Fundamenta Informaticae XX (2015) 1–21
DOI 10.3233/FI-2012-0000
IOS Press

Non-monotonic pre-fix points and Learning

Stefano Berardi *

Computer Science Department
Università di Torino
Corso Svizzera 185, 10149 Torino, Italy
stefano@di.unito.it

Ugo de' Liguoro †

Computer Science Department
Università di Torino
Corso Svizzera 185, 10149 Torino, Italy
ugo.deliguoro@unito.it

Abstract. We consider the problem of finding pre-fix points of interactive realizers over arbitrary knowledge spaces, obtaining a relative recursive procedure. Knowledge spaces and interactive realizers are an abstract setting to represent learning processes, that can interpret non-constructive proofs. Atomic pieces of information of a knowledge space are stratified into levels, and evaluated into truth values depending on knowledge states. Realizers are then used to define operators that extend a given state by adding answers and possibly forcing us to remove some: in the learning process states of knowledge change non-monotonically. Existence of a pre-fix point of a realizer is equivalent to the termination of the learning process with some state of knowledge which is free of patent contradictions and such that there is nothing to add. In this paper we generalize our previous results in the case of level 2 knowledge spaces and deterministic operators to the case of ω-level knowledge spaces and of non-deterministic operators.

Address for correspondence: Corso Svizzera 185, 10149 Torino, Italy
*This author was partially supported by PRIN "Metodi logici per il trattamento dell'informazione", Anno 2010-2011 - prot. 2010FP79LR_007
†This author was partially supported by ICT COST Action IC1201 BETTY, MIUR PRIN Project CINA Prot. 2010LHT4KM and Torino University/Compagnia San Paolo Project SALT.

1. Introduction

A fundamental aspect of constructive interpretations of classical arithmetic is how information is gathered and handled while looking for a witness of the proved formulas. This has been understood by several authors as a problem of control and side effects, although intended in different ways. Building over Coquand's semantics of evidence of classical arithmetic [9] and its representation as limiting interaction sequences [4], we have developed the concept of *interactive realizability* in [3, 5], we provided a mechanical interpretation of non constructive proofs as effective strategies that "learn" the witness. In this paper we leave aside the issue of a mechanical interpretation of classical proofs, and we develop a model of learning which may be used as a guideline to design and check programs which provide an output solving a given problem by "learning".

According to [6], learning the truth of a statement expressible in first order arithmetic can be abstractly presented as a process going through steps, which we call *states of knowledge*, or *state* for short, such that a (candidate) output can be relatively computed out of them. A knowledge state is a set of evidences called *answers*, giving a partial information about the truth of a formula; since total knowledge consists of infinite states in general, we understand classical truth as the infinite limit of an approximating process, whose finite steps are "effective" in the sense that they are defined by relative recursive functions.

Knowledge is improved by answering questions, which is the task of *realizers*, namely certain functions, taking some state of knowledge and returning some finite sets of answers that can be consistently added to the given state. The correct output of a learning program is obtained whenever the realizer does not provide any new answer with respect to a given state. The basic example is the learning of excluded middle with just one quantifier. A realizer, in this case a strategy for learning the truth of $\exists x P(x) \vee \forall x \neg P(x)$, with P recursive, would start by producing a sequence of true statements $\neg P(n_0), \ldots, \neg P(n_h)$, supporting the truth of $\forall x \neg P(x)$ until no n_{h+1} appears such that $P(n_{h+1})$; as soon as this happens, the process stops because the learner knows that $\exists x P(x)$ is true, and the current "candidate output" is taken as the final output. The crucial fact is that, even if no such an n_{h+1} exists, a "continuous" realizer depending only on a finite amount of information about the state must reach a pre-fix point w.r.t. subset inclusion, which we interpret as the situation in which no more knowledge is needed to produce the correct output. Remark that when we stop adding knowledge, the belief about whether the statement $\exists x P(x)$ is true or false may still be wrong, but we require that the program should provide the correct output even if its belief is wrong. We claim that this is enough for interpreting proofs in classical arithmetic (see [2, 3, 5], the examples in the present paper may give some hint of how it may be done).

In [3, 5] we have studied the case where answers are decidable arithmetical statements, which can be definitely known to be true, and considered the case where the value of a realizer is either a singleton or it is empty; we call such a case *deterministic*. Here we extend our approach in two directions. First we consider arbitrary finite sets of answers as values of realizers, which are not necessarily compatible among each other, so that the actual progress of knowledge will consist of a *non-deterministic* choice of the answers to be added to the current state. Second, we relax the requirement of decidability of the arithmetical statements in the universe of answers.

While non determinism is only a matter of efficiency, abstractly representing the parallel answers of several questions, admitting non decidable statements as answers rises an issue. The problem is that arbitrary arithmetical statements may contain nested quantifiers: think for example of an instance of excluded middle of the shape $\forall x \exists y Q(x,y) \vee \exists x \forall y \neg Q(x,y)$, where Q is recursive. Learning its truth might consist in producing a number of evidences $\neg Q(n_0, m_0), \ldots, \neg Q(n_0, m_k)$, making the statement $\forall y \neg Q(n_0, y)$ consistent with the current state, so that it can safely be added to the state. However, the statement $\forall y \neg Q(n_0, y)$ is not definitely true as it is the case for the statements $\neg Q(n_0, m_i)$. In fact if the next step produces $Q(n_0, m_{k+1})$ for some m_{k+1}, then $\forall y \neg Q(n_0, y)$ is no longer consistent with the new knowledge, and has to be dropped form the current knowledge state. Hence if we admit arbitrary arithmetical statements as answers (even if just two-quantifiers statements), states of knowledge in a learning process do not increase monotonically. We call this form of learning *non-monotonic learning*. In the case of non-monotonic learning it is much harder to find under which (reasonable) assumptions about realizers pre-fix points exist.

We solve this problem by introducing a topology over the set of answers and a stratification into levels of answers, modeling the degrees of their logical complexity: answers of level $0, 1, 2, \ldots$ represent statements with $0, 1, 2, \ldots$ nested quantifiers over individuals. Realizers are certain continuous functions over the space of states of knowledge. We then define a reduction relation over states such that, whenever a consistent subset of the new answers is produced by some fixed realizer, answers in conflict with them are removed from the given state. We prove that any reduction sequence out of a non-contradictory state (that we call *sound*) generates a sequence of states reaching a pre-fix point of the realizer w.r.t. subset inclusion, hence the sequence must be finite.

The plan of the paper is as follows. In §2 we introduce the concept of state of knowledge and the state topology. Then we define realizers and a reduction relation over states depending on a realizer r, which is the non-deterministic algorithm searching a pre-fix point of r. In §3 we discuss the use of knowledge space to design and prove correct monotonic and non-monotonic learning algorithms, which are the effective counterpart of the non-constructive principles used in the proofs from which we designed the algorithms. In §4 we prove that the set of states from which any execution of the algorithm terminates is an open set in the state topology. In §5 we use this fact to establish that if there is some reduction sequence of length ω out of some state, then there is a reduction sequence of length ω_1 out of the same state. Eventually in §6, we prove that reduction sequences of length ω_1 do not exist, so that we conclude that all reduction sequences are of finite length, hence any execution of the non-deterministic algorithm terminates. Finally in §7 we discuss some related works and conclude.

2. Knowledge spaces, realizers and reduction

We introduce the basic concepts and recall the definitions from [6], then we discuss them.

Definition 2.1. (Knowledge spaces)

1. A *knowledge space* is a tuple $(\mathbb{A}, \sim, \mathsf{lev})$ where $\mathbb{A}$ is a countable set, $\sim$ is an equivalence relation over $\mathbb{A}$, $\mathsf{lev} : \mathbb{A} \to \mathbb{N}$ is a map such that if $x \sim y$ then $\mathsf{lev}(x) = \mathsf{lev}(y)$.

2. A subset $X \subseteq \mathbb{A}$ is a *knowledge state* if for any distinct $a, b \in X$ it is the case that $a \not\sim b$.

3. The set $\mathbb{S}(\mathbb{A})$ (shortly $\mathbb{S}$ when $\mathbb{A}$ is understood) of all knowledge states of $\mathbb{A}$, is taken with the *state topology*, which is generated by the sub-basics:

$$A_a = \{X \in \mathbb{S} \mid a \in X\} \quad \text{and} \quad B_a = \{X \in \mathbb{S} \mid X \cap [a]_\sim = \emptyset\}.$$

We briefly discuss our definitions. The set $\mathbb{A}$ is the universe of answers. It is stratified according to the map lev, representing the logical complexity of the answers: we interpreted $\mathsf{lev}(x) = n$ as "a represents a statement with n quantifiers". The set $\mathbb{A}$ is equipped with the equivalence relation $a \sim b$, representing the fact that a and b are answers to the same question, so that questions could be identified with equivalence classes $[a]_\sim$ of answers. This explains why we asked that two answers to the same question have the same level. A state of knowledge X is a set of answers including no equivalent answers. To put otherwise, X is a set of answers such that each equivalence class of answers, that is, each question, has at most one answer in X. X is a partial function from questions to answers: any question may have one answer or no answer at all in X. We disallow multiple answers, thought they are representable by adding to the knowledge space several copies of the same question.

When X belongs to the open set A_a we have the *positive information* that the question $[a]_\sim$ has answer $a \in X$; if instead $X \in B_a$ we have the *negative information* that the question $[a]_\sim$ has no answer in X. Let X be any state and O be any open set. By unfolding Definition 2.3, we obtain that $X \in O$ if and only if there are two finite (possibly empty) sets $\{[a_1]_\sim, \ldots, [a_k]_\sim\}$ and $\{[b_1]_\sim, \ldots, [b_h]_\sim\}$ of questions, such that $X \in A_{a_1} \cap \cdots \cap A_{a_h} \cap B_{b_1} \cap \cdots \cap B_{b_k} \subseteq O$; that is, there are finitely many questions $[a_1]_\sim, \ldots, [a_k]_\sim$ of which some answer is known at X, and finitely many questions $[b_1]_\sim, \ldots, [b_h]_\sim$ of which it is known at X that yet no answer has been given.

One could think of $[a]_\sim$ as a memory cell with values from the set $[a]_\sim \cup \{\emptyset\}$, where $\emptyset$ is the special value for "no answer". If X is a state of knowledge then we call $\mathtt{lk}([a]_\sim, X) = [a]_\sim \cap X \in \mathcal{P}_{\mathit{fin}}(\mathbb{A})$ the "lookup function", yielding the value of the "cell" $[a]_\sim$ at state X. Then the state topology is the smallest topology over $\mathbb{S}$ making the lookup map continuous.

In the following we take $\mathbb{A}$ and $\mathtt{Bool} = \{\mathsf{T}, \mathsf{F}\}$ with the discrete topology and $\mathbb{A} \times \mathbb{S}$ with the product topology. We define the notion of valuation, telling whether an answer $a \in \mathbb{A}$ is true according to a knowledge state X, of sound (that is, self-consistent) state, and of realizer, which is a knowledge-extending map.

Definition 2.2. (Layered Valuations and Realizers)

1. A *layered valuation* over $(\mathbb{A}, \sim, \mathsf{lev})$, shortly a *valuation*, is a total continuous mapping $\mathtt{tr} : \mathbb{A} \times \mathbb{S} \to \mathtt{Bool}$ such that
$$\mathsf{tr}(a, X) = \mathsf{tr}(a, \{x \in X \mid \mathsf{lev}(x) < \mathsf{lev}(a)\}).$$
2. A state $X \in \mathbb{S}$ is *sound* if $\mathsf{tr}(x, X) = \mathsf{T}$ for all $x \in X$.
3. A *realizer* w.r.t. the valuation $\mathtt{tr}$ is a total continuous map $r : \mathbb{S} \to \mathcal{P}_{\mathit{fin}}(\mathbb{A})$, where $\mathcal{P}_{\mathit{fin}}(\mathbb{A})$ is taken with the discrete topology, which is such that:
$$\forall X \in \mathbb{S}\ \forall a \in r(X) \setminus X.\ X \cap [a]_\sim = \emptyset\ \ \&\ \ \mathtt{tr}(a, X) = \mathsf{T}.$$

The valuation $\mathsf{tr}(a, X)$ of an answer a (think of a logical formula) is relativized to a knowledge state X, and only depends on the answers in X of a lower level than a. If we think of answers of level n as n-quantifiers statements, then it is clear that the truth of an answer of level n should be defined in term of answers of lower level. For instance, in §1, the truth value of the formula $\forall y \neg Q(n_0, y)$, of level 1, was affected by the valuation of the formula $Q(n_0, m_{k+1})$ of level 0, but not vice versa. A sound state is a state which cannot contain answers that are false w.r.t. the state itself: a sound state is self-consistent.

A realizer r takes $X \in \mathbb{S}$ and returns finitely many answers $r(X)$ which either are in X or they are not equivalent to any other answer in X (hence they are "new") and that are true w.r.t. X. A naive way to model a step in the learning process guided by r is by moving from X to $X \cup r(X)$. However nothing prevents that for some $a, b \in r(X)$ it is $a \sim b$, so that we cannot add both a and b to X, because we forbid two answers to the same question. Worse than this, it is possible that $X \cup \{a\}$ is not sound for some $a \in r(X)$, although it might be true that X is sound and $\mathtt{tr}(a, X) = \mathsf{T}$. In fact this is precisely the case we outline in §1, in which a level 0 counterexample is discovered to some previously accepted answer of level 1. In order to search for a pre-fix point of r we have to define a way to select some new answers from $r(X)$ and adding them to X, simultaneously removing all answers becoming false.

The simplest (and the less efficient) method we found is the following. We select some non-empty subset $s \subseteq r(X) \setminus X$ of answers all of the same level n (we call this an *homogeneous* subset), and we simultaneously remove from X all answers of level $> n$ because, as in the example of §1, these answers *could* become false in X if the answers of level n in X change. In order to provide a formal definition of this method, we have to define a notion of restriction first. Given $n \in \mathbb{N}$ and a state X we define the subsets of X of level below, above and equal to n:

$$X \restriction_{<n} = \{x \in X \mid \mathsf{lev}(x) < n\}, \;\; X \restriction_{>n} = \{x \in X \mid \mathsf{lev}(x) > n\}, \;\; X \restriction_{=n} = \{x \in X \mid \mathsf{lev}(x) = n\}.$$

We also write $X \restriction_{\leq n} = X \restriction_{<n} \cup X \restriction_{=n}$. We denote by $\mathbb{S}_{\mathsf{fin}}$ the set of finite states; let $s, s', t, t', \ldots$ range over $\mathbb{S}_{\mathsf{fin}}$. We define a reduction relation $\twoheadrightarrow^r$ on $X \in \mathbb{S}$, adding and removing answers from X according to what r says. $\twoheadrightarrow^r$ translates the method with homogeneous sets we outlined above, and forms one step of a search for a pre-fix point of r.

Definition 2.3. (Reduction)
We say that a state $s \in \mathbb{S}_{\mathsf{fin}}$ is *homogeneous* if $s \neq \emptyset$ and for some $n \in \mathbb{N}$, $\mathsf{lev}(x) = n$ for all $x \in s$; then we write $\mathsf{lev}(s) = n$. For any homogeneous s of level n we define a map $\mathtt{R}_s : \mathbb{S} \to \mathbb{S}$ by:

$$\mathtt{R}_s(X) = X \restriction_{\leq n} \cup\, s = \{x \in X \mid \mathsf{lev}(x) \leq n\} \cup\, s.$$

Then, given a realizer r and an homogeneous s we define the binary *reduction relation* over $\mathbb{S}$ by:

$$X \twoheadrightarrow^{s,r} Y \Leftrightarrow s \subseteq r(X) \setminus X \;\;\&\;\; Y = \mathtt{R}_s(X).$$

We say that X *reduces to* Y *in one step* and we write $X \twoheadrightarrow^r Y$ if $X \twoheadrightarrow^{s,r} Y$ for some homogeneous s. The condition that s is a non-empty state is necessary for termination, while the restriction to homogeneous states, namely with answers of the same level, could be dropped; we make this assumption because it is a mild limitation to the "parallelism" of the algorithm, while it simplifies the technical treatment.

Clearly, for a large knowledge state this method is really inefficient: if we add answers of level 0 to some X then we remove all answers of level 1 from X, including those which are unrelated with the answers we added, and whose truth value therefore will not change. In order to produce an efficient algorithm we should record whether the truth value of answer may depend on the truth value of another answer, and we should not remove an answer of level 1 if we add an answer of level 0 unrelated to it. This will be the task of another paper: in this paper we prove termination and correctness for $\twoheadrightarrow^r$.

As an immediate consequence of the definitions of $\mathtt{tr}$, r and $\twoheadrightarrow^r$ we establish:

Lemma 2.4.

1. $X \twoheadrightarrow^r Y \;\;\&\;\; X \in \mathbb{S}_{\text{fin}} \Rightarrow Y \in \mathbb{S}_{\text{fin}}$.

2. $X \twoheadrightarrow^r Y \;\;\&\;\;$ X is sound $\Rightarrow Y$ is sound.

3. $\neg \exists Y.\, X \twoheadrightarrow^r Y \Leftrightarrow r(X) \subseteq X$.

Let us fix some realizer r. A reduction sequence of length n from X to Y is a sequence $X_0, \ldots, X_n$ such that $X = X_0 \twoheadrightarrow^r X_1 \twoheadrightarrow^r \ldots \twoheadrightarrow^r X_n = Y$. An infinite reduction sequence out of X is an endless sequence $X = X_0 \twoheadrightarrow^r X_1 \twoheadrightarrow^r \ldots \twoheadrightarrow^r X_n \ldots$ of reductions. For any integer $n \in \mathbb{N}$ we say that X reduces to Y in n steps and we write $X \twoheadrightarrow^r_n Y$ if there is a reduction sequence from X to Y of length n. We write $X \twoheadrightarrow^r_* Y$ if $X \twoheadrightarrow^r_n Y$ for some $n \in \mathbb{N}$.

We observe that X is a pre-fix point of r, that is $r(X) \subseteq X$, if and only if there is no homogeneous set $s \subseteq r(X)$ such that $X \cap s = \emptyset$, that is if and only if for all $Y \in \mathbb{S}$ we have $X \not\twoheadrightarrow^r Y$. If $\sim$ is decidable and both r and $\mathtt{tr}$ are relative recursive then we can see $Y \twoheadrightarrow^r Z$ as the one step relation of a non-deterministic algorithm computing a pre-fix point X of r starting with some $X_0 \in \mathbb{S}$; then such an X, if any, can be seen as a result of the computation out of X_0. By lemma 2.4 we know that if we move from some finite sound state s_0, e.g. $\emptyset$, the reduction relation $\twoheadrightarrow^r$ generates a tree with finite and sound states as nodes, which is finitary because $r(X)$ is finite even for infinite X so that there can be only finitely many homogeneous $s \subseteq r(X)$. In particular the relation $X \twoheadrightarrow^r Y$ is decidable for finite X and Y, and relative recursive in general.

We say that $X \in \mathbb{S}$ is *strongly normalizing* if all reduction sequences out of X are finite. We denote by $\mathtt{SN}$ the set of all strongly normalizing states. Our thesis is that $\mathtt{SN} = \mathbb{S}$, namely that the reduction tree out of any X is finite. This implies that if $s \in \mathbb{S}_{\text{fin}}$ and s is sound we can effectively find a finite and sound pre-fix point t of r by reducing s.

Remark 2.5. Before illustrating the concept of realizer by examples, and embarking in the proof of $\mathtt{SN} = \mathbb{S}$, let us observe that the existence of pre-fix points of realizers essentially depends on both continuity and totality. Consider the following example: let $\mathbb{A} = \mathbb{N}$ and $\sim$ be the identity, so that any subset $X \subseteq \mathbb{N}$ is a state of knowledge. Also let $\mathsf{lev}(n) = n$ for all $n \in \mathbb{N}$ and let $\mathtt{tr}$ be the constantly true function (all answers are true). Define the mapping:

$$r(X) = \min\{n \mid n \notin X\},$$

so that if $X = \{0, \ldots, n\}$ is an initial segment of $\mathbb{N}$ then $r(X) = \{n+1\}$, hence $r(X) \not\subseteq X$ (which is the case for any $X \neq \mathbb{N}$). Observe that r is continuous, since $r^{-1}(\{n\}) = \{X \mid n \notin X \;\;\&\;\; \forall i < n.(i \in X)\}$, and this is a finite intersection of sub-basics. This implies that the reduction $\{0, \ldots, n\} \twoheadrightarrow^r \{0, \ldots, n+1\}$ is infinite.

The key point here is that r is defined everywhere but in $\mathbb{N}$, since $r(\mathbb{N}) = \min\{n \mid n \notin \mathbb{N}\} = \min(\emptyset)$, so that r is *not* a total function. Moreover it doesn't admit a continuous total extension. Toward a contradiction suppose that r_I is the total extension of r such that $r_I(\mathbb{N}) = I$, for some finite $I \subset \mathbb{N}$. If I is not a singleton set then $X \neq \mathbb{N}$ implies $r_I(X) = r(X) \neq I$ as $r(X)$ is a singleton; hence $r_I^{-1}(I) = \{\mathbb{N}\}$ which is not open. So suppose that $I = \{n\}$ for some $n \in \mathbb{N}$; then we have $r_I^{-1}(I) = \{X \mid n \notin X \;\;\&\;\; \forall i < n.(i \in X)\} \cup \{\mathbb{N}\}$.

We claim that $r_I^{-1}(I)$ is not the union of basic opens. If it were, we would have $\mathbb{N} \in O$ for some basic open $O \subseteq r_I^{-1}(I)$, but there is no such a set. Indeed, if $\mathbb{N} \in O$ then we have no negative condition $B_a \subseteq O$, therefore $O = A_{i_1} \cap \cdots \cap A_{i_k} = \{X \mid i_1, \ldots, i_k \in X\}$ for some $i_1, \ldots, i_k \in \mathbb{N}$. Let $m = \max\{i_1, \ldots, i_k, n\}$: then $\{0, \ldots, m\} \in O$ because $i_1, \ldots, i_k \in \{0, \ldots, m\}$. By definition of r_I we have $r_I(\{0, \ldots, m\}) = \{m+1\} \neq \{n\} = I$, therefore $\{0, \ldots, m\} \notin r_I^{-1}(\{n\}) = r_I^{-1}(I)$. We conclude that $\{0, \ldots, m\} \in O \setminus r_I^{-1}(I)$, contradicting $O \subseteq r_I^{-1}(I)$.

In conclusion r cannot be extended to a realizer r' over $\mathbb{S}(\mathbb{N}) = \mathcal{P}(\mathbb{N})$, because either r' is *not* a total function or it is *not* continuous.

3. Examples of knowledge spaces and realizers

In each of the examples below we consider the problem on finding some $u \in U$, the universe of the problem, satisfying a certain property; then we introduce a knowledge space $(\mathbb{A}, \sim, \mathsf{lev})$ and a valuation $\mathtt{tr}$ to describe the current knowledge of a program looking for a solution of the problem. As in [7], we decompose the program in a continuous map $\alpha : \mathbb{S} \to U$, computing some perfectible guess $\alpha(X) \in U$ for a solution according to the current knowledge $X \in \mathbb{S}$, and a realizer $r : \mathbb{S} \to \mathcal{P}_{\mathit{fin}}(\mathbb{A})$ which describes how the program can update its knowledge state $X \in \mathbb{S}$. Then we say that the pair (α, r) solves the problem of finding some $u \in U$ with a decidable property P, which was written $r \vdash \alpha : U$ in [7], if whenever X is a pre-fix point of r then $\alpha(X) \in U$ satisfy the property P. A pre-fix point X of r (some $X \in \mathbb{S}$ such that $r(X) \subseteq X$) may be found by reduction, as outlined in the previous section. We did not prove yet that all reductions terminate, but, in the particular cases we consider, we will provide some direct argument for termination.

Continuity of r, $\mathtt{tr}$ plays an essential role. Recall that a realizer r is continuous if the finite set of answers $r(X)$ depends on the answers to finitely many questions of the form: $\exists a_1 \in [b_1]_\sim \cap X$, …, $\exists a_n \in [b_n]_\sim \cap X$. Each answer can be either the statement "there is no $a_i \in [b_i]_\sim \cap X$" or it may consist of the unique a such that $a \in [b_i]_\sim \cap X$. We will consider only continuous maps r which are relative recursive w.r.t. an oracle for the lookup function $\mathtt{lk}$, which is a computable function when X is finite. The same remark applies to the continuous map $\mathtt{tr}$.

Our first example is Coquand's basic example [9], which requires a knowledge space with just one level, corresponding to monotonic learning. Our second example is a variant of Coquand's example: we use it to compare a deterministic sequential search for a pre-fix point with a non-deterministic parallel search. We include a third example having two levels of answers, corresponding to the simplest cases of non-monotonic learning. We know of examples with three or more levels e.g. coming from Ramsey Theorem, Higman's Lemma or Kruskal Lemma, but they are too long to be included in this paper.

3.1. One-level knowledge spaces: solving an inequation by monotonic learning

Assume $f, a : \mathbb{N} \to \mathbb{N}$ are unary functions computable over the set $\mathbb{N}$ of natural numbers. We consider the problem of finding some value for x in $\mathbb{N}$ solving the inequation $f(x) \leq f(a(x))$. This example slightly generalizes Coquand's inequation $f(x) \leq f(x + 27)$ [9], and can be solved in a similar way.

Classical arithmetic proves that f has a minimum value $f(k)$: by definition we have $f(k) \leq f(z)$ for all $z \in \mathbb{N}$, and in particular that $f(k) \leq f(a(k))$. Thus the inequation has a solution. However, this argument seems helpless to effectively find a solution, except by blind search, because there is no algorithm computing a k such that $f(k)$ is the minimum of $f(\mathbb{N})$. As a matter of fact any algorithm can use just finitely many values of f, while an infinite information is needed to establish that there exists no q such that $f(q) < f(k)$. The classical proof bypasses the obstacle and requires, uniformly in p, a perfect knowledge about the existence of some q such that $f(p) > f(q)$.

Is it possible to turn the classical argument above into an algorithm that is not brute force? The basic remark is that it is not necessary to find a minimum of f; it is enough to find some x such that $f(x)$ is "low enough" to solve $f(x) \leq f(a(x))$. To make this precise, we introduce a knowledge space $(\mathbb{A}, \sim, \mathsf{lev})$ and a valuation $\mathtt{tr}$ with only one level, describing what a program knows about the minimum point of f after finitely many evaluations of f. The solution space in this case is $U = \mathbb{N}$ and learning is monotonic (we are never forced to reject an answer previously inserted in the knowledge space).

The set $\mathbb{A}$ of answers consists of all statements of the form $(f(p) > f(q))$ and the truth value $\mathtt{tr}((f(p) > f(q)), X)$ is the arithmetical truth value of $(f(p) > f(q))$, independently from the knowledge state X. The function $\mathtt{tr}(a, \_)$ is continuous just because it is constant. There is a unique level of answers: the map lev is constantly equal to 0. For each $p \in \mathbb{N}$, we represent the question: *"is there some q such that $f(p) > f(q)$?"* by the set of its possible answers: $\{(f(p) > f(q)) \mid q \in \mathbb{N}\}$. These sets are the equivalence classes of $\sim$. Any such question corresponds to a basic open in the State Topology and piking one of its possible answers $(f(p) > f(q))$ corresponds to a step in the computation of the realizer.

Given a knowledge state X, we consider the hypothesis "p is a minimum point of f" as true in X if X *knows of no counterexample*, that is if there are no answers $(f(p) > f(q)) \in X$ testifying the existence of a counterexample. Observe that the hypothesis "p is a minimum point of f" as it is cannot be added to the set of answers, as it depends on an infinite information, so that the continuous function $\mathtt{tr}$ could not be defined on it. Also note that, whenever we add a true counterexample $(f(p) > f(q))$ to X, we will never change our mind and we are never forced to remove it from X: this is why in this case the learning process is monotonic.

The next step is to design a strategy finding a solution of $f(x) \leq f(a(x))$, consisting of a realizer r updating the knowledge state of the program. We consider an auxiliary function $\alpha : \mathbb{S} \to \mathbb{N}$ reading off a solution $\alpha(X)$ when X is a pre-fix point of r. Both maps are relative recursive w.r.t. the lookup map $\mathtt{lk}$, hence are continuous, and are defined as follows. Given a knowledge state X we arbitrarily set $k_0 = 0$; if $(f(k_0) > f(k_1)) \in X$ for some k_1 then X knows of the counterexample $f(k_1)$ to the fact that $f(k_0)$ is a minimum in the image of f. Then we proceed by guessing k_1 as a candidate for a minimum point of f, and so on until a strictly decreasing sequence $f(k_0) > f(k_1) > \ldots > f(k_r)$ in $\mathbb{N}$ is produced. By well-foundation of the ordering of $\mathbb{N}$ this sequence must be finite, so that there is some k_r such that $(f(k_r) > f(q)) \notin X$ for any $q \in \mathbb{N}$, that is, such that the equivalence class $\{(f(p) > f(q)) \mid q \in \mathbb{N}\}$ has empty intersection with X. Such k_r may be assumed to be a minimum point of f, although not in true sense of the word, but only according to the knowledge in X. Then we have effectively defined $\alpha(X) = k_r$ using finitely many questions of the form $\exists a \in [b]_\sim \cap X$.

The realizer r checks whether k_r given above solves $f(x) \leq f(a(x))$. If this is the case, then we set $r(X) = \emptyset$, and $\alpha(X) = k_r$ yields a solution, no matter whether k_s is an actual minimum point of f or not. As we required in §1, the correctness of the result is independent from the correctness of the temporary hypothesis about the minimum point. If $\alpha(X)$ is not a solution, then $f(k_r) > f(a(k_r))$, and we put $r(X) = \{(f(k_r) > f(a(k_r)))\}$. Since $r(X)$ is effectively defined using only finitely many questions of the form $\exists a \in [b]_\sim \cap X$, that is finitely many applications of $\mathtt{lk}$, we have that r is relative recursive w.r.t. $\mathtt{lk}$, hence continuous.

Now we apply the method designed in §2 for searching a pre-fix point. Whenever $r(X) \not\subseteq X$, our method adds $(f(k_r) > f(a(k_r)))$ to X, and starts again. In this case, our method never removes answers from X. The search for a fix point always terminates: as we already said, the length of a chain $f(k_0) > f(k_1) > \ldots > f(k_r)$ is bounded above by $f(k_0)$.

Observe that we could easily improve this algorithm: instead of the whole knowledge state X we may keep a single variable x. We assign $x := 0$, and whenever $f(x) > f(a(x))$ we re-assign $x := a(x)$. The knowledge space is intended to explain the principle involved in the search of a solution, and to design a first version of a learning program. Usually this first version may work with much less information and can be improved.

3.2. One-level knowledge spaces: a system of inequations and monotonic learning

The main point in the previous example is that we turn a reasoning by cases about the truth of a Σ^0_1-statement, like $\exists y.f(x) > f(y)$, into the search for a pre-fix point in a knowledge space with only one level. By adapting this idea we may find many other examples, all concerning classical proofs in which we reason by cases about the truth of a Σ^0_1-statement, and all requiring one level of answers and monotonic learning.

Sometimes, $r(X)$ may include several alternative choices for extending X (non-determinism). Some of these choices may include two or more answers: we interpret this as the fact that the pre-fix point of r is found by a parallel computation. To illustrate the point we consider yet another variant of Coquand's example, which still requires one level of answers and monotonic learning, but in which non-determinism and parallel computations play an essential role. Assume $f, g : \mathbb{N} \to \mathbb{N}$ are unary functions and $a, b : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ are binary functions, all over the set $\mathbb{N}$ of natural numbers. We consider the problem of finding some values for x, y in $\mathbb{N}$ solving the system of inequations:

$$\begin{cases} f(x) \leq f(a(x,y)) \\ g(y) \leq g(b(x,y)) \end{cases}$$

Classical arithmetic proves that f, g have minimum values $f(k), g(h)$: by definition we have $f(k) \leq f(z)$ and $g(h) \leq g(t)$ for all $z, t \in \mathbb{N}$, and in particular: $f(k) \leq f(a(k,h))$ and $g(h) \leq g(b(k,h))$. Thus the system has some solution. In §3.1 we remarked that there is no algorithm computing a k such that $f(k)$ is the minimum of $f(\mathbb{N})$. The classical proof requires it, instead.

How do we solve the system, apart from checking all possible pairs k, h? As in 3.1, the idea is to find $f(k)$, $g(h)$ "low enough" to solve the system. We formalize this idea by introducing a knowledge space $(\mathbb{A}, \sim, \mathsf{lev})$ and a valuation tr with only one level, describing what a program knows about the minimum points of f and g after finitely many evaluations of f and g. The solution space in this case is $U = \mathbb{N} \times \mathbb{N}$. Again, learning is monotonic, as we are never forced to reject an answer previously inserted in the knowledge space.

The set $\mathbb{A}$ of answers consists of all statements of the form $(f(p) > f(q))$ and $(g(p) > g(q))$, where the truth value $\mathsf{tr}((f(p) > f(q)), X)$ is the arithmetical truth value of $(f(p) > f(q))$, and the same for $\mathsf{tr}((g(p) > g(q)), X)$, independently from the knowledge state X. Again, the function $\mathsf{tr}(a, \_)$ is continuous just because it is constant. There is a unique level of answers: the map lev is constantly equal to 0. The equivalence classes of $\sim$ are all sets of answers of the form $\{(f(p) > f(q) \mid q \in \mathbb{N}\}$, or of the form $\{(g(p) > g(q)) \mid q \in \mathbb{N}\}$, for some $p \in \mathbb{N}$. We interpret any equivalence class $\{(f(p) > f(q)) \mid q \in \mathbb{N}\}$ as the question: *"is there some q such that $f(p) > f(q)$?"*, and any equivalence class $\{(g(p) > g(q)) \mid q \in \mathbb{N}\}$ as the question: *"is there some q such that $g(p) > g(q)$?"* Any such question corresponds to a basic open in the State Topology and to a possible step in the computation of the realizer. Given a knowledge state X, we consider the hypothesis "p is a minimum point of f" as true in X if X *knows of no counterexample*, that is if there are no answers $(f(p) > f(q)) \in X$ testifying the existence of a counterexample, and the same for g. When we add a true counterexample $(f(p) > f(q))$ or $(g(p) > g(q))$ to X, we cannot ever change our mind about it, and we are never forced to remove it from X: therefore in this case the learning process is monotonic. The set $\mathbb{A}$ of answers consists of all statements of the form $(f(p) > f(q))$ and $(g(p) > g(q))$, where the truth value $\mathtt{tr}((f(p) > f(q)), X)$ is the arithmetical truth value of $(f(p) > f(q))$, and the same for $\mathtt{tr}((g(p) > g(q)), X)$, independently from the knowledge state X. Again, the function $\mathtt{tr}(a, \_)$ is continuous just because it is constant. There is a unique level of answers: the map lev is constantly equal to 0. The equivalence classes of $\sim$ are all sets of answers of the form $\{(f(p) > f(q) \mid q \in \mathbb{N}\}$, or of the form $\{(g(p) > g(q)) \mid q \in \mathbb{N}\}$, for some $p \in \mathbb{N}$. We interpret any equivalence class $\{(f(p) > f(q)) \mid q \in \mathbb{N}\}$ as the question: *"is there some q such that $f(p) > f(q)$?"*, and any equivalence class $\{(g(p) > g(q)) \mid q \in \mathbb{N}\}$ as the question: *"is there some q such that $g(p) > g(q)$?"* Any such question corresponds to a basic open in the State Topology and to a possible step in the computation of the realizer. Given a knowledge state X, we consider the hypothesis "p is a minimum point of f" as true in X if X *knows of no counterexample*, that is if there are no answers $(f(p) > f(q)) \in X$ testifying the existence of a counterexample, and the same for g. When we add a true counterexample $(f(p) > f(q))$ or $(g(p) > g(q))$ to X, we cannot ever change our mind about it, and we are never forced to remove it from X: therefore in this case the learning process is monotonic.

The next step is to design a strategy finding a solution of the system above, consisting of a realizer r updating the knowledge state of the program, and an $\alpha : \mathbb{S} \to \mathbb{N} \times \mathbb{N}$ computing a solution $\alpha(X)$ when X is a pre-fix point of X. Both maps are relative recursive w.r.t. the lookup map $\mathtt{lk}$, hence continuous, and are defined as follows. Given a knowledge state X we arbitrarily set $k_0 = 0$; if $(f(k_0) > f(k_1)) \in X$ for some k_1 then X knows of the counterexample $f(k_1)$ to the hypothesis that $f(k_0)$ is a minimum in the image of f. Then we proceed by guessing k_1 as a candidate for a minimum point of f, and so on until a strictly decreasing sequence $f(k_0) > f(k_1) > \ldots > f(k_r)$ in $\mathbb{N}$ is produced. By well-foundation of the ordering of $\mathbb{N}$ this sequence must be finite, so that there is some k_r such that $(f(k_r) > f(q)) \notin X$ for any $q \in \mathbb{N}$, that is, such that $\{(f(p) > f(q)) \mid q \in \mathbb{N}\} \cap X = \emptyset$. Such k_r can be assumed to be a minimum point of f, although not in the true sense of the word but only according to the knowledge in X. In the same way we can find a temporary guess for the minimum point h_s of g. We set $\alpha(X) = (k_r, h_s)$. Then we have effectively defined $\alpha(X)$ using finitely many questions of the form $\exists a \in [b]_\sim \cap X$, that is, finitely many applications of the lookup function $\mathtt{lk}$, therefore α is relative recursive w.r.t. $\mathtt{lk}$, hence continuous.

The realizer r checks whether k_r, h_s are a solution of the problem. If they are such, then we set $r(X) = \emptyset$, and $\alpha(X) = (k_r, h_s)$ yields a solution, no matter whether (k_s, h_s) are actual minimum points or not. As we asked in §1, the correctness of the result is independent from the correctness of the temporary hypotheses about minimum points. If (k_r, h_s) are not a correct result, then either $f(k_r) > f(a(k_r, h_s))$, or $g(h_s) > g(b(k_r, h_s))$, or both. In the first case we put $r(X) = \{(f(k_r) > f(a(k_r, h_s)))\}$, in the second one $r(X) = \{(g(h_s) > g(b(k_r, h_s)))\}$, in the third $r(X) = \{(f(k_r) > f(a(k_r, h_s))), (g(h_s) > g(b(k_r, h_s)))\}$. Since $r(X)$ is effectively defined using only finitely many questions of the form $\exists a \in [b]_\sim \cap X$, that is, finitely many applications of $\mathtt{lk}$, we have that r is relative recursive w.r.t. $\mathtt{lk}$, hence continuous.

Whenever $r(X) \not\subseteq X$, the method designed in §2 for searching a pre-fix point adds some non-empty consistent subset $\emptyset \subset Y \subseteq r(X)$ to X, and starts again. The method never removes answers from X.

Knowing that the search for a pre-fix point of the realizer r always terminates, as it will be established in the next sections, we conclude that some $\alpha(X) = (k_r, h_s)$ solving the system will be eventually found. For this particular problem, however, we have a more direct termination argument, since the length of the chains $f(k_0) > f(k_1) > \ldots > f(k_r)$ and $g(h_0) > g(h_1) > \ldots > g(h_s)$ are bounded above by $f(k_0)$ and $g(h_0)$ respectively.

Observe that while defining the process that uses r some arbitrary choices are made, so that the actual solution is not unique in general, yet it is always correct. We may replace r by some deterministic r' solving the same system, by arbitrarily defining some $r'(X) \subseteq r(X)$ which is a singleton if $r(X) \neq \emptyset$. For instance, we may define $r(X) = \{(f(k_r) > f(a(k_r, h_s)))\}$ whenever this answer is true, and $r(X) = \{(g(h_s) > g(b(k_r, h_s)))\}$ if $f(k_r) > f(a(k_r, h_s))$ is false and $g(h_s) > g(b(k_r, h_s))$ is true. However, if we run some random simulations, we can easily find examples in which there are pre-fix points found by a non-deterministic r that cannot be found by any deterministic $r'(X) \subseteq r(X)$.

Again, we could improve the algorithm we found: instead of the whole knowledge state X we might just keep x, y. We start assigning $x := 0$ and $y := 0$ (any other values would work). Each time $a(x, y)$ is a counterexample to the hypothesis $\forall z \in \mathbb{N}.f(x) \leq f(z)$ we re-assign $x := a(x, y)$; similarly we re-assign $y := b(x, y)$ whenever $b(x, y)$ is a counterexample to $\forall t \in \mathbb{N}.g(x) \leq g(t)$, and these events can happen at the same time. The computation always terminates into some correct result, but the actual result may depend on the non-deterministic choices which have been made during the computation.

3.3. Two-levels knowledge spaces: Stolzelberg Example and monotonic learning

There are classical proofs which cannot be interpreted in a knowledge space with just one level and with a monotonic search of the pre-fix point of a realizer. This is the case of a proof using the Infinite Pigeonhole Principle (IPP). We include one example.

Assume $\sigma : \mathbb{N} \to \{b, w\}$ is some infinite sequence of black/white balls. Then IPP says: either there are infinitely many black balls, or infinitely many white balls. The classical proof is immediate: toward a contradiction assume that there are only finitely many black balls, namely for some $n \in \mathbb{N}$ all black balls have index $< n$, and that there are only finitely many white balls, namely for some $m \in \mathbb{N}$ all white balls have index $< m$. If we set $p = \max(n, m)$, then the ball $\sigma(p)$ is neither black nor white, which is a contradiction. As a corollary, for any $k \in \mathbb{N}$ either there are k black or k white balls.

The IPP cannot be proved in Intuitionistic Arithmetic [1], because one cannot decide which is finite among two sets in case there is exactly one such. The degree of undecidability of this question is higher than the degree of the statement $\exists y.f(x) > f(y)$ considered in the previous examples. Indeed, the statement $\exists y.f(x) > f(y)$ is semi-decidable, while this is not the case of the predicate "being finite" and the predicate "being infinite", which require two nested quantifiers over integers. For this reason, a space of knowledge with just one level is not enough to give a concrete counterpart of any proof essentially using IPP.

We claim that we can design a two-level knowledge space to describe what we know about the existence of infinitely many black balls, or infinitely many white balls, and we use it to solve the problem of finding 10 balls (say) of the same color. We stress that the issue is not to find an algorithm whatsoever, which is otherwise very easy just by counting white and black balls until at worst $(10+10-1)$ balls have been checked. Rather our purpose is to illustrate how the non constructive argument sketched above can be interpreted as an algorithm. Incidentally, the algorithm that we obtain out of the constructive content of the classical proof that there are k balls of the same color, for any k, is the optimal one, solving a problem proposed by Gabriel Stolzenberg [12].

We define a knowledge space with two levels. Level 0 answers are sentences of the shape: "the t-th ball is black (white), with $t \geq n$", and level 1 answers are sentences of the shape "all black (white) balls have index $< n$". Level 0 answers are immediately checked, so that their valuation does not depend on the actual state of knowledge; on the contrary level 1 answers are relativized to the state, as if we were adding to their meaning the clause: "as far as we know". Now the task of the realizer is not to decide whether there are infinite black or white balls in σ (a non-effective task), rather to verify whether the current state of knowledge X is sufficient to select 10 balls of the same color, or X is incomplete w.r.t. that goal, so that some extra knowledge (namely some consistent set $\emptyset \subset s \subseteq r(X) \setminus X$) has to be added to X, and a further portion of σ has to be explored. Clearly the discovery of the truth of a new sentence of level 0 might contradict some sentence of level 1, that has to be withdrawn from the state of knowledge.

Formally the level 0 answers of the knowledge space are $\{b(t,n), w(t,n) \mid t,n \in \mathbb{N}\}$; their interpretation is defined by the valuation map: $\mathtt{tr}(b(t,n),X) = \mathsf{T}$ if and only if $(t \geq n) \wedge \sigma(t) = b$ (for "black") and $\mathtt{tr}(w(t,n),X) = \mathsf{T}$ if and only if $(t \geq n) \wedge \sigma(t) = w$ (for "white") respectively. The question "is there some black ball of index $u \geq n$?" is represented by introducing the equivalence class $b(\_,n) = \{b(u,n) \mid u \in \mathbb{N}\}$ of $\sim$. The question "is there some white ball of index $u \geq n$?" is represented by introducing the equivalence class $w(\_,n) = \{w(u,n) \mid u \in \mathbb{N}\}$ of $\sim$.

The level 1 answers are $\{W(n), B(n) \mid n \in \mathbb{N}\}$, interpreted as: "all white balls have index $< n$", "all black balls have index $< n$". We set $\mathtt{tr}(W(n),X) = \mathsf{T}$ if and only if $w(\_,n) \cap X = \emptyset$, giving to $W(n)$ the meaning: "as far as we know at X there are no white balls from the n-th one" (i.e., n is greater than all indexes of white balls). Dually, we set $\mathtt{tr}(B(n),X) = \mathsf{T}$ if and only if $b(\_,n) \cap X = \emptyset$: "as far as we know at X there are no black balls from the n-th one" (i.e., n is greater than the indexes of black balls). By construction and the definition of the space topology, $\mathtt{tr}(W(n),\_)$, $\mathtt{tr}(B(n),\_)$ are relative recursive w.r.t. $\mathtt{lk}$, hence continuous, and depend on X. Thus, the truth values of $W(n)$, $B(n)$ are intrinsically unsafe, as they depend on the imperfect knowledge X, instead of being grounded on finitely many verified facts. The answers $W(n)$, $B(n)$ form a second level of knowledge, representing what the algorithm conjectures about the infinity of either black or white balls. Since our goal is to find 10 balls of the same color, the algorithm is not committed to decide which conjecture is actually true, so that even a finite amount of information suffices. On the other hand, since all that matters is which color we can choose, there are just two equivalence classes w.r.t. $\sim$ among the level 1 answers, namely the set $W(\_) = \{W(n) \mid n \in \mathbb{N}\}$ and the set $B(\_) = \{B(n) \mid n \in \mathbb{N}\}$. They can be associated to the questions: "is there some n such that all white (black) balls in σ have index $< n$?"

Given some $\sigma : \mathbb{N} \to \{b,w\}$, the set U of solutions is the set $\mathbb{N}^*$ of finite lists of (indexes of) balls. We define a map $\alpha : \mathbb{S} \to U$ relative recursive in $\mathtt{lk}$, returning a guess for a solution, and a realizer r, relative recursive in $\mathtt{lk}$, whose pre-fix points are knowledge states X such that $\alpha(X)$ is a list of 10 white balls or of 10 black balls. Assume $X \in \mathbb{S}$ is any knowledge state. With two applications of $\mathtt{lk}$, r checks whether there is some $W(n) \in W(\_) \cap X$ and some $B(m) \in B(\_) \cap X$: note that each of these questions can be answered in the positive by a unique answer, $W(n)$ or $B(m)$. We then define $r(X)$ as the least set satisfying to the following conditions.

1. Case $W(\_) \cap X = \emptyset$ *(X believes there is no bound to the indexes of white balls)*. Either X knows of some white ball after 0, that is $w(t_1, 0) \in X$ for some $t_1 \geq 0$, or not. If $w(t_1,0) \in X$, then either X knows of some white ball after $t_1 + 1$, namely $w(t_2, t_1 + 1) \in X$ for some $t_2 \geq t_1 + 1$

or not, and so on for at most 10 steps. In this way we get an increasing list $t_1 < t_2 < \ldots$ of (indexes of) white balls. If we haven't found 10 white balls in this way, then there exists some u such that $w(u, \_) \cap X = \emptyset$. In this case $\mathsf{tr}(W(u), X) = \mathsf{T}$ and we require that $W(u) \in r(X)$. *u is a candidate for a strict upper bound to indexes of white balls.* Note that $W(u) \notin X$, because we assumed $W(\_) \cap X = \emptyset$.

2. Case $B(\_) \cap X = \emptyset$ *(X believes there is no bound to the indexes of black balls).* This is the dual case. By exchanging the roles of black and white balls, either we find 10 black balls, or there is some v such that $b(v, \_) \cap X = \emptyset$. In this case $\mathsf{tr}(B(v), X) = \mathsf{T}$ is true and we require that $B(v) \in r(X)$. *v is a candidate for a strict upper bound to indexes of black balls.* Observe that $B(v) \notin X$, because we assumed $B(\_) \cap X = \emptyset$.

3. Case $W(n) \in W(\_) \cap X$ and $B(m) \in B(\_) \cap X$, for some unique n, m *(X erroneously believes that there is some bound n to the indexes of white balls and some bound m to the indexes of black balls).* If $W(n)$, $B(m)$ are true in X (in particular, if X is sound), then there are no $w(t, n), b(t, m) \in X$, in particular X does not know the color of the ball with index $p = \max(n, m)$ after n, m, that is, $w(p, n) \notin r(X)$ and $b(p, n) \notin r(X)$. According to whether $\sigma(p)$ is b or w, we require $w(p, n) \in r(X)$ or $b(p, n) \in r(X)$ respectively: *r produces a counterexample to $W(n)$ or to $B(m)$.*

By construction we have $r(X) \cap X = \emptyset$, that is, X is a pre-fix point of X if and only if $r(X) = \emptyset$. Observe that the conditions above interpret a classical non-effective proof into a relative effective realizer r, replacing perfect knowledge about infinitely many balls by the process of extending the imperfect knowledge X, that we can assume to be finite. In the third case, when X erroneously believes $W(n)$ and $B(m)$, we have $w(p, n) \in r(X)$ or $b(p, n) \in r(X)$, and adding the answer $w(p, n)$ or $b(p, n)$ falsifies $W(n)$ (we have found some white ball after n) or $B(n)$ (we found some black ball after n) respectively. In this case the extension of X proposed by the realizer r is *non-monotonic*, because at least one belief among $W(n)$ or $B(m)$ must be removed from X, if X has to be sound (that is all answers in X are true w.r.t. X), so that the sequence of states of the learning process is non-monotonic. Actually the method for searching fix points from §2 would inefficiently remove both answers $W(n)$, $B(m)$ from X, and the fact that nonetheless a pre-fix point of r will be reached is a consequence of the non trivial result of Theorem 6.2.

If we find a sound pre-fix point X of r, then $r(X) = \emptyset$ by definition, which implies that we can obtain out of X either 10 white balls, or 10 black balls, or both. The learning algorithm works essentially as the classical proof, modulo the fact that it has imperfect knowledge of infinity of white and black balls, and sometimes it has to improve it. No improvement will ever produce a definite knowledge about the cardinality of white and black balls, yet the imperfect and perfectible knowledge bypasses the problem and suffices to solve the problem.

4. The set of strongly normalizing states is open

The first step toward establishing $\mathtt{SN} = \mathbb{S}$, namely that all states strongly normalize, is to prove that $\mathtt{SN}$ is open in the state topology. To do this we first characterize the reduction relation.

Lemma 4.1. (Reduction)
Let $s \in \mathbb{S}_{\text{fin}}$ be any homogeneous state of level n. Assume $X, Y \in \mathbb{S}$ and $X \twoheadrightarrow^r Y$. Let $m \in \mathbb{N}$.

1. $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$.
2. $X \not\twoheadrightarrow^r X$.
3. If $m \leq n$, then $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$.
4. If $X \upharpoonright_{<m+1} \not\subseteq Y \upharpoonright_{<m+1}$, then $Y \upharpoonright_{=m} = \emptyset$.
5. If $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ then $m \leq n$.
6. If $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ then $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$

Proof:

1. By definition of $X \twoheadrightarrow^{s,r} Y$ we have $s \neq \emptyset$, $X \cap s = \emptyset$ and $Y \upharpoonright_{=n} = X \upharpoonright_{=n} \cup s$. We conclude $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$.
2. By point 1, if $X \twoheadrightarrow^{s,r} Y$ then $X \upharpoonright_{=n} \subset Y \upharpoonright_{=n}$, hence $Y \neq X$.
3. Assume $m \leq n$ in order to prove $X \upharpoonright_{<m+1} \subseteq Y \upharpoonright_{<m+1}$. We reason by cases.
   (a) Let $m < n$. Then $m + 1 \leq n$. By definition of $X \twoheadrightarrow^r Y$ we have $X \upharpoonright_{<n} = Y \upharpoonright_{<n}$, and from $m + 1 \leq n$ we conclude $X \upharpoonright_{<m+1} = (X \upharpoonright_{<n}) \upharpoonright_{<m+1} = (Y \upharpoonright_{<n}) \upharpoonright_{<m+1} = Y \upharpoonright_{<m+1}$.
   (b) Let $m = n$. Then by point 1 above and $X \upharpoonright_{<n} = Y \upharpoonright_{<n}$ we have $X \upharpoonright_{<m+1} = X \upharpoonright_{<n+1} \subset Y \upharpoonright_{<n+1} = Y \upharpoonright_{<m+1}$
4. By point 3, if $X \upharpoonright_{<m+1} \not\subseteq Y \upharpoonright_{<m+1}$, then $m > n$. We deduce $Y \upharpoonright_{=m} \subseteq Y \upharpoonright_{>n} = \emptyset$.
5. Assume $X \upharpoonright_{<m} = Y \upharpoonright_{<m}$ in order to prove that $m \leq n$. If it were $m > n$, we would deduce $X \upharpoonright_{=n} = (X \upharpoonright_{<m}) \upharpoonright_{=n} = (Y \upharpoonright_{<m}) \upharpoonright_{=n} = Y \upharpoonright_{=n}$, contradicting point 1. Thus, $m \leq n$.
6. We apply points 5 and 3 in this order.

□

The next step is to prove that that $\mathtt{SN}$ is open in the state topology. For all $n \in \mathbb{N}$, $n > 0$ we define $\mathtt{SN}_n = \{X \in \mathbb{S} \mid \forall Y \in \mathbb{S}.\ X \not\twoheadrightarrow^r_n Y\}$, that is the set of states from which there is no reduction sequence of length n. The reduction tree $\mathtt{T}(X) = \{Y \in \mathbb{S} \mid \exists n.\ X \twoheadrightarrow^r_n Y\}$ from $X \in \mathbb{S}$ is finitely branching: for any node Y, the number of subsets of $r(Y)$, which is finite, bounds above the number of children of Y. By König's Lemma, $\mathtt{T}(X)$ is finite if and only if all branches of tree (all reduction sequences from X) are finite. Thus, $\mathtt{T}(X)$ is finite if and only if there is some upper bound $n \in \mathbb{N}$ to the reduction sequences out of X. This implies $\mathtt{SN} = \bigcup_{n \in \mathbb{N}} \mathtt{SN}_n$. Therefore to prove that $\mathtt{SN}$ is open it is enough to prove that all $\mathtt{SN}_n$ are open.

Lemma 4.2. ($\mathtt{SN}$ **is open)**
Assume $s \in \mathbb{S}_{\text{fin}}$ is any homogeneous state. Let $I, I_0, I_1 \in \mathcal{P}_{\mathit{fin}}(\mathbb{A})$ be finite sets of answers. Assume $X, Y, X', Y' \in \mathbb{S}$. Let $\mathtt{R}_s : \mathbb{S} \to \mathbb{S}$ be as in Def. 2.3.

1. For all $a \in \mathbb{A}$, $\{X \in \mathbb{S} \mid a \notin X\}$ is open.
2. If (I_0, I_1) is a partition of I, then $\{X \in \mathbb{S} \mid (I \cap X = I_0) \wedge (I \setminus X = I_1)\}$ is open.
3. $\mathtt{R}_s : \mathbb{S} \to \mathbb{S}$ is a continuous map.
4. $\mathtt{SN}_1$ is open.
5. For all $n \in \mathbb{N}$, $\mathtt{SN}_n$ is open.
6. $\mathtt{SN}$ is open.

Proof:

1. Assume $a \in \mathbb{A}$ and $O = \{X \in \mathbb{S} \mid a \notin X\}$. The set O consists of all states including some element of of $[a]_\sim$ different from a, or having empty intersection with $[a]_\sim$. Thus O is the union of all sets $\{X \in \mathbb{S} \mid b \notin X\}$ for $b \in [a]_\sim$ and $b \notin a$, and of the set $\{X \in \mathbb{S} \mid X \cap [a]_\sim = \emptyset\}$. All these sets are basic open of the state topology, therefore O is an open set of the state topology.
2. Assume that (I_0, I_1) is a partition of I and $O = \{X \in \mathbb{S} \mid (I \cap X = I_0) \wedge (I \setminus X = I_1)\}$. Since both $(I \cap X, I \setminus X)$ and (I_0, I_1) are partitions of I, the condition $(I \cap X = I_0) \wedge (I \setminus X = I_1)$ is equivalent to $(I \cap X \supseteq I_0) \wedge (I \setminus X \supseteq I_1)$. Thus, O is equal to the intersection of all sets $\{X \in \mathbb{S} \mid a \in X\}$, for any $a \in I_0$, and of all sets $\{X \in \mathbb{S} \mid a \notin X\}$, for $a \in I_1$. These sets are finitely many because I is finite, and are either sub-basic open, or are open by point 1 above. Thus, O, being a finite intersection of open sets, is open.
3. Assume $s \in \mathbb{S}_{\mathsf{fin}}$ is an homogeneous state of level n. Assume $a \in \mathbb{A}$ and $A_a = \{Z \in \mathbb{S} \mid a \in Z\}$, $B_a = \{Z \in \mathbb{S} \mid Z \cap [a]_\sim = \emptyset\}$ are sub-basic open. We have to prove that $\mathtt{R}_s^{-1}(A_a)$, $\mathtt{R}_s^{-1}(B_a)$ are open sets. We prove this statement by case analysis and definition unfolding.
   (a) If $\mathsf{lev}(a) > n$ then $\mathtt{R}_s^{-1}(A_a) = \emptyset$.
   (b) If $\mathsf{lev}(a) \leq n$ and $a \in s$ then $\mathtt{R}_s^{-1}(A_a) = \mathbb{S}$.
   (c) If $\mathsf{lev}(a) \leq n$ and $a \notin s$ then $\mathtt{R}_s^{-1}(A_a) = A_a$.
   (d) If $\mathsf{lev}(a) > n$ then $\mathtt{R}_s^{-1}(B_a) = \mathbb{S}$.
   (e) If $\mathsf{lev}(a) \leq n$ and $s \cap [a]_\sim \neq \emptyset$ then $\mathtt{R}_s^{-1}(B_a) = \emptyset$.
   (f) If $\mathsf{lev}(a) \leq n$ and $s \cap [a]_\sim = \emptyset$ then $\mathtt{R}_s^{-1}(B_a) = B_a$.
4. $\mathtt{SN}_1$ is the set of irreducible states, equivalently the set of states $X \in \mathbb{S}$ which are pre-fix points of r. Thus, we have to prove that if X is a pre-fix point of r, then there is some open set $X \in O$ such that all $Y \in O$ are pre-fix points of r. Let $O' = r^{-1}(\{r(X)\})$, $O'' = \{Y \in \mathbb{S} \mid (r(X) \cap Y = r(X)) \wedge (r(X) \setminus Y = \emptyset)\}$, and $O = O' \cap O''$. O' is open because $r : \mathbb{S} \to \mathcal{P}_{\mathit{fin}}(\mathbb{A})$ is continuous and $\mathcal{P}_{\mathit{fin}}(\mathbb{A})$ has the discrete topology. O'' is open by point 2, with $I_0 = r(X)$ and $I_1 = \emptyset$. Thus, O is open. By definition, $X \in O' = r^{-1}(\{r(X)\})$ and $X \in O'' = \{Y \in \mathbb{S} \mid Y \cap r(X) = r(X) \wedge r(X) \setminus Y = \emptyset\}$, because $r(X) \subseteq X$. Thus, $X \in O$. For any $Y \in O$ we have by definition of O: $r(Y) = r(X)$ and $r(Y) = r(X) \subseteq Y$, as we wished to show.

5. We prove that $\mathtt{SN}_n$ is open by induction over $n \in \mathbb{N}, n > 0$. The case $n = 1$ is the previous point. Assume $\mathtt{SN}_n$ is open in order to prove that $\mathtt{SN}_{n+1}$ is open. Let $X \in \mathtt{SN}_{n+1}$: we have to prove that there is some open set $X \in O \subseteq \mathtt{SN}_{n+1}$. $r(X) \setminus X$ is finite, therefore there are finitely many homogeneous states $s_1, \ldots, s_k \subseteq r(X) \setminus X$. These states define exactly all reductions from X: $X \twoheadrightarrow^{s_i,r} X_i$, for $i = 1, \ldots, k$. From $X \in \mathtt{SN}_{n+1}$ we deduce $X_i \in \mathtt{SN}_n$ for all $i = 1, \ldots, k$. Let $O_i = \mathtt{R}_{s_i}^{-1}(\mathtt{SN}_n)$: O_i is open by point 3 above, and $X \in O_i$ because $\mathtt{R}_{s_i}(X) \in \mathtt{SN}_n$ by the assumption $X \in \mathtt{SN}_{n+1}$. Let $O' = r^{-1}(\{r(X)\})$, $O'' = \{Y \in \mathbb{S} \mid (r(X) \cap Y = r(X) \cap X) \wedge (r(X) \setminus Y = r(X) \setminus X)\}$. By definition we have $X \in O'$, $X \in O''$. O' is open because r is continuous, and O'' is open by point 2. Let $O = O' \cap O'' \cap O_1 \cap \ldots \cap O_n$: then $X \in O$ and O is open. For all $Y \in O$ we have $r(Y) = r(X)$, and $r(Y) \setminus Y = r(X) \setminus Y = r(X) \setminus X$. Therefore the reductions from Y are exactly in number of k: $Y \twoheadrightarrow^{s_i,r} Y_i$ for all $i = 1, \ldots, k$. We have $Y_i \in \mathtt{SN}_n$ by $O \subseteq O_i = \mathtt{R}_{s_i}^{-1}(\mathtt{SN}_n)$. We conclude that $Y \in \mathtt{SN}_{n+1}$, as desired.

6. $\mathtt{SN}$ is the union of all $\mathtt{SN}_n$, therefore it is the union of open sets and it is open.

□

5. Reduction sequences of transfinite length

The next step is to prove that if there are states in $\mathbb{S} \setminus \mathtt{SN}$, then there are reduction sequences of any transfinite length. From this fact we will derive a contradiction.

We denote the class of ordinals with $\mathtt{ON}$, and ordinals with Greek letters $\alpha, \beta, \gamma, \lambda, \mu, \ldots$. We recall that a limit ordinal is any ordinal $\lambda > 0$ such that for all $\alpha < \lambda$ we have $\alpha + 1 < \lambda$. The first infinite ordinal ω and the first uncountable ordinal ω_1 are limit ordinals. Further ω_1 has the property that any l.u.b. of some countable set I of ordinals all $< \omega_1$ is some $\xi < \omega_1$.

A sequence of length α on $\mathbb{S}$ is any map $\sigma : [0, \alpha[\to \mathbb{S}$. We represent sequences of length α with indexed sets $\sigma = \{X_\beta \mid \beta < \alpha\}$. When α is some limit ordinal λ, the *limit of a sequence* $\{X_\beta \mid \beta < \lambda\}$ is defined as

$$\lim_{\beta \to \lambda} X_\beta = \cup_{\beta < \lambda} \cap_{\beta \leq \gamma < \lambda} X_\gamma.$$

To put this otherwise, $\lim_{\beta \to \lambda} X_\beta$ consists of all answers belonging to the states of $\{X_\beta \mid \beta < \lambda\}$ from some β up. A *limit sequence of length* α is any sequence $\{X_\beta | \beta < \alpha\}$ of length α such that for all limit ordinal $\lambda < \alpha$ we have $X_\lambda = \lim_{\beta \to \lambda} X_\beta$. A *limit reduction sequence of length* α is any limit sequence of length α such that for all $\beta + 1 < \alpha$ we have $X_\beta \twoheadrightarrow^r X_{\beta+1}$. We prove that if $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is some limit reduction sequence of length ω_1 over $\mathbb{S} \setminus \mathtt{SN}$. Then we prove that limit reduction sequence of length ω_1 over $\mathbb{S}$ (and a fortiori over $\mathbb{S} \setminus \mathtt{SN}$) cannot exist. The conclusion is that $\mathbb{S} \setminus \mathtt{SN} = \emptyset$, as desired.

If $X \in \mathbb{S} \setminus \mathtt{SN}$, then there is some infinite reduction sequence

$$X = X_0 \twoheadrightarrow^r X_1 \twoheadrightarrow^r \ldots \twoheadrightarrow^r X_n \ldots$$

out of X. Thus, there is some X_1 such that $X \twoheadrightarrow^r X_1$ and there is some infinite reduction sequence out of X_1, hence $X \twoheadrightarrow^r X_1$ for some $X_1 \in \mathbb{S} \setminus \mathtt{SN}$. By choice axiom, there is some choice map

$$\mathtt{next} : (\mathbb{S} \setminus \mathtt{SN}) \to (\mathbb{S} \setminus \mathtt{SN})$$

such that $X \twoheadrightarrow^r \mathtt{next}(X)$ for all $X \in \mathbb{S} \setminus \mathtt{SN}$. The function $\mathtt{next}$ is the empty map if $\mathtt{SN} = \mathbb{S}$.

Using `next`, from any $X \in \mathbb{S} \setminus \mathtt{SN}$ we can define an infinite reduction sequence $\mathtt{next}^n(X)$ all in $\mathbb{S} \setminus \mathtt{SN}$. We prove that such a reduction can be extended it to reduction sequence in $\mathbb{S} \setminus \mathtt{SN}$ of length ω_1. This is because closed sets in the state topology are closed under limits, and $\mathbb{S} \setminus \mathtt{SN}$ is a closed set.

In this part of the proof we need the notion of "definitely true".

Definition 5.1. (Definitely true)
Assume $\lambda \in \mathtt{ON}$ is limit and $\sigma = \{X_\beta \mid \beta < \lambda\}$ is any sequence of length λ on $\mathbb{S}$.

1. σ satisfies $X_\gamma \subseteq X_{\gamma+1}$ definitely if $\exists \beta < \alpha. \forall \gamma \in [\beta, \lambda[. X_\gamma \subseteq X_{\gamma+1}$.

2. σ is definitely weakly increasing if $\exists \beta < \alpha. \forall \gamma, \delta \in [\beta, \lambda[. (\gamma \leq \delta) \implies X_\gamma \subseteq X_\delta$.

3. σ is definitely constant if $\exists \beta < \alpha. \forall \gamma \in [\beta, \lambda[. X_\beta = X_\gamma$.

The next step is to prove some easy properties of limit reduction sequences.

Lemma 5.2. (Limit Reduction sequences)
Assume $\lambda \in \mathtt{ON}$ is a limit ordinal and $\sigma = \{X_\alpha \mid \alpha < \lambda\}$ is any limit sequence on $\mathbb{S}$ of length λ. Let $L = \lim_{\gamma \to \omega_1} X_\gamma$.

1. If for some $\alpha < \lambda$ and all $\alpha \leq \beta < \lambda$ we have $X_\alpha \subseteq X_\beta$, then $X_\alpha \subseteq L$.

2. If for some $\alpha < \lambda$ and all $\alpha \leq \beta < \lambda$ we have $X_\beta \subseteq X_{\beta+1}$, then σ is weakly increasing from the same α.

3. If σ is definitely increasing and $\lambda = \omega_1$, then σ is definitely stationary.

4. For any $n \in \mathbb{N}$, $\sigma \restriction_{<n} = \{X_\alpha \restriction_{<n} | \alpha < \lambda\}$ is a limit sequence.

Proof:

1. Assume $X_\alpha \subseteq X_\gamma$ for all $\alpha \leq \gamma < \lambda$. Then $X_\alpha \subseteq \bigcap_{\alpha \leq \gamma < \lambda} X_\gamma \subseteq \lim_{\gamma \to \lambda} X_\gamma = L$.

2. Assume $\alpha \leq \alpha' < \lambda$. We prove $X_{\alpha'} \subseteq X_\beta$ by induction on $\alpha' \leq \beta < \lambda$. Assume $\beta = \alpha'$. Then $X_{\alpha'} \subseteq X_{\alpha'}$. Assume $\beta = \gamma + 1 > \gamma \geq \alpha' \geq \alpha$. Then $X_{\alpha'} \subseteq X_\gamma$ by induction hypothesis and $X_\gamma \subseteq X_{\gamma+1}$ by hypothesis, hence $X_{\alpha'} \subseteq X_\beta$. Assume β is limit: then $X_{\alpha'} \subseteq X_\gamma$ for all $\alpha' \leq \gamma < \beta$ by induction hypothesis, therefore $X_{\alpha'} \subseteq X_\beta$ by point 1 applied to the sequence $\{X_\gamma | \gamma < \beta\}$.

3. Assume that σ is definitely increasing from some α and $\lambda = \omega_1$, in order to prove that σ is definitely stationary. For all $a \in L$ we have $a \in X_\gamma$ definitely, therefore there is a first $\alpha \leq \xi_a < \omega_1$ such that $a \in X_{\xi_a} \subseteq X_\gamma$ for all $\gamma \geq \xi_a$. Let ξ be l.u.b. of $\{\xi_a \mid a \in L\} \cup \{\alpha\}$. L is at most countable because $L \subseteq \mathbb{A}$, which is countable, and α and all ξ_a are $< \omega_1$, therefore $\xi < \omega_1$. We proved that there is some $\alpha \leq \xi < \omega_1$ such that for all $\alpha \leq \xi \leq \gamma < \omega_1$ we have $L \subseteq X_\gamma$. From point 1 and $X_\gamma \subseteq X_\delta$ for all $\gamma \leq \delta < \omega_1$ we have $X_\gamma \subseteq L$. We conclude $L = X_\gamma$ for all $\xi \leq \gamma < \omega_1$.

4. Assume $\mu < \lambda$ is limit. Then $X_\mu = \bigcup_{\alpha < \mu} \bigcap_{\alpha \leq \beta < \mu} X_\beta$, hence $X_\mu \restriction_{<n} = \bigcup_{\alpha < \mu} \bigcap_{\alpha \leq \beta < \mu} X_\beta \restriction_{<n}$. Thus, $\sigma \restriction_{<n}$ is a limit sequence.

□

Now we explain how to construct a limit reduction sequence in $\mathbb{S} \setminus \mathtt{SN}$ of length ω_1. The crucial remark is the following: for any answer in a limit reduction sequence, either the answer belongs to the limit of the sequence together with all answers of level less or equal than its own level, or in some subsequent step the answer is deleted together with all answers of the same level (see the first point of the next lemma).

Lemma 5.3. Assume $\lambda \in \mathtt{ON}$ is a limit ordinal and $\sigma = \{X_\beta \mid \beta < \lambda\}$ is any limit reduction sequence of length λ. Let $L = \lim_{\beta \to \lambda} X_\beta \in \mathbb{S}$, and $n \in \mathbb{N}$

1. For all $\alpha < \lambda$ and all $n \in \mathbb{N}$, either $X_\alpha \upharpoonright_{<n+1} \subseteq L$, or there is some $\alpha < \gamma < \lambda$ such that $X_\gamma \upharpoonright_{=m} = \emptyset$
2. L is topologically adherent to $\{X_\beta \mid \beta < \lambda\}$ (that is, any open set including L intersects $\{X_\beta \mid \beta < \lambda\}$.
3. If $C \subseteq \mathbb{S}$ is closed and $\{X_\beta \mid \beta < \lambda\} \subseteq C$ then $L \in C$
4. $\mathbb{S} \setminus \mathtt{SN}$ is closed
5. If $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is some length ω_1 limit reduction sequence in $\mathbb{S} \setminus \mathtt{SN}$.

Proof:

1. Consider the sequence $\tau = \{X_\beta \upharpoonright_{<n+1} | \beta < \lambda\}$: this is a limit sequence by Lemma 5.2.4. If $X_\beta \upharpoonright_{<n+1} \subseteq X_{\beta+1} \upharpoonright_{<n+1}$ for all $\alpha \leq \beta < \lambda$, then τ is weakly increasing from α by Lemma 5.2.2. In this case $X_\alpha \upharpoonright_{<n+1} \subseteq X_\gamma \upharpoonright_{<n+1} \subseteq X_\gamma$ for all $\alpha \leq \gamma < \lambda$, therefore $X_\alpha \upharpoonright_{<n+1} \subseteq L$ by Lemma 5.2.1. Assume instead that $X_\beta \upharpoonright_{<n+1} \not\subseteq X_{\beta+1} \upharpoonright_{<n+1}$ for some $\alpha \leq \beta < \lambda$. Then by Lemma 4.1.4 we have $X_{\beta+1} \upharpoonright_{=n} = \emptyset$.

2. Fix $\alpha < \lambda$, and assume O is any sub-basic open and $L \in O$, in order to prove that $X_\beta \in O$ for some $\alpha \leq \beta < \lambda$. For some $a \in \mathbb{A}$, either $O = A_a = \{X \in \mathbb{S} | a \in X\}$, or $O = B_a = \{X \in \mathbb{S} | X \cap [a]_\sim = \emptyset\}$. We reason by cases.

   (a) If $O = A_a$ we have $a \in L$. By definition of L, for some $\alpha < \lambda$ and all $\alpha \leq \beta < \lambda$ we have $a \in X_\beta$. In particular, $a \in X_\alpha$, hence $X_\alpha \in O$.

   (b) If $O = B_a$ we have $L \cap [a]_\sim = \emptyset$. Assume $n = \mathsf{lev}(a)$: by point 1 above there is some $\alpha \leq \beta < \lambda$ such that either $X_\beta \upharpoonright_{<n+1} \subseteq L$ or $X_\beta \upharpoonright_{=n} = \emptyset$. In both cases we have $X_\beta \upharpoonright_{=n} \subseteq L \upharpoonright_{=n}$, either because $X_\beta \upharpoonright_{=n} = (X_\beta \upharpoonright_{<n+1}) \upharpoonright_{=n} \subseteq L \upharpoonright_{=n}$, or because $X_\beta \upharpoonright_{=n} = \emptyset \subseteq L \upharpoonright_{=n}$. We deduce $X_\beta \cap [a]_\sim = (X_\beta \upharpoonright_{=n}) \cap [a]_\sim \subseteq (L \upharpoonright_{=n}) \cap [a]_\sim \subseteq L \cap [a]_\sim = \emptyset$. Thus, $X_\beta \in B_a$.

3. Assume $C \subseteq \mathbb{S}$ is closed and $\{X_\beta \mid \beta < \lambda\} \subseteq C$ in order to prove that $L \in C$. Assume for contradiction that $L \notin C$. Then $L \in \mathbb{S} \setminus C$, which is open. By point 2 we have $X_\alpha \in \mathbb{S} \setminus C$ for some $\alpha < \lambda$, contradicting $X_\alpha \in C$.

4. $\mathbb{S} \setminus \mathtt{SN}$ is closed because $\mathtt{SN}$ is open by Lemma 4.2.6.

5. From any $X \in \mathbb{S} \setminus \mathtt{SN}$ we may define a limit reduction sequence of length ω_1 (and in fact of *any* length). We set $X_0 = X$, $X_{\alpha+1} = \mathtt{next}(X_\alpha)$ for all $\alpha < \omega_1$ and $X_\lambda = \lim_{\beta \to \lambda} X_\beta$ for all limit $\lambda < \omega_1$. We check that the definition is correct. By assumption $X_0 = X \in \mathbb{S} \setminus \mathtt{SN}$. Assume $\alpha < \omega_1$ and $X_\alpha \in \mathbb{S} \setminus \mathtt{SN}$, then $X_{\alpha+1} = \mathtt{next}(X_\alpha) \in \mathbb{S} \setminus \mathtt{SN}$. Assume $\lambda < \omega_1$ is limit and $\{X_\beta | \beta < \lambda\} \subseteq \mathbb{S} \setminus \mathtt{SN}$. Since $\mathbb{S} \setminus \mathtt{SN}$ is closed by point 4, then by point 3 above we have $X_\lambda = \lim_{\beta \to \lambda} X_\beta \in \mathbb{S} \setminus \mathtt{SN}$.

□

6. Termination of the algorithm searching pre-fix points

In the previous section we proved that if $\mathbb{S} \setminus \mathtt{SN} \neq \emptyset$, then there is a limit reduction sequence of length ω_1. In this section we prove that no limit reduction sequence of length ω_1 may exists, and we conclude that $\mathbb{S} \setminus \mathtt{SN} = \emptyset$, as desired. We first prove that limit reduction sequences of length ω_1 are definitely stationary.

Lemma 6.1. (Stationarity)
Assume $\sigma = \{X_\alpha \mid \alpha < \omega_1\}$ is any sequence on $\mathbb{S}$. Let $n \in \mathbb{N}$.

1. For any limit reduction sequence $\{X_\beta \mid \beta < \omega_1\}$ of length ω_1 on $\mathbb{S}$, the sequence $\{X_\beta \upharpoonright_{<n} | \beta < \omega_1\}$ is definitely stationary.

2. Any limit reduction sequence $\{X_\beta \mid \beta < \omega_1\}$ of length ω_1 on $\mathbb{S}$ is definitely stationary.

Proof:

1. We argue by induction over $n \in \mathbb{N}$. Assume $n = 0$: then $X_\beta \upharpoonright_{<n} = \emptyset$ is definitely stationary. Assume $X_\beta \upharpoonright_{<n}$ is definitely stationary, in order to prove that $X_\beta \upharpoonright_{<n+1}$ is definitely stationary. If $X_\beta \upharpoonright_{<n} = X_{\beta+1} \upharpoonright_{<n}$ then $X_\beta \upharpoonright_{<n+1} \subseteq X_{\beta+1} \upharpoonright_{<n}$ by Lemma 4.1.6, hence we have that $X_\beta \upharpoonright_{<n+1} \subseteq X_{\beta+1} \upharpoonright_{<n}$ definitely. By Lemma 5.2.4 $X_\beta \upharpoonright_{<n+1}$ is a limit sequence, and by Lemma 5.2.2 it is weakly increasing. It has length ω_1, therefore by Lemma 5.2.3 it is definitely stationary.

2. By the previous point, for all $n \in \mathbb{N}$ there is a first $\alpha_n < \omega_1$ such that $X_\beta \upharpoonright_{<n}$ is stationary from α_n. Let $\alpha < \omega_1$ by the l.u.b. of $\{\alpha_n | n \in \mathbb{N}\}$: then for all $n \in \mathbb{N}$, $X_\beta \upharpoonright_{<n}$ is stationary from α. Thus, X_β is stationary from α.

□

Now the strong termination result for the reduction relation $\twoheadrightarrow^r$ easily follows.

Theorem 6.2. (Pre-fix point Theorem)
For all states $X \in \mathbb{S}$, all reduction sequences $X = X_0 \twoheadrightarrow^r X_1 \twoheadrightarrow^r X_2 \twoheadrightarrow^r \ldots$ out of X are finite.

Proof:
Assume there is some $X \in \mathbb{S} \setminus \mathtt{SN}$. By Lemma 5.3.5 there is some limit reduction sequence $\{X_\beta \mid \beta < \omega_1\} \subseteq (\mathbb{S} \setminus \mathtt{SN})$ out of X of length ω_1. By Lemma 6.1.2, $\{X_\beta \mid \beta < \omega_1\}$ is definitely stationary, therefore for some $\alpha < \omega_1$ we have $X_{\alpha+1} = X_\alpha$, hence $X_\alpha \twoheadrightarrow^r X_{\alpha+1} = X_\alpha$, against Lemma 4.1.2. □

7. Related works and conclusions

In this section we stress the most relevant differences of the present paper w.r.t. previous work by the authors themselves and by others. The setting of knowledge spaces and the way we understand states of knowledge are clearly related to the concept of Herbrand universes. Indeed, one way to look at knowledge spaces is as an abstract description of the Herbrand Theorem, on which e.g. [10, 8] are based. The key difference between our work and these papers is that we interpret Herbrand Theorem as a guideline to design and check programs which learn, a goal that apparently has not been considered before.

We now compare definitions and results from the present paper to previous work of ours. The difference w.r.t. [3] and [5] is that in those papers we only considered monotonic learning. In [2] also the case of non-monotonic learning is considered, though only deterministic learning processes are treated. In [7], which is the full version of [6], we propose a deterministic algorithm to compute a (finite) sound pre-fix point of any effective realizer; however we treated just the case in which the maximum level of answers is 2, while here we have a termination proof of a non-deterministic algorithm working on states with answers of arbitrary level $< \omega$.

We stress that proving the existence of a pre-fix point for non-deterministic realizers is no minor trick. Indeed, if the output $r(X)$ of a realizer may include more than one answer, then the convergence result also holds for any deterministic $r' : \mathbb{S} \to \mathcal{P}_{\mathit{fin}}(\mathbb{A})$, *even if* r' *is not continuous*, provided there is some continuous $r : \mathbb{S} \to \mathcal{P}_{\mathit{fin}}(\mathbb{A})$ such that $r'(X) \subseteq r(X)$ for all $X \in \mathbb{S}$. This simple remark shows that the result for the non-deterministic case is much stronger than for the deterministic one.

In [11] Mints considered the ω-level version of the problem. In our terminology, he introduced a reduction relation adding one answer at the time, which is non-deterministic because at each step many choices are possible, and proved a *weak* normalization result: there is a reduction sequence from the empty state to some sound irreducible state. However, in [11] there is no normalizing reduction strategy, and we suspect that the strong normalization result would fail in that setting.

In conclusion we have presented a new result that we consider a step toward a realistic use of non constructive proofs as a guideline to design and check learning programs. Improvements are certainly possible, such as for example a more sophisticated way of representing logical dependencies than just using the level. The aim is to find an algorithm removing the minimum amount of answers from a state when adding new ones, hence resulting into a faster computation.

Acknowledgments

The authors wish to thank the anonymous referees, whose comments have been of help for improving the final version of the paper.

References

[1] Akama, Y., Berardi, S., Hayashi, S., Kohlenbach, U.: An Arithmetical Hierarchy of the Law of Excluded Middle and Related Principles. In Proc. of LICS'04, IEEE Computer Society, 2004, pp. 192-201.

[2] F. Aschieri. *Learning, Realizability and Games in Classical Arithmetic*, Ph. d. thesis, University of Turin and Queen Mary University of London, 2010. http://www.di.unito.it/~stefano/Aschieri-PhDThesis.pdf

[3] F. Aschieri, S. Berardi, *Interactive Learning-Based Realizability Interpretation for Heyting Arithmetic with* $\mathtt{EM}_1$, Proceedings of TLCA 2009, SLNCS, vol. 5608, 2009, pp. 20-34. doi:10.1007/978-3-642-02273-9_4

[4] S. Berardi, U. de'Liguoro, *Toward the interpretation of non-constructive reasoning as non-monotonic learning*, Information and Computation, vol. 207, 1, 2009, pp. 63-81. doi:10.1016/j.ic.2008.10.003

[5] S. Berardi, U. de'Liguoro, *Interactive realizers. A new approach to program extraction from non constructive proofs*, ACM Transactions on Computational Logic, vol. 13 n. 2, 2012. doi:10.1145/2159531.2159533

[6] S. Berardi, U. de'Liguoro, *Knowledge Spaces and Interactive Realizers*. Proc. of CSL 2012, LIPICs vol. 16, 2012, pp. 77-91. doi:10.4230/LIPIcs.CSL.2012.77

[7] S. Berardi, U. de'Liguoro, *Knowledge Spaces and the Completeness of Learning Strategies*, Logical Methods in Computer Science, 10, 1, 2014, pp. 1-23. http://arxiv.org/pdf/1401.0885

[8] S. R. Buss, *On Herbrand's Theorem*, in Logic and Computational Complexity, International Workshop LCC '94, 1994 pp.. 195-209. http://dx.doi.org/10.1007/3-540-60178-3_85

[9] T. Coquand. *A semantics of evidence for classical arithmetic.* J. of Symbolic Logic, 60, 1995, pp. 325-337. http://projecteuclid.org/euclid.jsl/1183744693

[10] D. Miller, *Expansion Tree Proofs and Their Conversion to Natural Deduction Proofs*, in 7th International Conference on Automated Deduction, SLNCS 170, 1984, pp 375-393. http://link.springer.com/chapter/10.1007/978-0-387-34768-4_22

[11] G. Mints. *Non-Deterministic Epsilon Substitution Method for PA and ID_1*, Logic, Construction, Computation, Ontos-Verlag Series in Mathematical Logic, Berger et al. editors, 2012, pp. 325-342. http://www.degruyter.com/view/product/208943

[12] G. Stolzenberg *Seeing Constructions*, Conference given at Sophia Antipolis, Antibes, France, May 9, 1989.