The simplest computability-logic completeness proof

Giorgi Japaridze*

To Dick de Jongh, my cherished mentor and friend; the man whom I admire not only for being a prominent logician, but also for being an outstanding human with an incredibly pure soul...

Abstract

The paper presents a soundness and completeness proof for a propositional system that axiomatizes one of the most basic fragments of computability logic — the approach introduced by the author in [Annals of Pure and Applied Logic 123 (2003), pp. 1-99]. This proof is significantly simpler than the proofs of any other similar results generated so far within the framework of computability logic. This is due to the fact that completeness here is understood as completeness with respect to *uniform validity* rather than the weaker concept of *validity*.

1 Introduction and preliminaries

The recently initiated approach called computability logic is a logical theory of interactive computation. It understands computational problems as games played by a machine against the environment, and uses logical formalism to describe the valid principles of computability. The foundational paper [1] on computability logic, which introduced the subject only semantically, set the goal of exploring possible axiomatizations of the new logic. A significant progress has been made since then towards this goal: several nontrivial fragments of the logic have been successfully axiomatized, and more — incrementally strong — results in the same style are apparently still to come in the near future. The corresponding soundness and completeness proofs tend to be very long (so far the longest one taking a few dozen pages) and hard. The present paper is also devoted to a soundness/completeness proof for a fragment of computability logic. Technically its main theorem follows from some already known, significantly stronger results. However, the virtue of this contribution is that the completeness proof presented

*This material is based upon work supported by the National Science Foundation under Grant No. 0208816

in it is the only short and easy-to-comprehend one in the pipeline of similar results. Hence, familiarity with the present proof might be a good or even necessary starting point for someone (especially for a student) who wants to deeper explore computability logic in a step-by-step fashion.

This article is not self-contained and can only be understood by a reader familiar with [1], as it fully relies on the terminology and notation established in [1]. The latter contains and index which can be used to find any terms or notation not explained in the present paper.

Henceforth by "formula" we always mean a propositional, elementary-base, finite-depth formula of the universal language. For any such formula F, every interpretation * is F-admissible, so we can safely omit the word "admissible" before "interpretation". Note also that F^* is unistructural, which means that the valuation parameter can be safely ignored when we talk about what runs of F^* are legal.

In what follows, the letters E, F, G, H, L will be exclusively used as a metavariable for formulas, α, β for moves, * for interpretations and e for valuations.

We will be using the notation

$$\|F\|$$

for the elementarization of F. For a given valuation e and interpretation *, the *classical model induced by e and **, denoted

$$\mathcal{CLM}^*_e,$$

is the true/false assignment for atoms where an atom p is true iff the predicate p^* (which, note, may depend on some "hidden" variables) is true at valuation e; this assignment extends to all elementary formulas in the standard classical way.

The following lemma can be verified by straightforward induction on the complexity of F:

Lemma 1.1 $\mathbf{Wn}^{F^*}_e\langle\rangle = \top$ *iff* $\|F\|$ *is true in* $\mathcal{CLM}^*_e$ *(all F, e, *).*

Now we define a function that, for a formula F and a surface occurrence O in F, returns a string α called the *F-specification of O*, which is said to *F-specify O*. In particular:

- The occurrence of F in itself is F-specified by ϵ (the empty string).
- If $F = \neg G$, then an occurrence that happens to be in G is F-specified by the same string that G-specifies that occurrence.
- If F is $G_1 \wedge \ldots \wedge G_n$, $G_1 \vee \ldots \vee G_n$ or $G_1 \rightarrow G_2$, then an occurrence that happens to be in G_i is F-specified by $i.\alpha$, where α is the G_i-specification of that occurrence.

Example: The second occurrence of $p \sqcup q$ in $F = r \vee (p \sqcup q) \vee \neg(p \rightarrow (r \wedge (p \sqcup q)))$ is F-specified by the string "3.2.2.".

The following lemma can be easily verified by induction on the complexity of F, the routine details of which we omit:

Lemma 1.2 *For all F, α and *:*

a) $\langle \bot\alpha \rangle \in \mathbf{Lr}^{F^*}$ iff $\alpha = \beta i$, where β is the F-specification of a positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) and $i \in \{1, \ldots, n\}$. In this case $\langle \bot\alpha \rangle F^* = H^*$, where H is the result of substituting in F the above occurrence by G_i.

b) $\langle \top\alpha \rangle \in \mathbf{Lr}^{F^*}$ iff $\alpha = \beta i$, where β is the F-specification of a negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcap \ldots \sqcap G_n$) and $i \in \{1, \ldots, n\}$. In this case $\langle \top\alpha \rangle F^* = H^*$, where H is the result of substituting in F the above occurrence by G_i.

Logic **CL1**, which we call the propositional, finite-depth, elementary-base fragment of computability logic and which is a natural syntactic fragment of (the more expressive) logic **FD** introduced in [1], is given by the following two rules:

Rule (a): $\vec{H} \vdash F$, where F is stable and $\vec{H}$ is the smallest set of formulas satisfying the following condition: If F has a positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$), then, for each $i \in \{1, \ldots, n\}$, $\vec{H}$ contains the result of replacing this occurrence in F by G_i.

Rule (b): $H \vdash F$, where H is the result of replacing in F a negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) by G_i for some $i \in \{1, \ldots, n\}$.

Axioms are not explicitly stated, but note that the set of premises of Rule **(a)** may be empty, in which case the conclusion of that rule acts as an axiom. It is a very easy job to verify that a formula (in our present sense) is provable in **CL1** iff it is provable in **FD**.

The rest of this paper is devoted to a proof of the soundness and completeness of **CL1** with respect to uniform validity.

2 Soundness

Theorem 2.1 *If* **CL1** $\vdash F$, *then F is uniformly valid (any F).*

PROOF Assume **CL1** $\vdash F$. Let us fix a particular **CL1**-proof of F. We will be referring to at as "the proof", and referring to formulas occurring in the proof as "proof formulas". We construct the EPM $\mathcal{M}$ which works as follows. At the beginning, this machine creates a record E to hold proof formulas, initializes it to F, and then follows the following interactive algorithm:

> **Procedure** LOOP: Act depending on which of the two rules was used to derive E in the proof:
>
> *Case of Rule* **(a)***:* Keep granting permission until the adversary makes a move αi, where α E-specifies a positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) and $i \in \{1, \ldots, n\}$. Let H be the result of substituting the above occurrence by G_i in E. Then update E to H, and repeat LOOP.
>
> *Case of Rule* **(b)***:* Let H be the (only) premise of E in the proof. H is the result of substituting, in E, a certain negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) by G_i for some $i \in \{1, \ldots, n\}$. Let α be the E-specification of that occurrence. Then make the move αi, update E to H, and repeat LOOP.

Pick an arbitrary valuation e and an arbitrary e-computation branch B of $\mathcal{M}$. Let Γ be the run spelled by B. Our goal is to show that B is fair and, for every *, $\mathbf{Wn}_e^{F^*}\langle\Gamma\rangle = \top$, so that $\mathcal{M} \models F^*$. Consider the work of $\mathcal{M}$ along B. Let E_i denote the value of the record E at the beginning of the ith iteration of LOOP. Evidently E_{i+1} is always one of the premises of E_i in the proof, so that LOOP is iterated only a finite number of times. Fix l as the number of iterations of LOOP, and let $L = E_l$. The lth iteration deals with the case of Rule **(a)**, for otherwise there would be a next iteration. This guarantees that $\mathcal{M}$ will grant permission infinitely many times during that iteration, so that branch B is indeed fair. And the fact that L is derived by Rule **(a)** implies that

$$L \textit{ is stable.} \tag{1}$$

For each i with $1 \leq i \leq l$, let Θ_i be the sequence of the (correspondingly labeled) moves made by the players by the beginning of the ith iteration of LOOP.

$$\begin{array}{l}\textit{For all } ^* \textit{ and } 1 \leq i \leq l, \textit{ either } \Theta_i \textit{ is a } \bot\textit{-illegal position of } F^*, \\ \textit{or } \Theta_i \in \mathbf{Lr}^{F^*} \textit{ and } \langle\Theta_i\rangle F^* = E_i^*.\end{array} \tag{2}$$

This statement can be proven by induction on i. The basis case with $i = 1$ is trivial. Now consider an arbitrary i with $1 \leq i < l$. If Θ_i is a $\bot$-illegal position of F^*, then so is Θ_{i+1} as the latter is an extension of the former. Suppose now Θ_i is not a $\bot$-illegal position of F^*. Then, by the induction hypothesis, $\Theta_i \in \mathbf{Lr}^{F^*}$ and $\langle\Theta_i\rangle F^* = E_i^*$. If the ith iteration of LOOP deals with the case of Rule **(b)**, then exactly one move α is made during that iteration, and this move is by the machine, so that $\Theta_{i+1} = \langle\Theta_i, \top\alpha\rangle$. In view of Lemma 1.2(b), $\langle\top\alpha\rangle \in \mathbf{Lr}^{E_i^*}$ and $\langle\top\alpha\rangle E_i^* = E_{i+1}^*$. With the equalities $\Theta_{i+1} = \langle\Theta_i, \top\alpha\rangle$ and $E_i^* = \langle\Theta_i\rangle F^*$ in mind, the former then implies $\Theta_{i+1} \in \mathbf{Lr}^{F^*}$ and the latter implies $\langle\Theta_{i+1}\rangle F^* = E^*_{i+1}$. Suppose now the ith iteration of LOOP deals with the case of Rule **(a)**. Then the machine does not make a move; if its adversary makes a move α that is not $\bot$'s legal initial move for $E^*_i = \langle\Theta_i\rangle F^*$, then $\langle\Theta_i, \bot\alpha\rangle$ is a $\bot$-illegal position of F^* and so will be Θ_{i+1} as it will contain $\langle\Theta_i, \bot\alpha\rangle$ as an initial segment; otherwise, if $\langle\bot\alpha\rangle \in \mathbf{Lr}^{E^*_i}$, arguing as in the previous case (only using 1.1(a) instead of 1.1(b)), we can again conclude that $\Theta_{i+1} \in \mathbf{Lr}^{F^*}$ and $\langle\Theta_{i+1}\rangle F^* = E^*_{i+1}$.

$$\textit{For all } {}^*\textit{, either } \Gamma \textit{ is a } \bot\textit{-illegal run of } F^*\textit{, or } \mathbf{Wn}^{F^*}_e\langle\Gamma\rangle = \mathbf{Wn}^{L^*}_e\langle\rangle. \qquad (3)$$

To prove this statement, suppose Γ is not a $\bot$-illegal run of F^*. Θ_l is an initial segment of Γ, so Θ_l is not $\bot$-illegal, either. Then, according to (2), Θ_l is a legal position of F^* and $\langle\Theta_l\rangle F^* = L^*$, which implies that $\mathbf{Wn}^{F^*}_e\langle\Theta_l\rangle = \mathbf{Wn}^{L^*}_e\langle\rangle$. So, it would now be sufficient to show that $\Theta_l = \Gamma$. But indeed, as the lth iteration of LOOP deals with the case of Rule **(a)**, during that iteration $\top$ does not move; if $\bot$ makes a move α, it is clear from Lemma 1.2(a) that α should be an illegal initial move for L^* (otherwise there would be a next iteration), which, taking into account that $L^* = \langle\Theta_l\rangle F^*$, implies that $\langle\Theta_l, \bot\alpha\rangle$ is a $\bot$-illegal position of F^*; but $\langle\Theta_l, \bot\alpha\rangle$ must be an initial segment of Γ, so that Γ is a $\bot$-illegal run of F^*, contrary to our assumption. Thus, $\bot$ does not make any moves during the last iteration of LOOP either, and hence $\Gamma = \Theta_l$.

To finish our proof of the theorem, consider an arbitrary interpretation *. We want to show that $\mathbf{Wn}^{F^*}_e\langle\Gamma\rangle = \top$. If Γ is a $\bot$-illegal run of F^*, we are done. Suppose now Γ it is not an $\bot$-illegal run of F^*. Then, by (3), $\mathbf{Wn}^{F^*}_e\langle\Gamma\rangle = \mathbf{Wn}^{L^*}_e\langle\rangle$. Thus, it remains to show that $\mathbf{Wn}^{L^*}_e\langle\rangle = \top$. Suppose, for a contradiction, that $\mathbf{Wn}^{L^*}_e\langle\rangle = \bot$. Then, according Lemma 1.1, $\|L\|$ is false in $\mathcal{CLM}^*_e$. But this is impossible because, by (1), $\|L\|$ is a tautology. $\Box$

3 Completeness

In our completeness proof for **CL1** we will employ the complementary logic **CL1$'$** given by the following two rules:

Rule (a): $\vec{H} \vdash F$, where F is instable and $\vec{H}$ is the smallest set of formulas satisfying the following condition: If F has a negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$), then, for each $i \in \{1, \ldots, n\}$, $\vec{H}$ contains the result of replacing this occurrence in F by G_i.

Rule (b): $H \vdash F$, where H is the result of replacing in F a positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) by G_i for some $i \in \{1, \ldots, n\}$.

Lemma 3.1 *$\mathbf{CL1} \not\vdash F$, then $\mathbf{CL1'} \vdash F$ (any F).*

PROOF We prove this lemma by induction on the complexity of F. There are two cases to consider:

Case 1: F is stable. Then there must be a **CL1**-unprovable formula H that is the result of replacing in F some positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) by G_i for some $i \in \{1, \ldots, n\}$ (otherwise, F would be **CL1**-derivale by Rule **(a)**). By the induction hypothesis $\mathbf{CL1'} \vdash H$, whence, by Rule **(b)**, $\mathbf{CL1'} \vdash F$.

Case 2: F is instable. Let $\vec{H}$ be the smallest set of formulas satisfying the following condition: If F has a negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$), then, for each $i \in \{1, \ldots, n\}$, $\vec{H}$ contains the result of replacing this occurrence in F by G_i. None of the elements of $\vec{H}$ is **CL1**-provable (otherwise F would be derivable in **CL1** by Rule **(b)**). Therefore, by the induction hypothesis, each element of $\vec{H}$ is **CL1′**-provable, whence, by Rule **(a)**, $\mathbf{CL1'} \vdash F$. $\Box$

Theorem 3.2 *If $\mathbf{CL1} \not\vdash F$, then F is not uniformly valid (any F).*

PROOF Assume $\mathbf{CL1} \not\vdash F$. Then, by Lemma 3.1, $\mathbf{CL1'} \vdash F$. Let us fix a particular **CL1′**-proof for F. We will be referring to at as "the proof", and referring to formulas occurring in the proof as "proof formulas". We construct the EPM $\mathcal{N}$ that works as follows. At the beginning, this machine creates a record E to hold proof formulas, initializes it to F, and then follows the interactive algorithm described below:

> **Procedure** LOOP: Act depending on which of the two rules was used to derive E in the proof:
>
> *Case of Rule* **(a)***:* Keep granting permission until the adversary makes a move αi, where α E-specifies a negative (resp. positive) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) and $i \in \{1, \ldots, n\}$. Let H be the result of substituting the above occurrence by G_i in E. Then update E to H, and repeat LOOP.
>
> *Case of Rule* **(b)***:* Let H be the (only) premise of E in the proof. H is the result of substituting, in E, a certain positive (resp. negative) surface occurrence of a subformula $G_1 \sqcap \ldots \sqcap G_n$ (resp. $G_1 \sqcup \ldots \sqcup G_n$) by G_i for some $i \in \{1, \ldots, n\}$. Let α be the E-specification of that occurrence. Then make the move αi, update E to H, and repeat LOOP.

Note the similarity between $\mathcal{N}$ and the EPM $\mathcal{M}$ from Section 2. The descriptions of the algorithms that these two machines follow are exactly the same, only with the words "positive" and "negative" interchanged. In view of the perfect symmetry between $\mathcal{M}$ and $\mathcal{N}$, between **CL1** and **CL1**$'$ and between clauses (a) and (b) of Lemma 1.2, arguing as in the previous section, we can conclude that, in each computation branch of $\mathcal{N}$, LOOP will be iterated only a finite number of times and that $\mathcal{N}$ is fair.

Let us fix an arbitrary HPM $\mathcal{H}$ and an arbitrary valuation e. Let Γ be the $\mathcal{H}$ vs $\mathcal{C}$ run on e (see Definition 20.5 of [1]). Since $\mathcal{N}$ is fair, such a run is well-defined. Thus, Γ is the run cospelled by the $(\mathcal{N}, e, \mathcal{H})$-branch which, according to Lemma 20.4 of [1], is an e-computation branch of $\mathcal{N}$.

Let L denote the value the record E of $\mathcal{N}$ at the beginning of the last iteration of LOOP in the $(\mathcal{N}, e, \mathcal{H})$-branch. The following two statements can be proven in a way fully symmetric to the way statements (1) and (3) were proven in Section 2:

$$L \textit{ is instable.} \tag{4}$$

$$\textit{For all } {}^*\textit{, either } \Gamma \textit{ is a } \top\textit{-illegal run of } F^*\textit{, or } \mathbf{Wn}_e^{F^*}\langle\Gamma\rangle = \mathbf{Wn}_e^{L^*}\langle\rangle. \tag{5}$$

The instability of L means that $\|L\|$ is not a tautology. Let us fix any classical model M in which $\|L\|$ is false. Let * be the interpretation such that, for any atom p,

$$p^* = \begin{cases} \top & \text{if } p \text{ is true in } M; \\ \bot & \text{if } p \text{ is false in } M. \end{cases}$$

It is rather obvious that then $M = \mathcal{CLM}_e^*$. Therefore, by Lemma 1.1, $\mathbf{Wn}_e^{L^*}\langle\rangle = \bot$. Consequently, by (5), either Γ is a $\top$-illegal run of F^*, or $\mathbf{Wn}_e^{F^*}\langle\Gamma\rangle = \bot$. In either case we have $\mathbf{Wn}_e^{F^*}\langle\Gamma\rangle = \bot$. This means that $\mathcal{H}$ does not win F^* against $\mathcal{N}$ (see [1], Definition 20.7). This, in turn, by Proposition 20.8 of [1], implies that $\mathcal{H}$ does not win F^*.

Thus, for every HPM $\mathcal{H}$ there is an interpretation * such that $\mathcal{H}$ does not win F^*. This means nothing but that F is not uniformly valid. $\square$

References

[1] *G.Japaridze*, "Introduction to computability logic". **Annals of Pure and Applied Logic** 123 (2003), pp. 1-99.