A Big Step from Finite to Infinite Computations

Davide Ancona[a], Francesco Dagnino[a], Jurriaan Rot[b], Elena Zucca[a]

[a] *DIBRIS, University of Genova, Italy*
[b] *Radboud University, The Netherlands*

Abstract

We provide a construction that, given a big-step semantics describing finite computations and their observations, extends it to include infinite computations as well. The basic idea is that the finite behaviour uniquely determines the infinite behaviour once observations and their composition operators are fixed. Technically, the construction relies on the framework of *inference systems with corules*. The effectiveness and scope of the approach are illustrated by several examples. The correctness is formally justified by proving that, starting from a big-step semantics equivalent to a reference small-step semantics, such equivalence is preserved by the construction.

Keywords: operational semantics, infinite behaviour, coinduction

The known is finite, the unknown infinite
– Thomas Henry Huxley

1. Introduction

The behaviour of programs or software systems can be described by the final *results* of computations, and/or their interactions with the context, also seen as *observations*. For instance, a function call can terminate and return a value, as well as have output effects during its execution.

Here, we deal with semantic definitions covering both results and observations. Often, such definitions are provided for *finite* computations only. Notably, in big-step style, infinite computations are simply not modelled, hence diverging and stuck terms are not distinguished. This becomes even more unsatisfactory if we have observations, since we would like to express that a non terminating program has a significant infinite behaviour.

Recently, examples of big-step semantics modeling divergence have been provided [1, 2] by means of *generalized inference systems* [3, 4], which allow *corules* to control coinduction. Indeed, modeling infinite behaviour by a purely coinductive interpretation of big-step rules would lead to spurious results [5] and undetermined observation, whereas, by adding appropriate corules, we can correctly get divergence (∞) as the only result, and a uniquely determined observation. This approach has been adopted in [1, 2] to design big-step definitions including infinite behaviour for lambda-calculus and a simple imperative Java-like language. However, in

Email addresses: `davide.ancona@unige.it` (Davide Ancona), `francesco.dagnino@dibris.unige.it` (Francesco Dagnino), `jrot@cs.ru.nl` (Jurriaan Rot), `elena.zucca@unige.it` (Elena Zucca)

such works the designer of the semantics is in charge of finding the appropriate corules, and this is a non-trivial task.

In this paper, we show a *construction* that extends a given big-step semantics, modeling finite computations, to include infinite behaviour as well, notably *generating appropriate corules*. The construction consists of two steps:

1. Starting from a monoid O modeling finite observations (e.g., finite traces), we construct an *ω-monoid* $\langle O, O_\infty \rangle$ also modeling infinite observations (e.g., infinite traces). The latter structure is a variation of ω-semigroup [6], including a *mixed product* composing a finite with a possibly infinite observation, and an *infinite product* mapping an infinite sequence of finite observations into a single one (possibly infinite).
2. Starting from an inference system defining a big-step judgment $c \Rightarrow \langle r, o \rangle$, with c configuration, $r \in R$ result, and $o \in O$ finite observation, we construct an inference system with corules defining an extended big-step judgment $c \Rightarrow \langle r_\infty, o_\infty \rangle$ with $r_\infty \in R_\infty = R + \{\infty\}$ and $o_\infty \in O_\infty$. The construction generates additional rules for *propagating divergence*, and corules for *introducing divergence* in a controlled way, obtained as instances of two patterns (co-unit) and (co-mul).

To show the effectiveness of our approach, we provide several instances of the framework, with different kinds of (finite) observations. Depending on the nature of such observations, instantiations of only (co-unit) or both should be added to obtain the intended infinite behaviour.

Finally, we consider the issue of formally justifying that the construction is correct. To this end, we assume the original big-step semantics to be equivalent to (finite sequences of steps in) a reference small-step semantics, and we show that, by applying the construction, we obtain an extended big-step semantics which is still equivalent to the small-step semantics, where we consider possibly infinite sequences of steps. As hypothesis, rather than just equivalence in the finite case (which would be not enough), we assume a set of *equivalence conditions* between individual big-step rules and the small-step relation.

This proof of equivalence holds for deterministic semantics; issues arising in the non-deterministic case and a possible solution are sketched in the conclusion.

Outline Sect. 2 is a quick introduction to inference systems with corules. Sect. 3 informally introduces our approach on a simple example. Sect. 4 describes the construction of ω-monoids, and Sect. 5 the extension of big-step semantics. Sect. 6 treats several significant examples. Sect. 7 contains the proof of equivalence. Related work is summarized in Sect. 8, and conclusions drawn in Sect. 9. The appendix gives more details on ω-monoids (Appendix A), proofs (Appendix B and C), and small-step semantics of the examples (Appendix D).

2. Inference systems with corules

First we recall standard notions on inference systems [7, 5]. Assuming a *universe* $\mathcal{U}$ of *judgments*, an *inference system* $\mathcal{I}$ is 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* (a.k.a. *conclusion*). A rule with an empty set of premises is an *axiom*. A *proof tree* (a.k.a. *derivation*) for a judgment j is a tree whose nodes are (labeled with) judgments in $\mathcal{U}$, j is the root, and there is a node c with children Pr only if there is a rule $\frac{Pr}{c}$. The *inductive* and the *coinductive interpretation* of $\mathcal{I}$, denoted $Ind(\mathcal{I})$ and

$\mathit{CoInd}(\mathcal{I})$, are the sets of judgments with respectively a finite[1] and a possibly infinite proof tree. Set-theoretically, let $F_{\mathcal{I}} : \wp(\mathcal{U}) \to \wp(\mathcal{U})$, $F_{\mathcal{I}}(S) = \{c \mid Pr \subseteq S, \frac{Pr}{c} \in \mathcal{I}\}$; call a set S *closed* if $F_{\mathcal{I}}(S) \subseteq S$, and *consistent* if $S \subseteq F_{\mathcal{I}}(S)$. Then $\mathit{Ind}(\mathcal{I})$ is the smallest closed set, and $\mathit{CoInd}(\mathcal{I})$ is the largest consistent set.

We recall now the notion of inference system with corules [3, 4], which mixes induction and coinduction in a specific way. For a set $S \subseteq \mathcal{U}$, let $\mathcal{I}|_S$ denote the inference system obtained from $\mathcal{I}$ by keeping only rules with consequence in S.

Definition 2.1 (Inference system with corules)**.** *An* inference system with corules*, or* generalized inference system*, is a pair $\langle \mathcal{I}, \mathcal{C}\rangle$ where $\mathcal{I}$ and $\mathcal{C}$ are inference systems, whose elements are called* rules *and* corules*, respectively. The* interpretation *$\mathit{Gen}(\mathcal{I},\mathcal{C})$ of such a pair is defined by $\mathit{Gen}(\mathcal{I},\mathcal{C}) = \mathit{CoInd}(\mathcal{I}|_{\mathit{Ind}(\mathcal{I}\cup\mathcal{C})})$.*

Thus, the interpretation $\mathit{Gen}(\mathcal{I},\mathcal{C})$ is basically *coinductive*, but restricted to a universe of judgements which is *inductively defined* by the (potentially) larger system $\mathcal{I} \cup \mathcal{C}$. In proof-theoretic terms, $\mathit{Gen}(\mathcal{I},\mathcal{C})$ is the set of judgments which have a possibly infinite proof tree in $\mathcal{I}$ whose nodes all have a finite proof tree in $\mathcal{I}\cup\mathcal{C}$, that is, the (standard) inference system consisting of rules and corules. We will write $\langle \mathcal{I}, \mathcal{C}\rangle \vdash j$ when j is derivable in $\langle \mathcal{I}, \mathcal{C}\rangle$, that is, $j \in \mathit{Gen}(\mathcal{I},\mathcal{C})$.

We illustrate these notions by a simple example. As usual, sets of rules are expressed by *meta-rules* with side conditions, and analogously sets of corules are expressed by *meta-corules* with side conditions. (Meta-)corules will be written with thicker lines, to be distinguished from (meta-)rules. The following inference system defines the maximal element of a list of natural numbers, where ϵ is the empty list, and $x{:}u$ the list with head x and tail u.

$$\frac{}{\mathit{max}(x{:}\epsilon, x)} \qquad \frac{\mathit{max}(u, y)}{\mathit{max}(x{:}u, z)}\, z = \max(x, y)$$

The inductive interpretation is defined only on finite lists, since for infinite lists an infinite proof is needed. However, the coinductive interpretation fails to be a function. For instance, let $L = 1 : 2 : 1 : 2 : 1 : 2 : \ldots$. Then any judgment $\mathit{max}(L, x)$ with $x \geq 2$ can be derived, as illustrated by the following examples.

$$\frac{\dfrac{\dfrac{\cdots}{\mathit{max}(L, 2)}}{\mathit{max}(2{:}L, 2)}}{\mathit{max}(1{:}2{:}L, 2)} \qquad \frac{\dfrac{\dfrac{\cdots}{\mathit{max}(L, 5)}}{\mathit{max}(2{:}L, 5)}}{\mathit{max}(1{:}2{:}L, 5)}$$

By adding a corule (in this case a coaxiom), we add a constraint which forces the greatest element to belong to the list, so that wrong results are "filtered out":

$$\frac{}{\mathit{max}(x{:}\epsilon, x)} \qquad \frac{\mathit{max}(u, y)}{\mathit{max}(x{:}u, z)}\, z = \max(x, y) \qquad \genfrac{}{}{2pt}{}{}{\mathit{max}(x{:}u, x)}$$

Indeed, the judgment $\mathit{max}(1{:}2{:}L, 2)$ has the infinite proof tree shown above, and each node has a finite proof tree in the inference system extended by the corule:

$$\frac{\dfrac{\dfrac{\cdots}{\mathit{max}(L, 2)}}{\mathit{max}(2{:}L, 2)}}{\mathit{max}(1{:}2{:}L, 2)} \qquad \frac{\genfrac{}{}{2pt}{}{}{\mathit{max}(2{:}L, 2)}}{\mathit{max}(1{:}2{:}L, 2)}$$

[1]Under the common assumption that sets of premises are finite, otherwise we should say a well-founded tree, that is, a tree with no infinite paths.

On the other hand, the judgment $max(1{:}2{:}L, 5)$ has the infinite proof tree shown above, but has *no finite proof tree* in the inference system extended by the corule. Indeed, since 5 does not belong to the list, the corule can never be applied. Hence, this judgment cannot be derived in the inference system with corules. We refer to [3, 1, 2, 4] for other examples.

Let $\langle \mathcal{I}, \mathcal{C} \rangle$ be a generalized inference system. The *bounded coinduction principle*, a generalization of the standard coinduction principle, can be used to prove *completeness* of $\langle \mathcal{I}, \mathcal{C} \rangle$ w.r.t. a set $\mathcal{S}$ (for "specification") of *valid* judgments.

Theorem 2.1 (Bounded coinduction)**.** *If the following two conditions hold:*

1. *$\mathcal{S} \subseteq Ind(\mathcal{I} \cup \mathcal{C})$, that is, each valid judgment has a finite proof tree in $\mathcal{I} \cup \mathcal{C}$;*
2. *$\mathcal{S} \subseteq F_{\mathcal{I}}(\mathcal{S})$, that is, each valid judgment is the consequence of an inference rule in $\mathcal{I}$ where all premises are in $\mathcal{S}$,*

then $\mathcal{S} \subseteq Gen(\mathcal{I}, \mathcal{C})$.

3. Our approach

Notation for sequences. Given a set X, we denote by $X^\star$, $X^\omega = \{\sigma \mid \mathbb{N} \to X\}$, and $X^\infty = X^\star \cup X^\omega$, respectively, the sets of finite, infinite, and possibly infinite sequences of elements of X. We write $x{:}u$ for concatenation of $x \in X$ with $u \in X^\infty$, $u \cdot v$ (or just $u\,v$) for concatenation of $u \in X^\star$ with $v \in X^\infty$, and ε for the empty sequence. Given a function $f\colon X \to Y$, we obtain functions $f^\star\colon X^\star \to Y^\star$ and $f^\omega\colon X^\omega \to Y^\omega$, defined by elementwise application of f. For $u \in X^\star$ and $v \in X^\infty$, we say that u is a prefix of v, denoted by $u \triangleleft v$, if $u \cdot z = v$ for some $z \in X^\infty$.

3.1. An example of semantics with observations

We illustrate our approach on a call-by-value λ-calculus with output. The top section of

$$\begin{array}{rcll}
e & ::= & v \mid x \mid e_1\,e_2 \mid \texttt{out}\,e & \text{expression} \\
u, v & ::= & i \mid \lambda x.e & \text{result = value} \\
o & ::= & v_1 \ldots v_n & \text{finite observation}
\end{array}$$

$$\begin{array}{rcll}
\ell & ::= & v \mid \varepsilon & \text{elementary observation} \\
\mathcal{E}[\,] & ::= & \square \mid \mathcal{E}[\,]\,e \mid (\lambda x.e)\,\mathcal{E}[\,] \mid \texttt{out}\,\mathcal{E}[\,] & \text{evaluation context}
\end{array}$$

$$(\beta)\,\frac{}{(\lambda x.e)\,v \xrightarrow{\varepsilon} e[x/v]} \qquad (\text{OUT})\,\frac{}{\texttt{out}\,v \xrightarrow{v} v} \qquad (\text{CTX})\,\frac{e \xrightarrow{\ell} e'}{\mathcal{E}[e] \xrightarrow{\ell} \mathcal{E}[e']}\ \mathcal{E}[\,] \neq \square$$

$$(\text{VAL})\,\frac{}{v \Rightarrow \langle v,\, \varepsilon \rangle} \qquad (\text{APP})\,\frac{\begin{array}{c} e_1 \Rightarrow \langle \lambda x.e,\, o_1 \rangle \\ e_2 \Rightarrow \langle v,\, o_2 \rangle \\ e[v/x] \Rightarrow \langle u,\, o \rangle \end{array}}{e_1\,e_2 \Rightarrow \langle u,\, o_1 \cdot o_2 \cdot o \rangle} \qquad (\text{OUT})\,\frac{e \Rightarrow \langle v,\, o \rangle}{\texttt{out}\,e \Rightarrow \langle v,\, o \cdot v \rangle}$$

Figure 1: λ-calculus with output: syntax and finite semantics

Fig. 1 contains the syntax. We assume infinite sets of *variables x* and *integer constants i*. Results are either integer constants or λ-abstractions. Beyond standard constructs, we add expressions of shape $\texttt{out}\,e$, which output the result of the evaluation of e. Correspondingly, observations are sequences of such outputs, and the semantics of an expression consists of both its final result and the whole observation produced during the computation.

The mid section contains, as reference, the small-step semantics, defined by a labelled transition system $e \xrightarrow{\ell} e'$. Labels are observations produced on a single step, called *elementary*, that is, either an output observation, or no observation, represented by the empty sequence ε. Rule (CTX) is the usual contextual closure, and evaluation contexts define the standard call-by-value left-to-right strategy.

The bottom section contains the same semantics in big-step style. As expected, the big-step judgment $e \Rightarrow \langle v, o\rangle$ directly computes the semantics (result and observation), whereas in the small-step style such semantics is obtained from a finite sequence of steps $e{=}e_0 \xrightarrow{\ell_1} \ldots \xrightarrow{\ell_n} e_n{=}v$ such that $o{=}\ell_1 \cdot \ldots \cdot \ell_n$.

3.2. Extending observations

First of all we enrich results by a special element ∞ denoting divergence, and observations by considering infinite output sequences:

$$\begin{array}{lcll} v_\infty & ::= & v \mid \infty & \text{result or divergence} \\ o_\infty & ::= & o \mid v_1 \ldots v_n \ldots & \text{observation} \end{array}$$

The latter is an instance of a general construction, formally defined in Sect. 4. Briefly, assuming that finite observations are a *monoid* $\langle O,\ *,\ \mathsf{u}\rangle$, with $*$ (sequentially) combining two observations, and the identity u, also called *unit*, modeling absence of observation, we construct an *ω-monoid* $\langle O,\ O_\infty\rangle$, where O_∞ models possibly infinite observations, with a *mixed product* $*_\infty\colon O \times O_\infty \to O_\infty$ combining a finite with a possibly infinite observation, and an *infinite product* $\mathsf{p}\colon O^\omega \to O_\infty$ mapping an infinite sequence of finite observations into a possibly infinite observation. For details and a proper definition, see Section 4.

In the example, the monoid is $\langle Val^\star,\ \cdot,\ \varepsilon\rangle$, and the construction just adds infinite output sequences. Formally, we obtain the ω-monoid $\langle Val^\star,\ Val^\infty\rangle$, where the mixed product is the concatenation of a finite with a possibly infinite sequence, still denoted by $\cdot$, and the infinite product returns the concatenation of an infinite number of finite sequences.

3.3. Extending big-step semantics

The judgment is modified into $e \Rightarrow \langle v_\infty, o_\infty\rangle$ to include divergence and infinite observations. Correspondingly, we extend the inference system, as formalized in Sect. 5. Here we informally explain the extension using the example.

Divergence propagation We first present the easier part, which is how to add rules for divergence propagation, shown in Fig. 2.

$$\text{(DIV-APP1)}\ \frac{e_1 \Rightarrow \langle \infty,\ o_\infty\rangle}{e_1\ e_2 \Rightarrow \langle \infty,\ o_\infty\rangle} \qquad \text{(DIV-APP2)}\ \frac{e_1 \Rightarrow \langle \lambda x.e,\ o\rangle \quad e_2 \Rightarrow \langle \infty,\ o_\infty\rangle}{e_1\ e_2 \Rightarrow \langle \infty,\ o\ \cdot o_\infty\rangle}$$

$$\text{(DIV-APP3)}\ \frac{e_1 \Rightarrow \langle \lambda x.e,\ o_1\rangle \quad e_2 \Rightarrow \langle v,\ o_2\rangle \quad e[v/x] \Rightarrow \langle \infty,\ o_\infty\rangle}{e_1\ e_2 \Rightarrow \langle \infty,\ o_1 \cdot o_2 \cdot o_\infty\rangle} \qquad \text{(DIV-OUT)}\ \frac{e \Rightarrow \langle \infty,\ o_\infty\rangle}{\mathtt{out}\ e \Rightarrow \langle \infty,\ o_\infty\rangle}$$

Figure 2: λ-calculus with output: adding divergence propagation

These rules are not arbitrary: they are constructed in a systematic manner starting from the original (meta-)rules. That is, for each original meta-rule, we consider premises as ordered from

Infinite proof tree for any $\Omega \equiv \omega\,\omega \Rightarrow \langle v_\infty,\ o_\infty\rangle$ in $\mathcal{I}$

$$\text{(APP/DIV-APP3)}\ \dfrac{\text{(VAL)}\ \dfrac{}{\omega \Rightarrow \langle \omega,\ \varepsilon\rangle} \qquad \text{(VAL)}\ \dfrac{}{\omega \Rightarrow \langle \omega,\ \varepsilon\rangle} \qquad \text{(APP/DIV-APP3)}\ \dfrac{\vdots}{\omega\,\omega \equiv (x\,x)[\omega/x] \Rightarrow \langle v_\infty,\ o_\infty\rangle}}{\omega\,\omega \Rightarrow \langle v_\infty,\ \varepsilon \cdot \varepsilon \cdot o_\infty \equiv o_\infty\ \rangle}$$

Finite proof tree for $\Omega \equiv \omega\,\omega \Rightarrow \langle \infty,\ \varepsilon\rangle$ in $\mathcal{I} \cup \mathcal{C}$

$$\text{(CO-EMPTY)}\ \dfrac{}{\Omega \Rightarrow \langle \infty,\ \varepsilon\rangle}$$

Figure 3: Proof trees for Ω

$$\text{(CO-EMPTY)}\ \dfrac{}{e \Rightarrow \langle \infty,\ \varepsilon\rangle} \qquad \text{(CO-OUT)}\ \dfrac{e \Rightarrow \langle v,\ o\rangle}{\texttt{out}\ e \Rightarrow \langle \infty,\ o \cdot v \cdot o_\infty\rangle}$$

Figure 4: λ-calculus with output: adding corules

left to right.[2] For each premise, say, the i-th, we add a meta-rule where the first $i - 1$ premises are kept as they are (hence, the corresponding computations converge), whereas the i-th premise requires the corresponding computation to diverge. In the conclusion, we get ∞ as result and the mixed product of the observations in the premises (in the given order) as observation; only the last observation is possibly infinite.

Divergence introduction The rules in Fig. 2 ensure that divergent computations, if any, are correctly propagated. To discuss how to correctly *introduce* divergent computations, consider, for instance, the term $\Omega = \omega\ \omega$, where $\omega = \lambda x.x\ x$. We should derive $\Omega \Rightarrow \langle \infty,\ \varepsilon\rangle$, and *only this* judgment, modeling that Ω diverges without producing any output. Similarly to the example *max* in Sect. 2, no judgment can be derived for Ω in the inductive interpretation, and, in the coinductive interpretation, an infinite proof tree exists for *any* judgment $\Omega \Rightarrow \langle v_\infty,\ o_\infty\rangle$, as shown[3] in Fig. 3, where (APP/DIV-APP3) means either (APP), if v_∞ is a value v, or (DIV-APP3), if $v_\infty = \infty$.

In summary, divergent terms have no result (are stuck) in the inductive interpretation, and a fully non-deterministic result in the coinductive interpretation. Our approach is to add appropriate corules, so that, as in the *max* example, we add constraints to filter out wrong judgments.

In the example, we add to the rules in Fig. 1 and Fig. 2 the corules shown in Fig. 4, which again are obtained in a systematic manner. Notably, they are special cases of two patterns, named (CO-UNIT) and (CO-MUL), which handle two different cases of divergent computations. We explain the role of these rules; they will be formally defined in Sect. 5 (Def. 5.3).

The (CO-UNIT) pattern handles the case where the computation produces a *finite*[4] observation o. In this case, a purely coinductive approach obtains any v_∞, and any observation of shape $o * o_\infty$, and the aim of the corule is to only allow $v_\infty = \infty$ and $o_\infty = \mathsf{u}$. In the example, we use the specific name (CO-EMPTY), since the unit is the empty sequence. In the Ω case, $o = \varepsilon$, and, with (CO-EMPTY), we derive only the judgment $\Omega \Rightarrow \langle \infty,\ \varepsilon\rangle$. Indeed, consider one of the proof trees in

[2]Assuming an order on premises is transparent with respect to the finite semantics, but relevant for the construction, see Sect. 7.3 for a detailed discussion.

[3]Here and in the following examples we add to the proof tree some comments (with a grey background) showing an equivalent expression, to help the reader.

[4]More precisely, *finitely generated* by elementary observations, as defined in Def. 4.3.

Infinite proof tree for $\hat{\Omega} \equiv \hat{\omega}\,(\mathtt{out}\,\hat{\omega}) \Rightarrow \langle v_\infty,\ o_\infty \rangle$ in $\mathcal{I}$, enforcing $o_\infty = \hat{\omega} \cdot o_\infty$

$$\text{(app/div-app3)}\,\dfrac{\text{(val)}\,\dfrac{}{\hat{\omega} \Rightarrow \langle \hat{\omega},\ \varepsilon \rangle} \qquad \text{(out)}\,\dfrac{\text{(val)}\,\dfrac{}{\hat{\omega} \Rightarrow \langle \hat{\omega},\ \varepsilon \rangle}}{\mathtt{out}\,\hat{\omega} \Rightarrow \langle \hat{\omega},\ \hat{\omega} \rangle} \qquad \text{(app/div-app3)}\,\dfrac{\vdots}{\hat{\Omega} \equiv \hat{\omega}\,(\mathtt{out}\,\hat{\omega}) \equiv (x\,(\mathtt{out}\,x))[\hat{\omega}/x] \Rightarrow \langle v_\infty,\ o_\infty \rangle}}{\hat{\omega}\,(\mathtt{out}\,\hat{\omega}) \Rightarrow \langle v_\infty,\ \varepsilon \cdot \hat{\omega} \cdot o_\infty \equiv \hat{\omega} \cdot o_\infty \ \rangle}$$

Finite proof tree for $\hat{\Omega} \equiv \hat{\omega}\,(\mathtt{out}\,\hat{\omega}) \Rightarrow \langle \infty,\ \widehat{o_\infty} \rangle$ in $\mathcal{I} \cup \mathcal{C}$, with $\widehat{o_\infty} = \hat{\omega}\ \ldots\ \hat{\omega}\ \ldots$

$$\text{(div-app2)}\,\dfrac{\text{(val)}\,\dfrac{}{\hat{\omega} \Rightarrow (\hat{\omega}, \varepsilon)} \qquad \text{(co-out)}\,\dfrac{\text{(val)}\,\dfrac{}{\hat{\omega} \Rightarrow \langle \hat{\omega},\ \varepsilon \rangle}}{\mathtt{out}\,\hat{\omega} \Rightarrow \langle \infty,\ \varepsilon \cdot \hat{\omega} \cdot \widehat{o_\infty} \rangle}}{\hat{\omega}\,(\mathtt{out}\,\hat{\omega}) \Rightarrow \langle \infty,\ \varepsilon \cdot \varepsilon \cdot \hat{\omega} \cdot \widehat{o_\infty} \equiv \widehat{o_\infty} \ \rangle}$$

Figure 5: Proof trees for $\hat{\Omega}$

Fig. 3, which have an infinite path. For each node of such an infinite path[5] the corules should allow a finite proof tree. If the path consists of infinite nodes $\Omega \Rightarrow \langle v,\ o_\infty \rangle$, for some v and o_∞, then the corules do not allow any finite proof tree for this judgment, since they all have ∞ in the conclusion. If it consists of infinite nodes $\Omega \Rightarrow \langle \infty,\ o_\infty \rangle$, for some o_∞, then it is easy to see that only for $\Omega \Rightarrow \langle \infty,\ \varepsilon \rangle$ there is a finite proof tree, shown in the bottom section of Fig. 3.

The (co-mul) pattern, instead, handles the case where the computation produces an infinite observation, since *infinitely many elementary non-unit observations are produced*. In this case, a purely coinductive approach obtains any v_∞; on the other side, the observation is uniquely determined by this infinite sequence. Consider, for instance, the term $\hat{\Omega} = \hat{\omega}\,(\mathtt{out}\,\hat{\omega})$, with $\hat{\omega} = \lambda x.(x\,(\mathtt{out}\,x))$, for which the small-step semantics produces the output sequence consisting of infinite occurrences of the value $\hat{\omega}$. In the top part of Fig. 5 we show the infinite proof trees which can be constructed for $\hat{\Omega}$. Each of them forces the constraint $o_\infty = \hat{\omega} \cdot o_\infty$, which is solved only for $\widehat{o_\infty} = \hat{\omega}\ \ldots\ \hat{\omega}\ \ldots$. Hence, the aim of the corule is, on one hand, to force $v_\infty = \infty$, and, on the other hand, to allow a finite proof tree for any node in the infinite path. Since in this infinite path there are infinite nodes producing an observation, it is enough to add a corule for such nodes. In our running example, we use the specific name (co-out), since the only original meta-rule producing a (non-unit) observation is (out). Exactly as in the Ω case, the corules do not allow finite proof trees for judgments of shape $\hat{\Omega} \Rightarrow \langle v,\ o_\infty \rangle$. On the other hand, they should allow a finite proof tree for the judgment $\hat{\Omega} \Rightarrow \langle \infty,\ \widehat{o_\infty} \rangle$, which can be obtained by corule (co-out), as shown in the bottom section of Fig. 5.

We conclude by explaining how meta-corules are added in a systematic way.

- We always add a meta-coaxiom (co-unit) with conclusion $e \Rightarrow \langle \infty,\ \mathsf{u} \rangle$.

- Assuming that in each meta-rule the observation in the conclusion is the product of the observations $o_1, \ldots, o_n$ in the premises, followed by an elementary observation o, we add, for each meta-rule where $o \neq \mathsf{u}$, a corresponding meta-corule with the same premises[6]

[5]For other nodes the condition is true since they have a finite proof tree using the rules.

[6]Here we simplify a little; in the construction in Sect. 5, we assume a distinguished last premise called *continuation*, which is not kept in the corule.

and conclusion $e \Rightarrow \langle \infty, o_1 * \cdots * o_n * o * o_\infty \rangle$. In the example, only (out) has a non-unit elementary observation, therefore (co-out) is the only added meta-corule.

A formal account of this general construction is given in Sect. 5.

4. From finite to infinite observations

In this section, we formally define ω-monoids. They are a variation of ω-semigroups used in algebraic language theory [6]. Further, we introduce a completion construction from monoids to ω-monoids.

Definition 4.1 (ω-monoid)**.** *An* ω-monoid *is a pair $\langle M, N\rangle$ of sets together with a function $* : M \times M \to M$, called* finite product*, a function $*_\infty : M \times N \to N$ called* mixed product*, a function $\mathsf{p} : M^\omega \to N$ called* infinite product*, and a constant $\mathsf{u} \in M$ called* unit*, satisfying the following properties:*

1. *$\langle M, *, \mathsf{u}\rangle$ is a monoid:*
   for all $x, y, z \in M$: $(x * y) * z = x * (y * z)$ and $x * \mathsf{u} = x = \mathsf{u} * x$.
2. *$*_\infty$ is a left action:*
   for all $x, y \in M$ and $\alpha \in N$: $x *_\infty (y *_\infty \alpha) = (x * y) *_\infty \alpha$ and $\mathsf{u} *_\infty \alpha = \alpha$.
3. *p respects the mixed product: for all $x \in M$ and $\sigma \in M^\omega$, $x *_\infty \mathsf{p}(\sigma) = \mathsf{p}(x{:}\sigma)$.*
4. *p satisfies the infinite associative law[7]: for all $\sigma, \tau \in M^\omega$, if $\sigma <: \tau$, then $\mathsf{p}(\sigma)=\mathsf{p}(\tau)$.*

A *homomorphism* from an ω-monoid $\langle M, N\rangle$ to an ω-monoid $\langle M', N'\rangle$ is a pair of functions $f : M \to M'$ and $g : N \to N'$ which preserve the unit and all three products in the expected way.

Example 4.1. *We list a few basic examples of ω-monoids.*

1. *A main example is the pair $\langle A^\star, A^\infty\rangle$ of finite sequences and possibly infinite sequences over an alphabet A. Finite and infinite product are given by concatenation. The mixed product concatenates finite sequences (on the left) with arbitrary sequences (on the right). There is no concatenation with infinite sequences on the left.*
2. *As a special case of* (1), $\langle \mathbb{N}, +, 0\rangle$ *extends to the ω-monoid $\langle \mathbb{N}, \mathbb{N} + \{\infty\}\rangle$.*
3. *The monoid $\langle \mathbb{N}, \vee, 0\rangle$ (where $n_1 \vee n_2$ is the join of n_1 and n_2 w.r.t. the standard order) also extends to an ω-monoid $\langle \mathbb{N}, \mathbb{N} + \{\infty\}\rangle$. Here, the infinite product computes the supremum of values occuring in a sequence.*
4. *Let $\mathcal{P}(X)$ be the powerset of a set X, and $\mathcal{P}_f(X)$ the finite powerset, i.e., $\mathcal{P}_f(X) = \{S \subseteq X \mid S \text{ finite}\}$. The monoid $\langle \mathcal{P}_f(X), \cup, \emptyset\rangle$ extends to an ω-monoid, with the second component given by the (full) powerset $\mathcal{P}(X)$.*

Given a set A, recall that $A^\star$ is a *free monoid*: if M is a monoid, then for every map $f\colon A \to M$ there is a unique monoid homomorphism $f^\sharp\colon A^\star \to M$ such that $f^\sharp(a) = f(a)$ for all $a \in A$. In particular, starting from the identity $\mathsf{id}_M : M \to M$, we get the map $\mathsf{id}_M^\sharp : M^\star \to M$, which interprets a sequence of elements as a unique element, by iterating the operation $*$ on the sequence. We abbreviate $\mathsf{id}_M^\sharp$ by it_M (for "iterator"), dropping the subscript when clear from the context. The following result, which follows by a slight adaptation of the proof for ω-semigroups [6], characterises the free ω-monoid.

[7]The order $<: \subseteq M^\omega \times M^\omega$ is defined by: for $\sigma = (x_i)_{i\in\mathbb{N}}$ and $\tau = (y_i)_{i\in\mathbb{N}}$, $\sigma <: \tau$ iff there exists a strictly increasing sequence $(k_i)_{i\in\mathbb{N}}$ with $k_0 = 0$ such that $x_i = y_{k_i} * \cdots * y_{k_{i+1}-1}$.

Proposition 4.1 (Free ω-monoid). *For every set A, ω-monoid $\langle M, N\rangle$ and map $f : A \to M$ there are unique maps $f^\sharp$, $f^\sharp_\infty$ such that $\langle f^\sharp, f^\sharp_\infty\rangle$ is a homomorphism of ω-monoids from $\langle A^\star, A^\infty\rangle$ to $\langle M, N\rangle$.*

Analogously to the monoid case, we abbreviate $\langle f^\sharp, f^\sharp_\infty\rangle$ by $\langle \mathsf{it}, \mathsf{it}_\infty\rangle$ when $f = \mathsf{id}_M$. We will use this iterator in Sect. 7.1 to obtain a single observation from a sequence of elementary observations.

We define $x \preceq_* y \Leftrightarrow \exists z \in M.\ x * z = y$. It is easy to check that this relation is reflexive and transitive, hence it is a pre-order. Furthermore, all left multiplications (the functions $y \mapsto x * y$ for all $x \in M$) are monotone w.r.t. $\preceq_*$. Finally, we denote by $\preceq_\infty \subseteq M \times N$ the relation defined similarly by the mixed product, i.e., $x \preceq_\infty \alpha \Leftrightarrow \exists \beta \in N.\ x *_\infty \beta = \alpha$.

We present now a construction which, given a monoid $\langle M, *, \mathsf{u}\rangle$, produces an ω-monoid $C_\infty(M) = \langle M, M_\infty\rangle$ called the *completion* of M. The completion is order-theoretic, and follows essentially from [8].

The construction takes existing limits into account, such that, when completing, e.g., the powerset $\mathcal{P}(X)$ of a set X (cf. Example 4.1), one obtains no new elements (intuitively, all limits already exist). To make this precise, we define a *left continuous monoid* as a monoid $\langle M, *, \mathsf{u}\rangle$ such that $\preceq_*$ is anti-symmetric (hence, a partial order) and, for each $x \in M$, the left multiplication function $m_x : M \to M$ defined by $m_x(y) = x * y$ is (Scott) continuous w.r.t. $\preceq_*$. More explicitly, for each increasing sequence $\sigma = (y_i)_{i\in\mathbb{N}}$ admitting a least upper bound $\sup\sigma$, the least upper bound $\sup(x * \sigma)$ exists (where $x * \sigma = (x * y_i)_{i\in\mathbb{N}}$), and $x * \sup\sigma = \sup(x * \sigma)$ A *continuous homomorphism* is a monoid homomorphism $f : M \to N$ which is (Scott) continuous w.r.t. $\preceq_*$. Every (underlying) monoid listed in Example 4.1 is left continuous.

The completion construction described below turns a left continuous monoid M into an ω-monoid $C_\infty(M) = \langle M, M_\infty\rangle$, where M_∞ is presented as a quotient of the set M^ω. We start by defining a relation $\sqsubseteq$ on M^ω. For $\sigma \in M^\omega$ we denote by $\mathsf{S}(\sigma)$ the closure of the set $\{\mathsf{it}(u) \mid u \triangleleft \sigma\}$ w.r.t. least upper bounds of increasing sequences. Then, for all $\sigma, \tau \in M^\omega$, we define $\sigma \sqsubseteq \tau \Leftrightarrow \forall x \in \mathsf{S}(\sigma).\ \exists y \in \mathsf{S}(\tau).\ x \preceq_* y$. This relation is a pre-order. We denote by $\equiv$ the induced equivalence relation, that is, $\sigma \equiv \tau \Leftrightarrow \sigma \sqsubseteq \tau \wedge \tau \sqsubseteq \sigma$.

Definition 4.2 (Completion). *The* completion *of a left continuous monoid $\langle M, *, \mathsf{u}\rangle$ is the ω-monoid $C_\infty(M) = \langle M, M_\infty\rangle$ where: $M_\infty = M^\omega/\equiv$, the mixed product $*_\infty : M \times M_\infty \to M_\infty$ is given by $x *_\infty [\tau]_\equiv = [x{:}\tau]_\equiv$, and the infinite product $\mathsf{p} : M^\omega \to M_\infty$ is given by $\mathsf{p}(\tau) = [\tau]_\equiv$.*

The fact that $C_\infty(M)$ is indeed an ω-monoid follows from A.1. In fact, the completion extends to a functor, mapping a continuous homomorphism f between left continuous monoids to an ω-monoid homomorphism $\langle f, f^\omega\rangle$.

Example 4.2. *The ω-monoid $C_\infty(A^\star)$ is isomorphic to the (free) ω-monoid $\langle A^\star, A^\infty\rangle$. In fact, the first three ω-monoids in Example 4.1 arise as completions of their underlying monoid. For the fourth ($\mathcal{P}_f(X)$) this is the case if X is countable.*

For an ω-monoid, $\langle M, N\rangle$ consider the map $\iota_M : M \to N$ defined by $\iota_M(x) = \mathsf{p}(x{:}\mathsf{u}^\omega)$, where u^ω is the infinite sequence of u's. This map is not necessarily injective, but, if M is left continuous, it is, hence, this is the case of the completion $\langle M, M_\infty\rangle$. In the sequel, we therefore identify M with its image in M_∞, leaving the inclusion ι implicit.

We conclude this section with the definition of a technical property of ω-monoids, stated for the completion in particular, which will be used to guide the extension of big-step semantics presented in Sect. 5. We note that the use of this property is quite subtle, and most of the extension can be understood without it. Given $G \subseteq M$, and M_G the submonoid of M generated by G, a sequence $\sigma = (x_i)_{i\in\mathbb{N}}$ in G is *trivial* if it is eventually always the unit. An element $\alpha \in M_\infty$ is a *limit product* of σ if, for all $k \in \mathbb{N}$, $x_0 * \cdots * x_k \preceq_\infty \alpha$. Note that, thanks to the properties of the infinite product, $\mathsf{p}(\sigma)$is a limit product of σ. Moreover, for $\alpha \in M_\infty$, we define the set $\mathcal{F}(\alpha)$ of *factors* of α in M_G:

$$\mathcal{F}(\alpha) = \{x \in M_G \mid \exists x' \in M_G.\ \exists\sigma \in M_G^\omega.\ \alpha = x' * x *_\infty \mathsf{p}(\sigma)\}\,.$$

We can now define the properties we need:

Definition 4.3. *Let M be a monoid. For a subset $G \subseteq M$, we say G has* unique limits *in M_∞ if each non-trivial sequence $\sigma = (x_i)_{i\in\mathbb{N}}$ in G has a unique limit product. Further, an element $\alpha \in M_\infty$ is called* finitely generated by G *if $\mathcal{F}(\alpha)$ is non-empty and finite.*

5. Extending big-step semantics

In this section, we formally define the construction which extends big-step semantics. We start with a few basic definitions related to big-step semantics. Assume in the following:

- a set C of *configurations* c;
- a set $R \subseteq C$ of *results* r;
- a left continuous monoid $\langle O, *, \mathsf{u}\rangle$ of (finite) *observations* o.

In the example of Sect. 3, configurations are just language terms. In general, they could include additional components, such as memory. We take an "operational" view of big-step semantics, as in, e.g., [9], where results are special configurations, rather than separate semantic entities. This allows a more direct comparison with small-step semantics in Sect. 7.

Definition 5.1. *A* judgement *j is a triple denoted by $c \Rightarrow \langle r,\ o\rangle$, where $c \in C$, $r \in R$ and $o \in O$. We set $C(j) = c$, $R(j) = r$ and $O(j) = o$.*

A rule *ρ in big-step semantics has shape*

$$\frac{j_1 \quad \dots \quad j_n \quad j_{n+1}}{c \Rightarrow \langle R(j_{n+1}),\ O(j_1) * \cdots * O(j_n) * o * O(j_{n+1})\rangle}$$

where $c \in C \setminus R$, $j_1, \dots, j_n$ are judgements called dependencies *(with $n \geq 0$), j_{n+1} is a judgement called* continuation *and $o \in O$ is the* elementary observation.

A big-step semantics *is a collection $\mathcal{I}$ of rules of the above form, together with a single axiom $\dfrac{}{r \Rightarrow \langle r,\ \mathsf{u}\rangle}$ for each result $r \in R$.*

The notation for judgements is extended to rules by setting $C(\rho) = c$, $R(\rho) = R(j_{n+1})$ and $O(\rho) = O(j_1) * \cdots * O(j_n) * o * O(j_{n+1})$.

A rule ρ as above models that the evaluation of c consists of the following steps: first, for any dependency j_i, in the given order, the configuration $C(j_i)$ is evaluated, producing the observation $O(j_i)$; then, the elementary observation o is emitted and the configuration $C(j_{n+1})$ in the continuation is evaluated; finally, the result of the continuation $R(j_{n+1})$ is returned, and all the partial observations are composed, following the evaluation order.[8] The restriction that an observation is

[8]The term *continuation*, even though overloaded, is chosen to suggest that, after evaluating dependencies (subterms), some actual computation takes place leading to a new term, from which the computation *continues*, as in a computational step in the small-step style.

only emitted before the continuation is a simplification: emitting observations before evaluating a dependency can be encoded by factorizing the semantics, by introducing an intermediate term, as it should be done in small-step style as well.

Note that there are axioms only for results, which have no other rules. Hence, the only derivable judgment for r is $r \Rightarrow \langle r,\ \mathsf{u} \rangle$, which we call a *trivial* judgment.

A big-step semantics as defined above forms an inference system, whose inductive interpretation is the semantics of interest. However, these rules carry slightly more information than standard inference rules. Notably, premises are a non-empty sequence, rather than a set, and the last premise plays a special role. Such additional structure does not affect the semantic relation defined by the rules, but is relevant to define the construction.

It will be useful in the technical development to denote a rule as in Definition 5.1 as above more concisely as

$$\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)\,,$$

referred to as the *inline* format of rules.

[Elena: moved here before the examples] The following assumption means that meta-variables for observations can be freely instantiated, that is, observations do not influence the evaluation process.

Assumption 1. *Throughout this paper, we assume the following, for any big-step semantics $\mathcal{I}$: for all $\rho = \mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$ in $\mathcal{I}$, $o_1, \ldots, o_{n+1} \in O$, there is $\rho' = \mathsf{rule}(j'_1 \ldots j'_n, j'_{n+1}, c, o)$ s.t., for all $i \in 1 \ldots n+1$, $j'_i = C(j_i) \Rightarrow \langle R(j_i),\ o_i \rangle$.*

Example 5.1. *The big-step semantics of lambda-calculus with output, in the inline format of rules, is as follows.*

$$\text{(APP)}\quad \mathsf{rule}(e_1 \Rightarrow \langle \lambda x.e,\ o_1 \rangle\ e_2 \Rightarrow \langle v,\ o_2 \rangle, e[v/x] \Rightarrow \langle u,\ o \rangle, e_1\, e_2, \varepsilon)$$
$$\text{(OUT)}\quad \mathsf{rule}(e \Rightarrow \langle v,\ o_1 \rangle, v \Rightarrow \langle v,\ o_2 \rangle, \mathtt{out}\, e, v)$$

We compare this to the original definition in Fig. 1. Axiom (VAL) *is omitted since it is assumed implicitly in Definition 5.1, and* (APP) *is the original meta-rule in inline format (the third premise is the continuation). In* (OUT)*, the original meta-rule had no explicit continuation, hence it is expressed in a slightly different way, with a dummy continuation $v \Rightarrow \langle v,\ o_2 \rangle$. The two versions are clearly equivalent since this judgment can only be derived for $o_2 = \mathsf{u}$.*

Example 5.2. *The semantics of an application $e_1\, e_2$ in Fig. 1 formalizes a strategy which first (1) evaluates e_1, then (2) checks that the value of e_1 is a λ-abstraction, finally (3) evaluates e_2. That is, left-to-right evaluation with early error detection. Other strategies can be obtained by adjusting (small-step and) big-step rules. Notably, right-to-left evaluation (3)-(1)-(2), expressed in small-step style by evaluation contexts $\mathcal{E}[\ ]\, v$ and $e\, \mathcal{E}[\ ]$, can be expressed by just swapping the first two premises, that is:*

$$\text{(APP-RIGHT)}\quad \mathsf{rule}(e_2 \Rightarrow \langle v,\ o_2 \rangle\ e_1 \Rightarrow \langle \lambda x.e,\ o_1 \rangle, e[v/x] \Rightarrow \langle u,\ o \rangle, e_1\, e_2, \varepsilon)$$

Left-to-right evaluation with late error detection (1)-(3)-(2), expressed in small-step style by evaluation contexts $\mathcal{E}[\ ]\, e$ and $v\, \mathcal{E}[\ ]$, can be expressed as follows:

$$\text{(APP-LATE)}\quad \mathsf{rule}(e_1 \Rightarrow \langle v_1,\ o_1 \rangle\ e_2 \Rightarrow \langle v_2,\ o_2 \rangle\ v_1 \Rightarrow \langle \lambda x.e,\ o_3 \rangle, e[v_2/x] \Rightarrow \langle u,\ o \rangle, e_1\, e_2, \varepsilon)$$

We now turn to the extension of a big-step semantics $\mathcal{I}$, which consists in the addition of rules for divergence propagation as well as corules to rule out spurious results, as shown in the example in Section 3. It depends on properties of $\mathcal{I}$ and the observation monoid O which corules are actually added—see the final construction in Definition 5.4. We start by describing the basic rules for divergence propagation and the relevant corules.

Definition 5.2 (Rules for divergence propagation)**.** *The set of rules $\mathcal{I}_\infty$ is obtained by extending $\mathcal{I}$ as follows. For each $\rho = \mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$, $i \in 1..n+1$, and $o_\infty \in O_\infty$, we add a rule*

$$\frac{j_1 \quad \ldots \quad j_{i-1} \quad C(j_i) \Rightarrow \langle \infty,\ o_\infty \rangle}{c \Rightarrow \langle \infty,\ O(j_1) * \cdots * O(j_{i-1}) * o' *_\infty o_\infty \rangle} \quad o' = \begin{cases} o & \text{if } i = n+1 \\ \mathsf{u} & \text{otherwise} \end{cases}$$

denoted by $\mathsf{div}(\rho, i, o_\infty)$ to highlight the relationship with the original rule ρ.

Intuitively, we consider the possibility that evaluation of $C(j_i)$ diverges for one of the premises j_i. In that case, the subsequent premises should be ignored and the configuration c in the conclusion should diverge as well.

Definition 5.3 (Corules patterns)**.** *The set of corules $C^{\mathcal{I}}_{\mathsf{u}}$ is defined as follows:*

$$\text{(CO-UNIT)}\ \frac{}{c \Rightarrow \langle \infty,\ \mathsf{u} \rangle}$$

The set of corules $C^{\mathcal{I}}_{m}$ is defined as follows:

$$\text{(CO-MUL)}\ \frac{j_1 \quad \ldots \quad j_n}{c \Rightarrow \langle \infty,\ O(j_1) * \cdots * O(j_n) * o *_\infty o_\infty \rangle} \quad \begin{array}{l} o_\infty \in O_\infty \\ \mathsf{rule}(j_1 \ldots j_n, j, c, o) \in \mathcal{I} \\ o \neq \mathsf{u} \end{array}$$

We denote by $\mathsf{co\text{-}mul}(\rho, o_\infty)$ the corule in $C^{\mathcal{I}}_{m}$ constructed from $\rho = \mathsf{rule}(j_1 \ldots j_n, j, c, o)$ with observation o_∞. The set of corules $C^{\mathcal{I}}$ is defined as the union $C^{\mathcal{I}}_{\mathsf{u}} \cup C^{\mathcal{I}}_{m}$.

Example 5.3. *Due to the side condition $o \neq \mathsf{u}$, for instance, for the λ-calculus with output, the above definition associates no corule* (CO-MUL) *with* (APP)*, whereas for* (OUT) *we obtain rule* (CO-OUT)*, see Fig. 4.*

The properties of the semantics extension strongly depends on how the ω-monoid $C_\infty(O) = \langle O,\ O_\infty \rangle$ behaves w.r.t. the *elementary observations* $\mathrm{E}_{\mathcal{I}}$ **[Davide: I would suggest using a notation more different from evaluation contexts $\mathcal{E}[\,]$ [Elena: I tried an alternative]]** produced in the semantics, defined by

$$\mathrm{E}_{\mathcal{I}} = \{o \mid \text{there is a rule } \rho = \mathsf{rule}(j_1, \ldots, j_n, j_{n+1}, c, o)\}\,.$$

Further, we define $O_{\mathcal{I}}$ as the submonoid of O generated by $\mathrm{E}_{\mathcal{I}}$. This submonoid contains all the observations produced by finite computations, as stated in the following lemma.

Lemma 5.1. *If $\langle \mathcal{I},\ \emptyset \rangle \vdash c \Rightarrow \langle r,\ o \rangle$, then $o \in O_{\mathcal{I}}$.*

We are now ready to define the extension of big-step semantics, using the constructions in Definitions 5.2 and 5.3.

Definition 5.4 (Extending big-step semantics)**.** *The* extension *of a big-step semantics $\mathcal{I}$ is the inference system with corules $\langle \mathcal{I}_\infty,\ C^{\mathcal{I}} \rangle$ if $\mathrm{E}_{\mathcal{I}}$ has unique limits, otherwise it is $\langle \mathcal{I}_\infty,\ C^{\mathcal{I}}_{\mathsf{u}} \rangle$.*

In the above definition, if $\mathrm{E}_{\mathcal{I}}$ has unique limits, we take both corules patterns of Def. 5.3 and the construction is correct, as will be formally shown in Sect. 7.4 (Theorem 7.1 and Theorem 7.5). Indeed, given a computation which produces infinitely many elementary non-unit observations, pattern (CO-MUL) allows *any limit product* of this sequence (Lemma 7.6), hence the uniqueness of limits is needed to avoid spurious observations. The property of unique limits holds in many significant examples, see the next section, notably for the common case where observations are traces.

If this property does not hold, we can keep only pattern (CO-UNIT) and, in this way, the construction is correct for computations with observations which are finitely generated by $\mathrm{E}_{\mathcal{I}}$ (Theorem 7.3 and Theorem 7.2), as defined above. This is satisfactory in many examples, see again the next section.

We conclude with a basic but important property of semantics with corules. A set C of corules is called *conservative* if all rules are of the form

$$\frac{Pr}{c \Rightarrow \langle \infty,\, o_\infty \rangle}$$

For instance, $C^{\mathcal{I}}_{\mathsf{u}}$, $C^{\mathcal{I}}_{\mathsf{m}}$, and $C^{\mathcal{I}}$ are conservative sets of corules. By adding a conservative set of corules to $\mathcal{I}_\infty$ we do not affect finite computations, as formally stated below. The important consequence is that, for converging judgments, we can reason by standard inductive techniques (see, e.g., Lemma 7.2 and Lemma 7.5).

Theorem 5.1 (Conservativity)**.** *For each $\langle \mathcal{I}_\infty, C\rangle$ with C conservative, $\langle \mathcal{I}_\infty, C\rangle \vdash c \Rightarrow \langle r,\, o\rangle$ iff $\langle \mathcal{I}, \emptyset\rangle \vdash c \Rightarrow \langle r,\, o\rangle$.*

6. Examples of instantiation of the construction

In this section, we consider several examples, with different underlying monoids of finite observations. The reference small-step semantics are reported in Appendix D. For all examples, the original big-step semantics is equivalent to finite small-step computations.[9] For simplicity, we directly show the (possibly simplified) meta-rules obtained by the construction, using the following convention: non-bold for original meta-rules, bold black for added meta-(co)rules, bold gray for extended meta-rules (merging original and added meta-rules).

I/O events The first example, in Fig. 6, is a slight extension of the λ-calculus in Sect. 3: besides `out` e, we add the construct `in` to read input values. Single observations are no longer just values, but I/O events of shape either `in` v (value v has been read) or `out` v (value v has been output). The monoid of finite observations is $\{\mathtt{in}\, v, \mathtt{out}\, v \mid v \text{ value}\}^\star$, that is, the free monoid as in Sect. 3, but on top of a different set of single observations, and the ω-monoid completion adds infinite sequences of events.

The grammar also defines (divergence) propagation contexts $\mathcal{D}[\ ]$ [2] with one hole at fixed depth 1 to allow a more concise presentation of the meta-rules added for divergence propagation (see comments to rule (DIV) below). Meta-rules (VAL), (OUT), and (IN) are original meta-rules; (VAL) and (OUT) are analogous to those in Fig. 1. However, here configurations have shape $(e;\sigma)$ where σ is an infinite sequence of values modeling the input stream. In meta-rule (IN), a value is read from such a stream, emitting the corresponding elementary observation.

Meta-rule (APP) is the merge of two different meta-rules: the original one, analogous to (APP) in Fig. 1, and that added for divergence propagation from the third premise, analogous to (DIV-APP3) in Fig. 2. To this aim, the meta-variable w ranges over pairs of shape either $\langle(v;\sigma),\, o\rangle$,

[9] As can be proved by showing that the equivalence conditions of Sect. 7.3 hold.

$$
\begin{array}{rcll}
e & ::= & v \mid x \mid e_1\,e_2 \mid \mathtt{in} \mid \mathtt{out}\,e & \text{expression} \\
u, v & ::= & i \mid \lambda x.e & \text{value} \\
\theta & ::= & \mathtt{in}\,v \mid \mathtt{out}\,v & \text{I/O event} \\
o & ::= & \theta_1 \ldots \theta_n & \text{finite observation} \\
o_\infty & ::= & o \mid \theta_1 \ldots \theta_n \ldots & \text{observation} \\
\mathcal{D}[\;] & ::= & \Box\,e \mid \mathtt{out}\,\Box & \text{(divergence) propagation context}
\end{array}
$$

$$
\text{(VAL)}\;\frac{}{(v;\sigma) \Rightarrow \langle (v;\sigma),\ \varepsilon\rangle}
\qquad
\text{(OUT)}\;\frac{(e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\ o\rangle}{(\mathtt{out}\,e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\ o \cdot (\mathtt{out}\,v)\rangle}
$$

$$
\text{(IN)}\;\frac{}{(\mathtt{in};v{:}\sigma) \Rightarrow \langle (v;\sigma),\ \mathtt{in}\,v\rangle}
\qquad
\text{(APP)}\;\frac{\begin{array}{c}(e_1;\sigma_1) \Rightarrow \langle (\lambda x.e;\sigma_2),\ o_1\rangle \\ (e_2;\sigma_2) \Rightarrow \langle (v;\sigma_3),\ o_2\rangle\end{array} \quad (e[x \leftarrow v];\sigma_3) \Rightarrow w}{(e_1\ e_2;\sigma_1) \Rightarrow o_1 \cdot o_2 \cdot w}
$$

$$
\text{(DIV-APP2)}\;\frac{(e_1;\sigma_1)\Rightarrow\langle(\lambda x.e;\sigma_2),\ o\rangle \quad (e_2;\sigma_2)\Rightarrow\langle\infty,\ o_\infty\rangle}{(e_1\ e_2;\sigma_1)\Rightarrow\langle\infty,\ o\ \cdot o_\infty\rangle}
\qquad
\text{(DIV)}\;\frac{(e;\sigma) \Rightarrow \langle\infty,\ o_\infty\rangle}{(\mathcal{D}[e];\sigma) \Rightarrow \langle\infty,\ o_\infty\rangle}
$$

$$
\text{(CO-EMPTY)}\;\genfrac{}{}{2pt}{}{}{(e;\sigma) \Rightarrow \langle\infty,\ \varepsilon\rangle}
\qquad
\text{(CO-OUT)}\;\genfrac{}{}{2pt}{}{(e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\ o\rangle}{(\mathtt{out}\,e;\sigma_1) \Rightarrow \langle\infty,\ o \cdot (\mathtt{out}\,v) \cdot o_\infty\rangle}
$$

$$
\text{(CO-IN)}\;\genfrac{}{}{2pt}{}{}{(\mathtt{in};v{:}\sigma) \Rightarrow \langle\infty,\ (\mathtt{in}\,v) \cdot o_\infty\rangle}
$$

Figure 6: λ-calculus with I/O: meta-(co)rules generated by the construction

or $\langle\infty,\ o_\infty\rangle$; accordingly, $o' \cdot w$ denotes either $\langle(v;\sigma),\ o' \cdot o\rangle$ or $\langle\infty,\ o' \cdot o_\infty\rangle$. Meta-rule (DIV-APP2) is analogous to that in Fig. 2, added for divergence propagation from the second premise of (APP). Thanks to propagation contexts, the remaining meta-rule (DIV) represents those added for divergence propagation from the first premise of both (APP) and (OUT), analogously to (DIV-APP1) and (DIV-OUT) in Fig. 2.

The meta-corules (CO-EMPTY) and (CO-OUT) are analogous to those in Fig. 4, obtained as special cases of (CO-UNIT) and (CO-MUL) defined in Sect. 5, respectively, where the latter pattern is applied to meta-rule (OUT). The meta-corule (CO-IN) is obtained by applying the pattern (CO-MUL) to meta-rule (IN).

In this example, as in that of Sect. 3, by adding both the (CO-UNIT) and the (CO-MUL) patterns, as shown above, we get the expected semantics, that is, the same derived from the small-step computations. Notably, completeness holds adding both patterns (Theorem 7.1), and soundness holds since the monoid of finite observations has unique limits (Theorem 7.5).

I/O costs In the next example, the language is the same, but single observations are the (time) costs associated with each I/O operation. This could be easily generalized to other constructs, e.g., considering also the costs for function application; however, by considering I/O operations only, we can show that our construction leads to exactly the same meta-rules as the previous example, modulo the used monoid of finite observations.

This monoid is $\langle\mathbb{R}_{\geq 0},\ +,\ 0\rangle$, that is, non-negative real numbers with addition; the only infinite observation added by the completion is ∞, corresponding to diverging series, with the obvious behavior w.r.t. the mixed product.

The meta-rules in Fig. 7 differ from those in Fig. 6 mainly for the employed ω-monoids, and few other details. Namely, meta-variables c and c_∞ range over $\mathbb{R}_{\geq 0}$ (finite observations) and $\mathbb{R}_{\geq 0} + \{\infty\}$ (possibly infinite observations), and the semantics is parametric in the two functions $c_{\mathtt{in}} : \mathit{Val} \to \mathbb{R}_{\geq 0}$ and $c_{\mathtt{out}} : \mathit{Val} \to \mathbb{R}_{\geq 0}$ assigning costs to in and out operations, respectively, depending on the input/output value. The meta-corule corresponding to the pattern (co-unit) in Sect. 5 has been named (co-zero). As in Fig. 6, we overload notation by adopting the same symbol (+ in this case) for both finite and mixed product. In this case, the meta-variable w ranges over pairs of shape either $\langle (v;\sigma),\, c\rangle$, or $\langle \infty,\, c_\infty\rangle$; accordingly, $c' + w$ denotes either $\langle (v;\sigma),\, c' + c\rangle$ or $\langle \infty,\, c' + c_\infty\rangle$.

$$\text{(val)}\frac{}{(v;\sigma) \Rightarrow \langle (v;\sigma),\, 0\rangle} \qquad \text{(out)}\frac{(e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\, c\rangle}{(\mathtt{out}\, e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\, c + c_{\mathtt{out}}(v)\rangle}$$

$$\text{(in)}\frac{}{(\mathtt{in}; v{:}\sigma) \Rightarrow \langle (v;\sigma),\, c_{\mathtt{in}}(v)\rangle}$$

$$\text{(app)}\frac{(e_1;\sigma_1) \Rightarrow \langle (\lambda x.e;\sigma_2),\, c_1\rangle \quad (e_2;\sigma_2) \Rightarrow \langle (v;\sigma_3),\, c_2\rangle \quad (e[x \leftarrow v];\sigma_3) \Rightarrow w}{(e_1\, e_2;\sigma_1) \Rightarrow c_1 + c_2 + w}$$

$$\textbf{(div-app2)}\frac{(e_1;\sigma_1) \Rightarrow \langle (\lambda x.e;\sigma_2),\, c\rangle \quad (e_2;\sigma_2) \Rightarrow \langle \infty,\, c_\infty\rangle}{(e_1\, e_2;\sigma_1) \Rightarrow \langle \infty,\, c\ + c_\infty\rangle} \qquad \textbf{(div)}\frac{(e;\sigma) \Rightarrow \langle \infty,\, c_\infty\rangle}{(\mathcal{D}[e];\sigma) \Rightarrow \langle \infty,\, c_\infty\rangle}$$

$$\textbf{(co-zero)}\frac{}{(e;\sigma) \Rightarrow \langle \infty,\, 0\rangle} \qquad \textbf{(co-out)}\frac{(e;\sigma_1) \Rightarrow \langle (v;\sigma_2),\, c\rangle}{(\mathtt{out}\, e;\sigma_1) \Rightarrow \langle \infty,\, c + c_{\mathtt{out}}(v) + c_\infty\rangle}$$

$$\textbf{(co-in)}\frac{}{(\mathtt{in}; v{:}\sigma) \Rightarrow \langle \infty,\, c_{\mathtt{in}}(v) + c_\infty\rangle}$$

Figure 7: λ-calculus with I/O costs: meta-(co)rules generated by the construction

As in the previous example, by adding both the (co-unit) and (co-mul) patterns we get the expected semantics. Completeness is again ensured by Theorem 7.1. Soundness (Theorem 7.5) holds under the following assumption on the cost functions: $0 < \inf\{c_{\mathtt{in}}(v), c_{\mathtt{out}}(v) \mid v \in \mathit{Val}, 0 < c_{\mathtt{in}}(v), c_{\mathtt{out}}(v)\}$; that is, non-zero costs for I/O operations cannot be arbitrarily close to zero. This ensures that the set $\mathrm{E}_{\mathcal{I}}$ of elementary observations produced in the semantics has unique limits. This is a reasonable assumption: it means that diverging programs performing infinite I/O operations with non-zero costs cannot have a finite cost.

Executed branches We consider a λ-calculus with labelled conditional expressions and Boolean values b, and a semantics useful to reason about branch coverage. The syntax is as follows.

$$\begin{array}{rcll} e & ::= & v \mid x \mid e_1\, e_2 \mid e \mathbin{?_a} e_1 : e_2 & \text{expression} \\ u, v & ::= & b \mid \lambda x.e & \text{value} \\ \mathcal{D}[\,] & ::= & \Box\, e \mid \Box \mathbin{?_a} e_1 : e_2 & \text{propagation context} \end{array}$$

We assume each conditional expression $e \mathbin{?_a} e_1 : e_2$ in a program to be associated with a unique label a ranging over a countably infinite set $\mathcal{A}$ of labels, so that $a.\mathit{true}$ and $a.\mathit{false}$ denote the unique addresses inside the program of the *then* and *else* branches e_1 and e_2, respectively.

Here finite observations are the sets of the addresses of the branches executed by a program, represented by the monoid $\langle \mathcal{P}_f(\{a.\mathit{true}, a.\mathit{false} \mid a \in \mathcal{A}\}),\, \cup,\, \emptyset\rangle$ (see point 4 in Example 4.1). Again, in the meta-rules, the symbol $\cup$ denotes both the finite and the mixed product. The meta-variable w ranges over pairs of shape either $\langle v,\, A\rangle$, or $\langle \infty,\, A_\infty\rangle$, and, if $w = \langle v_\infty,\, A_\infty\rangle$, then $A \cup w$ denotes $\langle v_\infty,\, A \cup A_\infty\rangle$. Similarly to meta-rule (app), meta-rules (if-f) and (if-t) (name in gray bold) are obtained by merging two different meta-rules: the original one for the finite semantics of conditional expressions, and the meta-rule representing those added for divergence propagation in the second premise.

For what concerns meta-corules, in this example we add only the (co-unit) pattern, because adding (co-mul) would be unsound. Indeed, the completion produces the full powerset $\mathcal{P}(\{a.\mathit{true}, a.\mathit{false} \mid$

$$\text{(VAL)}\ \frac{}{v \Rightarrow \langle v,\ \emptyset\rangle} \qquad \text{(APP)}\ \frac{e_1 \Rightarrow \langle \lambda x.e,\ A_1\rangle \quad e_2 \Rightarrow \langle v,\ A_2\rangle \quad e[x \leftarrow v] \Rightarrow w}{e_1\ e_2 \Rightarrow A_1 \cup A_2 \cup w}$$

$$\text{(IF-F)}\ \frac{e \Rightarrow \langle \mathit{false},\ A\rangle \quad e_2 \Rightarrow w}{e\ ?_a\ e_1 : e_2 \Rightarrow A \cup \{a.\mathit{false}\} \cup w} \qquad \text{(IF-T)}\ \frac{e \Rightarrow \langle \mathit{true},\ A\rangle \quad e_1 \Rightarrow w}{e\ ?_a\ e_1 : e_2 \Rightarrow A \cup \{a.\mathit{true}\} \cup w}$$

$$\text{(DIV-APP2)}\ \frac{e_1 \Rightarrow \langle \lambda x.e,\ A\rangle \quad e_2 \Rightarrow \langle \infty,\ A_\infty\rangle}{e_1\ e_2 \Rightarrow \langle \infty,\ A \cup A_\infty\rangle} \qquad \text{(DIV)}\ \frac{e \Rightarrow \langle \infty,\ A_\infty\rangle}{\mathcal{D}[e] \Rightarrow \langle \infty,\ A_\infty\rangle}$$

$$\text{(CO-EMPTY)}\ \frac{}{e \Rightarrow \langle \infty,\ \emptyset\rangle}$$

Figure 8: λ-calculus with conditional: meta-(co)rules generated by the construction

$a \in \mathcal{A}\}$)), and the set of elementary observations $E_\mathcal{I}$ has no unique limits. However, since every program has a finite set of branches, it is easy to see that (even infinite) small-step computations produce only observations which are finitely generated by $E_\mathcal{I}$, hence Theorem 7.2 gives completeness of the big-step semantics. And, because we have only the (CO-UNIT) pattern, by Theorem 7.3 the big-step semantics is sound.

Maximum heap size In this last example we consider an imperative extension of the call-by-value λ-calculus with heap references which can be explicitly deallocated. The syntax is as follows.

$$\begin{array}{rcll} e & ::= & v \mid x \mid e_1\, e_2 \mid \texttt{ref}\ e \mid !\, e \mid e_1 = e_2 \mid \texttt{free}\ e & \text{expression} \\ v & ::= & \iota \mid \lambda x.e & \text{value} \\ \mathcal{D}[\,] & ::= & \square\, e \mid \texttt{ref}\ \square \mid !\, \square \mid \texttt{free}\ \square \mid \square = e & \text{propagation context} \end{array}$$

Values are either references ι or λ-abstractions. The syntax includes expressions of shape $\texttt{ref}\ e$ creating a new reference initialized with the value of e, $!\,e$ dereferencing the reference denoted by e, $e_1 = e_2$ updating the reference denoted by e_1 with the value of e_2, and $\texttt{free}\ e$ deallocating the reference denoted by e.

In this case we are interested in observing the maximum size of the heap used by a program: finite observations are the monoid $\langle \mathbb{N},\ \vee,\ 0\rangle$ (see point 3 in Example 4.1). This monoid can be employed whenever observing the maximum number of used resources, independently from the notion of resource (heap locations, files, locks, etc.). The completion is the ω-monoid $\langle \mathbb{N},\ \mathbb{N} + \{\infty\}\rangle$, whose infinite product computes the supremum of values in a given sequence.

As before, the same symbol $\vee$ denotes both the finite and the mixed product. The meta-variable w ranges over pairs of shape either $\langle (v;\mathcal{H}),\ s\rangle$, or $\langle \infty,\ s_\infty\rangle$; accordingly, $s' \vee w$ denotes either $\langle (v;\mathcal{H}),\ s' \vee s\rangle$ or $\langle \infty,\ s' \vee s_\infty\rangle$.

Configurations have shape $(e;\mathcal{H})$, where a heap $\mathcal{H}$ is a finite map from references to values. Heap extension is denoted by $\mathcal{H} \uplus \{\iota \mapsto v\}$ (where ι is not in the domain of $\mathcal{H}$); $|\mathcal{H}|$ denotes the cardinality of the domain of $\mathcal{H}$, i.e., its size[10]. If the computation converges, then $\langle (v;\mathcal{H}),\ s\rangle$ is returned (a value v, a heap $\mathcal{H}$, and a maximum size s); if it diverges, then a pair $\langle \infty,\ s_\infty\rangle$ is returned.

As in the previous example, the set $E_\mathcal{I}$ of the elementary observations produced in the semantics has no unique limits. Hence, we keep only the meta-coaxiom (CO-ZERO) corresponding to the pattern (CO-UNIT) to avoid unsoundness. The big-step semantics is sound by Theorem 7.3; however, as opposed to the previous example, the infinitely generated observation[11] ∞ is not obtained,

[10]With the simplifying assumption that the size does not depend on the values in the heap.

[11]Produced by diverging programs allocating infinite references, without deallocating them.

$$\text{(VAL)}\ \frac{}{(v;\mathcal{H}) \Rightarrow \langle (v;\mathcal{H}),\ 0\rangle} \qquad \text{(REF)}\ \frac{(e;\mathcal{H}_1) \Rightarrow \langle (v;\mathcal{H}_2),\ s\rangle}{(\texttt{ref}\, e;\mathcal{H}_1) \Rightarrow \langle (\iota;\mathcal{H}_2 \uplus \{\iota \mapsto v\}),\ s \vee (1 + |\mathcal{H}_2|)\rangle}$$

$$\text{(DEREF)}\ \frac{(e;\mathcal{H}_1) \Rightarrow \langle (\iota;\mathcal{H}_2),\ s\rangle}{(!\, e;\mathcal{H}_1) \Rightarrow \langle (v;\mathcal{H}_2),\ s \vee |\mathcal{H}_2|\rangle}\ \mathcal{H}_2(\iota) = v$$

$$\text{(FREE)}\ \frac{(e;\mathcal{H}_1) \Rightarrow \langle (\iota;\mathcal{H}_2),\ s\rangle}{(\texttt{free}\, e;\mathcal{H}_1) \Rightarrow \langle (v;\mathcal{H}_3),\ s \vee |\mathcal{H}_3|\rangle}\ \mathcal{H}_2 = \mathcal{H}_3 \uplus \{\iota \mapsto v\}$$

$$\text{(UPD)}\ \frac{(e_1;\mathcal{H}_1) \Rightarrow \langle (\iota;\mathcal{H}_2),\ s_1\rangle \quad (e_2;\mathcal{H}_2) \Rightarrow \langle (v;\mathcal{H}_3),\ s_2\rangle}{(e_1 = e_2;\mathcal{H}_1) \Rightarrow \langle (v;\mathcal{H}_2 \uplus \{\iota \mapsto v\}),\ s_1 \vee s_2 \vee |\mathcal{H}_3|\rangle}\ \mathcal{H}_3 = \mathcal{H}_2 \uplus \{\iota \mapsto v'\}$$

$$\text{(APP)}\ \frac{(e_1;\mathcal{H}_1) \Rightarrow \langle (\lambda x.e;\mathcal{H}_2),\ s_1\rangle \quad (e_2;\mathcal{H}_2) \Rightarrow \langle (v;\mathcal{H}_3),\ s_2\rangle \quad (e[x \leftarrow v];\mathcal{H}_3) \Rightarrow w}{(e_1\ e_2;\mathcal{H}_1) \Rightarrow s_1 \vee s_2 \vee |\mathcal{H}_3| \vee w}$$

$$\textbf{(DIV-UPD)}\ \frac{(e_1;\mathcal{H}_1) \Rightarrow \langle (\iota;\mathcal{H}_2),\ s\rangle \quad (e_2;\mathcal{H}_2) \Rightarrow \langle \infty,\ s_\infty\rangle}{(e_1 = e_2;\mathcal{H}_1) \Rightarrow \langle \infty,\ s \vee s_\infty\rangle} \qquad \textbf{(DIV)}\ \frac{(e;\mathcal{H}) \Rightarrow \langle \infty,\ s_\infty\rangle}{(\mathcal{D}[e];\mathcal{H}) \Rightarrow \langle \infty,\ s_\infty\rangle}$$

$$\textbf{(DIV-APP2)}\ \frac{(e_1;\mathcal{H}_1) \Rightarrow \langle (\lambda x.e;\mathcal{H}_2),\ s\rangle \quad (e_2;\mathcal{H}_2) \Rightarrow \langle \infty,\ s_\infty\rangle}{(e_1\ e_2;\mathcal{H}_1) \Rightarrow \langle \infty,\ s \vee s_\infty\rangle} \qquad \textbf{(CO-ZERO)}\ \frac{}{(e;\mathcal{H}) \Rightarrow \langle \infty,\ 0\rangle}$$

Figure 9: λ-calculus with references: meta-(co)rules generated by the construction

hence the semantics is not complete. However, by Theorem 7.2 the semantics is complete for finitely generated observations; since the only infinitely generated observation is ∞, the only case where the big-step semantics "gets stuck", hence does not return any result, as opposite to the small-step semantics, is for programs that would require an infinite heap to run. This is acceptable, as such programs are always doomed to crash.

7. Preservation of equivalence

In this section we prove the correctness of the construction in Sect. 5. To formulate and prove correctness, we assume a reference small-step semantics, where the definition of both finite and infinite computations is straightforward. Correctness then means that, starting from a big-step semantics equivalent to finite small-step computations, we get an extended big-step semantics equivalent to possibly infinite small-step computations.

This main correctness result requires several assumptions on the big-step semantics. In particular, we assume that it is *deterministic*. Both determinism and equivalence with finite small-step computations need to be expressed, rather than globally on the semantic relation, at the level of individual rules, since such properties should be preserved by the construction, which handles single rules.

We start by describing small-step semantics in our context.

7.1. Small-step semantics for (in)finite computations

A small-step semantics with observations is defined by a *labelled transition system*, that is, a set of *steps* $q \xrightarrow{o} q'$, for $q, q' \in C$, $o \in O$. Then:

- $c \rightsquigarrow \langle r,\ o\rangle$ means that there is a finite computation from c to r, and o is the interpretation as single observation of the finite sequence of the elementary observations of single steps.

Formally, let $\rightarrow^{\star} \subseteq C \times O^{\star} \times C$ be inductively defined by: $c \xrightarrow{\varepsilon}^{\star} c$, and if $c \xrightarrow{o} c'$ and $c' \xrightarrow{u}^{\star} c''$ then $c \xrightarrow{o:u}^{\star} c''$. Then the relation $\rightsquigarrow \subseteq C \times R \times O$ is defined as follows:

$$c \rightsquigarrow \langle r,\, o\rangle \Leftrightarrow \exists u \in O^{\star}.\ c \xrightarrow{u}^{\star} r \wedge \mathsf{it}(u) = o$$

- $c \rightsquigarrow \langle \infty,\, o_\infty\rangle$ means that there is an infinite computation starting from c, and o is the interpretation as single observation of the infinite sequence of the elementary observations of single steps. Formally, let $\rightarrow^{\omega} \subseteq C \times O^{\omega}$ be coinductively defined[12] by:

$$\frac{c' \xrightarrow{\sigma}^{\omega}}{c \xrightarrow{o:\sigma}^{\omega}}\ c \xrightarrow{o} c'$$

  Then the relation $\rightsquigarrow \subseteq C \times O_\infty$ is defined as follows:

$$c \rightsquigarrow \langle \infty,\, o_\infty\rangle \Leftrightarrow \exists \sigma \in O^{\omega}.\ c \xrightarrow{\sigma}^{\omega} \wedge \mathsf{it}_\infty(\sigma) = o_\infty$$

7.2. Determinism assumptions

Determinism is expressed at the level of single rules as follows:

Assumption 2. *For each ρ, ρ' such that $c{=}C(\rho) = C(\rho')$, the following holds:*

1. $\rho{=}\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$ *and* $\rho'{=}\mathsf{rule}(j'_1 \ldots j'_n, j'_{n+1}, c, o')$
2. *for all $k \in 1..n+1$, if, for all $h < k$, $R(j_h) = R(j'_h)$, then $C(j_k) = C(j'_k)$ and, if $k = n+1$, $o = o'$.*

To show that the above assumption actually models determinism, let us consider the (meta-)rule for application in a lambda-calculus where configurations are pairs $e|\mu$ with μ auxiliary structure, e.g., memory, modified by some constructs, so that the evaluation order is relevant.

$$\text{(APP)}\frac{e_1|\mu \Rightarrow \langle \lambda x.e|\mu_1,\, o_1\rangle \quad e_2|\mu_1 \Rightarrow \langle v|\mu_2,\, o_2\rangle \quad e[v/x]|\mu_2 \Rightarrow \langle u|\mu',\, o\rangle}{e_1\, e_2|\mu \Rightarrow \langle u|\mu',\, o_1 \cdot o_2 \cdot \varepsilon \cdot o\rangle}$$

Assumption 2.1 requires that all the rules with the same configuration in the consequence have the same number n of dependencies ($n = 2$ in the example), and this clearly holds, since they are all instances of (APP). However, for a fixed $e_1\, e_2|\mu$ in the consequence, there are infinitely many rules which can be obtained by instantiating the meta-variables. Assumption 2.2 imposes the following constraints, expressed in the meta-rule by using the same meta-variable:

- ($k = 1$) the configuration in the first premise is uniquely determined
- ($k = 2$) the configuration in the second premise is uniquely determined by the result of the first premise
- ($k = 3$) the configuration in the third premise is uniquely determined by the results of the first two premises. In (APP) we assume ε as elementary observation, in general the elementary observation as well could depend on the results of the first two premises.

Finally note that, accordingly with Assumption 1, configurations are *not* determined by previous observations, as expressed in the meta-rule by the fact that meta-variables for observations are freely instantiated.

[12]That is, the largest relation such that, if $c \xrightarrow{\sigma}^{\omega}$, there are o, σ' and c' s.t. $\sigma = o{:}\sigma'$, $c \xrightarrow{o} c'$ and $c' \xrightarrow{\sigma}^{\omega}$.

We present two lemmas, holding under Assumption 2, used in later proofs. The first states that big-step semantics is actually deterministic. Note that, by conservativity (Theorem 5.1), this ensures that any conservative set of corules preserves determinism for finite computations. Another source of non-determinism could be an infinite computation conflicting with a finite one, and the second lemma states that this is prevented as well.

Lemma 7.1. *If $\langle \mathcal{I}, \emptyset\rangle \vdash c \Rightarrow \langle r, o\rangle$ and $\langle \mathcal{I}, \emptyset\rangle \vdash c \Rightarrow \langle r', o'\rangle$, then $r{=}r'$ and $o{=}o'$.*

Lemma 7.2. *If $\langle \mathcal{I}_\infty, C\rangle \vdash c \Rightarrow \langle r, o\rangle$ with C conservative, then there is no $o_\infty \in O_\infty$ such that $\langle \mathcal{I}_\infty, C\rangle \vdash c \Rightarrow \langle \infty, o_\infty\rangle$.*

We proceed with a third assumption on the big-step semantics: well-foundedness of the predecessor relation. Its formulation relies on Assumption 2, and several auxiliary concepts that we need to introduce first. These will also be used in the equivalence conditions of the following section.

Definition 7.1. *For $\rho{=}\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$, set $\mathsf{f}(\rho)$ the first index in $1..n$ such that $C(j_i) \notin R$, if any, otherwise $\mathsf{f}(\rho) = n + 1$. We say ρ is*

- computational *if all dependencies are trivial (hence $\mathsf{f}(\rho) = n + 1$);*
- contextual *if $\mathsf{f}(\rho) \leq n$, and for all $k < \mathsf{f}(\rho)$, j_k is trivial;*
- stuck*, if, for some $k < \mathsf{f}(\rho)$, j_k is not trivial.*

In the third case, $C(j_k) \in R$ but j_k is *not* of shape $r \Rightarrow \langle r, \mathsf{u}\rangle$, hence the premise j_k cannot be derived, and the rule is not applicable. Note that an inline meta-rule, for instance (app) in the above example, can have instantiations which are computational (if e_1 and e_2 have shape $\lambda x.e$ and v, respectively), contextual (if either e_1 or e_2 are not values), or stuck (e.g., if e_1 is an integer constant i.)

Under Assumption 2, configurations can be classified analogously to rules. Notably, c is *stuck* if all its rules are stuck, that is, $C(\rho) = c$ implies ρ stuck; otherwise, c is *reduced* if all its non-stuck rules $\rho{=}\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$ are computational ($\mathsf{f}(\rho){=}n + 1$), *compound* if they are contextual ($\mathsf{f}(\rho){\leq}n$). Intuitively, a configuration is reduced if there are no dependencies to be evaluated, compound otherwise. In the above example, a configuration $(\lambda x.e)\, e'|\mu$ where e' is not a value is compound.

We define the *predecessor relation* pr on configurations as follows: c' is a *predecessor* of c if there is a contextual rule ρ for c such that $c' = C(j_i)$ where $i = \mathsf{f}(\rho)$. By Assumption 2, for any compound c there is a unique predecessor c', which we denote by $\mathsf{pr}(c)$. Intuitively, c' is the (sub)configuration which needs to be evaluated first. The predecessor relation allows us to formulate the following assumption, which abstractly models that "program parts" of configurations have an inductive structure.

Assumption 3. *The relation pr on configurations is* well-founded.

From now on, we assume $\mathcal{I}$ to satisfy Assumptions 1, 2 and 3.

7.3. Equivalence conditions

As anticipated, to prove that equivalence with small-step computations is preserved by the construction, the equivalence in the finite case is not enough as hypothesis. To see this, consider the small-step semantics of application in Fig. 1, where, for $e_1\, e_2$ such that e_1 reduces to an integer constant, reduction gets immediately stuck, and the big-step semantics with late error detection in Example 5.2:

$$\text{(APP-LATE)}\quad \mathsf{rule}(e_1 \Rightarrow \langle v_1,\ o_1\rangle\ e_2 \Rightarrow \langle v_2,\ o_2\rangle\ v_1 \Rightarrow \langle \lambda x.e,\ o_3\rangle, e[v_2/x] \Rightarrow \langle u,\ o\rangle, e_1\, e_2, \varepsilon)$$

Considering only finite computations, such two semantics are equivalent: in particular, no result can be derived for $e_1\, e_2$ in the big-step semantics as well, since a judgment $v \Rightarrow \langle \lambda x.e,\ o_3\rangle$ can only be derived if $v = \lambda x.e$. However, this no longer holds for infinite computations: if e_1 evaluates to an integer constant, and e_2 diverges, then we get a stuck computation in small-step semantics, whereas by applying the construction we get, among others, the big-step rule

$$\text{(DIV-APP-LATE2)}\ \frac{e_1 \Rightarrow \langle v,\ o\rangle \quad e_2 \Rightarrow \langle \infty,\ o_\infty\rangle}{e_1\ e_2 \Rightarrow \langle \infty,\ o\ \cdot o_\infty\rangle}$$

hence we get non-termination. Intuitively, in order equivalence to be preserved by the construction, individual big-step rules should "match" the small-step relation, whereas this is not the case for rule (APP-LATE). The following *equivalence conditions* formalize this required matching.

To express the conditions in a convenient way, we introduce an auxiliary relation on rules:

Definition 7.2. *For rules ρ and ρ', we write $\rho \xrightarrow{o} \rho'$ if*

$\rho = \mathsf{rule}(j_1 \ldots j_i \ldots j_n, j, c, o')$ *and* $i = \mathsf{f}(\rho)$
$\rho' = \mathsf{rule}(j'_1 \ldots j'_i\, j_{i+1} \ldots j_n, j, c', o')$ *and, for all* $k < i$, j_k *and* j'_k *are trivial,*
$c \xrightarrow{o} c'$, $C(j_i) \xrightarrow{o} C(j'_i)$, $O(j_i) = o * O(j'_i)$ *and* $R(j_i) = R(j'_i)$.

Note that ρ is a contextual rule. For instance, consider the following small-step transitions in the lambda-calculus in Fig. 1:

$$(\lambda x.x)\ \mathtt{out}\ \mathtt{out}\ 1 \xrightarrow{1} (\lambda x.x)\ \mathtt{out}\ 1 \xrightarrow{1} (\lambda x.x)\ 1 \xrightarrow{\varepsilon} 1$$

the first two are obtained by rule (CTX) from $\mathtt{out}\ 1 \xrightarrow{1} 1$, the last by rule (APP). Correspondingly, there are transitions $\rho_0 \xrightarrow{1} \rho_1 \xrightarrow{1} \rho_2$ where

$$\rho_0 = \mathsf{rule}(\lambda x.x \Rightarrow \langle \lambda x.x,\ \varepsilon\rangle\ \mathtt{out}\ \mathtt{out}\ 1 \Rightarrow \langle 1,\ 1 \cdot 1\rangle, 1 \Rightarrow \langle 1,\ \varepsilon\rangle, (\lambda x.x)\ \mathtt{out}\ \mathtt{out}\ 1, \varepsilon)$$

$$\rho_1 = \mathsf{rule}(\lambda x.x \Rightarrow \langle \lambda x.x,\ \varepsilon\rangle\ \mathtt{out}\ 1 \Rightarrow \langle 1,\ 1\rangle, 1 \Rightarrow \langle 1,\ \varepsilon\rangle, (\lambda x.x)\ \mathtt{out}\ 1, \varepsilon)$$

$$\rho_2 = \mathsf{rule}(\lambda x.x \Rightarrow \langle \lambda x.x,\ \varepsilon\rangle\ 1 \Rightarrow \langle 1,\ \varepsilon\rangle, 1 \Rightarrow \langle 1,\ \varepsilon\rangle, (\lambda x.x)\ 1, \varepsilon)$$

An important property of this relation is that, if $\rho \xrightarrow{o} \rho'$, then the number of dependencies still to be evaluated cannot increase. More precisely: if $C(j'_i)$ is a result, as in the transition $\rho_1 \xrightarrow{1} \rho_2$, then $\mathsf{f}(\rho') > \mathsf{f}(\rho)$; if $C(j'_i)$ is not a result, as in the transition $\rho_0 \xrightarrow{1} \rho_1$, then $\mathsf{f}(\rho') = \mathsf{f}(\rho)$. In the latter case, ρ' is still contextual, and $\mathsf{pr}(C(\rho')) = C(j'_i)$.

We can now formulate the equivalence conditions.

Condition 1 (Enough rules)**.** *If $c \xrightarrow{o} c'$, then there exists $\rho = \mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o')$ such that:*

1. *either ρ is computational, $o' = o$, and $C(j_{n+1}) = c'$*
2. *or ρ is contextual, and there exists ρ' s.t. $C(\rho') = c'$ and $\rho \xrightarrow{o} \rho'$.*

This condition requires, for each small-step transition $c \xrightarrow{o} c'$, the existence of a corresponding big-step rule. This requirement is essential for completeness.

For instance, for $(\lambda x.x)$ `out out` $1 \xrightarrow{1} (\lambda x.x)$ `out` 1, there is the contextual rule ρ_0 and the transition $\rho_0 \xrightarrow{1} \rho_1$. For $(\lambda x.x)\ 1 \xrightarrow{\varepsilon} 1$, there is the computational rule ρ_2 with continuation 1 and elementary observation ε.

Condition 2 (Forward compatibility)**.** *For each non-stuck ρ=$\mathsf{rule}(j_1 \ldots j_n, j, c, o)$:*

1. *if ρ is computational, then $c \xrightarrow{o} C(j)$*
2. *if ρ is contextual with $i = \mathsf{f}(\rho) \leq n$, j_i=$c_i \Rightarrow \langle r_i,\ o_i\rangle$, then:*
   if $c_i \xrightarrow{\hat{o}} c'_i$ and $o_i = \hat{o} * o'_i$, then there exists ρ'=$\mathsf{rule}(j'_1 \ldots j'_n, j', c', o')$ s.t. $\rho \xrightarrow{\hat{o}} \rho'$ and j'_i=$c'_i \Rightarrow \langle r_i,\ o'_i\rangle$.

This condition is in some way symmetric to the previous one: for each computational rule, there must be a corresponding (computational) small-step transition; for each contextual rule ρ, if the predecessor c_i has a compatible[13] small-step transition $\hat{o}$, then the rule ρ, hence in particular the configuration $C(\rho)$, has the same small-step transition.

For instance, for the non-stuck (contextual) rule ρ_1, where `out` $1 \xrightarrow{1} 1$, there is the transition $\rho_1 \xrightarrow{1} \rho_2$. For the computational rule ρ_2, there is the small-step transition $(\lambda x.x)\ 1 \xrightarrow{\varepsilon} 1$.

By iterating Condition (2), this requirement can be extended to the global semantics as follows:

> for each $\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$, if, for all $i \in 1..n+1$, $C(j_i) \rightsquigarrow \langle R(j_i),\ O(j_i)\rangle$, then $c \rightsquigarrow \langle (R(j_{n+1}),\ O(j_1) * \cdots * O(j_n) * o * O(j_{n+1})\rangle$.

Condition 3 (Backward compatibility)**.** *If $c \xrightarrow{o} c'$, then, for each non-stuck $\rho' = \mathsf{rule}(j'_1 \ldots j'_n, j'_{n+1}, c', o'')$, there exists $\rho = \mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o')$ such that:*

1. *either ρ is computational, $o' = o$, and j_{n+1}=$c' \Rightarrow \langle R(\rho'),\ O(\rho')\rangle$*
2. *or ρ is contextual, and $\rho \xrightarrow{o} \rho'$.*

This last condition requires the big-step rules to be "backward closed" w.r.t. the transition relation.

For instance, for $(\lambda x.x)$ `out` $1 \xrightarrow{1} (\lambda x.x)\ 1$, for the non-stuck rule ρ_2 there is the contextual rule ρ_1 such that $\rho_1 \xrightarrow{1} \rho_2$.

In the following we will assume all the above conditions to hold. The following lemmas state that they actually model equivalence of big-step semantics with finite small-step computations. In this case, proof techniques are pretty standard: for completeness we use arithmetic induction on the length of the sequence of steps and a subject expansion lemma (Lemma 7.3), while for soundness we use induction on big-step rules, which is applicable thanks to Theorem 5.1.

Lemma 7.3. *If $\langle \mathcal{I},\ \emptyset\rangle \vdash c' \Rightarrow \langle r,\ o'\rangle$ and $c \xrightarrow{o} c'$, then $\langle \mathcal{I},\ \emptyset\rangle \vdash c \Rightarrow \langle r,\ o * o'\rangle$.*

Lemma 7.4 (Completeness for convergence)**.** *If $c \rightsquigarrow \langle r,\ o\rangle$, then $\langle \mathcal{I},\ \emptyset\rangle \vdash c \Rightarrow \langle r,\ o\rangle$.*

Lemma 7.5 (Soundness for convergence)**.** *If $\langle \mathcal{I},\ \emptyset\rangle \vdash c \Rightarrow \langle r,\ o\rangle$ then $c \rightsquigarrow \langle r,\ o\rangle$.*

[13]That is, its label "can be seen as first part" of the observation in the big-step judgment.

7.4. Preservation of equivalence for infinite computations

This section presents the formal results about the preservation of equivalence. As already stated above, we assume a big-step semantics $\mathcal{I}$ satisfying Assumptions 1, 2 and 3, and we assume compatibility with a small-step semantics through Conditions 1, 2 and 3 as formulated in the previous subsection.

First we discuss how *completeness* for finite computations (Lemma 7.4) is extended to infinite ones. The following theorem states completeness for $\mathcal{C}^{\mathcal{I}}$, that is, the extension obtained by both the (co-unit) and (co-mul) patterns. The result is stated only for infinite computations, since by Theorem 5.1 we derive the same converging judgements as before, hence Lemma 7.4 is enough.

Theorem 7.1 (Completeness)**.** *If* $c \rightsquigarrow \langle \infty, o_\infty \rangle$*, then* $\langle \mathcal{I}_\infty, \mathcal{C}^{\mathcal{I}} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$*.*

An additional completeness result characterizes infinite computations which can be derived by restricting corules to the set $\mathcal{C}^{\mathcal{I}}_{\mathsf{u}}$ of those of shape (co-unit). Recall from Sect. 5 that $\mathrm{E}_{\mathcal{I}}$ is the set of the elementary observations produced in $\mathcal{I}$, $O_{\mathcal{I}}$ the submonoid of O generated by $\mathrm{E}_{\mathcal{I}}$, and observations finitely generated (by $\mathrm{E}_{\mathcal{I}}$) are those with a non-empty and finite set of factors in $O_{\mathcal{I}}$ (see Def. 4.3). Then, the completeness result for $\mathcal{C}^{\mathcal{I}}_{\mathsf{u}}$ can be stated as follows.

Theorem 7.2 (Completeness for $\mathcal{C}^{\mathcal{I}}_{\mathsf{u}}$)**.** *If* $c \rightsquigarrow \langle \infty, o_\infty \rangle$ *and* o_∞ *is finitely generated by* $\mathrm{E}_{\mathcal{I}}$*, then* $\langle \mathcal{I}_\infty, \mathcal{C}^{\mathcal{I}}_{u} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$*.*

Now we discuss how *soundness* for finite computations (Lemma 7.5) is extended to infinite ones. Soundness is always preserved by restricting corules to the set $\mathcal{C}^{\mathcal{I}}_{\mathsf{u}}$, and in such case, as stated above, completeness can be kept as well if the produced observations are finitely generated, see the third example of Sect. 6.

Theorem 7.3 (Soundness for $\mathcal{C}^{\mathcal{I}}_{\mathsf{u}}$)**.** *If* $\langle \mathcal{I}_\infty, \mathcal{C}^{\mathcal{I}}_{u} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$*, then* $c \rightsquigarrow \langle \infty, o_\infty \rangle$*.*

Let us now consider the pattern (co-mul). We formally motivate that, as shown by the examples of Sect. 6, in this case soundness requires unique limits (see Def. 4.3).

Let $\mathcal{C}$ be a set of corules, and $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$ by an infinite proof tree t. We can associate with t the infinite sequence σ_t of finite observations produced by the computation. Formally, for each level i of t there is a rule $\mathsf{div}(\rho_i, k_i, o_\infty^{i+1})$ with $\rho_i = \mathsf{rule}(j_1^i \ldots j_{n_i}^i, j^i, c_i, o_i')$, and we define $\sigma_t = (o_i)_{i \in \mathbb{N}} \in O^\omega$ as follows:

$$o_i = \begin{cases} O(j_1^i) * \cdots * O(j_{k_i-1}^i) & k_i \leq n_i \\ O(j_1^i) * \cdots * O(j_{n_i}^i) * o_i' & k_i = n_i + 1 \end{cases}$$

Set $o_\infty^0 = o_\infty$, we get, for all $i \in \mathbb{N}$, $o_\infty^i = o_i *_\infty o_\infty^{i+1}$ and $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash c_i \Rightarrow \langle \infty, o_\infty^i \rangle$ by a proof tree t_i such that $\sigma_{t_0} = \sigma_t$ and $\sigma_{t_i} = o_i{:}\sigma_{t_{i+1}}$. Moreover, o_∞ is a limit product of σ_t, that is, for all $n \in \mathbb{N}$, $o_0 * \cdots * o_n \preceq_\infty o_\infty$. Actually we have the following result.

Lemma 7.6. *If* $\langle \mathcal{I}_\infty, \mathcal{C}^{\mathcal{I}} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$ *holds by a derivation* t*,* σ_t *is non-trivial and* o'_∞ *is a limit product of* σ_t*, then* $\langle \mathcal{I}_\infty, \mathcal{C}^{\mathcal{I}} \rangle \vdash c \Rightarrow \langle \infty, o'_\infty \rangle$*.*

This lemma shows why adding the pattern (co-mul) could be not sound: for each derivation t for a judgement $c \Rightarrow \langle \infty, o_\infty \rangle$, where σ_t is non-trivial, we can derive $c \Rightarrow \langle \infty, o'_\infty \rangle$ for all limit products o'_∞ of σ_t. Hence, an easy sufficient condition to ensure soundness is to require such limit to be unique.

By Lemma 5.1, given an infinite derivation t in $\mathcal{I}_\infty$, σ_t belongs to $O_{\mathcal{I}}^\omega$ by definition. Then, for soundness it is enough that $O_{\mathcal{I}}$ (in fact, thanks to the next proposition, $\mathrm{E}_{\mathcal{I}}$), has unique limits.

Proposition 7.4. *$O_{\mathcal{I}}$ has unique limits if and only if $E_{\mathcal{I}}$ has unique limits.*

Theorem 7.5 (Soundness). *If $\langle \mathcal{I}_\infty, C^{\mathcal{I}} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$ and $E_{\mathcal{I}}$ has unique limits, then $c \rightsquigarrow \langle \infty, o_\infty \rangle$.*

8. Related work

The notion of ω-monoid is a variation of the more standard ω-semigroups used in algebraic language theory [6]. Algebraic structures with a similar aim are proposed more often in the context of type systems, where the notion corresponding to observation is called *effect*. Such effect systems are traditionally commutative, hence effects typically form a bounded join semilattice, where the join is used to overapproximate different execution branches. This allows general formulations to study common properties, see, e.g., [10]. More recently, *sequential* effect systems have been proposed [11], which take into account evaluation order as in our case. However, differently from our observations, effects in type systems do not need to be "infinite", since they model static approximations.

Concerning operational semantics modeling divergence, we follow a stream of work dating back to [12], who proposed a stratified approach, investigated in [5] as well, with a separate judgment for divergence, defined coinductively. In this way, however, rules are duplicated, and there is no unique formal definition of the behaviour. An alternative possibility, also investigated in [5], is to interpret coinductively the standard big-step rules (*coevaluation*). Unfortunately, as discussed in Sect. 3, coevaluation is non-deterministic and, thus, may fail to correctly capture the correct infinite behavior of a computation: a diverging term, such as Ω, evaluates to any value.

Pretty big-step semantics [13] handles the issue of duplication of rules by a unified judgment with a unique set of rules, which can be interpreted either inductively or coinductively, getting termination and divergence, respectively. To factorise rules, the approach requires the introduction of new specific syntactic forms representing intermediate computation steps, as in small-step semantics. Poulsen and Mosses [14] subsequently present *flag-based big-step semantics*, which further streamlines the approach by combining it with the M-SOS technique (modular structural operational semantics), thereby reducing the number of rules and premises, avoiding the need for intermediate forms.

Both approaches [13, 14] do not prevent spurious results in non-terminating computations, as happens in coevaluation.

In [1] it has been firstly shown that with corules [3, 4] one can define a unified big-step judgment with a unique set of rules avoiding spurious evaluations. This can be seen as *constrained coevaluation*. Indeed, corules add constraints on the infinite derivations to filter out spurious results, so that, for diverging terms, it is only possible to get ∞ as result. This is extended to include observations as traces in [2]. In this case, the effect of spurious evaluations can be more detrimental; indeed, when a divergent computation produces a finite observation o, coevaluation returns any observation of shape $o * o_\infty$, and, thus, fails to correctly specify the effects of a non-terminating computation, as discussed on page 6. In comparison to [2], the present paper provides a receipe for a *fully systematic* approach, and, furthermore, allows reasoning on a *more general notion of observation*: at our knowledge, there is no other operational semantics framework able to capture divergence in conjunction with a notion of observation not limited to traces, as shown in Sect. 6.

Other proposals [15, 16, 17] are inspired by *definitional interpreters* [18], based on a step-indexed approach (a.k.a."fuel"-based semantics) where computations are approximated to some finite amount of steps (typically with a counter); in this way divergence can be modeled by induction. In [15] functional big-step semantics is investigated for proving by induction compiler correctness. In [16] inductive proof strategies are explored for type soundness properties for the polymorphic type systems $F_{<:}$, and equivalence with small-step semantics. An inductive proof of type soundness for the big-step semantics of a Java-like language is proposed in [17].

Conditional coinduction is employed in [19] to combine induction and coinduction in definitions of total parser combinators. In [20], inspired by [5], the coinductive partiality monad allows the definition of big/small-step semantics for lambda-calculi and virtual machines as total, computable functions able to capture divergence.

Coinductive trace semantics in big-step style have been studied by Nakata and Uustalu [21, 22, 23]. Their investigation started with the semantics of an imperative While language with no I/O [21] where traces are possibly infinite sequences of states; semantic rules are all coinductive and define two mutually dependent judgments. Based on such a semantics, they define a Hoare logic [22]; differently to our approach, weak bisimilarity between traces is required for proving that programs exhibit equivalent observable behaviors. This is due to the fact that "silent effects" (that is, non-observable internal steps) must be explicitly represented to guarantee guardedness conditions which ensure productivity of co-recursive definitions. This problem is overcome with corules in generalized inference systems.

The semantics has been subsequently extended with interactive I/O [23], by exploiting the notion of resumption monad: a tree representing possible runs of a program to model its non-deterministic behavior due to input values. Also in this case a big-step trace semantics is defined with two mutually recursive coinductive judgments, and weak bisimilarity is needed; however, the definition of the observational equivalence is more involved, since it requires nesting inductive definitions in coinductive ones. A generalised notion of resumption has been introduced later by Piróg and Gibbons and studied in a category-theoretic and coalgebraic context [24].

In [21, 22] equivalence of the big-step and small-step semantics is proved; expressions and statements are distinct, and expressions cannot diverge. This is another significant difference with the languages considered in this paper; the semantics of `out` e becomes simpler, since the corresponding corule could be turned into a coaxiom.

The resumption monad of Nakata and Uustalu is inspired by the seminal work of Capretta [25] on the *delay monad*, where coinductive types are exploited to model infinite computations by means of a type constructor for partial elements, which allows the formal definition of convergence and divergence and a type-theoretic representation of general recursive functions; this type constructor is proved to constitute a strong monad, upon which subsequent related papers elaborated [26, 27, 28] to define other monads for managing divergence. In particular, McBride [27] has proposed a more general approach based on a free monad for which the delay monad is an instantiation obtained through a monad morphism. All these proposals are based on the step-indexed approach.

More recently, interaction trees (ITrees) [29] have been presented as a coinductive variant of free monads with the main aim of defining the denotational semantics for effectful and possibly nonterminating computations, to allow compositional reasoning for mutually recursive components of an interactive system with fully mechanized proofs in Coq. Interaction trees are coinductively defined trees which directly support a more general fixpoint combinator which does need a step-indexed approach, as happens for the general monad of McBride. A Tau constructor is introduced to represent a silent step of computation, to express silently diverging computations without violating Coq's guardedness condition; as a consequence, generic definition of weak bisimulation on ITrees is required to remove any finite number of Tau's, similarly as happens in the approach of Nakata and Uustalu.

9. Discussion and future work

The big-step style can be useful for abstracting details, directly deriving the implementation of an interpreter, formally verifying compilers [5], cost semantics, and soundness and completeness proofs for program logics [13]. However, modeling divergence is a non trivial problem, even more when a non terminating program can have a significant behaviour through observations. Roughly, the main difficulty is that we do not describe the infinite behaviour "a posteriori", that is, given the whole infinite sequence of finite observations. Instead, we want to directly get the possibly infinite observation produced by a diverging computation. In succinct words, *we want to deal directly with infinity*.

The paper proposes a solution to this problem which uses *inference systems with corules* [3, 4]. This solution has been first adopted in [1] for pure divergence, and then in [2] by adding observations as traces. However, in these works the approach was only applied in certain examples and with observations, if any, limited to traces. In this paper, we go much further showing that the approach can be applied uniformly to a wider class of big-step definitions once the specific nature of observations, not limited to traces, and their composition is determined.

The quest for the above described solution has led to two other contributions which are important in themselves. The first is the definition of an algebraic notion, *ω-monoids*, which nicely models finite and infinite observations and provides the ingredients to express infinite behaviour both in the small-step and big-step approach, through infinite product and finite/mixed product, respectively. The second is an abstract notion of "what is a big-step semantics", general enough to capture typical examples. Having such an abstract notion, we are able to formally define the extension on a generic instance, and even to formalize in a generic way equivalence conditions between big-step rules and small-step relation which one typically uses in proofs of equivalence.

Given a big-step semantics obtained by our approach, as one of the examples in Sect. 6, it is desirable to have a mechanized support to prove its properties, also including infinite behaviour. A proof-of-concept is provided in [30], which shows how to implement in Agda examples of inference systems with corules, the corresponding bounded coinduction principle, and proofs of soundness/completeness with respect to a given specification.

A related issue is to check derivability of a judgment in an inference system with corules, notably existence of an infinite proof tree. Restricting to regular trees, this can be done by expressing the inference system as a Prolog program and applying *co-SLD resolution* [31, 32], where an atom of the goal can be successfully resolved if it can be unified with one of the atoms, called *coinductive hypotheses*, that have been already resolved. To deal with corules, co-SLD resolution needs to be modified: an atom that unifies with a coinductive hypothesis triggers standard SLD-resolution in the system extended with corules [33].

A typical drawback of standard big-step semantics is that stuck computations and divergence are identified, hence it is not possible to reason on properties of non-terminating computations. In the extended big-step semantics obtained by our approach, this becomes possible (see, e.g., the proof of soundness of a type system given in [1]). On the other hand, the property that a *single step* preserves type makes only sense in small-step semantics. In big-step semantics, the corresponding property is strong soundness [34], that is, computations cannot be stuck, and furthermore, produce a result of the same type if they converge. A finer grained property of big-step rules which more closely resembles small-step type preservation is *local preservation*, introduced in [35] as one of three conditions which ensure (strong) soudness. Local preservation requires, roughly, that well-typedness of the consequence implies well-typedness of the premises.

In the general context of inference systems with corules, it would be interesting to enhance the associated proof method. More specifically, the notion of bounded coinduction is a combination of a standard coinductive proof method (establishing a suitable post-fixed) and a separate inclusion in a domain determined by rules and corules. The coinductive part is amenable to up-to techniques [36, 37], which may help to simplify proofs using bounded coinduction, and parametric coinduction [38], which may be useful in formalisation in a proof assistant. We leave a more thorough investigation of their use in the context of inference systems with corules for future work.

We will also explore refined corule patterns able to achieve soundness for arbitrary ω-monoids. Moreover, we plan to improve the construction to support non-determinism. Indeed, whereas the current extension also works for many realistic non-deterministic examples, there are subtle cases where unsound judgments can be derived. Roughly, in case two judgments for the same configuration, say j_1 and j_2, have an infinite proof tree, to construct a finite proof tree using corules for j_1 we can use rules from the proof tree of j_2. To avoid this, we plan to explicitly add in the judgment the partially obtained observation.

A more general long-term goal is to investigate an abstract notion of "inference system with metarules", which would possibly make both the extension with corules and the general construction in this paper smoother.

References

[1] D. Ancona, F. Dagnino, E. Zucca, Reasoning on divergent computations with coaxioms, PACMPL 1 (OOPSLA) (2017) 81:1–81:26. `doi:10.1145/3133905`.

[2] D. Ancona, F. Dagnino, E. Zucca, Modeling infinite behaviour by corules, in: T. D. Millstein (Ed.), ECOOP'18 - Object-Oriented Programming, Vol. 109 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 21:1–21:31. `doi:10.4230/LIPIcs.ECOOP.2018.21`.

[3] D. Ancona, F. Dagnino, E. Zucca, Generalizing inference systems by coaxioms, in: H. Yang (Ed.), ESOP 2017 - European Symposium on Programming, Vol. 10201 of Lecture Notes in Computer Science, Springer, 2017, pp. 29–55. `doi:10.1007/978-3-662-54434-1_2`.

[4] F. Dagnino, Coaxioms: flexible coinductive definitions by inference systems, Logical Methods in Computer Science 15 (1). `doi:10.23638/LMCS-15(1:26)2019`.

[5] X. Leroy, H. Grall, Coinductive big-step operational semantics, Information and Computation 207 (2) (2009) 284–304. `doi:10.1016/j.ic.2007.12.004`.

[6] D. Perrin, J. Pin, Infinite words - automata, semigroups, logic and games, Vol. 141 of Pure and applied mathematics series, Elsevier Morgan Kaufmann, 2004.

[7] P. Aczel, An introduction to inductive definitions, in: J. Barwise (Ed.), Handbook of Mathematical logic, North Holland, 1977, pp. 739–782. `doi:10.1016/S0049-237X(08)71120-0`.

[8] G. Markowsky, Chain-complete posets and directed sets with applications, Algebra universalis 6 (1) (1976) 53–68. `doi:10.1007/BF02485815`.

[9] B. C. Pierce, Types and programming languages, The MIT Press, 2002.

[10] D. Marino, T. D. Millstein, A generic type-and-effect system, in: A. Kennedy, A. Ahmed (Eds.), TLDI'09: Types in Languages Design and Implementation, ACM Press, 2009, pp. 39–50. `doi:10.1145/1481861.1481868`.

[11] R. Tate, The sequential semantics of producer effect systems, in: R. Giacobazzi, R. Cousot (Eds.), ACM Symp. on Principles of Programming Languages 2013, ACM Press, 2013, pp. 15–26. `doi:10.1145/2429069.2429074`.

[12] P. Cousot, R. Cousot, Inductive definitions, semantics and abstract interpretations, in: R. Sethi (Ed.), ACM Symp. on Principles of Programming Languages 1992, ACM Press, 1992, pp. 83–94. `doi:10.1145/143165.143184`.

[13] A. Charguéraud, Pretty-big-step semantics, in: M. Felleisen, P. Gardner (Eds.), ESOP 2013 - European Symposium on Programming, Vol. 7792 of Lecture Notes in Computer Science, Springer, 2013, pp. 41–60. `doi:10.1007/978-3-642-37036-6_3`.

[14] C. B. Poulsen, P. D. Mosses, Flag-based big-step semantics, Journal of Logical and Algebraic Methods in Programming 88 (2017) 174–190. `doi:10.1016/j.jlamp.2016.05.001`.

[15] S. Owens, M. O. Myreen, R. Kumar, Y. K. Tan, Functional big-step semantics, in: P. Thiemann (Ed.), ESOP 2016 - European Symposium on Programming, Vol. 9632 of Lecture Notes in Computer Science, Springer, 2016, pp. 589–615. `doi:10.1007/978-3-662-49498-1\_23`.

[16] N. Amin, T. Rompf, Type soundness proofs with definitional interpreters, in: G. Castagna, A. D. Gordon (Eds.), ACM Symp. on Principles of Programming Languages 2017, ACM Press, 2017, pp. 666–679. doi:10.1145/3009837.

[17] D. Ancona, How to prove type soundness of Java-like languages without forgoing big-step semantics, in: D. J. Pearce (Ed.), FTfJP'14 - Formal Techniques for Java-like Programs, ACM Press, 2014, pp. 1:1–1:6. doi:10.1145/2635631.2635846.

[18] J. C. Reynolds, Definitional interpreters for higher-order programming languages, in: ACM '72, Proceedings of the ACM annual conference, Vol. 2, ACM Press, 1972, pp. 717–740. doi:10.1145/800194.805852.

[19] N. A. Danielsson, Total parser combinators, in: P. Hudak, S. Weirich (Eds.), Intl. Conf. on Functional Programming 2010, ACM Press, 2010, pp. 285–296. doi:10.1145/1863543.1863585.

[20] N. A. Danielsson, Operational semantics using the partiality monad, in: P. Thiemann, R. B. Findler (Eds.), Intl. Conf. on Functional Programming 2012, ACM Press, 2012, pp. 127–138. doi:10.1145/2364527.2364546.

[21] K. Nakata, T. Uustalu, Trace-based coinductive operational semantics for while, in: S. Berghofer, T. Nipkow, C. Urban, M. Wenzel (Eds.), Theorem Proving in Higher Order Logics - TPHOLs 2009, Vol. 5674 of Lecture Notes in Computer Science, Springer, 2009, pp. 375–390. doi:10.1007/978-3-642-03359-9_26.

[22] K. Nakata, T. Uustalu, A Hoare logic for the coinductive trace-based big-step semantics of while, in: A. D. Gordon (Ed.), ESOP 2010 - European Symposium on Programming, Vol. 6012 of Lecture Notes in Computer Science, Springer, 2010, pp. 488–506. doi:10.1007/978-3-642-11957-6_26.

[23] K. Nakata, T. Uustalu, Resumptions, weak bisimilarity and big-step semantics for while with interactive I/O: an exercise in mixed induction-coinduction, in: L. Aceto, P. Sobocinski (Eds.), SOS'10 - Structural Operational Semantics, Vol. 32 of Electronic Proceedings in Theoretical Computer Science, 2010, pp. 57–75. doi:10.4204/EPTCS.32.5.

[24] M. Piróg, J. Gibbons, The coinductive resumption monad, in: MFPS 2014 - Mathematical Foundations of Programming Semantics, 2014, pp. 273–288. doi:10.1016/j.entcs.2014.10.015.

[25] V. Capretta, General recursion via coinductive types, Logical Methods in Computer Science 1 (2). doi:10.2168/LMCS-1(2:1)2005.

[26] A. Abel, J. Chapman, Normalization by evaluation in the delay monad: A case study for coinduction via copatterns and sized types, in: P. Levy, N. Krishnaswami (Eds.), MSFP@ETAPS 2014 - Mathematically Structured Functional Programming, Vol. 153 of Electronic Proceedings in Theoretical Computer Science, 2014, pp. 51–67. doi:10.4204/EPTCS.153.4.

[27] C. McBride, Turing-completeness totally free, in: R. Hinze, J. Voigtländer (Eds.), MPC 2015 - Mathematics of Program Construction, Vol. 9129 of Lecture Notes in Computer Science, Springer, 2015, pp. 257–275. doi:10.1007/978-3-319-19797-5_13.

[28] J. Chapman, T. Uustalu, N. Veltri, Quotienting the delay monad by weak bisimilarity, Mathematical Structures in Computer Science 29 (1) (2019) 67–92. doi:10.1017/S0960129517000184.

[29] L. Xia, Y. Zakowski, P. He, C. Hur, G. Malecha, B. C. Pierce, S. Zdancewic, Interaction trees: representing recursive and impure programs in Coq, PACMPL 4 (ACM Symp. on Principles of Programming Languages 2020) (2020) 51:1–51:32. doi:10.1145/3371119.

[30] L. Ciccone, Flexible coinduction in Agda, Master Thesis in Computer Science, University of Genova (2019).

[31] L. Simon, A. Bansal, A. Mallya, G. Gupta, Co-logic programming: Extending logic programming with coinduction, in: L. Arge, C. Cachin, T. Jurdzinski, A. Tarlecki (Eds.), ICALP 2007 - Automata, Languages and Programming, Vol. 4596 of Lecture Notes in Computer Science, Springer, 2007, pp. 472–483. doi:10.1007/978-3-540-73420-8_42.

[32] D. Ancona, A. Dovier, A theoretical perspective of coinductive logic programming, Fundamenta Informaticae 140 (3-4) (2015) 221–246. doi:10.3233/FI-2015-1252.

[33] D. Ancona, F. Dagnino, E. Zucca, Extending coinductive logic programming with co-facts, Electronic Proceedings in Theoretical Computer Science 258 (2017) 1–18. doi:10.4204/eptcs.258.1.

[34] A. K. Wright, M. Felleisen, A syntactic approach to type soundness, Information and Computation 115 (1) (1994) 38–94. doi:10.1006/inco.1994.1093.

[35] F. Dagnino, V. Bono, E. Zucca, M. Dezani-Ciancaglini, Soundness conditions for big-step semantics, in: ESOP 2020 - European Symposium on Programming, 2020, to appear.

[36] D. Pous, D. Sangiorgi, Enhancements of the bisimulation proof method, in: D. Sangiorgi, J. Rutten (Eds.), Advanced Topics in Bisimulation and Coinduction, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2012, pp. 233–289. doi:10.1017/CBO9780511792588.007.

[37] D. Pous, Coinduction all the way up, in: M. Grohe, E. Koskinen, N. Shankar (Eds.), LICS 2016 - Logic in Computer Science, ACM Press, 2016, pp. 307–316. doi:10.1145/2933575.2934564.

[38] C. Hur, G. Neis, D. Dreyer, V. Vafeiadis, The power of parameterization in coinductive proof, in: ACM Symp. on Principles of Programming Languages 2013, ACM Press, 2013, pp. 193–206.

A. Details on the completion from monoids to ω-monoids

In this section, we provide details on the completion of a monoid, which was presented (without proofs) in Sect. 4. We will present this completion as a functor $C_\infty : \mathsf{Mon}_c \to \omega\text{-}\mathsf{Mon}$, where Mon_c is the category of left continuous monoids and continuous homomorphisms, and ω-Mon is the category of ω-monoids and ω-monoid homomorphisms. This functor, given a left continuous monoid M, will produce an ω-monoid $C_\infty(M) = \langle M, M_\infty\rangle$, where the second component M_∞ is presented as a quotient of M^ω by a suitable equivalence relation.

In proofs, it is sometimes useful to slightly reformulate $S(\sigma)$, as follows:

$$S(\sigma) = \begin{cases} \{\mathsf{it}(u) \mid u \triangleleft \sigma\} \cup \{\sup_{u \triangleleft \sigma}(\mathsf{it}(u))\} & \text{if this supremum is defined} \\ \{\mathsf{it}(u) \mid u \triangleleft \sigma\} & \text{otherwise} \end{cases} \tag{A.1}$$

Indeed, the closure adds at most a single element.

Lemma A.1. *The following hold for any left continuous monoid M:*

1. *for all $z \in M$ and $\sigma, \tau \in M^\omega$, if $\sigma \equiv \tau$, then $z{:}\sigma \equiv z{:}\tau$,*
2. *for all $\sigma, \tau \in M^\omega$: if $\sigma <: \tau$ then $\sigma \equiv \tau$,*
3. *for all $z_1, z_2 \in M$, $z_1{:}\mathsf{u}^\omega \equiv z_2{:}\mathsf{u}^\omega$ iff $z_1 = z_2$,*
4. *for any continuous monoid homorphism $f : M \to M'$ and $\sigma, \tau \in M^\omega$: $\sigma \equiv \tau$ implies $f^\omega(\sigma) \equiv f^\omega(\tau)$.*

Proof.

1. We have to prove the two inequalities $z{:}\sigma \sqsubseteq z{:}\tau$ and $z{:}\tau \sqsubseteq z{:}\sigma$; we prove only the first, as the other is analogous.
   Let $x \in S(z : \sigma)$, then $x = \sup_i(\mathsf{it}(u_i))$ for some increasing sequence $(u_i)_i$ of prefixes of $z : \sigma$, i.e., $u_i \triangleleft z : \sigma$ for all i. Without loss of generality, assume that $(u_i)_i$ does not contain the empty word (since that's the least element; and the case that the supremum is the empty word is trivial) so each u_i is of the form $u_i = z : u_i'$, with $u_i' \triangleleft \sigma$. Then $\sup_i(\mathsf{it}(u_i')) \in S(\sigma)$, so since $\sigma \sqsubseteq \tau$ there is an increasing sequence $(v_i)_i$ with each v_i a prefix of τ, and $\sup_i(\mathsf{it}(u_i')) \preceq_* \sup_i(\mathsf{it}(v_i))$. So we get
   $$\begin{aligned} x &= \sup(\mathsf{it}(u_i)) \\ &= \sup(\mathsf{it}(z : u_i')) \\ &= \sup(z * \mathsf{it}(u_i')) \\ &= z * \sup(\mathsf{it}(u_i')) \\ &\preceq_* z * \sup(\mathsf{it}(v_i)) \\ &= \sup(\mathsf{it}(z : v_i)) \end{aligned}$$
   and since $sup(\mathsf{it}(z : v_i)) \in S(z : \tau)$ we get $z : \sigma \sqsubseteq z : \tau$ as needed.
2. Suppose $\sigma <: \tau$, with $\sigma = (x_i)_{i\in\mathbb{N}}$ and $\tau = (y_i)_{i\in\mathbb{N}}$, i.e., there is a strictly increasing sequence $(k_i)_{i\in\mathbb{N}}$ with $k_0 = 0$ and $x_i = y_{k_i} * \cdots * y_{k_{i+1}-1}$. Towards a proof of $\sigma \sqsubseteq \tau$, let $x \in S(\sigma)$. We use the characterisation in (A.1) and proceed with a case distinction.
   - If $x = \mathsf{it}(u)$ for some $u \triangleleft \sigma$, then it is easy to see there is $v \triangleleft \tau$ such that $\mathsf{it}(u) = \mathsf{it}(v)$, hence $\mathsf{it}(u) \preceq_* \mathsf{it}(v)$, and since $\mathsf{it}(v) \in S(\tau)$ this suffices.
   - Otherwise, we have $x = \sup(\mathsf{it}(u_i))$ where $(u_i)_{i\in\mathbb{N}}$ is such that u_i is the prefix of length i of σ. Then define the sequence $(v_i)_{i\in\mathbb{N}}$ of prefixes of τ by $v_i = \tau(0) : \tau(1) : \ldots : \tau(k_i - 1)$, so $\sup(\mathsf{it}_(v_i)) \in S(\tau)$. Then $\mathsf{it}(u_i) = \mathsf{it}(v_i)$, so $\sup(\mathsf{it}(u_i)) = \sup(\mathsf{it}(v_i))$, and $\sup(\mathsf{it}(u_i)) \preceq_* \sup(\mathsf{it}(v_i))$.

For $\tau \sqsubseteq \sigma$, we let $x \in S(\tau)$ and again use a case distinction.

- If $x = \mathsf{it}(u)$ for some $u \triangleleft \tau$, then there are v, w such that $\mathsf{it}(uv) = \mathsf{it}(w)$, for some $w \triangleleft \sigma$. Thus $\mathsf{it}(w) \in S(\sigma)$, and $\mathsf{it}(u) \preceq_* \mathsf{it}(u) * \mathsf{it}(v) = \mathsf{it}(uv) = \mathsf{it}(w)$.
- Otherwise $x = \sup(\mathsf{it}_{u_i})$ for some increasing and infinite sequence $(u_i)_{i\in\mathbb{N}}$. Without loss of generality, we assume that the length of each u_i equals k_i. Let $(v_i)_{i\in\mathbb{N}}$ be such that v_i is the prefix of τ of length i. $\mathsf{it}(u_i) = \mathsf{it}(v_i)$ for each i, and the proof concludes as the second case above.

3. Suppose $z_1 : \mathsf{u}^\omega \equiv z_2 : \mathsf{u}^\omega$. We prove $z_1 \preceq_* z_2$ and $z_2 \preceq_* z_1$; this suffices, since $\preceq_*$ is anti-symmetric. To this end, consider the prefix z_1 of $z_1 : \mathsf{u}^\omega$. Since $z_1 : \mathsf{u}^\omega \sqsubseteq z_2 : \mathsf{u}^\omega$ there is $y \in S(\tau)$ with $\mathsf{it}(z_1) \preceq_* y$. It is easy to see that either $y = \mathsf{it}(z_2)$ or $y = \mathsf{u}$, since it maps every prefix of $z_2 : \mathsf{u}^\omega$ to either one of those. If $y = \mathsf{it}(z_2)$, we are done, as $z_1 = \mathsf{it}(z_1) \preceq_* \mathsf{it}(z_2) = z_2$. In the second case, i.e., if $y = \mathsf{u}$, then $z_1 \preceq_* \mathsf{u}$, and by anti-symmetry of $\preceq_*$ we get $z_1 = \mathsf{u}$ and, since u is the least element, we get $z_1 \preceq_* z_2$. Analogously, from $z_2 : \mathsf{u}^\omega \sqsubseteq z_1 : \mathsf{u}^\omega$ we then get $z_2 \preceq_* z_1$, hence $z_2 = z_1$ by antisymmetry.
4. Suppose $f : M \to M'$ is a continuous monoid homomorphism, $\sigma, \tau \in M^\omega$, and $\sigma \sqsubseteq \tau$. We need to prove $f^\omega(\sigma) \sqsubseteq f^\omega(\tau)$. To this end, let $x \in S(\sigma)$. Then $x = \sup(\mathsf{it}(f^\star(u_i)))$ for some increasing sequence u_i of prefixes of σ. Since $\sigma \sqsubseteq \tau$, there exists $\sup(\mathsf{it}(v_i)) \in S(\tau)$, with $(v_i)_i$ an increasing sequence of prefixes of τ, and $\sup(\mathsf{it}(u_i)) \preceq_* \sup(\mathsf{it}(v_i))$. Now, we obtain:
$$\begin{aligned} x &= \sup(\mathsf{it}(f^\star(u_i))) \\ &= \sup(f(\mathsf{it}(u_i))) \\ &= f(\sup(\mathsf{it}(u_i))) \\ &\preceq_* f(\sup(\mathsf{it}(v_i))) \\ &= \sup(\mathsf{it}(f^\star(v_i))) \end{aligned}$$
The second step holds since f is a monoid homomorphism, the third since f is continuous.

□

We are now ready to prove that the completion is a functor $C_\infty : \mathsf{Mon}_c \to \omega\text{-}\mathsf{Mon}$. The main fact to show is that $C_\infty(f)$ is well defined; compatibility with composition and identities is immediate by definition.

Proposition A.1. *The construction $C_\infty(M) = \langle M, M_\infty \rangle$ extends to a functor from the category of left continuous monoids and continuous homomorphisms to the category of ω-monoids, given on a continuous homomorphism $f : M \to M'$ by $C_\infty(f) = \langle f, f^\omega \rangle$ (the map f^ω is well-defined on M_∞).*

Proof. Given a monoid M, $C_\infty(M)$ is an ω-monoid by Lemma A.1. By Lemma A.1 (4) the pair of functions $C_\infty(f)$ is well-defined, that is, $C_\infty(f) : C_\infty(M) \to C_\infty(M')$. We now check that $C_\infty(f)$ is a homomorphism of ω-monoids. By hypothesis the first component is a homomorphism of monoids, hence we have only to check compatibility with mixed product and infinite product. More precisely, we have to check the following equivalences: for all $x \in M$ and $\tau \in M^\omega$
$$f^\omega(x{:}\tau) \equiv f(x){:}f^\omega(\tau) \qquad f^\omega(\mathsf{p}_M(\tau)) \equiv \mathsf{p}_{M'}(f^\omega(\tau))$$
The former trivially holds by definition of f^ω (it is actually an equality), and the latter also trivially holds since at this level p_M is the identity, hence the equivalence can be rewritten as $f^\omega(\tau) \equiv f^\omega(\tau)$ that is true by reflexivity of $\equiv$.

Then the fact that C_∞ preserves composition and identities is straightforward, because composition is made pointwise and $(-)^\omega$ is functorial, and we have $C_\infty(\mathsf{id}_M) = \langle \mathsf{id}_M,\, \mathsf{id}_M^\omega \rangle = \langle \mathsf{id}_M,\, \mathsf{id}_{M^\omega} \rangle = \mathsf{id}_{C_\infty(M)}$. □

B. Proofs of main results

In this section we provide the technical development of completeness and soundness results stated in Sect. 7.4.

B.1. Completeness

We start by proving the completeness result (Theorem 7.1) for $C^{\mathcal{I}}$. The proof uses a slight variation of a proof principle from [1], reported below, which is an easy consequence of bounded coinduction (Theorem 2.1).

Theorem B.1. *Let $P \subseteq C \times O_\infty$ be a predicate. If for each $\langle c,\, o_\infty \rangle \in P$:*

Boundedness *$\langle \mathcal{I}_\infty \cup C,\, \emptyset \rangle \vdash c \Rightarrow \langle \infty,\, o_\infty \rangle$, and*

Consistency *$c \Rightarrow \langle \infty,\, o_\infty \rangle$ is the consequence of a rule where, for all premises of shape $c' \Rightarrow \langle \infty,\, o'_\infty \rangle$, $\langle c',\, o'_\infty \rangle \in P$, and, for all premises of shape $c' \Rightarrow \langle r,\, o \rangle$, $\langle \mathcal{I}_\infty,\, C \rangle \vdash c' \Rightarrow \langle r,\, o \rangle$,*

then $\langle c,\, o_\infty \rangle \in P$ implies that $\langle \mathcal{I}_\infty,\, C \rangle \vdash c \Rightarrow \langle \infty,\, o_\infty \rangle$.

The proof is omitted since analogous to the one in [1]. We start by showing boundedness, which needs two auxiliary lemmas.

We write $\bar{j}$ for a list of judgments $j_1 \ldots j_n$. **[Elena: moved here]**

Lemma B.1. *If $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}},\, \emptyset \rangle \vdash c' \Rightarrow \langle r_\infty,\, o_\infty \rangle$ and $c \xrightarrow{o} c'$, then $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}},\, \emptyset \rangle \vdash c \Rightarrow \langle r_\infty,\, o *_\infty o_\infty \rangle$.*

Proof. We proceed by induction on pr and we distinguish two cases.

1. If c is reduced then we distinguish two cases:
   - if $r_\infty = \infty$, then, by Condition 1, there exists a computational rule ρ=$\mathsf{rule}(j_1 \ldots j_n, j, c, o)$ such that $C(j) = c'$; hence, the thesis follows by rule $\mathsf{div}(\rho, \mathsf{f}(\rho), o_\infty)$, whose first n premises are $j_1, \ldots, j_n$, which are trivial, and the last one is $c' \Rightarrow \langle \infty,\, o_\infty \rangle$, which is derivable by hypothesis.
   - If $r_\infty \in R$, then the last applied rule ρ' in the proof tree for $c' \Rightarrow \langle r_\infty,\, o_\infty \rangle$ belongs to $\mathcal{I}$, hence, by Condition 3, there exists a computational rule ρ=$\mathsf{rule}(\bar{j}, j, c, o)$ with j=$c' \Rightarrow \langle r_\infty,\, o_\infty \rangle$; hence we get the thesis since all judgements in $\bar{j}$ are trivial and j is derivable by hypothesis.
2. If c is compound, then we have the following cases on the last applied rule in the proof tree for $c' \Rightarrow \langle r_\infty,\, o_\infty \rangle$:
   - if the rule is (co-unit), then $r_\infty = \infty$ and $o_\infty = \mathsf{u}$; by Condition 1 there is a contextual rule ρ=$\mathsf{rule}(j_1 \ldots j_i \ldots j_n, j, c, o')$ where $i = \mathsf{f}(\rho)$, $\mathsf{pr}(c) = C(j_i) = c_i$, and $c_i \xrightarrow{o} c''$. Applying again (co-unit) we can derive $c'' \Rightarrow \langle \infty,\, \mathsf{u} \rangle$, hence, by inductive hypothesis $c_i \Rightarrow \langle \infty,\, o \rangle$ is derivable, hence we get the thesis by rule $\mathsf{div}(\rho, i, o)$.

- In all other cases there is a rule $\rho' \in \mathcal{I}$ associated with the last applied rule[14] in the proof tree for $c' \Rightarrow \langle r_\infty, o_\infty\rangle$, hence, by Condition 3, there exists a contextual rule ρ=$\mathsf{rule}(j_1 \ldots j_i \ldots j_n, j, c, o')$ where $i = \mathsf{f}(\rho)$ and $\rho \xrightarrow{o} \rho'$. Hence, we have $C(j_i) = \mathsf{pr}(c) = c_i$, $c_i \xrightarrow{o} c'' = C(j'_i)$ where j'_i is a dependency of ρ', $O(j_i) = o * O(j'_i)$, and $R(j_i) = R(j'_i)$. Therefore, since by hypothesis j'_i is derivable, by inductive hypothesis j_i is derivable as well, hence we get the thesis by applying the rule associated with ρ of the same type of the one associated with ρ'.

$\square$

Lemma B.2. *If $c \xrightarrow{o} c'$ and $o \neq \mathsf{u}$, then $\langle \mathcal{I}_\infty \cup \mathcal{C}^{\mathcal{I}}, \emptyset\rangle \vdash c \Rightarrow \langle\infty, o *_\infty o_\infty\rangle$ for $o_\infty \in O_\infty$.*

Proof. We proceed by induction on pr. By Condition 1, there exists a rule ρ=$\mathsf{rule}(\bar{j}, j, c, o')$ and we distinguish two cases:

1. if c is reduced, then ρ is computational, and, since $o \neq \mathsf{u}$, $\mathsf{co\text{-}mul}(\rho, o_\infty) \in \mathcal{C}^{\mathcal{I}}$ and this gives us the thesis, since all judgements in $\bar{j}$ are trivial.
2. if c is compound, then ρ is contextual with $\bar{j} = j_1 \ldots j_n$, $i = \mathsf{f}(\rho)$, and $C(j_i) = \mathsf{pr}(c) \xrightarrow{o} c''$; therefore, by inductive hypothesis, we get $\langle \mathcal{I}_\infty \cup \mathcal{C}^{\mathcal{I}}, \emptyset\rangle \vdash \mathsf{pr}(c) \Rightarrow \langle\infty, o *_\infty o_\infty\rangle$, hence we get the thesis by rule $\mathsf{div}(\rho, i, o *_\infty o_\infty)$.

$\square$

Lemma B.3 (Boundedness)**.** *If $c \rightsquigarrow \langle\infty, o_\infty\rangle$, then $\langle \mathcal{I}_\infty \cup \mathcal{C}^{\mathcal{I}}, \emptyset\rangle \vdash c \Rightarrow \langle\infty, o_\infty\rangle$.*

Proof. By definition, $c \xrightarrow{\sigma}{}^{\omega}$ for some $\sigma \in O^\omega$ such that $\mathsf{it}_\infty(\sigma) = o_\infty$. We distinguish two cases:

- if $\sigma = \mathsf{u}^\omega$, then $o_\infty = \mathsf{u}$ and the thesis follows from the coaxiom (co-unit)
- otherwise, there exists $o \neq \mathsf{u}$ and $n \in \mathbb{N}$ such that $\sigma = \mathsf{u}^n(o{:}\sigma')$, hence $o_\infty = o *_\infty o'_\infty$, with $o'_\infty = \mathsf{it}_\infty(\sigma')$, and $c \xrightarrow{\mathsf{u}^n}{}^{\star} c' \xrightarrow{o} c''$. By Lemma B.2 we get that $\langle \mathcal{I}_\infty \cup \mathcal{C}^{\mathcal{I}}, \emptyset\rangle \vdash c' \Rightarrow \langle\infty, o *_\infty o'_\infty\rangle$ and by a transitive closure[15] of Lemma B.1 we get $\langle \mathcal{I}_\infty \cup \mathcal{C}^{\mathcal{I}}, \emptyset\rangle \vdash c \Rightarrow \langle\infty, \mathsf{u} * o *_\infty o'_\infty\rangle$.

$\square$

To prove consistency, we need that, in an infinite small-step reduction sequence, we always reach a configuration c which is either reduced or compound with $\mathsf{pr}(c)$ diverging too.

Lemma B.4. *Given sequences $(c_i)_{i\in\mathbb{N}}$ and $(o_i)_{i\in\mathbb{N}}$ such that, for all $i \in \mathbb{N}$, $c_i \xrightarrow{o_i} c_{i+1}$, then there exists an $n \in \mathbb{N}$ such that, for each $k < n$, c_k is compound and:*

1. *either c_n is reduced*
2. *or c_n is compound and, for all $k \geq n$, $\mathsf{pr}(c_k) \xrightarrow{o_k} \mathsf{pr}(c_{k+1})$.*

[14]This rule can be either $\mathsf{div}(\rho', i, o'_\infty)$, or $\mathsf{co\text{-}mul}(\rho', o'_\infty)$, or ρ' itself.
[15]More precisely by induction on n.

Proof. By Assumption 2, all non-stuck rules ρ for c_0 have the same number of dependencies, say d, and the same index $\mathsf{f}(\rho)$. Furthermore, since c_0 is non-stuck, by Condition 1 there exists at least one non-stuck ρ for it, hence, it is uniquely defined the number of dependencies we still have to evaluate, that is $d + 1 - \mathsf{f}(\rho)$. Let us denote by $\ell(c_0)$ such number. We can rephrase the statement as follows:

> for all $h \in \mathbb{N}$, for all $(c_i)_{i\in\mathbb{N}}$ and $(o_i)_{i\in\mathbb{N}}$ such that $c_i \xrightarrow{o_i} c_{i+1}$ and $\ell(c_0) = h$, there exists $n \in \mathbb{N}$ such that c_n satisfies either 1 or 2.

Therefore, we can proceed by complete arithmetic induction on $\ell(c_0)$.

Base If $\ell(c_0) = 0$, then c_0 is reduced by definition, hence point 1 trivially holds.

Induction Suppose that $\ell(c_0) > 0$ and let us start with some observations. Since by hypothesis for all $i \in \mathbb{N}$ we have $c_i \xrightarrow{o_i} c_{i+1}$, for all $i \in \mathbb{N}$, if c_i is compound, then either $\mathsf{pr}(c_i) \xrightarrow{o_i} r$ or c_{i+1} is compound and $\mathsf{pr}(c_i) \xrightarrow{o_i} \mathsf{pr}(c_{i+1})$. Furthermore, if c_i is compound, then $\mathsf{pr}(c_i) \xrightarrow{o_i} c'$, and $\ell(c_{i+1}) \le \ell(c_i)$; more precisely, if $c' \in R$, then $\ell(c_{i+1}) < \ell(c_i)$, otherwise $\ell(c_{i+1}) = \ell(c_i)$.

So we have two cases:

- if for all compound configurations c_i, we have $\mathsf{pr}(c_i) \xrightarrow{o_i} \mathsf{pr}(c_{i+1})$, then it is easy to prove by induction on i that all configurations c_i are compound and so $\ell(c_i) = \ell(c_{i+1})$ and $\mathsf{pr}(c_i) \xrightarrow{o_i} \mathsf{pr}(c_{i+1})$. Hence $\mathsf{pr}(c_0)$ diverges as required in 2.
- Otherwise, let n be the first (least) index such that c_n is compound and $\mathsf{pr}(c_n) \xrightarrow{o_n} r$. Again, it is easy to check by induction on i, that for all $i < n$, c_i is compound, hence $\mathsf{pr}(c_i) \xrightarrow{o_i} \mathsf{pr}(c_{i+1})$ and $\ell(c_{n+1}) < \ell(c_n)$. Therefore, considering the sequences $(c_k)_{k>n}$ and $(o_k)_{k>n}$, by inductive hypothesis, there is an index $m \ge k > n$ satisfying one of the two conditions, hence the thesis holds also for the original sequences, since c_m belongs also to it.

□

Lemma B.5. *Given finite sequences $(\rho_i)_{i\le m}$ and $(o_i)_{i<m}$ such that*

- *for all $i < m$, $\rho_i \xrightarrow{o_i} \rho_{i+1}$,*
- *$\rho_0 = \mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$ is non-stuck, and*
- *$\rho_m = \mathsf{rule}(j'_1 \ldots j'_n, j'_{n+1}, c', o')$, with $l = \mathsf{f}(\rho_m)$,*

then:

- *for all $k < l$ we have $C(j_k) \rightsquigarrow \langle R(j_k), O(j_k)\rangle$, and*
- *$C(j_l) \xrightarrow{v}^{\star} C(j'_l)$ and $O(j_l) = \mathsf{it}(v) * O(j'_l)$.*

Proof. We proceed by arithmetic induction on m.

Base If $m = 0$, then $\rho_0 = \rho_m$, hence the thesis holds, since ρ_0 is non-stuck, and so, by definition, for all $k < l$, j_k is trivial, that is, $j_k = r_k \Rightarrow \langle r_k, \mathsf{u}\rangle$, and so the first part trivially holds, and the second one holds as well, by taking $v = \varepsilon$, since $j_l = j'_l$.

Induction Let us assume the thesis for m and prove it for $m+1$. By hypothesis $\rho_0 \xrightarrow{o_0} \rho_1$ and by inductive hypothesis the thesis holds for ρ_1. Let $h = \mathsf{f}(\rho_0)$ and $k = \mathsf{f}(\rho_1)$, then $h \leq k \leq l$. Since $\rho_0 \xrightarrow{o_0} \rho_1$, ρ_0 is contextual, hence all j_i for $i < h$ are trivial, hence satisfy point 1, while all j_i for $h < i \leq l$ are also premises of ρ_1, hence satisfy the thesis by inductive hypothesis. Therefore, we have only to prove the thesis for j_h.

Let j be the h-th premise of ρ_1, we know that $C(j_h) \xrightarrow{o_0} C(j)$, $R(j_h) = R(j)$, and $O(j_h) = o_0 * O(j)$. We distinguish two cases:

- if $h < l$, then by inductive hypothesis $C(j) \rightsquigarrow \langle R(j), O(j)\rangle$ holds, hence we can derive $C(j_h) \rightsquigarrow \langle R(j), o_0 * O(j)\rangle$, that is, $C(j_h) \rightsquigarrow \langle R(j_h), O(j_h)\rangle$.
- if $h = k = l$, then by inductive hypothesis $C(j) \xrightarrow{v} C(j'_l)$ for some finite sequence v, hence the thesis follows, since $C(j_h) \xrightarrow{o_0} C(j)$ and $O(j_h) = o_0 * O(j) = o_0 * \mathsf{id}(v) * O(j'_l) = \mathsf{it}(o_0{:}v) * O(j'_l)$.

□

We have now all the elements to prove completeness.

Proof of Theorem 7.1. We use Theorem B.1. Here the predicate P is the set $\{\langle c, o_\infty\rangle \mid c \rightsquigarrow \langle \infty, o_\infty\rangle\}$. This set is bounded by Lemma B.3, hence we have only to check the second condition.

Let us consider $\langle c, o_\infty\rangle$ such that $c \rightsquigarrow \langle \infty, o_\infty\rangle$, hence, by definition, there are $(c_i)_{i\in\mathbb{N}}$ and $\sigma = (o_i)_{i\in\mathbb{N}}$ such that $c_0 = c$, $c_i \xrightarrow{o_i} c_{i+1}$ and $\mathsf{it}_\infty(\sigma) = o_\infty$. By Lemma B.4 there is an index $n \in \mathbb{N}$ such that all c_k for $k < n$ are compound and either c_n is reduced, or c_n is compound and $\mathsf{pr}(c_n)$ diverges. In both cases, by Condition 1 there is a rule ρ_n with $C(\rho_n) = c_n$ and, by an iteration of Condition 3, there is a sequence of rules $(\rho_i)_{i\leq n}$ such that $\rho_i \xrightarrow{o_i} \rho_{i+1}$ and $C(\rho_i) = c_i$. By Lemma B.5, if $\rho_0 = \mathsf{rule}(j_1 \ldots j_m, j_{m+1}, c, o)$, $l = \mathsf{f}(\rho_n)$ and j is the l-th premise of ρ_n, then, for all $k < l$, $C(j_k) \rightsquigarrow \langle R(j_k), O(j_k)\rangle$, $C(j_l) \xrightarrow{v}{}^{\star} C(j)$, and $O(j_l) = \mathsf{it}(v) * O(j)$. Therefore, by Lemma 7.4 and Theorem 5.1 we get $\langle \mathcal{I}_\infty, C^{\mathcal{I}}\rangle \vdash j_k$.

Now, let us consider $\rho = \mathsf{div}(\rho_0, l, o'_\infty)$. We have two cases following Lemma B.4:

1. If c_n is reduced, then $l = m+1$, ρ_n is computational, and has shape $\mathsf{rule}(\bar{j}, j, c_n, o_n)$ by Condition 1. Since reduction of rules does not change the continuation, $j_l{=}j_{m+1}{=}j$, hence $v = \varepsilon$ and so we set $o'_\infty = \mathsf{it}_\infty(\sigma_{n+1})$, where $\sigma_{n+1} = (o_i)_{i\geq n+1}$; hence, it is easy to prove by coinduction that $c_{n+1} \xrightarrow{\sigma_{n+1}}{}^{\omega}$, and this implies $c_{n+1} \rightsquigarrow \langle \infty, o'_\infty\rangle$, hence $\langle c_{n+1}, o'_\infty\rangle \in P$.
2. If c_n is compound, then ρ_n is contextual, $C(j) = \mathsf{pr}(c_n)$ and for all $i \geq n$, $\mathsf{pr}(c_i) \xrightarrow{o_i} \mathsf{pr}(c_{i+1})$, hence $\mathsf{pr}(c_n) \xrightarrow{\sigma_n}{}^{\omega}$, where $\sigma_n = (o_i)_{i\geq n}$. Hence, since $C(j_l) \xrightarrow{v}{}^{\star} C(j)$, we get $C(j_l) \xrightarrow{v\sigma_n}{}^{\omega}$, and so, setting $o'_\infty = \mathsf{it}_\infty(v\sigma_n)$, we get $C(j_l) \rightsquigarrow \langle \infty, o'_\infty\rangle$. Therefore, $\langle C(j_l), o'_\infty\rangle$ belongs to P.

In both cases ρ satisfies the requirements of Theorem B.1, hence this proves the thesis. □

We now prove the completeness result for $C^{\mathcal{I}}_{\mathsf{u}}$ (Theorem 7.2), stating that all infinite computations producing a finitely generated observation can be derived using only (co-unit).

To show this result, we first need some key properties of finitely generated observations reported in the following lemma:

Lemma B.6. *If $\sigma \in M^\omega_G$ is a sequence such that $z = \mathsf{p}(\sigma)$ is finitely generated, then:*

1. *there exists a finite prefix v of σ such that $z = \mathsf{p}(v\mathsf{u}^\omega)$*

2. *for all decomposition $\sigma = v\sigma'$, $\mathsf{p}(\sigma')$ is finitely generated.*

Proof.

1. Let $\sigma = (o_i)_{i\in\mathbb{N}}$ and $u_i = o_0 \dots o_i$ for all $i \in \mathbb{N}$, then, since $o_\infty = \mathsf{p}(\sigma)$ is finite, there is $n \in \mathbb{N}$ such that, for all $k \geq n$, $\mathsf{it}(u_n) = \mathsf{it}(u_k)$, otherwise we would have infinitely many elements in $\mathcal{F}(o_\infty)$ that is absurd. Then, $\sigma \equiv u_n\mathsf{u}^\omega$, which trivially holds from what we just observed, hence we have the thesis.
2. Let $\sigma = v\sigma'$ and note that $\mathcal{F}(\sigma')$ is not empty since the head of σ' belongs to it. Then, consider $o \in \mathcal{F}(\mathsf{p}(\sigma'))$, by definition of factor we have $\mathsf{p}(\sigma') = o' * o *_\infty \mathsf{p}(\sigma'')$, hence $(\mathsf{it}(v) * o') * o *_\infty \mathsf{p}(\sigma'') = \mathsf{p}(\sigma)$, that is, $o \in \mathcal{F}(o_\infty)$. This proves that $\mathcal{F}(\mathsf{p}(\sigma')) \subseteq \mathcal{F}(o_\infty)$, hence $\mathcal{F}(\mathsf{p}(\sigma'))$ is finite, and this proves the thesis.

□

The next lemma shows that, thanks to the equivalence conditions, observations produced by single steps in small-step semantics are themselves in $\mathrm{E}_\mathcal{I}$.

Lemma B.7. *If $c \xrightarrow{o} c'$ then $o \in E_\mathcal{I}$.*

Proof. We proceed by well-founded induction on pr: if c is reduced, then by Condition 1 there is a computational rule ρ=$\mathsf{rule}(\bar{j}, j, c, o)$, hence $o \in \mathrm{E}_\mathcal{I}$, and if c is compound, again by Condition 1, $\mathsf{pr}(c) \xrightarrow{o} c''$, hence $o \in \mathrm{E}_\mathcal{I}$ holds by inductive hypothesis. □

The proof of Theorem 7.2 is essentially the same as Theorem 7.1: we apply Theorem B.1, where Lemma B.6 guarantees boundedness and consistency.

Proof of Theorem 7.2. The proof follows exactly that of Theorem 7.1, hence we have first to prove boundedness. If $c \rightsquigarrow \langle \infty, o_\infty \rangle$, by definition there is $\sigma = (o_i)_{i\in\mathbb{N}}$ such that $o_\infty = \mathsf{p}(\sigma)$. By Lemma B.7, $o_i \in \mathrm{E}_\mathcal{I}$ for all $i \in \mathbb{N}$, hence, by Lemma B.6 (1) there exists $v = o_0 \dots o_{n-1}$ such that $o_\infty = \mathsf{p}(v\mathsf{u}^\omega)$. By hypothesis $c \xrightarrow{v}{}^\star c'$ and by (co-unit) we get $\langle \mathcal{I}_\infty \cup \mathcal{C}^\mathcal{I}_\mathsf{u}, \emptyset \rangle \vdash c' \Rightarrow \langle \infty, \mathsf{u} \rangle$; then, by Lemma B.1, restricted to $\mathcal{C}^\mathcal{I}_\mathsf{u}$, we get $\langle \mathcal{I}_\infty \cup \mathcal{C}^\mathcal{I}_\mathsf{u}, \emptyset \rangle \vdash c \Rightarrow \langle \infty, \mathsf{p}(v\mathsf{u}^\omega) \rangle$ as needed.

To prove consistency, we do exactly the same construction done in the proof of Theorem 7.1, hence we get a rule $\mathsf{div}(\rho, i, o'_\infty)$, where the i-th premise hase shape $c' \Rightarrow \langle \infty, o'_\infty \rangle$, where $o'_\infty = \mathsf{p}(\sigma')$ for some σ' such that $\sigma = v\sigma'$. Hence we have only to prove that o'_∞ is finitely generated, but this holds thanks to Lemma B.6 (2), hence we get the thesis by Theorem B.1. □

B.2. Soundness

We start by some results and notations which hold for any conservative set of corules, hence in particular for both $\mathcal{C}^\mathcal{I}$ and $\mathcal{C}^\mathcal{I}_\mathsf{u}$.

Lemma B.8 (Forward compatibility). *If $\mathcal{C}$ is conservative, $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash c \Rightarrow \langle \infty, o_\infty \rangle$ by a rule $\mathsf{div}(\rho, i, o''_\infty)$, with $\rho = \mathsf{rule}(j_1 \dots j_n, j_{n+1}, c, o)$, then $c \xrightarrow{u}{}^\star c'$ with $u = u_1 \dots u_{i-1}$ and $\mathsf{it}(u_k) = O(j_k)$ for all $k < i$, $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash c' \Rightarrow \langle \infty, o'_\infty \rangle$ and*

1. *if c' is reduced, then $c' \xrightarrow{o} C(j_{n+1})$ and $o'_\infty = o *_\infty o''_\infty$ and $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash C(j_{n+1}) \Rightarrow \langle \infty, o''_\infty \rangle$*
2. *if c' is compound, then $\mathsf{pr}(c')$=$C(j_i)$, o'_∞=o''_∞, and $\langle \mathcal{I}_\infty, \mathcal{C} \rangle \vdash \mathsf{pr}(c') \Rightarrow \langle \infty, o''_\infty \rangle$.*

Proof. By construction for all $k < i$, j_k is a convergent judgement of shape $c_k \Rightarrow \langle r_k,\ o_k\rangle$ and by hypothesis we have $\langle \mathcal{I}_\infty,\ C\rangle \vdash j_k$; therefore, by Theorem 5.1 and Lemma 7.5 we get that $c_k \rightsquigarrow \langle r_k,\ o_k\rangle$, that is, $c_k \xrightarrow{u_k}{}^\star r_k$ for $u_k \in O^\star$ and $\mathsf{it}(u_k) = o_k$. By iteratively applying Condition 2, we get that $c \xrightarrow{u}{}^\star c'$ with $u = u_1 \cdots u_{i-1}$ and $\rho \xrightarrow{u}{}^\star \rho'$[16] with $\rho' = \mathsf{rule}(j'_1 \ldots j'_{i-1} j_i \ldots j_n, j_{n+1}, c', o)$ and $j'_k = r_k \Rightarrow \langle r_k,\ \mathsf{u}\rangle$. Then by rule $\mathsf{div}(\rho', i, o''_\infty)$ we get that $\langle \mathcal{I}_\infty,\ C\rangle \vdash c' \Rightarrow \langle \infty,\ o'_\infty\rangle$ and distinguish the following cases:

1. If c' is reduced, then ρ' is computational and, by Condition 2, we get that $c' \xrightarrow{o} C(j_{n+1}) = c''$, hence $i = \mathsf{f}(\rho') = n + 1$, $o'_\infty = o *_\infty o''_\infty$ and $c'' \Rightarrow \langle \infty,\ o''_\infty\rangle$ is derivable since it is a premise of $\mathsf{div}(\rho', i, o''_\infty)$.
2. If c' is compound, then ρ' is contextual, $i = \mathsf{f}(\rho')$, $\mathsf{pr}(c') = C(j_i)$, $o'_\infty = o''_\infty$ and $\mathsf{pr}(c') \Rightarrow \langle \infty,\ o''_\infty\rangle$ is derivable since it is a premise of $\mathsf{div}(\rho', i, o''_\infty)$.

□

We now show that, if $\langle \mathcal{I}_\infty,\ C\rangle \vdash c \Rightarrow \langle \infty,\ o_\infty\rangle$, then there is an infinite reduction $c \xrightarrow{\sigma}{}^\omega$ in the small-step semantics where σ is equivalent to σ_t (see the definition introduced for Lemma 7.6), as stated in the following lemma.

Lemma B.9. *If $\langle \mathcal{I}_\infty,\ C\rangle \vdash c \Rightarrow \langle \infty,\ o_\infty\rangle$ by an infinite derivation t, then $c \xrightarrow{\sigma}{}^\omega$ for some $\sigma \in O^\omega$ such that $\mathsf{it}_\infty(\sigma) = \mathsf{it}_\infty(\sigma_t)$.*

Proof. Let t be an infinite derivation for $c \Rightarrow \langle \infty,\ o_\infty\rangle$, and for each level i of the tree we have a rule of shape $\mathsf{div}(\rho_i, k_i, o^{i+1}_\infty)$ where $\rho_i = \mathsf{rule}(j^i_1 \ldots j^i_{n_i}, j^i, c_i, o'_i)$, and denote by t_i the subtree starting with such rule. We assume $\sigma_t = (o_i)_{i\in\mathbb{N}}$, hence $\sigma_{t_i} = o_i{:}\sigma_{t_{i+1}}$.

Consider the sequence of configurations $(c_i)_{i\in\mathbb{N}}$, hence $\langle \mathcal{I}_\infty,\ C\rangle \vdash c_i \Rightarrow \langle \infty,\ o^i_\infty\rangle$ holds by the derivation t_i, then, by Lemma B.8, we get that $c_i \xrightarrow{v_i}{}^\star c'_i$ where either $k_i \le n_i$ and $\mathsf{pr}(c'_i) = c_{i+1}$ or c'_i is reduced (hence $k_i = n_i + 1$) and $c'_i \xrightarrow{o'_i} c_{i+1}$. Then, let us define

$$v'_i = \begin{cases} v_i & k_i \le n_i \\ v_i o'_i & k_i = n_i + 1 \end{cases}$$

then $\mathsf{it}(v'_i) = o_i$ by definition, hence, if σ is the flattening of the sequence $(v'_i)_{i\in\mathbb{N}}$, we get that $\mathsf{p}(\sigma) = \mathsf{p}(\sigma_t)$ by the infinite associative law (see Def. 4.1). Finally, since σ and σ_t are both infinite sequences, we have that $\mathsf{it}_\infty(\sigma) = \mathsf{p}(\sigma) = \mathsf{p}(\sigma_t) = \mathsf{it}_\infty(\sigma_t)$ as needed.

Now we have to prove that $c \xrightarrow{\sigma}{}^\omega$. First of all, note that it is easy to check that there is a strictly increasing sequence of natural numbers $(h_i)_{i\in\mathbb{N}}$ such that v'_{h_i} is not empty and for all k s.t. $h_i < k < h_{i+1}$ or $k < h_0$ we have $v'_k = \varepsilon$, because, otherwise, if for all $i \ge k$ we have $v'_i = \varepsilon$, we would also have $c'_i = c_i$, $k_i \le n_i$ and $\mathsf{pr}(c_i = c_{i+1})$, and this is not possible since $(c_i)_{i\ge k}$ is an infinite descending chain in pr, which is well-founded. Then, we can decompose σ into a sequence $(\sigma_i)_{i\in\mathbb{N}}$ such that $\sigma = \sigma_0$ and $\sigma_i = v'_{h_i}\sigma_{i+1}$, since all other v'_k are empty; furthermore, for all $i \in \mathbb{N}$, we know that $c_{h_i} \xrightarrow{v'_{h_i}}{}^\star c''_{h_i}$ such that, either $c''_{h_i} = c'_{h_i}$ and $\mathsf{pr}^{d_i}(c''_{h_i}) = c_{h_{i+1}}$, with $d_i = h_{i+1} - h_i$, or $c''_{h_i} = c_{h_i+1}$ and $\mathsf{pr}^{d_i}(c''_{h_i}) = c_{h_{i+1}}$, with $d_i = h_{i+1} - h_i - 1$.

We can inductively construct a sequence $(c^*_i)_{i\in\mathbb{N}}$ such that $c^*_0 = c_0$, $\mathsf{pr}^{D_i}(c^*_i) = c_{h_i}$, for $D_i = h_0 + d_0 + \ldots + d_{i-1}$, and $c^*_i \xrightarrow{v'_{h_i}}{}^\star c^*_{i+1}$ as follows:

[16]Note that this is needed only if some u_k is not empty, otherwise $\rho = \rho'$ and $c = c'$.

- $c_0^* = c_0$, hence, $D_0 = h_0$ and, by construction, $c_{h_0} = \mathsf{pr}^{h_0}(c_0)$
- since $c_{h_i} \xrightarrow{v'_i}{}^{\star} c''_{h_i}$, $\mathsf{pr}^{d_i}(c''_{h_i}) = c_{h_{i+1}}$ and $\mathsf{pr}^{D_i}(c_i^*) = c_{h_i}$, by iteratively applying Condition 2, we get that $c_i^* \xrightarrow{v'_i}{}^{\star} c''$, with $\mathsf{pr}^{D_i}(c'') = c''_{h_i}$. We set $c_{i+1}^* = c''$ and note that $\mathsf{pr}^{D_{i+1}}(c_{i+1}^*) = \mathsf{pr}^{D_i}(\mathsf{pr}^{d_i}(c_{i+1}^*)) = \mathsf{pr}^{D_i}(c''_{h_i}) = c_{h_{i+1}}$.

Finally, we can prove by coinduction on the definition of $\rightarrow^{\omega}$ that for all $i \in \mathbb{N}$, $c_i^* \xrightarrow{\sigma_i}{}^{\omega}$, since each prefix v'_{h_i} is not empty by construction. Hence, as a particular case we get $c \xrightarrow{\sigma}{}^{\omega}$, since $c_0 = c$ and $\sigma_0 = \sigma$. $\square$

Lemma B.10. *If $\langle \mathcal{I}_\infty \cup C_u^{\mathcal{I}}, \emptyset\rangle \vdash c \Rightarrow \langle r, o\rangle$, $\langle \mathcal{I}_\infty \cup C_u^{\mathcal{I}}, \emptyset\rangle \vdash c \Rightarrow \langle \infty, o_\infty\rangle$, then $o_\infty = o' \in O$ and $o' \preceq_* o$.*

Proof. The fact that $o_\infty = o' \in O$ can be easily proved by induction on the (finite) derivation for $c' \Rightarrow \langle \infty, o_\infty\rangle$ in $\mathcal{I}_\infty \cup C_{\mathsf{u}}^{\mathcal{I}}$. Then we have only to show that $o' \preceq_* o$, and this can be done by induction on the (finite) derivation for $c \Rightarrow \langle r, o\rangle$ in $\mathcal{I}_\infty \cup C_{\mathsf{u}}^{\mathcal{I}}$.

Base If $c = r$, then $o = \mathsf{u}$. Since the only rule with conclusion $r \Rightarrow \langle \infty, o_\infty\rangle$ is (co-unit), we get $o_\infty = \mathsf{u}$ and this proves the thesis since $\mathsf{u} \preceq_* \mathsf{u}$.

Induction If $c \notin R$, then it is derived by a rule ρ=$\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o_\rho)$. We have two cases:

- if $c \Rightarrow \langle \infty, o_\infty\rangle$ is derived by (co-unit), we get $o_\infty = \mathsf{u}$, hence $\mathsf{u} \preceq_* o$, since u is the least element in O.
- if $c \Rightarrow \langle \infty, o_\infty\rangle$ is derived by a rule $\mathsf{div}(\rho', i, o'_\infty)$ with ρ'=$\mathsf{rule}(j'_1 \ldots j'_m, j'_{m+1}, c, o'_\rho)$, with $m = n$ by Assumption 2, we can prove, by complete arithmetic induction, using Assumption 2, that, for all $k \in 1..i$, $C(j_k) = C(j'_k)$. Therefore, by induction hypothesis, we get $o'_\infty = o'' \preceq_* O(j_i)$. Furthermore, by Lemma 7.1, for all $k \in 1..i-1$, we get $R(j_k) = R(j'_k)$ and $O(j_k) = O(j'_k) = o_k$, and, in addition, if $i = n+1$, by Assumption 2 we also have $o_\rho = o_{\rho'}$.
  Hence, if $i \leq n$, then $o_\infty = o_1 * \cdots * o_{i-1} *_\infty o'_\infty$, thus, $o' = o_1 * \cdots * o_{i-1} * o'' \preceq_* o_1 * \cdots * o_{i-1} * O(j_i) \preceq_* o$; otherwise, $i = n+1$, $o_\infty = o_1 * \cdots * o_n * o_\rho *_\infty o'_\infty$, hence, $o' = o_1 * \cdots * o_n * o_\rho * o'' \preceq_* o_1 * \cdots * o_n * o_\rho * O(j_{n+1}) = o$.

$\square$

We can now prove the soundness result for $C_{\mathsf{u}}^{\mathcal{I}}$ (Theorem 7.3).

Proof of Theorem 7.3. In the proof, given a rule ρ=$\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$, we set $C(\rho, i) = C(j_i)$, $R(\rho, i) = R(j_i)$ and $O(\rho, i) = O(j_i)$.

Let t be the infinite derivation of $c \Rightarrow \langle \infty, o_\infty\rangle$ in $\mathcal{I}_\infty$, hence for each level i there is a rule $\mathsf{div}(\rho_i, k_i, o_\infty^{i+1})$, and let $\sigma_t = (o_i)_{i\in\mathbb{N}}$ be the infinite sequence associated with t (see the definition introduced for Lemma 7.6); hence $o_\infty = o_\infty^0$ and $o_\infty^i = o_i *_\infty o_\infty^{i+1}$. By definition, the judgement has also a finite derivation t' in $\mathcal{I}_\infty \cup C_{\mathsf{u}}^{\mathcal{I}}$. In t', at each level i, we have three possibilities: we have applied a divergence propagation rule $\mathsf{div}(\rho'_i, k_i, {o'_\infty}^{i+1})$, we have applied a divergence propagation rule $\mathsf{div}(\rho'_i, k'_i, {o'_\infty}^{i+1})$, or we have applied the coaxiom (co-unit).

Let h be the least level for which the first situation does not happen, then, for each level $i < h$, we are in the first one. We can prove, by a double induction, using Lemma 7.1 and Assumption 2, that, for all $i < h$ and $l \leq k_i$, $C(\rho_i) = C(\rho'_i)$, $C(\rho_i, l) = C(\rho'_i, l)$ and, if $l \neq k_i$, $O(\rho_i, l) = O(\rho'_i, l)$. This implies ${o'_\infty}^i = o_i *_\infty {o'_\infty}^{i+1}$ and $o_\infty = o_0 *_\infty {o'_\infty}^1$. Then we have two cases for level h:

- if we have applied the coaxiom, then either $h = 0$ and $o_\infty = \mathsf{u}$, or $h > 0$ and ${o_\infty}^{h-1} = o_{h-1}$. In both cases we get $o_\infty{=}o_0 \cdots o_{h-1} * o'_h$ with $o'_h = \mathsf{u}$.
- if we have applied a divergence propagation rule $\mathsf{div}(\rho, k, o'_\infty)$ with $k \neq k_h$, from what we have just proved, we have $C(\rho_h) = C(\rho_{h-1}, k_{h-1}) = C(\rho'_{h-1}, k_{h-1}) = C(\rho)$. Thus, we can prove, by complete arithmetic induction, using Lemma 7.1 and Assumption 2, that, for all $l \leq k' = \min\{k, k_h\}$, $C(\rho_h, l) = C(\rho, l)$ and, if $l \neq k'$, $O(\rho_h, l) = O(\rho, l)$. Hence, we get $k = \min\{k, k_h\}$, because, if $k > k_h$, then $C(\rho_h, k_k) \Rightarrow \langle R(\rho_h, k_k), O(\rho_h, k_h)\rangle$ is derivable, but also $C(\rho_h, k_h) \Rightarrow \langle \infty, {o_\infty}^{h+1}\rangle$ is derivable, and this is absurd by Lemma 7.2.

  Set $j_l = C(\rho_h, l) \Rightarrow \langle R(\rho_h, l), O(\rho_h, l)\rangle$, for all $l \leq k$. Then, since $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}}_{\mathsf{u}}, \emptyset\rangle \vdash j_k$ and $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}}_{\mathsf{u}}, \emptyset\rangle \vdash C(j_k) \Rightarrow \langle \infty, o'_\infty\rangle$, as $C(j_k) = C(\rho, k)$, by Lemma B.10, we get $o'_\infty = o'$ with $o' \preceq_* O(j_k)$. Therefore, ${o_\infty}^h = o'_h = O(j_1) * \cdots * O(j_{k-1}) * o' \preceq_* O(j_1) * \cdots * O(j_{k-1}) * O(j_k) \preceq_* o_h$, as $k < k_k$, and this implies $o_\infty = o_0 * \cdots * o_{h-1} * o'_h \preceq_* o_0 * \cdots * o_h$.

Therefore, in all cases, we have $o_\infty = o_0 * \cdots * o_{h-1} * o'_h *_\infty \mathsf{p}(\mathsf{u}^\omega) = \mathsf{p}(o_0 \ldots o_{h-1} o'_h \mathsf{u}^\omega)$ with $o'_h \preceq_* o_h$.

Now, by Lemma B.9 we have $c \xrightarrow{\sigma}^\omega$ and $\mathsf{it}_\infty(\sigma) = \mathsf{it}_\infty(\sigma_t)$ for some $\sigma \in O^\omega$, that is, $[\sigma]_\equiv = [\sigma_t]_\equiv$. Therefore, in order to get the thesis, it is enough to show that $\sigma_t \equiv v\mathsf{u}^\omega$, where $v = o_0 \ldots o_{h-1} o'_h$.

Set $\tau = v\mathsf{u}^\omega$, hence $\mathsf{it}_\infty(\tau) = o_\infty$. Then, we get $\tau \sqsubseteq \sigma_t$, since $\mathsf{it}(v) = o_0 * \cdots * o_{h-1} * o'_h \preceq_* o_0 * \cdots * o_h = \mathsf{it}(v')$, where $v' = o_0 \ldots o_h \triangleleft \sigma_t$. On the other hand, recall that, by construction, $\sigma_t = (o_i)_{i \in \mathbb{N}}$ and, for all $n \in \mathbb{N}$, $o_\infty = o_0 * \cdots * o_n *_\infty {o_\infty}^{n+1} = \mathsf{it}_\infty(o_0 \ldots o_n \sigma_{n+1})$ for some $\sigma_{n+1} \in O^\omega$. Hence, since $\mathsf{it}_\infty(\tau) = \mathsf{it}_\infty(o_0 \ldots o_n \sigma_{n+1})$, we have $\tau \equiv o_0 \ldots o_n \sigma_{n+1}$, for all $n \in \mathbb{N}$. Therefore, by definition of $\equiv$, there is a prefix $v' \triangleleft \tau$ such that $o_0 * \cdots * o_n \preceq_* \mathsf{it}(v')$ and, since $v' \triangleleft \tau$, $\mathsf{it}(v') \preceq_* \mathsf{it}(v)$. Hence, for all $n \in \mathbb{N}$, $o_0 * \cdots * o_n \preceq_* \mathsf{it}(v)$, thus, the least upper bound of $(o_0 * \cdots * o_n)_{n \in \mathbb{N}}$, if any, is below $\mathsf{it}(v)$, and this shows $\sigma_t \sqsubseteq \tau$. So, we have the thesis. □

We now prove the two results concerning the set of corules $C^{\mathcal{I}}$.

Proof of Lemma 7.6. At each level i of t we have a rule $\mathsf{div}(\rho_i, k_i, o^{i+1}_\infty)$, with $C(\rho_i) = c_i$, $c_0 = c$, $o^0_\infty = o_\infty$ and $o^i_\infty = o_i *_\infty o^{i+1}_\infty$. Since o'_∞ is a limit of the series on σ_t we have that $o'_\infty = o_0 * \cdots o_n *_\infty {o'_\infty}^{n+1}$ for all $n \in \mathbb{N}$, hence each judgement of shape $c_i \Rightarrow \langle \infty, {o'_\infty}^i\rangle$ is the consequence of the rule $\mathsf{div}(\rho_i, k_i, o^{i+1}_\infty)$ and this shows that we can build an infinite derivation for each of them.

To build finite proofs in $\mathcal{I}_\infty \cup C^{\mathcal{I}}$, it is enough to show a finite derivation for nodes at level i where $o_i \neq \mathsf{u}$, since the sequence is non-trivial, hence for each $o_i = \mathsf{u}$ there is $k > i$ such that $o_k \neq \mathsf{u}$. Suppose that $\rho_i = \mathsf{rule}(j^i_1 \ldots j^i_{n_i}, j^i, c_i, o'_i)$, then we have two cases:

- if $k_i = n_i + 1$ and for all $k \leq n_i$ we have $O(j^i_k) = \mathsf{u}$, then $o'_i \neq \mathsf{u}$ and so we get the proof by $\mathsf{co\text{-}mul}(\rho_i, {o'_\infty}^{i+1})$.
- Otherwise, let j^i_k be the first premise such that $O(j^i_k) \neq \mathsf{u}$ with $k \leq \min\{k_i, n_i\}$; we know by hypothesis that $\langle \mathcal{I}_\infty, C^{\mathcal{I}}\rangle \vdash j^i_k$, hence by Lemma 7.5 we have $C(j^i_k) \rightsquigarrow \langle R(j^i_k), O(j^i_k)\rangle$ and, since $O(j^i_k) \neq \mathsf{u}$, then $C(j^i_k) \xrightarrow{o} c'$ with $O(j^i_k) = o * o'$ and $o \neq \mathsf{u}$. Therefore, by Lemma 7.5 and Condition 2 we get $c_i \xrightarrow{v}^\star c'_i$ with $v = \mathsf{u}^h$ and $\mathsf{pr}(c'_i) = C(j^i_k)$, then, again by Condition 2, we get $c'_i \xrightarrow{o} c''_i$, hence, since ${o'_\infty}^i = o *_\infty o''_\infty$, for some o''_∞, by Lemma B.2 we

get $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}}, \emptyset\rangle \vdash c_i'' \Rightarrow \langle\infty, o *_\infty o''_\infty\rangle$, and by iterating Lemma B.1 we get $\langle \mathcal{I}_\infty \cup C^{\mathcal{I}}, \emptyset\rangle \vdash c_i \Rightarrow \langle\infty, o^i_\infty\rangle$ as needed.

□

Proof of Prop. 7.4. Since $\mathrm{E}_{\mathcal{I}} \subseteq O_{\mathcal{I}}$, the implication $\Leftarrow$ is trivial. To prove the other direction, consider $\sigma \in O^\omega_{\mathcal{I}}$. We first show that there exists $\tau \in \mathrm{E}^\omega_{\mathcal{I}}$ such that $\sigma <: \tau$. Assume $\sigma = (o_i)_{i\in\mathbb{N}}$, then, by definition of $O_{\mathcal{I}}$, we have that $o_i = o_{i1} * \cdots * o_{in_i}$ with $o_{i1}, \ldots, o_{in_i} \in \mathrm{E}_{\mathcal{I}}$; hence if τ is the sequence of all o_{ik_i} in the correct order we get the thesis. Since O is left-continuous (in particular $\leq_*$ is a partial order), we have that σ is trivial iff τ is trivial. Then, it is easy to check that o_∞ is a limit product of σ iff it is a limit product of τ, and this gives the thesis. □

Proof of Theorem 7.5. Let t be the infinite derivation of $c \Rightarrow \langle\infty, o_\infty\rangle$ and $\sigma_t = (o_i)_{i\in\mathbb{N}}$, then by Lemma B.9 there is $\sigma \in O^\omega$ such that $c \xrightarrow{\sigma}{}^\omega$ and $\mathsf{it}_\infty(\sigma) = \mathsf{it}_\infty(\sigma_t)$. We have two cases:

- if σ_t is trivial, that is, for all $k \geq n$, $o_k = \mathsf{u}$, for some $n \in \mathbb{N}$, then $\langle \mathcal{I}_\infty, C^{\mathcal{I}}_{\mathsf{u}}\rangle \vdash c \Rightarrow \langle\infty, o_\infty\rangle$, since we can cut t at depth n using (co-unit), hence by Theorem 7.3 we get the thesis;
- if σ_t is non-trivial, then, since $\mathsf{it}_\infty(\sigma_t)$ and o_∞ are both limits of the series on σ_t and $\mathrm{E}_{\mathcal{I}}$ has unique limits, hence $O_{\mathcal{I}}$ has unique limits as well, we have $o_\infty = \mathsf{it}_\infty(\sigma_t) = \mathsf{it}_\infty(\sigma)$, hence $c \rightsquigarrow \langle\infty, o_\infty\rangle$ as needed.

□

C. Other proofs

Proof of Theorem 5.1. We prove that the following conditions are equivalent, where points 1 and 4 are those in the statement of the theorem:

1. $\langle \mathcal{I}_\infty, C\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$ and $r_\infty \in R$ and $o_\infty \in O$
2. $\langle \mathcal{I}_\infty \cup C, \emptyset\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$ and $r_\infty \in R$ and $o_\infty \in O$
3. $\langle \mathcal{I}_\infty, \emptyset\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$
4. $\langle \mathcal{I}, \emptyset\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$.

It is easy to check that $r_\infty \in R$ implies $o_\infty \in O$, hence we do not care about o_∞.

The implication 1 $\Rightarrow$ 2 trivially holds by definition of derivability in an inference system with corules.

Suppose now we have a finite proof tree t in $\mathcal{I}_\infty \cup C$ for the judgement $c \Rightarrow \langle r_\infty, o_\infty\rangle$, then it can be proved by induction on the depth of the tree that $r_\infty = \infty$ if and only if we have applied a corule in t. This is due to the fact that in $\mathcal{I}_\infty$ all rules having a divergent judgement in the conclusion have also a divergent judgement in the premises and viceversa. Therefore we get ($\star$) that $r_\infty \in R$ if and only if we do not use corules in t, if and only if t is a finite proof tree in $\mathcal{I}_\infty$, if and only if t is a finite proof tree in $\mathcal{I}$, since without corules also rules for divergence propagation are not applicable. This proves the equivalences 2 $\Leftrightarrow$ 3 $\Leftrightarrow$ 4.

Finally, if $\langle \mathcal{I}_\infty, \emptyset\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$, then $r_\infty \in R$ by ($\star$), and $\langle \mathcal{I}_\infty, C\rangle \vdash c \Rightarrow \langle r_\infty, o_\infty\rangle$ since derivability is preserved by the addition of corules. □

Proof of Lemma 5.1. It follows by a straightforward induction on rules. □

Proof of Lemma 7.1. The proof is by induction on the derivation of $c \Rightarrow \langle r,\ o\rangle$, and denote by RH the induction hypothesis.

Base We have $c = r$ and $o = \mathsf{u}$, hence $r' = r$ and $o' = \mathsf{u}$, as the only rule for results is the axiom.

Induction If $c \notin R$, then $c \Rightarrow \langle r,\ o\rangle$ is derived by a rule $\rho{=}\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o_\rho)$ and $c \Rightarrow \langle r',\ o'\rangle$ is derived by a rule $\rho'{=}\mathsf{rule}(j'_1 \ldots j'_m, j'_{m+1}, c, o_{\rho'})$, with $m = n$ by Assumption 2. We prove, by complete arithmetic induction, that, for all $k \in 1..n+1$, $j_k = j'_k$. Let $k \in 1..n+1$, then, by induction hypothesis, we know that, for all $h < k$, $j_h = j'_h$, hence, in particular, $R(j_h) = R(j'_h)$. Thus, by Assumption 2, we get $C(j_k) = C(j'_k)$. Then, by RH, we get $R(j_k) = R(j'_k)$ and $O(j_k) = O(j'_k)$, that is, $j_k = j'_k$.

Therefore, we immediately get $r = R(j_{n+1}) = R(j'_{n+1}) = r'$. Furthermore, we know that $o = O(j_1) * \cdots * O(j_n) * o_\rho * O(j_{n+1})$ and $o' = O(j'_1) * \cdots * O(j'_n) * o_{\rho'} * O(j'_{n+1})$, by hypothesis, and we have just proved that, for all $k \in 1..n+1$, $O(j_k) = O(j'_k)$, hence, to conclude, it is enough to show $o_\rho = o_{\rho'}$, but this follows from Assumption 2, since we know that, for all $k \in 1..n$, $R(j_k) = R(j'_k)$.

□

Proof of Lemma 7.2. Thanks to Theorem 5.1, we proceed by induction on the derivation of $c \Rightarrow \langle r,\ o\rangle$.

Base If $c = r$, then it cannot diverge because there is no divergence propagation rule having a result in the conclusion.

Induction If $c \notin R$, then $c \Rightarrow \langle r,\ o\rangle$ is derived by a rule $\rho{=}\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o')$. Assume now that $\langle \mathcal{I}_\infty,\ C\rangle \vdash c \Rightarrow \langle \infty,\ o_\infty\rangle$ for some $o_\infty \in O_\infty$, then we have applied a rule $\mathsf{div}(\rho', i, o'_\infty)$ where $\rho'{=}\mathsf{rule}(j'_1 \ldots j'_m, j'_{m+1}, c, o'')$ with $m = n$ by Assumption 2. We prove, by complete arithmetic induction, that, for all $k \in 1..i$, $C(j_i) = C(j'_i)$. Let $k \in 1..i$, then, by induction hypothesis, we have, for all $h < k$, $C(j_h) = C(j'_h)$, and, by Lemma 7.1, we get $R(j_h) = R(j'_h)$. Thus, by Assumption 2, we get $C(j_k) = C(j'_k)$.

In particular we have $C(j_i) = C(j'_i)$ and, by hypothesis, we have $\langle \mathcal{I}_\infty,\ C\rangle \vdash C(j'_i) \Rightarrow \langle \infty,\ o'_\infty\rangle$, which is not possible by induction hypothesis, since $\langle \mathcal{I}_\infty,\ C\rangle \vdash j_i$.

□

Proof of Lemma 7.3. It immediately follows from Lemma B.1 and Theorem 5.1. □

Proof of Lemma 7.4. By induction on the number of steps, using Lemma 7.3. □

Proof of Lemma 7.5. We reason by induction on rules.

Base Axioms have conclusion $r \Rightarrow \langle r,\ \mathsf{u}\rangle$, and $r \rightsquigarrow \langle r,\ \mathsf{u}\rangle$ trivially holds since $r \xrightarrow{\varepsilon}{}^{\star}\ r$ and $\mathsf{it}(\varepsilon) = \mathsf{u}$.

Induction Given a rule $\mathsf{rule}(j_1 \ldots j_n, j_{n+1}, c, o)$ we assume that for all $i \in 1..n+1$ we have $C(j_i) \rightsquigarrow \langle R(j_i),\ O(j_i)\rangle$, then the thesis follows from an iteration of Condition 2.

□

D. Small-step rules of the examples

In this section we report the axioms of the small-step relation of the examples in Sect. 6. The definition of evaluation contexts is pretty standard, following the early detection of stuck terms as explained in Sect. 3, hence we omit it, as well as the rule for contextual closure, since it is again standard.

I/O events

$$((\lambda x.e)\,v;\sigma) \xrightarrow{\varepsilon} (e[v/x];\sigma) \qquad (\mathtt{out}\,v;\sigma) \xrightarrow{\mathtt{out}\,v} (v;\sigma) \qquad (\mathtt{in};v{:}\sigma) \xrightarrow{\mathtt{in}\,v} (v;\sigma)$$

I/O costs

$$((\lambda x.e)\,v;\sigma) \xrightarrow{0} (e[v/x];\sigma) \qquad (\mathtt{out}\,v;\sigma) \xrightarrow{c_{\mathtt{out}}(v)} (v;\sigma) \qquad (\mathtt{in};v{:}\sigma) \xrightarrow{c_{\mathtt{in}}(v)} (v;\sigma)$$

Executed branches

$$(\lambda x.e)\,v \xrightarrow{\emptyset} e[v/x] \qquad \mathit{true}\ ?_a\, e_1 : e_2 \xrightarrow{\{a.\mathit{true}\}} e_1 \qquad \mathit{false}\ ?_a\, e_1 : e_2 \xrightarrow{\{a.\mathit{false}\}} e_2$$

Heap size

$$((\lambda x.e)\,v;\mathcal{H}) \xrightarrow{|\mathcal{H}|} (e[v/x];\mathcal{H})$$

$$(\mathtt{ref}\,v;\mathcal{H}) \xrightarrow{|\mathcal{H}|+1} (\iota;\mathcal{H} \uplus \{\iota \mapsto v\}) \qquad (\mathtt{free}\,\iota;\mathcal{H} \uplus \{\iota \mapsto v\}) \xrightarrow{|\mathcal{H}|} (v;\mathcal{H})$$

$$(!\,\iota;\mathcal{H} \uplus \{\iota \mapsto v\}) \xrightarrow{|\mathcal{H}|+1} (v;\mathcal{H} \uplus \{\iota \mapsto v\}) \qquad (\iota = v;\mathcal{H} \uplus \{\iota \mapsto v'\}) \xrightarrow{|\mathcal{H}|+1} (v;\mathcal{H} \uplus \{\iota \mapsto v\})$$