Annals of Pure and Applied Logic 164 (2013) 591–617

Contents lists available at SciVerse ScienceDirect

Annals of Pure and Applied Logic

www.elsevier.com/locate/apal

Learning based realizability for HA + EM1 and 1-Backtracking games: Soundness and completeness

Federico Aschieri [a,b,*]

[a] *Dipartimento di Informatica, Università di Torino, Italy*
[b] *School of Electronic Engineering and Computer Science, Queen Mary, University of London, UK*

ARTICLE INFO

Article history:
Available online 17 May 2012

MSC:
03F03
03F30
03F55

Keywords:
Classical realizability
Backtracking games
Learning
Classical logic

ABSTRACT

We prove a soundness and completeness result for Aschieri and Berardi's learning based realizability for Heyting Arithmetic plus Excluded Middle over semi-decidable statements with respect to 1-Backtracking Coquand game semantics. First, we prove that learning based realizability is sound with respect to 1-Backtracking Coquand game semantics. In particular, any realizer of an implication-and-negation-free arithmetical formula embodies a winning recursive strategy for the 1-Backtracking version of Tarski games. We also give examples of realizers and winning strategy extraction for some classical proofs. Secondly, we extend our notion of realizability to a total recursive learning based realizability and show that the notion is complete with respect to 1-Backtracking Coquand game semantics.

1. Introduction

In this paper we show that learning based realizability (see Aschieri and Berardi [4]) relates to 1-Backtracking Tarski games as intuitionistic realizability (see Kleene [15]) relates to Tarski games, when one considers implication-and-negation-free formulas. The relationship we refer to is between realizability on one hand, and existence of winning strategies on the other. In particular, it is folklore that a negation-and-implication-free arithmetical formula is Kleene realizable if and only if Eloise has a recursive winning strategy in the associated Tarski game. We show as well that an implication-and-negation-free arithmetical formula is "learning realizable" if and only if Eloise has a recursive winning strategy in the associated 1-Backtracking Tarski game.

It is well known that Tarski games (which were actually introduced by Hintikka, see [14] and Definition 12) are just a simple way of rephrasing the concept of classical truth in terms of a game between two players – the first one, Eloise, trying to show the truth of a formula, the second, Abelard, its falsehood – and that a Kleene realizer gives a recursive winning strategy to the first player. The result is quite expected: since a realizer gives a way of computing all the information about the truth of a formula, the player trying to prove the truth of that formula has a recursive winning strategy. However, not at all *any* classically provable arithmetical formula allows a winning recursive strategy for that player; otherwise, the decidability of the Halting problem would follow.

In [7], Coquand introduced a new game semantics for Peano Arithmetic, centered on the concept of "Backtracking Tarski game": a special Tarski game in which players have the additional possibility of correcting their moves and backtracking to a previous position of the game anytime they wish. Coquand then showed that for any provable negation-and-implication-free arithmetical formula A, Eloise has a *recursive* winning strategy in the Backtracking Tarski game associated to A. Remarkably,

* Correspondence to: Dipartimento di Informatica, Università di Torino, Italy.
E-mail address: aschieri@di.unito.it.

0168-0072/$ – see front matter
http://dx.doi.org/10.1016/j.apal.2012.05.002

a proof in Peano Arithmetic thus hides a non-trivial computational content that can be described as a recursive strategy that produces witnesses in classical Arithmetic by interaction and learning.

In the first part of the paper, we show that learning based realizers have direct interpretation as recursive winning strategies in 1-Backtracking Tarski games (which are a particular case of Coquand games: see Berardi et al. [5] and Definition 11 below). The result was desired, because interactive learning based realizers, by design, are similar to strategies in games with backtracking: they improve their computational ability by learning from interaction and counterexamples in a convergent way; eventually, they gather enough information about the truth of a formula to win its associated game.

An interesting but incomplete step towards our result was the Hayashi limit realizability [12,13]. Indeed, a realizer in the sense of Hayashi represents a recursive winning strategy in 1-Backtracking Tarski games. However, from the computational point of view, Hayashi realizers do not relate to 1-Backtracking Tarski games in a significant way: Hayashi winning strategies work by exhaustive search and, actually, do not learn from the game and from the *interaction* with the other player. As a result of this issue, constructive upper bounds on the length of games cannot be obtained, whereas using our realizability this is possible. For example, in the case of the 1-Backtracking Tarski game for the formula $\exists x^{\mathbb{N}}\forall y^{\mathbb{N}} f(x) \leqslant f(y)$, the Hayashi realizer checks all the natural numbers to be sure that an n such that $\forall y^{\mathbb{N}} f(n) \leqslant f(y)$ is eventually found. On the contrary, our realizer yields a strategy for Eloise which bounds the number of backtrackings by $f(0)$, as shown in this paper; moreover, what the strategy learns is determined by the interaction with the other player. In this case, the Hayashi strategy is the same one suggested by the classical *truth* of the formula, whereas ours is the constructive strategy suggested by its classical *proof*. The same consideration is also true in general, and it is due to the fact that the excluded middle is not interpreted constructively by Hayashi: in order to prove that it is realizable, he needs classical reasoning.

Since learning based realizers are extracted from proofs in $\mathsf{HA} + \mathsf{EM}_1$ (Heyting Arithmetic with excluded middle over existential sentences, see [4]), one also has an interpretation of classical proofs as strategies with 1-Backtracking. Moreover, studying learning based realizers in terms of 1-Backtracking games also sheds light on their behavior and offers an interesting case study in program extraction and interpretation in classical Arithmetic.

In the second part of the paper, we extend the class of learning based realizers from a classical oracle-equipped version of Gödel's system T to a classical oracle-equipped version of $\mathcal{PCF}$ and define a more general "total recursive learning based realizability". This step is analogous to the (conceptual, rather than chronological) step leading from Kreisel modified realizability [16] to Kleene realizability: one extends the computational power of realizers. We then prove a completeness theorem: for every implication-and-negation-free arithmetical formula A, if Eloise has recursive winning strategy in the 1-Backtracking Tarski game associated to A, then A is also realizable.

The *plan of the paper* is the following. In Section 2, we recall the definitions and results from Aschieri and Berardi [4] that we shall need in the following. In Section 3, we prove our first main theorem: a realizer of an arithmetical formula embodies a winning strategy in its associated 1-Backtracking Tarski game. In Section 4, we extract realizers from two classical proofs and study their behavior as learning strategies. In Section 5, we define an extension of learning based realizability of [4] and in Section 6 prove its completeness with respect to 1-Backtracking Tarski games.

2. Learning-based realizability for $\mathsf{HA} + \mathsf{EM}_1$

The whole content of this section is based on Aschieri and Berardi [4], where the reader may also find full motivations and proofs. We recall here not only all the definitions and results we need in the rest of the paper, but we also provide detailed comments, because they are essential to understand the program extraction techniques that we will employ in the examples of Section 4 and in the completeness proof of Section 6.

Learning based realizers are written in an extension of Gödel's system T. For a complete definition of T we refer to Girard [9]. T is simply typed λ-calculus, with atomic types $\mathtt{N}$ (representing the set $\mathbb{N}$ of natural numbers) and $\mathtt{Bool}$ (representing the set $\mathbb{B} = \{\text{True}, \text{False}\}$ of booleans), product types $T \times U$ and arrows types $T \to U$, constants $0 : \mathtt{N}$, $\mathsf{S} : \mathtt{N} \to \mathtt{N}$, $\mathtt{True}, \mathtt{False} : \mathtt{Bool}$, pairs $\langle .,. \rangle$, projections π_0, π_1, conditional if_T and primitive recursion R_T in all types, and the usual reduction rules $(\beta), (\pi), (\mathsf{if}), (\mathsf{R})$ for λ, $\langle .,. \rangle, \mathsf{if}_T, \mathsf{R}_T$. From now on, if t, u are terms of T we denote with $t = u$ provable intensional equality in T. If $k \in \mathbb{N}$, the numeral denoting k is the closed normal term $\mathsf{S}^k(0)$ of type $\mathtt{N}$. Terms of the form $\mathsf{if}_T t_1 t_2 t_3$ will be written in the more legible form $\mathsf{if}\ t_1\ \mathsf{then}\ t_2\ \mathsf{else}\ t_3$. All closed normal terms of T of type $\mathtt{N}$ are numerals. Any closed normal term of type $\mathtt{Bool}$ in T is $\mathtt{True}$ or $\mathtt{False}$.

We introduce a notation for ternary projections: if $T = A \times (B \times C)$, with p_0, p_1, p_2 we respectively denote the terms π_0, $\lambda x : T.\pi_0(\pi_1(x))$, $\lambda x : T.\pi_1(\pi_1(x))$. If $u = \langle u_0, \langle u_1, u_2 \rangle\rangle : T$, then $\mathsf{p}_i u = u_i$ in T for $i = 0, 1, 2$. We abbreviate $\langle u_0, \langle u_1, u_2 \rangle\rangle : T$ with $\langle u_0, u_1, u_2 \rangle : T$.

Learning based realizability [4] is an extension of Kreisel modified realizability [16] to $\mathsf{HA} + \mathsf{EM}_1$ where

$$\mathsf{EM}_1 := \forall x^{\mathbb{N}}.\ \exists y^{\mathbb{N}} Pxy \vee \forall y^{\mathbb{N}} \neg Pxy$$

with Pxy decidable. In a few words, it is a way of making oracle computations more effective, through the use of approximations of oracle values and learning of new values by counterexamples. A learning based realizer is in the first place a term of Gödel's $\mathcal{T}_{\text{Class}}$, which is Gödel's system T plus oracles $\mathsf{X}_P : \mathtt{N} \to \mathtt{Bool}$, $\Phi_P : \mathtt{N} \to \mathtt{N}$ of the same Turing degree of an oracle for the Halting problem. Of course, if a realizer was only this, it would be ineffective and so useless. Therefore, learning based realizers are computed with respect to *approximations* of the oracles X_P, Φ_P and thus effectiveness is recovered. Since approximations may be sometimes inadequate, results of computations may be wrong. But a learning based