Infinitary Classical Logic:
Recursive Equations and Interactive Semantics

Michele Basaldella*

Université d'Aix–Marseille, CNRS, I2M, Marseille, France

`michele.basaldella@gmail.com`

In this paper, we present an interactive semantics for derivations in an infinitary extension of classical logic. The formulas of our language are possibly infinitary trees labeled by propositional variables and logical connectives. We show that in our setting every recursive formula equation has a unique solution. As for derivations, we use an infinitary variant of Tait–calculus to derive sequents.

The interactive semantics for derivations that we introduce in this article is presented as a debate (interaction tree) between a test $\mathscr{T}$ (derivation candidate, Proponent) and an environment $\neg\mathbf{S}$ (negation of a sequent, Opponent). We show a completeness theorem for derivations that we call interactive completeness theorem: the interaction between $\mathscr{T}$ (test) and $\neg\mathbf{S}$ (environment) does not produce errors (i.e., Proponent wins) just in case $\mathscr{T}$ comes from a syntactical derivation of $\mathbf{S}$.

1 Introduction

In this article, we present an *interactive semantics* for *derivations* — i.e., formal proofs — in a proof–system that we call *infinitary classical logic*.

Infinitary classical logic. The system we consider is an infinitary extension of *Tait–calculus* [8], a sequent calculus for *classical logic* which is often used to analyze the proof theory of classical arithmetic and its fragments. In Tait–calculus, formulas are built from positive and negated propositional variables by using disjunctions $\vee$ and conjunctions $\wedge$ of arbitrary (possibly infinite) arity. Negation $\neg$ is defined by using a generalized form of De Morgan's laws. As for derivations, sequents of formulas are derived by means of rules of inference with a possibly infinite number of premises. In Tait–calculus, formulas and derivations — when seen as trees — while not necessarily finitarily branching, are *well–founded*. In this work, we remove the assumption of well–foundedness, and we let formulas and derivations be infinitary in a broader sense. What we get is the system that we call *infinitary classical logic*.

Recursive equations. The main reason for introducing infinitary classical logic is our interest in studying *recursive (formula) equations* in a classical context. Roughly speaking, by recursive equation we mean a pair of formulas $(\mathbf{v}, \mathbf{F})$, that we write as $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$, where $\mathbf{v}$ is an atom and $\mathbf{F}$ is a formula which depends on $\mathbf{v}$, i.e., such that $\mathbf{v}$ occurs in $\mathbf{F}$. For instance, $\mathbf{v} \stackrel{\text{REC}}{=} \neg\mathbf{v} \vee \mathbf{v}$ is a recursive equation. A *solution* of a recursive equation, say $\mathbf{v} \stackrel{\text{REC}}{=} \neg\mathbf{v} \vee \mathbf{v}$, is any formula $\mathbf{G}$ which is equal to $\neg\mathbf{G} \vee \mathbf{G}$. Solutions of recursive equations are often called *recursive types*, and they have been studied extensively in the literature (see e.g., [2, 7] and the references therein). In this area, one usually aims at finding a mathematical space in which the (interpretations of) equations have a unique solution (or, at least, a canonical one). In this paper, we *define* formulas as (possibly infinitary) labeled trees, and we prove existence and uniqueness of solutions of equations within the "space" of the formulas. As it turns out, if $\mathbf{G}$ is the solution of a recursive equation (i.e., a recursive type), then $\mathbf{G}$ is not well–founded. This fact motivates us to consider

---

infinitary formulas in our broader sense.

Derivations and tests. Since formulas are infinitary, we let derivations be infinitary as well. We obtain a cut–free sequent calculus in which the solutions of all recursive equations are derivable (in the sense of Theorem 3.16(2)). As expected, since we deal with ill–founded (i.e., non–well–founded) derivations, the price to pay for this huge amount of expressivity is that our calculus is inconsistent (in the sense of Remark 3.17(c)). In spite of this, it is precisely this notion of infinitary derivation that we want to study in this work. To this aim, we introduce a semantics *for derivations*, as we now explain.

Traditionally, the proof theory of classical logic is centered around the notion of *derivability*. In this paper, we are interested in analyzing the structure of our infinitary *derivations*. To this aim, we introduce the notion of *test*. A test $\mathscr{T}$ is a tree labeled by logical rules (no sequents), and the fundamental relation between tests and derivations can be informally stated as follows: given a sequent $\mathbf{S}$, if it is the case that by adding sequent information in an appropriate way to $\mathscr{T}$ we obtain a derivation of $\mathbf{S}$, then we say that $\mathscr{T}$ *comes from a derivation of* $\mathbf{S}$. In this article, the *syntactical* concept that we investigate is "$\mathscr{T}$ comes from a derivation of $\mathbf{S}$" rather than the traditional one "$\mathbf{S}$ is derivable." Also, note that our concept is *stronger* than the usual one: if $\mathscr{T}$ comes from a derivation of $\mathbf{S}$, then $\mathbf{S}$ is obviously derivable!

To grasp the idea behind our concept from another viewpoint, consider the lambda calculus. By the Curry–Howard correspondence, untyped lambda terms can be seen as "tests" for natural deduction derivations, and "$\mathscr{T}$ comes from a derivation of $\mathbf{S}$" can be read as "the untyped lambda term $\mathscr{T}$ has (simple) type $\mathbf{S}$ in the Curry–style type assignment."

Interactive semantics and completeness. Traditionally, in order to study the concept of derivability, one introduces a notion of model and eventually shows a *completeness theorem*: the *syntactical* notion of *derivability* and the *semantical* notion of *validity* (as usual, *valid* means "true in every model") coincide. Here, we are interested in proving a completeness theorem as well. But, since we replace the syntactical concept of derivability with "$\mathscr{T}$ comes from a derivation of $\mathbf{S}$", we need to replace the semantical notion of validity with something else. So, we now let our interactive semantics enter the stage.

In few words, our interactive semantics is organized as follows: first, we introduce the notion of *environment* $\neg\mathbf{S}$ (the negation of a sequent $\mathbf{S}$) and then, we make the test $\mathscr{T}$ and the environment $\neg\mathbf{S}$ *interact*. More precisely, we introduce the notion of *configuration* (a pair of the form $(\mathscr{T}, \neg\mathbf{S})$) and define a procedure which makes configurations evolve from the initial configuration $(\mathscr{T}, \neg\mathbf{S})$ by means of a transition relation. As a result, we get a tree of configurations that we call an *interaction tree*. The procedure which determines the interaction tree is *interactive* in the sense that it can be seen as a debate between two players: Proponent (the test $\mathscr{T}$) and Opponent (the environment $\neg\mathbf{S}$). Opponent *wins the debate* if the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ produces an *error*, i.e., a position in the interaction tree generated from $(\mathscr{T}, \neg\mathbf{S})$ which is labeled by an error symbol. Otherwise, Proponent *wins the debate*. Our main result is the *interactive completeness theorem*:

$$\mathscr{T} \text{ comes from a derivation of } \mathbf{S} \quad \text{if and only if} \quad \begin{array}{l} \text{the interaction between } \mathscr{T} \text{ and } \neg\mathbf{S} \\ \text{does not produce errors.} \end{array} \quad (\star)$$

That is, $\mathscr{T}$ comes from a derivation of $\mathbf{S}$ if and only if Proponent wins the debate.

The motivation for introducing our semantics comes from our interest in extending the completeness theorem of ludics [5] to logics which are not necessarily polarized fragments of linear logic. The completeness theorem of [1] (called interactive completeness also there) can be stated, up to terminology and notation, in our setting as

$$\mathscr{T} \text{ comes from a derivation of } \mathbf{S} \quad \text{if and only if} \quad \begin{array}{l} \text{for no } \mathscr{M} \in \neg\mathbf{S}, \text{ the interaction between} \\ \mathscr{T} \text{ and } \mathscr{M} \text{ produces errors.} \end{array} \quad (\star\star)$$

The crucial point is, of course, the RHS of $(\star\star)$. Here, $\mathbf{S}$ represents a formula of a polarized fragment

of linear logic, and $\mathscr{T}$ and $\mathscr{M}$ are designs (proof–like objects similar to our tests). In [1], formulas are interpreted as sets of designs such that $\neg\neg\mathbf{S} = \mathbf{S}$, where $\neg\mathbf{S}$ is the set of designs given by: $\mathscr{M} \in \neg\mathbf{S}$ just in case for every $\mathscr{P} \in \mathbf{S}$, the interaction between $\mathscr{P}$ and $\mathscr{M}$ does not produce errors (so, the RHS of $(\star\star)$ means $\mathscr{T} \in \neg\neg\mathbf{S} = \mathbf{S}$). In [1], the result of the interaction between $\mathscr{T}$ and $\mathscr{M}$ is determined by a procedure of reduction for designs which reflects the procedure of cut–elimination of the underlying logic. Of course, it would be very nice if we could use the same approach in our setting!

Unfortunately, if one tries to remove the polarities from ludics, then one encounters several *technical* problems related to cut–elimination (that we do not discuss here). To the present author, the most convenient way to cope with non–polarized logics, is to build a new framework from the very beginning, keeping the *format* of the statement $(\star\star)$ as guiding principle, the rest being — in case — sacrificed. The choice here is purely personal: the present author believes that the significance of ludics is ultimately justified by the completeness theorem not e.g., by the fact that the interpretation of the logical formulas is induced by a procedure of reduction for designs. This describes the origin of this work.

Finally, we note that the RHS of $(\star)$ is just the RHS of $(\star\star)$ in case $\neg\mathbf{S}$ is a singleton (and so, it can be identified with $\neg\mathbf{S}$ itself). Indeed, this is the case for the interactive semantics presented in this paper.

As for future work, we plan to adapt our interactive semantics to analyze derivations in second–order propositional classical logic.

Outline. This paper is organized as follows. In Section 2 we recall some preliminary notions about labeled trees. In Section 3 we introduce formulas and recursive formula equations, and we prove the existence and the uniqueness of solutions of equations. We also define the notion of derivation and discuss the derivability of some sequents. In Section 4 we present our interactive semantics and prove the interactive completeness theorem.

2   Preliminaries: Positions and Labeled Trees

In this section, we recall the basic notions of position and labeled tree. We also establish some notation and terminology that we extensively use in the sequel.

Definition 2.1 (Position, length)**.** Let $\mathbb{N}$ be the set of the natural numbers. Let ∞ be any object such that $\infty \notin \mathbb{N}$. We call **position** any function $p : \mathbb{N} \longrightarrow \mathbb{N} \cup \{\infty\}$ which satisfies the following property:

 (P)   for some $n \in \mathbb{N}$, $p(k) \in \mathbb{N}$ for all $k < n$ and $p(k) = \infty$ for all $k \geq n$.

Since $\infty \notin \mathbb{N}$, the natural number n above is unique. We call it the **length** of the position and denote it by $\lg(p)$. The set of all positions is denoted by $\mathbb{N}^\star$ and we use $p, q, r \ldots$ to range over its elements. $\triangle$

Notation and Terminology 2.2. Let p, q and r be positions, and let U, V and W be sets of positions.

(1) A position p is also written as $\langle p(0), \ldots, p(\lg(p) - 1) \rangle$. In particular, we write $\langle \rangle$ for the unique position of length 0 that we call the *empty position*.

(2) The *concatenation* of p and q is the position $p \star q$ defined as follows:

$$p \star q(k) \;\stackrel{\text{DEF}}{=}\; \begin{cases} p(k) & \text{if } k < \lg(p) \\ q(k - \lg(p)) & \text{if } k \geq \lg(p) \;, \end{cases} \qquad \text{for } k \in \mathbb{N} \;.$$

In other words, $p \star q = \langle p(0), \ldots, p(\lg(p) - 1), q(0), \ldots, q(\lg(q) - 1) \rangle$. The operation of concatenation is associative (i.e., $(p \star q) \star r = p \star (q \star r)$) and it has $\langle \rangle$ as neutral element (i.e., $\langle \rangle \star p = p = p \star \langle \rangle$). Note also that $\lg(p \star q) = \lg(p) + \lg(q)$.

(3) We write $p \sqsubseteq q$ if $q = p \star t$ for some $t \in \mathbb{N}^\star$, and we say that q is an *extension of* p and that p is a *restriction of* q. If $\lg(t) \geq 1$ (resp. $\lg(t) = 1$), then we also say that q is a *proper* (resp. *immediate*) extension of p and that p is a *proper* (resp. *immediate*) restriction of q, and we write $p \sqsubset q$ (resp. $p \sqsubset_1 q$).

(4) We write $U \star V$ for the set $\{p \star q \mid p \in U \text{ and } q \in V\}$. Note that $U \star (V \star W) = (U \star V) \star W$, and that $U \star \bigcup_{i \in I} V_i = \bigcup_{i \in I} (U \star V_i)$, for every family $\{V_i\}_{i \in I}$ of sets of positions. $\qquad \triangle$

Definition 2.3 (Labeled tree, domain)**.** Let L be a set. Let ∞_L be any object such that $\infty_L \notin L$. A **tree labeled by** L is a function $T : \mathbb{N}^\star \longrightarrow L \cup \{\infty_L\}$ which satisfies the following properties:

(T$_1$) $T(\langle \rangle) \in L$;

(T$_2$) if $T(p) \in L$ and $q \sqsubseteq p$, then $T(q) \in L$.

We call the set $\{p \in \mathbb{N}^\star \mid T(p) \in L\}$ the **domain** of T and we denote it as $\mathrm{dom}(T)$. $\qquad \triangle$

Let T and U be trees labeled by L. Since $T(q) = \infty_L$ for all $q \notin \mathrm{dom}(T)$, the tree T is completely determined by the values it takes on its domain. In particular,

$$T = U \quad \text{if and only if} \quad \mathrm{dom}(T) = \mathrm{dom}(U) \ \text{and} \ T(p) = U(p) \text{ for all } p \in \mathrm{dom}(T) \ .$$

Notation and Terminology 2.4. Let T and U be trees labeled by L, and let $p \in \mathrm{dom}(T)$.

(1) We say that p is a *leaf of* T if there is no $q \in \mathrm{dom}(T)$ such that $p \sqsubset q$.

(2) The *subtree of* T *above* p is the tree T_p labeled by L defined as follows:

$$\mathrm{dom}(T_p) \overset{\text{DEF}}{=} \{q \in \mathbb{N}^\star \mid p \star q \in \mathrm{dom}(T)\} \quad \text{and} \quad T_p(q) \overset{\text{DEF}}{=} T(p \star q) \ , \text{ for } q \in \mathrm{dom}(T_p) \ .$$

Note that we have $T_{\langle \rangle} = T$ and $(T_p)_q = T_{p \star q}$, for every p and q in $\mathbb{N}^\star$ such that $p \star q \in \mathrm{dom}(T)$.

We say that U is a *subtree of* T if $U = T_q$, for some $q \in \mathrm{dom}(T)$. If $\mathrm{lg}(q) = 1$, then we also say that U is an *immediate subtree of* T.

(3) If L is a product, i.e., $L = A \times B$ for some sets A and B, then we write $T_\mathsf{L}(p)$ and $T_\mathsf{R}(p)$ for the left and the right component of $T(p)$ respectively (i.e., if $T(p) = (a,b)$, then $T_\mathsf{L}(p) = a$ and $T_\mathsf{R}(p) = b$).

(4) We say that T is *ill–founded* if there exists a function $f : \mathbb{N} \longrightarrow \mathrm{dom}(T)$ such that $f(n) \sqsubset f(n+1)$ for every $n \in \mathbb{N}$. We also say that T is *well–founded* if it is not ill–founded. $\qquad \triangle$

3 Infinitary Classical Logic

In this section, we present our infinitary version of classical logic. In Subsection 3.1 we introduce formulas as possibly infinitary labeled trees, and in Subsection 3.2 we introduce the notion of recursive formula equation and prove that solutions of equations exist and they are unique. Finally, in Subsection 3.3 we introduce the concept of derivation and observe some basic properties.

3.1 Formulas

We now define the concept of formula and the operation of negation.

Definition 3.1 (Propositional variable)**.** Let $+$ and $-$ be two distinct symbols. Let $\mathscr{V} \overset{\text{DEF}}{=} \{+\} \times \mathbb{N}$ and $\neg \mathscr{V} \overset{\text{DEF}}{=} \{-\} \times \mathbb{N}$. We call the elements of $\mathscr{V}$ (resp. $\neg \mathscr{V}$) **positive** (resp. **negated**) **propositional variables**, and we denote a generic element $(+, v)$ of $\mathscr{V}$ (resp. $(-, v)$ of $\neg \mathscr{V}$) by $\mathbf{v}$ (resp. $\neg \mathbf{v}$). $\qquad \triangle$

Definition 3.2 (Formula)**.** Let $\vee$ and $\wedge$ be two distinct symbols such that $\{\vee, \wedge\} \cap (\mathscr{V} \cup \neg \mathscr{V}) = \emptyset$. We call **formula** any tree T labeled by $\{\vee, \wedge\} \cup \mathscr{V} \cup \neg \mathscr{V}$ which satisfies the following property:

(F) $\qquad$ for every $p \in \mathrm{dom}(T)$, if $T(p) \in \mathscr{V} \cup \neg \mathscr{V}$, then p is a leaf of T.

In the sequel, we use the letters $\mathbf{F}, \mathbf{G}, \mathbf{H}, \dots$ to range over formulas. $\qquad \triangle$

Notation and Terminology 3.3. Let $\mathbf{F}$ be a formula.

(1) Since for every $p \in \mathrm{dom}(\mathbf{F})$, the subtree $\mathbf{F}_p$ is a formula, we say that $\mathbf{F}_p$ is a *subformula of* $\mathbf{F}$ and that $\mathbf{F}_p$ *occurs in* $\mathbf{F}$. If $\mathrm{lg}(p) = 1$ then we also say that $\mathbf{F}_p$ is an *immediate subformula of* $\mathbf{F}$.

(2) If $\mathbf{F}(\langle\rangle) \in \mathcal{V} \cup \neg\mathcal{V}$, then $\mathrm{dom}(\mathbf{F}) = \{\langle\rangle\}$ and we say that $\mathbf{F}$ is an **atom**. If $\mathbf{F}(\langle\rangle) = \mathbf{v} \in \mathcal{V}$ (resp. $\mathbf{F}(\langle\rangle) = \neg\mathbf{v} \in \neg\mathcal{V}$), then we abusively denote $\mathbf{F}$ by $\mathbf{v}$ (resp. $\neg\mathbf{v}$) and we say that $\mathbf{F}$ is a **positive** (resp. **negated**) **atom**.

(3) If $\mathbf{F}(\langle\rangle) \in \{\vee, \wedge\}$, then we call $\mathbf{F}$ a **compound formula**. The set of natural numbers

$$I \stackrel{\text{DEF}}{=} \{i \in \mathbb{N} \mid \langle i\rangle \in \mathrm{dom}(\mathbf{F})\}$$

is said to be the **arity of F**. Let $\mathbf{F}(\langle\rangle) = \vee$ (resp. $\mathbf{F}(\langle\rangle) = \wedge$). Since for every $i \in I$ the formula $\mathbf{F}_{\langle i\rangle}$ is an immediate subformula of $\mathbf{F}$, we also denote $\mathbf{F}$ by $\vee_I \mathbf{F}_{\langle i\rangle}$ (resp. $\wedge_I \mathbf{F}_{\langle i\rangle}$) and we say that $\mathbf{F}$ is a **disjunctive** (resp. **conjunctive**) **formula**. If $I = \{0, \ldots, n-1\}$ for some $n \in \mathbb{N}$, then we also write $\vee[\mathbf{F}_{\langle 0\rangle}, \ldots, \mathbf{F}_{\langle n-1\rangle}]$ (resp. $\wedge[\mathbf{F}_{\langle 0\rangle}, \ldots, \mathbf{F}_{\langle n-1\rangle}]$) for $\mathbf{F}$. Obviously, if we write something like "let $\mathbf{F}$ be $\vee[\mathbf{G}_0, \ldots, \mathbf{G}_{n-1}] \ldots$" (resp. $\wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{n-1}]$), then we mean that $\mathbf{F}$ is a disjunctive (resp. conjunctive) formula of arity $\{0, \ldots, n-1\}$ and that $\mathbf{G}_k = \mathbf{F}_{\langle k\rangle}$, for each $k < n$. Similarly, we may use the expression $\vee[\mathbf{G}, \mathbf{H}]$ (resp. $\wedge[\mathbf{G}, \mathbf{H}]$) to denote the disjunctive (resp. conjunctive) formula $\mathbf{F}$ whose arity is $\{0, 1\}$ and such that $\mathbf{F}_{\langle 0\rangle} = \mathbf{G}$ and $\mathbf{F}_{\langle 1\rangle} = \mathbf{H}$, and so on. Finally, if $I = \emptyset$ — that is, $\mathrm{dom}(\mathbf{F}) = \{\langle\rangle\}$ — then we write f (for *false*) and t (for *true*) rather than $\vee[\,]$ (resp. $\wedge[\,]$) respectively. $\triangle$

Definition 3.4 (Negation). Let $\mathbf{F}$ be a formula. The formula $\neg\mathbf{F}$, that we call the **negation of F**, is defined as follows: $\mathrm{dom}(\neg\mathbf{F}) \stackrel{\text{DEF}}{=} \mathrm{dom}(\mathbf{F})$ and

$$\neg\mathbf{F}(p) \stackrel{\text{DEF}}{=} \begin{cases} \neg\mathbf{v} & \text{if } \mathbf{F}(p) = \mathbf{v} \\ \mathbf{v} & \text{if } \mathbf{F}(p) = \neg\mathbf{v} \\ \vee & \text{if } \mathbf{F}(p) = \wedge \\ \wedge & \text{if } \mathbf{F}(p) = \vee, \end{cases} \qquad \text{for } p \in \mathrm{dom}(\neg\mathbf{F}) . \qquad \triangle$$

We observe that the negation is involutive, i.e., $\neg\neg\mathbf{F} = \mathbf{F}$, for every formula $\mathbf{F}$. Furthermore,

$$\neg \vee_I \mathbf{F}_{\langle i\rangle} = \wedge_I \neg\mathbf{F}_{\langle i\rangle} \quad \text{and} \quad \neg \wedge_I \mathbf{F}_{\langle i\rangle} = \vee_I \neg\mathbf{F}_{\langle i\rangle} .$$

According to Definition 3.2, formulas are allowed to be *infinitary*: they may have an infinite set $I \subseteq \mathbb{N}$ as arity, and they can also be ill–founded. In Tait's work [8], the situation is rather different: the arity of a compound formula need not be a subset of $\mathbb{N}$, and only well–founded formulas are considered.

Let us discuss our choices. As for the first difference, we restrict attention to sets of natural numbers mainly for expository reasons; a pleasant consequence of this choice is that we can define sequents as *formulas* (Definition 3.11) rather than finite sets (as in [8]), multi–sets, or sequences of formulas. As for the second one, we have to consider ill–founded formulas because solutions of recursive equations are formulas which *always* have this property (see Theorem 3.9).

We finally observe that the principles of (transfinite) induction and recursion cannot be applied to ill–founded formulas. In particular, — even though we are in classical world — it is not possible to give to our formulas a Tarskian definition of *truth*. Nevertheless, in our setting it is possible to define a reasonable notion of *derivation*, as we do in Subsection 3.3.

3.2 Recursive Formula Equations

In this subsection, we define the concepts of recursive equation and solution of an equation. We prove that every recursive equation has a unique solution. We also give some concrete examples.

Definition 3.5 (Substitution). Let $\mathbf{F}$ and $\mathbf{G}$ be formulas, and let $\mathbf{v}$ be a *positive* atom. We define the formula $\mathbf{F}[\mathbf{G}/\mathbf{v}]$ obtained by the **substitution of G for v and of $\neg$G for $\neg$v in F** as follows. Let $R \stackrel{\text{DEF}}{=} \{r \in \mathrm{dom}(\mathbf{F}) \mid \mathbf{F}(r) \in \{\mathbf{v}, \neg\mathbf{v}\}\}$ and $S \stackrel{\text{DEF}}{=} \mathrm{dom}(\mathbf{F}) \setminus R$. We set $\mathrm{dom}(\mathbf{F}[\mathbf{G}/\mathbf{v}]) \stackrel{\text{DEF}}{=} S \cup (R \star \mathrm{dom}(\mathbf{G}))$ and

$$\mathbf{F}[\mathbf{G}/\mathbf{v}](p) \stackrel{\text{DEF}}{=} \begin{cases} \mathbf{F}(p) & \text{if } p \in S \\ \mathbf{G}(q) & \text{if } p = r \star q \text{ and } \mathbf{F}(r) = \mathbf{v} \\ \neg \mathbf{G}(q) & \text{if } p = r \star q \text{ and } \mathbf{F}(r) = \neg \mathbf{v} \end{cases}, \quad \text{for } p \in \text{dom}(\mathbf{F}[\mathbf{G}/\mathbf{v}]) . \qquad \triangle$$

The correctness of the previous definition is justified by the following lemma.

Lemma 3.6. *Let R and S be as in* Definition 3.5.

(a) *For every $r \in R$ and every $t \in \mathbb{N}^\star$, if $p = r \star t$ then $p \notin S$.*

(b) *Let $V \subseteq \mathbb{N}^\star$, and let $p \in R \star V$. Suppose that there are r, r' in R and q, q' in V such that $p = r \star q$ and $p = r' \star q'$. Then, $r = r'$ and $q = q'$.*

Proof. (a) If $t = \langle \rangle$, then $p = r \star \langle \rangle = r \in R$. If $t \neq \langle \rangle$, then $p \notin \text{dom}(\mathbf{F})$, as r is a leaf of $\mathbf{F}$. Hence, $p \notin S$.

(b) Since both r and r' are restrictions of p, we have $r \sqsubseteq r'$ or $r' \sqsubseteq r$. Since r and r' are leaves of $\mathbf{F}$, we conclude $r = r'$ (and hence $q = q'$). $\qquad \square$

The following proposition easily follows from our definition of substitution.

Proposition 3.7. *Let $\mathbf{F}$ and $\mathbf{G}$ be formulas, and let $\mathbf{v}$ be a positive atom. Then:*

(1) $\mathbf{v}[\mathbf{G}/\mathbf{v}] = \mathbf{G}$ *and* $(\neg \mathbf{v})[\mathbf{G}/\mathbf{v}] = \neg \mathbf{G}$;

(2) $\left(\vee_I \mathbf{F}_{\langle i \rangle} \right)[\mathbf{G}/\mathbf{v}] = \vee_I \left(\mathbf{F}_{\langle i \rangle}[\mathbf{G}/\mathbf{v}] \right)$ *and* $\left(\wedge_I \mathbf{F}_{\langle i \rangle} \right)[\mathbf{G}/\mathbf{v}] = \wedge_I \left(\mathbf{F}_{\langle i \rangle}[\mathbf{G}/\mathbf{v}] \right)$;

(3) $\mathbf{F}[\mathbf{G}/\mathbf{v}] = \mathbf{F}$, *if neither $\mathbf{v}$ nor $\neg \mathbf{v}$ is a subformula of $\mathbf{F}$;*

(4) $\neg(\mathbf{F}[\mathbf{G}/\mathbf{v}]) = (\neg \mathbf{F})[\mathbf{G}/\mathbf{v}]$. $\qquad \square$

Definition 3.8 (Recursive formula equation, solution)**.** A **recursive formula equation** (or **recursive equation**, or just **equation**) is an ordered pair of formulas $(\mathbf{v}, \mathbf{F})$, that we write as $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$, such that:

(R$_1$) $\mathbf{v}$ is a positive atom;

(R$_2$) $\mathbf{F}$ is a compound formula;

(R$_3$) $\mathbf{F}(p) \in \{\mathbf{v}, \neg \mathbf{v}\}$, for some $p \in \text{dom}(\mathbf{F})$.

A **solution** of $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$ is a formula $\mathbf{G}$ such that $\mathbf{G} = \mathbf{F}[\mathbf{G}/\mathbf{v}]$. $\qquad \triangle$

We now discuss the previous definition. To begin with, note that by (R$_2$), a pair of atoms such as $(\mathbf{v}, \mathbf{a})$ is *not* a recursive equation. The reason to exclude such pairs is to avoid to consider $(\mathbf{v}, \mathbf{v})$ — which would be a trivial equation, as every formula would be a solution — and $(\mathbf{v}, \neg \mathbf{v})$ — which would have no solution at all. Up to now, our choices are perfectly in line with [3, 2]. In our setting, we also impose the additional condition (R$_3$). The aim of this clause is to exclude pairs of the form $(\mathbf{v}, \mathbf{F})$ where neither $\mathbf{v}$ nor $\neg \mathbf{v}$ actually occurs in $\mathbf{F}$. The reason is that pairs like the one above would be trivial equations as well: by Proposition 3.7(3), $\mathbf{F}$ itself would be the unique solution, as we have $\mathbf{F}[\mathbf{G}/\mathbf{v}] = \mathbf{F}$ for every formula $\mathbf{G}$.

We now turn attention to the literature on recursive types. In this topic, the theorem which states existence and uniqueness of solutions of recursive equations is perhaps the most important result. One usual way to prove it is to show that the mathematical space (in our case, it would be the set of formulas) forms a complete metric space with respect to some metric. Then, the result follows by applying Banach's fixpoint theorem, by using the fact that the operation of substitution induces a contractive map from the space to itself [3, 2]. Another traditional way to prove that result requires to introduce in the space some notion of approximation. One then shows that suitable sequences of "lower" (resp. "upper") approximations converge to a "lower" (resp. "upper") solution of the equation. To prove the result, one eventually proves that the two solutions coincide (see e.g., [7, 1] and the references therein).

By contrast, in our setting we are able to prove the result in a direct and elementary way; our method does not require to explicitly introduce any sort of metric or notions of approximation. We do not claim

that our *result* is new, as there are several *similar* results in the literature. However, we believe that our *proof* deserves some attention, as it is quite simple and self–contained.

Theorem 3.9 (Existence and uniqueness of solutions). *Every recursive equation* $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$ *has a unique solution* $\mathbf{G}$. *Furthermore,* $\mathbf{G}$ *is ill–founded.*

Proof. Let R and S be as in Definition 3.5. First, we observe that by condition (R_2) of Definition 3.8, $\langle\rangle \in S$. In particular, $\langle\rangle \notin R$. Furthermore, by condition (R_3), R is non–empty.

We now remark that by definition of substitution, if $\mathbf{H}$ is a solution of the equation $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$, then the set of positions $\text{dom}(\mathbf{H})$ has to satisfy $\text{dom}(\mathbf{H}) = S \cup (R \star \text{dom}(\mathbf{H}))$. So, we are interested in studying those sets of positions X such that $X = S \cup (R \star X)$. To this aim, we define

$$A_0 \stackrel{\text{DEF}}{=} S \;, \qquad A_{n+1} \stackrel{\text{DEF}}{=} R \star A_n \;, \qquad A_{<n} \stackrel{\text{DEF}}{=} \bigcup_{k<n} A_k \quad \text{and} \quad A \stackrel{\text{DEF}}{=} \bigcup_{n\in\mathbb{N}} A_n \;.$$

The key property, that we show in a moment, is that A is the *unique* set such that $A = S \cup (R \star A)$. This fact is known as *Arden's Rule* in the literature of formal languages theory (see e.g., [6]).

We now prove this fact in our setting (our proof is adapted from the one given in [6]).

(i) *Arden's Rule:* $\quad X = S \cup (R \star X) \quad$ if and only if $\quad X = A$.

Proof of (i). If $X = A$, then $A = \bigcup_{n\in\mathbb{N}} A_n = A_0 \cup \bigcup_{n>0} A_n = A_0 \cup (R \star \bigcup_{n\in\mathbb{N}} A_n) = S \cup (R \star A)$.

To show the converse, let X be such that $X = S \cup (R \star X)$. For each $n \in \mathbb{N}$ we define $X_0 \stackrel{\text{DEF}}{=} X$, $X_{n+1} \stackrel{\text{DEF}}{=} R \star X_n$. By induction on n, we show that for every $n \in \mathbb{N}$ we have $X = A_{<n} \cup X_n$. If $n = 0$, then $X = \emptyset \cup X_0 = X$. Let $n = m+1$ and assume $A_{<m} \cup X_m = X$. We have $A_{<m+1} \cup X_{m+1} = A_0 \cup (R \star A_{<m}) \cup (R \star X_m) = A_0 \cup (R \star (A_{<m} \cup X_m)) = S \cup (R \star X) = X$. Now, since $\langle\rangle \notin R$, for each $p \in X$ we have $p \notin X_{\text{lg}(p)+1}$, as $\text{lg}(q) > \text{lg}(p)$ for every $q \in X_{\text{lg}(p)+1}$. Hence, if $p \in X = A_{<\text{lg}(p)+1} \cup X_{\text{lg}(p)+1}$, then $p \in A_{<\text{lg}(p)+1} \subseteq A$. Finally, if $p \in A$, then $p \in A_n$ for some $n \in \mathbb{N}$. Therefore, $p \in A_n \subseteq A_{<n+1} \cup X_{n+1} = X$. $\qquad\square$

By (i), the domain of *any* solution of the recursive equation $\mathbf{v} \stackrel{\text{REC}}{=} \mathbf{F}$ has to be A. However, this fact does not give us any hint about how to define the *values* of a solution $\mathbf{G}$ (i.e., $\mathbf{G}(p)$, for $p \in A$).

To tackle this problem, we now show the following properties.

(ii) (a) For every $n \in \mathbb{N}$, each $p \in A_n$ can be factorized in a *unique* way as $p = r_0^p \star \cdots \star r_{n-1}^p \star s^p$, where $r_0^p, \ldots, r_{n-1}^p \in R$ and $s^p \in S$.

    (b) For every $n \in \mathbb{N}$ and every $m \in \mathbb{N}$, if $n \neq m$ then A_n and A_m are disjoint.

Proof of (ii). (a) By induction on n. If $n = 0$, then $p \in A_0 = S$. By Lemma 3.6(a), for no $r \in R$ and $t \in \mathbb{N}^\star$ we have $p = r \star t$. So, we set $s^p \stackrel{\text{DEF}}{=} p$. If $n = m+1$, then $p \in A_{m+1} = R \star A_m$. By Lemma 3.6(b), there is a unique $r \in R$ and a unique $q \in A_m$ such that $p = r \star q$. By inductive hypothesis, q can be written as $q = r_0^q \star \cdots \star r_{m-1}^q \star s^q$. Hence, p can be factorized as $p = r_0^p \star \cdots \star r_{n-1}^p \star s^p$, where $r_0^p \stackrel{\text{DEF}}{=} r$, $r_k^p \stackrel{\text{DEF}}{=} r_{k-1}^q$ for every $1 \leq k < n$, and $s^p \stackrel{\text{DEF}}{=} s^q$.

(b) Suppose, for a contradiction, that for some $p \in A$ there are n and m in $\mathbb{N}$ such that $n < m$, $p \in A_n$ and $p \in A_m$. Let n be the least natural number for which this fact holds. By Lemma 3.6(a), we have $n > 0$, as for no $t \in \mathbb{N}^\star$ it is the case that $p \in A_0 = S$ and $p = r \star t \in A_m$. Suppose now that $n > 0$, $p = r \star t \in A_n$ and $p = r' \star t' \in A_m$. Since $p \in R \star A$, we can apply Lemma 3.6(b) and we obtain $r = r'$ and $t = t'$. But then, we have $t \in A_{n-1}$ and $t \in A_{m-1}$. This contradicts the minimality of n. $\qquad\square$

By (ii), it follows that each $p \in A$ can be uniquely factorized as $p = r_0^p \star \cdots \star r_{n_p-1}^p \star s^p$, where n_p is the unique $n \in \mathbb{N}$ such that $p \in A_n$. We are now ready to define $\mathbf{G}$.

Let $p \in A$ and let $m_p \in \mathbb{N}$ be the cardinality of $\{j < n_p \mid \mathbf{F}(r_j^p) = \neg\mathbf{v}\}$. We set $\text{dom}(\mathbf{G}) \stackrel{\text{DEF}}{=} A$ and

$$\mathbf{G}(p) \stackrel{\text{DEF}}{=} \begin{cases} \mathbf{F}(s^p) & \text{if } m_p \text{ is even} \\ \neg\mathbf{F}(s^p) & \text{if } m_p \text{ is odd} \end{cases}, \quad \text{for } p \in \text{dom}(\mathbf{G}) \;.$$

We now show that $\mathbf{G}$ is a formula and that it is a solution of $\mathbf{v} \overset{\text{REC}}{=} \mathbf{F}$.

(iii) (1) $\langle \rangle \in \text{dom}(\mathbf{G})$ and if $q \sqsubseteq p$, then $q \in \text{dom}(\mathbf{G})$.

  (2) If $\mathbf{G}(p) \in \mathscr{V} \cup \neg\mathscr{V}$, then p is a leaf of $\mathbf{G}$.

  (3) $\mathbf{G} = \mathbf{F}[\mathbf{G}/\mathbf{v}]$.

Proof of (iii). (1) $\langle \rangle \in S = A_0 \subseteq \text{dom}(\mathbf{G})$. If $q \sqsubseteq p$ then either $q = r_0^p \star \cdots \star r_{n_p-1}^p \star t$ and $t \sqsubseteq s^p$, or $q = r_0^p \star \cdots \star r_{k-1}^p \star t$, for $k < n_p$ and $t \sqsubset r_k^p$. In both cases, $t \in S$. Thus, $q \in A_{n_p} \cup A_k \subseteq A = \text{dom}(\mathbf{G})$.

  (2) Let $\mathbf{G}(p) \in \mathscr{V} \cup \neg\mathscr{V}$. As $\mathbf{G}(p) \in \{\mathbf{F}(s^p), \neg\mathbf{F}(s^p)\}$, we have $\mathbf{G}(p) \in \mathscr{V} \cup \neg\mathscr{V}$ just in case $\mathbf{F}(s^p) \in \mathscr{V} \cup \neg\mathscr{V}$. Since s^p is a leaf of $\mathbf{F}$ and $s^p \notin R$, no proper extension of p is in $\text{dom}(\mathbf{G})$. So, p is a leaf of $\mathbf{G}$.

  (3) By (i), $\text{dom}(\mathbf{G}) = A = S \cup (R \star A) = S \cup (R \star \text{dom}(\mathbf{G})) = \text{dom}(\mathbf{F}[\mathbf{G}/\mathbf{v}])$. Let $p \in \text{dom}(\mathbf{G})$. If $n_p = 0$, then $p = s^p \in S$ and $\mathbf{G}(p) = \mathbf{F}(p) = \mathbf{F}[\mathbf{G}/\mathbf{v}](p)$. Suppose now that $n_p > 0$. Let q be $r_1^p \star \cdots \star r_{n_p-1}^p \star s^p$. By definition, $s^p = s^q$ (see the proof of (ii)(a)). If $\mathbf{F}(r_0^p) = \mathbf{v}$, then $m_p = m_q$ and $\mathbf{G}(p) = \mathbf{G}(q) = \mathbf{F}[\mathbf{G}/\mathbf{v}](p)$. If $\mathbf{F}(r_0^p) = \neg\mathbf{v}$, then $m_p = m_q + 1$ and $\mathbf{G}(p) = \neg\mathbf{G}(q) = \mathbf{F}[\mathbf{G}/\mathbf{v}](p)$. $\qquad\square$

We now prove that $\mathbf{G}$ is the unique solution of the recursive equation $\mathbf{v} \overset{\text{REC}}{=} \mathbf{F}$.

(iv) If $\mathbf{H}$ is a formula such that $\mathbf{H} = \mathbf{F}[\mathbf{H}/\mathbf{v}]$, then $\mathbf{H} = \mathbf{G}$.

Proof of (iv). By (i), $\text{dom}(\mathbf{H}) = A$. We now prove that for every $n \in \mathbb{N}$, for each $p \in A_n$ we have $\mathbf{H}(p) = \mathbf{G}(p)$. We reason by induction on n. If $n = 0$, then $p \in S$. Hence, $\mathbf{H}(p) = \mathbf{F}[\mathbf{H}/\mathbf{v}](p) = \mathbf{F}(p) = \mathbf{G}(p)$. Suppose now that $n = m+1$. Let q be $r_1^p \star \cdots \star r_{n-1}^p \star s^p \in A_m$. By inductive hypothesis, $\mathbf{H}(q) = \mathbf{G}(q)$. If $\mathbf{F}(r_0^p) = \mathbf{v}$, then $\mathbf{H}(p) = \mathbf{F}[\mathbf{H}/\mathbf{v}](p) = \mathbf{H}(q) = \mathbf{G}(q) = \mathbf{G}(p)$. If $\mathbf{F}(r_0^p) = \neg\mathbf{v}$, then $\mathbf{H}(p) = \mathbf{F}[\mathbf{H}/\mathbf{v}](p) = \neg\mathbf{H}(q) = \neg\mathbf{G}(q) = \mathbf{G}(p)$. $\qquad\square$

Finally, we show that $\mathbf{G}$ is ill–founded.

(v) There exists a function $f : \mathbb{N} \longrightarrow \text{dom}(\mathbf{G})$ such that $f(n) \sqsubset f(n+1)$, for every $n \in \mathbb{N}$.

Proof of (v). Recall that R is non–empty and that $\langle \rangle \notin R$. Let $r \in R$. Define $r^0 \overset{\text{DEF}}{=} \langle \rangle$ and $r^{n+1} \overset{\text{DEF}}{=} r \star r^n$, for each $n \in \mathbb{N}$. As $\langle \rangle \in S$, we have $r^n = r^n \star \langle \rangle \in A_n$ and $r^n \sqsubset r^{n+1}$, for every $n \in \mathbb{N}$. So, $\{r^n \mid n \in \mathbb{N}\} \subseteq A = \text{dom}(\mathbf{G})$. Thus, the function f given by: $f(n) \overset{\text{DEF}}{=} r^n$ for every $n \in \mathbb{N}$, has the required property. $\qquad\square$

The proof of Theorem 3.9 is now complete. $\qquad\square$

Example 3.10. Let us consider the recursive equations

$$(1) \ \mathbf{u} \overset{\text{REC}}{=} \vee[\mathbf{u}, \mathbf{u}] \ , \qquad (2) \ \mathbf{w} \overset{\text{REC}}{=} \wedge[\mathbf{w}, \mathbf{w}] \quad \text{and} \quad (3) \ \mathbf{v} \overset{\text{REC}}{=} \vee[\neg\mathbf{v}, \mathbf{v}] \ .$$

Let $\mathbf{U}$, $\mathbf{W}$ and $\mathbf{V}$ be the solutions of the equations (1), (2) and (3) respectively. We have $\text{dom}(\mathbf{U}) = \text{dom}(\mathbf{W}) = \text{dom}(\mathbf{V}) = A$, where $A \overset{\text{DEF}}{=} \{p \in \mathbb{N}^\star \mid p(k) \in \{0, 1\}, \text{ for all } k < \lg(p)\}$. Moreover, for $p \in A$, we have $\mathbf{U}(p) = \vee$, $\mathbf{W}(p) = \wedge$ and

$$\mathbf{V}(p) = \begin{cases} \vee & \text{if the cardinality of } \{k \mid p(k) = 0\} \text{ is even} \\ \wedge & \text{if the cardinality of } \{k \mid p(k) = 0\} \text{ is odd} \end{cases} .$$

In our setting, $\mathbf{v} \overset{\text{REC}}{=} \vee[\neg\mathbf{v}, \mathbf{v}]$ can be used to represent the equation "$X = X \to X$" which is a well–known example of equation of *mixed variance* in the literature of recursive types (see e.g., [7]). $\qquad\triangle$

3.3 Derivations

In this subsection, we introduce the notions of sequent and derivation. We also show some sequents which are derivable in our framework. In this work, we define sequents as special disjunctive formulas. We think that this choice is convenient, as it makes clearer the "duality" between sequents and environments (Definition 4.1(2)). Derivations are defined to be trees labeled by sequents and rules. Our

definition of derivation is actually very similar to that of *pre–proof* in [4] (very roughly, a pre–proof is a not necessarily well–founded derivation in a sequent calculus for classical logic with the ω–rule).

Definition 3.11 (Sequent). We call **sequent** any *disjunctive* formula $\mathbf{F}$ whose arity is $\{0, \ldots, n-1\}$, for some $n \in \mathbb{N}$. The set of all sequents is denoted by SEQ. $\triangle$

Notation 3.12. Let $\mathbf{S} = \vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}]$ be a sequent. Let $\mathbf{G}$ be a formula. We write $\vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}, \mathbf{G}]$, and sometimes also $\mathbf{S} \vee \mathbf{G}$, for the sequent $\mathbf{T}$ of arity $\{0, \ldots, n\}$ defined as follows: $\mathrm{dom}(\mathbf{T}) \overset{\text{DEF}}{=} \mathrm{dom}(\mathbf{S}) \cup \{\langle n \rangle \star q \mid q \in \mathrm{dom}(\mathbf{G})\}$ and

$$\mathbf{T}(p) \overset{\text{DEF}}{=} \begin{cases} \mathbf{S}(p) & \text{if } p \in \mathrm{dom}(\mathbf{S}) \\ \mathbf{G}(q) & \text{if } p = \langle n \rangle \star q, \end{cases} \quad \text{for } p \in \mathrm{dom}(\mathbf{T}) \ .$$

In particular, if $n = 0$ then $\mathbf{T} = \boldsymbol{f} \vee \mathbf{G} = \vee[\mathbf{G}]$ (recall that $\boldsymbol{f}$ is an abbreviation for $\vee[\,]$). $\triangle$

Definition 3.13 (Rule). A **rule** is either an *axiom rule* or a *disjunctive rule* or a *conjunctive rule*.

- An **axiom rule** is an ordered triple $(\mathbf{v}, k, \ell)$, where $\mathbf{v} \in \mathcal{V}$, $k \in \mathbb{N}$ and $\ell \in \mathbb{N}$.
- A **disjunctive rule** is an ordered triple $(\vee, k, i_0)$, where $k \in \mathbb{N}$ and $i_0 \in \mathbb{N}$.
- A **conjunctive rule** is a ordered pair $(\wedge, k)$, where $k \in \mathbb{N}$.

The set of all rules is denoted by RULES. $\triangle$

Definition 3.14 (Derivation). A **derivation** is a tree T labeled by $\text{SEQ} \times \text{RULES}$ such that for each $p \in \mathrm{dom}(T)$ one of the following conditions (ax), $(\vee)$ and $(\wedge)$ holds.

$$(\text{ax}): \begin{cases} \text{(i)} & T(p) = \left(\vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}], (\mathbf{v}, k, \ell) \right); \\ \text{(ii)} & k < n, \ell < n, \mathbf{F}_k = \mathbf{v} \text{ and } \mathbf{F}_\ell = \neg\mathbf{v}; \\ \text{(iii)} & p \text{ is a leaf of } T. \end{cases}$$

$$(\vee): \begin{cases} \text{(i)} & T(p) = \left(\vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}], (\vee, k, i_0) \right); \\ \text{(ii)} & k < n, \mathbf{F}_k = \vee_I \mathbf{G}_{\langle i \rangle} \text{ and } i_0 \in I; \\ \text{(iii)} & p \star \langle i \rangle \in \mathrm{dom}(T) \text{ if and only if } i = i_0; \\ \text{(iv)} & T_{\mathrm{L}}(p \star \langle i_0 \rangle) = \vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}, \mathbf{G}_{\langle i_0 \rangle}]. \end{cases}$$

$$(\wedge): \begin{cases} \text{(i)} & T(p) = \left(\vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}], (\wedge, k) \right); \\ \text{(ii)} & k < n \text{ and } \mathbf{F}_k = \wedge_I \mathbf{G}_{\langle i \rangle}; \\ \text{(iii)} & p \star \langle i \rangle \in \mathrm{dom}(T) \text{ if and only if } i \in I; \\ \text{(iv)} & T_{\mathrm{L}}(p \star \langle i \rangle) = \vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}, \mathbf{G}_{\langle i \rangle}], \text{ for every } i \in I. \end{cases}$$

In the sequel, we use $\pi, \rho, \sigma, \ldots$ to range over derivations. We say that π **is a derivation of S** if $\pi_{\mathrm{L}}(\langle \rangle) = \mathbf{S}$, and we say that **S is derivable** if there exists a derivation π of it. $\triangle$

Remark 3.15. Let π be a derivation. Let $\pi_{\mathrm{L}}(\langle \rangle) = \vee[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}]$, and let $\pi_{\mathrm{L}}(p) = \vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}]$, for some $p \in \mathrm{dom}(\pi)$.

(a) *The subformula property.* For every $k < n$, there exists $\ell < m$ such that $\mathbf{F}_k$ is a subformula of $\mathbf{G}_\ell$.

(b) *Leaves.* The position p is a leaf of π if and only if p is as in (ax) of Definition 3.14, or p is as in $(\wedge)$ of Definition 3.14 and $\mathbf{F}_k = \boldsymbol{t}$ (recall that $\boldsymbol{t}$ is an abbreviation for $\wedge[\,]$).

(c) *Rules.* The rule $\pi_{\mathrm{R}}(p)$ and the sequent $\pi_{\mathrm{L}}(p)$ completely determine the sequent $\pi_{\mathrm{L}}(q)$, for each q which immediately extends p in $\mathrm{dom}(\pi)$. In other words, if ρ and σ are two derivations of the same sequent $\mathbf{S}$, then $\mathrm{dom}(\rho) = \mathrm{dom}(\sigma)$ and $\rho_{\mathrm{R}}(p) = \sigma_{\mathrm{R}}(p)$ for all $p \in \mathrm{dom}(\rho)$ together imply $\rho = \sigma$. $\triangle$

Using a more traditional notation for derivations in the sequent calculus, conditions (ax), $(\vee)$ and $(\wedge)$ of Definition 3.14 can be respectively written as follows:

$$\frac{}{\vdash \Gamma, \mathbf{F}, \Delta, \neg\mathbf{F}, \Sigma} \ (\text{ax}) \qquad \frac{\vdash \Gamma, \vee_I \mathbf{F}_{\langle i \rangle}, \Delta, \mathbf{F}_{\langle i_0 \rangle}}{\vdash \Gamma, \vee_I \mathbf{F}_{\langle i \rangle}, \Delta} \ (\vee) \qquad \frac{\vdash \Gamma, \wedge_I \mathbf{F}_{\langle i \rangle}, \Delta, \mathbf{F}_{\langle i \rangle} \quad \ldots \text{ for all } i \in I}{\vdash \Gamma, \wedge_I \mathbf{F}_{\langle i \rangle}, \Delta} \ (\wedge)$$

where $\{\mathbf{F}, \neg\mathbf{F}\} = \{\mathbf{v}, \neg\mathbf{v}\}$ in (ax), $i_0 \in I$ in $(\vee)$, and in $(\wedge)$ the expression "$\ldots$ for all $i \in I$" means that there is one premise for each $i \in I$. In particular, if $I = \emptyset$, then there is no premise above the conclusion.

The *rules of inference* displayed above essentially correspond to the *normal rules* of Tait–calculus [8]. But in contrast with [8], in our setting ill–founded derivations are permitted. As a consequence, we have the following results.

Theorem 3.16.

(1) *Let* $\mathbf{S} = \vee[\mathbf{F}_0, \ldots, \mathbf{F}_{n-1}]$ *be a sequent. Suppose that for some $k < n$ either $\mathbf{F}_k$ is a disjunctive formula whose arity is not the empty set, or $\mathbf{F}_k$ is a conjunctive formula. Then, $\mathbf{S}$ is derivable.*

(2) *Let* $\mathbf{v} \overset{\mathrm{REC}}{=} \mathbf{F}$ *be a recursive equation, and let $\mathbf{G}$ be its unique solution. Then, the sequent $\vee[\mathbf{G}]$ is derivable.*

Proof. (1) Let $\mathbf{F}_k = \vee_I \mathbf{G}_{\langle i \rangle}$, for some $I \neq \emptyset$. Let $i_0 \in I$. We define a tree T labeled by SEQ $\times$ RULES as follows: $\mathrm{dom}(T) \overset{\mathrm{DEF}}{=} \{ p \in \mathbb{N}^\star \mid p(k) = i_0, \text{ for all } k < \lg(p) \}$ and

$$
T(p) \overset{\mathrm{DEF}}{=} \begin{cases} \big(\mathbf{S}, \, (\vee, k, i_0)\big) & \text{if } p = \langle\rangle \\ \big(T_{\mathsf{L}}(q) \vee \mathbf{G}_{\langle i_0 \rangle}, \, (\vee, k, i_0)\big) & \text{if } p = q \star \langle i_0 \rangle, \end{cases} \qquad \text{for } p \in \mathrm{dom}(T) \ .
$$

Suppose now that $\mathbf{F}_k = \wedge_I \mathbf{G}_{\langle i \rangle}$. We analogously define a tree U labeled by SEQ $\times$ RULES as follows: $\mathrm{dom}(U) \overset{\mathrm{DEF}}{=} \{ p \in \mathbb{N}^\star \mid p(k) \in I, \text{ for all } k < \lg(p) \}$ and

$$
U(p) \overset{\mathrm{DEF}}{=} \begin{cases} \big(\mathbf{S}, \, (\wedge, k)\big) & \text{if } p = \langle\rangle \\ \big(U_{\mathsf{L}}(q) \vee \mathbf{G}_{\langle i \rangle}, \, (\wedge, k)\big) & \text{if } p = q \star \langle i \rangle, \end{cases} \qquad \text{for } p \in \mathrm{dom}(U) \ .
$$

Since each $p \in \mathrm{dom}(T)$ (resp. $p \in \mathrm{dom}(U)$) satisfies condition $(\vee)$ (resp. $(\wedge)$) of Definition 3.14, T (resp. U) is a derivation of $\mathbf{S}$.

(2) By condition (R_2) of Definition 3.8, either $\mathbf{F} = \vee_I \mathbf{F}_{\langle i \rangle}$ or $\mathbf{F} = \wedge_I \mathbf{F}_{\langle i \rangle}$. Furthermore, by (R_3) we have that $I \neq \emptyset$, as $\mathbf{v}$ or $\neg\mathbf{v}$ (or both) must occur in $\mathbf{F}$. For $\diamond \in \{\vee, \wedge\}$, we have, by Proposition 3.7(2), that $\mathbf{G} = \mathbf{F}[\mathbf{G}/\mathbf{v}] = (\diamond_I \mathbf{F}_{\langle i \rangle})[\mathbf{G}/\mathbf{v}] = \diamond_I (\mathbf{F}_{\langle i \rangle}[\mathbf{G}/\mathbf{v}]) = \diamond_I \mathbf{G}_{\langle i \rangle}$. Hence, $\mathbf{G}$ is a compound formula whose arity is $I \neq \emptyset$. Then, we can apply (1) above to $\mathbf{S} = \vee[\mathbf{G}]$ and obtain the desired result. $\qquad\square$

Remark 3.17. Let $\mathbf{U}$, $\mathbf{W}$, and $\mathbf{V}$ be the formulas defined in Example 3.10.

(a) By Theorem 3.16(2), for each $\mathbf{M} \in \{\mathbf{U}, \mathbf{W}, \mathbf{V}\}$ the sequent $\vee[\mathbf{M}]$ is derivable. But, since the subformulas of $\mathbf{M}$ are neither atoms nor equal to $\boldsymbol{t}$, it follows from Remark 3.15(a)(b) that every derivation of $\vee[\mathbf{M}]$ has to be ill–founded.

(b) There are several sequents which are not derivable. For instance, $\boldsymbol{f}$, $\vee[\boldsymbol{f}], \vee[\boldsymbol{f}, \boldsymbol{f}], \ldots$ Also, for each atom $\mathbf{a}$ the sequents $\vee[\mathbf{a}]$, $\vee[\mathbf{a}, \mathbf{a}]$, $\vee[\mathbf{a}, \mathbf{a}, \mathbf{a}], \ldots$ are not derivable.

(c) Even though $\boldsymbol{f}$ is not derivable, our system is *inconsistent* in the sense that there is some formula $\mathbf{F}$ such that both $\vee[\mathbf{F}]$ and $\vee[\neg\mathbf{F}]$ are derivable. For, consider $\mathbf{U}$ and $\mathbf{W}$, and note that $\mathbf{U} = \neg\mathbf{W}$. By (a) above, $\vee[\mathbf{U}]$ and $\vee[\mathbf{W}]$ are derivable.

(d) The *cut–rule* is *not admissible* in our system, i.e., it is not true that

(CUT)   for every sequent $\mathbf{S}$ and every formula $\mathbf{F}$, if $\mathbf{S} \vee \mathbf{F}$ and $\mathbf{S} \vee \neg\mathbf{F}$ are derivable, then so is $\mathbf{S}$.

For, let $\mathbf{S}$ be $\boldsymbol{f}$. By (c) above, both $\boldsymbol{f} \vee \mathbf{U} = \vee[\mathbf{U}]$ and and $\boldsymbol{f} \vee \neg\mathbf{U} = \boldsymbol{f} \vee \mathbf{W} = \vee[\mathbf{W}]$ are derivable. But by (b) above, $\boldsymbol{f}$ is not derivable. $\qquad\triangle$

4   Interactive Semantics

We now present our interactive semantics for derivations in infinitary classical logic.

Let $\mathbf{S}$ be a sequent. Recall that in this paper we are not interested in studying the concept of derivability, i.e., "$\mathbf{S}$ is derivable." Rather, the concept that we want to analyze is "the test $\mathscr{T}$ comes from a derivation of $\mathbf{S}$", in the sense we now make clear. Before giving the formal definitions, we explain at an

informal level the three basic notions of our semantics: *test*, *environment* and *interaction tree*.

Tests.   In this paper, a test is nothing but a tree $\mathscr{T}$ labeled by RULES such that $\text{dom}(\mathscr{T}) = \mathbb{N}^\star$. This is actually the official definition of test (that we repeat, for convenience, in the next subsection). Tests have to be thought as "derivation candidates" for a derivation π of a sequent **S**. More precisely, let the *skeleton of the derivation π of* **S** be the tree T labeled by RULES whose domain is equal to $\text{dom}(\pi)$ and such that $T(p) = \pi_{\mathsf{R}}(p)$, for all $p \in \text{dom}(\pi)$. In other words, the skeleton is just the object obtained from a derivation by erasing all the sequent information. Then, we say that *the test $\mathscr{T}$ comes from the derivation π of* **S** (or that $\mathscr{T}$ is a "successful candidate") if the skeleton of π is "contained" into $\mathscr{T}$ (see Definition 4.4 for the precise definition). In general, it is not always the case that a test contains the skeleton of some derivation. In this paper, tests have to be understood as *untyped* objects in the sense that they are not necessarily related to formulas and sequents: it may be the case that a test comes from a single derivation, or several derivations, or no derivation at all. The semantics we define gives us a precise answer to the problem of determining when a test comes from a derivation or not.

Finally, we also point out that the definition of "to come from" is *syntactical*, as we use the syntactical notions of derivation and skeleton to define it.

Environments and interaction trees.   Technically speaking, an environment is just the negation $\neg$**S** of a sequent **S** (in particular, it is not a sequent). In our setting, tests and environments *interact*. To be more specific, a pair $(\mathscr{T}, \neg\mathbf{S})$, called *configuration*, uniquely determines a tree which is labeled by configurations of the previous kind, or by an error symbol $\Uparrow$. We call this tree the *interaction tree of* $(\mathscr{T}, \neg\mathbf{S})$. The construction of the interaction tree is *dynamical* in the sense that it can be seen as the (possibly infinitary) "unfolding" of the transition relation — of a suitable transition system — starting from $(\mathscr{T}, \neg\mathbf{S})$. The procedure which determines the interaction tree has also a very simple and natural *game interpretation*. Now, recall that the statement of the *interactive completeness theorem* is:

$\mathscr{T}$ comes from a derivation of **S**   if and only if   the interaction between $\mathscr{T}$ and $\neg$**S** does not produce errors.

We can now intuitively explain how errors come into play: if for some position p in the domain of the interaction tree of $(\mathscr{T}, \neg\mathbf{S})$ it is the case that the label is $\Uparrow$, then we say that *the interaction between the test $\mathscr{T}$ and the environment $\neg$**S** produces an error*. Observe that the notion of "to produce an error" is *semantical* in the sense that we use the interaction tree to determine the eventual presence of errors. In particular, we make no use of the syntactical notions of derivation and skeleton.

We now develop the technical part of this section. In Subsection 4.1 we give the formal definitions of test, environment and interaction tree. We also give a game theoretical interpretation of our interaction trees. In Subsection 4.2 we prove the interactive completeness theorem.

4.1   Interaction trees

Definition 4.1 (Test, environment, configuration)**.**

(1) A **test** is a tree T labeled by RULES such that $\text{dom}(T) = \mathbb{N}^\star$. The set of all tests is denoted by TESTS, and we use $\mathscr{T}, \mathscr{U}, \mathscr{W}, \ldots$ to denote its elements.

(2) We call **environment** any formula which is the negation of a sequent. That is, an environment is a *conjunctive* formula whose arity is $\{0, \ldots, m-1\}$, for some $m \in \mathbb{N}$. The set of all environments is denoted by ENV.

(3) We define the set CONF of **configurations** as

$$\text{CONF} \overset{\text{DEF}}{=} (\text{TESTS} \times \text{ENV}) \cup \{\Uparrow\},$$

where ⇑ (pronounced "error") is any object such that $\Uparrow \notin \text{TESTS} \times \text{ENV}$. In the sequel, we use the letters $c, d, e, \ldots$ to denote configurations. △

Notation 4.2. Let $\mathbf{E} = \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}]$ be an environment. Let $\mathbf{H}$ be a formula. In the sequel, we write $\wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}, \mathbf{H}]$, and when convenient also $\mathbf{E} \wedge \mathbf{H}$, for the environment $\neg(\vee[\neg\mathbf{G}_0, \ldots, \neg\mathbf{G}_{m-1}, \neg\mathbf{H}])$ of arity $\{0, \ldots, m\}$. Recall that the sequent $\vee[\neg\mathbf{G}_0, \ldots, \neg\mathbf{G}_{m-1}, \neg\mathbf{H}]$ is defined in Notation 3.12. △

Definition 4.3 (Interaction tree, production of errors)**.** Let c be a configuration. The **interaction tree of** c is the tree T labeled by CONF defined as follows.

(C$_1$) $\langle\rangle \in \text{dom}(T)$, and $T(\langle\rangle) \overset{\text{DEF}}{=} c$.

(C$_2$) Suppose that $p \in \text{dom}(T)$ has been defined. We define the immediate extensions of p in $\text{dom}(T)$ and their labels by cases as follows:

   (⇑$_1$) If $T(p) = \Uparrow$, then p is a leaf of T.
   (ax) If $T(p) = (\mathscr{T}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}])$, $\mathscr{T}(\langle\rangle) = (\mathbf{v}, k, \ell)$, $k < m$, $\ell < m$, $\mathbf{G}_k = \neg\mathbf{v}$ and $\mathbf{G}_\ell = \mathbf{v}$, then p is a leaf of T.
   (∨) If $T(p) = (\mathscr{T}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}])$, $\mathscr{T}(\langle\rangle) = (\vee, k, i_0)$, $k < m$, $\mathbf{G}_k = \wedge_I \mathbf{H}_{\langle i\rangle}$ and $i_0 \in I$, then $p \star \langle i\rangle \in \text{dom}(T)$ if and only if $i = i_0$, and $T(p \star \langle i_0\rangle) \overset{\text{DEF}}{=} (\mathscr{T}_{\langle i_0\rangle}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}, \mathbf{H}_{\langle i_0\rangle}])$.
   (∧) If $T(p) = (\mathscr{T}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}])$, $\mathscr{T}(\langle\rangle) = (\wedge, k)$, $k < m$ and $\mathbf{G}_k = \vee_I \mathbf{H}_{\langle i\rangle}$, then $p \star \langle i\rangle \in \text{dom}(T)$ if and only if $i \in I$, and $T(p \star \langle i\rangle) \overset{\text{DEF}}{=} (\mathscr{T}_{\langle i\rangle}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}, \mathbf{H}_{\langle i\rangle}])$ for each $i \in I$.
   (⇑$_2$) In all the other cases, $p \star \langle i\rangle \in \text{dom}(T)$ if and only if $i = 0$ and $T(p \star \langle 0\rangle) \overset{\text{DEF}}{=} \Uparrow$.

We denote the interaction tree of a configuration c as $\text{IT}(c)$.

Let $\mathscr{T}$ be a test, and let $\mathbf{E}$ be an environment. We say that the **the interaction between $\mathscr{T}$ and $\mathbf{E}$ does not produce errors** if

$$\text{there is } no \ p \in \text{dom}(\text{IT}((\mathscr{T}, \mathbf{E}))) \text{ such that } \text{IT}((\mathscr{T}, \mathbf{E}))(p) = \Uparrow \ . \qquad \triangle$$

We now discuss the previous bunch of definitions.

Global view of the interaction tree: interaction and transition systems. The procedure which determines the interaction tree has to be read as follows. First, we have $\langle\rangle \in \text{dom}(\text{IT}(c))$, by (C$_1$). From $\langle\rangle$ and its label, we calculate, by using the clauses in (C$_2$), all the positions of length one in $\text{dom}(\text{IT}(c))$. Now, we do the same thing for each position of length one in $\text{dom}(\text{IT}(c))$, and we obtain all the positions of length two in $\text{dom}(\text{IT}(c))$, and so on. This procedure might not stop and produce an ill–founded tree.

We call $\text{IT}(c)$ *interaction* tree because (in the case $c \neq \Uparrow$) when a test and an environment are combined into a configuration, they *interact*. Namely, they evolve into new configurations according to the clauses given in Definition 4.3. Furthermore, given a configuration c its interaction tree $\text{IT}(c)$ is defined *dynamically*: at each step one checks which clause of Definition 4.3 holds and then (possibly) proceeds to the next step. Indeed, one can recognize the computational structure of *transition system* here: CONF is the set of *states* and the *transition relation* $\rightsquigarrow$ is given according to the clauses in (C$_2$). Here, the configuration $\Uparrow$ represents a final state. Thus, the interaction tree of c can be seen as the (possibly infinitary) "unfolding" of the transition relation $\rightsquigarrow$ starting from the state c.

Local view of the interaction tree: a game theoretical interpretation. At first view, clauses (C$_2$)(ax), (C$_2$)(∨) and (C$_2$)(∧) of Definition 4.3 seem to be just the "dual" of conditions (ax), (∨) and (∧) of Definition 3.14 respectively. This is admittedly true, but notice that in Definition 3.14 we use (ax), (∨) and (∧) to *define* the notion of derivation, whereas here we use (C$_2$)(ax), (C$_2$)(∨) and (C$_2$)(∧) to (possibly) *find errors*. Furthermore, there is nothing which corresponds to clauses (C$_2$)(⇑$_1$) and (C$_2$)(⇑$_2$)

in Definition 3.14. On the other hand, these clauses are crucial to the aim of finding errors. Hence, even though derivations and interaction trees seem to be similar, they have different purposes.

We can now give a very simple and natural game interpretation to the objects introduced so far. Specifically, the interaction tree can be seen as a *debate* between two players: P (Proponent) and O (Opponent). The moves of the game are determined by the clauses given in Definition 4.3. In our game, P is always in charge of tests, whereas O is always in charge of environments. Given a configuration c, P (possibly) makes a question and O answers to the question by producing a set of configurations. Then, for each configuration produced by O, P (possibly) makes a question and O answers to the question by producing a set of configurations, and so on. The debate starts with a given configuration c different from $\Uparrow$. Then, for each configuration d given or produced so far, the two players behave as follows.

($\Uparrow$G) If $d = \Uparrow$, then P does not make any question and O does not produce any configuration.

Let $d = \big(\mathscr{T}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}]\big)$. Then, P asks $\mathscr{T}(\langle\rangle)$ and O answers as follows.

(axG) If P asks $\big(\mathbf{v}, k, \ell\big)$, then O checks the formulas $\mathbf{G}_k$ and $\mathbf{G}_\ell$. If $\mathbf{G}_k = \neg\mathbf{v}$ and $\mathbf{G}_\ell = \mathbf{v}$, then O does not produce any configuration. Otherwise, O produces $\Uparrow$.

($\vee$G) If P asks $\big(\vee, k, i_0\big)$, then O checks the formula $\mathbf{G}_k$. If $\mathbf{G}_k = \wedge_I \mathbf{H}_{\langle i\rangle}$ and $i_0 \in I$, then O produces $\big(\mathscr{T}_{\langle i_0\rangle}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}, \mathbf{H}_{\langle i_0\rangle}]\big)$. Otherwise, O produces $\Uparrow$.

($\wedge$G) If P asks $\big(\wedge, k\big)$, then O checks the formula $\mathbf{G}_k$. If $\mathbf{G}_k = \vee_I \mathbf{H}_{\langle i\rangle}$, then O produces $\big(\mathscr{T}_{\langle i\rangle}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}, \mathbf{H}_{\langle i\rangle}]\big)$, for each $i \in I$. Otherwise, O produces $\Uparrow$.

Of course, up to terminology and the rearrangement of clause ($\Uparrow_2$) inside the other clauses, this is just a more informal reformulation of the clauses given in Definition 4.3. We say that *O wins the debate starting from c* if at some stage it is able to exhibit an error, i.e., to produce the configuration $\Uparrow$. Otherwise, we say that *P wins the debate starting from c*. With this game theoretical intuition in mind, the statement of the interactive completeness theorem can be reformulated as follows: P wins the debate starting from $c = (\mathscr{T}, \neg\mathbf{S})$ just in case $\mathscr{T}$ comes from a derivation of $\mathbf{S}$.

Special cuts. A configuration of the form $\big(\mathscr{T}, \wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}]\big)$ can also be understood as a *special cut*, i.e., as a cut of the following special form (here we use a more traditional notation)

$$
\frac{\begin{array}{c}\vdots\pi\\\vdash \mathbf{F}_0, \ldots, \mathbf{F}_{m-1}\end{array} \qquad \begin{array}{c}\vdots z_{\mathbf{G}_0}\\\vdash \mathbf{G}_0\end{array} \quad \ldots \quad \begin{array}{c}\vdots z_{\mathbf{G}_{m-1}}\\\vdash \mathbf{G}_{m-1}\end{array}}{}\ \text{cut}
$$

where we simultaneously cut $\mathbf{F}_k$ with $\mathbf{G}_k$ for each $k < m$. Here, the derivation π (in the sense of the previous section) corresponds to the tests $\mathscr{T}$ and, for each formula $\mathbf{G}$ of our logic, the derivation $z_{\mathbf{G}}$ is given by: if $\mathbf{G}$ is an atom, then $\vdash \mathbf{G}$ has no premises; if $\mathbf{G} = \vee_I \mathbf{F}_{\langle i\rangle}$ or $\mathbf{G} = \wedge_I \mathbf{F}_{\langle i\rangle}$, then there is one premise $\vdash \mathbf{F}_{\langle i\rangle}$ for each $i \in I$ (thus, $z_{\mathbf{G}}$ is not a derivation in the sense of the previous section, and it is akin to a derivation of the subformula tree of $\mathbf{G}$). In addition, the formulas $\mathbf{F}_k$ and $\mathbf{G}_k$ need not be the negation of each other. Hence, *special cuts are not cuts in the ordinary sense*, and in particular, special cuts have nothing to do with the notion of cut discussed in the previous section.

Under this interpretation, an environment $\wedge[\mathbf{G}_0, \ldots, \mathbf{G}_{m-1}]$ can be naturally understood as a *conjunction* of formulas, as it represents a sequence $\vdash \mathbf{G}_0 \ldots \vdash \mathbf{G}_{m-1}$ of (unary) traditional sequents. With this picture in mind, the clauses given in Definition 4.3 transform a special cut into a set of special cuts or $\Uparrow$. So, they work as steps of reduction for a procedure of cut–elimination for special cuts. For instance, in the case of $(C_2)(\vee)$ we obtain after one step of reduction (here $\mathbf{F}_k = \vee_J \mathbf{L}_{\langle i\rangle}$ is any disjunctive formula such that $i_0 \in J$)

$$
\frac{\begin{array}{c}\vdots\pi_{\langle i_0\rangle}\\\vdash \mathbf{F}_0, \ldots, \vee_J \mathbf{L}_{\langle i\rangle}, \ldots, \mathbf{F}_{m-1}, \mathbf{L}_{\langle i_0\rangle}\end{array} \ \begin{array}{c}\vdots z_{\mathbf{G}_0}\\\vdash \mathbf{G}_0\end{array}\ \ldots\ \begin{array}{c}\vdots z_{\mathbf{G}_k}\\\vdash \wedge_I \mathbf{H}_{\langle i\rangle}\end{array}\ \ldots\ \begin{array}{c}\vdots z_{\mathbf{G}_{m-1}}\\\vdash \mathbf{G}_{m-1}\end{array}\ \begin{array}{c}\vdots z_{\mathbf{H}_{\langle i_0\rangle}}\\\vdash \mathbf{H}_{\langle i_0\rangle}\end{array}}{}\ \text{cut}
$$

which is again a special cut. The present discussion on special cuts is quite informal, but the point here is to develop another intuition about the concepts introduced so far.

4.2 Interactive completeness

In this section, we prove the interactive completeness theorem. First, we give the formal definition of "to come from", and then we move to the statement and the proof of the theorem.

Definition 4.4 (To come from). Let **S** be a sequent, and let π be a derivation of **S**. Let $\mathscr{T}$ be a test. We say that $\mathscr{T}$ **comes from the derivation π of S** if

$$\mathscr{T}(p) \;=\; \pi_{\mathsf{R}}(p) \;, \;\; \text{for every } p \in \text{dom}(\pi) \;. \hspace{2cm} \triangle$$

In other words, $\mathscr{T}$ comes from the derivation π of **S** if $\mathscr{T}$ "contains" the *skeleton* of π, in the sense of the beginning of this section.

Theorem 4.5 (Interactive completeness). *Let **S** be a sequent, and let $\mathscr{T}$ be a test. Then, the following claims are equivalent.*

*(1) The test $\mathscr{T}$ comes from a derivation of **S**.*

*(2) The interaction between $\mathscr{T}$ and $\neg$**S** does not produce errors.*

Proof. (1) implies (2) : Let π be a derivation of **S**, and assume that $\mathscr{T}$ comes from π. Let c be $(\mathscr{T}, \neg\mathbf{S})$. To show that the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ does not produce errors, we now prove that every $p \in \mathbb{N}^\star$ satisfies one of the following (mutually exclusive) conditions:

(P_1) : $p \in \text{dom}(\pi)$, $p \in \text{dom}(\text{IT}(c))$ and $\text{IT}(c)(p) = (\mathscr{T}_p, \neg\pi_{\mathsf{L}}(p))$.

(P_2) : $p \notin \text{dom}(\pi)$ and $p \notin \text{dom}(\text{IT}(c))$.

We proceed by induction on the length of p. The position $\langle\rangle$ satisfies (P_1), as $\langle\rangle \in \text{dom}(\pi)$, $\langle\rangle \in \text{dom}(\text{IT}(c))$ and $\text{IT}(c)(\langle\rangle) = c = (\mathscr{T}, \neg\mathbf{S}) = (\mathscr{T}_{\langle\rangle}, \neg\pi_{\mathsf{L}}(\langle\rangle))$. Consider now an arbitrary position $p \in \mathbb{N}^\star$. Assume that p satisfies (P_1) or (P_2). We show that for each $j \in \mathbb{N}$, the position $p \star \langle j \rangle$ satisfies (P_1) or (P_2). If p satisfies (P_2), then $p \star \langle j \rangle \notin \text{dom}(\pi)$ and $p \star \langle j \rangle \notin \text{dom}(\text{IT}(c))$, as π and $\text{IT}(c)$ are labeled trees. Hence, $p \star \langle j \rangle$ satisfies (P_2). Otherwise, p satisfies (P_1). Since $p \in \text{dom}(\pi)$, we have $\mathscr{T}(p) = \pi_{\mathsf{R}}(p)$, as $\mathscr{T}$ comes from π by assumption. Moreover, since π is a derivation, p satisfies one of conditions (ax), ($\vee$) and ($\wedge$) of Definition 3.14. Let $\pi_{\mathsf{L}}(p)$ be $\vee[\mathbf{G}_0, \dots, \mathbf{G}_{n-1}]$, so that $\neg\pi_{\mathsf{L}}(p) = \wedge[\neg\mathbf{G}_0, \dots, \neg\mathbf{G}_{n-1}]$. We now consider the following subcases.

   (i) If $\mathscr{T}(p) = (\mathbf{v}, k, \ell)$, then p satisfies (ax) of Definition 3.14. Hence, $k < n$, $\ell < n$, $\mathbf{G}_k = \mathbf{v}$, $\mathbf{G}_\ell = \neg\mathbf{v}$ and p is a leaf of π. Since $\neg\mathbf{G}_k = \neg\mathbf{v}$ and $\neg\mathbf{G}_\ell = \mathbf{v}$, $\text{IT}(c)(p)$ is as in (C_2)(ax) of Definition 4.3. Hence, p is a leaf of $\text{IT}(c)$. In particular, $p \star \langle j \rangle$ satisfies (P_2).

   (ii) If $\mathscr{T}(p) = (\vee, k, i_0)$, then p satisfies ($\vee$) of Definition 3.14. This means that $k < n$, $\mathbf{G}_k = \vee_I \mathbf{H}_{\langle i \rangle}$, $i_0 \in I$, only $p \star \langle i_0 \rangle$ immediately extends p in $\text{dom}(\pi)$, and $\pi_{\mathsf{L}}(p \star \langle i_0 \rangle) = \pi_{\mathsf{L}}(p) \vee \mathbf{H}_{\langle i_0 \rangle}$. Since $\neg\mathbf{G}_k = \wedge_I \neg\mathbf{H}_{\langle i \rangle}$ and $i_0 \in I$, $\text{IT}(c)(p)$ is as in $(C_2)(\vee)$ of Definition 4.3. Hence, only $p \star \langle i_0 \rangle$ immediately extends $p \in \text{dom}(\text{IT}(c))$ and $\text{IT}(c)(p \star \langle i_0 \rangle) = ((\mathscr{T}_p)_{\langle i_0 \rangle}, \neg\pi_{\mathsf{L}}(p) \wedge \neg\mathbf{H}_{\langle i_0 \rangle}) = (\mathscr{T}_{p \star \langle i_0 \rangle}, \neg\pi_{\mathsf{L}}(p \star \langle i_0 \rangle))$. So, $p \star \langle j \rangle$ satisfies (P_1) if $j = i_0$, and $p \star \langle j \rangle$ satisfies (P_2) otherwise.

   (iii) Finally, if $\mathscr{T}(p) = (\wedge, k)$, then p satisfies ($\wedge$) of Definition 3.14. So, $k < n$, $\mathbf{G}_k = \wedge_I \mathbf{H}_{\langle i \rangle}$, $p \star \langle i \rangle$ immediately extends p in $\text{dom}(\pi)$ if and only if $i \in I$, and $\pi_{\mathsf{L}}(p \star \langle i \rangle) = \pi_{\mathsf{L}}(p) \vee \mathbf{H}_{\langle i \rangle}$ for each $i \in I$. Since $\neg\mathbf{G}_k = \vee_I \neg\mathbf{H}_{\langle i \rangle}$, $\text{IT}(c)(p)$ is as in $(C_2)(\wedge)$ of Definition 4.3. Thus, $p \star \langle i \rangle$ immediately extends p in $\text{dom}(\text{IT}(c))$ if and only if $i \in I$ and $\text{IT}(c)(p \star \langle i \rangle) = ((\mathscr{T}_p)_{\langle i \rangle}, \neg\pi_{\mathsf{L}}(p) \wedge \neg\mathbf{H}_{\langle i \rangle}) = (\mathscr{T}_{p \star \langle i \rangle}, \neg\pi_{\mathsf{L}}(p \star \langle i \rangle))$, for every $i \in I$. Hence, $p \star \langle j \rangle$ satisfies (P_1) if $j \in I$, and $p \star \langle j \rangle$ satisfies (P_2) otherwise.

This proves that each $p \in \mathbb{N}^\star$ satisfies either (P$_1$) or (P$_2$). From this fact, it follows that $\mathrm{IT}(c)(p) = \big(\mathscr{T}_p, \neg\pi_{\mathsf{L}}(p)\big)$ for all $p \in \mathrm{dom}(\mathrm{IT}(c))$. In particular, there is no $p \in \mathrm{dom}(\mathrm{IT}(c))$ such that $\mathrm{IT}(c)(p) = \Uparrow$, i.e., the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ does not produce errors.

(2) implies (1) : Let c be $\big(\mathscr{T}, \neg\mathbf{S}\big)$ and assume that (2) holds. This means that each position $p \in \mathrm{dom}(\mathrm{IT}(c))$ is labeled by a member of TESTS $\times$ ENV. In such a situation, it immediately follows from the definition of interaction tree that for each $p \in \mathrm{dom}(\mathrm{IT}(c))$ we have $\mathrm{IT}(c)(p) = \big(\mathscr{T}_p, \mathbf{E}\big)$, for some $\mathbf{E} \in$ ENV. We define a tree T labeled SEQ $\times$ RULES as follows: $\mathrm{dom}(T) \overset{\text{DEF}}{=} \mathrm{dom}(\mathrm{IT}(c))$ and

$$T(p) \overset{\text{DEF}}{=} \big(\neg\mathbf{E}, \mathscr{T}(p)\big) \ , \quad \text{for } p \in \mathrm{dom}(T) \text{ with } \mathrm{IT}(c)(p) = \big(\mathscr{T}_p, \mathbf{E}\big) \ .$$

We now show that T is a derivation. To do this, we need to check that for each $p \in \mathrm{dom}(T)$ one of conditions (ax), ($\vee$) and ($\wedge$) of Definition 3.14 holds. Let $\mathrm{IT}(c)(p) = \big(\mathscr{T}_p, \mathbf{E}\big)$ and let $\mathbf{E}$ be $\wedge[\mathbf{F}_0, \ldots, \mathbf{F}_{m-1}]$, so that $\neg\mathbf{E} = \vee[\neg\mathbf{F}_0, \ldots, \neg\mathbf{F}_{m-1}]$. There are the following cases to consider.

(1) Suppose that $\mathscr{T}(p) = \big(\mathbf{v}, k, \ell\big)$. Since the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ does not produce errors, the configuration $\mathrm{IT}(c)(p)$ is as in (C$_2$)(ax) of Definition 4.3. This means that $k < m$, $\ell < m$, $\mathbf{F}_k = \neg\mathbf{v}$, $\mathbf{F}_\ell = \mathbf{v}$ and p is a leaf of $\mathrm{IT}(c)$. Since $\neg\mathbf{F}_k = \mathbf{v}$ and $\neg\mathbf{F}_\ell = \neg\mathbf{v}$, and since p is a leaf of T, the position p satisfies (ax) of Definition 3.14.

(2) Suppose that $\mathscr{T}(p) = \big(\vee, k, i_0\big)$. Since the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ does not produce errors, the situation is as in (C$_2$)($\vee$) of Definition 4.3. Hence, $k < m$, $\mathbf{F}_k = \wedge_I \mathbf{H}_{\langle i \rangle}$, $i_0 \in I$, only $p \star \langle i_0 \rangle$ immediately extends p in $\mathrm{dom}(\mathrm{IT}(c))$, and $\mathrm{IT}(c)(p \star \langle i_0 \rangle) = \big(\mathscr{T}_{p \star \langle i_0 \rangle}, \mathbf{E} \wedge \mathbf{H}_{\langle i_0 \rangle}\big)$. Since $\neg\mathbf{F}_k = \vee_I \neg\mathbf{H}_{\langle i \rangle}$, $i_0 \in I$, only $p \star \langle i_0 \rangle$ immediately extends p in $\mathrm{dom}(T)$ and $T_{\mathsf{L}}(p \star \langle i_0 \rangle) = \neg(\mathbf{E} \wedge \mathbf{H}_{\langle i_0 \rangle}) = \neg\mathbf{E} \vee \neg\mathbf{H}_{\langle i_0 \rangle}$, we conclude that ($\vee$) of Definition 3.14 holds for the position p.

(3) Finally, suppose that $\mathscr{T}(p) = \big(\wedge, k\big)$. Since the interaction between $\mathscr{T}$ and $\neg\mathbf{S}$ does not produce errors, the configuration $\mathrm{IT}(c)(p)$ is as in (C$_2$)($\wedge$) of Definition 4.3. Thus, $k < m$, $\mathbf{F}_k = \vee_I \mathbf{H}_{\langle i \rangle}$, $p \star \langle i \rangle$ immediately extends p in $\mathrm{dom}(\mathrm{IT}(c))$ if and only $i \in I$, and $\mathrm{IT}(c)(p \star \langle i \rangle) = \big(\mathscr{T}_{p \star \langle i \rangle}, \mathbf{E} \wedge \mathbf{H}_{\langle i \rangle}\big)$ for every $i \in I$. Since $\neg\mathbf{F}_k = \wedge_I \neg\mathbf{H}_{\langle i \rangle}$, $p \star \langle i \rangle$ immediately extends p in $\mathrm{dom}(T)$ if only if $i \in I$, $T_{\mathsf{L}}(p \star \langle i \rangle) = \neg(\mathbf{E} \wedge \mathbf{H}_{\langle i \rangle}) = \neg\mathbf{E} \vee \neg\mathbf{H}_{\langle i \rangle}$ for every $i \in I$, we conclude that the position p satisfies ($\wedge$) of Definition 3.14.

Hence, T is a derivation of $\mathbf{S}$. By construction, $T_{\mathsf{R}}(p) = \mathscr{T}(p)$ for all $p \in \mathrm{dom}(T)$. Therefore, the test $\mathscr{T}$ comes from a derivation of $\mathbf{S}$, namely T. $\square$

References

[1] M. Basaldella & K. Terui (2010): *Infinitary Completeness in Ludics*. In: *Proceedings of LICS 2010*, pp. 294–303, doi:10.1109/LICS.2010.47.

[2] M. Coppo (1998): *Recursive Types: the syntactic and semantic approaches*. In: *Type Theory and its Application to Computer Systems*, *RIMS Lecture Notes* 1023, RIMS, Kyoto University, pp. 16–41. Available at `http://hdl.handle.net/2433/61723`.

[3] B. Courcelle (1983): *Fundamental properties of infinite trees*. *Theor. Comput. Sci.* 25, pp. 95–169, doi:10.1016/0304-3975(83)90059-2.

[4] J.-Y. Girard (1987): *Proof Theory and Logical Complexity: Volume I*. Bibliopolis, Napoli.

[5] J.-Y. Girard (2001): *Locus Solum: From the rules of logic to the logic of rules*. *Mathematical Structures in Computer Science* 11(3), pp. 301–506, doi:10.1017/S096012950100336X.

[6] Z. Manna (1974): *Mathematical Theory of Computation*. McGraw–Hill, New York.

[7] P.-A. Melliès & J. Vouillon (2005): *Recursive polymorphic types and parametricity in an operational framework*. In: *Proceedings of LICS 2005*, pp. 82–91, doi:10.1109/LICS.2005.42.

[8] W.W. Tait (1968): *Normal derivability in classical logic*. In: *The Syntax and Semantics of Infinitary Languages*, chapter 12, *LNM* 72, Springer–Verlag, pp. 204–236, doi:10.1007/BFb0079691.