha(X)| < k \wedge \\
&\qquad \alpha(X) \ni m < n \wedge f(m) > f(n)\}
\end{aligned}$$

then $S = \alpha(X)$ and we get the equivalent algorithm:

$X := \emptyset$
while $r(X) \neq \emptyset$ do
    $S := \alpha(X)$
    $n := \min\{m \mid \max(S) < m \ \wedge \ \neg\exists p > m.\ `f(m) > f(p)' \in X\}$
    $X := X \cup \{`f(m) > f(n)' \mid m \in S \wedge f(m) > f(n)\}$
return $\alpha(X)$

Learning

- learning is knowledge changing with time, proceeding by asking *questions* and obtaining *answers*
- the steps of this process are *states of knowledge*, namely sets of answers (e.g. logical formulas), supposed to be true

Learning

- learning is knowledge changing with time, proceeding by asking *questions* and obtaining *answers*
- the steps of this process are *states of knowledge*, namely sets of answers (e.g. logical formulas), supposed to be true
- to avoid ambiguities we require that at most one answer is known for each question at each state
- the truth value of an answer may depend on the truth value of answers of lower complexity in the same state
- adding a new and even true answer to a state can make false other answers in the state itself, so that the process of "expanding" a knowledge state is a non-monotonic one

States of Knowledge

- $\mathbb{A}$ is the (countable) set of *answers*
- $x \sim y$ if $x, y \in \mathbb{A}$ are answers to the same question
- $\mathbb{Q} = \mathbb{A}/_{\sim}$ is the set of *questions*
- $X \subseteq \mathbb{A}$ is a *state of knowledge* if it includes at most one answer for each question, i.e.

$$x, y \in X \;\Rightarrow\; x = y \vee x \not\sim y$$

- $\mathbb{S}$ is the set of states of knowledge

States of Knowledge

- $\mathbb{A}$ is the (countable) set of *answers*
- $x \sim y$ if $x, y \in \mathbb{A}$ are answers to the same question
- $\mathbb{Q} = \mathbb{A}/_\sim$ is the set of *questions*
- $X \subseteq \mathbb{A}$ is a *state of knowledge* if it includes at most one answer for each question, i.e.

$$x, y \in X \;\Rightarrow\; x = y \vee x \not\sim y$$

- $\mathbb{S}$ is the set of states of knowledge

Remark: the relation

$$x \# y \Leftrightarrow x = y \;\&\; x \not\sim y$$

makes $(\mathbb{A}, \#)$ into the web of the coherence space $\mathbb{S}$, hence $(\mathbb{S}, \subseteq)$ is a domain.

The topology $\Omega(\mathbb{S})$

We define the following map, called the *query map*:

$$\mathrm{q} : \mathbb{S} \times \mathbb{Q} \to \mathcal{P}_{fin}(\mathbb{A}), \qquad \mathrm{q}(X, [x]) \triangleq X \cap [x]$$

meaning:

- $\mathrm{q}(X, [x]) = \{x\}$: the answer to question $[x]$ at X is x
- $\mathrm{q}(X, [x]) = \emptyset$: at X no answer is known to the question $[x]$

The topology $\Omega(\mathbb{S})$

We define the following map, called the *query map*:

$$q : \mathbb{S} \times \mathbb{Q} \to \mathcal{P}_{fin}(\mathbb{A}), \qquad q(X, [x]) \triangleq X \cap [x]$$

meaning:

- $q(X, [x]) = \{x\}$: the answer to question $[x]$ at X is x
- $q(X, [x]) = \emptyset$: at X no answer is known to the question $[x]$

The *state topology* $\Omega(\mathbb{S})$ over $\mathbb{S}$ is the least topology making continuous the query map.

$\Omega(\mathbb{S})$ is generated by the subbasics A_x, B_x, with $x \in \mathbb{A}$:

$$\begin{aligned} A_x &= \{X \in \mathbb{S} \mid x \in X\} = \{X \in \mathbb{S} \mid X \cap [x] = \{x\}\} \\ B_x &= \{X \in \mathbb{S} \mid X \cap [x] = \emptyset\} \end{aligned}$$

Relative Truth

Relative Truth

Define the maps:

- lev : $\mathbb{A} \to \mathit{Ord}$, level function stratifying $\mathbb{A}$ and such that

$$x \sim y \Rightarrow \text{lev}(x) = \text{lev}(y)$$

Relative Truth

Define the maps:

- lev : $\mathbb{A} \to Ord$, level function stratifying $\mathbb{A}$ and such that

$$x \sim y \Rightarrow \mathrm{lev}(x) = \mathrm{lev}(y)$$

- tr : $\mathbb{A} \times \mathbb{S} \to \{\mathrm{true}, \mathrm{false}\}$, relative evaluation assigning truth values to answers w.r.t. knowledge states, which is continuous and s.t.

$$\mathrm{tr}(x, X) = \mathrm{tr}(x, X \upharpoonright \mathrm{lev}(x))$$

where $X \upharpoonright \gamma = \{x \in X \mid \mathrm{lev}(x) < \gamma\}$ for $\gamma \in Ord$.

Relative Truth

Define the maps:

- lev : $\mathbb{A} \to \mathit{Ord}$, level function stratifying $\mathbb{A}$ and such that

$$x \sim y \Rightarrow \text{lev}(x) = \text{lev}(y)$$

- tr : $\mathbb{A} \times \mathbb{S} \to \{\text{true}, \text{false}\}$, relative evaluation assigning truth values to answers w.r.t. knowledge states, which is continuous and s.t.

$$\text{tr}(x, X) = \text{tr}(x, X \upharpoonright \text{lev}(x))$$

where $X \upharpoonright \gamma = \{x \in X \mid \text{lev}(x) < \gamma\}$ for $\gamma \in \mathit{Ord}$.

We say that $(\mathbb{A}, \sim, \text{lev}, \text{tr})$ is a *layered knowledge structure*.

Models

Let $X \in \mathbb{S}$, $x \in \mathbb{A}$:

- X is *sound* if $x \in X \Rightarrow \mathrm{tr}(x, X) = \mathrm{true}$;
- X is *complete* if $X \cap [x] = \emptyset \Rightarrow \mathrm{tr}(x, X) = \mathrm{false}$;
- X is a *model* if it is sound and complete.

Models

Let $X \in \mathbb{S}$, $x \in \mathbb{A}$:

- X is *sound* if $x \in X \Rightarrow \mathrm{tr}(x, X) = \mathrm{true}$;
- X is *complete* if $X \cap [x] = \emptyset \Rightarrow \mathrm{tr}(x, X) = \mathrm{false}$;
- X is a *model* if it is sound and complete.

that reads as:

- X is sound if it is a consistent (but possibly partial, not necessarily extensible to a complete) view of the world
- X is complete if no answer can be consistently added to X (but X itself is not necessarily made of true answers)
- X is a model if it is a perfect (but not unique in general!) representation of the world

Models existence

$\emptyset$ is trivially sound; an X including exactly one $x \in [y]$ for each question $[y] \in \mathbb{Q}$ is complete, but not necessarily sound (it depends on the choice of tr).

Theorem

For every layered knowledge structure $(\mathbb{A}, \sim, \text{lev}, \text{tr})$ and space of knowledge $\mathbb{S}$ over it, there exists a model $X \in \mathbb{S}$.

Solutions

Let $(\mathbb{A}, \sim, \mathrm{lev}, \mathrm{tr})$ be a layered knowledge structure and $\mathbb{S}$ its knowledge space.

$\alpha : \mathbb{S} \to \mathbb{N}$ is a *solution* to the *problem* $P \subseteq \mathbb{N}$ if it is continuous and

$$X \text{ is a model } \Rightarrow \alpha(X) \in P$$

$\alpha(X) = n$ is the guess at X for solving a problem P; it is a solution if the knowledge represented by any model X suffices for $n \in P$.

Solutions

Let $(\mathbb{A}, \sim, \mathrm{lev}, \mathrm{tr})$ be a layered knowledge structure and $\mathbb{S}$ its knowledge space.

$\alpha : \mathbb{S} \to \mathbb{N}$ is a *solution* to the *problem* $P \subseteq \mathbb{N}$ if it is continuous and

$$X \text{ is a model } \Rightarrow \alpha(X) \in P$$

$\alpha(X) = n$ is the guess at X for solving a problem P; it is a solution if the knowledge represented by any model X suffices for $n \in P$.

Problem

Models are ideal objects, infinite and non computable in general. Are there finitary and effective counterparts of the concepts of model and solution?

Interactive Realizers

An *interactive realizer* is a continuous map $r : \mathbb{S} \to \mathcal{P}_{fin}(\mathbb{A})$ s.t.

$$x \in r(X) \Rightarrow X \cap [x] = \emptyset \ \& \ \mathrm{tr}(x, X) = \mathrm{true}$$

$r(X)$ yields a finite set of answers to questions which are unanswered at X, and which are true w.r.t. X (if any such answer exists).

Interactive Realizers

An *interactive realizer* is a continuous map $r : \mathbb{S} \to \mathcal{P}_{fin}(\mathbb{A})$ s.t.

$$x \in r(X) \Rightarrow X \cap [x] = \emptyset \ \& \ \mathrm{tr}(x, X) = \mathrm{true}$$

$r(X)$ yields a finite set of answers to questions which are unanswered at X, and which are true w.r.t. X (if any such answer exists).

Given a continuous $\alpha : \mathbb{S} \to \mathbb{N}$ and any $P \subseteq \mathbb{N}$, we say that r *interactively realizes* α w.r.t. P, written $r \vdash \alpha : P$, if r is a realizer and for all $X \in \mathbb{S}$:

$$X \text{ sound } \ \& \ r(X) = \emptyset \Rightarrow \alpha(X) \in P$$

Finite approximations to models suffice

Theorem

If r is a realizer then there exists a **finite** sound $X \in \mathbb{S}$ such that $r(X) = \emptyset$

Finite approximations to models suffice

Theorem

If r is a realizer then there exists a **finite** sound $X \in \mathbb{S}$ such that $r(X) = \emptyset$

Proof. Since the following "algorithm" always terminates:

$X :=$ any finite sound state of knowledge, e.g. $\emptyset$

while $r(X) \neq \emptyset$ do

let $\emptyset \neq U \subseteq r(X)$ be such that $x \not\sim y$ for any $x, y \in U$

(at worst $U = \{z\}$ for some $z \in r(X)$)

$Y := \{x \in X \mid \mathrm{tr}(x, X \cup U) = \mathrm{true}\}$

$X := Y \cup U$

return X

Note that the values of X do not increase monotonically in general.

Finite approximations to models suffice

Theorem

If r is a realizer then there exists a **finite** sound $X \in \mathbb{S}$ such that $r(X) = \emptyset$

Proof. Since the following "algorithm" always terminates:

$X :=$ any finite sound state of knowledge, e.g. $\emptyset$

while $r(X) \neq \emptyset$ do

  let $\emptyset \neq U \subseteq r(X)$ be such that $x \not\sim y$ for any $x, y \in U$

   (at worst $U = \{z\}$ for some $z \in r(X)$)

  $Y := \{x \in X \mid \text{tr}(x, X \cup U) = \text{true}\}$ **backtracking!**

  $X := Y \cup U$

return X

Note that the values of X do not increase monotonically in general.

Completeness of Realizability

Theorem

There exists a solution α of a problem P relative to a knowledge space $\mathbb{S}$ if and only if there exists a realizer r such that $r \vdash \alpha : P$

Completeness of Realizability

Theorem

There exists a solution α of a problem P relative to a knowledge space $\mathbb{S}$ if and only if there exists a realizer r such that $r \vdash \alpha : P$

- models are perfect knowledge and represent *iterated limits* (according to the state topology) of finite and sound states of knowledges

Completeness of Realizability

Theorem

There exists a solution α of a problem P relative to a knowledge space $\mathbb{S}$ if and only if there exists a realizer r such that $r \vdash \alpha : P$

- models are perfect knowledge and represent *iterated limits* (according to the state topology) of finite and sound states of knowledges
- a *realizer* is an effective tool to learn a *finite approximation* to the ideal model, which is enough to find a witness of any (given) decidable predicate

Completeness of Realizability

Theorem

There exists a solution α of a problem P relative to a knowledge space $\mathbb{S}$ if and only if there exists a realizer r such that $r \vdash \alpha : P$

- models are perfect knowledge and represent *iterated limits* (according to the state topology) of finite and sound states of knowledges
- a *realizer* is an effective tool to learn a *finite approximation* to the ideal model, which is enough to find a witness of any (given) decidable predicate
- the completeness theorem implies that whenever a solution classically exists, it can be effectively learned

Conclusions

- interactive realizability provides a constructive interpretation of Tarskian's truth
- we have shown elsewhere (although in the monotonic case only, but see also Aschieri's work) that interactive realizers and solutions also interpret classical proofs
- we propose interactive realizability as a method for understanding the behaviour of programs extracted from non-constructive proofs, and in general to investigate proof mining

References

- S. Berardi, U. de'Liguoro, *Toward the interpretation of non constructive reasoning as non-monotonic learning*, Info. Comp. 207(1), 2009.
- F. Aschieri, S. Berardi, *Interactive learning-based realizability for Heyting arithmetic with* EM_1, LMCS 6(3), 2010.
- S. Berardi, U. de'Liguoro, *Interactive realizers. A new approach to program extraction form non-constructive proofs*, ACM Trans. Comp. Log. 13(2), 2012.