Bi-inductive Structural Semantics [*]

Patrick Cousot

*Département d'informatique, École normale supérieure, 45 rue d'Ulm,
75230 Paris cedex 05, France*

Radhia Cousot

CNRS & École polytechnique, 91128 Palaiseau cedex, France

Abstract

We propose a simple order-theoretic generalization, possibly non monotone, of set-theoretic inductive definitions. This generalization covers inductive, co-inductive and bi-inductive definitions and is preserved by abstraction. This allows structural operational semantics to describe simultaneously the finite/terminating and infinite/diverging behaviors of programs. This is illustrated on grammars and the structural bifinitary small/big-step trace/relational/operational semantics of the call-by-value λ-calculus (for which co-induction is shown to be inadequate).

Key words: fixpoint definition, inductive definition, co-inductive definition, bi-inductive definition, non-monotone definition, grammar, structural operational semantics, SOS, trace semantics, relational semantics, small-step semantics, big-step semantics, divergence semantics.

---

1 Introduction

The connection between the use of fixpoints in *denotational semantics* [24] and the use of rule-based inductive definitions in *axiomatic semantics* [15] and

---

structural operational semantics (SOS) [28,30,29] can be made by a generalization of inductive definitions [2] to include co-inductive definitions [11]. It is then possible to generalize *natural semantics* describing finite input/output behaviors [17] so as to also include infinite behaviors [10]. This is necessary since the definition of the infinite behaviors cannot be derived from the finite big-step SOS behaviors.

1.1 Motivating example

Let us consider for example the choice operator $E_1 \mid E_2$ where the evaluation of expression E_1 either terminates (returning the value a, written $E_1 \Rightarrow a$) or does not terminate (written $E_1 \Rightarrow \perp$). Similarly for expression E_2, either $E_2 \Rightarrow b$ or $E_2 \Rightarrow \perp$. For the semantics of the choice operator, we have three possible results $\{r \mid E_1 \mid E_2 \Rightarrow r\} \subseteq \{a, b, \perp\}$, depending upon its operational semantics. Several alternatives are considered below.

- *Nondeterministic*: an internal choice is made initially to evaluate E_1 or to evaluate E_2;

$E_1 \mid E_2$		E_1	
		a	$\perp$
E_2	b	$\{a, b\}$	$\{\perp, b\}$
	$\perp$	$\{a, \perp\}$	$\{\perp\}$

- *Parallel*: evaluate E_1 and E_2 concurently, with an unspecified scheduling, and return the first available result a or b;

$E_1 \mid E_2$		E_1	
		a	$\perp$
E_2	b	$\{a, b\}$	$\{b\}$
	$\perp$	$\{a\}$	$\{\perp\}$

- *Eager:* evaluate E_1 (or respectively E_2) first and then E_2 (resp. E_1) and return either result a or b;

$E_1 \mid E_2$		E_1	
		a	$\perp$
E_2	b	$\{a, b\}$	$\{\perp\}$
	$\perp$	$\{\perp\}$	$\{\perp\}$

- *Mixed left-to-right:* evaluate E_1 first and either return its result a or evaluate E_2 and return its results b;

$E_1 \mid E_2$		E_1	
		a	$\perp$
E_2	b	$\{a, b\}$	$\{\perp\}$
	$\perp$	$\{a, \perp\}$	$\{\perp\}$

2

- *Mixed right-to-left:* evaluate E_2 first and either return its result b or evaluate E_1 and return its results a;

		E_1	
$E_1 \mid E_2$		a	$\bot$
E_2	b	$\{a, b\}$	$\{\bot, b\}$
	$\bot$	$\{\bot\}$	$\{\bot\}$

Observe that all evaluations have exactly the same convergence big-step semantics. However, they differ on their divergence behaviors. It follows, for example, that an implementation of the natural semantics [17] will have its diverging behaviors undefined by the formal semantics hence determined by the behavior of the implementation. This is the case with left-to-right evaluation Prolog implementation [3,13], but the problem is general and concerns the class of all implementations that conform to the semantics, regardless of how they were produced. So the natural big-step convergence semantics is an abstract semantics of programs which is not an exact match for its concrete operational semantics. This shows the need to extend big-step/natural semantics to cope with infinite behaviors.

1.2   Summary

The paper develops and illustrates the use of "bi-inductive" definitions in operational semantics.

Bi-inductive definitions enable both finitary and infinitary behaviors to be described simultaneously [10,11].

Section 2 describes the general methodology. Hilbert proof systems [2] are extended by replacing the powerset $\langle \wp(U), \subseteq \rangle$ of the universe U by a complete partial order $\langle \mathcal{D}, \sqsubseteq \rangle$. The method for defining a map from a well-founded set to complete partial orders combines well-founded recursion and structural inductive definitions described by using different, but equivalent, forms: fixpoint definition, equational definition, constraint-based (inequational) definition, and rule-based definition.

Section 3 recalls a few elements of abstract interpretation, including soundness and completeness.

Section 4 is a simple illustration of this approach to give a trace semantics to transition systems [6].

The semantics of context-free grammars in Sect. 5 combines the classical definitions of the finite and infinite languages generated by a grammar, which can be recovered by simple abstractions.

Section 6 is an application to the call-by-value λ-calculus. We introduce an original big-step trace semantics that gives operational meaning to both convergent and divergent behaviors of programs. The compositional structural definition mixes induction for finite behaviors and co-induction for infinite behaviors while avoiding duplication of rules between the two cases. This big-step trace semantics excludes erroneous behaviors that go wrong. The other semantics are then systematically derived by abstraction.

The big-step trace semantics is first abstracted to a relational semantics and then to the standard big-step or natural semantics. These abstractions are sound and complete in that the big-step trace and relational semantics describe the same converging or diverging behaviors while the big-step trace and natural semantics describe the same finite behaviors. The big-step trace semantics is then abstracted into a small-step semantics, by collecting transitions along traces. This abstraction is sound but incomplete in that the traces generated by the small-step semantics describes convergent, divergent, but also erroneous behaviors of programs. This shows that trace-based operational semantics can be much more informative that small-step operational semantics.

2 Structural order-theoretic inductive definitions

We introduce different forms of structural order-theoretic inductive definitions and prove their equivalence.

2.1 Dcpos and complete lattices

Let $\langle S, \sqsubseteq \rangle$ be a poset [12]. A chain in the poset $\langle S, \sqsubseteq \rangle$ is a subset of S such that any two elements in the chain are comparable by $\sqsubseteq$. A directed complete partial order (dcpo) is a poset such that any chain has a least upper bound (lub denoted $\sqcup$). For the empty chain the lub is the infimum $\bot$ of S. A complete lattice is a poset such that any subset has a lub. If I is a set and $\langle S, \sqsubseteq \rangle$ is a poset (resp. dcpo, complete lattice) then the pointwise extension $\langle I \longrightarrow S, \dot{\sqsubseteq} \rangle$ with $f \dot{\sqsubseteq} g \triangleq \forall i \in I : f(i) \sqsubseteq g(i)$ is a poset (resp. dcpo, complete lattice) and similarly for the pointwise extension $\langle I' \longrightarrow (I \longrightarrow S), \ddot{\sqsubseteq} \rangle$ of $\langle I \longrightarrow S, \dot{\sqsubseteq} \rangle$.

2.2 Syntax

Structural inductive definitions are by induction on the syntactic structure of the program. We understand a language $\mathbb{L}$ as a set of non-empty "syntactic components" (including programs). For example, the λ-calculus has $\boldsymbol{\lambda} y \cdot \boldsymbol{\lambda} x \cdot a$,

y, $\boldsymbol{\lambda}\,x \cdot a$, x and a among its "syntactic components". A component is "atomic" or else has finitely many "strict subcomponents" such as y, $\boldsymbol{\lambda}\,x \cdot a$, x and a for $\boldsymbol{\lambda}\,y \cdot \boldsymbol{\lambda}\,x \cdot a$. For simplicity, these subcomponents are assumed to be distinct two-by-two (for example thanks to unique labels). The corresponding cover relation is $\ell \longrightarrow\!\!\!\prec \ell'$ on $\mathbb{L}$ meaning that ℓ is a "strict immediate syntactic sub-component" of ℓ'. For example, $y \longrightarrow\!\!\!\prec \boldsymbol{\lambda}\,y \cdot \boldsymbol{\lambda}\,x \cdot a$ and $\boldsymbol{\lambda}\,x \cdot a \longrightarrow\!\!\!\prec \boldsymbol{\lambda}\,y \cdot \boldsymbol{\lambda}\,x \cdot a$ while $x \longrightarrow\!\!\!\prec \boldsymbol{\lambda}\,x \cdot a$ and $a \longrightarrow\!\!\!\prec \boldsymbol{\lambda}\,x \cdot a$ but $a \;\not\!\!\longrightarrow\!\!\!\prec \boldsymbol{\lambda}\,y \cdot \boldsymbol{\lambda}\,x \cdot a$. As a shorthand reminiscent of the grammatical notation, we write $\boldsymbol{\lambda}\,y \cdot \boldsymbol{\lambda}\,x \cdot a ::= y,\ \boldsymbol{\lambda}\,x \cdot a$ and $\boldsymbol{\lambda}\,x \cdot a ::= x,\ a$ where the "strict immediate syntactic subcomponents" are given in left-to-right order (in fact any total order would do).

More generally, to completely abstract away from syntax, we let $\langle \mathbb{L},\ \preccurlyeq \rangle$ be a partially ordered set where $\preccurlyeq$ is well-founded and $\prec$ is the corresponding strict relation. We write $\longrightarrow\!\!\!\prec$ for the corresponding cover relation that is $x \longrightarrow\!\!\!\prec y$ if and only if $x \prec y$ and $\nexists z : x \prec z \prec y$. The cover relation $\longrightarrow\!\!\!\prec$ should have finite left images $\forall \ell \in \mathbb{L} : |\{\ell' \in \mathbb{L} \mid \ell' \longrightarrow\!\!\!\prec \ell\}| \in \mathbb{N}^{1}$. We let $\prod_{\ell' \longrightarrow\!\!\!\prec \ell} \ell'$ be the tuple of elements covered by ℓ and given in some total order $\prod_{\ell' \longrightarrow\!\!\!\prec \ell} \ell' = \ell_1, \ldots, \ell_n$ so that $\{\ell_1, \ldots, \ell_n\} = \{\ell' \in \mathbb{L} \mid \ell' \longrightarrow\!\!\!\prec \ell\}$ and write $\ell ::= \ell_1, \ldots, \ell_n$ for brevity with $n = 0$ for atoms (such that $\forall \ell' \in \mathbb{L} : \ell' \;\not\!\!\longrightarrow\!\!\!\prec \ell$).

2.3   Semantic domains

For each "component" $\ell \in \mathbb{L}$, we consider a semantic domain $\langle \mathcal{D}_\ell, \sqsubseteq_\ell, \perp_\ell, \sqcup_\ell \rangle$ which is assumed to be a dcpo.

2.4   Variables

For each "component" $\ell \in \mathbb{L}$, we consider variables X_ℓ, Y_ℓ, ... ranging over the semantic domain $\mathcal{D}_\ell$. We drop the subscript ℓ when the corresponding semantic domain is clear from the context (e.g. the semantic domain is the same for all "components" i.e. $\forall \ell \in \mathbb{L} : \mathcal{D}_\ell = \mathcal{D}$).

2.5   Transformers

For each "component" $\ell \in \mathbb{L}$, we let Δ_ℓ be indexed sequences (totally ordered sets). For example if the semantics of the "component" ℓ is defined by a sequence of rules labeled (R_1), ..., (R_n) in that order, then we can define $\Delta_\ell = R_1$, ...,

---

[1] $|S|$ is the cardinality of set S and $\mathbb{N}$ is the set of natural numbers.

R_n. We write $\prod_{i \in \Delta_\ell} x_i$ when considering the sequence $\langle x_i, \ i \in \Delta_\ell \rangle \in \Delta_\ell \longrightarrow S$ of elements of a set S as a vector of $\prod_{i \in \Delta_\ell} S$.

For each element $i \in \Delta_\ell$ of the sequence, we consider transformers $F_\ell^i \in \mathcal{D}_\ell \times \mathcal{D}_{\ell_1} \dots \times \mathcal{D}_{\ell_n} \longrightarrow \mathcal{D}_\ell$ where $\ell ::= \ell_1 \dots \ell_n$. When $n = 0$, we have $F_\ell^i \in \mathcal{D}_\ell \longrightarrow \mathcal{D}_\ell$.

The transformers are said to be $\sqsubseteq_\ell$-*monotone in their first parameter* ($\sqsubseteq_\ell$-*monotone* for brevity), whenever $\forall i \in \Delta_\ell, \ \ell ::= \ell_1, \dots, \ell_n, \ X, Y \in \mathcal{D}_\ell, X_1 \in \mathcal{D}_{\ell_1}, \dots, X_n \in \mathcal{D}_{\ell_n} \colon X \sqsubseteq_\ell Y \implies F_\ell^i(X, X_1, \dots, X_n) \sqsubseteq_\ell F_\ell^i(Y, X_1, \dots, X_n)$.

2.6   Join

For each "component" $\ell \in \mathbb{L}$, the *join* $\curlyvee_\ell \in (\Delta_\ell \longrightarrow \mathcal{D}_\ell) \longrightarrow \mathcal{D}_\ell$ is used to gather alternatives in formal definitions. For brevity, we write $\curlyvee_\ell (\prod_{i \in \Delta_\ell} X_i) = \curlyvee_\ell{}_{i \in \Delta_\ell} X_i$, leaving implicit the fact that the X_i should be considered in the total order given by the sequence Δ_ℓ.

Most often, the order of presentation of these alternatives in the formal definition is not significant. In this case, Δ_ℓ is just a set and the join may often be defined in term of a *binary join* $\curlyvee_\ell \in (\mathcal{D}_\ell \times \mathcal{D}_\ell) \longrightarrow \mathcal{D}_\ell$, which is assumed to be associative and commutative, as $\curlyvee_\ell (\prod_{i \in \Delta_\ell} X_i) \triangleq \curlyvee_\ell{}_{i \in \Delta_\ell} X_i$. The binary join may be different from the least upper bound (lub) $\sqcup_\ell$ of the semantic domain $\mathcal{D}_\ell$.

The join operator is said to be *componentwise $\sqsubseteq_\ell$-monotone* whenever ($\forall \langle X_i, \ i \in \Delta_\ell \rangle : \forall \langle Y_i, \ i \in \Delta_\ell \rangle : (\forall i \in \Delta_\ell : X_i \sqsubseteq_\ell Y_i) \implies \curlyvee_\ell (\prod_{i \in \Delta_\ell} X_i) \sqsubseteq_\ell \curlyvee_\ell (\prod_{i \in \Delta_\ell} Y_i))$.
This is the case when the binary join is $\sqsubseteq_\ell$-monotone,

2.7   Fixpoint definitions

A *fixpoint definition* has the form

$$\forall \ell \in \mathbb{L} : \mathcal{S}_f[\![\ell]\!] = \mathbf{lfp}^{\sqsubseteq_\ell} \mathcal{F}_f[\![\ell]\!]$$

$$\text{where} \qquad \mathcal{F}_f[\![\ell]\!] \triangleq \boldsymbol{\lambda} X \cdot \curlyvee_\ell{}_{i \in \Delta_\ell} F_\ell^i(X, \prod_{\ell' \prec \ell} \mathcal{S}_f[\![\ell']\!])$$

6

and $\mathbf{lfp}^{\sqsubseteq}$ is the partially defined $\sqsubseteq$-least fixpoint operator on a poset $\langle P, \sqsubseteq \rangle$ [2]. To emphasize structural composition when $\ell ::= \ell_1, \ldots, \ell_n$, we write

$$\forall \ell \in \mathbb{L} : \mathcal{S}_f[\![\ell ::= \ell_1, \ldots, \ell_n]\!] = \mathbf{lfp}^{\sqsubseteq_\ell} \, \lambda \, X \bullet \bigvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(X, \mathcal{S}_f[\![\ell_1]\!], \ldots, \mathcal{S}_f[\![\ell_n]\!]) \ .$$

Hypothesis 1 *It is assumed that the least fixpoint* $\mathbf{lfp}^{\sqsubseteq_\ell} \mathcal{F}_f[\![\ell]\!]$ *does exist.* □

Hyp. 1 holds in the event of monotony.

Lemma 2 *If* $\lambda \, X \bullet F_\ell^i(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$ *is monotone for all* $i \in \Delta_\ell$ *and* Υ_ℓ *is monotone then* $\forall \ell \in \mathbb{L} : \mathcal{S}_f[\![\ell]\!]$ *is well defined.* □

PROOF Assume, by induction on $\prec$, that $\mathcal{S}_f[\![\ell']\!]$ is well-defined for all $\ell' \prec\!\!\prec \ell$. $\lambda \, X \bullet \bigvee_{\substack{\ell \\ i \in \Delta}} F_\ell^i(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$ is monotone since $\lambda \, X \bullet F_\ell^i(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$ is monotone for all $i \in \Delta_\ell$ and Υ_ℓ is monotone by hypothesis. It follows that the least fixpoint $\mathbf{lfp}^{\sqsubseteq_\ell} \mathcal{F}_f[\![\ell]\!]$ does exist in the dcpo $\langle \mathcal{D}_\ell, \sqsubseteq_\ell \rangle$ as shown by [7] [3] (or [27] without the axiom of choice, see [18,21] for historical perspectives), proving that $\mathcal{S}_f[\![\ell]\!]$ is well-defined. ∎

Definitions without fixpoint or join can nevertheless be encompassed as fixpoints such as $\bigvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(\mathcal{S}_f[\![\ell_1]\!], \ldots, \mathcal{S}_f[\![\ell_n]\!]) = \mathbf{lfp}^{\sqsubseteq_\ell} \, \lambda \, X \bullet \bigvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(\mathcal{S}_f[\![\ell_1]\!], \ldots, \mathcal{S}_f[\![\ell_n]\!])$ or without join $F_\ell^i(\mathcal{S}_f[\![\ell_1]\!], \ldots, \mathcal{S}_f[\![\ell_n]\!]) = \mathbf{lfp}^{\sqsubseteq_\ell} \, \lambda \, X \bullet \bigvee_{\substack{\ell \\ i' \in \{i\}}} F_\ell^{i'}(\mathcal{S}_f[\![\ell_1]\!], \ldots, \mathcal{S}_f[\![\ell_n]\!])$.

2.8   Equational definitions

An *equational definition* has the form:

$\langle \mathcal{S}_e[\![\ell]\!], \ell \in \mathbb{L} \rangle$ *is the componentwise* $\sqsubseteq_\ell$*-least* $\langle X_\ell, \ell \in \mathbb{L} \rangle$ *satisfying the system of equations*

$$X_\ell = \bigvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(X_\ell, \prod_{\ell' \prec\!\!\prec \ell} X_{\ell'}), \qquad \ell \in \mathbb{L} \ .$$

---

[2] We write $\mathbf{lfp}_a^{\sqsubseteq} f$ for the $\sqsubseteq$-least fixpoint of $f \in P \longrightarrow P$ which is $\sqsubseteq$-greater than or equal to $a \in P$ if any. If P has an infimum $\bot \in P$ then $\mathbf{lfp}^{\sqsubseteq} f = \mathbf{lfp}_\bot^{\sqsubseteq} f$. The dual partially defined greatest fixpoint operator is $\mathbf{gfp}^{\sqsubseteq}$.

[3] The complete lattice hypothesis is not used in [7] to prove the existence of the least fixpoint of monotone partial functions on a poset. It follows from the well-definedness of transfinite iterates from pre-fixpoints, in particular for limit ordinals. This hypothesis, which is weaker than dcpos, would also be sufficient in this paper when assuming monotony.

Lemma 3 *If* **Hyp. 1** *holds then* $\forall \ell \in \mathbb{L} : \mathcal{S}_e[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$. □

PROOF We prove, by induction on $\prec$, that the componentwise $\sqsubseteq_\ell$-least such $\langle X_\ell, \ell \in \mathbb{L} \rangle$ satisfies $\forall \ell \in \mathbb{L} : \mathcal{S}_e[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$. For the base case $\nexists \ell' \prec\!\!\!\!-\!\prec \ell$, $\mathcal{S}_e[\![\ell]\!]$ is the $\sqsubseteq_\ell$-least X_ℓ such that $X_\ell = \curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(X_\ell)$ that is $\mathcal{S}_f[\![\ell]\!] = $
$\mathbf{lfp}^{\sqsubseteq_\ell} \boldsymbol{\lambda} X \bullet \curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(X)$ by definition and existence of the $\sqsubseteq_\ell$-least fixpoint.
Otherwise $X_{\ell'} = \mathcal{S}_e[\![\ell']\!] = \mathcal{S}_f[\![\ell']\!]$ for all $\ell' \prec\!\!\!\!-\!\prec \ell$ by induction hypothesis and so $\mathcal{S}_e[\![\ell]\!]$ is the $\sqsubseteq_\ell$-least X_ℓ such that $X_\ell = \curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(X_\ell, \prod\limits_{\ell' \prec\!\!-\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$, that is
$\mathbf{lfp}^{\sqsubseteq_\ell} \mathcal{F}_f[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$ by definition and existence of the $\sqsubseteq_\ell$-least fixpoint. ∎
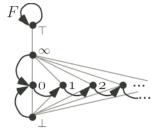
2.9 Constraint-based definitions

A *constraint-based definition* has the form:

$\langle \mathcal{S}_c[\![\ell]\!], \ell \in \mathbb{L} \rangle$ is the componentwise $\sqsubseteq_\ell$-least $\langle X_\ell, \ell \in \mathbb{L} \rangle$ satisfying the system of constraints (inequations)

$$\curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(X_\ell, \prod\limits_{\ell' \prec\!\!-\!\prec \ell} X_{\ell'}) \sqsubseteq_\ell X_\ell, \qquad \ell \in \mathbb{L}\,.$$

Lemma 4 *If* $\forall \ell \in \mathbb{L}$, $\mathcal{F}_f[\![\ell]\!]$ *is* $\sqsubseteq_\ell$-*monotone then* $\forall \ell \in \mathbb{L} : \mathcal{S}_c[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$. □

PROOF We prove, by induction on $\prec$, that $\mathcal{S}_c[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$. Assume this is true for all $\ell' \prec\!\!\!\!-\!\prec \ell$. So $\mathcal{S}_c[\![\ell]\!]$ is the $\sqsubseteq_\ell$-least X_ℓ such that $\curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(X_\ell, \prod\limits_{\ell' \prec\!\!-\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$
$\sqsubseteq_\ell X_\ell$. Then the fixpoint property $\mathcal{S}_f[\![\ell]\!] = \curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(\mathcal{S}_f[\![\ell]\!], \prod\limits_{\ell' \prec\!\!-\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$ implies
$\curlyvee_\ell\limits_{i \in \Delta_\ell} F_\ell^i(\mathcal{S}_f[\![\ell]\!], \prod\limits_{\ell' \prec\!\!-\!\prec \ell} \mathcal{S}_f[\![\ell']\!]) \sqsubseteq_\ell \mathcal{S}_f[\![\ell]\!]$ since $\sqsubseteq_\ell$ is reflexive, proving that at least one such X_ℓ does exist. By transfinite induction, all transfinite iterates for $\mathcal{F}_f[\![\ell]\!]$ from $\bot_\ell$ (which do exist in a dcpo [7]) are $\sqsubseteq_\ell$-less than or equal to any such X_ℓ. Because $\mathcal{S}_f[\![\ell]\!] = \mathbf{lfp}^{\sqsubseteq_\ell} \mathcal{F}_f[\![\ell]\!]$ is one of these iterates we conclude that $\mathcal{S}_c[\![\ell]\!]$ does exist and, by antisymmetry, is $\mathcal{S}_f[\![\ell]\!]$. ∎

In absence of monotony, as shown on the opposite example, the least fixpoint definition and the constraint-based definition may not coincide, since $0 = F(\infty) \sqsubset \infty \sqsubset \top = \mathbf{lfp}^{\sqsubseteq} F$

2.10 Rule-based definitions

A *rule-based definition* is a sequence of rules of the form

$$\frac{X_\ell}{F_\ell^i(X_\ell, \displaystyle\prod_{\ell' \rightarrowtail \ell} \mathcal{S}_r[\![\ell']\!])} \sqsubseteq_\ell \qquad \ell \in \mathbb{L}, \quad i \in \Delta_\ell$$

where the premise and conclusion are elements of the $\langle \mathcal{D}_\ell, \sqsubseteq_\ell \rangle$ cpo. When understanding the rule in logical form (where the premise is a statement that is assumed to be true and from which a conclusion can be drawn), the following form might be preferred.

$$\frac{X_\ell \sqsubseteq_\ell \mathcal{S}_r[\![\ell]\!]}{F_\ell^i(X_\ell, \displaystyle\prod_{\ell' \rightarrowtail \ell} \mathcal{S}_r[\![\ell']\!]) \sqsubseteq_\ell \mathcal{S}_r[\![\ell]\!]} \sqsubseteq_\ell \qquad \ell \in \mathbb{L}, \quad X_\ell \in \mathcal{D}_\ell, \quad i \in \Delta_\ell$$

If F_ℓ^i does not depend upon the premise X_ℓ, it is an axiom. In such presentations, the join $\curlyvee_\ell$ of the alternatives is left implicit[4]. To make it explicit, we rewrite such definitions in the form

$$\frac{X_\ell \sqsubseteq_\ell \mathcal{S}_r[\![\ell]\!]}{\displaystyle\curlyvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(X_\ell, \displaystyle\prod_{\ell' \rightarrowtail \ell} \mathcal{S}_r[\![\ell']\!]) \sqsubseteq_\ell \mathcal{S}_r[\![\ell]\!]} \sqsubseteq_\ell \qquad \ell \in \mathbb{L}, \quad X_\ell \in \mathcal{D}_\ell \ . \qquad (1)$$

The formal definition of the join makes explicit whether the order of presentation of the rules does matter, or not. When it doesn't, the join can be defined using a binary associative and commutative join. This binary join can even be left implicit and, by associativity and commutativity, the rules can be given in any order. This will be the case for the examples provided in Sect. 5 and Sect. 6.

The *meaning* of a rule-based definition (1) is

$$\mathcal{S}_r[\![\ell]\!] \triangleq \mathbf{lfp}^{\sqsubseteq_\ell} \ \boldsymbol{\lambda} X \bullet \curlyvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(X, \prod_{\ell' \rightarrowtail \ell} \mathcal{S}_f[\![\ell']\!]) \ .$$

where, by **Hyp. 1**, the fixpoint of the consequence operator is assumed to exist.

---

[4] This is the case in Hilbert's formal systems, see Sect. 2.12.

A $D \in \mathcal{D}_\ell$ is *provable* if and only if it has a *proof* that is a transfinite sequence [5] $D_0, \ldots, D_\lambda$ of elements of $\mathcal{D}_\ell$ such that $D_0 = \perp_\ell$, $D_\lambda = D$ and for all $0 < \delta \leqslant \lambda$,
$$D_\delta \sqsubseteq_\ell \bigcurlyvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(\bigsqcup_{\substack{\ell \\ \beta < \delta}} D_\beta, \prod_{\ell' \prec \ell} \mathcal{S}_r[\![\ell']\!]).$$

The *proof-theoretic meaning* of a rule-based definition (1) is

$$\mathcal{S}_p[\![\ell]\!] \triangleq \bigsqcup_\ell \{D \in \mathcal{D}_\ell \mid D \text{ is provable}\} .$$

Lemma 5 *If $\forall \ell \in \mathbb{L}$, $\mathcal{F}_f[\![\ell]\!]$ is $\sqsubseteq_\ell$-monotone then $\forall \ell \in \mathbb{L} : \mathcal{S}_p[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!].$* □
PROOF The proof is by induction on $\prec$, so assume $\forall \ell' \prec \ell : \mathcal{S}_p[\![\ell']\!] = \mathcal{S}_f[\![\ell']\!]$. The limit $\mathcal{S}_f[\![\ell]\!]$ of the ultimately stationary transfinite iterates for $\boldsymbol{\lambda} X \cdot \bigcurlyvee_{\substack{\ell \\ i \in \Delta_\ell}} F_\ell^i(X, \prod_{\ell' \prec \ell} \mathcal{S}_p[\![\ell']\!]) = \mathcal{F}_f[\![\ell]\!]$ from $\perp_\ell$ (which does exist in a dcpo [7]) belongs to $\{D \in \mathcal{D}_\ell \mid D \text{ is provable}\}$ since $\sqsubseteq_\ell$ is reflexive. Any other proof is upper-bounded by these iterates and so if D *is provable* then $D \sqsubseteq_\ell \mathcal{S}_f[\![\ell]\!]$ proving that the least upper bound (lub) $\bigsqcup_\ell \{D \in \mathcal{D}_\ell \mid D \text{ is provable}\}$ does exist and is precisely $\mathcal{S}_f[\![\ell]\!]$. ∎

2.11 Equivalence of the order-theoretic inductive definitions

Theorem 6 *Hyp. 1 implies that $\forall \ell \in \mathbb{L} : \mathcal{S}[\![\ell]\!] \triangleq \mathcal{S}_f[\![\ell]\!] = \mathcal{S}_e[\![\ell]\!] = \mathcal{S}_r[\![\ell]\!]$. If $\forall \ell \in \mathbb{L}$, $\mathcal{F}_f[\![\ell]\!]$ is $\sqsubseteq_\ell$-monotone then $\mathcal{S}[\![\ell]\!] = \mathcal{S}_c[\![\ell]\!] = \mathcal{S}_p[\![\ell]\!]$.* □

This generalization of [2] could also include a game-theoretic version (the game semantics [1] being of quite different nature). The closure-condition version [2] is also easy to adapt.

2.12 Example: inductive definitions

The classical inductive definition [2] of a subset $\mathcal{S}$ of a universe U by rules $\left\{ \frac{P_i}{c_i} \mid i \in I \right\}$ where $P_i \subseteq U$ and $c_i \in U$, $i \in I$ can be written $\dfrac{X \subseteq \mathcal{S}}{\{c_i \mid P_i \subseteq X\} \subseteq \mathcal{S}}_\subseteq$, $i \in I$ or $\dfrac{P_i \subseteq X, \quad X \subseteq \mathcal{S}}{c_i \in \mathcal{S}}_\subseteq$, $i \in I$ that is $\dfrac{P_i \subseteq \mathcal{S}}{c_i \in \mathcal{S}}_\subseteq$, $i \in I$ for short. So $\langle \mathbb{L}, \preccurlyeq \rangle \triangleq \langle \{\blacksquare\}, \Longrightarrow \rangle$ where $\blacksquare$ stands for the void syntactic component, $\langle \mathcal{D}_\blacksquare, \sqsubseteq_\blacksquare, \perp_\blacksquare, \sqcup_\blacksquare \rangle \triangleq \langle \wp(\mathcal{U}), \subseteq, \varnothing, \cup \rangle$, $\Delta_\blacksquare \triangleq I$, for a given $i \in I$, $F_\blacksquare^i \in \wp(U) \longrightarrow \wp(U)$ is $F_\blacksquare^i(X) \triangleq \{c_i \mid P_i \subseteq X\}$ and $\bigcurlyvee_\blacksquare \triangleq \bigcup$ thus defining $\mathcal{S} = \mathbf{lfp}^\subseteq \boldsymbol{\lambda} X \cdot \{c_i \mid i \in I \wedge P_i \subseteq X\}$.

---

[5] In the classical case [2], the fixpoint operator is continuous hence proofs are finite.

2.13 Reduction of order-theoretic inductive definitions

An element x of a poset $\langle \mathcal{D}, \sqsubseteq \rangle$ is *(complete) join irreducible* if and only if for all $X \subseteq \mathcal{D}$ such that $x = \bigsqcup X$ we necessarily have $x \in X$. Observe that if $\langle \mathcal{D}, \sqsubseteq \rangle$ has an infimum $\bot$ then $\bot$ is not join irreducible since $\bot = \bigsqcup \varnothing$ but $\bot \notin \varnothing$. We let $\mathcal{J}(\mathcal{D})$ be the set of join irreducibles of $\mathcal{D}$. If $x \in \mathcal{D}$, we define $\mathcal{J}(x) \triangleq \{ y \in \mathcal{J}(\mathcal{D}) \mid y \sqsubseteq x \}$.

A poset $\langle \mathcal{D}, \sqsubseteq \rangle$ satisfies the *descending chain condition (DCC)* if and only if every denumerable descending chain $x_0 \sqsupseteq x_1 \sqsupseteq \ldots$ in $\mathcal{D}$ is finite that is $x_k = x_{k+1} = \ldots$ for some $k \in \mathbb{N}$.

If $\langle \mathcal{D}, \sqsubseteq \rangle$ is a poset satisfying (DCC) then for all $\forall x \in \mathcal{D} : x = \bigsqcup \mathcal{J}(x)$. The proof is an easy generalization of [12, **Prop. 2.45**].

In case $\langle \mathcal{D}_\ell, \sqsubseteq_\ell \rangle$ satisfies (DCC), we let $\{ F_\ell^{ij}(P_i, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!]) \mid j \in \Delta_\ell^i \}$ $\triangleq \mathcal{J}(F_\ell^i(P_i, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!]))$ for all $i \in \Delta_\ell$ and define the *reduced inductive definition* as

$$\forall \ell \in \mathbb{L} : \mathcal{S}_f^{\mathcal{J}}[\![\ell]\!] = \mathbf{lfp}^{\sqsubseteq_\ell} \, \boldsymbol{\lambda} X \bullet \bigvee_{\substack{\ell \\ i \in \Delta_\ell}} \bigsqcup_{\substack{\ell \\ j \in \Delta_\ell^i}} F_\ell^{ij}(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!]) \, .$$

and similarly for the equivalent forms.

Lemma 7 *If* $\langle \mathcal{D}_\ell, \sqsubseteq_\ell \rangle$, $\ell \in \mathbb{L}$ *satisfies (DCC) then* $\mathcal{S}_f^{\mathcal{J}}[\![\ell]\!] = \mathcal{S}_f[\![\ell]\!]$. $\qquad \square$

PROOF By join irreducibility $F_\ell^i(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!]) = \bigsqcup_\ell \mathcal{J}(F_\ell^i(X, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!]))$ $= \bigsqcup_{\substack{\ell \\ j \in \Delta_\ell^i}} F_\ell^{ij}(P_i, \prod_{\ell' \prec\!\!\prec \ell} \mathcal{S}_f[\![\ell']\!])$ by definition. $\qquad \blacksquare$

For example $0 \in \mathcal{E}, \; \dfrac{X \subseteq \mathcal{E}}{\{ n \mid n+2 \in X \} \subseteq \mathcal{E}} \subseteq$ can be simplified into $0 \in \mathcal{E}, \; \dfrac{n \in \mathcal{E}}{n+2 \in \mathcal{E}} \subseteq$.

2.14 Bi-semantic domains

To account for terminating/finite and diverging/infinite program behaviors, we consider bi-semantic domains [10] consisting, for each $\ell \in \mathbb{L}$, of a finitary semantic domain (of finite program behaviors) $\langle \mathcal{D}_\ell^+, \sqsubseteq_\ell^+, \bot_\ell^+, \bigsqcup_\ell^+ \rangle$ and a infinitary semantic codomain (of infinite program behaviors) $\langle \mathcal{D}_\ell^-, \sqsubseteq_\ell^-, \bot_\ell^-, \bigsqcup_\ell^- \rangle$ which are assumed to be dcpos (respectively complete lattices). They are combined into a bi-semantic domain (of bifinite program behaviors) $\mathcal{D}_\ell$ thanks to a projection $\pi_\ell^+ \in \mathcal{D}_\ell \longrightarrow \mathcal{D}_\ell^+$, a coprojection $\pi_\ell^- \in \mathcal{D}_\ell \longrightarrow \mathcal{D}_\ell^-$, and a constructor $\pi_\ell \in \mathcal{D}_\ell^+ \times \mathcal{D}_\ell^- \longrightarrow \mathcal{D}_\ell$ satisfying $\forall x \in \mathcal{D}_\ell^+, y \in \mathcal{D}_\ell^- : \pi_\ell^+(\pi_\ell(x,y)) = x$ and $\pi_\ell^-(\pi_\ell(x,y)) = y$ while $\forall X \in \mathcal{D} : \pi_\ell(\pi_\ell^+(X), \pi_\ell^-(X)) = X$. Examples are

the cartesian product, disjoint union or union of disjoint sets. The bi-semantic domain $\langle \mathcal{D}_\ell, \sqsubseteq_\ell, \perp_\ell, \sqcup_\ell \rangle$ is then a dcpo (respectively a complete lattice) by defining $X^+ \triangleq \pi_\ell^+(X)$, $X^- \triangleq \pi_\ell^-(X)$, $X \sqsubseteq_\ell Y \triangleq (X^+ \sqsubseteq_\ell^+ Y^+) \wedge (X^- \sqsubseteq_\ell^- Y^-)$, and $\bigsqcup_\ell \underset{i \in I}{} X_i \triangleq \pi_\ell(\bigsqcup_\ell^+ \underset{i \in I}{} X_i^+, \bigsqcup_\ell^- \underset{i \in I}{} X_i^-)$.

2.15 Bi-semantic fixpoints

Lemma 8 *Let L^+ and L^- be a partition of the set L. For all $X, Y \subseteq L$, define $X^+ \triangleq X \cap L^+$, $X^- \triangleq X \cap L^-$, and $(X \sqsubseteq Y) \triangleq (X^+ \subseteq Y^+) \wedge (X^- \supseteq Y^-)$. Let $F \in \wp(L) \longrightarrow \wp(L)$ be $\subseteq$-monotone* [6] *such that $\forall X \subseteq L : (F(X))^+ = F(X^+)$ and $F(X) \subseteq F((\mathbb{S}^+ \cap X^+) \cup X^-)$ where $F^+(X) \triangleq (F(X^+))^+$, $\mathbb{S}^+ = \mathbf{lfp}^{\subseteq} F^+$, $F^-(X) \triangleq (F(\mathbb{S}^+ \cup X^-))^-$, $\mathbb{S}^- = \mathbf{gfp}^{\subseteq} F^-$. Then $\mathbb{S} \triangleq \mathbb{S}^+ \cup \mathbb{S}^- = \mathbf{lfp}^{\sqsubseteq} F$.* □

PROOF $\langle \wp(L), \subseteq \rangle$ is a complete lattice and F is $\subseteq$-monotone when so are F^+ and F^- proving that $\mathbf{lfp}^{\subseteq} F^+$ and $\mathbf{gfp}^{\subseteq} F^-$ exist by Tarski's fixpoint theorem [33]. We first prove that $\mathbb{S}$ is a fixpoint of F.

$$
\begin{aligned}
& \mathbb{S} \\
=\ & \mathbb{S}^+ \cup \mathbb{S}^- \\
=\ & F^+(\mathbb{S}^+) \cup F^-(\mathbb{S}^-) \\
& \qquad \qquad \langle \text{by fixpoint definitions } \mathbb{S}^+ \triangleq \mathbf{lfp}^{\subseteq} F^+ \text{ and } \mathbb{S}^- \triangleq \mathbf{gfp}^{\subseteq} F^- \rangle \\
=\ & (F(\mathbb{S}^+))^+ \cup (F(\mathbb{S}^+ \cup \mathbb{S}^-))^- \qquad\qquad\qquad\quad \langle \text{def. } F^+ \text{ and } F^- \rangle \\
=\ & (F(\mathbb{S}))^+ \cup (F(\mathbb{S}))^- \quad \langle \text{since } (F(S^+))^+ = (F(S))^+ \text{ and } \mathbb{S} = \mathbb{S}^+ \cup \mathbb{S}^- \rangle \\
=\ & F(\mathbb{S}) \qquad\qquad\qquad\qquad\qquad \langle \text{since } \forall X \subseteq L : X = X^+ \cup X^- \rangle
\end{aligned}
$$

To prove that $\mathbb{S}$ is the $\sqsubseteq$-least fixpoint of F, let T be another fixpoint of F that is $T = F(T)$. It follows that $T^+ \cup T^- = (F(T))^+ \cup (F(T))^-$ so $T^+ = (F(T))^+$ and $T^- = (F(T))^-$ since $L^+ \cap L^- = \varnothing$. Therefore $T^+ = (F(T))^+ = (F(T^+))^+ = F^+(T^+)$ hence $\mathbb{S}^+ \subseteq T^+$ since $\mathbb{S}^+ \triangleq \mathbf{lfp}^{\subseteq} F^+$. Moreover $T^- = (F(T))^- = (F(T^+ \cup T^-))^- = (F(\mathbb{S}^+ \cup T^-))^- = F^-(T^-)$ by $\subseteq$-monotony of F, hypothesis $F(T) \subseteq F((\mathbb{S}^+ \cap T^+) \cup T^-)$ and antisymmetry. It follows that $T^- \subseteq \mathbb{S}^-$ by Tarski's fixpoint theorem [33] for $\mathbf{gfp}^{\subseteq} F^-$. We conclude that $\mathbb{S} \sqsubseteq T$ by def. of $\sqsubseteq$. ∎

The lemma can be easily generalized to any bi-semantic domain as defined in the previous Sect. 2.14.

---

[6] but not necessarily $\sqsubseteq$-monotone.

Given a set $\mathcal{S}$ (for example, a set of states in Sect. 4, a finite terminal alphabet in Sect. 5 or a set of terms in Sect. 6), we let $\mathcal{S}^\star$ be the set of finite sequences over the set $\mathcal{S}$ including the empty sequence ϵ, $\mathcal{S}^+ \triangleq \mathcal{S}^\star \setminus \{\epsilon\}$, $\mathcal{S}^\omega$ be the set of infinite sequences over $\mathcal{S}$, $\mathcal{S}^\propto \triangleq \mathcal{S}^\star \cup \mathcal{S}^\omega$ be the set of finite or infinite sequences over $\mathcal{S}$ [7], and $\mathcal{S}^\infty \triangleq \mathcal{S}^+ \cup \mathcal{S}^\omega$ be the set of nonempty finite or infinite sequences over $\mathcal{S}$. We let $|\sigma| \in \mathbb{N} \cup \{\omega\}$ be the length of $\sigma \in \mathcal{S}^\propto$, in particular $|\epsilon| = 0$ and $\mathcal{S}^n \triangleq \{\sigma \in \mathcal{S}^\star \mid |\sigma| = n\}$. We let $\bullet$ be the concatenation of traces so that $\epsilon \bullet \sigma = \sigma \bullet \epsilon = \sigma$ and $\sigma \bullet \varsigma = \sigma$ when $\sigma \in \mathcal{S}^\omega$. If $\sigma \in \mathcal{S}^+$ then $|\sigma| > 0$ and $\sigma = \sigma_0 \bullet \sigma_1 \bullet \ldots \bullet \sigma_{|\sigma|-1}$. If $\sigma \in \mathcal{S}^\omega$ then $|\sigma| = \omega$ and $\sigma = \sigma_0 \bullet \sigma_1 \bullet \ldots \bullet \sigma_n \bullet \ldots$. For sentences over an alphabet in Sect. 5, we denote concatenation $\bullet$ by juxtaposition so $\sigma = \sigma_0 \sigma_1 \ldots \sigma_{|\sigma|-1} \in \mathcal{S}^\star$ and $\sigma = \sigma_0 \sigma_1 \ldots \sigma_n \ldots \in \mathcal{S}^\omega$.

Given $X, Y \in \wp(\mathcal{S}^\propto)$, we define $X^\star \triangleq X \cap \mathcal{S}^\star$, $X^+ \triangleq X \cap \mathcal{S}^+$, $X^\omega \triangleq X \cap \mathcal{S}^\omega$ and $X \sqsubseteq Y \triangleq X^\star \subseteq Y^\star \wedge X^\omega \supseteq Y^\omega$, so that $\langle \wp(\mathcal{S}^\propto), \sqsubseteq, \mathcal{S}^\omega, \mathcal{S}^\star, \sqcup, \sqcap \rangle$ is an example of bi-semantic domain as defined in Sect. 2.14. It is a complete lattice with lub $X \sqcup Y \triangleq (X^\star \cup Y^\star)) \cup (X^\omega \cap Y^\omega)$. Similarly, for the bi-semantic domain $\langle \wp(\mathcal{S}^\infty), \sqsubseteq, \mathcal{S}^\omega, \mathcal{S}^+, \sqcup, \sqcap \rangle$.

3 Abstraction

We consider a simple form of abstraction based on a continuous abstraction function α [9], which includes the particular case of a Galois connection [8] (denoted $\langle P, \preceq \rangle \xleftarrow[\alpha]{\gamma} \langle Q, \sqsubseteq \rangle$, or $\langle P, \preceq \rangle \xleftarrow[\alpha]{\gamma} \langle Q, \sqsubseteq \rangle$ when α is onto, where $\langle P, \preceq \rangle$ and $\langle Q, \sqsubseteq \rangle$ are posets, and $\forall x \in P : \forall y \in Q : \alpha(x) \sqsubseteq y \Longleftrightarrow x \preceq \gamma(y)$).

For all $\ell \in \mathbb{L}$, we let $\langle \overline{\mathcal{D}}_\ell, \overline{\sqsubseteq}_\ell, \overline{\perp}_\ell, \overline{\sqcup}_\ell \rangle$ be dcpos, $\overline{F}_\ell^i \in \overline{\mathcal{D}}_\ell \times \overline{\mathcal{D}}_{\ell_1} \ldots \times \overline{\mathcal{D}}_{\ell_n} \longrightarrow \overline{\mathcal{D}}_\ell$, $i \in \Delta_\ell$ be monotone in their first parameter, and define the abstract semantics $\overline{\mathcal{S}}_f[\![\ell]\!]$ in one of the equivalent forms of **Th. 6**.

If $\alpha_\ell \in \mathcal{D}_\ell \longrightarrow \overline{\mathcal{D}}_\ell$, we say that the abstract semantics $\langle \overline{\mathcal{S}}[\![\ell]\!], \ell \in \mathbb{L} \rangle$ is *sound* with respect to the concrete semantics $\langle \mathcal{S}[\![\ell]\!], \ell \in \mathbb{L} \rangle$ if and only if $\forall \ell \in \mathbb{L} : \alpha_\ell(\mathcal{S}[\![\ell]\!]) \overline{\sqsubseteq}_\ell \overline{\mathcal{S}}[\![\ell]\!]$. It is *complete* whenever $\forall \ell \in \mathbb{L} : \overline{\mathcal{S}}[\![\ell]\!] \overline{\sqsubseteq}_\ell \alpha_\ell(\mathcal{S}[\![\ell]\!])$. The following theorem provides a sufficient soundness and completeness condition.

Theorem 9 *If the F_ℓ^i and $\overline{F}_\ell^i$ are monotone in their first parameter, the join operators $\curlyvee_\ell$ and $\overline{\curlyvee}_\ell$ are componentwise monotone, the $\alpha_\ell \in \mathcal{D}_\ell \longrightarrow \overline{\mathcal{D}}_\ell$, $\ell \in \mathbb{L}$ are strict and continuous (in particular $\langle \mathcal{D}_\ell, \sqsubseteq_\ell \rangle \xleftarrow[\alpha_\ell]{\gamma_\ell} \langle \overline{\mathcal{D}}_\ell, \overline{\sqsubseteq}_\ell \rangle$ is a Galois*

---

[7] The "proportional to" symbol $\propto$ is used as a pictogram similar to "infinity" ∞ but with the possibility of emptiness.

connection) and the F_ℓ^i commute with the $\overline{F}_\ell^i$ up to α_ℓ i.e. $\forall \ell \in \mathbb{L} : \forall X_\ell \sqsubseteq_\ell$
$\mathcal{S}[\![\ell]\!] : \forall X_{\ell'} \in \mathcal{D}_{\ell'}, \ell' \prec \ell$:

$$\overline{\curlyvee_\ell}_{i \in \Delta_\ell} \overline{F}_\ell^i(\alpha_\ell(X_\ell), \prod_{\ell' \prec \ell} \alpha_{\ell'}(X_{\ell'})) = \alpha_\ell(\curlyvee_\ell_{i \in \Delta_\ell} F_\ell^i(X_\ell, \prod_{\ell' \prec \ell} X_{\ell'}))$$

then $\overline{\mathcal{S}}[\![\ell]\!] = \alpha_\ell(\mathcal{S}[\![\ell]\!])$. $\square$

PROOF By induction on $\prec$ using the fixpoint definition of $\mathcal{S}[\![\ell]\!]$ and $\overline{\mathcal{S}}[\![\ell]\!]$, and [8, 7.1.0.4-(3)]. ∎

4 Order-theoretic inductive definitions of the trace semantics of transition systems

We let Σ be a set of states and $\tau \subseteq \Sigma \times \Sigma$ be a transition relation on Σ. We consider the bi-semantic domain $\langle \wp(\Sigma^\infty), \sqsubseteq, \Sigma^\omega, \Sigma^+, \sqcup, \sqcap \rangle$ defined in Sect. 2.16 and the trace semantics

$$\mathcal{S} \triangleq \bigcup_{n>0} \{\sigma \in \Sigma^n \mid \forall i \in [0, n-1[: \langle \sigma_i, \sigma_{i+1} \rangle \in \tau \wedge \forall s \in \Sigma : \langle \sigma_{n-1}, s \rangle \notin \tau\}$$
$$\cup \{\sigma \in \Sigma^\omega \mid \forall i \in \mathbb{N} : \langle \sigma_i, \sigma_{i+1} \rangle \in \tau\} \ .$$

Since the semantics is not defined by structural induction, we define $\mathbb{L} \triangleq \{\blacksquare\}$ where $\blacksquare$ is a void syntactic component and $\prec \triangleq \varnothing$. We let

$$\dot{\tau} \triangleq \{\sigma \in \Sigma^+ \mid |\sigma| = 1 \wedge \forall s \in \Sigma : \langle \sigma_0, s \rangle \notin \tau\} \qquad \text{blocking state traces}$$
$$\tau \circ X \triangleq \{\sigma_0 \bullet \sigma_1 \bullet \varsigma \in \Sigma^\infty \mid \langle \sigma_0, \sigma_1 \rangle \in \tau \wedge \sigma_1 \bullet \varsigma \in X\} \quad \text{transition prefix}$$
$$F(X) \triangleq \dot{\tau} \cup \tau \circ X \qquad \qquad \text{trace transformer.}$$

The trace transformer F is $\sqsubseteq$-monotone, indeed upper-continuous. The i-th iterate F^i of F from Σ^ω is

$$F^i = \bigcup_{n=0}^{i-1} \{\sigma \in \Sigma^n \mid \forall i \in [0, n-1[: \langle \sigma_i, \sigma_{i+1} \rangle \in \tau \wedge \forall s \in \Sigma : \langle \sigma_{n-1}, s \rangle \notin \tau\}$$
$$\cup \{\sigma_0 \bullet \ldots \bullet \sigma_i \ldots \bullet \varsigma \in \Sigma^\omega \mid \forall k \in [0, i-1[: \langle \sigma_k, \sigma_{k+1} \rangle \in \tau\}$$

so that $\mathcal{S} = \bigsqcup_{i \in \mathbb{N}} F^i = \mathbf{lfp}^\sqsubseteq F$ [6]. In rule-based form, we have

$$\dot{\tau} \in \mathcal{S} \qquad \frac{\sigma \in \mathcal{S}}{\tau \circ \sigma \in \mathcal{S}} \sqsubseteq .$$

The trace transformer F is $\sqsubseteq$-monotone for transition systems and for grammars considered in next Sect. 5 but no longer for the big-step trace semantics of the call-by-value λ-calculus considered in Sect. 6.3.

5 Structural order-theoretic inductive definitions of the semantics of context-free grammars

The Ginsburg-Rice/Chomsky-Schützenberger theorem [4,14,31] shows that the terminal language generated by a context-free grammar can be expressed in $\dot{\subseteq}$-least fixpoint form. This was extended to the infinite language generated by a context-free grammar by Nivat [26] using $\dot{\subseteq}$-greatest fixpoints. To illustrate bi-inductive structural definition on a simple example, we define the bifinite semantics of grammars mixing the least fixpoint for finite sentences and the greatest fixpoint for infinite sentences.

5.1 Metasyntax of grammars

The (meta-)language $\mathbb{L} \triangleq \{\epsilon\} \cup \mathbb{T} \cup \mathbb{N} \cup \mathbb{R} \cup \mathbb{P} \cup \mathbb{S} \cup \mathbb{G}$ is defined by the following (meta-)grammar $(\mathbb{T} \cap \mathbb{N} = \varnothing)$

$\epsilon \notin \mathbb{T} \cup \mathbb{N}$	empty sentence		$P \in \mathbb{P}$	productions	
$T \in \mathbb{T}$	terminals		$P ::= N \rightarrow R$		
$N \in \mathbb{N}$	nonterminals[8]		$S \in \mathbb{S}$	sets of productions	
$R \in \mathbb{R}$	righthand sides		$S ::= P \mid P\,S$		
$R ::= T\,R \mid N\,R \mid \epsilon$			$G \in \mathbb{G}$	grammars	
			$G ::= S$		

As usual $N ::= \alpha \mid \beta$ is a shorthand for the two grammar rules $N ::= \alpha$ and $N ::= \beta$. To avoid confusion, the left-hand side N of a grammar rule is separated from the right-hand side α by $::=$ in the meta-grammar ($N ::= \alpha$) and by

---

[8] $\mathbb{N}$ is the set of nonterminals while $\mathbb{N}$ is the set of natural numbers.

$\rightarrow$ in the grammar ($N \rightarrow \alpha$). The "strict immediate subcomponent" relation $\prec$ on the meta-language $\mathbb{L}$ is defined as $T \prec T R$, $R \prec T R$, $N \prec N R$, $R \prec N R$, $\epsilon \prec R$ (when $R ::= \epsilon$), $R \prec N \rightarrow R$, $P \prec S$ (when $S ::= P$), $P \prec P S$, $S \prec P S$, and $S \prec G$ (when $G ::= S$). Hence $\prec$ is well-founded since sentences in the meta-language (that is grammars) are assumed to be finite [9].

5.2 Fixpoint structural bifinite semantics of grammars

The bifinite semantics $\mathcal{S}[\![G]\!] \in \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)$ of grammars G is defined in fixpoint form by structural induction (on $\prec$). Recall from Sect. 2.16 that $\langle \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty), \dot{\sqsubseteq}, \boldsymbol{\lambda} N \cdot T^\omega, \boldsymbol{\lambda} N \cdot T^\star, \dot{\sqcup}, \dot{\sqcap} \rangle$ is a complete lattice for the pointwise ordering $\dot{\sqsubseteq}$.

$$\mathcal{S}[\![\epsilon]\!] \in \{\epsilon\} \qquad \mathcal{S}[\![T]\!] \in \mathbb{T} \qquad \mathcal{S}[\![N]\!] \in \mathbb{N}$$
$$\mathcal{S}[\![\epsilon]\!] \triangleq \epsilon \qquad \mathcal{S}[\![T]\!] \triangleq T \qquad \mathcal{S}[\![N]\!] \triangleq N$$

$$\mathcal{S}[\![R]\!] \in (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)) \longrightarrow \wp(\mathbb{T}^\infty)$$
$$\mathcal{S}[\![R ::= T\ R']\!] \triangleq (\boldsymbol{\lambda} \rho \cdot \{\mathcal{S}[\![T]\!]\}) \mathbin{\dot{\cdot}} \mathcal{S}[\![R']\!]$$
$$\mathcal{S}[\![R ::= N\ R']\!] \triangleq (\boldsymbol{\lambda} \rho \cdot \rho(\mathcal{S}[\![N]\!])) \mathbin{\dot{\cdot}} \mathcal{S}[\![R']\!]$$
$$\mathcal{S}[\![R ::= \epsilon]\!] \triangleq \boldsymbol{\lambda} \rho \cdot \{\mathcal{S}[\![\epsilon]\!]\}$$

$$\mathcal{S}[\![P]\!] \in (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)) \longrightarrow (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty))$$
$$\mathcal{S}[\![P ::= N \rightarrow R]\!] \triangleq \boldsymbol{\lambda} \rho \cdot \boldsymbol{\lambda} N' \cdot (\!|N' = N \mathbin{?} \mathcal{S}[\![R]\!]\rho \mathbin{\vdots} \mathbb{T}^\omega|\!)$$

$$\mathcal{S}[\![S]\!] \in (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)) \longrightarrow (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty))$$
$$S[\![S ::= P]\!] \triangleq \mathcal{S}[\![P]\!]$$
$$\mathcal{S}[\![S ::= P\ S']\!] \triangleq \mathcal{S}[\![P]\!] \mathbin{\dot{\cup}} \mathcal{S}[\![S']\!]$$

$$\mathcal{S}[\![G]\!] \in \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)$$
$$\mathcal{S}[\![G ::= S]\!] \triangleq \mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![S]\!] \quad .$$

where $\dot{\cdot}$ is sentence concatenation $\cdot$ extended elementwise and pointwise; $\dot{\cup}$ is the pointwise extension of $\dot{\cup}$, itself the pointwise extension of $\cup$; and $(\!|\mathrm{tt} \mathbin{?} a \mathbin{\vdots} b|\!) = a$, $(\!|\mathrm{ff} \mathbin{?} a \mathbin{\vdots} b|\!) = b$ is the conditional.

[9] Observe that for a meta-grammar rule $A ::= B_1 \ldots B_n$ where the nonterminals B_1, $\ldots$, B_n respectively derive into the terminal sentences β_1, $\ldots$, β_n so that A derives into $\alpha = \beta_1 \ldots \beta_n$, we have $\beta_1 \prec \alpha$, $\ldots$, and $\beta_n \prec \alpha$ and we write, as defined in Sect. 2.2, $\alpha ::= \beta_1 \ldots \beta_n$. In the metagrammar, we use the same symbols for A, $B_1 \ldots$, B_n and α, $\beta_1 \ldots$, β_n!

Theorem 10 $\forall \mathsf{G} \in \mathbb{G} : \mathcal{S}[\![\mathsf{G}]\!]$ *is well-defined.* □

PROOF If $L, L_1, L_2 \subseteq \mathbb{T}^\infty$ and $L_1 \sqsubseteq L_2$ then $L_1^\star \subseteq L_2^\star$ and $L_1^\omega \supseteq L_2^\omega$ so $L_1 \bullet L = L_1^\star \bullet L \cup L_1^\omega \sqsubseteq L_2^\star \bullet L \cup L_2^\omega = L_2 \bullet L$ since $L_1^\star \bullet L \subseteq L_2^\star \bullet L$ and $L_1^\omega \supseteq L_2^\omega$. Moreover $L \bullet L_1 = L \bullet L_1 \cup L^\omega = L \bullet L_1^\star \cup L \bullet L_1^\omega \cup L^\omega \sqsubseteq L \bullet L_2^\star \cup L \bullet L_2^\omega \cup L^\omega = L \bullet L_2 \cup L^\omega = L \bullet L_2$ since $L \bullet L_1^\star \subseteq L \bullet L_2^\star$ and $L \bullet L_1^\omega \cup L^\omega \supseteq L \bullet L_2^\omega \cup L^\omega$. Therefore $\bullet$ is $\sqsubseteq$-monotone hence so is $\mathbin{\raisebox{0.2ex}{$\bullet$}}$ pointwise. It follows, by induction on the "strict immediate subcomponent" relation $\prec$, that $\forall \mathsf{S} \in \mathbb{S} : \mathcal{S}[\![\mathsf{S}]\!]$ is $\dot{\sqsubseteq}$-monotone so $\mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![\mathsf{S}]\!]$ exists and $\forall \mathsf{G} \in \mathbb{G} : \mathcal{S}[\![\mathsf{G}]\!]$ is well-defined. ∎

Example 11 For the grammar G defined by the rules $\mathsf{X} \to \mathsf{aX}$, $\mathsf{X} \to \mathsf{b}$, we have the following metasyntax tree whose nodes are decorated with their semantics

The semantics of the grammar G is therefore

$$\mathcal{S}[\![\mathsf{G}]\!] \triangleq \mathbf{lfp}^{\dot{\sqsubseteq}} \boldsymbol{\lambda}\rho \bullet \boldsymbol{\lambda}\mathsf{N}' \bullet \big(\!\big|\, \mathsf{N}' = \mathsf{X} \,\mathbin{\raisebox{0.3ex}{?}}\, (\{\mathsf{a}\} \bullet \rho(\mathsf{X})) \cup \{\mathsf{b}\} \,\mathbin{\raisebox{0.3ex}{$\mathring{,}$}}\, \mathbb{T}^\omega \,\big|\!\big)$$

$$= \boldsymbol{\lambda}\mathsf{N}' \bullet \big(\!\big|\, \mathsf{N}' = \mathsf{X} \,\mathbin{\raisebox{0.3ex}{?}}\, \{\mathsf{a}^\omega\} \cup \{\mathsf{a}^n\mathsf{b} \mid n \geqslant 0\} \,\mathbin{\raisebox{0.3ex}{$\mathring{,}$}}\, \mathbb{T}^\omega \,\big|\!\big)$$

which is the $\dot{\sqsubseteq}$-least solution of the more traditional system of equations [4,14,26,31] (where $X \triangleq \rho(\mathsf{X})$ and $N \triangleq \rho(\mathsf{N})$)

$$\begin{cases} X = (\{\mathsf{a}\} \bullet X) \cup \{\mathsf{b}\} \\ N = \mathbb{T}^\omega \qquad\qquad \text{when } \mathsf{N} \neq \mathsf{X} \;. \end{cases}$$

The $\dot{\sqsubseteq}$-least solution for the X component is computed iteratively as

$$X^0 \;=\; \mathbb{T}^\omega$$

$$X^1 \;=\; (\{a\} \bullet X^0) \cup \{b\}$$

$$\;=\; (\{a\} \bullet \mathbb{T}^\omega) \cup \{b\}$$

$$X^2 \;=\; (\{a\} \bullet X^1) \cup \{b\}$$

$$\;=\; (\{a\} \bullet ((\{a\} \bullet \mathbb{T}^\omega) \cup \{b\})) \cup \{b\}$$

$$\;=\; (\{aa\} \bullet \mathbb{T}^\omega) \cup \{ab, b\}$$

$$\ldots \ldots \ldots$$

$$X^n \;=\; (\{a^n\} \bullet \mathbb{T}^\omega) \cup \bigcup_{0 \leqslant i < n} \{a^i b\} \qquad\qquad \text{induction hypothesis}$$

$$X^{n+1} \;=\; (\{a\} \bullet X^n) \cup \{b\}$$

$$\;=\; (\{a\} \bullet ((\{a^n\} \bullet \mathbb{T}^\omega) \cup \bigcup_{0 \leqslant i < n} \{a^i b\})) \cup \{b\}$$

$$\;=\; \{a^{n+1}\} \bullet \mathbb{T}^\omega \cup \bigcup_{0 \leqslant i < n} \{a^{i+1} b\} \cup \{b\}$$

$$\;=\; \{a^{n+1}\} \bullet \mathbb{T}^\omega \cup \bigcup_{0 \leqslant j < n+1} \{a^j b\} \qquad\qquad \text{where } j = i + 1$$

$$\ldots \ldots \ldots$$

$$X^\omega \;=\; \bigcap_{n \geqslant 0} (\{a^n\} \bullet \mathbb{T}^\omega) \cup \bigcup_{n \geqslant 0} (\bigcup_{0 \leqslant i < n} \{a^i b\})$$

$$\;=\; \{a^\omega\} \cup \bigcup_{n \geqslant 0} \{a^n b\}$$

$$X^{\omega+1} \;=\; X^\omega \; . \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \square$$

5.3 Rule-based structural bifinite semantics of grammars

An equivalent definition of the bifinite semantics $\mathcal{S}[\![\mathsf{G}]\!] \in \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)$ of grammars G can be given in rule-based form by structural induction (on $\longrightarrow\!\!\!\prec$) as follows ($\rho \in \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)$)

$$\epsilon \in \mathcal{S}[\![\epsilon]\!] \qquad\qquad\qquad\qquad \mathsf{T} \in \mathcal{S}[\![\mathsf{T}]\!]$$

$$\mathsf{N} \in \mathcal{S}[\![\mathsf{N}]\!] \qquad\qquad (\boldsymbol{\lambda}\rho \bullet \{\mathcal{S}[\![\mathsf{T}]\!]\}) \,\dot{;}\, \mathcal{S}[\![\mathsf{R}']\!] \,\dot{\sqsubseteq}\, \mathcal{S}[\![\mathsf{R} ::= \mathsf{T}\ \mathsf{R}']\!]$$

$$(\boldsymbol{\lambda}\rho \bullet \rho(\mathcal{S}[\![\mathsf{N}]\!])) \,\dot{;}\, \mathcal{S}[\![\mathsf{R}']\!] \,\dot{\sqsubseteq}\, \mathcal{S}[\![\mathsf{R} ::= \mathsf{N}\ \mathsf{R}']\!] \qquad \boldsymbol{\lambda}\rho \bullet \{\mathcal{S}[\![\epsilon]\!]\} \,\dot{\sqsubseteq}\, \mathcal{S}[\![\mathsf{R} ::= \epsilon]\!]$$

$$\lambda\rho \cdot \lambda\,\mathsf{N}' \cdot \left(\mathsf{N}' = \mathsf{N}\ \mathring{?}\ \mathcal{S}[\![\mathsf{R}]\!](\rho)\ \mathring{\mathstrut}\ \mathbb{T}^\omega\right) \dot{\sqsubseteq}$$
$$\mathcal{S}[\![\mathsf{P} ::= \mathsf{N}\ \rightarrow\ \mathsf{R}]\!] \qquad\qquad \mathcal{S}[\![\mathsf{P}]\!] \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{S} ::= \mathsf{P}]\!]$$

$$\mathcal{S}[\![\mathsf{P}]\!] \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{S} ::= \mathsf{P}\ \mathsf{S}']\!] \qquad\qquad \mathcal{S}[\![\mathsf{S}']\!] \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{S} ::= \mathsf{P}\ \mathsf{S}']\!]$$

$$\dfrac{\rho \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{G} ::= \mathsf{S}]\!]}{\mathcal{S}[\![\mathsf{S}]\!](\rho) \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{G} ::= \mathsf{S}]\!]}\ {\dot{\sqsubseteq}}\ .$$

Example 12 The bi-inductive definition of the semantics $\mathcal{S}[\![\mathsf{G}]\!]$ of the grammar G defined by the rules $\mathsf{X}\ \rightarrow\ \mathsf{aX}, \mathsf{X}\ \rightarrow\ \mathsf{b}$ is

$$\dfrac{\rho \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{G}]\!]}{\lambda\,\mathsf{N}' \cdot \left(\mathsf{N}' = \mathsf{X}\ \mathring{?}\ (\{\mathsf{a}\} \bullet \rho(\mathsf{X})) \cup \{\mathsf{b}\}\ \mathring{\mathstrut}\ \mathbb{T}^\omega\right) \dot{\sqsubseteq} \mathcal{S}[\![\mathsf{G}]\!]}\ {\dot{\sqsubseteq}}$$

which, letting $X \triangleq \rho(\mathsf{X})$, simplifies into

$$\mathbb{T}^\omega \sqsubseteq \mathcal{S}[\![\mathsf{G}]\!]\mathsf{N}, \quad \mathsf{N} \neq \mathsf{X} \qquad \dfrac{X \sqsubseteq \mathcal{S}[\![\mathsf{G}]\!]\mathsf{X}}{(\{\mathsf{a}\} \bullet X) \cup \{\mathsf{b}\} \sqsubseteq \mathcal{S}[\![\mathsf{G}]\!]\mathsf{X}}\ {\sqsubseteq}\ .$$

The proof that the finite word $\mathsf{a}^n\mathsf{b}$ is generated by G is (each theorem is followed by a proof argument given between curly brackets $\wr...\wr$)

$\mathbb{T}^\omega \qquad \wr$ basis $\wr$

$\{\mathsf{b}\} \qquad \wr \{\mathsf{b}\} \sqsubseteq (\{\mathsf{a}\} \bullet \mathbb{T}^\omega) \cup \{\mathsf{b}\}\wr$

$\{\mathsf{ab}\} \qquad \wr \{\mathsf{ab}\} \sqsubseteq (\{\mathsf{a}\} \bullet \{\mathsf{b}\}) \cup \{\mathsf{b}\} \sqsubseteq (\{\mathsf{a}\} \bullet (\mathbb{T}^\omega \sqcup \{\mathsf{b}\})) \cup \{\mathsf{b}\}\wr$

$\{\mathsf{a}^2\mathsf{b}\} \qquad \wr \{\mathsf{a}^2\mathsf{b}\} \sqsubseteq (\{\mathsf{a}\} \bullet \{\mathsf{ab}\}) \cup \{\mathsf{b}\} \sqsubseteq (\{\mathsf{a}\} \bullet (\mathbb{T}^\omega \sqcup \{\mathsf{b}\} \sqcup \{\mathsf{ab}\})) \cup \{\mathsf{b}\}\wr$

$\cdots$

$\{\mathsf{a}^n\mathsf{b}\} \qquad \wr \{\mathsf{a}^n\mathsf{b}\} \sqsubseteq (\{\mathsf{a}\} \bullet \{\mathsf{a}^{n-1}\mathsf{b}\}) \cup \{\mathsf{b}\}\wr$.

The transfinite proof that the infinite word a^ω is generated by G is

$\mathbb{T}^\omega \qquad\qquad \wr$ basis $\wr$

$\{\mathsf{a}\} \bullet \mathbb{T}^\omega \qquad \wr \{\mathsf{a}\} \bullet \mathbb{T}^\omega \sqsubseteq (\{\mathsf{a}\} \bullet \mathbb{T}^\omega) \cup \{\mathsf{b}\}\wr$

$\{\mathsf{a}^2\} \bullet \mathbb{T}^\omega \qquad \wr \{\mathsf{a}^2\} \bullet \mathbb{T}^\omega \sqsubseteq (\{\mathsf{a}\} \bullet \{\mathsf{a}\} \bullet \mathbb{T}^\omega) \cup \{\mathsf{b}\}\wr]$

$\cdots$

$\{\mathsf{a}^{n-1}\} \bullet \mathbb{T}^\omega \qquad \wr$ induction hypothesis, $n > 0$, $\mathsf{a}^0 = \epsilon\wr$

$\{a^n\} \bullet \mathbb{T}^\omega \qquad \wr \{a^n\} \bullet \mathbb{T}^\omega \sqsubseteq (\{a\} \bullet \{a^{n-1}\} \bullet \mathbb{T}^\omega) \cup \{b\} \wr$

$\ldots$

$\{a^\omega\} \qquad \wr \{a^\omega\} \sqsubseteq ((\{a\} \bullet \mathbb{T}^\omega) \cup \{b\}) \sqcup ((\{a^2\} \bullet \mathbb{T}^\omega) \cup \{b\}) \ldots ((\{a^{n+1}\} \bullet \mathbb{T}^\omega)$

$\qquad\qquad \cup \{b\}) \sqcup \ldots \wr$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

5.4 Abstraction into the finite language generated by a context-free grammar

The abstraction is $\alpha \in \wp(\mathbb{T}^\infty) \longrightarrow \wp(\mathbb{T}^\star)$, $\alpha(X) \triangleq X \cap \mathbb{T}^\star$ extended pointwise to $\dot{\alpha} \in (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)) \longrightarrow (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\star))$ as $\dot{\alpha}(\rho) \triangleq \boldsymbol{\lambda}\,\mathsf{N} \bullet \alpha(\rho(\mathsf{N}))$. We have $\langle \wp(\mathbb{T}^\infty), \sqsubseteq \rangle \xleftrightarrow[\alpha]{\mathbf{1}} \langle \wp(\mathbb{T}^\star), \subseteq \rangle$ (where $\mathbf{1}$ is the injection of $\wp(\mathbb{T}^\star)$ into $\wp(\mathbb{T}^\infty)$) hence $\langle \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty), \dot{\sqsubseteq} \rangle \xleftrightarrow[\dot{\alpha}]{\dot{\mathbf{1}}} \langle \wp(\mathbb{N} \longrightarrow \mathbb{T}^\star), \dot{\subseteq} \rangle$ pointwise. We get Ginsburg-Rice/Chomsky-Schützenberger's fixpoint characterization of the finite language generated by a context-free grammar [4,14,31] by abstracting $\mathcal{S}[\![\mathsf{G} ::= \mathsf{S}]\!] \triangleq \mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![\mathsf{S}]\!]$ into $\mathcal{S}^\star[\![\mathsf{G} ::= \mathsf{S}]\!] \triangleq \dot{\alpha}(\mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![\mathsf{S}]\!]) = \mathbf{lfp}^{\dot{\subseteq}} \mathcal{S}[\![\mathsf{S}]\!]$.

5.5 Abstraction into the infinite language generated by a context-free grammar

The abstraction $\dot{\alpha} \in (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty)) \longrightarrow (\mathbb{N} \longrightarrow \wp(\mathbb{T}^\omega))$, $\dot{\alpha}(\rho) \triangleq \boldsymbol{\lambda}\,\mathsf{N} \bullet \rho(\mathsf{N}) \cap \mathbb{T}^\omega$, such that $\langle \mathbb{N} \longrightarrow \wp(\mathbb{T}^\infty), \dot{\sqsubseteq} \rangle \xleftrightarrow[\alpha]{\dot{\mathbf{1}}} \langle \wp(\mathbb{N} \longrightarrow \mathbb{T}^\omega), \dot{\supseteq} \rangle$, leads to Nivat's fixpoint characterization of the infinite language generated by a context-free grammar [26] that is $\mathcal{S}[\![\mathsf{G}::=\mathsf{S}]\!] \triangleq \mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![\mathsf{S}]\!]$ abstracted into $\mathcal{S}^\omega[\![\mathsf{G}::=\mathsf{S}]\!] \triangleq \dot{\alpha}(\mathbf{lfp}^{\dot{\sqsubseteq}} \mathcal{S}[\![\mathsf{S}]\!]) = \mathbf{gfp}^{\dot{\subseteq}} \mathcal{S}[\![\mathsf{S}]\!]$.

6 Structural order-theoretic inductive definitions of the semantics of the call-by-value λ-calculus

The next example of structural order-theoretic inductive definition is inspired by [29,22]. We introduce a maximal trace semantics describing terminating and diverging computations. The trace semantics is then abstracted into a sound and complete relational semantics. In turn this relational semantics is abstracted into a sound reduction semantics which is incomplete since the future of computations is unpredictable. Each semantics can be defined using small steps or big steps of computation. Each semantics can be defined in fixpoint or rule-based form.

Semantics		Fixpoint definition		Rule-based definition	
		big-step	small-step	big-step	small-step
Trace	$\vec{\mathbb{S}}$	$\mathbf{lfp}^{\sqsubseteq}\,\vec{F}$	$\mathbf{lfp}^{\sqsubseteq}\,\vec{f}$	$\longmapsto$	$\overset{\ast}{\longmapsto}$
Relational	$\vec{\widehat{\mathbb{S}}}$	$\mathbf{lfp}^{\sqsubseteq}\,\vec{\widehat{F}}$	$\mathbf{lfp}^{\sqsubseteq}\,\vec{\widehat{f}}$	$\Longrightarrow$	$\overset{\ast}{\Longrightarrow}$
Reduction	$\mathbb{S}$		$\mathbf{lfp}^{\subseteq}\,f = \mathbf{gfp}^{\subseteq}\,f$		$\longrightarrow$.

These semantics including the maximal trace semantics $\vec{\mathbb{S}}$ of *Sect. 6.3.1* and the bifinitary relational semantics $\widehat{\mathbb{S}}$ of *Sect. 6.4* specify the correct finite computations which end with a value and the infinite computations but do not describe the erroneous computations so the semantics of a term that "goes wrong" is empty. Describing these erroneous computations would present no difficulty but is often irrelevant. For example in typing it must be proved that well-typed programs cannot "go wrong" (which requires to describe erroneous computations) or equivalently that well-typed programs "go well" that is have correct finite computations or diverge (in which case the semantics is simpler since erroneous computations need not to be described). The practice is also quite common in natural languages for which no one cares to describe the syntax and semantics of incorrect or meaningless sentences.

6.1   Syntax

The syntax of the λ-calculus with constants is

$$
\begin{aligned}
\mathsf{x}, \mathsf{y}, \mathsf{z}, \ldots \;\in\; & \mathbb{X} && \text{variables} \\
\mathsf{c} \;\in\; & \mathbb{C} && \text{constants } (\mathbb{X} \cap \mathbb{C} = \varnothing) \\
\mathsf{c} ::= & \; 0 \mid 1 \mid \ldots \\
\mathsf{v} \;\in\; & \mathbb{V} && \text{values} \\
\mathsf{v} ::= & \; \mathsf{c} \mid \boldsymbol{\lambda}\mathsf{x}\bullet\mathsf{a} \\
\mathsf{e} \;\in\; & \mathbb{E} && \text{errors} \\
\mathsf{e} ::= & \; \mathsf{c}\,\mathsf{a} \mid \mathsf{e}\,\mathsf{a} \\
\mathsf{a}, \mathsf{a}', \mathsf{a}_1, \ldots, \mathsf{b}, \ldots \;\in\; & \mathbb{T} && \text{terms} \\
\mathsf{a} ::= & \; \mathsf{x} \mid \mathsf{v} \mid \mathsf{a}\,\mathsf{a}'
\end{aligned}
$$

We write $\mathsf{a}[\mathsf{x} \leftarrow \mathsf{b}]$ for the capture-avoiding substitution of b for all free occurences of x within a. We let $\mathrm{FV}(\mathsf{a})$ be the free variables of a. We define the call-by-value semantics of closed terms (without free variables) $\overline{\mathbb{T}} \triangleq \{\mathsf{a} \in \mathbb{T} \mid \mathrm{FV}(\mathsf{a}) = \varnothing\}$.

The application $(\boldsymbol{\lambda}\mathsf{x}\boldsymbol{\cdot}\mathsf{a})\,\mathsf{v}$ of a function $\boldsymbol{\lambda}\mathsf{x}\boldsymbol{\cdot}\mathsf{a}$ to a value v is evaluated by substitution $\mathsf{a}[\mathsf{x}\leftarrow\mathsf{v}]$ of the actual parameter v for the formal parameter x in the function body a. This cannot be understood as induction on the program syntax since $\mathsf{a}[\mathsf{x}\leftarrow\mathsf{v}]$ is not in general a strict syntactic subcomponent of $(\boldsymbol{\lambda}\mathsf{x}\boldsymbol{\cdot}\mathsf{a})\,\mathsf{v}$.

Hence the various semantics in this Sect. 6 cannot be defined by structural induction on the syntax of λ-expressions as was the case in the previous Sect. 5. So the framework of Sect. 2 is instantiated with $\mathbb{L}=\{\blacksquare\}$ and $\prec$ is defined to be false on $\mathbb{L}$ which prevents the use of structural induction on program syntax. For brevity we omit the void syntactic component $\blacksquare$ writing e.g. F for $F[\![\blacksquare]\!]$, $\mathcal{D}$ for $\mathcal{D}_{\blacksquare}$, Δ for $\Delta_{\blacksquare}$, etc.

6.2   Trace domain

Recursion will be handled using fixpoints in the trace domain $\langle\mathcal{D}_{\blacksquare},\,\sqsubseteq_{\blacksquare}\rangle\triangleq\langle\wp(\mathbb{T}^{\infty}),\,\sqsubseteq\rangle$, which is the complete lattice $\langle\wp(\mathbb{T}^{\infty}),\,\sqsubseteq,\,\mathbb{T}^{\omega},\,\mathbb{T}^{+},\,\sqcup,\,\sqcap\rangle$ defined in Sect. 2.16.

We define the *application* $\mathsf{a}@\sigma$ *of a term* $\mathsf{a}\in\mathbb{T}$ *to a trace* $\sigma\in\mathbb{T}^{\infty}$ to be $\sigma'\in\mathbb{T}^{\infty}$ such that $|\sigma'|=|\sigma|$ and $\forall i<|\sigma|:\sigma'_i=\mathsf{a}\,\sigma_i$. Similarly the *application* $\sigma@\mathsf{a}$ *of a trace* $\sigma\in\mathbb{T}^{\infty}$ *to a term* $\mathsf{a}\in\mathbb{T}$ is σ' such that $|\sigma'|=|\sigma|$ and $\forall i<|\sigma|:\sigma'_i=\sigma_i\,\mathsf{a}$.

6.3   Big-step maximal trace semantics of the call-by-value λ-calculus

6.3.1   Fixpoint big-step maximal trace semantics

The bifinitary trace semantics $\vec{\mathbb{S}}\in\wp(\overline{\mathbb{T}}^{\infty})$ of the closed call-by-value λ-calculus $\mathbb{T}$ can be specified in fixpoint form $\vec{\mathbb{S}}=\mathbf{lfp}^{\sqsubseteq}\,\vec{F}$ where the set of traces transformer $\vec{F}\in\wp(\overline{\mathbb{T}}^{\infty})\longrightarrow\wp(\overline{\mathbb{T}}^{\infty})$ describes big steps of computation

$$\vec{F}(S)\triangleq\{\mathsf{v}\in\overline{\mathbb{T}}^{\infty}\mid\mathsf{v}\in\mathbb{V}\}\tag{a}$$

$$\cup\,\{(\boldsymbol{\lambda}\mathsf{x}\boldsymbol{\cdot}\mathsf{a})\,\mathsf{v}\boldsymbol{\cdot}\mathsf{a}[\mathsf{x}\leftarrow\mathsf{v}]\boldsymbol{\cdot}\sigma\mid\mathsf{v}\in\mathbb{V}\wedge\mathsf{a}[\mathsf{x}\leftarrow\mathsf{v}]\boldsymbol{\cdot}\sigma\in S\}\tag{b}$$

$$\cup\,\{\sigma@\mathsf{b}\mid\sigma\in S^{\omega}\}\tag{c}$$

$$\cup\,\{(\sigma@\mathsf{b})\boldsymbol{\cdot}(\mathsf{v}\,\mathsf{b})\boldsymbol{\cdot}\sigma'\mid\sigma\neq\epsilon\wedge\sigma\boldsymbol{\cdot}\mathsf{v}\in S^{+}\wedge\mathsf{v}\in\mathbb{V}\wedge(\mathsf{v}\,\mathsf{b})\boldsymbol{\cdot}\sigma'\in S\}\tag{d}$$

$$\cup\,\{\boldsymbol{\lambda}\mathsf{x}\boldsymbol{\cdot}\mathsf{a}@\sigma\mid\mathsf{a}\in\mathbb{V}\wedge\sigma\in S^{\omega}\}\tag{e}$$

$$\cup\,\{(\mathsf{a}@\sigma)\boldsymbol{\cdot}(\mathsf{a}\,\mathsf{v})\boldsymbol{\cdot}\sigma'\mid\mathsf{a},\mathsf{v}\in\mathbb{V}\wedge\sigma\neq\epsilon\wedge\sigma\boldsymbol{\cdot}\mathsf{v}\in S^{+}\wedge(\mathsf{a}\,\mathsf{v})\boldsymbol{\cdot}\sigma'\in S\}\,.\tag{f}$$

The definition of $\vec{F}$ has (a) for termination, (b) for call-by-value β-reduction, (c) and (d) for left reduction under applications and (e) and (f) for right reduction under applications, corresponding to left-to-right evaluation. (b), (d) and (f) cope both with terminating and diverging traces. In the framework of Sect. 2, we have $\Delta_\bullet \triangleq \{a, b, c, d, e, f\}$ where $\vec{F}_\bullet^i(S)$, $i \in \Delta_\bullet$ is defined by equation (i), $(i) = (a), (b), \ldots, (f)$. The join operator is chosen in binary form as $\curlyvee_\bullet \triangleq \cup$.

Lemma 13 $\vec{F}$ *is $\subseteq$-monotone but not $\sqsubseteq$-monotone.* $\square$
PROOF $\subseteq$-monotony holds for (a) and $\cup$ and can be proved for all cases (b)–(f) of the form $F(S) = \{f(\mathsf{a}, \mathsf{a}', ..., \sigma, \sigma') \mid p(\mathsf{a}, \mathsf{a}', ...) \wedge g(\sigma) \in S^+ \wedge h(\sigma') \in S\}$ so that $S \subseteq S'$ implies $F(S) \subseteq F(S')$.

For a counter-example to $\sqsubseteq$-monotony, define $X^+ \triangleq X \cap \mathbb{T}^+$, $X^\omega \triangleq X \cap \mathbb{T}^\omega$ and consider $\theta \triangleq \boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \mathsf{x}$, $X = \{(\theta\,\theta)^\omega\}$ (where $\mathsf{a}^\omega \triangleq \mathsf{a} \cdot \mathsf{a} \cdot \mathsf{a} \cdot ...$) and $Y = \{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \theta) \cdot \theta, \, (\theta\,\theta)^\omega\}$. We have $X \sqsubseteq Y$ since $X^+ = \varnothing \subseteq \{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \theta) \cdot \theta\} = Y^+$ and $X^\omega = \{(\theta\,\theta)^\omega\} \supseteq \{(\theta\,\theta)^\omega\} = Y^\omega$. However $\vec{F}(X) \not\sqsubseteq \vec{F}(Y)$. Indeed by (d), we have $((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \theta)\,\theta) \cdot (\theta\,\theta) \cdot (\theta\,\theta)^\omega = ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \theta)\,\theta) \cdot (\theta\,\theta)^\omega \in \vec{F}(Y)$ while $((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x} \, \theta)\,\theta) \cdot (\theta\,\theta)^\omega \notin \vec{F}(X)$ by examining all cases (a)—(f). ∎

So we must prove $\mathbf{lfp}^{\sqsubseteq} \vec{F}$ to exist. However, because $\vec{F}$ is not $\sqsubseteq$-monotone, $\mathbf{lfp}^{\sqsubseteq} \vec{F}$ cannot be constructed by iteration of $\vec{F}$ from $\mathbb{T}^\omega$ since infinite traces starting with a finite prefix which is not yet constructed at some iterate would definitely be eliminated in the next iterate.

Recall that $S^+ \triangleq S \cap \mathbb{T}^+$, $S^\omega \triangleq S \cap \mathbb{T}^\omega$ so $S^+ \cap S^\omega = \varnothing$ and define

$$\vec{\mathbb{S}}^+ \triangleq \mathbf{lfp}^{\subseteq} \vec{F}^+ \quad \text{where} \quad \vec{F}^+(S) \triangleq (\vec{F}(S^+))^+ \ .$$

By Lem. 13, $\vec{F}^+ \in \wp(\mathbb{T}^+) \longrightarrow \wp(\mathbb{T}^+)$ is $\subseteq$-monotone so $\mathbf{lfp}^{\subseteq} \vec{F}^+$ does exist on the complete lattice $\langle \wp(\mathbb{T}^+), \subseteq, \varnothing, \mathbb{T}^+, \cup, \cap \rangle$.

Define

$$\vec{\mathbb{S}}^\omega \triangleq \mathbf{gfp}^{\subseteq} \vec{F}^\omega \quad \text{where} \quad \vec{F}^\omega(S) \triangleq (\vec{F}(\vec{\mathbb{S}}^+ \cup S^\omega))^\omega \ .$$

By Lem. 13, $\vec{F}^\omega \in \wp(\mathbb{T}^\omega) \longrightarrow \wp(\mathbb{T}^\omega)$ is $\subseteq$-monotone so $\mathbf{gfp}^{\subseteq} \vec{F}^\omega$ does exist on the complete lattice $\langle \wp(\mathbb{T}^\omega), \subseteq, \varnothing, \mathbb{T}^\omega, \cup, \cap \rangle$.

Theorem 14

$$\vec{\mathbb{S}} \triangleq \vec{\mathbb{S}}^+ \cup \vec{\mathbb{S}}^\omega = \boldsymbol{lfp}^{\sqsubseteq} \vec{F} \ .$$

PROOF By Lem. 13 and 8. ∎

The trace semantics can also be defined coinductively (as is the case for transition systems [6, Th. 13]):

Theorem 15

$$\vec{\mathbb{S}} = \mathbf{gfp}^{\subseteq} \vec{F} \ .$$

PROOF By Lem. 13, $\vec{F}$ is $\subseteq$-monotone so $\mathbf{gfp}^{\subseteq} \vec{F}$ exists by Tarski's fixpoint theorem [33].

By Th. 14, $\vec{F}(\mathbf{lfp}^{\sqsubseteq} \vec{F}) = \mathbf{lfp}^{\sqsubseteq} \vec{F}$ so $\mathbf{lfp}^{\sqsubseteq} \vec{F} \subseteq \mathbf{gfp}^{\subseteq} \vec{F}$ by def. $\mathbf{gfp}$, proving $(\mathbf{lfp}^{\sqsubseteq} \vec{F})^{+} \subseteq (\mathbf{gfp}^{\subseteq} \vec{F})^{+}$ and $(\mathbf{lfp}^{\sqsubseteq} \vec{F})^{\omega} \subseteq (\mathbf{gfp}^{\subseteq} \vec{F})^{\omega}$. Moreover $\vec{F}(\mathbf{gfp}^{\subseteq} \vec{F}) = \mathbf{gfp}^{\subseteq} \vec{F}$ so $\mathbf{lfp}^{\sqsubseteq} \vec{F} \sqsubseteq \mathbf{gfp}^{\subseteq} \vec{F}$ by def. $\mathbf{lfp}$, proving that $(\mathbf{lfp}^{\sqsubseteq} \vec{F})^{\omega} \supseteq (\mathbf{gfp}^{\subseteq} \vec{F})^{\omega}$ hence $(\mathbf{lfp}^{\sqsubseteq} \vec{F})^{\omega} = (\mathbf{gfp}^{\subseteq} \vec{F})^{\omega}$ by antisymmetry.

It remains to prove $(\mathbf{lfp}^{\sqsubseteq} \vec{F})^{+} \supseteq (\mathbf{gfp}^{\subseteq} \vec{F})^{+}$. Given a trace $\varsigma \in (\mathbf{gfp}^{\subseteq} \vec{F})^{+} = (\vec{F}(\mathbf{gfp}^{\subseteq} \vec{F}))^{+}$, we prove that $\varsigma \in (\vec{F}(\mathbf{lfp}^{\sqsubseteq} \vec{F}))^{+} = (\mathbf{lfp}^{\sqsubseteq} \vec{F})^{+}$. The case (a) is trivial, the cases (c) and (e) are impossible since ς is finite and cases (b), (d), and (f) follow by induction on the length $|\varsigma|$ of ς. In all these case, we have $\varsigma = f(\sigma, \sigma') \in (\vec{F}(\mathbf{gfp}^{\subseteq} \vec{F}))^{+}$ with $|\sigma| < |\varsigma|$ and $|\sigma'| < |\varsigma|$ so $\sigma, \sigma' \in (\mathbf{lfp}^{\sqsubseteq} \vec{F})^{+}$ by induction hypothesis proving that $\varsigma = f(\sigma, \sigma') \in (\vec{F}(\mathbf{lfp}^{\sqsubseteq} \vec{F}))^{+} = (\mathbf{lfp}^{\sqsubseteq} \vec{F})^{+}$ by respective def. (b), (d), and (f) of $\vec{F}$. ∎

6.3.2 Properties of the maximal trace semantics

Lemma 16 *The bifinitary trace semantics* $\vec{\mathbb{S}}$ *is* suffix-closed *in that*

$$\forall \sigma \in \mathbb{T}^{\infty} : \mathsf{a} \bullet \sigma \in \vec{\mathbb{S}} \implies \sigma \in \vec{\mathbb{S}} \ .$$

PROOF We proceed by structural induction on the closed term a. Assume $\mathsf{a} \bullet \sigma \in \vec{\mathbb{S}} = \vec{F}(\vec{\mathbb{S}})$. The case $\mathsf{a} \bullet \sigma = \mathsf{v}$ is impossible since $\forall \sigma \in \mathbb{T}^{\infty} : \sigma \neq \epsilon$.

If $\mathsf{a} \bullet \sigma = (\boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{a}') \, \mathsf{v} \bullet \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma'$ then $\sigma = \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma' \in \vec{\mathbb{S}}$ by def. of $\vec{F}$.

If $\mathsf{a} \bullet \sigma = \sigma' @ \mathsf{b}$ where $\sigma' \in \vec{\mathbb{S}}^{\omega} \subseteq \vec{\mathbb{S}}$ then $\mathsf{a} = (\mathsf{a}' \, \mathsf{b})$ and $\sigma' = \mathsf{a}' \bullet \sigma'' \in \vec{\mathbb{S}}$ so $\sigma'' \in \vec{\mathbb{S}}^{\omega} \subseteq \vec{\mathbb{S}}$ by induction hypothesis proving that $\sigma = \sigma'' @ \mathsf{b} \in \vec{F}(\vec{\mathbb{S}}) = \vec{\mathbb{S}}$.

If $\mathsf{a} \bullet \sigma = (\sigma' @ \mathsf{b}) \bullet (\mathsf{v} \, \mathsf{b}) \bullet \sigma''$ where $\sigma' \bullet \mathsf{v} \in \vec{\mathbb{S}}^{+}$ and $(\mathsf{v} \, \mathsf{b}) \bullet \sigma'' \in \vec{\mathbb{S}}$ then $\sigma' = \mathsf{a}' \bullet \sigma'''$ where $\mathsf{a} = (\mathsf{a}' \, \mathsf{b})$ so $\mathsf{a}' \bullet \sigma''' \bullet \mathsf{v} \in \vec{\mathbb{S}}^{+} \subseteq \vec{\mathbb{S}}$ proving $\sigma''' \bullet \mathsf{v} \in \vec{\mathbb{S}}^{+} \subseteq \vec{\mathbb{S}}$ by induction hypothesis and so $\sigma = (\sigma''' @ \mathsf{b}) \bullet (\mathsf{v} \, \mathsf{b}) \bullet \sigma'' \in \vec{F}(\vec{\mathbb{S}}) = \vec{\mathbb{S}}$.

If $\mathsf{a} \bullet \sigma = \mathsf{a}' @ \sigma'$ where $\sigma' \in \vec{\mathbb{S}}^{\omega} \subseteq \vec{\mathbb{S}}$ then $\mathsf{a} = (\mathsf{a}' \, \mathsf{b})$ and $\sigma' = \mathsf{b} \bullet \sigma''$ so $\sigma'' \in \vec{\mathbb{S}}^{\omega} \subseteq \vec{\mathbb{S}}$ by induction hypothesis proving that $\sigma = \mathsf{a}' @ \sigma'' \in \vec{F}(\vec{\mathbb{S}}) = \vec{\mathbb{S}}$.

Finally, if $\mathsf{a} \bullet \sigma = (\mathsf{a}' @ \sigma') \bullet (\mathsf{a}' \, \mathsf{v}) \bullet \sigma''$ where $\mathsf{a}', \mathsf{v} \in \mathbb{V}$, $\sigma' \bullet \mathsf{v} \in \vec{\mathbb{S}}^+$, and $(\mathsf{a}' \, \mathsf{v}) \bullet \sigma'' \in \vec{\mathbb{S}}$ then $\mathsf{a} = (\mathsf{a}' \, \mathsf{b})$ and $\sigma' = \mathsf{b} \bullet \sigma'''$ so $\mathsf{b} \bullet \sigma''' \bullet \mathsf{v} \in \vec{\mathbb{S}}^+$ proving that $\sigma''' \bullet \mathsf{v} \in \vec{\mathbb{S}}^+$ by induction hypothesis hence $\sigma = (\mathsf{a}' @ \sigma''') \bullet (\mathsf{a}' \, \mathsf{v}) \bullet \sigma'' \in \vec{F}(\vec{\mathbb{S}}) = \vec{\mathbb{S}}$. ∎

Lemma 17 *The bifinitary trace semantics $\vec{\mathbb{S}}$ is* total *in that it excludes intermediate or result errors*

$$\forall \mathsf{a} \in \overline{\mathbb{T}} : \nexists \sigma, \sigma' \in \overline{\mathbb{T}}^\infty, \mathsf{e} \in \mathbb{E} : \mathsf{a} \bullet \sigma \bullet \mathsf{e} \bullet \sigma' \in \vec{\mathbb{S}} \ .$$

PROOF Assume, by reductio ad absurdum, that $\mathsf{a} \bullet \sigma \bullet \mathsf{e} \bullet \sigma' \in \vec{\mathbb{S}}$ then $\mathsf{e} \bullet \sigma' \in \vec{\mathbb{S}}$ since $\vec{\mathbb{S}}$ is suffix-closed. By structural induction on e, if $\mathsf{e} = \mathsf{e}_1 \, \mathsf{a}$ then, by definition of $\vec{\mathbb{S}} = \vec{F}(\vec{\mathbb{S}})$, $\exists \sigma'' : \mathsf{e}_1 \bullet \sigma'' \in \vec{\mathbb{S}}$, which is impossible by induction, or $\mathsf{e} = \mathsf{c} \, \mathsf{a}$ and then $\exists \sigma'' : \mathsf{c} \bullet \sigma'' \in \vec{\mathbb{S}} = \vec{F}(\vec{\mathbb{S}})$ so $\sigma'' = \epsilon$, which excludes all cases (c)–(f), the only possible ones for e. ∎

Lemma 18 *The finite maximal traces are* blocking *in that the result of a finite computation is always a final value*

$$\forall \sigma \in \mathbb{T}^\infty \cup \{\epsilon\} : \sigma \bullet \mathsf{b} \in \vec{\mathbb{S}}^+ \implies \mathsf{b} \in \mathbb{V} \ .$$

PROOF By induction on the length of σ and definition of $\vec{\mathbb{S}}^+ = \vec{F}(\vec{\mathbb{S}}) \cap \mathbb{T}^+$. ∎

6.3.3 Rule-based big-step maximal trace semantics

The maximal trace semantics $\vec{\mathbb{S}}$ can also be defined as follows

$$\mathsf{v} \in \vec{\mathbb{S}}, \quad \mathsf{v} \in \mathbb{V} \qquad\qquad \frac{\mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma \in \vec{\mathbb{S}}}{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a}) \, \mathsf{v} \bullet \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma \in \vec{\mathbb{S}}} \, \sqsubseteq, \quad \mathsf{v} \in \mathbb{V}$$

$$\frac{\sigma \in \vec{\mathbb{S}}^\omega}{\sigma @ \mathsf{b} \in \vec{\mathbb{S}}} \, \sqsubseteq \qquad\qquad \frac{\sigma \bullet \mathsf{v} \in \vec{\mathbb{S}}^+, \ (\mathsf{v} \, \mathsf{b}) \bullet \sigma' \in \vec{\mathbb{S}}}{(\sigma @ \mathsf{b}) \bullet (\mathsf{v} \, \mathsf{b}) \bullet \sigma' \in \vec{\mathbb{S}}} \, \sqsubseteq, \quad \mathsf{v} \in \mathbb{V}$$

$$\frac{\sigma \in \vec{\mathbb{S}}^\omega}{\mathsf{a} @ \sigma \in \vec{\mathbb{S}}} \, \sqsubseteq, \quad \mathsf{a} \in \mathbb{V} \qquad\qquad \frac{\sigma \bullet \mathsf{v} \in \vec{\mathbb{S}}^+, \ (\mathsf{a} \, \mathsf{v}) \bullet \sigma' \in \vec{\mathbb{S}}}{(\mathsf{a} @ \sigma) \bullet (\mathsf{a} \, \mathsf{v}) \bullet \sigma' \in \vec{\mathbb{S}}} \, \sqsubseteq, \quad \mathsf{v}, \mathsf{a} \in \mathbb{V} \ .$$

PROOF The set of rules $\dfrac{p_i^1(\sigma_i^1) \in \vec{\mathbb{S}}, \dots, p_i^n(\sigma_i^n) \in \vec{\mathbb{S}}}{c_i(\sigma_i^1, \dots, \sigma_i^n) \in \vec{\mathbb{S}}} \, \sqsubseteq, \ i \in \Delta$ is a shorthand

for $\dfrac{S \sqsubseteq \vec{\mathbb{S}}}{\curlyvee_\ell \{c_i(\sigma_i^1, \dots, \sigma_i^n) \mid p_i^1(\sigma_i^1) \in S, \dots, p_i^n(\sigma_i^n) \in S\} \sqsubseteq \vec{\mathbb{S}}} \, \sqsubseteq$ and $\curlyvee_\ell$ is $\cup$ in this

$\underset{i \in \Delta}{}$

example. ∎

Defining $\vec{\mathbb{S}}[\![a]\!] \triangleq \{a \bullet \sigma \mid a \bullet \sigma \in \vec{\mathbb{S}}\}$, $\vec{\mathbb{S}}^+[\![a]\!] \triangleq \{a \bullet \sigma \mid a \bullet \sigma \in \vec{\mathbb{S}}^+\}$, and $\vec{\mathbb{S}}^\omega[\![a]\!] \triangleq \{a \bullet \sigma \mid a \bullet \sigma \in \vec{\mathbb{S}}^\omega\}$, we can also write for brevity

$$v \in \vec{\mathbb{S}}[\![v]\!], \quad v \in \mathbb{V} \qquad\qquad \frac{\sigma \in \vec{\mathbb{S}}[\![a[x \leftarrow v]]\!]}{(\boldsymbol{\lambda} x \bullet a)\, v \bullet \sigma \in \vec{\mathbb{S}}[\![(\boldsymbol{\lambda} x \bullet a)\, v]\!]} \sqsubseteq, \quad v \in \mathbb{V}$$

$$\frac{\sigma \in \vec{\mathbb{S}}^\omega[\![a]\!]}{\sigma @ b \in \vec{\mathbb{S}}[\![a\, b]\!]} \sqsubseteq \qquad\qquad \frac{\sigma \bullet v \in \vec{\mathbb{S}}^+[\![a]\!], \ \sigma' \in \vec{\mathbb{S}}[\![v\, b]\!]}{(\sigma @ b) \bullet \sigma' \in \vec{\mathbb{S}}[\![a\, b]\!]} \sqsubseteq, \quad v \in \mathbb{V}$$

$$\frac{\sigma \in \vec{\mathbb{S}}^\omega[\![b]\!]}{a @ \sigma \in \vec{\mathbb{S}}[\![a\, b]\!]} \sqsubseteq, \quad a \in \mathbb{V} \qquad\qquad \frac{\sigma \bullet v \in \vec{\mathbb{S}}^+[\![b]\!], \ \sigma' \in \vec{\mathbb{S}}[\![a\, v]\!]}{(a @ \sigma) \bullet \sigma' \in \vec{\mathbb{S}}[\![a\, b]\!]} \sqsubseteq, \quad a, v \in \mathbb{V} \ .$$

PROOF $\curlyvee_\ell$ is $\cup$ and $\vec{\mathbb{S}} = \bigcup_{a \in \mathbb{T}} \vec{\mathbb{S}}[\![a]\!]$. ∎

Observe that the inductive definition of $\vec{\mathbb{S}}[\![a]\!]$ should neither be understood as a *structural induction* [28] on a (since $a[x \leftarrow v] \not\prec (\boldsymbol{\lambda} x \bullet a)\, v$) nor as *action induction* [23] (because of infinite traces). The definition could be split in inductive rules for termination and co-inductive rules for divergence, as shown in Th. 14, but the above bi-inductive definition avoids the duplication of common rules. Defining $a \Longmapsto \sigma \triangleq \sigma \in \vec{\mathbb{S}}[\![a]\!]$, we can also write

$$v \Longmapsto v, \quad v \in \mathbb{V} \qquad\qquad \frac{a[x \leftarrow v] \Longmapsto \sigma}{(\boldsymbol{\lambda} x \bullet a)\, v \Longmapsto (\boldsymbol{\lambda} x \bullet a)\, v \bullet \sigma} \sqsubseteq, \quad v \in \mathbb{V}$$

$$\frac{a \Longmapsto \sigma}{a\, b \Longmapsto \sigma @ b} \sqsubseteq, \quad \sigma \in T^\omega \qquad\qquad \frac{a \Longmapsto \sigma \bullet v, \quad v\, b \Longmapsto \sigma'}{a\, b \Longmapsto (\sigma @ b) \bullet \sigma'} \sqsubseteq, \quad v \in \mathbb{V}, \sigma \in T^+$$

$$\frac{b \Longmapsto \sigma}{a\, b \Longmapsto a @ \sigma} \sqsubseteq, \quad a \in \mathbb{V}, \sigma \in T^\omega \qquad \frac{b \Longmapsto \sigma \bullet v, \quad a\, v \Longmapsto \sigma'}{a\, b \Longmapsto (a @ \sigma) \bullet \sigma'} \sqsubseteq, \quad a, v \in \mathbb{V}, \sigma \in T^+ \ .$$

6.4 Abstraction of the big-step trace semantics into the big-step relational semantics of the call-by-value λ-calculus

6.4.1 Relational abstraction of traces

The relational abstraction of sets of traces is

$$\alpha \in \wp(\mathbb{T}^\infty) \longrightarrow \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\})) \tag{2}$$

$$\alpha(S) \triangleq \{\langle \sigma_0, \sigma_{n-1} \rangle \mid \sigma \in S \wedge |\sigma| = n\} \cup \{\langle \sigma_0, \bot \rangle \mid \sigma \in S \wedge |\sigma| = \omega\}$$

$$\gamma \in \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\})) \longrightarrow \wp(\mathbb{T}^\infty)$$

$$\gamma(T) \triangleq \{\sigma \in \mathbb{T}^\infty \mid (|\sigma| = n \wedge \langle \sigma_0, \sigma_{n-1} \rangle \in T) \vee (|\sigma| = \omega \wedge \langle \sigma_0, \bot \rangle \in T)\}$$

so that

$$\langle \wp(\mathbb{T}^\infty), \subseteq \rangle \xleftarrow[\alpha]{\gamma} \langle \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\})), \subseteq \rangle . \tag{3}$$

PROOF

$$\alpha(S) \subseteq T$$

$$\Longleftrightarrow \quad \{\langle \sigma_0, \sigma_{n-1} \rangle \mid \sigma \in S \wedge |\sigma| = n\} \cup \{\langle \sigma_0, \bot \rangle \mid \sigma \in S \wedge |\sigma| = \omega\} \subseteq T$$
$$\wr \text{def. } \alpha \wr$$

$$\Longleftrightarrow \quad \forall \sigma \in S^+ : \langle \sigma_0, \sigma_{|\sigma|-1} \rangle \in T^+ \wedge \forall \sigma \in S^\omega : \langle \sigma_0, \bot \rangle \in T^\omega$$
$$\wr \text{def. } \subseteq, \ S^+ \triangleq S \cap \mathbb{T}^+, \text{ and } S^\omega \triangleq S \cap \mathbb{T}^\omega \wr$$

$$\Longleftrightarrow \quad S^+ \subseteq \{\sigma \mid |\sigma| = n \wedge \langle \sigma_0, \sigma_{n-1} \rangle \in T\} \wedge S^\omega \subseteq \{\sigma \mid |\sigma| = \omega \wedge \langle \sigma_0, \bot \rangle \in T\}$$
$$\wr \text{def. } \subseteq, \ T^+ \triangleq T \cap (\mathbb{T} \times \mathbb{T}), \text{ and } T^\omega \triangleq T \cap (\mathbb{T} \times \{\bot\}) \wr$$

$$\Longleftrightarrow \quad S \subseteq \gamma(T) \qquad\qquad \wr S = S^+ \cup S^\omega \text{ and def. } \gamma(T) \wr \quad \blacksquare$$

6.4.2 Bifinitary relational semantics

The bifinitary relational semantics $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) \in \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\}))$ is the relational abstraction of the trace semantics mapping an expression to its final value or $\bot$ in case of divergence.

6.4.3 Fixpoint big-step bifinitary relational semantics

The bifinitary relational semantics $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) = \alpha(\mathbf{lfp}^{\sqsubseteq} \vec{F})$ can be defined in fixpoint form as $\mathbf{lfp}^{\sqsubseteq} \widehat{F}$ where the big-step transformer $\widehat{F} \in \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\})) \longrightarrow \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\}))$ is

$$\widehat{F}(T) \triangleq \{\langle \mathsf{v}, \mathsf{v} \rangle \mid \mathsf{v} \in \mathbb{V}\} \tag{4}$$

$$\cup \{\langle (\boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{a}) \, \mathsf{v}, \, r \rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}], \, r \rangle \in T\}$$

27

$\cup \{\langle (\mathsf{a}\ \mathsf{b}),\ \bot\rangle \mid \langle \mathsf{a},\ \bot\rangle \in T\}$

$\cup \{\langle (\mathsf{a}\ \mathsf{b}),\ r\rangle \mid \mathsf{a} \notin \mathbb{V} \wedge \langle \mathsf{a},\ \mathsf{v}\rangle \in T^+ \wedge \mathsf{v} \in \mathbb{V} \wedge \langle (\mathsf{v}\ \mathsf{b}),\ r\rangle \in T\}$

$\cup \{\langle (\mathsf{a}\ \mathsf{b}),\ \bot\rangle \mid \mathsf{a} \in \mathbb{V} \wedge \langle \mathsf{b},\ \bot\rangle \in T\}$

$\cup \{\langle (\mathsf{a}\ \mathsf{b}),\ r\rangle \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{b},\ \mathsf{v}\rangle \in T^+ \wedge \langle (\mathsf{a}\ \mathsf{v}),\ r\rangle \in T\}$.

Lemma 19 $\vec{F}$ *is* $\subseteq$*-monotone but not* $\sqsubseteq$*-monotone.* $\qquad\square$

PROOF $\subseteq$-monotony holds for the first constant case and $\cup$ and can be proved for all other cases of the form $F(S) = \{f(\mathsf{a}, \mathsf{a}', ..., \sigma, \sigma') \mid p(\mathsf{a}, \mathsf{a}', ...) \wedge g(\sigma) \in S^+ \wedge h(\sigma') \in S\}$ so that $S \subseteq S'$ implies $F(S) \subseteq F(S')$.

The counter-example of Lem. 13, $X = \{\langle (\theta\ \theta),\ \bot\rangle\}$ and $Y = \{\langle \boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{x}\ \theta, \theta\rangle, \langle \theta\ \theta,\ \bot\rangle\}$ with $X \sqsubseteq Y$ but $\vec{F}(X) \not\sqsubseteq \vec{F}(Y)$ shows the absence of monotony.$\blacksquare$

Lemma 20 $\alpha(\vec{F}(S)) = \widehat{F}(\alpha(S))$ $\qquad\square$

PROOF α is a complete $\cup$-morphism, so we calculate $\alpha(\vec{F}(S))$ by cases.

— $\alpha(\{\mathsf{v} \in \overline{\mathbb{T}}^\infty \mid \mathsf{v} \in \mathbb{V}\})$

$= \{\langle \mathsf{v},\ \mathsf{v}\rangle \mid \mathsf{v} \in \mathbb{V}\}$ $\qquad\qquad$ ⎰def. α and $|\mathsf{v}| = 1$⎱

— $\alpha(\{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v} \cdot \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \mid \mathsf{v} \in \mathbb{V} \wedge \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \in S\})$

$= \alpha(\{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v} \cdot \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \mid \mathsf{v} \in \mathbb{V} \wedge \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \in S^+\}) \cup$
$\ \alpha(\{(\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v} \cdot \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \mid \mathsf{v} \in \mathbb{V} \wedge \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \cdot \sigma \in S^\omega\})$
$\qquad\qquad\qquad\qquad$ ⎰$S = S^+ \cup S^\omega$ and α preserves lubs⎱

$= \{\langle (\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v},\ r\rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}],\ r\rangle \in \alpha(S)^+\} \cup$
$\ \{\langle (\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v},\ \bot\rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}],\ \bot\rangle \in \alpha(S)^\omega\}$ $\qquad$ ⎰def. α⎱

$= \{\langle (\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v},\ r\rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}],\ r\rangle \in \alpha(S)\}$
$\qquad\qquad$ ⎰def. $T^+ \triangleq T \cap (\mathbb{T} \times \mathbb{T})$ and $T^\omega \triangleq T \cap (\mathbb{T} \times \{\bot\})$⎱

— $\alpha(\{\sigma @ \mathsf{b} \mid \sigma \in S^\omega\})$

$= \{\langle (\sigma_0\ \mathsf{b}),\ \bot\rangle \mid \sigma \in S^\omega\}$ $\qquad\qquad\qquad$ ⎰def. α and $@$⎱

$= \{\langle (\sigma_0\ \mathsf{b}),\ \bot\rangle \mid \langle \sigma_0,\ \bot\rangle \in \alpha(S)\}$ $\qquad\qquad\qquad$ ⎰def. α⎱

$= \{\langle (\mathsf{a}\ \mathsf{b}),\ \bot\rangle \mid \langle \mathsf{a},\ \bot\rangle \in \alpha(S)\}$ $\qquad\qquad$ ⎰$S \subseteq \mathbb{T}^\infty$ so $\sigma_0 \in \mathbb{T}$⎱

— $\alpha(\{(\sigma @ \mathsf{b}) \cdot (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \mid \sigma \cdot \mathsf{v} \in S^+ \wedge \mathsf{v} \in \mathbb{V} \wedge (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \in S\})$

$= \alpha(\{(\sigma @ \mathsf{b}) \cdot (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \mid \sigma \cdot \mathsf{v} \in S^+ \wedge \mathsf{v} \in \mathbb{V} \wedge (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \in S^+\}) \cup$
$\ \alpha(\{(\sigma @ \mathsf{b}) \cdot (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \mid \sigma \cdot \mathsf{v} \in S^+ \wedge \mathsf{v} \in \mathbb{V} \wedge (\mathsf{v}\ \mathsf{b}) \cdot \sigma' \in S^\omega\})$
$\qquad\qquad\qquad\qquad$ ⎰$S = S^+ \cup S^\omega$ and α preserves lubs⎱

28

$=$ $\{\langle(\sigma_0\ \mathsf{b}),\ r\rangle \mid \sigma \bullet \mathsf{v} \in S^+ \wedge \mathsf{v} \in \mathbb{V} \wedge \langle(\mathsf{v}\ \mathsf{b}),\ r\rangle \in \alpha(S)^+\} \cup$
$\{\langle(\sigma\ \mathsf{b}),\ \bot\rangle \mid \sigma \bullet \mathsf{v} \in S^+ \wedge \mathsf{v} \in \mathbb{V} \wedge \langle(\mathsf{v}\ \mathsf{b}),\ \bot\rangle \in \alpha(S)^\omega\})$ ⟨def. α and @⟩

$=$ $\{\langle(\sigma_0\ \mathsf{b}),\ r\rangle \mid \langle\sigma_0,\ \mathsf{v}\rangle \in \alpha(S)^+ \wedge \mathsf{v} \in \mathbb{V} \wedge \langle(\mathsf{v}\ \mathsf{b}),\ r\rangle \in \alpha(S)\}$

⟨def. $T^+ \triangleq T \cap (\mathbb{T} \times \mathbb{T})$, $T^\omega \triangleq T \cap (\mathbb{T} \times \{\bot\})$, and α⟩

$=$ $\{\langle(\mathsf{a}\ \mathsf{b}),\ r\rangle \mid \langle\mathsf{a},\ \mathsf{v}\rangle \in \alpha(S)^+ \wedge \mathsf{v} \in \mathbb{V} \wedge \langle(\mathsf{v}\ \mathsf{b}),\ r\rangle \in \alpha(S)\}$

⟨$S \subseteq \mathbb{T}^\infty$ so $\sigma_0 \in \mathbb{T}$⟩

— $\alpha(\{\mathsf{a} @ \sigma \mid \mathsf{a} \in \mathbb{V} \wedge \sigma \in S^\omega\})$

$=$ $\{\langle(\mathsf{a}\ \sigma_0),\ \bot\rangle \mid \mathsf{a} \in \mathbb{V} \wedge \sigma \in S^\omega\}$ ⟨def. α and @⟩

$=$ $\{\langle(\mathsf{a}\ \sigma_0),\ \bot\rangle \mid \mathsf{a} \in \mathbb{V} \wedge \langle\sigma_0,\ \bot\rangle \in \alpha(S)\}$ ⟨def. α and $T^\omega \triangleq T \cap (\mathbb{T} \cup \{\bot\})$⟩

$=$ $\{\langle(\mathsf{a}\ \mathsf{b}),\ \bot\rangle \mid \mathsf{a} \in \mathbb{V} \wedge \langle\mathsf{b},\ \bot\rangle \in \alpha(S)\}$ ⟨$S \subseteq \mathbb{T}^\infty$ so $\sigma_0 \in \mathbb{T}$⟩

— $\alpha(\{(\mathsf{a} @ \sigma) \bullet (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \sigma \bullet \mathsf{v} \in S^+ \wedge (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \in S\})$

$=$ $\alpha(\{(\mathsf{a} @ \sigma) \bullet (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \sigma \bullet \mathsf{v} \in S^+ \wedge (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \in S^+\}) \cup$
$\alpha(\{(\mathsf{a} @ \sigma) \bullet (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \sigma \bullet \mathsf{v} \in S^+ \wedge (\mathsf{a}\ \mathsf{v}) \bullet \sigma' \in S^\omega\})$

⟨$S = S^+ \cup S^\omega$ and α preserves lubs⟩

$=$ $\{\langle(\mathsf{a}\ \sigma_0),\ r\rangle \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \langle\sigma_0,\ \mathsf{v}\rangle \in \alpha(S)^+ \wedge \langle(\mathsf{a}\ \mathsf{v}),\ r\rangle \in \alpha(S)^+\} \cup$
$\{\langle(\mathsf{a}\ \sigma_0),\ \bot\rangle \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \langle\sigma_0,\ \mathsf{v}\rangle \in \alpha(S)^+ \wedge \langle(\mathsf{a}\ \mathsf{v}),\ \bot\rangle \in \alpha(S)^\omega\}$ ⟨def. α⟩

$=$ $\{\langle(\mathsf{a}\ \mathsf{b}),\ r\rangle \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \langle\mathsf{b},\ \mathsf{v}\rangle \in \alpha(S) \wedge \langle(\mathsf{a}\ \mathsf{v}),\ r\rangle \in \alpha(S)\}$

⟨$T^\omega \triangleq T \cap (\mathbb{T} \cup \{\bot\})$ and $S \subseteq \mathbb{T}^\infty$ so $\sigma_0 \in \mathbb{T}$⟩ .

Hence, we have the commutation property $\alpha(\vec{F}(S)) = \widehat{F}(\alpha(S))$ when defining $\widehat{F}$ by (4). ∎

Lemma 21 $\widehat{\mathbb{S}}^+ \triangleq \alpha(\vec{\mathbb{S}}^+) = \mathbf{lfp}^\subseteq \widehat{F}^+$ *where* $\widehat{F}^+(S) \triangleq \widehat{F}(S^+)$. □
PROOF To prove that $\alpha(\vec{\mathbb{S}}^+) = \alpha(\mathbf{lfp}^\subseteq \vec{F}^+)$ is equal to $\mathbf{lfp}^\subseteq \widehat{F}^+ = \widehat{\mathbb{S}}^+$, we observe that α preserves $\cup$ and $\alpha \circ \vec{F}^+ = \widehat{F}^+ \circ \alpha$ by **Lem. 20** so $\alpha(\mathbf{lfp}^\subseteq \vec{F}^+)$ $= \mathbf{lfp}^\subseteq \widehat{F}^+$ by [6, Th. 3]. ∎

Lemma 22 $\widehat{\mathbb{S}}^\omega \triangleq \alpha(\vec{\mathbb{S}}^\omega) = \mathbf{gfp}^\subseteq \widehat{F}^\omega$ *where* $\widehat{F}^\omega(S) \triangleq (\widehat{F}(\widehat{\mathbb{S}}^+ \cup S^\omega))^\omega$. □
PROOF We must prove that $\alpha(\vec{\mathbb{S}}^\omega) = \alpha(\mathbf{gfp}^\subseteq \vec{F}^\omega)$ is equal to $\mathbf{gfp}^\subseteq \widehat{F}^\omega = \widehat{\mathbb{S}}^\omega$.

— To prove that $\alpha(\mathbf{gfp}^\subseteq \vec{F}^\omega) \subseteq \mathbf{gfp}^\subseteq \widehat{F}^\omega$, we let X^δ, $\delta \in \mathcal{O}$ and $\overline{X}^\delta$, $\delta \in \mathcal{O}$ be the respective transfinite iterates of $\vec{F}^\omega$ and $\widehat{F}^\omega$ from $X^0 = \mathbb{T}^\omega$ and $\overline{X}^0 = \mathbb{T} \times \{\bot\}$ so that $\alpha(X^0) \subseteq \overline{X}^0$ hence $X^0 \subseteq \gamma(\overline{X}^0)$ by (3) in Sect. 6.4.1. Assume, by induction hypothesis, that $\forall \beta < \delta : X^\beta \subseteq \gamma(\overline{X}^\beta)$. We have $\forall \beta < \delta : (\bigcap_{\beta' < \delta} X^{\beta'}) \subseteq \gamma(\overline{X}^\beta)$ hence $(\bigcap_{\beta < \delta} X^\beta) \subseteq (\bigcap_{\beta < \delta} \gamma(\overline{X}^\beta))$ by definition of the greatest lower bound (glb) $\cap$ and therefore $(\bigcap_{\beta < \delta} X^\beta) \subseteq \gamma(\bigcap_{\beta < \delta} \overline{X}^\beta)$ by (3) in Sect. 6.4.1 so $X^\delta = \vec{F}^\omega(\bigcap_{\beta < \delta} X^\beta) \subseteq \vec{F}^\omega(\gamma(\bigcap_{\beta < \delta} \overline{X}^\beta))$ by monotony. It follows that $X^\delta \subseteq \gamma(\widehat{F}^\omega(\bigcap_{\beta < \delta} \overline{X}^\beta)) = \gamma(\overline{X}^\delta)$ since $\alpha \circ \vec{F}^\omega = \widehat{F}^\omega \circ \alpha$ by

29

Lem. 20 implies $\alpha \circ \vec{F}^\omega \circ \gamma = \widehat{\vec{F}}^\omega \circ \alpha \circ \gamma$ hence $\alpha \circ \vec{F}^\omega \circ \gamma \subseteq \widehat{\vec{F}}^\omega$ by (3) in Sect. 6.4.1 and monotony that is $\vec{F}^\omega \circ \gamma \mathrel{\dot{\subseteq}} \gamma \circ \widehat{\vec{F}}^\omega$ by (3) in Sect. 6.4.1. Hence $\exists \lambda \in \mathcal{O} : \mathbf{gfp}^{\subseteq} \vec{F}^\omega = X^\lambda \subseteq \gamma(\overline{X}^\lambda) = \gamma(\mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega)$ and we conclude by (3) in Sect. 6.4.1.

— To prove that $\mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega \subseteq \alpha(\mathbf{gfp}^{\subseteq} \vec{F}^\omega)$, we show that $\forall \langle \mathsf{a}, \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega :$ $\exists \sigma \in \mathbf{gfp}^{\subseteq} \vec{F}^\omega : \sigma_0 = \mathsf{a}$. To do so for any $\langle \mathsf{a}, \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega$, we prove by transfinite induction on δ that

$$\forall \delta \in \mathcal{O} > 0 : \forall \langle \mathsf{a}, \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega : \exists \sigma \in \mathbb{T}^\omega : \sigma_0 = \mathsf{a} \wedge \sigma \in \bigcap_{\beta < \delta} X^\beta .$$

For $\delta = 1$, $\bigcap_{\beta < \delta} X^\beta = X^0 = \mathbb{T}^\omega$ and $\mathsf{a} \in \mathbb{T}$.

Assume by induction hypothesis, that $\exists \sigma \in \mathbb{T}^\omega : \sigma_0 = \mathsf{a} \wedge \forall \eta \in \mathcal{O} : 0 < \eta < \delta : \sigma \in \bigcap_{\beta < \eta} X^\beta$. We have $\sigma \in \bigcap_{\eta < \delta} \bigcap_{\beta < \eta} X^\beta = \bigcap_{\beta < \delta} X^\beta$ et we must show that $\exists \sigma \in \mathbb{T}^\omega : \sigma_0 = \mathsf{a} \wedge \sigma \in X^\delta = \vec{F}^\omega(\bigcap_{\beta < \delta} X^\beta)$. Because the iterates X^δ, $\delta \in \mathcal{O}$ are decreasing, this implies $\exists \sigma \in \mathbb{T}^\omega : \sigma_0 = \mathsf{a} \wedge \sigma \in \bigcap_{\beta < \delta} X^\beta$.

It remains to show, by structural case analysis on a, that if $\sigma \in S : \sigma_0 = \mathsf{a}$, then $\exists \sigma' \in \vec{F}(S) : \sigma'_0 = \mathsf{a}$ where $S = \bigcap_{\beta < \delta} X^\beta$.

— If $\mathsf{a} \in \mathbb{V}$ then $\langle \mathsf{a}, \bot \rangle \notin \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega$.

— If $\mathsf{a} = (\boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{a}') \, \mathsf{v}$, $\mathsf{v} \in \mathbb{V}$ then $\langle \mathsf{a}, \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega = \widehat{\vec{F}}^\omega(\mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega)$ so by (4), $\langle \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}], \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega$. By induction on δ, we have $\exists \sigma' \in \mathbb{T}^\omega : \sigma'_0 = \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \wedge \sigma' \in \bigcap_{\beta < \delta} X^\beta$ so that, by (b), $(\boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{a}') \, \mathsf{v} \bullet \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma' \in \vec{F}(\bigcap_{\beta < \delta} X^\beta) = X^\delta$.

— If $\mathsf{a} = (\mathsf{a}' \, \mathsf{b})$ then there are four subcases.

— If $\langle \mathsf{a}', \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega \subseteq \bigcap_{\beta < \delta} X^\beta$ then, by induction hypothesis on δ, we have $\exists \sigma' \in \mathbb{T}^\omega : \sigma'_0 = \mathsf{a}' \wedge \sigma' \in \bigcap_{\beta < \delta} X^\beta$ so that, by (c), $\sigma' \, @ \, \mathsf{b} \in \vec{F}(\bigcap_{\beta < \delta} X^\beta) = X^\delta$ is such that $\sigma'_0 = (\mathsf{a}' \, \mathsf{b}) = \mathsf{a}$ by definition of $@$.

— If $\langle \mathsf{a}', \mathsf{v} \rangle \in \widehat{\vec{S}}^+ = \alpha(\vec{S}^+)$, $\mathsf{v} \in \mathbb{V}$, and $\langle (\mathsf{v} \, \mathsf{b}), \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega$ then, by induction hypothesis on δ, we have $\exists \sigma' \in \mathbb{T}^\omega : \sigma'_0 = (\mathsf{v} \, \mathsf{b}) \wedge \sigma' \in \bigcap_{\beta < \delta} X^\beta$. By definition (2) of α in Sect. 6.4.1, there exists $\varsigma \in \mathbb{T}^+ : \varsigma \in \vec{S}^+ \wedge |\varsigma| = n \wedge \langle \varsigma_0, \varsigma_{n-1} \rangle = \langle \mathsf{a}', \mathsf{v} \rangle$ proving by definition (d) of $\widehat{\vec{F}}$ that $\exists \sigma'' = (\varsigma \, @ \mathsf{b}) \mathbin{\mathsf{\$}} \sigma' \in \widehat{\vec{F}}(\bigcap_{\beta < \delta} X^\beta) = X^\delta$ where, by definition, $\varsigma \bullet \mathsf{c} \mathbin{\mathsf{\$}} \mathsf{c} \bullet \varsigma' \triangleq \varsigma \bullet \mathsf{c} \bullet \varsigma'$. We have $\sigma''_0 = (\varsigma \, @ \mathsf{b})_0 = (\varsigma_0 \, @ \mathsf{b}) = (\mathsf{a}' \, @ \mathsf{b}) = \mathsf{a}$.

— If $\mathsf{a}' \in \mathbb{V}$ and $\langle \mathsf{b}, \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{\vec{F}}^\omega$ then by induction hypothesis on δ, $\exists \sigma' \in \mathbb{T}^\omega : \sigma_0 = \mathsf{b} \wedge \sigma' \in \bigcap_{\beta < \delta} X^\beta$ proving by definition (e) of $\widehat{\vec{F}}$ that $\sigma = \mathsf{a}' \, @ \, \sigma' \in \widehat{\vec{F}}(\bigcap_{\beta < \delta} X^\beta) = X^\delta$ with $\sigma_0 = (\mathsf{a}' \, @ \, \sigma')_0 = (\mathsf{a}' \, \sigma'_0) = (\mathsf{a}' \, \mathsf{b}) = \mathsf{a}$.

30

— If $a', v \in \mathbb{V}$, $\langle b, v \rangle \in \widehat{\mathbb{S}}^+ = \alpha(\vec{\mathbb{S}}^+)$, and $\langle (a'\ v), \bot \rangle \in \mathbf{gfp}^{\subseteq} \widehat{F}^{\omega}$ then, by induction hypothesis on δ, we have $\exists \sigma' \in \mathbb{T}^{\omega} : \sigma'_0 = (a'\ v) \wedge \sigma' \in \bigcap_{\beta < \delta} X^{\beta}$. By definition (2) in Sect. 6.4.1 of α, there exists $\varsigma \in \mathbb{T}^+ : \varsigma \in \vec{\mathbb{S}}^+ \wedge |\varsigma| = n \wedge \langle \varsigma_0, \varsigma_{n-1} \rangle = \langle b, v \rangle$ proving by definition (f) of $\vec{F}$ that $(a'@\varsigma)_{\S}\sigma' \in \vec{F}(\bigcap_{\beta < \delta} X^{\beta}) = X^{\delta}$ with $\sigma_0 = (a'\ @\ \varsigma)_0 = (a'\ \varsigma_0) = (a'\ b) = a$. ∎

Theorem 23 $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) = \alpha(\mathit{lfp}^{\sqsubseteq} \vec{F}) = \mathit{lfp}^{\sqsubseteq} \widehat{F}$. □

PROOF By Th. 14 and Lem. 20, we have $\vec{\mathbb{S}} = \vec{F}(\vec{\mathbb{S}})$ so $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) = \alpha(\vec{F}(\vec{\mathbb{S}})) = \widehat{F}(\alpha(\vec{\mathbb{S}})) = \widehat{F}(\widehat{\mathbb{S}})$ proving that $\widehat{\mathbb{S}}$ is a fixpoint of $\widehat{F}$. By Lem. 21, 22, and 8, we have $\widehat{\mathbb{S}} = \mathbf{lfp}^{\sqsubseteq} \widehat{F}$. ∎

Contrary to the case of the trace semantics Th. 15, the relational semantics _cannot_ be defined coinductively which would validate incorrect evaluations of the form $a \Rightarrow v$ where a actually diverges [22]. This phenomenon was already observed for transition systems [6, Sect. 5].

Theorem 24 $(\mathit{lfp}^{\sqsubseteq} \widehat{F})^+ \subsetneq (\mathit{gfp}^{\subseteq} \widehat{F})^+$ _and_ $(\mathit{lfp}^{\sqsubseteq} \widehat{F})^{\omega} = (\mathit{gfp}^{\subseteq} \widehat{F})^{\omega}$ _so_

$$\widehat{\mathbb{S}} \neq \mathit{gfp}^{\subseteq} \widehat{F} \ .$$

PROOF By Lem. 19, $\widehat{F}$ is $\subseteq$-monotone so $\mathbf{gfp}^{\subseteq} \widehat{F}$ exists by Tarski's fixpoint theorem [33].

By Th. 23, $\widehat{F}(\mathbf{lfp}^{\sqsubseteq} \widehat{F}) = \mathbf{lfp}^{\sqsubseteq} \widehat{F}$ so $\mathbf{lfp}^{\sqsubseteq} \widehat{F} \subseteq \mathbf{gfp}^{\subseteq} \widehat{F}$ by def. $\mathbf{gfp}$, proving $(\mathbf{lfp}^{\sqsubseteq} \widehat{F})^+ \subseteq (\mathbf{gfp}^{\subseteq} \widehat{F})^+$ and $(\mathbf{lfp}^{\sqsubseteq} \widehat{F})^{\omega} \subseteq (\mathbf{gfp}^{\subseteq} \widehat{F})^{\omega}$. Moreover $\widehat{F}(\mathbf{gfp}^{\subseteq} \widehat{F}) = \mathbf{gfp}^{\subseteq} \widehat{F}$ so $\mathbf{lfp}^{\sqsubseteq} \widehat{F} \sqsubseteq \mathbf{gfp}^{\subseteq} \widehat{F}$ by def. $\mathbf{lfp}$, proving that $(\mathbf{lfp}^{\sqsubseteq} \widehat{F})^{\omega} \supseteq (\mathbf{gfp}^{\subseteq} \widehat{F})^{\omega}$ hence $(\mathbf{lfp}^{\sqsubseteq} \widehat{F})^{\omega} = (\mathbf{gfp}^{\subseteq} \widehat{F})^{\omega}$ by antisymmetry.

Let $\theta \triangleq \boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{x}\ \mathsf{x}$ and $0 \triangleq \boldsymbol{\lambda} \mathsf{f} \bullet \boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{x}$. $\langle \theta\ \theta, 0 \rangle$ belongs to $\overline{\mathbb{T}}^{\infty}$. If $\langle \theta\ \theta, 0 \rangle = \langle \mathsf{x}\ \mathsf{x}[\mathsf{x} \leftarrow \theta], 0 \rangle$ belongs to an iterate of $\widehat{F}$ then, by def. (4) of $\widehat{F}$, $\langle (\boldsymbol{\lambda} \mathsf{x} \bullet \mathsf{x}\ \mathsf{x})\ \theta, 0 \rangle = \langle \theta\ \theta, 0 \rangle$ belongs to the next iterate, hence, by transfinite induction on the iterates, to $\mathbf{gfp}^{\subseteq} \widehat{F}$. However, there is no finite trace in $\vec{\mathbb{S}}$ starting with term $\theta\ \theta$ and ending with term 0 so, by Th. 23, $\langle \theta\ \theta, 0 \rangle \notin \alpha(\vec{\mathbb{S}}) = \mathbf{lfp}^{\sqsubseteq} \widehat{F}$, proving $(\mathbf{lfp}^{\sqsubseteq} \widehat{F})^+ \neq (\mathbf{gfp}^{\subseteq} \widehat{F})^+$. ∎

6.4.4 Rule-based big-step bifinitary relational semantics

The big-step bifinitary relational semantics $\Rightarrow$ is defined as $a \Rightarrow r \triangleq \langle a, r \rangle \in \alpha(\vec{\mathbb{S}}[\![a]\!])$ where $a \in \mathbb{T}$ and $r \in \mathbb{T} \cup \{\bot\}$. It is

$$v \Rightarrow v, \quad v \in \mathbb{V} \qquad\qquad \frac{a[x \leftarrow v] \Rightarrow r}{(\lambda x \cdot a)\, v \Rightarrow r} \; \sqsubseteq, \quad v \in \mathbb{V},\; r \in \mathbb{V} \cup \{\bot\}$$

$$\frac{a \Rightarrow \bot}{a\,b \Rightarrow \bot} \; \sqsubseteq \qquad\qquad \frac{a \Rightarrow v, \quad v\,b \Rightarrow r}{a\,b \Rightarrow r} \; \sqsubseteq, \quad a \notin \mathbb{V},\; v \in \mathbb{V},\; r \in \mathbb{V} \cup \{\bot\}$$

$$\frac{b \Rightarrow \bot}{a\,b \Rightarrow \bot} \; \sqsubseteq, \quad a \in \mathbb{V} \qquad \frac{b \Rightarrow v, \quad a\,v \Rightarrow r}{a\,b \Rightarrow r} \; \sqsubseteq, \quad a \in \mathbb{V},\; v \in \mathbb{V},\; r \in \mathbb{V} \cup \{\bot\} \, .$$

Again this should neither be understood as a structural induction (since $a[x \leftarrow v] \not\prec (\lambda x \cdot a)\, v$) nor as action induction (because of infinite behaviors) nor as co-induction by Th. 24. The abstraction $\alpha(T) \triangleq T \cap (\mathbb{T} \times \mathbb{T})$ yields (a variant of) the classical natural semantics [17] (where all rules with $\bot$ are eliminated and $\sqsubseteq$ becomes $\subseteq$ in the remaining ones). The abstraction $\alpha(T) \triangleq T \cap (\mathbb{T} \times \{\bot\})$ yields the divergence semantics (keeping only the rules with $\bot$, $\sqsubseteq$ is $\supseteq$, and $a \Rightarrow \bot$ is written $a \overset{\infty}{\Rightarrow}$ in [22]).

The above big-step bifinitary relational semantics $\Rightarrow$ is equivalent but not identical to the standard big-step semantics whose bifinitary generalization would be

$$v \Rightarrow v, \quad v \in \mathbb{V} \qquad \frac{a \Rightarrow \lambda x \cdot c, \quad b \Rightarrow v', \quad c[x \leftarrow v'] \Rightarrow r}{a\,b \Rightarrow r} \; \sqsubseteq, \quad \begin{array}{l} v, v' \in \mathbb{V}, \\ r \in \mathbb{V} \cup \{\bot\} \end{array}$$

$$\frac{a \Rightarrow \bot}{a\,b \Rightarrow \bot} \; \sqsubseteq \qquad \frac{a \Rightarrow v, \quad b \Rightarrow \bot}{a\,b \Rightarrow \bot} \; \sqsubseteq, \quad v \in \mathbb{V}$$

We have chosen to break evaluations of applications in smaller chunks instead so as to enforce evaluation of the function before that of the arguments and to make explicit the reduction step in the trace semantics.

6.5 Abstraction of the big-step trace semantics into the small-step reduction semantics of the call-by-value λ-calculus

The small-step reduction semantics abstracts the trace semantics by collecting all transitions along any trace.

6.5.1 Small-step abstraction of traces

The abstraction is

$$\alpha_s \in \wp(\mathbb{T}^\infty) \longrightarrow \wp(\mathbb{T} \times \mathbb{T})$$
$$\alpha_s(S) \triangleq \{\langle \sigma_i, \sigma_{i+1}\rangle \mid \sigma \in S \wedge 0 \leqslant i \wedge i+1 < |\sigma|\} \ .$$

Since the bifinitary trace semantics is suffix-closed, we can also use

$$\alpha \in \wp(\mathbb{T}^\infty) \longrightarrow \wp(\mathbb{T} \times \mathbb{T})$$
$$\alpha(S) \triangleq \{\langle \sigma_0, \sigma_1\rangle \mid \sigma \in S \wedge |\sigma| > 1\}$$

so that we have $\alpha_s(S) = \alpha(S)$ whenever S is suffix-closed. By defining $\overline{\wp}(\mathbb{T}^\infty)$ to be the set of suffix-closed and blocking subsets of $\mathbb{T}^\infty$ and $\gamma(\tau)$ to be the set of maximal traces generated by the transition relation $\tau \in \wp(\mathbb{T} \times \mathbb{T})$ that is

$$\gamma^+(\tau) \triangleq \{\sigma \in \mathbb{T}^+ \mid \forall i < |\sigma| : \langle \sigma_i, \sigma_{i+1}\rangle \in \tau \wedge \forall \mathsf{a} \in \mathbb{T} : \langle \sigma_{<|\sigma|-1}, \mathsf{a}\rangle \notin \tau\}$$
$$\gamma^\omega(\tau) \triangleq \{\sigma \in \mathbb{T}^\omega \mid \forall i \in \mathbb{N} : \langle \sigma_i, \sigma_{i+1}\rangle \in \tau\}$$
$$\gamma(\tau) \triangleq \gamma^+(\tau) \cup \gamma^\omega(\tau) \ ,$$

we have

$$\langle \overline{\wp}(\mathbb{T}^\infty), \subseteq\rangle \xleftarrow[\alpha]{\gamma} \langle \wp((\mathbb{T} \setminus \mathbb{V}) \times \mathbb{T}), \subseteq\rangle \ . \tag{5}$$

PROOF Assume that $S \in \overline{\wp}(\mathbb{T}^\infty)$, $\tau \subseteq (\mathbb{T} \setminus \mathbb{V}) \times \mathbb{T}$, and $\alpha(S) \subseteq \tau$ so that the first transition along a trace $\sigma \in S$ is also a transition in τ hence any transition along a trace $\sigma \in S$ is also a transition in τ since S is suffix-closed. If σ is infinite, it is a trace generated by τ hence $\sigma \in \gamma(\tau)$. If $\sigma \in S$ is finite, then $\sigma_{|\sigma|-1} \in \mathbb{V}$ since S is blocking so $\sigma_{|\sigma|-1}$ has no possible successor by $\tau \subseteq (\mathbb{T} \setminus \mathbb{V}) \times \mathbb{T}$ proving again that $\sigma \in \gamma(\tau)$ that is $S \subseteq \gamma(\tau)$.

Reciprocally, if $S \subseteq \gamma(\tau)$ and $\sigma \in S$, then by definition of γ, all transitions in σ are also transitions of τ proving that $\alpha(\{\sigma\}) \subseteq \tau$ hence $\alpha(S) = \alpha(\bigcup_{\sigma \in S}\{\sigma\}) \subseteq \tau$. ∎

Observe that this Galois connection is relative to $\langle \overline{\wp}(\mathbb{T}^\infty), \subseteq\rangle$ and is not valid for $\langle \overline{\wp}(\mathbb{T}^\infty), \sqsubseteq\rangle$. Besides absence of monotony, this is another reason why the abstraction theorem Th. 9 is not applicable for $\sqsubseteq$. Indeed the small-step reduction semantics is essentially incomplete in that it cannot anticipate that a computation will go wrong as was the case for the trace semantics and its relational abstraction.

6.5.2   Small-step reduction semantics

The small-step reduction semantics or transition semantics $\mathbb{S}$ is defined as

$$\mathbb{S} \triangleq \mathbf{lfp}^{\subseteq} f \tag{6}$$

$$f(\tau) \triangleq \{\langle (\boldsymbol{\lambda} x \cdot a)\; v,\; a[x \leftarrow v]\rangle\} \cup \{\langle a_0\; b,\; a_1\; b\rangle \mid \langle a_0,\, a_1\rangle \in \tau\} \cup$$

$$\{\langle v\; b_0,\; v\; b_1\rangle \mid v \in \mathbb{V} \wedge \langle b_0,\, b_1\rangle \in \tau\}\;.$$

$\langle \wp((\mathbb{T} \setminus \mathbb{V}) \times \mathbb{T}),\, \subseteq\rangle$ is a complete lattice and f is $\subseteq$–monotone so $\mathbf{lfp}^{\subseteq} f$ does exist by Tarski's fixpoint theorem [33]. The rule-based presentation of (6) has a call-by-value β-reduction axiom plus two context rules for reducing under applications, corresponding to left-to-right evaluation [29]. $a \longrightarrow b$ stands for $\langle a,\, b\rangle \in \mathbb{S}$ and $v \in \mathbb{V}$.

$$((\boldsymbol{\lambda} x \cdot a)\; v) \longrightarrow a[x \leftarrow v] \qquad \frac{a_0 \longrightarrow a_1}{a_0\; b \longrightarrow a_1\; b} \subseteq \qquad \frac{b_0 \longrightarrow b_1}{v\; b_0 \longrightarrow v\; b_1} \subseteq \;.$$

Lemma 25 $\alpha \circ \vec{F} \circ \gamma \mathrel{\dot{\subsetneq}} f$ $\square$

PROOF α is a complete $\cup$-morphism so we calculate $\alpha(\vec{F}(S))$ by cases.

$\quad$— $\quad \alpha(\{a \in \overline{\mathbb{T}}^{\infty} \mid a \in \mathbb{V}\})$

$\quad=\quad \varnothing$ $\wr\text{def. }\alpha\wr$

$\quad$— $\quad \alpha(\{(\boldsymbol{\lambda} x \cdot a)\; v \bullet a[x \leftarrow v] \bullet \sigma \mid v \in \mathbb{V} \wedge a[x \leftarrow v] \bullet \sigma \in S\})$

$\quad=\quad \{\langle (\boldsymbol{\lambda} x \cdot a)\; v,\; a[x \leftarrow v]\rangle\}$ $\wr\text{def. }\alpha\wr$

$\quad$— $\quad \alpha(\{\sigma \,@\, b \mid \sigma \in S^{\omega}\})$

$\quad=\quad \{\langle \sigma_0\; b,\; \sigma_1\; b\rangle \mid \sigma \in S^{\omega}\}$ $\wr\text{def. }\alpha\text{ and }@\wr$

$\quad=\quad \{\langle a_0\; b,\; a_1\; b\rangle \mid \langle a_0,\, a_1\rangle \in \alpha(S^{\omega})\}$ $\wr\text{def. }\alpha\text{ and }S^{\omega} \subseteq \mathbb{T}^{\omega}\wr$

$\quad$— $\quad \alpha(\{(\sigma \,@\, b) \bullet (v\; b) \bullet \sigma' \mid \sigma \bullet v \in S^{+} \wedge v \in \mathbb{V} \wedge (v\; b) \bullet \sigma' \in S\})$

$\quad=\quad \{\langle a\; b,\; v\; b\rangle \mid a \bullet v \in S^{+} \wedge v \in \mathbb{V} \wedge (v\; b) \bullet \sigma' \in S\} \cup$
$\quad\quad\quad \{\langle \sigma_0\; b,\; \sigma_1\; b\rangle \mid \sigma \bullet v \in S^{+} \wedge |\sigma| > 1 \wedge v \in \mathbb{V} \wedge (v\; b) \bullet \sigma' \in S\}$
                                                          $\wr\text{def. }\alpha\text{ and }|\sigma| > 0\wr$

$\quad\subseteq\quad \{\langle a\; b,\; v\; b\rangle \mid a \bullet v \in S^{+}\} \cup \{\langle \sigma_0\; b,\; \sigma_1\; b\rangle \mid \sigma \bullet v \in S^{+} \wedge |\sigma| > 1\}$
                                             $\wr\text{ignoring that }a\text{ or }(v\; b)\text{ might "go wrong"}\wr$

$\quad=\quad \{\langle \sigma_0\; b,\; \sigma_1\; b\rangle \mid \sigma \in S^{+} \wedge |\sigma| > 1\}$

$\quad=\quad \{\langle a_0\; b,\; a_1\; b\rangle \mid \langle a_0,\, a_1\rangle \in \alpha(S^{+})\}$ $\wr\text{def. }\alpha\text{ and }S^{+} \subseteq \mathbb{T}^{+}\wr$

— $\quad \alpha(\{\mathsf{a} \, @ \, \sigma \mid \mathsf{a} \in \mathbb{V} \wedge \sigma \in S^\omega\})$

$= \quad \{\langle \mathsf{a} \, \sigma_0, \, \mathsf{a} \, \sigma_1 \rangle \mid \mathsf{a} \in \mathbb{V} \wedge \sigma \in S^\omega\}$ $\qquad\qquad$ ⁋def. α and @⁋

$= \quad \{\langle \mathsf{v} \, \mathsf{b}_0, \, \mathsf{v} \, \mathsf{b}_1 \rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{b}_0, \, \mathsf{b}_1 \rangle \in \alpha(S^\omega)\}$ $\qquad$ ⁋def. α and $S^\omega \subseteq \mathbb{T}^\omega$⁋

— $\quad \alpha(\{(\mathsf{a} \, @ \, \sigma) \bullet (\mathsf{a} \, \mathsf{v}) \bullet \sigma' \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \sigma \bullet \mathsf{v} \in S^+ \wedge (\mathsf{a} \, \mathsf{v}) \bullet \sigma' \in S\})$

$\subseteq \quad \alpha(\{(\mathsf{a} \, @ \, \sigma) \bullet (\mathsf{a} \, \mathsf{v}) \bullet \sigma' \mid \mathsf{a}, \mathsf{v} \in \mathbb{V} \wedge \sigma \bullet \mathsf{v} \in S^+\})$

$\qquad\qquad\qquad\qquad\qquad$ ⁋ignoring that $(\mathsf{a} \, \mathsf{v})$ might "go wrong"⁋

$= \quad \{\langle \mathsf{v} \, \mathsf{b}_0, \, \mathsf{v} \, \mathsf{b}_1 \rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{b}_0, \, \mathsf{b}_1 \rangle \in \alpha(S^+)\}$ $\qquad$ ⁋def. α and $S^+ \subseteq \mathbb{T}^+$⁋

and so $\alpha(\vec{F}(S)) \subseteq f(\alpha(S))$ by defining f as in (6), proving that $\alpha \circ \vec{F} \circ \gamma \mathrel{\dot{\subseteq}} f$ since f is $\subseteq$-monotone, $\alpha \circ \gamma$ is reductive, and by pointwise definition of $\dot{\subseteq}$.

— $\quad$ We have $\alpha \circ \vec{F} \circ \gamma \mathrel{\dot{\subsetneq}} f$ since a single transition cannot anticipate whether the future computation can "go wrong". For example $((\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{x} \, 0) \, 0) \longrightarrow (0 \, 0) \in f \circ f(\varnothing)$ while $((\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{x} \, 0) \, 0) \longrightarrow (0 \, 0) \notin \alpha \circ \vec{F} \circ \gamma \circ \alpha \circ \vec{F} \circ \gamma(\varnothing)$ since there is no trace of the form $\sigma \bullet ((\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{x} \, 0) \, 0) \bullet (0 \, 0) \bullet \sigma'$ in $\vec{F} \circ \gamma \circ \alpha \circ \vec{F} \circ \gamma(\varnothing)$. $\blacksquare$

It follows that the small-step operational semantics or transition semantics $\mathbb{S}$ is sound but incomplete in that the set $\gamma(\mathbb{S})$ of maximal traces generated by the transition relation $\mathbb{S}$ includes the bifinitary trace semantics $\vec{S}$ plus spurious traces for computations that can "go wrong" that is terminate with a runtime error $\mathsf{e} \in \mathbb{E}$. Indeed, the transition semantics $\mathbb{S}$ is an α-overapproximation of the bifinitary trace semantics $\vec{S}$.

Theorem 26 $\vec{S} \subsetneq \gamma(\mathbb{S})$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

PROOF We prove $\vec{S}^+ \subseteq \gamma(\mathbb{S})$ and $\vec{S}^\omega \subseteq \gamma(\mathbb{S})$ so $\vec{S} = \vec{S}^+ \cup \vec{S}^\omega \subseteq \gamma(\mathbb{S})$.

— $\quad \alpha \circ \vec{F} \mathrel{\dot{\subseteq}} f \circ \alpha$ $\qquad\qquad\qquad\qquad\qquad\qquad$ ⁋by Lem. 25⁋

$\Longrightarrow \alpha \circ (\vec{F}(X)^+ \cup \vec{F}(X)^\omega) \subseteq f \circ \alpha(X)$ $\qquad\qquad\qquad$ ⁋partitionning⁋

$\Longrightarrow \alpha \circ \vec{F}(X)^+ \subseteq f \circ \alpha(X)$ $\qquad$ ⁋α is monotone by (5) in Sect. 6.5.1⁋

$\Longrightarrow \alpha \circ \vec{F}^+(X) \subseteq f \circ \alpha(X)$ $\qquad\qquad$ ⁋def. $\vec{F}^+(X) = \vec{F}(X^+)^+ = \vec{F}(X)^+$⁋

$\Longrightarrow \alpha(\mathbf{lfp}^{\subseteq} \vec{F}^+) \subseteq \mathbf{lfp}^{\subseteq} f$ $\qquad\qquad\qquad\qquad$ ⁋[7, Th. 7.1.0.4.(2)]⁋

$\Longrightarrow \vec{S}^+ \subseteq \gamma(\mathbb{S})$ $\qquad\qquad\qquad$ ⁋def. $\vec{S}^+$, $\mathbb{S}$ and (5) in Sect. 6.5.1.⁋

— $\quad \alpha \circ \vec{F} \mathrel{\dot{\subseteq}} f \circ \alpha$ $\qquad\qquad\qquad\qquad\qquad\qquad$ ⁋by Lem. 25⁋

$\Longrightarrow \alpha \circ (\vec{F}(X)^+ \cup \vec{F}(X)^\omega) \subseteq f \circ \alpha(X)$ $\qquad\qquad\qquad$ ⁋partitionning⁋

$\Longrightarrow \alpha \circ \vec{F}(X)^\omega \subseteq f \circ \alpha(X)$ $\qquad$ ⁋α is monotone by (5) in Sect. 6.5.1⁋

$\Longrightarrow \alpha((\vec{F}(\vec{S}^+ \cup S^\omega)^\omega) \subseteq f \circ \alpha(\vec{S}^+ \cup S^\omega)$ $\qquad$ ⁋for $X = (\vec{S}^+ \cup S^\omega)$⁋

$\Longrightarrow \alpha \circ \vec{F}^\omega(S^\omega) \subseteq f \circ \alpha(S^\omega)$

$$\langle\text{since } \vec{F}^\omega(S) \triangleq (\vec{F}(\vec{\mathbb{S}}^+ \cup S^\omega))^\omega \text{ and } \alpha \text{ is monotone by (5) in Sect. 6.5.1}\rangle$$

$$\implies \alpha \circ \vec{F}^\omega \circ \gamma^\omega(X) \subseteq f \circ \alpha \circ \gamma^\omega(X) \subseteq f(X)$$

$$\langle\text{for } S^\omega = \gamma^\omega(X) \text{ and } f \text{ monotone}\rangle$$

$$\implies \vec{F}^\omega \circ \gamma^\omega \ \dot{\subseteq}\ \gamma^\omega \circ f \qquad \langle\text{by (5) in Sect. 6.5.1restricted to infinite traces}\rangle$$

$$\implies \mathbf{gfp}^{\subseteq}\ \vec{F}^\omega \subseteq \gamma^\omega(\mathbf{lfp}^{\subseteq}\ f) \quad \langle\text{dual of [7, Th. 7.1.0.4.(2)] and (5) in Sect. 6.5.1}\rangle$$

$$\implies \vec{\mathbb{S}}^\omega \subseteq \gamma(\mathbb{S}) \qquad\qquad\qquad \langle\text{def. } \vec{\mathbb{S}}^\omega,\ \mathbb{S} \text{ and } \gamma^\omega \ \dot{\subseteq}\ \gamma.\rangle$$

The strict inclusion follows from spurious traces for computations that can "go wrong".   ∎

The inductive definition of $\mathbb{S}$ can also be understood as co-inductive since
Theorem 27 $\mathit{lfp}^{\subseteq}\ f\ =\ \mathit{gfp}^{\subseteq}\ f$.   □

PROOF The iterates $F^\delta, \delta \leqslant \omega$ of $\mathbf{lfp}^{\subseteq}\ f$ are (we write $\mathsf{a} \longrightarrow \mathsf{b}$ for the pair $\langle \mathsf{a},\ \mathsf{b}\rangle$)

— $\ F^0 = \varnothing$

— $\ F^1 = f(F^0) = \{((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v}) \longrightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\}$

— $\ F^n = \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n\}$

   $\langle\text{ind. hyp., where } \mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \text{ is assumed to be parenthesized so that}$
   $((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v}) \text{ is the leftmost reducible term}\rangle$

— $\ F^{n+1} = f(F^n)$

$=\quad \{((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v}) \longrightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\}\ \cup$
   $\{\mathsf{a}_0\ \mathsf{b} \longrightarrow \mathsf{a}_1\ \mathsf{b} \mid \mathsf{a}_0 \longrightarrow \mathsf{a}_1 \in \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant$
   $i + j < n\}\}\ \cup$
   $\{\mathsf{v}\ \mathsf{b}_0 \longrightarrow \mathsf{v}\ \mathsf{b}_1 \mid \mathsf{b}_0 \longrightarrow \mathsf{b}_1 \in \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant$
   $i + j < n\}\}$

$\langle\text{def. } \vec{F}\rangle$

$=\quad \{((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v}) \longrightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\}\ \cup$
   $\{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^{j+1} \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^{j+1} \mid 0 \leqslant i + j < n\}\ \cup$
   $\{\mathsf{v}^{i+1}\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n\}$

$\langle\text{def. } \in\rangle$

$=\quad \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n + 1\}$

— $\ F^\omega = \bigcup_{n \in \mathbb{N}} F^n$

$=\quad \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x} \cdot \mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid i, j \in \mathbb{N}\}$

The iterates $G^\delta, \delta \leqslant \omega$ of $\mathbf{gfp}^{\subseteq}\ f$ are

— $G^0 = \{\mathsf{y} \longrightarrow \mathsf{z}\}$ ⟨for any $\mathsf{y}, \mathsf{z} \in \mathbb{T}$⟩

— $G^1 = f(G^0) = \{(\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v} \longrightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\} \cup \{\mathsf{y}\ \mathsf{b} \longrightarrow \mathsf{z}\ \mathsf{b}\} \cup \{\mathsf{v}\ \mathsf{y} \longrightarrow \mathsf{v}\ \mathsf{z}\}$

— $G^n = \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n\} \cup$
  $\{\mathsf{v}^i\ \mathsf{y}\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{z}\ \mathsf{b}^j \mid i + j = n\}$ ⟨ind. hyp.⟩

— $G^{n+1} = f(G^n)$

$=$ $\{(\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v} \longrightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\} \cup$
  $\{\mathsf{a}_0\ \mathsf{b} \longrightarrow \mathsf{a}_1\ \mathsf{b} \mid \mathsf{a}_0 \longrightarrow \mathsf{a}_1 \in \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n\}\} \cup$
  $\{\mathsf{v}\ \mathsf{b}_0 \longrightarrow \mathsf{v}\ \mathsf{b}_1 \mid \mathsf{b}_0 \longrightarrow \mathsf{b}_1 \in \{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n\}\} \cup$
  $\{\mathsf{a}_0\ \mathsf{b} \longrightarrow \mathsf{a}_1\ \mathsf{b} \mid \mathsf{a}_0 \longrightarrow \mathsf{a}_1 \in \{\mathsf{v}^i\ \mathsf{y}\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{z}\ \mathsf{b}^j \mid i + j = n\}\} \cup$
  $\{\mathsf{v}\ \mathsf{b}_0 \longrightarrow \mathsf{v}\ \mathsf{b}_1 \mid \mathsf{b}_0 \longrightarrow \mathsf{b}_1 \in \{\mathsf{v}^i\ \mathsf{y}\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{z}\ \mathsf{b}^j \mid i + j = n\}\}$
  ⟨def. f and G^n⟩

$=$ $\{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid 0 \leqslant i + j < n + 1\} \cup$
  $\{\mathsf{v}^i\ \mathsf{y}\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{z}\ \mathsf{b}^j \mid i + j = n + 1\}$

— $G^\omega = \bigcap_{n \in \mathbb{N}} G^n$

$=$ $\{\mathsf{v}^i\ ((\boldsymbol{\lambda}\mathsf{x}\cdot\mathsf{a})\ \mathsf{v})\ \mathsf{b}^j \longrightarrow \mathsf{v}^i\ \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]\ \mathsf{b}^j \mid i, j \in \mathbb{N}\}$

proving that $\mathbf{lfp}^{\subseteq} f = F^\omega = G^\omega = \mathbf{gfp}^{\subseteq} f$. ∎

6.6 Small-step maximal trace semantics of the call-by-value λ-calculus

Coming back to the small-step maximal trace semantics $\overset{\infty}{\longrightarrow}$ of a transition relation $\longrightarrow$ considered in Sect. 4, let us define

$\overset{n}{\twoheadrightarrow} \triangleq \{\sigma \in \mathbb{T}^+ \mid |\sigma| = n > 0 \wedge \forall i : 0 \leqslant i < n - 1 : \sigma_i \longrightarrow \sigma_{i+1}\}$ partial traces

$\overset{n}{\longrightarrow} \triangleq \{\sigma \in \overset{n}{\twoheadrightarrow} \mid \sigma_{n-1} \in \mathbb{V}\}$ maximal execution traces of length n

$\overset{+}{\longrightarrow} \triangleq \bigcup_{n > 0} \overset{n}{\longrightarrow}$ maximal finite execution traces

$\overset{\omega}{\longrightarrow} \triangleq \{\sigma \in \mathbb{T}^\omega \mid \forall i \in \mathbb{N} : \sigma_i \longrightarrow \sigma_{i+1}\}$ infinite execution traces

$\overset{\infty}{\longrightarrow} \triangleq \overset{+}{\longrightarrow} \cup \overset{\omega}{\longrightarrow}$ maximal finite and diverging execution traces.

6.6.1 Fixpoint small-step maximal trace semantics

To express the small-step maximal trace semantics $\overset{\infty}{\longrightarrow}$ in fixpoint form, let us define the junction $\mathbin{\fatsemi}$ of set of traces as

$$S \, \S \, T \triangleq S^\omega \cup \{\sigma_0 \bullet \ldots \bullet \sigma_{|\sigma|-2} \bullet \sigma' \mid \sigma \in S^+ \wedge \sigma_{|\sigma|-1} = \sigma_0' \wedge \sigma' \in T\} \, ,$$

and the $\sqsubseteq$-monotone small-step set of traces transformer $\vec{f} \in \wp(\overline{\mathbb{T}}^\infty) \longrightarrow \wp(\overline{\mathbb{T}}^\infty)$

$$\vec{f}(T) \triangleq \{\mathsf{v} \in \overline{\mathbb{T}}^\infty \mid \mathsf{v} \in \mathbb{V}\} \cup \xrightarrow{2} \S \, T \tag{7}$$

describing small steps of computation.

Lemma 28 *We have*

$$\xrightarrow{\infty} \, = \, \mathit{lfp}^{\sqsubseteq} \, \vec{f} \, = \, \mathit{gfp}^{\subseteq} \, \vec{f} \, .$$

PROOF By [6, **Th. 13**]. ∎

Theorem 29 *The big-step and small-step trace semantics are the same*

$$\vec{\mathbb{S}} = \xrightarrow{\infty} \, .$$

PROOF —— We first prove that $\vec{f}(\vec{\mathbb{S}})^+ \subseteq \vec{\mathbb{S}}^+$. By definition (7) of $\vec{f}$, $\vec{\mathbb{S}}^+ = \vec{F}(\vec{\mathbb{S}})^+ = \vec{F}(\vec{\mathbb{S}}^+)$ and definition of $\vec{F}$ by (a)—(f) in Sect. 6.3.1, we must prove that $\xrightarrow{2} \S \, \vec{\mathbb{S}}^+ \subseteq \vec{\mathbb{S}}^+$ that is $\mathsf{a} \rightarrow \sigma_0 \wedge \sigma \in \vec{\mathbb{S}}^+ \Longrightarrow \mathsf{a} \bullet \sigma \in \vec{\mathbb{S}}^+$. By (6) in Sect. 6.5.2, we proceed by structural induction on a.

—— If $\mathsf{a} \rightarrow \sigma_0 = (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}') \, \mathsf{v} \rightarrow \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}]$ then $\mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{\mathbb{S}}^+$ implies $\mathsf{a} \bullet \sigma = (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}') \, \mathsf{v} \bullet \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}] \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{F}(\vec{\mathbb{S}}^+) = \vec{\mathbb{S}}^+$ by (b).

—— If $\mathsf{a} \rightarrow \sigma_0 = \mathsf{a}_0 \, \mathsf{b} \rightarrow \mathsf{a}_1 \, \mathsf{b}$ where $\mathsf{a}_0 \rightarrow \mathsf{a}_1$ so $\mathsf{a}_0 \notin \mathbb{V}$ and $\sigma \in \vec{\mathbb{S}}^+ = \vec{F}(\vec{\mathbb{S}}^+)$. By (a)—(f), there are 3 cases for $\mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots$.

—— In case (b), $\mathsf{a}_1 \, \mathsf{b} = (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}) \, \mathsf{v}$. We have $\mathsf{a}_0 \rightarrow \mathsf{a}_1$ and $\mathsf{a}_1 = \boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}$ so, by (6) in Sect. 6.5.2, $\mathsf{a}_0 = (\boldsymbol{\lambda}\mathsf{y} \bullet \mathsf{a}') \, \mathsf{v}'$ and $\mathsf{a}_1 = \mathsf{a}'[\mathsf{y} \leftarrow \mathsf{v}'] = \boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}$. Since $\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a} \in \mathbb{V}$ we have $\mathsf{a}'[\mathsf{y} \leftarrow \mathsf{v}'] = \boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a} \in \vec{\mathbb{S}}$ by (a) so $(\boldsymbol{\lambda}\mathsf{y} \bullet \mathsf{a}') \, \mathsf{v}' \bullet \mathsf{a}'[\mathsf{y} \leftarrow \mathsf{v}'] \in \vec{\mathbb{S}}$ by (b), that is $\mathsf{a}_0 \bullet \mathsf{a}_1 \in \vec{\mathbb{S}}$. By (d), $\mathsf{a}_0 \bullet \mathsf{a}_1 \in \vec{\mathbb{S}}$, $\mathsf{a}_1 \in \mathbb{V}$, and $\sigma = \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{\mathbb{S}}$ imply that $\mathsf{a} \bullet \sigma = \mathsf{a}_0 \, \mathsf{b} \bullet \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{F}(\vec{\mathbb{S}}^+) = \vec{\mathbb{S}}^+$.

—— In case (d), $\sigma = \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots = (\sigma' \, @ \, \mathsf{b}) \bullet (\mathsf{v}' \, \mathsf{b}) \bullet \sigma''$ where $\sigma' \neq \epsilon$, $\mathsf{a}_1 \bullet \sigma_1' \bullet \sigma_2' \bullet \ldots = \sigma' \bullet \mathsf{v}' \in \vec{\mathbb{S}}^+$, $\mathsf{v}' \in \mathbb{V}$, and $(\mathsf{v}' \, \mathsf{b}) \bullet \sigma'' \in \vec{\mathbb{S}}^+$. By induction hypothesis, $\mathsf{a}_0 \rightarrow \mathsf{a}_1$ and $\mathsf{a}_1 \bullet \sigma_1' \bullet \sigma_2' \bullet \ldots \in \vec{\mathbb{S}}^+$ imply that $\mathsf{a}_0 \bullet \mathsf{a}_1 \bullet \sigma_1' \bullet \sigma_2' \bullet \ldots = \mathsf{a}_0 \bullet \sigma' \bullet \mathsf{v}' \in \vec{\mathbb{S}}^+$ hence, by (d), $\mathsf{a} \bullet \sigma = (\mathsf{a}_0 \, \mathsf{b}) \bullet (\sigma' \, @ \, \mathsf{b}) \bullet (\mathsf{v}' \, \mathsf{b}) \bullet \sigma'' = ((\mathsf{a}_0 \bullet \sigma') \, @ \, \mathsf{b}) \bullet (\mathsf{v}' \, \mathsf{b}) \bullet \sigma'' \in \vec{\mathbb{S}}^+$.

—— In case (f), $\sigma = \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots = (\mathsf{a}' @ \sigma') \bullet (\mathsf{a}' \, \mathsf{v}') \bullet \sigma''$ where $\mathsf{a}_1 = \mathsf{a}'$, $\mathsf{v}' \in \mathbb{V}$, $\sigma' \neq \epsilon$, $\sigma' \bullet \mathsf{v}' \in \vec{\mathbb{S}}^+$, and $(\mathsf{a}' \, \mathsf{v}') \bullet \sigma'' \in \vec{\mathbb{S}}^+$. By (6) in Sect. 6.5.2, $\mathsf{a}_0 \rightarrow \mathsf{a}_1$ and $\mathsf{a}_1 \in \mathbb{V}$ imply $\mathsf{a}_0 = (\boldsymbol{\lambda}\mathsf{y} \bullet \mathsf{c}) \, \mathsf{w}$ and $\mathsf{a}_1 = \mathsf{c}[\mathsf{y} \leftarrow \mathsf{w}]$. Hence, by (a) and (b), $\mathsf{a}_0 \bullet \mathsf{a}_1 \in \vec{\mathbb{S}}^+$. Then, by (d), $\mathsf{a}_0 \bullet \mathsf{a}_1 \neq \epsilon$, $\mathsf{a}_0 \bullet \mathsf{a}_1 \in \vec{\mathbb{S}}^+$, $\mathsf{a}_1 \in \mathbb{V}$, and $\sigma = \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{\mathbb{S}}^+$ imply $\mathsf{a} \bullet \sigma = \mathsf{a}_0 \, \mathsf{b} \bullet \sigma = \mathsf{a}_0 \, \mathsf{b} \bullet \mathsf{a}_1 \, \mathsf{b} \bullet \sigma_1 \bullet \sigma_2 \bullet \ldots \in \vec{\mathbb{S}}^+$.

— If $a \twoheadrightarrow \sigma_0 = v\ b_0 \twoheadrightarrow v\ b_1$ where $v \in \mathbb{V}$ and $b_0 \twoheadrightarrow b_1$ then $b_0 \notin \mathbb{V}$ so, by (a)—(f), there are 3 cases for $v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots$.

   — If, by (b), $v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots = (\boldsymbol{\lambda} x \bullet a')\ v' \bullet a'[x \leftarrow v'] \bullet \sigma_2 \bullet \dots$ then $b_1 = v' \in \mathbb{V}$ so $b_1 \in \vec{\mathbb{S}}^+$ hence $b_0 \twoheadrightarrow b_1$ implies, by induction hypothesis, that $b_0 \bullet b_1 \in \vec{\mathbb{S}}^+$. By $b_0 \bullet b_1 \in \vec{\mathbb{S}}$, $b_1 \in \mathbb{V}$, and $v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots \in \vec{\mathbb{S}}^+$, we conclude, by (f), that $a \bullet \sigma = v\ b_0 \bullet v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots \in \vec{\mathbb{S}}^+$

   — The case where, by (d), $v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots = (\sigma' @ b') \bullet (v'\ b') \bullet \sigma''$, with $\sigma' \neq \epsilon$, $\sigma' \bullet v' \in \vec{\mathbb{S}}^+$, $v' \in \mathbb{V}$, and $(v'\ b') \bullet \sigma'' \in \vec{\mathbb{S}}^+$ would have $\sigma'_0 = v$ hence $v \bullet \sigma'_1 \bullet \dots v' \in \vec{\mathbb{S}}^+$, which is impossible.

   — If, by (f), $v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots = (a'@\sigma') \bullet (a'\ v') \bullet \sigma''$ where $v = a'$, $v' \in \mathbb{V}$, $\sigma' \neq \epsilon$, $\sigma' \bullet v' \in \vec{\mathbb{S}}^+$, and $(a'\ v') \bullet \sigma'' \in \vec{\mathbb{S}}^+$ then $\sigma'_0 = b_1$ so, by induction hypothesis, $b_1 \bullet \sigma'_1 \bullet \dots \bullet v' \in \vec{\mathbb{S}}^+$ and $b_0 \twoheadrightarrow b_1$ imply $b_0 \bullet b_1 \bullet \sigma'_1 \bullet \dots \bullet v' \in \vec{\mathbb{S}}^+$ so, $a \bullet \sigma = v\ b_0 \bullet v\ b_1 \bullet \sigma_1 \bullet \sigma_2 \bullet \dots = v\ b_0 \bullet (a' @ \sigma') \bullet (a'\ v') \bullet \sigma'' = a'\ b_0 \bullet (a' @ \sigma') \bullet (a'\ v') \bullet \sigma'' = (a' @ (b_0 \bullet \sigma')) \bullet (a'\ v') \bullet \sigma'' \in \vec{\mathbb{S}}^+$, by (f).

— To prove that $\vec{f}(\vec{\mathbb{S}})^\omega \supseteq \vec{\mathbb{S}}^\omega$, we must, by definition (7) of $\vec{f}$, prove that $\sigma \in \vec{\mathbb{S}}^\omega$ implies $\sigma \in \xrightarrow{2}\ \mathbin{\fgebreak} \vec{\mathbb{S}}^\omega$, that is $\sigma_0 \twoheadrightarrow \sigma_1$ and $\sigma_1 \bullet \sigma_2 \bullet \dots \in \vec{\mathbb{S}}^\omega$.

But $\sigma \in \vec{\mathbb{S}}^\omega \implies \sigma_0 \twoheadrightarrow \sigma_1$ is equivalent to $\vec{\mathbb{S}}^\omega \subseteq \gamma(\twoheadrightarrow)$ that is $\mathbf{gfp}^\subseteq \vec{F} \subseteq \gamma(\mathbf{gfp}^\subseteq f)$ which, by the dual of [6, **Th. 1**], follows from $\vec{F} \circ \gamma \mathbin{\dot{\subseteq}} \gamma \circ f$ or equivalently, $\alpha \circ \vec{F} \circ \gamma \mathbin{\dot{\subseteq}} f$.

Moreover $\vec{\mathbb{S}}$ hence $\vec{\mathbb{S}}^\omega$ is suffix closed and therefore $\sigma_1 \bullet \sigma_2 \bullet \dots \in \vec{\mathbb{S}}^\omega$.

— We have $\vec{f}(\vec{\mathbb{S}})^+ \subseteq \vec{\mathbb{S}}^+$ and $\vec{f}(\vec{\mathbb{S}})^\omega \supseteq \vec{\mathbb{S}}^\omega$ so $\vec{f}(\vec{\mathbb{S}}) \sqsubseteq \vec{\mathbb{S}}$, proving, by Tarski's fixpoint theorem [33] for the $\sqsubseteq$-monotone $\vec{f}$ on the complete lattice $\langle \wp(\overline{\mathbb{T}}^\infty), \sqsubseteq \rangle$, that $\mathbf{lfp}^\sqsubseteq \vec{f} \sqsubseteq \vec{\mathbb{S}}$ hence $\xrightarrow{\infty} \sqsubseteq \vec{\mathbb{S}}$.

— We now prove that $\vec{F}(\xrightarrow{+}) \subseteq \xrightarrow{+}$ that is $\forall \sigma \in \vec{F}(\xrightarrow{+}) : \sigma \in \xrightarrow{+}$. If $|\sigma| = 1$ then $\sigma = \sigma_0 \in \mathbb{V}$ so $\sigma \in \xrightarrow{1} \subseteq \xrightarrow{+}$. Otherwise $|\sigma| > 1$ and we proceed by case analysis on the syntax of σ_0.

   — If $\sigma = (\boldsymbol{\lambda} x \bullet a)\ v \bullet a[x \leftarrow v] \bullet \sigma' \in \vec{F}(\xrightarrow{+})$, $v \in \mathbb{V}$ then $a[x \leftarrow v] \bullet \sigma' \in \xrightarrow{+}$ by (b) and $(\boldsymbol{\lambda} x \bullet a) \twoheadrightarrow a[x \leftarrow v]$ by (6) in Sect. 6.5.2 so $\sigma = (\boldsymbol{\lambda} x \bullet a)\ v \bullet a[x \leftarrow v] \bullet \sigma' \in \xrightarrow{+}$ by definition of $\xrightarrow{+}$.

   — If $\sigma = (\sigma' @ b) \bullet (v\ b) \bullet \sigma'' \in \vec{F}(\xrightarrow{+})$, where by (d), $\sigma' \neq \epsilon$, $\sigma' \bullet v \in \xrightarrow{+}$, $v \in \mathbb{V}$, and $(v\ b) \bullet \sigma'' \in \xrightarrow{+}$ then, by definition of $\xrightarrow{+}$, $\sigma'_0 \twoheadrightarrow \sigma'_1 \twoheadrightarrow \dots \twoheadrightarrow \sigma'_{|\sigma'|-1} \twoheadrightarrow v$ and $(v\ b) \twoheadrightarrow \sigma''_0 \twoheadrightarrow \sigma''_1 \twoheadrightarrow \dots \twoheadrightarrow \sigma''_{|\sigma''|-1}$, and so by (6) in Sect. 6.5.2, $(\sigma'_0\ b) \twoheadrightarrow (\sigma'_1\ b) \twoheadrightarrow \dots \twoheadrightarrow (\sigma'_{|\sigma'|-1}\ b) \twoheadrightarrow (v\ b)$ hence $\sigma = (\sigma'_0\ b) \bullet (\sigma'_1\ b) \bullet \dots \bullet (\sigma'_{|\sigma'|-1}\ b) \bullet (v\ b) \bullet \sigma''_0 \bullet \sigma''_1 \bullet \dots \bullet \sigma''_{|\sigma''|-1} \in \xrightarrow{+}$ by definition of $\xrightarrow{+}$.

   — If $\sigma = (a @ \sigma') \bullet (a\ v) \bullet \sigma'' \in \vec{F}(\xrightarrow{+})$, where by (f), $a, v \in \mathbb{V}$, $\sigma' \neq \epsilon$, $\sigma' \bullet v \in \xrightarrow{+}$, $(a\ v) \bullet \sigma'' \in \xrightarrow{+}$ then, by definition of $\xrightarrow{+}$, $\sigma'_0 \twoheadrightarrow \dots \twoheadrightarrow \sigma'_{|\sigma'|-1} \twoheadrightarrow v$ and $(a\ v) \twoheadrightarrow \sigma''_0 \twoheadrightarrow \dots \twoheadrightarrow \sigma''_{|\sigma''|-1}$, and so by (6) in Sect. 6.5.2,

(a σ_0') $\rightarrow$ ... $\rightarrow$ (a $\sigma_{|\sigma'|-1}'$) $\rightarrow$ (a v) proving, by definition of $\xrightarrow{+}$, that
$\sigma = $ (a σ_0') $\bullet$ .... $\bullet$ (a $\sigma_{|\sigma'|-1}'$) $\bullet$ (a v) $\bullet$ σ_0'' $\bullet$ .... $\bullet$ $\sigma_{|\sigma''|-1}''$ $\in \xrightarrow{+}$.

— Next, we prove that $\xrightarrow{\omega} \subseteq \vec{F}(\xrightarrow{\omega})$. If $\sigma \in \xrightarrow{\omega}$ then $\sigma_0 \notin \mathbb{V}$ and $\sigma_0 \rightarrow$
$\sigma_1 ... \rightarrow \sigma_n \rightarrow$ .... By (6) in Sect. 6.5.2, there are three cases.

— If $\sigma_0 \rightarrow \sigma_1 = (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a})$ v $\rightarrow \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]$ then $\mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \bullet$ .... $\bullet \sigma_n \bullet$ ... $\in \xrightarrow{\omega}$
so $\sigma = (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}) \bullet \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}] \bullet$ .... $\bullet \sigma_n \bullet$ ... $\in \vec{F}(\xrightarrow{\omega})$ by (b).
— If $\sigma_0 \rightarrow \sigma_1 = \mathsf{a}_0$ b $\rightarrow \mathsf{a}_1$ b where $\mathsf{a}_0 \rightarrow \mathsf{a}_1$, then there are two cases.
  — Either all σ_i, $i \in N$ are of the form a_i b in which case, by (6) in Sect. 6.5.2,
  $\mathsf{a}_0 \rightarrow \mathsf{a}_1 \rightarrow$ ... $\rightarrow \mathsf{a}_n \rightarrow$ ... hence, by definition of $\xrightarrow{\omega}$, $\mathsf{a}_0 \bullet \mathsf{a}_1 \bullet$ .... $\bullet \mathsf{a}_n \bullet$ ... $\in$
  $\xrightarrow{\omega}$, proving, by (c), that $\sigma = (\mathsf{a}_0$ b$) \bullet (\mathsf{a}_1$ b$) \bullet$ .... $\bullet (\mathsf{a}_n$ b$) \bullet$ ... $\in \vec{F}(\xrightarrow{\omega})$.
  — Or $\sigma = (\mathsf{a}_0$ b$) \bullet$ .... $\bullet (\mathsf{a}_{i-1}$ b$) \bullet \sigma_i \bullet \sigma i + 1 \bullet$ ... and σ_i is not of the form
  $(\mathsf{a}_i$ b$)$. $(\mathsf{a}_0$ b$) \rightarrow$ ... $\rightarrow (\mathsf{a}_{i-1}$ b$) \rightarrow \sigma_i$ implies $\mathsf{a}_0 \rightarrow \mathsf{a}_1 \rightarrow$ ... $\rightarrow \mathsf{a}_{i-1}$,
  by (6) in Sect. 6.5.2. Since σ_i is not of the form $(\mathsf{a}_i$ b$)$, there are, according
  to (6) in Sect. 6.5.2, only two possibe cases for $(\mathsf{a}_{i-1}$ b$) \rightarrow \sigma_i$.
    – Either $((\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a})$ v$) = (\mathsf{a}_{i-1}$ b$) \rightarrow \sigma_i = \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}]$, or
    – $(\mathsf{v}$ b$) = (\mathsf{a}_{i-1}$ b$) \rightarrow \sigma_i = (\mathsf{v}$ b$_1)$ where $\mathsf{v} \in \mathbb{V}$ and $\mathsf{b}_0 \rightarrow \mathsf{b}_1$.
  In both cases $\mathsf{a}_{i-1} \in \mathbb{V}$, so $\mathsf{a}_{i-1} \not\rightarrow$ hence $\mathsf{a}_0 \bullet \mathsf{a}_1 \bullet$ .... $\bullet \mathsf{a}_{i-1} \in \xrightarrow{+}$, and
  $(\mathsf{a}_{i-1}$ b$) \bullet \sigma_i \bullet \sigma_{i+1} \bullet$ ... $\in \xrightarrow{\omega}$ so $\sigma = (\mathsf{a}_0$ b$) \bullet (\mathsf{a}_1$ b$) \bullet$ .... $\bullet (\mathsf{a}_{i-1}$ b$) \bullet \sigma_i \bullet \sigma_{i+1} \bullet$ ... $\in$
  $\vec{F}(\xrightarrow{\omega})$ by (d).

— Otherwise $\sigma_0 \rightarrow \sigma_1 = \mathsf{v}$ b$_0 \rightarrow \mathsf{v}$ b$_1$ where $\mathsf{b}_0 \rightarrow \mathsf{b}_1$ and there are two
cases.
  — Either $\forall i \in N : \sigma_i = (\mathsf{v}$ b$_i)$ hence, by (6) in Sect. 6.5.2, $\mathsf{b}_0 \rightarrow \mathsf{b}_1 \rightarrow$
  ... $\rightarrow \mathsf{b}_n \rightarrow$ ... so $\mathsf{b}_0 \bullet \mathsf{b}_1 \bullet$ .... $\bullet \mathsf{b}_n \bullet$ ... $\in \xrightarrow{\omega}$ proving that $\sigma = (\mathsf{v}$ b$_0) \bullet$
  $(\mathsf{v}$ b$_1) \bullet$ .... $\bullet (\mathsf{v}$ b$_n) \bullet$ ... $\in \vec{F}(\xrightarrow{\omega})$ by (e).
  — Or $\sigma = (\mathsf{v}$ b$_0) \bullet$ .... $\bullet (\mathsf{v}$ b$_{i-1}) \bullet \sigma_i \bullet$ ... where σ_i is not of the form $(\mathsf{v}$ b$_i)$.
  By (6) in Sect. 6.5.2, $(\mathsf{v}$ b$_{i-1}) \rightarrow \sigma_i$ and σ_i is not of the form $(\mathsf{v}$ b$_i)$ imply
  that $((\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a}')$ v$') = (\mathsf{v}$ b$_{i-1}) \rightarrow \sigma_i = \mathsf{a}'[\mathsf{x} \leftarrow \mathsf{v}']$ so $\mathsf{b}_{i-1} = \mathsf{v}' \in \mathbb{V}$. Therefore
  $\mathsf{v} \in \mathbb{V}$, $\mathsf{b}_0 \bullet$ .... $\bullet \mathsf{b}_{i-1} \in \xrightarrow{+}$ since $\mathsf{b}_{i-1} \in \mathbb{V}$, and $(\mathsf{v}$ b$_{i-1}) \bullet \sigma_i \bullet$ ... $\in \xrightarrow{\omega}$ by
  definition of $\xrightarrow{\omega}$ imply that $\sigma = (\mathsf{v}$ b$_0) \bullet$ .... $\bullet (\mathsf{v}$ b$_{i-1}) \bullet \sigma_i \bullet$ ... $\in \vec{F}(\xrightarrow{\omega})$ by
  (f).

— We have $\vec{F}(\xrightarrow{+}) \subseteq \xrightarrow{+}$ so $\vec{F}^+(\xrightarrow{+}) \triangleq \vec{F}((\xrightarrow{+})^+)^+ = \vec{F}(\xrightarrow{+})^+ \subseteq (\xrightarrow{+})^+ =$
$\xrightarrow{+}$ and $\vec{F}^+$ is $\subseteq$-monotone on the complete lattice $\langle \wp(\mathbb{T}^+), \subseteq \rangle$ so $\vec{\mathbb{S}}^+ \triangleq$
$\mathbf{lfp}^{\subseteq} \vec{F}^+ = \bigcap\{X \subseteq \mathbb{T}^+ \mid \vec{F}^+(X) \subseteq X\} \subseteq \xrightarrow{+}$ by Tarski's fixpoint theorem [33].

Moreover by $\subseteq$-monotony of $\vec{F}$ and $\vec{F}(\xrightarrow{\omega}) \supseteq \xrightarrow{\omega}$, we have $\vec{F}^\omega(\xrightarrow{\omega}) = \vec{F}^\omega(S) \triangleq$
$(\vec{F}(\vec{\mathbb{S}}^+ \cup (\xrightarrow{\omega})^\omega))^\omega = (\vec{F}(\vec{\mathbb{S}}^+ \cup \xrightarrow{\omega}))^\omega \supseteq (\vec{F}(\xrightarrow{\omega}))^\omega \supseteq (\xrightarrow{\omega})^\omega = \xrightarrow{\omega}$ so $\vec{\mathbb{S}}^\omega \triangleq$
$\mathbf{gfp}^{\subseteq} \vec{F}^\omega = \bigcup\{X \subseteq \mathbb{T}^\omega \mid X \subseteq \vec{F}^\omega(X)\} \supseteq \xrightarrow{\omega}$ by Tarski's fixpoint theorem [33]
on the complete lattice $\langle \wp(\mathbb{T}^\omega), \subseteq \rangle$.

In follows that $\vec{\mathbb{S}} \triangleq \vec{\mathbb{S}}^+ \cup \vec{\mathbb{S}}^\omega \sqsubseteq \xrightarrow{+} \cup \xrightarrow{\omega} = \xrightarrow{\infty}$.

— In conclusion, we have $\vec{\mathbb{S}} = \overset{\infty}{\longrightarrow}$ by antisymmetry. ∎

6.6.2 Rule-based small-step maximal trace semantics

The maximal trace semantics $\vec{\mathbb{S}} = \overset{\infty}{\longrightarrow} = \mathbf{lfp}^{\sqsubseteq}\,\vec{f}$ where $\vec{f}$ is defined by (7) in Sect. 6.6.1 can be defined inductively with small-steps as

$$\mathsf{v} \in \vec{\mathbb{S}}, \quad \mathsf{v} \in \mathbb{V} \qquad \dfrac{\mathsf{a} \longrightarrow \mathsf{b}, \quad \mathsf{b} \bullet \sigma \in \vec{\mathbb{S}}}{\mathsf{a} \bullet \mathsf{b} \bullet \sigma \in \vec{\mathbb{S}}} \; \sqsubseteq$$

that is, writing $\mathsf{a} \Longmapsto \sigma$ for $\sigma \in \vec{\mathbb{S}}$ and $\sigma_0 = \mathsf{a}$

$$\mathsf{v} \Longmapsto \mathsf{v}, \quad \mathsf{v} \in \mathbb{V} \qquad \dfrac{\mathsf{a} \longrightarrow \mathsf{b}, \quad \mathsf{b} \Longmapsto \sigma}{\mathsf{a} \Longmapsto \mathsf{a} \bullet \sigma} \; \sqsubseteq$$

6.7 Small-step bifinitary relational semantics of the call-by-value λ-calculus

The bifinitary relational semantics was defined as $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}})$ (where α is the relational abstraction of sets of traces (2) in Sect. 6.4.1) and given in big-step form in *Sect. 6.4*. It can be given in small-step form by abstraction of the small-step bifinitary maximal trace semantics of *Sect. 6.6.1*.

6.7.1 Fixpoint small-step bifinitary relational semantics

The bifinitary relational semantics $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) = \alpha(\mathbf{lfp}^{\sqsubseteq}\,\vec{f})$ can be defined in fixpoint form as $\mathbf{lfp}^{\sqsubseteq}\,\widehat{f}$ where the small-step transformer $\widehat{f} \in \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\})) \longrightarrow \wp(\mathbb{T} \times (\mathbb{T} \cup \{\bot\}))$ is

$$\widehat{f}(R) \triangleq \{\langle \mathsf{v},\, \mathsf{v} \rangle \mid \mathsf{v} \in \mathbb{V}\} \tag{8}$$

$$\cup \{\langle (\boldsymbol{\lambda}\mathsf{x} \bullet \mathsf{a})\,\mathsf{v},\, r \rangle \mid \mathsf{v} \in \mathbb{V} \wedge \langle \mathsf{a}[\mathsf{x} \leftarrow \mathsf{v}],\, r \rangle \in R\}$$

$$\cup \{\langle \mathsf{a}_0\,\mathsf{b},\, r \rangle \mid \mathsf{a}_0 \longrightarrow \mathsf{a}_1 \wedge \langle \mathsf{a}_1\,\mathsf{b},\, r \rangle \in R\}$$

$$\cup \{\langle \mathsf{v}\,\mathsf{b}_0,\, r \rangle \mid \mathsf{b}_0 \longrightarrow \mathsf{b}_1 \wedge \langle \mathsf{v}\,\mathsf{b}_1,\, r \rangle \in R\}\;.$$

PROOF We have

41

$$\alpha(\vec{f}(T))$$

$$= \alpha(\{v \in \overline{\mathbb{T}}^\infty \mid v \in \mathbb{V}\} \cup \xrightarrow{2} \, {}^{\circ}_{9}\, T) \qquad\qquad \wr\text{def. (7) in Sect. 6.6.1 of } \vec{f}\, \wr$$

$$= \alpha(\{v \in \overline{\mathbb{T}}^\infty \mid v \in \mathbb{V}\}) \cup \alpha(\xrightarrow{2} \, {}^{\circ}_{9}\, T) \qquad \wr \alpha \text{ preserves } \cup \text{ by (3) in Sect. 6.4.1} \wr$$

$$= \{\langle v, v\rangle \mid v \in \mathbb{V}\} \cup \alpha(\xrightarrow{2} \, {}^{\circ}_{9}\, T) \qquad\qquad \wr\text{def. (2) of } \alpha \text{ in Sect. 6.4.1} \wr$$

$$= \{\langle v, v\rangle \mid v \in \mathbb{V}\} \cup \{a \bullet b \bullet \sigma \mid a \longrightarrow b \wedge b \bullet \sigma \in T\} \qquad \wr\text{def. } \xrightarrow{2} \text{ and } {}^{\circ}_{9}\wr$$

$$= \{\langle v, v\rangle \mid v \in \mathbb{V}\} \cup \{\langle a, \bot\rangle \mid a \longrightarrow b \wedge b \bullet \sigma \in T^\omega\} \cup$$
$$\quad \{\langle a, r\rangle \mid a \longrightarrow b \wedge b \bullet \sigma \bullet r \in T^+\} \qquad \wr\text{def. (2) of } \alpha \text{ in Sect. 6.4.1} \wr$$

$$= \{\langle v, v\rangle \mid v \in \mathbb{V}\} \cup \{\langle a, r\rangle \mid a \longrightarrow b \wedge \langle b, r\rangle \in \alpha(T)\}$$
$$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wr\text{def. (2) of } \alpha \text{ in Sect. 6.4.1} \wr$$

$$= \widehat{f}(\alpha(T))$$

by defining $\widehat{f}(R) \triangleq \{\langle v, v\rangle \mid v \in \mathbb{V}\} \cup \{\langle a, r\rangle \mid a \longrightarrow b \wedge \langle b, r\rangle \in R\}$. The commutation property $\alpha \circ \vec{f} = \widehat{f} \circ \alpha$ implies that $\widehat{\mathbb{S}} \triangleq \alpha(\vec{\mathbb{S}}) = \alpha(\mathbf{lfp}^{\sqsubseteq} \vec{f}) = \mathbf{lfp}^{\sqsubseteq} \widehat{f}$. Using the fixpoint property (6) in Sect. 6.5.2 of $\mathbb{S}$, we get (8). ∎

6.7.2 Rule-based small-step bifinitary relational semantics

The bifinitary rule-base form is ($a \Rightarrow b$ stands for $\langle a, b\rangle \in \widehat{\mathbb{S}}$ and $r \in \mathbb{V} \cup \{\bot\}$)

$$v \Rightarrow v, \quad v \in \mathbb{V} \qquad \dfrac{a \longrightarrow b, \quad b \Rightarrow r}{a \Rightarrow r}{}_\sqsubseteq$$

7 Related work

Divergence/nonterminating behaviors are needed in static program analysis [25] [10] or typing [5,22]. Such divergence information is part of the classical order-theoretic fixpoint denotational semantics [24] but not explicit in small-step/abstract-machine-based operational semantics [28,29,30] and absent of big-step/natural operational semantics [17]. A standard approach is therefore to generate an execution trace semantics from a (labelled) transition system/small-step operational semantics, using either an order-theoretic [6] or metric [35]
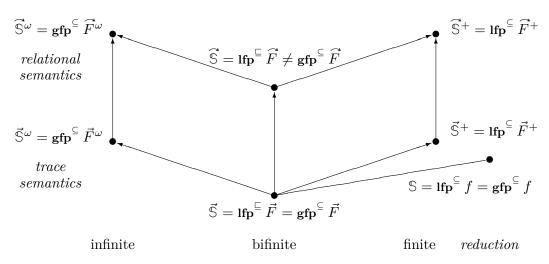
---

[10] For example, the authors of [32] claim that their "work is the first provably correct strictness analysis and call-by-name to call-by-value transformation for an untyped higher-order language" but since the considered big-step semantics does not account for divergence, the considered analysis is not *strictness* but a weaker *needness* analysis.

fixpoint definition or else a categorical definition as a final coalgebra for a behaviour functor (modeling the transition relation) up to a weak bisimulation [16,34,20] or using an equational definition for recursion in an order-enriched category [19]. However, the description of execution traces by small steps may be impractical as compared to a compositional definition using big steps. Moreover, execution traces are not always at an appropriate level of abstraction and relational semantics often look more natural.

8   Conclusion

We have introduced bi-inductive definitions, an order-theoretic approach to inductive definitions which allows the simultaneous definition of finite and infinite behaviors in structural operational semantics — both big-step and small-step styles. We have related various presentations of the bi-inductive semantics, such as explicit fixpoint definitions and the familiar rule-based definitions including in absence of monotony. Bi-induction simultaneously define the finite behaviors by induction and the infinite behaviors by co-induction. Using induction only would exclude infinite behaviors while using co-induction only might introduce spurious finite behaviors (for example in big-step relational semantics).

We have given two examples of using the approach: specifying the finite and infinite semantics of context-free grammars and of the call-by-value λ-calculus, both in small/big-step style and at various levels of abstractions for trace/relational/operational semantics. The lattice of abstractions of the big-step bifinite trace semantics is the following

$$\widehat{\mathbb{S}}^{\omega} = \mathbf{gfp}^{\subseteq} \widehat{F}^{\omega} \qquad \widehat{\mathbb{S}}^{+} = \mathbf{lfp}^{\subseteq} \widehat{F}^{+}$$

relational
semantics
$$\widehat{\mathbb{S}} = \mathbf{lfp}^{\sqsubseteq} \widehat{F} \neq \mathbf{gfp}^{\subseteq} \widehat{F}$$

$$\vec{\mathbb{S}}^{\omega} = \mathbf{gfp}^{\subseteq} \vec{F}^{\omega} \qquad \vec{\mathbb{S}}^{+} = \mathbf{lfp}^{\subseteq} \vec{F}^{+}$$

trace
semantics
$$\mathbb{S} = \mathbf{lfp}^{\subseteq} f = \mathbf{gfp}^{\subseteq} f$$

$$\vec{\mathbb{S}} = \mathbf{lfp}^{\sqsubseteq} \vec{F} = \mathbf{gfp}^{\subseteq} \vec{F}$$

infinite   bifinite   finite   *reduction*

and the lattice of abstractions of the small-step bifinite trace semantics $\vec{\mathbb{S}} = \mathbf{lfp}^{\sqsubseteq} \vec{f}$ is isomorphic.

Sound (and sometimes complete) abstractions are essential to establish this hierarchy of semantics [6] and to prove that all the semantics are well-behaved in the sense that they abstract the small-step trace semantics.

In conclusion bi-inductive definitions should satisfy the need for formal finite and infinite semantics, at various levels of abstraction, and in different styles, despite the possible absence of monotony.

References

[1] S. Abramsky. Semantics of interaction: an introduction to game semantics. In A.M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, pp. 1–32. Cambridge University Press, Cambridge, United Kingdom, 1997.

[2] P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, pp. 739–782. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1977.

[3] I. Attali, J. Chazarain, and S. Gilette. Incremental evaluation of natural semantics specifications. In M. Bruynooghe and M. Wirsing, editors, *Proceedings of the Fourth International Symposium on Programming Language Implementation and Logic Programming, PLILP '92*, Leuven, Belgium, 26–28 August 1992, Lecture Notes in Computer Science 631, pp. 87–99. Springer, Berlin, Germany, 1992.

[4] N. Chomsky and M.P. Schützenberger. The algebraic theory of context-free languages. In P. Bradford and D. Hirschberg, editors, *Computer programming and Formal Systems*, pp. 118–161. North-Holland Pub. Co., Amsterdam, The Netherlands, 1963.

[5] P. Cousot. Types as abstract interpretations, invited paper. In *Conference Record of the Twentyfourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 316–331, Paris, France, January 1997. ACM Press, New York, United States.

[6] P. Cousot. Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. *Theoretical Computer Science*, 277(1—2):47–103, 2002.

[7] P. Cousot and R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific Journal of Mathematics*, 82(1):43–57, 1979.

[8] P. Cousot and R. Cousot. Systematic design of program analysis frameworks. In *Conference Record of the Sixth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 269–282, San Antonio, Texas, 1979. ACM Press, New York, United States.

[9] P. Cousot and R. Cousot. Abstract interpretation frameworks. *Journal of Logic and Computation*, 2(4):511–547, August 1992.

[10] P. Cousot and R. Cousot. Inductive definitions, semantics and abstract interpretation. In *Conference Record of the Ninthteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pp. 83–94, Albuquerque, New Mexico, United States, 1992. ACM Press, New York, United States.

[11] P. Cousot and R. Cousot. Compositional and inductive semantic definitions in fixpoint, equational, constraint, closure-condition, rule-based and game-theoretic form, invited paper. In P. Wolper, editor, *Proceedings of the Seventh International Conference on Computer Aided Verification, CAV '95*, Liège, Belgium, Lecture Notes in Computer Science 939, pp. 293–308. Springer, Berlin, Germany, 3–5 July 1995.

[12] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order, Second Edition.* Cambridge University Press, Cambridge, United Kingdom, 2002.

[13] Th. Despeyroux. TYPOL: a formalism to implement natural semantics. Technical report RT–0094, INRIA Sophia Antipolis, March 1988.

[14] S. Ginsburg and G. Rice. Two families of languages related to ALGOL. *Journal of the Association for Computing Machinery*, 9:350–371, 1962.

[15] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the Association for Computing Machinery*, 12(10):576–580, October 1969.

[16] B. Jacobs and J. Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:222–269, 1997.

[17] G. Kahn. Natural semantics. In K. Fuchi and M. Nivat, editors, *Programming of Future Generation Computers*, pp. 237–258. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1988.

[18] A. Kanamori. The mathematical import of Zermelo's well-ordering theorem. *Bull. Symbolic Logic*, 3:281–311, 1997.

[19] B. Klin. Adding recursive constructs to bialgebraic semantics. *Journal of Logic and Algebraic Programming*, 60—61:259–286, July–December 2004.

[20] B. Klin. Bialgebraic methods in structural operational semantics. *Electronic Notes in Theoretical Computer Science*, 175(1):33–43, May 2007.

[21] J.-L. Lassez, V.L. Nguyen, and L. Sonenberg. Fixed point theorems and semantics: A folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[22] X. Leroy. Coinductive big-step operational semantics. In P. Sestoft, editor, *Proceedings of the Fifteenth European Symposium on Programming Languages and Systems, ESOP '2006*, Vienna, Austria, Lecture Notes in Computer Science 3924, pp. 54–68. Springer, Berlin, Germany, 27–28 March 2006.

[23] R. Milner. Operational and algebraic semantics of concurrent processes. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, chapter 19, pp. 1201–1242. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1990.

[24] P.D. Mosses. Denotational semantics. In J. van Leeuwen, editor, *Formal Models and Semantics*, volume B of *Handbook of Theoretical Computer Science*, chapter 11, pp. 575–631. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1990.

[25] A. Mycroft. The theory and practice of transforming call-by-need into call-by-value. In B. Robinet, editor, *Proceedings of the Fourth International Symposium on Programming*, Paris, France, 22–24 April 1980, Lecture Notes in Computer Science 83, pp. 270–281. Springer, Berlin, Germany, 1980.

[26] M. Nivat. Sur les ensembles de mots infinis engendrés par une grammaire algébrique. *Informatique Théorique et Applications*, 12(3):259–278, 1978.

[27] D. Pataria. A constructive proof of Tarski's fixed-point theorem for dcpo's. *65th Peripatetic Seminar on Sheaves and Logic, Århus, Denmark*, November 1997. Reported by M.H. Escardó in "Joins in the frame of nuclei", *Applied Categorical Structures*, Vol. 11, Issue 2, pp. 117–124, April 2003.

[28] G.D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN–19, Aarhus University, Denmark, September 1981. Reprinted in [30].

[29] G.D. Plotkin. The origins of structural operational semantics. *Journal of Logic and Algebraic Programming*, 60—61:3–15, July–December 2004.

[30] G.D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60—61:17–139, July–December 2004.

[31] M.P. Schützenberger. On a theorem of R. Jungen. *Proceedings of the American Mathematical Society*, 13:885–889, 1962.

[32] P. Steckler and M. Wand. Selective thunkification. In B. Le Charlier, editor, *Proceedings of the First International Symposium on Static Analysis, SAS '94*, Namur, Belgium, 20–22 September 1994, Lecture Notes in Computer Science 864, pp. 162–178. Springer, Berlin, Germany, 1994.

[33] A. Tarski. A lattice theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–310, 1955.

[34] D. Turi and G.D. Plotkin. Towards a mathematical operational semantics. In G. Winskel, editor, *Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science, LICS '1997*, pp. 280–291, Warsaw, Poland, 29 June–2 July 1997. IEEE Computer Society Press, Los Alamitos, California, United States.

[35] F. van Breugel. An introduction to metric semantics: operational and denotational models for programming and specification languages. *Theoretical Computer Science*, 258:1–98, May 2001.