Generalizing Inference Systems by Coaxioms

Davide Ancona$^{(\boxtimes)}$, Francesco Dagnino, and Elena Zucca

DIBRIS, Universitá di Genova, Genoa, Italy
{davide.ancona,elena.zucca}@unige.it, fra.dagn@gmail.com

Abstract. We introduce a generalized notion of inference system to support structural recursion on non well-founded datatypes. Besides axioms and inference rules with the usual meaning, a generalized inference system allows *coaxioms*, which are, intuitively, axioms which can only be applied "at infinite depth" in a proof tree. This notion nicely subsumes standard inference systems and their inductive and coinductive interpretation, while providing more flexibility. Indeed, the classical results on the existence and constructive characterization of least and greatest fixed points can be extended to our generalized framework, interpreting recursive definitions as fixed points which are not necessarily the least, nor the greatest one. This allows formal reasoning in cases where the inductive and coinductive interpretation do not provide the intended meaning, or are mixed together.

1 Introduction

Recently several approaches [5,10,11,18,19,25,32] have been proposed to program with coinductive (coalgebraic) datatypes to support corecursion, that is, the ability of defining predicates or functions by structural recursion on non-well-founded datatypes. Such solutions are generally characterized by a strong dichotomy between inductive and coinductive definitions, the former being based on the notion of least fixed point, and the latter on that of greatest fixed point. Moreover, some proposals provide language abstractions to allow the programmer to interpret corecursive definitions not in the standard coinductive way. As a consequence, formal reasoning about programs that exploit such abstractions cannot be based on usual proof principles.

In this paper, we introduce a framework for interpreting recursive definitions as fixed points which are not necessarily the least, nor the greatest one. This allows formal reasoning in cases where the inductive and coinductive interpretation do not provide the intended meaning, or are mixed together.

To introduce the idea, let us consider the following recursive definitions of functions on lists of integers, with the meaning suggested by the name.

Special thanks go to all anonymous reviewers, who helped us improve this paper, and to Bart Jacobs for an enlightening discussion with him on the use of coaxioms for modeling divergence with big-step semantics.

H. Yang (Ed.): ESOP 2017, LNCS 10201, pp. 29–55, 2017.
DOI: 10.1007/978-3-662-54434-1_2

```
let rec allPos = function [] -> true | x::l -> x >0 &&
allPos l
let rec member y =
   function [] -> false | x::l -> x==y||member y l
let rec elems = function
   [] -> [] |
   x::l -> let xs = elems l in if member x xs then xs else
   x::xs
let rec maxElem = function [x] -> x | x::l -> max x
(maxElem l)
```

These definitions are written above in a widely-known programming language syntax (OCaml) for concreteness, but this is not relevant here: for such first-order functions, in most programming languages we can write analogous recursive definitions, and they are usually interpreted *inductively*. This means that, turning, more abstractly, such recursive definitions into meta-rules of an inference system, they are interpreted as the set of judgments which have a finite proof tree. For instance, the meta-rules for the judgment $allPos(l, b)$ are as follows:

$$\frac{}{allPos(\Lambda, T)} \quad \frac{}{allPos(x{:}l, F)}\, x \le 0 \quad \frac{allPos(l, b)}{allPos(x{:}l, b)}\, x > 0$$

where Λ and : denote the empty list, and the list constructor, respectively, and T and F denote the boolean values. This interpretation works perfectly well on finite lists. However, with the inductive interpretation the above functions may happen to be undefined on infinite lists. For instance, the judgment $allPos(l, b)$ obviously has no finite proof tree if l is an infinite list of positives.

Indeed, to support structural recursion on non-well-founded structures, such as infinite lists or graphs, we typically have to use *coinduction*. The coinductive interpretation of an inference system is the set of judgments which have a (finite or infinite) proof tree.

In some cases, the coinductive interpretation actually yields the intended meaning. For instance, taking a slightly different version of `allPos` as a unary predicate $allPos(l)$, as it would be expressed in a logic program:

$$\frac{}{allPos(\Lambda)} \quad \frac{allPos(l)}{allPos(x : l)}\, x > 0$$

it is easy to see that with the coinductive interpretation we obtain the intended meaning on infinite lists as well, since we get an infinite proof tree if and only if all the elements in the list are positive. Indeed, this interpretation has been fruitfully used in coinductive logic programming (coLP) [3,31–33].

However, considering instead the previous relation $allPos(l, b)$, the coinductive interpretation fails to be a function, since for infinite lists of positives both the judgment $allPos(l, T)$ and $allPos(l, F)$ can be proved. Moreover, if we consider the predicate corresponding to the boolean function `member`:

$$\frac{}{member(x, x : l)} \quad \frac{member(x, l)}{member(x, y : l)}\, x \ne y$$

then the correct interpretation is the inductive one. Indeed, the coinductive interpretation contains all judgments $\mathit{member}(x, l)$ where l is an infinite list. Finally, for the predicates corresponding to the other example functions, which do not return a boolean, neither the inductive nor the coinductive interpretation yields the intended semantics. In particular, the coinductive interpretation contains too many elements. For instance, taking l the infinite list of 1s, by coinductively interpreting *elems* and *maxElem* we get, together with the correct judgments, also wrong ones, as will be formally shown in the following section.

All these examples suggest the idea that we should be able to "filter out" in some way the (infinite) proof trees corresponding to the coinductive interpretation, keeping only some of them. We make this possible by introducing *coaxioms*. A coaxiom is, intuitively, an axiom which can only be applied "at infinite depth" in a proof tree. An inference system interpreted inductively corresponds to a generalized inference system with no coaxioms, while an inference system interpreted coinductively corresponds to a generalized inference system where there is a coaxiom for each judgment.

From the model-theoretic point of view, coaxioms allow the programmer to choose the desired fixed point for a recursive definition, by selecting also fixed points which are neither the least, nor the greatest one. For instance, in the inference system for $\mathit{allPos}(l, b)$, the intended meaning is the set of judgments $\mathit{allPos}(l, b)$ where b is true if and only if the (finite or infinite) list l contains only positives. This set *is a fixed point* which lies between the least, which is undefined on infinite lists of positives, and the greatest, which returns both boolean values, hence is undetermined, on such lists.

Coaxioms are partly inspired by an extension of coLP and coinductive SLD resolution (coSLD) [31–33] with `finally` clauses [5], to allow more flexible interpretations of corecursive definitions of predicates, and by a related proposal in the context of object-oriented programming [10,11]. In this paper we take a more abstract and general approach and provide a framework for interpreting corecursive definitions in a flexible way and to formally reason on their correctness.

The rest of the paper is organized as follows: in Sect. 2 we introduce the notion of generalized inference system with coaxioms, and show how to express the previous examples and others. In Sect. 3 we formally define the fixed point semantics of inference systems with coaxioms in the more general setting of complete lattices. In Sect. 4 we discuss the equivalent semantics based on the proof-theoretic approach, and in Sect. 5 we illustrate the related proof techniques on some of the examples. In Sect. 6 we show some more involved examples and discuss some subtleties, Sect. 7 surveys related work, and finally in Sect. 8 we summarize our contribution and discuss further work. A prototype meta-interpreter[1] has been developed to test the examples provided in Sects. 2 and 6.

[1] Available at http://www.disi.unige.it/person/AnconaD/Software/esop17artifact.zip.

2 Inference Systems with Coaxioms

We recall some standard notions about inference systems [1,23].

Assume in the following a set $\mathcal{U}$ called the *universe*, whose elements are called *judgments*.

An *inference system* $\mathcal{I}$ consists of a set of *inference rules*, which are pairs $\frac{Pr}{c}$, with $Pr \subseteq \mathcal{U}$ the set of *premises*, $c \in \mathcal{U}$ the *consequence*.

The intuitive interpretation of a rule is that if the premises Pr hold then the consequence c should hold as well. In particular, an *axiom* is (the consequence of) a rule with empty set of premises, which necessarily holds.

The *(one step) inference* operator $F_{\mathcal{I}} : \wp(\mathcal{U}) \rightarrow \wp(\mathcal{U})$ associated with an inference system $\mathcal{I}$ is defined by:

$$F_{\mathcal{I}}(S) = \{c \mid Pr \subseteq S, \frac{Pr}{c} \in \mathcal{I}\}$$

That is, $F_{\mathcal{I}}(S)$ is the set of judgments that can be inferred (in one step) from the judgments in S using the inference rules. Note that this set always includes axioms.

A set S is *closed* if $F_{\mathcal{I}}(S) \subseteq S$, and *consistent* if $S \subseteq F_{\mathcal{I}}(S)$. That is, no new judgments can be inferred from a closed set, and all judgments in a consistent set can be inferred from the set itself.

The *inductive interpretation* of $\mathcal{I}$, denoted $Ind(\mathcal{I})$, is the smallest closed set, that is, the intersection of all closed sets, and the *coinductive interpretation* of $\mathcal{I}$, denoted $CoInd(\mathcal{I})$, is the largest consistent set, that is, the union of all consistent sets. Both interpretations are well-defined and can be equivalently expressed as the least (respectively, greatest) fixed point of the inference operator. Moreover, under continuity hypotheses on $F_{\mathcal{I}}$, they can be computed as follows:

$$\begin{aligned} &Ind(\mathcal{I}) = \bigcup\{F^n_{\mathcal{I}}(\emptyset) \mid n \geq 0\} \\ &CoInd(\mathcal{I}) = \bigcap\{F^n_{\mathcal{I}}(\mathcal{U}) \mid n \geq 0\} \end{aligned}$$

The inductive and coinductive interpretation can also be characterized in terms of proof trees. That is, defining a proof tree as a tree whose nodes are (labeled with) judgments in $\mathcal{U}$, and there is a node c with set of children Pr only if there exists a rule $\frac{Pr}{c}$, it can be shown [23] that $Ind(\mathcal{I})$ and $CoInd(\mathcal{I})$ are the sets of judgments which are the root of a finite[2] and an arbitrary (finite or infinite) proof tree, respectively.

We introduce now our generalization.

An *inference system with coaxioms* is a pair $(\mathcal{I}, \gamma)$ consisting of an inference system $\mathcal{I}$ and a set of *coaxioms* γ, with $\gamma \subseteq \mathcal{U}$. A coaxiom c will be written $\frac{\bullet}{c}$, very much like an axiom, and analogously to an axiom it can be used as an initial assumption to derive other judgments. However, coaxioms will be used in a special way, explained in the following.

[2] Under the common assumption that the set of premises of all the rules are finite, otherwise we should say a finite depth tree.

To illustrate the notion, we will consider an introductory example which computes the judgment $n \xrightarrow{\star} \mathcal{N}$ meaning that $\mathcal{N}$ is the set of nodes reachable from a node n of a given graph. Let us represent a graph by its set of nodes V and a function *adj* which returns all the adjacent nodes. As usual, sets of rules can be expressed by a *metarule* with side conditions, and the same can be done for sets of coaxioms.

$$\frac{n_1 \xrightarrow{\star} \mathcal{N}_1 \ \ldots \ n_k \xrightarrow{\star} \mathcal{N}_k}{n \xrightarrow{\star} \{n\} \cup \mathcal{N}_1 \cup \ldots \cup \mathcal{N}_k}\, adj(n) = \{n_1, \ldots, n_k\} \qquad \frac{\bullet}{n \xrightarrow{\star} \emptyset}\, n \in V$$

For instance, in the case of a graph with nodes a, b, c, with an arc from a into b and conversely, and c isolated, we would get the following metarules and coaxioms:

$$\frac{b \xrightarrow{\star} \mathcal{N}}{a \xrightarrow{\star} \{a\} \cup \mathcal{N}} \quad \frac{a \xrightarrow{\star} \mathcal{N}}{b \xrightarrow{\star} \{b\} \cup \mathcal{N}} \quad \frac{}{c \xrightarrow{\star} \{c\}} \quad \frac{\bullet}{a \xrightarrow{\star} \emptyset} \quad \frac{\bullet}{b \xrightarrow{\star} \emptyset} \quad \frac{\bullet}{c \xrightarrow{\star} \emptyset}$$

If we interpret the metarules inductively (excluding the coaxioms), then we get only the judgment $c \xrightarrow{\star} \{c\}$. In other words, a visit computing $n \xrightarrow{\star} \mathcal{N}$, like other judgments on graphs, should mark already encountered nodes to avoid non termination, since the graph structure is not well-founded. On the other hand, if we interpret the metarules coinductively (excluding again the coaxioms), then we get the correct judgments $a \xrightarrow{\star} \{a, b\}$ and $b \xrightarrow{\star} \{a, b\}$, but we also get the wrong judgments $a \xrightarrow{\star} \{a, b, c\}$ and $b \xrightarrow{\star} \{a, b, c\}$.

We define a different interpretation, called *interpretation generated by the coaxioms* and denoted $Gen(\mathcal{I}, \gamma)$, which takes into account the coaxioms in the following way.

1. First, we take the smallest closed superset of the set of coaxioms. In other words, we consider the inference system $\mathcal{I}_{\sqcup\gamma}$ obtained enriching $\mathcal{I}$ by judgments in γ considered as axioms, and we take its inductive interpretation $Ind(\mathcal{I}_{\sqcup\gamma})$.
2. Then, we take the largest consistent subset of $Ind(\mathcal{I}_{\sqcup\gamma})$. In other words, we take the coinductive interpretation of the inference system obtained from $\mathcal{I}$ by keeping only rules with consequence in $Ind(\mathcal{I}_{\sqcup\gamma})$, that is, we define
   $Gen(\mathcal{I}, \gamma) = CoInd(\mathcal{I}_{\sqcap Ind(\mathcal{I}_{\sqcup\gamma})})$

where $\mathcal{I}_{\sqcap S}$, with $\mathcal{I}$ inference system and $S \subseteq \mathcal{U}$, denotes the inference system obtained from $\mathcal{I}$ by keeping only rules with consequence in S.

In the example, in the first phase we obtain the following judgments (each line corresponds to an iteration of the inference operator):

$$\begin{array}{l}
a \xrightarrow{\star} \emptyset,\ b \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \{c\} \\
a \xrightarrow{\star} \emptyset,\ b \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \{c\},\ a \xrightarrow{\star} \{a\},\ b \xrightarrow{\star} \{b\} \\
a \xrightarrow{\star} \emptyset,\ b \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \emptyset,\ c \xrightarrow{\star} \{c\},\ a \xrightarrow{\star} \{a\},\ b \xrightarrow{\star} \{b\},\ a \xrightarrow{\star} \{a, b\}, b \xrightarrow{\star} \{a, b\}
\end{array}$$

The last set is closed, hence it is $Ind(\mathcal{I}_{\sqcup\gamma})$.

In the second phase, each iteration of the inference operator removes judgments which cannot be inferred from the previous step, that is, we get:

$c\xrightarrow{\star}\{c\}$, $a\xrightarrow{\star}\{a\}$, $b\xrightarrow{\star}\{b\}$, $a\xrightarrow{\star}\{a,b\}, b\xrightarrow{\star}\{a,b\}$
$c\xrightarrow{\star}\{c\}$, $a\xrightarrow{\star}\{a,b\}$, $b\xrightarrow{\star}\{a,b\}$

This last set is consistent, hence it is $\mathit{Gen}(\mathcal{I},\gamma)$, and it is indeed the expected result.

Note that the inductive and coinductive interpretation can be obtained as special cases of the interpretation generated by coaxioms of an inference system, notably:

– the inductive interpretation when the set of coaxioms is empty
– the coinductive interpretation when the set of coaxioms is the universe.

In terms of proof trees, judgments in $\mathit{Gen}(\mathcal{I},\gamma)$ are those which have an arbitrary (finite or infinite) proof tree t in the inference system $\mathcal{I}$, whose nodes all have a finite proof tree in $\mathcal{I}_{\sqcup\gamma}$. Note that for nodes in t which are roots of a finite subtree this always holds (a finite proof tree in $\mathcal{I}$ is a finite proof tree in $\mathcal{I}_{\sqcup\gamma}$ as well), hence the condition is only significant for nodes which are roots of an infinite path in the proof tree.

For instance, in the example, the judgment $a\xrightarrow{\star}\{a,b\}$ has an infinite proof tree in $\mathcal{I}$ where each node has a finite proof tree in $\mathcal{I}_{\sqcup\gamma}$, as shown below.

$$\dfrac{\dfrac{\dfrac{\dots}{a\xrightarrow{\star}\{a,b\}}}{b\xrightarrow{\star}\{a,b\}}}{a\xrightarrow{\star}\{a,b\}} \qquad \dfrac{\dfrac{\dfrac{}{a\xrightarrow{\star}\emptyset}}{b\xrightarrow{\star}\{b\}}}{a\xrightarrow{\star}\{a,b\}} \qquad \dfrac{\dfrac{\dfrac{}{b\xrightarrow{\star}\emptyset}}{a\xrightarrow{\star}\{a\}}}{b\xrightarrow{\star}\{a,b\}}$$

Moreover, there is another important property which will be proved in Sect. 4: if a judgment belongs to $\mathit{Gen}(\mathcal{I},\gamma)$, then, for all $n \geq 0$, it has a proof tree in the inference system $\mathcal{I}_{\sqcup\gamma}$ where coaxioms can only be used at depth greater than n.

For instance, in the example, it is easy to see that, for any n, we can obtain a finite proof tree for the judgment $a\xrightarrow{\star}\{a,b\}$ in $\mathcal{I}_{\sqcup\gamma}$ where coaxioms are used at depth greater than n, as shown below.

$$\dfrac{\dfrac{\dfrac{}{a\xrightarrow{\star}\emptyset}}{b\xrightarrow{\star}\{b\}}}{a\xrightarrow{\star}\{a,b\}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{b\xrightarrow{\star}\emptyset}}{a\xrightarrow{\star}\{a\}}}{b\xrightarrow{\star}\{a,b\}}}{a\xrightarrow{\star}\{a,b\}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{a\xrightarrow{\star}\emptyset}}{b\xrightarrow{\star}\{b\}}}{a\xrightarrow{\star}\{a,b\}}}{b\xrightarrow{\star}\{a,b\}}}{a\xrightarrow{\star}\{a,b\}} \qquad \dots$$

This last property motivates the name "coaxioms". Indeed, dually to axioms, which can be used in the proof tree at every depth, including 0, coaxioms can only be used "at an infinite depth" in the proof tree. Therefore, coaxioms filter out undesired infinite proof trees; in other words, they bound from above the greatest fixed point corresponding to the semantics of the generalized inference system.

As a second example, we consider the definition of the *first* sets in a grammar. Let us represent a context-free grammar by its set of terminals T, its set of non-terminals N, and all the productions $A ::= \beta_1 \mid \dots \mid \beta_n$ with left-hand side A, for each non-terminal A. Recall that, for each $\alpha \in (T \cup N)^+$, we can define the set $\mathit{first}(\alpha) = \{\sigma \mid \sigma \in T, \alpha \rightarrow^\star \sigma\beta\}$. Informally, $\mathit{first}(\alpha)$ is the set of the initial terminal symbols of the strings which can be derived from a string α in 0 or more steps.

The following inference system with coaxioms defines the judgment $\mathit{first}(\alpha, \mathcal{F})$, with $\mathcal{F} \subseteq T$.

$$\frac{}{\mathit{first}(\sigma\alpha, \{\sigma\})}\sigma \in T \qquad \frac{\mathit{first}(A, \mathcal{F})}{\mathit{first}(A\alpha, \mathcal{F})}\begin{array}{l}A \in N\\ A \not\rightarrow^\star \epsilon\end{array} \qquad \frac{\mathit{first}(A, \mathcal{F}) \ \ \mathit{first}(\alpha, \mathcal{F}')}{\mathit{first}(A\alpha, \mathcal{F} \cup \mathcal{F}')}\begin{array}{l}A \in N\\ A \rightarrow^\star \epsilon\end{array}$$

$$\frac{}{\mathit{first}(\epsilon, \emptyset)} \qquad \frac{\mathit{first}(\beta_1, \mathcal{F}_1) \ \ldots \ \mathit{first}(\beta_n, \mathcal{F}_n)}{\mathit{first}(A, \mathcal{F}_1 \cup \ldots \cup \mathcal{F}_n)} A ::= \beta_1 \mid \ldots \mid \beta_n \qquad \frac{\bullet}{\mathit{first}(A, \emptyset)} A \in N$$

The rules of the inference system correspond to the natural recursive definition of *first*. Note, in particular, that in a string of shape $A\alpha$, if the non-terminal A is *nullable*, that is, we can derive from it the empty string, then the *first* set for $A\alpha$ should also include the *first* set for α.

As in the previous example on graphs, the problem with this recursive definition is that, since the non-terminals in a grammar can mutually refer to each other, the function defined by the inductive interpretation can be undefined. That is, a naive top-down implementation might not terminate. For this reason, *first* sets are typically computed by an imperative bottom-up algorithm, or the top-down implementation is corrected by marking already encountered non-terminals, analogously to what is done for visiting graphs. Again as in the previous example, the coinductive interpretation may fail to be a function, whereas, with the coaxioms, we get the expected result.

We express now as inference systems with coaxioms the recursive definitions of functions shown at the beginning of Sect. 1. Let $\mathbb{Z}$ denote the set of integers, and $\mathbb{L}$ the set of (finite and infinite) lists of integers.

The first example is the function which checks whether all the elements of a list are positive, expressed by judgments of shape $\mathit{allPos}(l, b)$ with $l \in \mathbb{L}$ and $b \in \{T, F\}$.

$$\frac{}{\mathit{allPos}(\Lambda, T)} \qquad \frac{}{\mathit{allPos}(x{:}l, F)} x \leq 0 \qquad \frac{\mathit{allPos}(l, b)}{\mathit{allPos}(x{:}l, b)} x > 0 \qquad \frac{\bullet}{\mathit{allPos}(l, T)}$$

With the coaxioms, we obtain the expected function also on infinite lists of positives: indeed, we only consider the infinite trees where the nodes have a finite proof tree in the inference system enriched by the coaxioms. In this way, the infinite tree where $b = F$ is filtered out.

The function which checks whether an element belongs to a list, expressed by judgments of shape $\mathit{member}(x, l, b)$ with $x \in \mathbb{Z}$, $l \in \mathbb{L}$ and $b \in \{T, F\}$, is a very similar example, with the difference that the coaxioms map every list into false rather than true.

$$\frac{}{\mathit{member}(x, \Lambda, F)} \qquad \frac{}{\mathit{member}(x, x{:}l, T)} \qquad \frac{\mathit{member}(x, l, b)}{\mathit{member}(x, y : l, b)} x \neq y \qquad \frac{\bullet}{\mathit{member}(x, l, F)}$$

Analogously to the previous example, with the coaxioms we obtain the expected result also on infinite lists which do not contain the element.

The function which returns the set of the elements contained in a list is expressed by judgments of shape $\mathit{elems}(l, \mathit{xs})$, with $l \in \mathbb{L}$ and $\mathit{xs} \in \wp(\mathbb{Z})$.

$$\frac{}{\mathit{elems}(\Lambda, \emptyset)} \qquad \frac{\mathit{elems}(l, \mathit{xs})}{\mathit{elems}(x{:}l, \{x\} \cup \mathit{xs})} \qquad \frac{\bullet}{\mathit{elems}(l, \emptyset)}$$

In this case, the inductive interpretation gives the expected result only on finite lists, and the coinductive interpretation fails to be a function on infinite lists. For instance, for l the infinite list of 1s, any judgment $\mathit{elems}(l, \mathit{xs})$ with $1 \in \mathit{xs}$ can be derived. Indeed, for any such judgment we can construct an infinite proof tree which is a chain of applications of the last metarule. With the coaxioms, we only consider the infinite trees where the node $\mathit{elems}(l, \mathit{xs})$ has a finite proof tree in the inference system enriched by the coaxioms. This is only true for $\mathit{xs} = \{1\}$.

Note that coaxioms are needed to get the expected result not only on regular lists. Considering for example the infinite list $1 : 2 : 1 : 1 : 2 : 1 : 1 : 1 : 2 : ...$, it is easy to see that the same reasoning holds.

Finally, the function which returns the greatest element contained in a (non-empty) list is expressed by judgments of shape $\mathit{max}(l, x)$, with $l \in \mathbb{L}$ and $x \in \mathbb{Z}$.

$$\frac{}{\mathit{max}(x{:}\Lambda, x)} \qquad \frac{\mathit{max}(l, y)}{\mathit{max}(x{:}l, z)}\, z = \max(x, y) \qquad \frac{\bullet}{\mathit{max}(x{:}l, x)}$$

Analogously to the previous example, the coinductive interpretation fails to be a function (for instance, for l the infinite list of 1s, any judgment $\mathit{max}(l, x)$ with $x \geq 1$ can be derived), and the coaxioms "filter out" the wrong results.

3 Bounded Fixed Points

In this section, after recalling basic definitions, we define the *bounded fixed point generated by an element*, justifying its existence by the Knaster-Tarski theorem [34]. Then, we show that the interpretation generated by coaxioms of an inference system corresponds to a bounded fixed point in the powerset lattice. Finally, we provide a constructive characterization of bounded fixed points, again justified by a classical result (Kleene theorem). We refer to [22] for an history of these theorems with a number of good references.

In the following we assume a complete lattice $(L, \leq)$ with top and bottom elements $\top$ and $\bot$, and meet and join operations $\sqcap$ and $\sqcup$. Moreover, we use $\bigsqcap$ and $\bigsqcup$ to denote meet (greatest lower bound) and join (least upper bound) of a set, respectively.

Basic Definitions. Let $F : L \to L$, and $x \in L$. Then, x is a pre-fixed point of F iff $F(x) \leq x$; x is a post-fixed point of F iff $x \leq F(x)$; and x is a fixed point of F iff $x = F(x)$. Pre-fixed points will be also called *closed*, and post-fixed points will be also called *consistent* points. A function $F : L \to L$ is *monotone* if, for all $x, y \in L$, $x \leq y \Rightarrow F(x) \leq F(y)$.

In this general setting, the role of the universe is played by the top $\top$ of L, that of the inference system by a monotone function F, and that of the co-axioms by a distinguished element $\gamma \in L$, called *generator*.

Definition of Bounded Fixed-Point. In the following we assume a monotone function $F : L \to L$. The *bounded fixed point* generated by an element γ is the greatest fixed point of the monotone function obtained by restricting F to the down-set of the least pre-fixed point above γ. The construction is detailed and justified below. First of all we introduce two notations.

Definition 1. *Let $x \in L$. Then:*

- *The* closure *of x w.r.t. F is the element $\nabla_F(x)$ of L defined by* $\nabla_F(x) = \bigsqcap\{y \in L \mid x \leq y,\ F(y) \leq y\}$.
- *The* kernel *of x w.r.t. F is the element $\Delta_F(x)$ of L defined by* $\Delta_F(x) = \bigsqcup\{y \in L \mid y \leq x,\ y \leq F(y)\}$.

We can also see Δ_F and ∇_F as endofunctions on L, which are instances of well-known notions in lattice theory: closure and kernel operators.

From this definition immediately follows the *bounded coinduction principle.* Indeed, given $\beta \in L$, we have:

(CoInd) If $x \leq F(x)$ (x post-fixed), and $x \leq \beta$, then $x \leq \Delta_F(\beta)$.

The standard coinduction principle can be obtained as a specific instance of the more general principle above, by taking $\beta = \top$; for this particular case the hypothesis $x \leq \beta$ can be omitted, since it trivially holds. We will show in detail how to use this proof principle in Sect. 5.

The closure of an arbitrary element γ turns out to be *the best closed approximation of* γ, that is, the least pre-fixed point of F above γ, as shown below.

Proposition 1. *Let $\gamma \in L$. Then, $z = \nabla_F(\gamma)$ is the least pre-fixed point of F above γ.*

Proof. Set $S = \{x \in L \mid \gamma \leq x,\ F(x) \leq x\}$. We have to prove that $z \in S$, which then implies, by definition, that it is its least element. Since γ is a lower bound for all $x \in S$, by definition of meet we get $\gamma \leq z$. We can show that z is a pre-fixed point of F by the following steps:

- for all $x \in S$, $F(x) \leq x$ (def. of S) and $z \leq x$ (def. of $\bigsqcap$);
- for all $x \in S$, $F(x) \leq x$ (def. of S) and $F(z) \leq F(x)$ (F is monotone);
- for all $x \in S$, $F(z) \leq x$ (transitivity);
- $F(z) \leq z$ (def. of $\bigsqcap$).

$\square$

Note that if $\gamma = \bot$ we have that $\nabla_F(\bot)$ is the least pre-fixed point of F, that, thanks to the Knaster-Tarski theorem, is the least fixed point of F.
The kernel of a pre-fixed point β turns out to be *the greatest (post-)fixed-point of F below β*, as shown below.

Proposition 2. *Let $\beta \in L$. If β is a pre-fixed point of F and $z = \Delta_F(\beta)$, then $F(z) = z$.*

Proof. If β is an element of a complete lattice, then $L_\beta = \{x \in L \mid x \leq \beta\}$ is also a complete lattice, with top element β. If β is a pre-fixed point of F, then F is a monotone endofunction on L_β. Therefore, by the Knaster-Tarski theorem $F(z) = z$.

We can now define bounded fixed points generated by an element.

Definition 2 (Bounded fixed point). *Let $\gamma \in L$. The* bounded fixed point of F generated by γ, *denoted $Gen(F, \gamma)$, is the greatest fixed point of F below the closure of γ, that is, $Gen(F, \gamma) = \Delta_F(\nabla_F(\gamma))$.*

The bounded fixed point is well-defined since, thanks to Proposition 2, there exists the greatest fixed point below β, provided that the bound is a pre-fixed point. Since in general γ might not be pre-fixed, we need to construct a pre-fixed point from γ. Note that the first step of this construction *cannot* be expressed as the least fixed point of F on the complete lattice $\{x \in L \mid x \geq \gamma\}$, since in general F may fail to be an endofunction (e.g., if F is the function which maps any element to $\bot < \gamma$). Indeed, $\nabla_F(\gamma)$ is *not* a fixed point in general, but only a pre-fixed point: we need the two steps to obtain a fixed point.

Note also that the definition of bounded fixed point is asymmetric, that is, we take the greatest fixed point bounded from above by a least (pre-)fixed point, rather than the other way round. This is motivated by the intuition, explained in the previous section, that we essentially need a greatest fixed point, since we want to deal with non-well-founded structures, but we want to "constrain" in some way such greatest fixed point. Investigating the symmetric construction is a matter of further work (see the Conclusion).

An important fact is that bounded fixed points are a generalization of both least and greatest fixed points, since they can be obtained by taking particular generators, as stated in the following proposition.

Proposition 3.

1. *$Gen(F, \top)$ is the greatest fixed point of F*
2. *$Gen(F, \bot)$ is the least fixed point of F.*

Proof. 1. Note that $\nabla_F(\top) = \top$, since the only pre-fixed point above $\top$ is $\top$ itself, therefore we get $Gen(F, \top) = \Delta_F(\top)$, that is, the greatest fixed point of F, by Proposition 2.

2. As already noted $\nabla_F(\bot)$ is the least fixed point of F, in particular $\nabla_F(\bot)$ is post-fixed, therefore we get $Gen(F, \bot) = \Delta_F(\nabla_F(\bot)) = \nabla_F(\bot)$, namely it is the least fixed point of F.

□

Coaxioms as Generators. In Sect. 2 we have described two steps to construct $\mathit{Gen}(\mathcal{I},\gamma)$, the interpretation generated by coaxioms γ of an inference system $\mathcal{I}$.

1. First, we consider the inference system $\mathcal{I}_{\sqcup\gamma}$ obtained enriching $\mathcal{I}$ by judgments in γ considered as axioms, and we take its inductive interpretation $\mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$.
2. Then, we take the coinductive interpretation of the inference system obtained from $\mathcal{I}$ by keeping only rules with consequence in $\mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$, that is, we define
   $\mathit{Gen}(\mathcal{I},\gamma) = \mathit{CoInd}(\mathcal{I}_{\sqcap \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})})$

The definition of bounded fixed point is the formulation of these two steps in the general setting of complete lattices. Indeed, the inference operator $F_{\mathcal{I}}$ is a monotone function on the complete lattice $\wp(\mathcal{U})$ obtained by taking set inclusion as order, and specifying the coaxioms γ corresponds to fixing an arbitrary element of L as generator. To show the correspondence in a precise way, we give an alternative and equivalent characterization of closure.

Proposition 4. *Let $\gamma \in L$ and consider the function $F_{\sqcup\gamma} : L \to L$ defined by $F_{\sqcup\gamma}(x) = F(x) \sqcup \gamma$, that is clearly monotone. Then, $\nabla_{F_{\sqcup\gamma}}(\bot) = \nabla_F(\gamma)$.*

Proof. To prove the statement it is enough to show that $y \in L$ is a pre-fixed point of $F_{\sqcup\gamma}$ iff y is a pre-fixed point of F and $y \geq \gamma$. This trivially follows from the definition of $F_{\sqcup\gamma}$ and $\sqcup$, indeed $F(y) \sqcup \gamma = F_{\sqcup\gamma}(y) \leq y$ is equivalent to $F(y) \leq y$ and $\gamma \leq y$. $\square$

By this alternative characterization we can formally state the correspondence with the two steps for defining $\mathit{Gen}(\mathcal{I},\gamma)$.

Theorem 1. *Let $\mathcal{I}$ be an inference system and $\gamma, \beta \in \wp(\mathcal{U})$, with β closed w.r.t. $F_{\mathcal{I}}$, then the following facts hold:*

1. $(F_{\mathcal{I}})_{\sqcup\gamma} = F_{(\mathcal{I}_{\sqcup\gamma})}$ *(so we can safely omit brackets)*
2. $\nabla_{F_{\mathcal{I}}}(\gamma) = \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$
3. $\Delta_{F_{\mathcal{I}}}(\beta) = \mathit{CoInd}(\mathcal{I}_{\sqcap\beta})$.

Proof. 1. We have to show that, for $S \subseteq \mathcal{U}$, $(F_{\mathcal{I}})_{\sqcup\gamma}(S) = F_{(\mathcal{I}_{\sqcup\gamma})}(S)$. If $c \in (F_{\mathcal{I}})_{\sqcup\gamma}(S)$, then either $c \in \gamma$ or $c \in F_{\mathcal{I}}(S)$; in the former case there exists $\frac{}{c} \in \mathcal{I}_{\sqcup\gamma}$ by definition, in the latter there exists $\frac{Pr}{c} \in \mathcal{I}$ such that $Pr \subseteq S$, and this implies $\frac{Pr}{c} \in \mathcal{I}_{\sqcup\gamma}$. Therefore in both cases $c \in F_{(\mathcal{I}_{\sqcup\gamma})}(S)$.

   Conversely, if $c \in F_{(\mathcal{I}_{\sqcup\gamma})}(S)$, then there exists $\frac{Pr}{c} \in \mathcal{I}_{\sqcup\gamma}$ such that $Pr \subseteq S$. By definition of $\mathcal{I}_{\sqcup\gamma}$, either $\frac{Pr}{c} \in \mathcal{I}$ or $c \in \gamma$ and $Pr = \emptyset$, therefore in the former case $c \in F_{\mathcal{I}}(S)$ and in the latter $c \in \gamma$, thus in both cases $c \in (F_{\mathcal{I}})_{\sqcup\gamma}(S)$.
2. By Proposition 4 we get that $\nabla_{F_{\mathcal{I}}}(\gamma) = \nabla_{F_{\mathcal{I}_{\sqcup\gamma}}}(\emptyset)$, that is, the least fixed point of $F_{\mathcal{I}_{\sqcup\gamma}}$, thanks to statement (1) of this proposition and Proposition 2. Therefore, it corresponds to the *inductive interpretation* of the inference system $\mathcal{I}_{\sqcup\gamma}$, $\mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$.

3. Let $X = \mathit{CoInd}(\mathcal{I}_{\sqcap\beta})$, we have to show the two inclusions. First note that X is a post-fixed point w.r.t. $F_{\mathcal{I}}$, indeed $X \subseteq F_{\mathcal{I}_{\sqcap\beta}}(X)$, by definition of the coinductive interpretation, and $F_{\mathcal{I}_{\sqcap\beta}}(X) \subseteq F_{\mathcal{I}}(X)$, since each $c \in F_{\mathcal{I}_{\sqcap\beta}}(X)$ is the consequence of a rule $\frac{Pr}{c} \in \mathcal{I}_{\sqcap\beta}$ and by construction of $\mathcal{I}_{\sqcap\beta}$, this rule is also a rule of $\mathcal{I}$, therefore $c \in F_{\mathcal{I}}(X)$. In addition $c \in \beta$ again by definition of $\mathcal{I}_{\sqcap\beta}$, thus $X \subseteq \beta$, therefore by (CoInd) we get $X \subseteq \Delta_{F_{\mathcal{I}}}(\beta)$.

   On the other hand $\Delta_{F_{\mathcal{I}}}(\beta)$ is a post-fixed point of $F_{\mathcal{I}_{\sqcap\beta}}$. To show this fact first we note that for each $S \subseteq \beta$ we have $F_{\mathcal{I}}(S) \subseteq F_{\mathcal{I}_{\sqcap\beta}}(S)$, indeed if $c \in F_{\mathcal{I}}(S)$ then there exists a rule $\frac{Pr}{c} \in \mathcal{I}$ such that $Pr \subseteq S$, moreover we have that $F_{\mathcal{I}}(S) \subseteq F_{\mathcal{I}}(\beta) \subseteq \beta$ since β is closed, so $\frac{Pr}{c} \in \mathcal{I}_{\sqcap\beta}$ that implies that $c \in F_{\mathcal{I}_{\sqcap\beta}}$. Then, since $\Delta_{F_{\mathcal{I}}}(\beta)$ is a post-fixed point of $F_{\mathcal{I}}$ below β, we get that $\Delta_{F_{\mathcal{I}}}(\beta) \subseteq F_{\mathcal{I}}(\Delta_{F_{\mathcal{I}}}(\beta)) \subseteq F_{\mathcal{I}_{\sqcap\beta}}(\Delta_{F_{\mathcal{I}}}(\beta))$, so it is a post-fixed point. Therefore by the coinduction principle we get the other inclusion.

Thanks to Theorem 1, we can conclude that, given an inference system with coaxioms $(\mathcal{I}, \gamma)$:

$$\mathit{Gen}(\mathcal{I}, \gamma) = \mathit{CoInd}(\mathcal{I}_{\sqcap \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})}) = \Delta_{F_{\mathcal{I}}}(\nabla_{F_{\mathcal{I}}}(\gamma)) = \mathit{Gen}(F_{\mathcal{I}}, \gamma)$$

That is, the interpretation generated by coaxioms γ of the inference system $\mathcal{I}$ is exactly the bounded fixed point of $F_{\mathcal{I}}$ generated by γ.

Constructive Characterization of Bounded Fixed Point. The Kleene's theorem states that, under continuity hypotheses on F, we can characterize its greatest fixed point as the greatest lower bound of the descending chain obtained by repeatedly applying F to $\top$. By considering this theorem for the sublattice obtained as down-set of the bound, we can obtain a constructive characterization of the bounded fixed point generated by an element.

We recall some basic definitions. A *descending chain* in L is a set $C = \{x_i \mid i \in \mathbb{N}\} \subseteq L$ such that, for each $i \in \mathbb{N}$, $x_i \geq x_{i+1}$. A function $F : L \to L$ *preserves meet of descending chains* if and only if, for all descending chains C in L, we have $F(\bigsqcap C) = \bigsqcap F(C)$ where $F(C) = \{F(x_i) \mid x_i \in C\}$.

Given a function $F : L \to L$ and an element $\beta \in L$, set $C_{F,\beta} = \{F^n(\beta) \mid n \in \mathbb{N}\}$.

Proposition 5. *Let $F : L \to L$ be a function that preserves meet of descending chains, and $\beta \in L$ a pre-fixed point of F. Then:*

1. *$C_{F,\beta}$ is a descending chain in L*
2. *$\Delta_F(\beta) = \bigsqcap C_{F,\beta}$, that is, $\bigsqcap C_{F,\beta}$ is the greatest fixed point of F below β.*

Proof. 1. Since F preserves meet of descending chains, it is monotone, therefore, since β is pre-fixed, we get the thesis.

2. If β is an element of a complete lattice, then $L_\beta = \{x \in L \mid x \leq \beta\}$ is also a complete lattice, with top element β. If β is a closed point of F, then F is a monotone endofunction on L_β and it still preserves meet of descending chains. Therefore applying Kleene's theorem to L_β we get the thesis.

□

Note that for this constructive characterization we need an additional hypothesis on F. Under this assumption, the result of Proposition 5 immediately applies to our construction, as stated in the following corollary.

Corollary 1. *Let $F : L \to L$ be a function that preserves meet of descending chains, and $\gamma \in L$. Set $\beta = \nabla_F(\gamma)$. Then $Gen(F, \gamma) = \bigsqcap C_{F,\beta}$.*

Proof. By definition $Gen(F, \gamma) = \mathit{\Delta}_F(\beta)$. Since β is pre-fixed by Proposition 1 and F preserves meet of descending chains, by Proposition 5 we get the thesis. □

The characterization introduced above is important, but requires stronger assumptions on the function F. We now state a weaker result that is often enough for proving soundness, as will be illustrated in Sect. 5.

Proposition 6. *Let $F : L \to L$ be monotone and $\beta \in L$ a pre-fixed point, then*

$$\mathit{\Delta}_F(\beta) = \mathit{\Delta}_F\left(\bigsqcap C_{F,\beta}\right)$$

hence, in particular, $\mathit{\Delta}_F(\beta) \leq \bigsqcap C_{F,\beta}$.

Proof. Set $z = \bigsqcap C_{F,\beta}$. First of all we note that z is pre-fixed, indeed $F(z) \leq \bigsqcap F^{n+1}(\beta) = \beta \sqcap \bigsqcap F^{n+1}(\beta) = z$. We prove separately the two inequalities.

- $\mathit{\Delta}_F(z) \leq \mathit{\Delta}_F(\beta)$. By Proposition 2 $\mathit{\Delta}_F(z)$ is a fixed point, so in particular it is a post-fixed point, below z, by definition of $\bigsqcap$ we get $z \leq \beta$, so by transitivity $\mathit{\Delta}_F(z) \leq \beta$. By (CoInd) we get $\mathit{\Delta}_F(z) \leq \mathit{\Delta}_F(\beta)$.
- $\mathit{\Delta}_F(\beta) \leq \mathit{\Delta}_F(z)$. By Proposition 2 $\mathit{\Delta}_F(\beta)$ is a fixed point, so in particular a post-fixed point, below β. We prove by arithmetic induction that $\mathit{\Delta}_F(\beta) \leq F^n(\beta)$ for all $n \in \mathbb{N}$.
  Base $\mathit{\Delta}_F(\beta) \leq F^0(\beta) = \beta$ already proved.
  Induction Let us assume $\mathit{\Delta}_F(\beta) \leq F^n(\beta)$, so by monotonicity of F we get $F(\mathit{\Delta}_F(\beta)) \leq F^{n+1}(\beta)$. Since $\mathit{\Delta}_F(\beta)$ is a post-fixed point, we have that $\mathit{\Delta}_F(\beta) \leq F(\mathit{\Delta}_F(\beta))$, therefore by transitivity we get $\mathit{\Delta}_F(\beta) \leq F^{n+1}(\beta)$.
  By definition of $\bigsqcap$ we get $\mathit{\Delta}_F(\beta) \leq \bigsqcap C_{F,\beta} = z$, so by (CoInd) we get $\mathit{\Delta}_F(\beta) \leq \mathit{\Delta}_F(z)$.

Finally by anti-symmetry we get the equality. □

Another way to read the lemma above is that, given a bound β, we obtain the same greatest fixed point if we take as bound $\bigsqcap C_{F,\beta}$. Indeed from Proposition 6 and point 1 of Proposition 5 we can say more: given a bound β which is pre-fixed, we obtain the same greatest fixed point below β if we take as bound any element $F^n(\beta)$ of the descending chain.

4 Proof Trees

In this section we formally define several proof-theoretic characterizations of inference systems with coaxioms, and prove their equivalence[3] with the model-theoretic characterization given in the previous section.

First of all we recall the standard definition of proof trees and proof-theoretic characterization of inference systems.

Definition 3. *Given an inference system $\mathcal{I}$, a* proof tree *in $\mathcal{I}$ is a tree whose nodes are (labeled with) judgments in $\mathcal{U}$, and there is a node c with set of children Pr only if there exists a rule $\frac{Pr}{c}$. If a proof tree t in $\mathcal{I}$ has root j, then we say that t is a proof tree for j, or that j has proof tree t, in $\mathcal{I}$.*

Theorem 2. *Given an inference system $\mathcal{I}$, and a judgment $j \in \mathcal{U}$,*

1. $j \in \mathit{CoInd}(\mathcal{I})$ iff j has a proof tree in $\mathcal{I}$
2. $j \in \mathit{Ind}(\mathcal{I})$ iff j has a finite proof tree in $\mathcal{I}$.

See [15,23].

The first proof-theoretic characterization is based on the following theorem, which slightly generalizes the standard correspondence between proof trees in $\mathcal{I}$ and the coinductive interpretation of $\mathcal{I}$.

Theorem 3. *Given an inference system $\mathcal{I}$, and $\beta \subseteq \mathcal{U}$ a closed set of judgments, we have that, for all $j \in \mathcal{U}$, $j \in \Delta_{F_\mathcal{I}}(\beta)$ iff there exists a proof tree t for j in $\mathcal{I}$ such that each node of t is in β.*

Proof. By Theorem 1, $\Delta_{F_\mathcal{I}}(\beta) = \Delta_{F_{\mathcal{I}_{\sqcap\beta}}}(\mathcal{U}) = \mathit{CoInd}(\mathcal{I}_{\sqcap\beta})$. Thanks to Theorem 2 (1), we get that $j \in \mathit{CoInd}(\mathcal{I}_{\sqcap\beta})$ iff there exists a proof tree t for j in $\mathcal{I}_{\sqcap\beta}$. By Definition 3, each node of t is (labeled by) a consequence c of a rule in $\mathcal{I}_{\sqcap\beta}$, that is, $c \in \beta$ by definition of $\mathcal{I}_{\sqcap\beta}$, and this implies the thesis. $\square$

As a particular case, we get our first proof-theoretic characterization

Corollary 2. *Given an inference system with coaxioms $(\mathcal{I}, \gamma)$ and a judgment $j \in \mathcal{U}$, we have that $j \in \mathit{Gen}(\mathcal{I}, \gamma)$ iff there exists a proof tree t for j in $\mathcal{I}$ such that each node of t has a finite proof tree in $\mathcal{I}_{\sqcup\gamma}$.*

Proof. By Theorem 1, $\mathit{Gen}(\mathcal{I}, \gamma) = \Delta_{F_\mathcal{I}}(\beta)$, with $\beta = \nabla_{F_\mathcal{I}}(\gamma)$. Thanks to Theorem 3, we get that, for all $j \in \mathcal{U}$, $j \in \mathit{Gen}(\mathcal{I}, \gamma)$ iff there exists a proof tree t for j in $\mathcal{I}$ such that each node of t is in β. Again by Theorem 1 we get that $\beta = \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$, so by Theorem 2 (2) we get that a node j' of t is in β iff there exists a finite proof tree for j' in $\mathcal{I}_{\sqcup\gamma}$. $\square$

For the second proof-theoretic characterization, we need to define *approximated proof trees.*

In the definition below, let us denote by j_t the root of tree t.

[3] For the last, under the hypotheses of Proposition 5.

Definition 4. *Given an inference system with coaxioms* $(\mathcal{I}, \gamma)$*, the sets* $\mathcal{T}_n$ *of approximated proof trees of level* n *in* $(\mathcal{I}, \gamma)$*, for* $n \in \mathbb{N}$*, are inductively defined as follows:*

$$t \in \mathcal{T}_0 \quad \textit{if } t \textit{ finite proof tree in } \mathcal{I}_{\sqcup\gamma}$$

$$\frac{\mathcal{T}}{c} \in \mathcal{T}_n \textit{ if } \frac{Pr}{c} \in \mathcal{I},\ Pr = \{j_t \mid t \in \mathcal{T}\},\ \textit{and } \mathcal{T} \subseteq \mathcal{T}_{n-1}$$

In other words, an approximated proof tree of level n in $(\mathcal{I}, \gamma)$ is a finite proof tree in $(\mathcal{I}, \gamma)$ where coaxioms can only be used at depth $\geq n$.

The following lemma states that approximated proof trees of level n correspond to the n-th element of the descending chain $C_{F_{\mathcal{I}},\beta} = \{F_{\mathcal{I}}^n(\beta) \mid n \in \mathbb{N}\}$, with $\beta = \nabla_{F_{\mathcal{I}}}(\gamma) = \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$.

Lemma 1. *Given an inference system with coaxioms* $(\mathcal{I}, \gamma)$*, and a judgment* $j \in \mathcal{U}$*, we have that, for all* $n \in \mathbb{N}$*,* $j \in F_{\mathcal{I}}^n(\nabla_{F_{\mathcal{I}}}(\gamma))$ *iff* j *has an approximated proof tree of level* n *in* $(\mathcal{I}, \gamma)$*.*

Proof. Let β be $\nabla_{F_{\mathcal{I}}}(\gamma)$. We prove the thesis by induction on n.

Base If $n = 0$, then, by Theorem 1, $\beta = \nabla_{F_{\mathcal{I}}}(\gamma)$ corresponds to the inductive interpretation of $\mathcal{I}_{\sqcup\gamma}$, therefore the equivalence holds by Theorem 2 (2).

Induction We assume the equivalence for n and prove it for $n + 1$. We prove separately the two implications.

$\Rightarrow$ If $c \in F_{\mathcal{I}}^{n+1}(\beta)$, then there exists $\frac{Pr}{c} \in \mathcal{I}$ such that $Pr \subseteq F_{\mathcal{I}}^n(\beta)$. Hence, by inductive hypothesis, each judgment in Pr has an approximated proof tree of level n, that is, $Pr = \{j_t \mid t \in \mathcal{T}\}$, with $\mathcal{T} \subseteq \mathcal{T}_n$. Hence, $t = \frac{\mathcal{T}}{c}$ is a proof tree for c, and by definition, $t \in \mathcal{T}_{n+1}$.

$\Leftarrow$ If $t \in \mathcal{T}_{n+1}$ is an approximated proof tree for $c \in \mathcal{U}$, then, by definition, there exists $\frac{Pr}{c} \in \mathcal{I}$ such that $t = \frac{\mathcal{T}}{c}$, $Pr = \{j_t \mid t \in \mathcal{T}\}$, and $\mathcal{T} \subseteq \mathcal{T}_n$. By inductive hypothesis we have $Pr \subseteq {F_{\mathcal{I}}}^n(\beta)$, and, by definition of $F_{\mathcal{I}}$, this implies $c \in F_{\mathcal{I}}^{n+1}(\beta)$ as needed.

$\square$

Corollary 3. *Given an inference system with coaxioms* $(\mathcal{I}, \gamma)$*, and a judgment* $j \in \mathcal{U}$*, the following are equivalent:*

1. $j \in \mathit{Gen}(\mathcal{I}, \gamma)$
2. *there exists a proof tree* t *for* j *in* $\mathcal{I}$ *such that each node has an approximated proof tree of level* n *in* $(\mathcal{I}, \gamma)$*, for all* $n \in \mathbb{N}$*.*

Proof. By Theorem 1, Proposition 6, and Theorem 3, we get that, for all $j \in \mathcal{U}$, $j \in \mathit{Gen}(\mathcal{I}, \gamma)$ iff there exists a proof tree t for j in $\mathcal{I}$ such that each node j' of t is in $\bigsqcap C_{F_{\mathcal{I}},\beta}$ with $\beta = \nabla_{F_{\mathcal{I}}}(\gamma)$. By Lemma 1, $j' \in \bigsqcap C_{F_{\mathcal{I}},\beta}$ iff has an approximated proof tree of level n, for all $n \in \mathbb{N}$. $\square$

If the hypotheses of Proposition 5 are satisfied, then we get a simpler equivalent proof-theoretic characterization.

Corollary 4. *Given an inference system with coaxioms $(\mathcal{I}, \gamma)$, and a judgment $j \in \mathcal{U}$, if $F_{\mathcal{I}}$ preserves meet of descending chains, then the following are equivalent:*

1. *$j \in \mathit{Gen}(\mathcal{I}, \gamma)$*
2. *j has an approximated proof tree of level n in $(\mathcal{I}, \gamma)$, for all $n \in \mathbb{N}$.*

Proof. Let β be $\nabla_{F_{\mathcal{I}}}(\gamma)$. By Theorem 1 and Proposition 5, we get that $\mathit{Gen}(\mathcal{I}, \gamma) = \bigsqcap C_{F_{\mathcal{I}},\beta}$, therefore the thesis follows immediately from Lemma 1. $\square$

5 Reasoning with Coaxioms

In this section we discuss proof techniques for inference systems with coaxioms.

Assume that $G = \mathit{Gen}(\mathcal{I}, \gamma)$ is the interpretation generated by coaxioms for some $(\mathcal{I}, \gamma)$, and that S (for "specification") is the intended set of judgments, called *valid* in the following.

Typically, we are interested in proving $S \subseteq G$ (*completeness*, that is, each valid judgment can be derived) and/or $G \subseteq S$ (*soundness*, that is, each derivable judgment is valid). Proving both properties amounts to say that the inference system with coaxioms actually defines the intended set of judgments.

In the following, set $\beta = \nabla_{F_{\mathcal{I}}}(\gamma) = \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$.

Completeness Proofs. To show completeness, we can use the bounded coinduction principle. Indeed, since $G = \Delta_{F_{\mathcal{I}}}(\beta)$, if $S \leq \beta$ and S is a post-fixed point of $F_{\mathcal{I}}$, by (CoInd) we get that $S \leq G$. That is, using the notations of inference systems, to prove completeness it is enough to show that:

- $S \subseteq \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$
- $S \subseteq F_{\mathcal{I}}(S)$

We illustrate the technique on the inference system with coaxioms $(\mathcal{I}, \gamma)$ which defines the judgment $\mathit{allPos}(l, b)$ (page 7). Set S^{allPos} the set of pairs (l, b) where b is T if all the elements in l are positive, F otherwise. Completeness means that the judgment $\mathit{allPos}(l, b)$ can be proved, for all $(l, b) \in S^{\mathit{allPos}}$. By the bounded coinduction principle, it is enough to show that

- $S^{\mathit{allPos}} \subseteq \mathit{Ind}(\mathcal{I}_{\sqcup\gamma})$
- $S^{\mathit{allPos}} \subseteq F_{\mathcal{I}}(S^{\mathit{allPos}})$

To prove the first condition, we have to show that, for each $(l, b) \in S^{\mathit{allPos}}$, $\mathit{allPos}(l, b)$ has a *finite* proof tree in $\mathcal{I}_{\sqcup\gamma}$. This can be easily shown, indeed:

- If l contains a (first) non-positive element, hence $l = x_1 : \ldots : x_n : x : l'$ with $x_i > 0$ for $i \in [1..n]$, $x \leq 0$, and $b = F$ then we can reason by arithmetic induction on n. Indeed, for $n = 0$, (l, b) is the consequence of the second rule with no premises, and for $n > 0$ it is the consequence of the third rule where we can apply the inductive hypothesis to the premise.

– If l contains only positive elements, hence $b = T$, then (l, b) is a coaxiom, hence it is the consequence of a rule with no premises in $\mathcal{I}_{\sqcup\gamma}$.

To prove the second condition, we have to show that, for each $(l, b) \in S^{allPos}$, $allPos(l, b)$ is the consequence of a rule with premises in S^{allPos}. This can be easily shown, indeed:

– If $l = \Lambda$, hence $b = T$, then $allPos(\Lambda, T)$ is the consequence of the first rule with no premises.
– If $l = x : l'$ with $x \leq 0$, hence $b = F$, then $allPos(l, F)$ is the consequence of the second rule with no premises.
– If $l = x : l'$ with $x > 0$, and $b = T$, hence $(l', T) \in S^{allPos}$, then $allPos(l, T)$ is the consequence of the third rule with premise (l', T), and analogously if $b = F$.

Soundness Proofs. To show soundness, it is convenient to use the alternative characterization in terms of approximated proof trees given in Sect. 4, as detailed below. First of all, from Proposition 6, $G \subseteq \bigcap\{F^n_{\mathcal{I}}(\beta) \mid n \geq 0\}$. Hence, to prove $G \subseteq S$, it is enough to show that $\bigcap\{F^n_{\mathcal{I}}(\beta) \mid n \geq 0\} \subseteq S$. Moreover, by Lemma 1, for all $n \in \mathbb{N}$, judgments in $F^n_{\mathcal{I}}(\beta)$ are those which have an approximated proof tree of level n. Hence, to prove set inclusion, we can show that all judgments which have an approximated proof tree of level n for each n are in S or equivalently, by contraposition, that judgments which are not in S, that is, non-valid judgments, fail to have an approximated proof tree of level n for some n.

We illustrate the technique again on the example of *allPos*. We have to show that, for each $(l, b) \notin S^{allPos}$, there exists $n \geq 0$ such that (l, b) cannot be proved by using coaxioms at level greater than n. By cases:

– If l contains a (first) non-positive element, hence $l = x_1 : \ldots : x_n : x : l'$ with $x_i > 0$ for $i \in [1..n]$, $x \leq 0$, then, assuming that coaxioms can only be used at a level greater than $n + 1$, (l, b) can only be derived by instantiating n times the third rule, and once the second rule, hence b cannot be T.
– If l contains only positive elements, then it is immediate to see that there is no finite proof tree for (l, F).

6 Taming Coaxioms: Advanced Examples

Mutual Recursion. Circular definitions involving inductive and coinductive judgments have no semantics in standard inference systems, because all judgments have to be interpreted either inductively, or coinductively. The same problem arises in the context of coinductive logic programming [32], where a logic program has a well-defined semantics only if inductive and coinductive predicates can be stratified: each stratum defines only inductive or coinductive predicates (possibly defined in a mutually recursive way), and cannot depend on predicates defined in upper strata. Hence, it is possible to define the semantics of a logic program only if there are no mutually recursive definitions involving both inductive and coinductive predicates.

We have already seen that an inductive inference system corresponds to a generalized inference system with no coaxioms, while a coinductive one corresponds to a generalized one where coaxioms consist of all judgments in $\mathcal{U}$; however, between these two extremes, coaxioms offer many other possibilities thus allowing a finer control on the semantics of the defined judgments.

There exist cases where two or more related judgments need to be defined recursively, but for some of them the correct interpretation is inductive, while for others is coinductive [5,31,32]. In such cases, coaxioms may be employed to provide the correct definition in terms of a single inference system with no stratification, although special care is required to get from the inference system the intended meaning of judgments. To see this, let us consider the judgment $\mathit{path0}(t)$, where t is an infinite tree[4] over $\{0,1\}$, which holds iff there exists a path starting from the root of t and containing just 0s; trees are represented as infinite terms of shape $\mathit{tree}(n,l)$, where $n \in \{0,1\}$ is the root of the tree, and l is the infinite list of its direct subtrees. For instance, if t_1 and t_2 are the trees defined by the syntactic equations

$$t_1 = \mathit{tree}(0,l_1) \qquad l_1 = t_2{:}t_1{:}l_1 \qquad t_2 = \mathit{tree}(0,l_2) \qquad l_2 = \mathit{tree}(1,l_1){:}l_2$$

then we expect $\mathit{path0}(t_1)$ to hold, but not $\mathit{path0}(t_2)$.

To define $\mathit{path0}$, we introduce an auxiliary judgment $\mathit{is\_in0}(l)$ testing whether an infinite list l of trees contains a tree t such that $\mathit{path0}(t)$ holds. Intuitively, we expect $\mathit{path0}$ and $\mathit{is\_in0}$ to be interpreted coinductively and inductively, respectively; this reflects the fact that $\mathit{path0}$ checks a property universally quantified over an infinite sequence (a *safety* property in the terminology of concurrent systems): all the elements of the path must equal 0; on the contrary, $\mathit{is\_in0}$ checks a property existentially quantified over an infinite sequence (a *liveness* property in the terminology of concurrent systems): the list must contain a tree t with a specific property (that is, $\mathit{path0}(t)$ must hold). Driven by this intuition, one could be tempted to define the following inference system with coaxioms for all judgments of shape $\mathit{path0}(t)$, and no coaxioms for judgments of shape $\mathit{is\_in0}(l)$:

$$\frac{\mathit{is\_in0}(l)}{\mathit{path0}(\mathit{tree}(0,l))} \qquad \frac{\bullet}{\mathit{path0}(t)} \qquad \frac{\mathit{path0}(t)}{\mathit{is\_in0}(t{:}l)} \qquad \frac{\mathit{is\_in0}(l)}{\mathit{is\_in0}(t{:}l)}$$

Unfortunately, because of the mutual recursion between $\mathit{is\_in0}$ and $\mathit{path0}$, the inference system above does not capture the intended behavior: $\mathit{is\_in0}(l)$ is derivable for every infinite list of trees l, even when l does not contain a tree t with an infinite path starting from its root and containing just 0s.

To overcome this problem, we replace the judgment $\mathit{is\_in0}$ with the more general one $\mathit{is\_in}$, such that $\mathit{is\_in}(t,l)$ holds iff the infinite list l contains the tree t. Consequently, we can define the following generalized inference system:

$$\frac{\mathit{is\_in}(t,l) \quad \mathit{path0}(t)}{\mathit{path0}(\mathit{tree}(0,l))} \qquad \frac{\bullet}{\mathit{path0}(t)} \qquad \frac{}{\mathit{is\_in}(t,t{:}l)} \qquad \frac{\mathit{is\_in}(t,l)}{\mathit{is\_in}(t,t'{:}l)}$$

[4] For the purpose of this example, we only consider trees with infinite depth and branching.

Now the semantics of the system corresponds to the intended one, and we do not need to stratify the definitions into two separate inference systems.

Following the characterization in terms of proof trees and the proof techniques provided in Sects. 4 and 5, we can sketch a proof of correctness. Let S be the set where elements have either shape $\mathit{path0}(t)$, where t represents a tree with an infinite path of just 0s starting from its root, or $\mathit{is\_in}(t, l)$, where l represents an infinite list containing the tree t; then a judgment belongs to S iff it can be derived in the generalized inference system defined above.

Completeness: We first show that the set S is a post-fixed point, that is, it is consistent w.r.t. the inference rules, coaxioms excluded. Indeed, if t has an infinite path of 0s, then it has necessarily shape $\mathit{tree}(0, l)$, where l must contain a tree t' with an infinite path of 0s. Hence, the inference rule for $\mathit{path0}$ can be applied with premises $\mathit{is\_in}(t', l) \in S$, and $\mathit{path0}(t') \in S$. If an infinite list contains a tree t, then it has necessarily shape $t'{:}l$ where, either $t = t'$, and hence the axiom for $\mathit{is\_in}$ can be applied, or $t \neq t'$ and t is contained in l, and hence the inference rule for $\mathit{is\_in}$ can be applied with premise $\mathit{is\_in}(t, l) \in S$.

We then show that S is bounded by the closure of the coaxioms. For the elements of shape $\mathit{path0}(t)$ it suffices to directly apply the corresponding coaxiom; for the elements of shape $\mathit{is\_in}(t, l)$ we can show that there exists a finite proof tree built possibly also with the coaxioms by induction on the first position (where the head of the list corresponds to 0) in the list where t occurs. If the position is 0 (base case), then $l = t{:}l'$, and the axiom can be applied; if the position is $n > 0$ (inductive step), then $l = t'{:}l'$ and t occurs in l' at position $n - 1$, therefore, by inductive hypothesis, there exists a finite proof tree for $\mathit{is\_in}(t, l')$, therefore we can build a finite proof tree for $\mathit{is\_in}(t, l)$ by applying the inference rule for $\mathit{is\_in}$.

Soundness: We first observe that the only finite proof trees that can be derived for $\mathit{is\_in}(t, l)$ are obtained by application of the axiom for $\mathit{is\_in}$, hence $\mathit{is\_in}(t, l)$ holds iff there exists a finite proof tree for $\mathit{is\_in}(t, l)$ built with the inference rules for $\mathit{is\_in}$. Then, we can prove that, if $\mathit{is\_in}(t, l)$ holds, then t is contained in l by induction on the inference rules for $\mathit{is\_in}$. For the axiom (base case) the claim trivially holds, while for the other inference rule we have that if t belongs to l, then trivially t belongs to $t'{:}l$.

For the elements of shape $\mathit{path0}(t)$ we first observe that by the coaxioms they all trivially belong to the closure of the coaxioms. Then, any proof tree for $\mathit{path0}(t)$ must be infinite, because there are no axioms but only one inference rule for $\mathit{path0}$ where $\mathit{path0}$ is referred in the premises; furthermore, such a rule is applicable only if the root of the tree is 0. We have already proved that if $\mathit{is\_in}(t, l)$ is derivable, then t belongs to l, therefore we can conclude that if $\mathit{path0}(t)$ is derivable, then t contains an infinite path starting from its root, and containing just 0s.

A Numerical Example. It is well known that real numbers in the closed interval [0,1] can be represented by infinite sequences $(d_i)_{i \in \mathbb{N}^+}$ of decimal[5] digits,

[5] Of course the example can be generalized to any base $B \geq 2$.

where $\mathbb{N}^+$ denotes the set of all positive natural numbers. Indeed, $(d_i)_{i\in\mathbb{N}^+}$ represents the real number which is the limit of the series $\sum_{i=1}^{\infty} 10^{-i} d_i$ in the standard complete metric space of real numbers (such a limit always exists by completeness, because the associated sequence of partial sums is always a Cauchy sequence). Such a representation is not unique for all rational numbers in [0,1] (except for the bounds 0 and 1) that can be represented by a finite sequence of digits followed by an infinite sequence of 0s; for instance, 0.42 can be represented either by the sequence $42\bar{0}$, or by the sequence $41\bar{9}$, where $\bar{d}$ denotes the infinite sequence containing just the digit d.

For brevity, for $r = (d_i)_{i\in\mathbb{N}^+}$, $[\![r]\!]$ denotes $\sum_{i=1}^{\infty} 10^{-i} d_i$ (that is, the real number represented by r). We want to define the judgment $add(r_1, r_2, r, c)$ which holds iff $[\![r_1]\!] + [\![r_2]\!] = [\![r]\!] + c$ with c an integer number; that is, $add(r_1, r_2, r, c)$ holds iff the addition of the two real numbers represented by the sequences r_1 and r_2 yields the real number represented by the sequence r with carry c. We will soon discover that, to get a complete definition for add, c is required to range over a proper superset of the set $\{0, 1\}$, differently from what one could initially expect.

We can define the judgment add with the following generalized inference system, where $\div$ and mod denote the integer division, and the remainder operator, respectively.

$$\frac{add(r_1, r_2, r, c)}{add(d_1{:}r_1, d_2{:}r_2, (s \bmod 10){:}r, s \div 10)}\, s = d_1 + d_2 + c$$

$$\frac{\bullet}{add(r_1, r_2, r, c)}$$

A real number in [0,1] is represented by an infinite list of decimal digits, which, therefore, can always be decomposed as $d{:}r$, where d is the first digit (corresponding to the exponent -1), and r is the rest of the list of digits. Here, r_1, r_2, and r range over the set of infinite lists of decimal digits, while the carry must range over $\{-1, 0, 1, 2\}$ to support a complete definition. As clearly emerges from the proof of completeness provided below, besides the obvious values 0 and 1, the values -1 and 2 have to be considered for the carry to ensure a complete definition of add because both $add(\bar{0}, \bar{0}, \bar{9}, -1)$ and $add(\bar{9}, \bar{9}, \bar{0}, 2)$ hold, and, hence, should be derivable; these two judgments allow the derivation of an infinite number of other valid judgments, as, for instance, $add(1\bar{0}, 1\bar{0}, 1\bar{9}, 0)$ and $add(1\bar{9}, 1\bar{9}, 4\bar{0}, 0)$, respectively.

Also in this case we can sketch a proof of correctness: for all infinite sequences of decimal digits r_1, r_2 and r, and all $c \in \{-1, 0, 1, 2\}$, $add(r_1, r_2, r, c)$ is derivable iff $[\![r_1]\!] + [\![r_2]\!] = [\![r]\!] + c$.

Completeness: By the coaxioms we trivially have that each element $add(r_1, r_2, r, c)$ such that $[\![r_1]\!] + [\![r_2]\!] = [\![r]\!] + c$ with $c \in \{-1, 0, 1, 2\}$ belongs to the closure of the coaxioms.

To show that the unique inference rule of the system is consistent with the set of all correct judgments, let us assume that $[\![r_1']\!] + [\![r_2']\!] = [\![r']\!] + c'$ with $r_1' = d_1{:}r_1$, $r_2' = d_2{:}r_2$, $r' = d{:}r$ and $c' \in \{-1, 0, 1, 2\}$. Let us set $s = 10c' + d$, and $c = s - d_1 - d_2$, then $s \bmod 10 = d$ and $s \div 10 = c'$, and we get the desired conclusion of the inference rule, and the side condition holds; it remains to show that $[\![r_1]\!] + [\![r_2]\!] = [\![r]\!] + c$ with $c \in \{-1, 0, 1, 2\}$.

We first observe that by the properties of limits w.r.t. the usual arithmetic operations, and by definition of $[\![-]\!]$, for all infinite sequence r of decimal digits, if $r = d{:}r'$, then $[\![r]\!] = 10^{-1}(d + [\![r']\!])$; then, from the hypotheses we get the equality $d_1 + [\![r_1]\!] + d_2 + [\![r_2]\!] = d + [\![r]\!] + 10c'$, and, therefore, $[\![r_1]\!] + [\![r_2]\!] = [\![r]\!] + c$; finally, since $c = [\![r_1]\!] + [\![r_2]\!] - [\![r]\!]$, and $0 \leq [\![r_1]\!], [\![r_2]\!], [\![r]\!] \leq 1$, we get $c \in \{-1, 0, 1, 2\}$.

Soundness: Let $r'_1 = d_1{:}r_1$, $r'_2 = d_2{:}r_2$, and $r' = d{:}r$ be infinite sequences of decimal digits, and $c' \in \{-1, 0, 1, 2\}$; we observe that the judgment $add(r'_1, r'_2, r', c')$ can be derived from the unique inference rule only with the premise $add(r_1, r_2, r, c)$ where c must equal $10c' + d - d_1 - d_2$ and must range over $\{-1, 0, 1, 2\}$.

To prove soundness we show that if $[\![r'_1]\!] + [\![r'_2]\!] \neq [\![r']\!] + c'$, then $add(r'_1, r'_2, r', c')$ cannot be derived in the inference system. Let us set $\delta' = |[\![r']\!] + c' - [\![r'_1]\!] - [\![r'_2]\!]|$; obviously, under the hypothesis $[\![r'_1]\!] + [\![r'_2]\!] \neq [\![r']\!] + c'$, we get $\delta' > 0$. In particular, the following fact holds: if $\delta' \geq 4 \cdot 10^{-1}$, then $10c' + d - d_1 - d_2 \notin \{-1, 0, 1, 2\}$. Indeed, by the identity $[\![r]\!] = 10^{-1}(d + [\![r']\!])$ already used for the proof of completeness, we have that $\delta' = 10^{-1}|[\![r]\!] + c - [\![r_1]\!] - [\![r_2]\!]|$, with $c = 10c' + d - d_1 - d_2$; $10^{-1}([\![r]\!] + c - [\![r_1]\!] - [\![r_2]\!]) \geq 4 \cdot 10^{-1}$ implies $c \geq 3$ ($[\![r_1]\!], [\![r_2]\!], [\![r]\!] \in [0, 1]$), and, hence, $c = 10c' + d - d_1 - d_2 \notin \{-1, 0, 1, 2\}$. By duality, $10^{-1}([\![r]\!] + c - [\![r_1]\!] - [\![r_2]\!]) \leq -4 \cdot 10^{-1}$ implies $c \leq -2$, hence $c = 10c' + d - d_1 - d_2 \notin \{-1, 0, 1, 2\}$.

By virtue of this fact, and thanks to the hypotheses, we can prove by arithmetic induction over n that for all $n \geq 1$, if $\delta' \geq 4 \cdot 10^{-n}$, then there exists a finite proof tree for $add(r'_1, r'_2, r', c')$ where the coaxioms are applied at most at depth $n - 1$. The base case is directly derived from the previously proven fact. For the inductive step we observe that if the inference rule is applicable for deriving the conclusion $add(r'_1, r'_2, r', c')$, then we can apply the inductive hypothesis for the premise $add(r_1, r_2, r, c)$ since we have already shown that $\delta' = 10^{-1}\delta$, therefore $\delta \geq 4 \cdot 10^{-(n-1)}$.

We can now conclude by observing that if $[\![r'_1]\!] + [\![r'_2]\!] \neq [\![r']\!] + c'$, then there exists n such that $\delta' \geq 4 \cdot 10^{-n}$, therefore, by the previous result, we deduce that it is not possible to build a finite tree for $add(r'_1, r'_2, r', c')$ where the coaxioms are applied at arbitrary depth k (in particular, k is bounded by $n - 1$); therefore $add(r'_1, r'_2, r', c')$ cannot be derived in the inference system.

From the proof of soundness we can also deduce that if we let c range over $\mathbb{Z}$, then the inference system becomes unsound; for instance, $add(\bar{0}, \bar{0}, \bar{0}, 1)$ would be derivable, but $[\![\bar{0}]\!] + [\![\bar{0}]\!] \neq [\![\bar{0}]\!] + 1$:

$$\cfrac{\cfrac{\vdots}{add(\bar{0}, \bar{0}, \bar{0}, 10^1)}}{add(\bar{0}, \bar{0}, \bar{0}, 10^0)}$$

Big-Step Operational Semantics with Divergence. It is well-known that divergence cannot be captured by the big-step operational semantics of a programming language when semantic rules are interpreted inductively (that is, in the standard way) [4,6,23]. When rules are interpreted coinductively some partial result can be obtained under suitable hypotheses, but a practical way to capture divergence with a big-step operational semantics is to introduce two different forms of judgment [14,23]: one corresponds to the standard big-step evaluation relation, and is defined inductively, while the other one captures divergence, and is defined coinductively in terms of the inductive judgment, thus requiring stratification. Other approaches consist in considering coinductive trace-based big-step semantics [27], and flag-based big-step semantics [29].

Syntax of terms and values

$$e ::= v \mid x \mid e\ e \qquad v ::= \lambda x.e \qquad v_\infty ::= v \mid \infty$$

Semantic rules

$$(\text{coax})\ \frac{\bullet}{e \Rightarrow \infty} \qquad (\text{val})\ \frac{}{v \Rightarrow v} \qquad (\text{app})\ \frac{e_1 \Rightarrow \lambda x.e \quad e_2 \Rightarrow v \quad e[x \leftarrow v] \Rightarrow v_\infty}{e_1\ e_2 \Rightarrow v_\infty}$$

$$(\text{l-inf})\ \frac{e_1 \Rightarrow \infty}{e_1\ e_2 \Rightarrow \infty} \qquad (\text{r-inf})\ \frac{e_1 \Rightarrow v \quad e_2 \Rightarrow \infty}{e_1\ e_2 \Rightarrow \infty}$$

Fig. 1. Call-by-value big-step semantics of λ-calculus with divergence

With coaxioms a unique judgment can be defined in a more direct and compact way. We show[6] how this is possible for the standard call-by-value operational semantics of the λ-calculus in big-step style. Figure 1 defines syntax, values, and semantic rules. The meta-variable v ranges over standard values, that is, lambda abstractions, while v_∞ includes also divergence, represented by ∞. The evaluation judgment has the general shape $e \Rightarrow v_\infty$, meaning that either e evaluates to a value v (when $v_\infty \neq \infty$) or diverges (when $v_\infty = \infty$).

For what concerns the semantic rules, only a coaxiom is needed, stating that every expression may diverge. This ensures that ∞ can be the only allowed outcome for the evaluation of an expression which diverges; this can only happen when the corresponding derivation tree is infinite. Rule (val) is standard. Rule (app) deals with the evaluation of application when both expressions e_1 and e_2 do not diverge; the meta-variable v is required for the judgment $e_2 \Rightarrow v$ to guarantee convergence of e_2, while v_∞ is used for the result of the whole application, since the evaluation of the body of the lambda abstraction could diverge. As usual, $e[x \leftarrow v]$ denotes capture-avoiding substitution modulo α-renaming. Rules (l-inf) and (r-inf) cover the cases when either e_1 or e_2 diverges when trying to evaluate application, assuming that a left-to-right evaluation strategy has been imposed.

We show that the only judgment derivable for $e_\Delta = (\lambda x.x\ x)\lambda x.x\ x$ is $e_\Delta \Rightarrow \infty$. To this aim, we first disregard the coaxiom and exhibit an infinite derivation tree for the judgment $e_\Delta \Rightarrow v_\infty$, derivable for all v_∞:

[6] This example was inspired by Bart Jacobs.

$$\text{(app)}\ \dfrac{\text{(val)}\ \dfrac{}{\lambda x.x\ x \Rightarrow \lambda x.x\ x} \qquad \text{(val)}\ \dfrac{}{\lambda x.x\ x \Rightarrow \lambda x.x\ x} \qquad \text{(app)}\ \dfrac{\vdots}{(x\ x)[x \leftarrow \lambda x.x\ x] \Rightarrow v_\infty}}{(x\ x)[x \leftarrow \lambda x.x\ x] = e_\Delta \Rightarrow v_\infty}$$

In this particular case the derivation tree is also regular, but of course there are examples of divergent computations whose derivation tree is not regular. The vertical dots indicate that the derivation continues with the same repeated pattern. The derivation corresponds to the coinductive interpretation of the standard big-step semantics rules [4,23], which may exhibit non-deterministic behavior as happens for this example; however, here the coaxiom plays a crucial role by filtering out all undesired values, and, thus, leaving only the value ∞ representing divergence; indeed, by employing also the coaxiom, finite derivation trees can be built for $e_\Delta \Rightarrow v_\infty$ only when $v_\infty = \infty$. By Corollary 3 we can get an infinite sequence of approximating derivation trees of arbitrarily increasing level:

$$\text{(coax)}\ \dfrac{}{e_\Delta \Rightarrow \infty}$$

$$\text{(app)}\ \dfrac{\text{(val)}\ \dfrac{}{\lambda x.x\ x \Rightarrow \lambda x.x\ x} \qquad \text{(val)}\ \dfrac{}{\lambda x.x\ x \Rightarrow \lambda x.x\ x} \qquad \text{(coax)}\ \dfrac{}{(x\ x)[x \leftarrow \lambda x.x\ x] \Rightarrow \infty}}{(x\ x)[x \leftarrow \lambda x.x\ x] = e_\Delta \Rightarrow \infty}$$

$$\vdots$$

As a consequence, in the inference system with the coaxiom a valid infinite derivation tree can be built for $e_\Delta \Rightarrow v_\infty$ only when $v_\infty = \infty$.

7 Related Work

Inference systems [1] are widely adopted to formally define operational semantics, language translations, type systems, subtyping relations, and other relevant judgments. Although inference systems have been introduced for dealing with inductive recursive definitions, in the last two decades several authors have focused on their coinductive interpretation.

Cousot and Cousot [14] define divergence by the coinductive interpretation of an inference system which extends the big-step operational semantics. The same approach is followed by other authors [16,23,30]. Leroy and Grall [23] analyze two kinds of coinductive big-step operational semantics for the call-by-value λ-calculus, and study their relationships with the small-step and denotational semantics, and their suitability for compiler correctness proofs. Coinductive big-step semantics is used as well to reason on cyclic objects stored in memory [24,26], and to prove type soundness in Java-like languages [4,6]. Coinductive inference systems are also considered in the context of type analysis and subtyping for object-oriented languages [7,9].

More recently, several solutions have been proposed to extend existing programming languages to support corecursion, and are, therefore, more focused on operational aspects, and their corresponding implementation issues; contributions can be found for all main computational paradigms: logic [5,21,25], functional [18,19], and object-oriented programming [10,11].

For the logic paradigm, the starting point is *coinductive logic programming* (coLP) [32], an extension of logic programming which provides both a declarative and a sound but not complete operational semantics for coinductive predicates, the former based on the notion of complete Herbrand base (finite and infinite terms) and greatest fixed point. However, only the standard coinductive interpretation is supported, and mixing between inductive and coinductive predicates is only partially supported through stratification. Structural resolution [21] is an extension of the operational semantics for coLP not limited to regular derivations. Other proposals [5,25] provide more flexible operational semantics. The notion of `finally` clause [5] has inspired our notion of coaxiom: `finally` clauses are user-defined facts that are resolved when an infinite, but regular, derivation is detected, in replacement of the standard coinductive semantics. Despite the existing strong correlation with this paper, the semantics of `finally` clauses does not always coincide with a fixed point of the one step inference operator. Similar considerations apply also to the work on coFJ [10,11], where `with` clauses play a role similar to that of `finally` clauses for coLP. A first attempt to provide a denotational model for this language, overtaken by the present work, has been provided in [8].

CoCaml [18,19] is an extension of OCaml where the semantics of recursive functions can be parametric in an equation solver which can be either predefined, or explicitly provided by the programmer to support corecursion. The intuition suggests that choosing a solver corresponds to choose a specific partial order, in such a way that the desired function is a fixed point in the corresponding CPO. Among the several proposed solvers, the pre-defined *iterator solver* has an expressive power similar to that of the `finally` and `with` clauses mentioned above.

As already mentioned, the spirit of our work is very different from that on CoCaml, since we do not aim to extend a practical language with corecursion, but, rather, to provide a very general framework which smoothly extends the well-known notion of inference system, and that could be used in many useful contexts, as shown in Sect. 6. On the other hand, definitions of higher order functions cannot be directly supported by inference systems. The foundation of CoCaml [20] is based on the theory of recursion in the framework of coalgebras. Our approach, instead, relies on the standard complete lattice of powersets, with set inclusion as partial order. In this way, a single and simple model based on classical results works uniformly for any possible recursive definition expressed in terms of a generalized inference system.

Recursive and well-founded coalgebras [12] are a framework for generalized structural recursion.

Completely iterative algebras [2] and corecursive and anti-founded algebras [12] are frameworks for generalized structural corecursion; iterative algebras correspond to the rational case as opposed to the coinductive one.

In *guarded recursion* [13,28] a judgment can be proved by also using recursively the judgment itself, provided that such recursive call is *guarded* by introduction rules. The goal, similar to ours, is to obtain a unique fixed point, however, there is no counterpart of the guard notion in the general framework of inference systems.

8 Conclusion

We have presented a generalized notion of inference system by introducing coaxioms, to support flexible definitions of judgments by structural recursion on non-well-founded datatypes.

Consequently, we have generalized the meta-theory of inference systems by providing two equivalent semantics, one based on fixed points in a complete lattice, and the other on the notion of proof tree. In the former case, the semantics of an inference system is the greatest fixed point of its corresponding one step inference operator, below the least pre-fixed point containing the coaxioms; in the latter case, the standard notion of proof tree for the coinductive case is generalized by requiring coaxioms to be applicable "at an infinite depth".

We have provided proof techniques for proving soundness and completeness results and shown their application to a range of different examples.

A compelling direction for further developments is exploring mechanization in proof assistants and other proof techniques [17] for coaxioms.

A necessarily not complete prototype meta-interpreter has been implemented in SWI-Prolog[7] to test the examples provided in Sects. 2 and 6. SWI-Prolog offers a natural support to regular terms (a.k.a. cyclic terms) through unification, but examples involving non-regular terms (or derivations) cannot terminate.

Extending the notion of coaxiom in the setting of object-oriented and functional programming is more challenging, because of the gap between the underlying theories.

We plan to investigate the dual notion studied here: one could consider the least fixed point above the greatest post-fixed point contained in the coaxioms, instead of the greatest fixed point below the least pre-fixed point containing the coaxioms. In particular, it would be interesting studying inference systems for which the two different semantics coincide.

References

1. Aczel, P.: An introduction to inductive definitions. In: Handbook of Mathematical Logic. North Holland (1977)
2. Adámek, J., Milius, S., Velebil, J.: Iterative algebras at work. Math. Struct. Comput. Sci. **16**(6), 1085–1131 (2006)
3. Ancona, D.: Regular corecursion in Prolog. In: SAC 2012 - ACM Symposium on Applied Computing, pp. 1897–1902 (2012)
4. Ancona, D.: Soundness of object-oriented languages with coinductive big-step semantics. In: Noble, J. (ed.) ECOOP 2012. LNCS, vol. 7313, pp. 459–483. Springer, Heidelberg (2012)
5. Ancona, D.: Regular corecursion in Prolog. Comput. Lang. Syst. Struct. **39**(4), 142–162 (2013)

[7] Available at http://www.disi.unige.it/person/AnconaD/Software/esop17artifact.zip.

6. Ancona, D.: How to prove type soundness of Java-like languages without forgoing big-step semantics. In: FTfJP 2014 - Formal Techniques for Java-like Programs, pp. 1:1–1:6. ACM Press (2014)
7. Ancona, D., Corradi, A.: Sound and complete subtyping between coinductive types for object-oriented languages. In: Jones, R. (ed.) ECOOP 2014. LNCS, vol. 8586, pp. 282–307. Springer, Heidelberg (2014)
8. Ancona, D., Dagnino, F., Zucca, E.: Towards a model of corecursion with default. In: FTfJP 2016 - Formal Techniques for Java-like Programs (2016)
9. Ancona, D., Lagorio, G.: Coinductive type systems for object-oriented languages. In: Drossopoulou, S. (ed.) ECOOP 2009. LNCS, vol. 5653, pp. 2–26. Springer, Heidelberg (2009)
10. Ancona, D., Zucca, E.: Corecursive featherweight Java. In: FTfJP 2012 - Formal Techniques for Java-like Programs (2012)
11. Ancona, D., Zucca, E.: Safe corecursion in coFJ. In: FTfJP 2013 - Formal Techniques for Java-like Programs (2012)
12. Capretta, V., Uustalu, T., Vene, V.: Corecursive algebras: a study of general structured corecursion. In: Oliveira, M.V.M., Woodcock, J. (eds.) SBMF 2009. LNCS, vol. 5902, pp. 84–100. Springer, Heidelberg (2009). doi:10.1007/978-3-642-10452-7_7
13. Coquand, T.: Infinite objects in type theory. In: Barendregt, H., Nipkow, T. (eds.) TYPES 1993. LNCS, vol. 806, pp. 62–78. Springer, Heidelberg (1994)
14. Cousot, P., Cousot, R.: Inductive definitions, semantics and abstract interpretations. In: ACM Symposium on Principles of Programming Languages, pp. 83–94. ACM Press (1992)
15. Grall, H.: Proving fixed point. In: FICS 2010 - Fixed Points in Computer Science (2010)
16. Hughes, J., Moran, A.: Making choices lazily. In: FPCA 1995 - Functional Programming Languages and Computer Architecture, pp. 108–119. ACM Press (1995)
17. Hur, C., Neis, G., Dreyer, D., Vafeiadis, V.: The power of parameterization in coinductive proof. In: ACM Symposium on Principles of Programming Languages, pp. 193–206 (2013)
18. Jeannin, J., Kozen, D., Silva, A.: CoCaml: programming with coinductive types. Technical report, Computing and Information Science, Cornell University, December 2012
19. Jeannin, J.-B., Kozen, D., Silva, A.: Language constructs for non-well-founded computation. In: Felleisen, M., Gardner, P. (eds.) ESOP 2013. LNCS, vol. 7792, pp. 61–80. Springer, Heidelberg (2013)
20. Jeannin, J., Kozen, D., Silva, A.: Well-founded coalgebras, revisited. Math. Struct. Comput. Sci. FirstView, 1–21 (2016)
21. Johann, P., Komendantskaya, E., Komendantskiy, V.: Structural resolution for logic programming. In: Technical Communications of (ICLP 2015) (2015)
22. Lassez, J., Nguyen, V.L., Sonenberg, L.: Fixed point theorems and semantics: a folk tale. Inf. Process. Lett. **14**(3), 112–116 (1982)
23. Leroy, X., Grall, H.: Coinductive big-step operational semantics. Inf. Comput. **207**(2), 284–304 (2009)
24. Leroy, X., Rouaix, F.: Security properties of typed applets. In: MacQueen, D.B., Cardelli, L. (eds.) ACM Symposium on Principles of Programming Languages, pp. 391–403. ACM Press (1998)
25. Mantadelis, T., Rocha, R., Moura, P.: Tabling, rational terms, and coinduction finally together!. Theory Pract. Log. Program. **14**(4–5), 429–443 (2014)

26. Milner, R., Tofte, M.: Co-induction in relational semantics. Theor. Comput. Sci. **87**(1), 209–220 (1991)
27. Nakata, K., Uustalu, T.: Trace-based coinductive operational semantics for while. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) TPHOLs 2009. LNCS, vol. 5674, pp. 375–390. Springer, Heidelberg (2009)
28. Pavlovic, D.: Guarded induction on final coalgebras. Electron. Notes Theor. Comput. Sci. **11**, 140–157 (1998)
29. Poulsen, C.B., Mosses, P.D.: Flag-based big-step semantics. J. Log. Algebr. Methods Program. (2016). http://www.sciencedirect.com/science/article/pii/S2352220816300311
30. Schmidt, D.A.: Trace-based abstract interpretation of operational semantics. Lisp Symb. Comput. **10**(3), 237–271 (1998)
31. Simon, L.: Extending logic programming with coinduction. Ph.D. thesis, University of Texas at Dallas (2006)
32. Simon, L., Bansal, A., Mallya, A., Gupta, G.: Co-logic programming: extending logic programming with coinduction. In: ICALP 2007 - International Colloquium on Automata, Languages and Programming, pp. 472–483 (2007)
33. Simon, L., Mallya, A., Bansal, A., Gupta, G.: Coinductive logic programming. In: Etalle, S., Truszczyński, M. (eds.) ICLP 2006. LNCS, vol. 4079, pp. 330–345. Springer, Heidelberg (2006)
34. Tarski, A.: A lattice-theoretical fixpoint theorem and its applications. Pac. J. Math. **5**(2), 285–309 (1955)