Inductive Definitions, Semantics and Abstract Interpretation*

Patrick Cousot
LIENS, École Normale Supérieure
45, rue d'Ulm
75230 Paris cedex 05 (France)
cousot@dmi.ens.fr

Radhia Cousot
LIX, École Polytechnique
91128 Palaiseau cedex (France)
cousot@polytechnique.fr

Abstract

We introduce and illustrate a *specification method* combining rule-based inductive definitions, well-founded induction principles, fixed-point theory and abstract interpretation for general use in computer science. Finite as well as infinite objects can be specified, at various levels of details related by abstraction. General proof principles are applicable to prove properties of the specified objects.

The specification method is illustrated by introducing G$^\infty$SOS, a structured operational semantics generalizing Plotkin's [28] structured operational semantics (SOS) so as to describe the finite, as well as the infinite behaviors of programs in a uniform way and by constructively deriving inductive presentations of the other (relational, denotational, predicate transformers, ...) semantics from G$^\infty$SOS by abstract interpretation.

1 Inductive definitions

Inductive definitions which are widely used in mathematical logic to define sets inductively generated by closure conditions, have popularized in computer science over the past few years. Classical or positive inductive definitions, co-inductive, kernel or negative definitions [1] as well as bi-inductive definitions mixing these two kinds of definitions are first captured by a general definition of the notion of rule-based inductive definition which is then generalized to systems of iterated inductive definitions so as to define inductively cartesian products of sets of finite and infinite objects indexed by a well-founded ordering. Examples show that many concepts related to programming come out of such inductive definitions.

*This work was supported in part by Esprit BRA 3124 *Sémantique* and Greco CNRS C^3.

Appeared in the "Conference Record of the Nineteenth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages", Albuquerque, New Mexico, USA, January 19-22, 1992, pp. 84–94.

1.1 Rule-based inductive definitions

Example 1 (Even numbers) Let the *universe* U be the set $\mathcal{N}$ of natural numbers. The subset $2\mathcal{N}$ of even numbers can be defined by the following axiom and rule schema:

$$0 \in 2\mathcal{N} \qquad \frac{n \in 2\mathcal{N}}{n+2 \in 2\mathcal{N}} \tag{1}$$

Instead of considering a formal language for writing such rules and axioms schemata, we reason upon a set Φ of rules instances where axioms have empty premises:

$$\Phi = \left\{ \frac{\emptyset}{0} \right\} \cup \left\{ \frac{\{n\}}{n+2} \middle| \, n \in \mathcal{N} \right\} \tag{2}$$

The *induced operator* is:

$$\overline{\Phi}(X) = \{0\} \cup \{n+2 \mid \{n\} \subseteq X\} \tag{3}$$

Let the *basis* be $\perp = \emptyset$ and the *join* be $\cup$ which induces the *ordering* $\subseteq$. The $\subseteq$-smallest set $|\Phi|$ greater than $\perp$ which satisfies the rules Φ is the least fixpoint of $\overline{\Phi}$ that is $2\mathcal{N} = \overline{\Phi}^\omega = \cup_{n \geq 0} \overline{\Phi}^n$ where $\overline{\Phi}^0 = \perp = \emptyset$ and $\overline{\Phi}^{n+1} = \overline{\Phi}(\overline{\Phi}^n) = \{0, 2, \ldots, 2n\}$. The *closure ordinal* of $\overline{\Phi}$ is the first infinite limit ordinal ω. □

We now give a general set-theoretic definition of such inductive definitions. We write $\wp(S)$ for the powerset of the set S and $S \mapsto S'$ (respectively $S \hookrightarrow S'$) for the set of total (respectively partial) maps from S into the set S'.

Definition 2 (Inductive definition) An *inductive definition* $\Im$ is a quadruple $\langle U,\ \Phi,\ \perp,\ \sqcup \rangle$ such that:

- The *universe* U is a set,
- Φ is a set of *rules instances* $\frac{P}{c}$ where $P \subseteq U$ and $c \in U$,
- $\perp \subseteq U$ is the *basis*,
- $\sqcup \in \wp(\wp(U)) \hookrightarrow \wp(U)$ is the *join* and the *induced ordering* is $x \sqsubseteq y \overset{\text{def}}{=} x \sqcup y = y$ is a partial order on $\wp(U)$.

□

Definition 3 (Operator induced by an inductive definition) The *operator* $\overline{\Phi}$ induced by $\Im = \langle U,\ \Phi,\ \perp,\ \sqcup \rangle$ is $\overline{\Phi} \in \wp(U) \mapsto \wp(U)$ such that[1]:

$$\overline{\Phi}(X) = \{c \in U \mid \exists P \subseteq X : \frac{P}{c} \in \Phi\} \tag{4}$$

□

[1] More generality can be obtained when replacing $\subseteq$ by an arbitrary relation.

Definition 4 (Iteration of an operator) The *iteration* of $F \in S \mapsto S$ for $\sqcup \in \wp(S) \hookrightarrow S$ is the partial operator $F_{\sqcup}^{\infty} \in S \hookrightarrow S$ such that $F_{\sqcup}^{\infty}(X) = F^{\epsilon}$ where:

- $F^0 = X$,
- $F^{\lambda} = F(\bigsqcup_{\alpha<\lambda} F^{\alpha})$ if $0 < \lambda \in \mathit{Ord}$,
- $F^{\lambda} = F^{\epsilon}$ if $\lambda > \epsilon$.

$\mathit{clo}_X^{\sqcup}(F) \stackrel{\text{def}}{=} \epsilon$ is called the *closure ordinal* of F for X. $\square$

A *complete partial order* $L \langle\sqsubseteq, \bot, \sqcup\rangle$ (*cpo* for short) is a set partially ordered by $\sqsubseteq$ such that all increasing chains C have least upper bounds $\sqcup C$ and $\bot \in L$ ($\bot$ is usually the infimum of L). It is a *complete lattice* if and only if all subsets X have least upper bounds $\sqcup X$ and therefore greatest lower bounds $\sqcap X$ (in which case we usually write $\bot = \sqcap L$ and $\top = \sqcup L$). $F \in L \mapsto L$ is *monotonic* (written $F \in L \stackrel{mo\sqsubseteq}{\longmapsto} L$) if and only if $\forall x, y \in L : x \sqsubseteq y \Rightarrow F(x) \sqsubseteq F(y)$. It is *extensive* (written $F \in L \stackrel{ex\sqsubseteq}{\longmapsto} L$) if and only if $\forall x \in L : x \sqsubseteq F(x)$). It is a *complete* $\sqcup$*-morphism* (written $F \in L \stackrel{cm\sqcup}{\longmapsto} L$) if for all families of sets $X_i \subseteq L$ such that $\sqcup_i X_i$ exists in L, we have $\sqcup_i F(X_i) = F(\sqcup_i X_i)$.

The propositions below immediately follow from the constructive proof of Tarski's fixpoint theorem [9]:

Proposition 5 If $S \langle\sqsubseteq, \bot, \sqcup\rangle$ is a cpo and $F \in S \mapsto S$ is monotonic or extensive then $F_{\sqcup}^{\infty} \in \mathit{pre}(F) \mapsto S$ where $\mathit{pre}(F) = \{x \in S \mid x \sqsubseteq F(x)\}$ (so that $F_{\sqcup}^{\infty}$ is totally defined for pre-fixpoints of F). $\square$

Definition 6 (Well-formed inductive definition) The inductive definition $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ is *well-formed* if and only if $\overline{\Phi}_{\sqcup}^{\infty}(\bot)$ exists in which case it is the *set* $|\Im|$ *inductively defined by* $\Im$. $\square$

Definition 7 (Inductive definitions on a cpo or a complete lattice) Let $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ be an inductive definition. $\Im$ *is on a cpo* (respectively *on a complete lattice*) if and only if $\wp(U) \langle\sqsubseteq, \bot, \sqcup\rangle$ is a cpo (respectively $\wp(U) \langle\sqsubseteq, \bot, \top, \sqcup, \sqcap\rangle$ is a complete lattice). $\square$

Definition 8 (Monotonic and extensive inductive definitions) The inductive definition $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ is *monotonic* (respectively *extensive*) whenever $\overline{\Phi}$ is monotonic and $\bot \sqsubseteq \overline{\Phi}(\bot)$ (respectively $\overline{\Phi}$ is extensive). $\square$

Corollary 9 A monotonic or extensive inductive definition on a cpo is well-formed. $\square$

When it exists, we write $\mathit{lfp}_x^{\leq} F$ for the least fixpoint of F greater than or equal to x : $F(\mathit{lfp}_x^{\leq} F) = \mathit{lfp}_x^{\leq} F$, $x \leq \mathit{lfp}_x^{\leq} F$ and $\forall y : [F(y) = y \wedge x \leq y] \Rightarrow [\mathit{lfp}_x^{\leq} F \leq y]$. The same way, $\mathit{gfp}_x^{\leq} F$ is the greatest fixpoint of F less than or equal to x.

Proposition 10 If $S \langle\sqsubseteq, \bot, \sqcup\rangle$ is a cpo and $F \in S \stackrel{mo\sqsubseteq}{\longmapsto} S$ then $F_{\sqcup}^{\infty} \in \mathit{pre}(F) \mapsto S$ is an upper closure operator, for all $X \in \mathit{pre}(F)$, $F_{\sqcup}^{\infty}(X)$ is the least fixpoint $\mathit{lfp}_X^{\sqsubseteq} F$ of F greater than or equal to X. $\square$

Corollary 11 If $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ is a monotonic inductive definition on a cpo then $\overline{\Phi}_{\sqcup}^{\infty}$ is an upper closure operator, $\overline{\Phi}_{\sqcup}^{\infty}(X)$ is the least fixpoint $\mathit{lfp}_X^{\sqsubseteq} \overline{\Phi}$ of $\overline{\Phi}$ greater than or equal to $X \in \mathit{pre}(\overline{\Phi})$ for the ordering $\sqsubseteq$ induced by $\sqcup$ and $|\Im| = \overline{\Phi}_{\sqcup}^{\infty}(\bot) = \mathit{lfp}_{\bot}^{\sqsubseteq} \overline{\Phi}$. $\square$

Definition 12 (Rule satisfaction) The set $S \subseteq U$ is said to *satisfy the rules* Φ if and only if $\mathit{sat}_{\Im}(S) = (\bot \sqsubseteq S) \wedge (\overline{\Phi}(S) \sqsubseteq S)$ holds. $\square$

Proposition 13 If $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ is a monotonic inductive definition on a complete lattice then $|\Im| = \sqcap\{S \subseteq U \mid \mathit{sat}_{\Im}(S)\}$ is the least set (for $\sqsubseteq$) which satisfies the rules. $\square$

1.2 Positive inductive definitions

Definition 14 (Positive inductive definition) A *positive inductive definition* $\Im = \langle U, \Phi\rangle$ is $\Im^+ = \langle U, \Phi, \emptyset, \cup\rangle$. $\square$

Proposition 15 A positive inductive definition $\Im^+ = \langle U, \Phi\rangle^+$ is well-formed and $|\Im^+| = \overline{\Phi}_{\cup}^{\infty}(\emptyset) = \mathit{lfp}\, \overline{\Phi} = \cap\{S \subseteq U \mid \mathit{sat}_{\Im^+}(S)\}$ is the least (for $\subseteq$) *closed* set, that is which satisfies the rules: $\mathit{closed}_{\Phi}(S) \stackrel{\text{def}}{=} \mathit{sat}_{\Im^+}(S) = [\forall \frac{P}{c} \in \Phi : (P \subseteq S) \Rightarrow (c \in S)]$. $\square$

Example 16 (Logic programs) Using the notations of [2], let P be a logic program (containing at least one constant), B_P be its Herbrand universe and $\mathit{ground}(P)$ be the set of all ground instances of clauses in P. The inductive definition corresponding to P is $\Im^+ = \langle B_P, \Phi\rangle^+$ where:

$$\Phi = \left\{ \frac{B_1,\ldots,B_n}{A} \mid A \leftarrow B_1,\ldots,B_n \in \mathit{ground}(P) \right\} \qquad (5)$$

The operator $\overline{\Phi}$ induced by $\Im^+$ is the immediate consequence operator T_P. Then proposition (15) implies the characterization theorem of van Emden and Kowalski [30]: A closed set $I \subseteq B_P$ (such that $T_P(I) \subseteq I$) is a model of P, the least model M_P being the least fixpoint ${T_P}_{\cup}^{\infty}(\emptyset)$ of T_P. Lassez and Maher [20] observed that ${T_P}_{\cup}^{\infty}$ is a closure operator. $\square$

Example 17 (Maximal finite execution traces) Following [21], let $T = \langle S, \mathit{Act}, \{\xrightarrow{\alpha} \mid \alpha \in \mathit{Act}\}, \mathit{Init}\rangle$ be a *labelled transition system* where S is a set of states, Act is a set of actions, each $\xrightarrow{\alpha} \subseteq S \times S$ is the transition relation for action $\alpha \in \mathit{Act}$ and $\mathit{Init} \subseteq S$ is the set of initial states. We write $s \xrightarrow{\alpha} s'$ for $\langle s, s'\rangle \in \xrightarrow{\alpha}$, $s \stackrel{\alpha}{\not\rightarrow} s'$ for $\langle s, s'\rangle \notin \xrightarrow{\alpha}$, $s \longrightarrow s'$ for $\exists \alpha \in \mathit{Act} : s \xrightarrow{\alpha} s'$ and $s \not\rightarrow$ for $\forall \alpha \in \mathit{Act} : \forall s' \in S : s \stackrel{\alpha}{\not\rightarrow} s'$.

$\Sigma^{<\omega}$ is the universe of finite traces $\sigma = \overline{\sigma}_0 \overset{\vec{\sigma}_0}{\dashrightarrow} \overline{\sigma}_1 \cdots \overline{\sigma}_{\ell-1} \overset{\vec{\sigma}_{\ell-1}}{\dashrightarrow} \overline{\sigma}_\ell$ such that $\ell = |\sigma| - 1$ where $|\sigma|$ is the length of σ, $1 \leq |\sigma| < \omega$, $\overline{\sigma}_i \in S$ for $0 \leq i \leq \ell$, $\vec{\sigma}_j \in Act$ for $0 \leq j < \ell$ and $\sigma \overset{\alpha}{\dashrightarrow} \sigma'$ is the concatenation of σ and σ' through action α, also written $\sigma \xrightarrow{\alpha} \sigma'$ when $\overline{\sigma}_{|\sigma|-1} \xrightarrow{\alpha} \overline{\sigma'}_0$. Σ^ω is the universe of infinite traces $\sigma = \overline{\sigma}_0 \overset{\vec{\sigma}_0}{\dashrightarrow} \overline{\sigma}_1 \cdots \overline{\sigma}_n \overset{\vec{\sigma}_n}{\dashrightarrow} \overline{\sigma}_{n+1} \cdots$ such that $|\sigma| = \omega$, $\overline{\sigma}_n \in S$ and $\vec{\sigma}_n \in Act$ for $n \geq 0$. $\Sigma^{\leq\omega} \stackrel{\text{def}}{=} \Sigma^{<\omega} \cup \Sigma^\omega$ is the universe of traces.

The set $T^{<\omega}$ of *maximal finite execution traces* of this transition system T is defined by the following schemata (where free variables $\alpha \in Act$, $s \in S$ and $\sigma \in \Sigma^{<\omega}$ are universally quantified):

$$\frac{s \not\longrightarrow}{s \in T^{<\omega}} \qquad\qquad \frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^{<\omega}}{s \xrightarrow{\alpha} \sigma \in T^{<\omega}} \tag{6}$$

which stands for the set Φ of rule instances:

$$\frac{\emptyset}{s}\,{}^{+\Sigma^{<\omega}} \qquad\qquad \frac{\{\sigma\}}{s' \xrightarrow{\alpha} \sigma}\,{}^{+\Sigma^{<\omega}} \tag{7}$$

for all $s \in S$ such that $s \not\longrightarrow$ and $s' \in S$, $\alpha \in Act$ and $\sigma \in \Sigma^{<\omega}$ such that $s' \xrightarrow{\alpha} \overline{\sigma}_0$. The rules are decorated by the $+$ sign to indicate that $\bot = \emptyset$ and $\sqcup = \cup$ followed by the universe $U = \Sigma^{<\omega}$. One or both indications can be left out when clear from the context.

The monotone operator $\overline{\Phi} \in \wp(\Sigma^{<\omega}) \overset{mo\ \subseteq}{\longmapsto} \wp(\Sigma^{<\omega})$ induced by this inductive definition is:

$$\begin{aligned} \overline{\Phi}(X) \ = \ & \{s \mid s \not\longrightarrow\} \cup \\ & \{s \xrightarrow{\alpha} \sigma \mid s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in X\} \end{aligned} \tag{8}$$

Let us define $T^{0\dashv} \stackrel{\text{def}}{=} \emptyset$ and the set $T^{n\dashv} \stackrel{\text{def}}{=} \big\{\sigma \in \Sigma^{<\omega} \;\big|\; |\sigma| = n \wedge \forall i < |\sigma| - 1 : \overline{\sigma}_i \xrightarrow{\vec{\sigma}_i} \overline{\sigma}_{i+1} \wedge \overline{\sigma}_{n-1} \not\longrightarrow\big\}$ of execution traces of T of length $n > 0$. Observe, by induction on n, that the iterates of the operator $\overline{\Phi}$ are $\overline{\Phi}^n = \cup_{i=0}^n T^{i\dashv}$ so that $\overline{\Phi}^\omega = \cup_{n\geq 0} T^{n\dashv} = T^{<\omega}$. Moreover $\overline{\Phi}^{\omega+1} = \overline{\Phi}^\omega$ so that the closure ordinal of $\overline{\Phi}$ is ω. Finally, we have defined $\overline{\Phi}^\omega = \mathit{lfp}^{\subseteq}_{\emptyset}\, \overline{\Phi} = T^{<\omega} = \big\{\sigma \in \Sigma^{<\omega} \;\big|\; \forall i < |\sigma| - 1 : \overline{\sigma}_i \xrightarrow{\vec{\sigma}_i} \overline{\sigma}_{i+1} \wedge \overline{\sigma}_{|\sigma|-1} \not\longrightarrow\big\}$. $\square$

1.3 Negative or co-inductive definitions

Definition 18 (Negative co-inductive definition) A *negative inductive definition* $\Im = \langle U, \Phi\rangle$ is $\Im^- = \langle U, \Phi, U, \cap\rangle$. $\square$

Proposition 19 A negative inductive definition $\Im^- = \langle U, \Phi\rangle^-$ is well-formed and $|\Im^-| = \overline{\Phi}_\cap^\infty(U) = \mathit{gfp}\, \overline{\Phi} = \cup\{S \subseteq U \mid \mathit{sat}_{\Im^-}(S)\}$ is the greatest (for $\subseteq$) *dense* set, i.e. satisfying the rules: $\mathit{dense}_\Phi(S) \stackrel{\text{def}}{=} \mathit{sat}_{\Im^-}(S) = [\forall c \in S : \exists P \subseteq S : \frac{P}{c} \in \Phi]$. $\square$

The definition of observation equivalence in [21] and of static typing in [22] are examples of negative co-inductive definitions.

Example 20 (Infinite execution traces) The set T^ω of *infinite execution traces* of a transition system $T = \langle S, Act, \{\xrightarrow{\alpha} \mid \alpha \in Act\}, Init\rangle$ is defined by the following schema (where free variables $\alpha \in Act$, $s \in S$ and $\sigma \in \Sigma^\omega$ are universally quantified):

$$\frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^\omega}{s \xrightarrow{\alpha} \sigma \in T^\omega}\,{}^{-} \tag{9}$$

which stands for the set Φ of rule instances:

$$\frac{\{\sigma\}}{s \xrightarrow{\alpha} \sigma}\,{}^{-\Sigma^\omega} \tag{10}$$

for all $s \in S$, $\alpha \in Act$ and $\sigma \in \Sigma^\omega$ such that $s \xrightarrow{\alpha} \overline{\sigma}_0$.

The monotone operator $\overline{\Phi} \in \wp(\Sigma^\omega) \overset{mo\ \subseteq}{\longmapsto} \wp(\Sigma^\omega)$ induced by this inductive definition is:

$$\overline{\Phi}(X) \ = \ \{s \xrightarrow{\alpha} \sigma \mid s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in X\} \tag{11}$$

Let us define $T^{0\infty} \stackrel{\text{def}}{=} \Sigma^\omega$ and the set $T^{n\infty} \stackrel{\text{def}}{=} \big\{\sigma \in \Sigma^\omega \;\big|\; \forall i < n : \overline{\sigma}_i \xrightarrow{\vec{\sigma}_i} \overline{\sigma}_{i+1}\big\}$ of infinite traces starting with $n \geq 1$ transitions. Observe, by induction on n, that the iterates of the operator $\overline{\Phi}$ are $\overline{\Phi}^n = \cap_{i=0}^n T^{i\infty}$ so that $\overline{\Phi}^\omega = \cap_{n\geq 0} T^{n\infty} = T^\omega$. Moreover $\overline{\Phi}^{\omega+1} = \overline{\Phi}^\omega$ so that the closure ordinal of $\overline{\Phi}$ is ω. Finally, we have defined $\overline{\Phi}^\omega = \mathit{gfp}^{\subseteq}_{\Sigma^\omega}\, \overline{\Phi} = T^\omega = \big\{\sigma \in \Sigma^\omega \;\big|\; \forall n \geq 0 : \overline{\sigma}_n \xrightarrow{\vec{\sigma}_n} \overline{\sigma}_{n+1}\big\}$. $\square$

Example 21 (Maximal execution traces I) The set $T^{\leq\omega} \stackrel{\text{def}}{=} T^{<\omega} \cup T^\omega$ of *execution traces* of a transition system $T = \langle S, Act, \{\xrightarrow{\alpha} \mid \alpha \in Act\}, Init\rangle$ can be characterized by a negative inductive definition $\Im^- = \langle U, \Phi^-, U, \cap\rangle$ where $U = \Sigma^{\leq\omega}$ and Φ^- is specified by the following rule schemata:

$$\frac{s \not\longrightarrow}{s \in T^{\leq\omega}}\,{}^{-} \qquad\qquad \frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^{\leq\omega}}{s \xrightarrow{\alpha} \sigma \in T^{\leq\omega}}\,{}^{-} \tag{12}$$

The monotone operator $\overline{\Phi^-} \in \wp(\Sigma^{\leq\omega}) \overset{mo\ \subseteq}{\longmapsto} \wp(\Sigma^{\leq\omega})$ induced by this inductive definition is:

$$\begin{aligned} \overline{\Phi^-}(X) \ = \ & \{s \in S \mid s \not\longrightarrow\} \cup \\ & \{s \xrightarrow{\alpha} \sigma \mid s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in X\} \end{aligned} \tag{13}$$

Define $T^{nm} \stackrel{\text{def}}{=} \big\{\sigma \in \Sigma^{<\omega} \;\big|\; |\sigma| = m > n \wedge \forall i < n : \overline{\sigma}_i \xrightarrow{\vec{\sigma}_i} \overline{\sigma}_{i+1}\big\}$ to be the set of finite traces of length m starting with n transitions. Observe, by induction on n, and using the notations of examples 17 and 20 that the iterates of the operator $\overline{\Phi^-}$ are $\overline{\Phi^-}^n = (\cup_{i=0}^n T^{i\dashv}) \cup (\cup_{m>n} T^{nm}) \cup (\cap_{i=0}^n T^{i\infty})$. For all m, $n \geq 0$, if $\sigma \in T^{m\dashv}$ then $\sigma \in \overline{\Phi^-}^n$ since either $n \geq m$ and $\sigma \in T^{m\dashv} \subseteq \overline{\Phi^-}^n$ or $n < m$ and $\sigma \in T^{nm} \subseteq \overline{\Phi^-}^n$. Also $\sigma \in T^\omega$ implies $\sigma \in T^{i\infty}$ for all $i \geq 0$. Reciprocally $\sigma \notin T^{\leq\omega}$ has either $\overline{\sigma}_n \overset{\vec{\sigma}_n}{\not\longrightarrow} \overline{\sigma}_{n+1}$ for some n such that $0 \leq n < |\sigma| - 1$, in which case $\sigma \notin \overline{\Phi^-}^{n+1}$ or $|\sigma| = \ell$ and $\exists s \in S : \overline{\sigma}_{\ell-1} \longrightarrow s$ so that again $\sigma \notin \overline{\Phi^-}^{\ell+1}$. It follows that: $\overline{\Phi^-}^\omega = \cap_{n\geq 0} \overline{\Phi^-}^n = \cap_{n\geq 0}[(\cup_{i=0}^n T^{i\dashv}) \cup (\cup_{m>n} T^{nm}) \cup (\cap_{i=0}^n T^{i\infty})] = (\cup_{n\geq 0} T^{i\dashv}) \cup (\cap_{i=0}^n T^{i\infty}) = T^{<\omega} \cup T^\omega = T^{\leq\omega}$. Moreover $\overline{\Phi^-}^{\omega+1} = \overline{\Phi^-}^\omega$ so that the closure ordinal of $\overline{\Phi^-}$ is ω. Finally, we have defined $\overline{\Phi^-}^{\omega} = gfp^{\subseteq}_{\Sigma^{\leq\omega}} \overline{\Phi^-} = T^{\leq\omega}$. However this definition is not quite satisfactory because of the term $\cup_{m>n} T^{nm}$ in $\overline{\Phi^-}^{n}$ which disappears when passing to the limit $\overline{\Phi^-}^{\omega}$ (since $\cap_{n\geq 0} \cup_{m>n} T^{nm} = \emptyset$). A definition of $T^{\leq\omega}$ such that $\overline{\Phi^-}^{\omega}$ is a conservative extension of the $\overline{\Phi^-}^{n}$ is proposed in example 28. $\square$

If $F \in B \mapsto B$ is an operator on a complete boolean lattice $B\ \langle\sqsubseteq,\ \bot,\ \top,\ \sqcup,\ \sqcap,\ \neg\rangle$, its *dual* is $\widetilde{F} \in B \mapsto B$ is defined by $\forall x \in B : \widetilde{F}(x) = \neg F(\neg x)$. If F is monotonic, so is $\widetilde{F}$. By duality, a negative inductive definition can always be given an indirect equivalent positive definition. However, as shown by the above examples, most of the time, using a direct co-inductive definition is much more clear.

Proposition 22 (Duality of positive and negative inductive definitions) If $\Im = \langle U,\ \Phi\rangle$ is an inductive definition then $sat_{\Im^+}(S) = sat_{\Im^-}(\neg S)$, $|\Im^+| = lfp\,\overline{\Phi} = \neg gfp\,\widetilde{\overline{\Phi}}$, $|\Im^-| = gfp\,\overline{\Phi} = \neg lfp\,\widetilde{\overline{\Phi}}$ and $|\Im^+| = \neg|\Im^-|$. $\square$

1.4 Bi-inductive definitions

Assume the universe U is covered by two subsets U^+ and U^-, that is $U = U^+ \cup U^-$. For example U^+ may be the set of finite objects of U while U^- is the set of infinite objects of U. To define a subset $|\Im|$ of U, we could define separately the finite objects $|\Im| \cap U^+$ using a positive inductive definition and the infinite objects $|\Im| \cap U^-$ using a negative inductive definition. In practice, this must often be done simultaneously. For example in denotational semantics the terminating and non-terminating behaviors of programs are defined at the same time using a single fixpoint operator. Since, for clarity, we insist upon using rule-based inductive definitions (preferred to fixpoint definitions [5]), we propose a method for combining inductive and co-inductive definitions which allows for the simultaneous use of positive and negative axioms and induction rules. We then give equivalent (but maybe less intuitive) fixpoint characterizations.

Definition 23 (Bi-inductive definition) A *bi-inductive definition* $\Im = \langle U,\ \pi^+,\ \pi^-,\ \Phi\rangle$, where π^+, $\pi^- \in \wp(U) \mapsto \wp(U)$ is $\Im^{\pm} = \langle U^{\pm}/\equiv^{\pm},\ \Phi^{\pm},\ \bot^{\pm},\ \sqcup^{\pm}\rangle$ with, for all S, T, $S_i \subseteq U$:

- $U^{\pm} \stackrel{\text{def}}{=} \pi^+(U) \cup \pi^-(U)$,
- $S \equiv^{\pm} T \stackrel{\text{def}}{=} [\pi^+(S) = \pi^+(T)] \wedge [\pi^-(S) = \pi^-(T)]$,
- $\Phi^+ \stackrel{\text{def}}{=} \left\{ \frac{\pi^+(P)}{c'} \middle| \frac{P}{c} \in \Phi \ \wedge\ c' \in \pi^+(\{c\})\right\}$,
- $\Phi^- \stackrel{\text{def}}{=} \left\{ \frac{\pi^-(P)}{c'} \middle| \frac{P}{c} \in \Phi \ \wedge\ c' \in \pi^-(\{c\})\right\}$,
- $\Phi^{\pm} \stackrel{\text{def}}{=} \Phi^+ \cup \Phi^-$,
- $\bot^{\pm} \stackrel{\text{def}}{=} \pi^-(U)$ and
- $\sqcup^{\pm}_i S_i \stackrel{\text{def}}{=} \bigcup_i \pi^+(S_i) \cup \bigcap_i \pi^-(S_i)$.

$\square$

We often use bi-inductive definitions to simultaneously define subsets of the universe which are separated in the following sense:

Definition 24 (Separated bi-inductive definition) A bi-inductive definition $\Im = \langle U,\ \pi^+,\ \pi^-,\ \Phi\rangle$ is *separated* if and only if π^+ and π^- are monotonic for $\subseteq$, idempotent and satisfy the *separation property*: $\forall P \subseteq \pi^+(U) : \forall M \subseteq \pi^-(U) : \pi^+(P \cup M) = \pi^+(P)$ and $\pi^-(P \cup M) = \pi^-(M)$. $\square$

Proposition 25 Let $\Im = \langle U,\ \pi^+,\ \pi^-,\ \Phi\rangle$ be a separated bi-inductive definition. Let us define, for all S, T, $S_i \subseteq U$:

- $S \sqsubseteq^{\pm} T \stackrel{\text{def}}{=} [\pi^+(S) \subseteq \pi^+(T)] \wedge [\pi^-(S) \supseteq \pi^-(T)]$,
- $\top^{\pm} \stackrel{\text{def}}{=} \pi^+(U)$ and
- $\sqcap^{\pm}_i S_i \stackrel{\text{def}}{=} \bigcap_i \pi^+(S_i) \cup \bigcup_i \pi^-(S_i)$.

Then:

1) $\wp(U)/\equiv^{\pm}\ \langle\sqsubseteq^{\pm},\ \bot^{\pm},\ \top^{\pm},\ \sqcup^{\pm},\ \sqcap^{\pm}\rangle$ is a complete lattice,
2) $\overline{\Phi^{\pm}}$ is monotonic for $\sqsubseteq^{\pm}$,
3) $\Im^{\pm}$ is well-formed.

$\square$

Disjoined bi-inductive definitions are used to simultaneously define subsets of disjoined universes:

Definition 26 (Disjoined bi-inductive definition) A *disjoined bi-inductive definition* $\Im = \langle U^+,\ U^-,\ \Phi\rangle$ where $U^+ \cap U^- = \emptyset$ is $\Im^{\pm} = \langle U^+ \cup U^-,\ \pi^+,\ \pi^-,\ \Phi\rangle$ with $\pi^+(X) = X \cap U^+$ and $\pi^-(X) = X \cap U^-$. $\square$

Proposition 27 A disjoined bi-inductive definition $\Im^{\pm} = \langle U^+,\ U^-,\ \Phi\rangle^{\pm}$ is separated, hence well-formed and such that $\equiv^{\pm}$ is equality and $|\Im^{\pm}| = lfp^{\sqsubseteq^{\pm}}\,\overline{\Phi^{\pm}} = |\Im^+| \cup |\Im^-| = lfp^{\subseteq}\,\overline{\Phi^+} \cup gfp^{\subseteq}\,\overline{\Phi^-}$. $\square$

Example 28 (Maximal execution traces II) The set $T^{\leq\omega} \stackrel{\text{def}}{=} T^{<\omega} \cup T^{\omega}$ of *execution traces* of a transition system $T = \langle S,\ Act,\ \{\xrightarrow{\alpha} \mid \alpha \in Act\},\ Init\rangle$ can be defined by a disjoined bi-inductive definition $\Im^{\pm} = \langle U^+,\ U^-,\ \Phi^{\pm}\rangle^{\pm}$ where $U^+ = \Sigma^{<\omega}$ is the set of finite traces, $U^- = \Sigma^{\omega}$ is the set of infinite traces over S and Act and $\Phi^{\pm}$ is specified by the following rule schemata:

$$\frac{s \not\rightarrow}{s \in T^{\leq\omega}}{}^{+} \qquad\qquad \frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^{\leq\omega}}{s \xrightarrow{\alpha} \sigma \in T^{\leq\omega}}{}^{+} \tag{14}$$

$$\frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^{\leq\omega}}{s \xrightarrow{\alpha} \sigma \in T^{\leq\omega}}{}^{-} \tag{15}$$

For simplicity we write $\sigma \in T^{\leq\omega}$ instead of $\sigma \in \pi^+(T^{\leq\omega})$ (that is $\sigma \in T^{<\omega}$) in positive rule schemata and the same way $T^{\leq\omega}$ stands for $\pi^-(T^{\leq\omega})$ that is T^{ω}

in the negative ones. For short, the rule schemata (14) and (15) can be written as follows:

$$\frac{s \not\to}{s \in T^{\leq\omega}}\,+ \qquad\qquad \frac{s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in T^{\leq\omega}}{s \xrightarrow{\alpha} \sigma \in T^{\leq\omega}}\,\pm \qquad (16)$$

The monotone operator $\overline{\Phi^{\pm}} \in \wp(\Sigma^{\leq\omega}) \overset{mo\,\sqsubseteq^{\pm}}{\longmapsto} \wp(\Sigma^{\leq\omega})$ induced by this inductive definition is:

$$\begin{aligned}\overline{\Phi^{\pm}}(X) \ = \ & \{s \in S \mid s \not\to\} \cup \\ & \{s \xrightarrow{\alpha} \sigma \mid s \xrightarrow{\alpha} \overline{\sigma}_0 \ \wedge\ \sigma \in X\}\end{aligned} \qquad (17)$$

Observe, by induction on n, and using the notations of examples 17 and 20 that the iterates of the operator $\overline{\Phi^{\pm}}$ are $\overline{\Phi^{\pm}}^{n} = (\cup_{i=0}^{n} T^{i\dashv}) \cup (\cap_{i=0}^{n} T^{i\infty})$ so that $\overline{\Phi^{\pm}}^{\omega} = (\cup_{n\geq 0} T^{n\dashv}) \cup (\cap_{n\geq 0} T^{n\infty}) = T^{<\omega} \cup T^{\omega} = T^{\leq\omega}$. Moreover $\overline{\Phi^{\pm}}^{\omega+1} = \overline{\Phi^{\pm}}^{\omega}$ so that the closure ordinal of $\overline{\Phi^{\pm}}$ is ω. Finally, we have defined $\overline{\Phi^{\pm}}^{\omega} = \mathit{lfp}_{\perp^{\pm}}^{\sqsubseteq^{\pm}} \overline{\Phi^{\pm}} = T^{\leq\omega}$. □

Our approach for defining program behaviors, and more generally a subset of a space with finite and infinite computable objects, is characterized by the following remarks:

1. The finite objects are constructed from their finite components using positive rules;
2. The infinite objects are not obtained as limits of finite ones (this may be done once for all in the domain theoretical definition of the universe U) but selected among all possible ones by successive inspections of finite parts using negative rules;
3. By combining the two methods, one can define other fixpoints, in addition to the usual least (for $\subseteq$) and greatest fixpoints.

The usefulness of such non-extremal fixpoints is illustrated by the example below:

Example 29 (Infinitary languages) Let $A = \{a, b, c\}$ be an alphabet, $U = U^{+} \cup U^{-}$ where $U^{+} = A^{*}$ and $U^{-} = A^{\omega}$ are respectively the sets of finite and infinite words written on A. The infinitary language $L = (a \cup b)^{*}c \cup b^{\omega}$ is defined by the following axioms and rule schemata (which are disjoined since $U^{+} \cap U^{-} = \emptyset$):

$$c \in L\,+ \qquad \frac{\sigma \in L}{a\sigma \in L}\,+ \qquad \frac{\sigma \in L}{b\sigma \in L}\,\pm \qquad (18)$$

The operator associated with the instances of rules schemata (18) is $\overline{\Phi^{\pm}} = \{c\} \cup \{a\sigma \mid \sigma \in X \cap A^{*}\} \cup \{b\sigma \mid \sigma \in X\}$. It is a monotone operator on $\wp(U)\langle\sqsubseteq,^{\pm} \perp^{\pm}, \top^{\pm}, \sqcup^{\pm}, \sqcap^{\pm}\rangle$ and $\mathit{lfp}^{\sqsubseteq^{\pm}} \overline{\Phi^{\pm}} = (a \cup b)^{*}c \cup b^{\omega} = L$. Observe however that L is neither the least fixpoint $(a \cup b)^{*}c$ nor the greatest fixpoint $(a \cup b)^{*}c \cup (a \cup b)^{\omega}$ of the context-free grammar $X ::= c \mid aX \mid bX$ (these fixpoints being extremal for $\subseteq$). □

1.5 Proof methods

It is interesting to notice that numerous mathematical tools come of such inductive definitions. In particular, methods for proving properties of fixpoints (such as Park's fixpoint induction) can be transcribed to prove properties of inductively defined sets.

Proposition 30 (Fixpoint induction) Let $L\ \langle\sqsubseteq, \perp, \sqcup\rangle$ be a cpo and $F \in L \overset{mo\,\sqsubseteq}{\longmapsto} L$. Then $[\mathit{lfp}_{\perp}^{\sqsubseteq} F \sqsubseteq Y] \Leftrightarrow [\exists Z \in L : \perp \sqsubseteq Z \ \wedge\ F(Z) \sqsubseteq Z \ \wedge\ Z \sqsubseteq Y]$. If $\sqcap$ is well-defined then $[X \sqcap \mathit{lfp}_{\perp}^{\sqsubseteq} F \sqsubseteq Y] \Leftrightarrow [\exists Z \in L : \perp \sqsubseteq Z \ \wedge\ F(Z) \sqsubseteq Z \ \wedge\ X \sqcap Z \sqsubseteq Y]$ □

Corollary 31 If $\Im = \langle U, \Phi, \perp, \sqcup\rangle$ is a monotonic inductive definition on a cpo then $|\Im| \sqsubseteq S$ if and only if $\exists S' \subseteq U : (S' \sqsubseteq S) \wedge \mathit{sat}_{\Im}(S')$. □

Moreover $x \prec y \overset{\text{def}}{=} \exists \alpha, \beta \in \mathit{Ord} : \alpha < \beta \wedge x \in \overline{\Phi}^{\alpha} \wedge y \in \overline{\Phi}^{\beta} - \overline{\Phi}^{\alpha}$ is a strict well-founded partial ordering so that transfinite induction can be used to prove the inverse inequality:

Definition 32 (Well-foundedness) A relation $\prec$ on a class W is *well-founded* if and only if $\mathit{WellFounded}(W, \prec) = \forall E \subseteq W : [E \neq \emptyset \Rightarrow \exists y \in E : (\neg\exists z \in E : z \prec y)]$ holds. □

Proposition 33 (Iteration induction) Let $L\ \langle\sqsubseteq, \perp, \sqcup\rangle$ be a cpo with infimum $\perp = \sqcup\emptyset$, $F \in L \overset{mo\,\sqsubseteq}{\longmapsto} L$ and $Q \in L$. We have: $[Q \sqsubseteq \mathit{lfp}\, F] \Leftrightarrow [\exists W : \exists \prec \subseteq W \times W : \mathit{WellFounded}(W, \prec) \wedge \exists I \in W \mapsto L : (\forall x \in W : I(x) \sqsubseteq F(\bigsqcup_{x' \prec x} I(x'))) \wedge (Q \sqsubseteq \bigsqcup_{x \in W} I(x))]$ □

Another useful method for proving properties of fixpoints is computational induction. We let Ord be the class of ordinals and $\mathit{Limit} = \{\alpha \in \mathit{Ord} \mid \cup\alpha = \alpha \neq 0\}$ be the class of limit ordinals.

Proposition 34 (Computational induction) Let $L\ \langle\sqsubseteq, \perp, \sqcup\rangle$ be a cpo, $F \in L \overset{mo\,\sqsubseteq}{\longmapsto} L$ and $P \in \wp(L)$. We have: $(\mathit{lfp}_{\perp}^{\sqsubseteq} F \in P) \Leftrightarrow (\exists I \in \mathit{Ord} \mapsto L : [\perp \in I(0)] \wedge [\forall \alpha \in \mathit{Ord} : \forall X \in I(\alpha) : F(X) \in I(\alpha+1)] \wedge [\forall \alpha \in \mathit{Limit} : \forall X \in \alpha \overset{mo\,<}{\longmapsto} L : [\forall \beta < \alpha : X(\beta) \in I(\beta)] \Rightarrow [\bigsqcup_{\beta<\alpha} X(\beta) \in I(\alpha)]] \wedge [\forall X \in \mathit{Ord} \overset{mo\,<}{\longmapsto} L : (\forall \alpha \in \mathit{Ord} : X(\alpha) \in I(\alpha)) \Rightarrow (\bigsqcup_{\alpha \in \mathit{Ord}} X(\alpha) \in P)])$ □

An interesting particular case (which amounts to Scott induction when the function is *upper continuous* and the property is *admissible*) consists in choosing the invariant $I(\alpha)$ as P, but this is not a semantically complete proof method.

Proposition 35 (Stepwise computational induction) Let $L\ \langle\sqsubseteq, \perp, \sqcup\rangle$ be a cpo, $F \in L \overset{mo\,\sqsubseteq}{\longmapsto} L$ and $P \in \wp(L)$. We have: $([\perp \in P] \wedge [\forall X \in P : F(X) \in P] \wedge [\forall \alpha \in \mathit{Limit} : \forall X \in \alpha \overset{mo\,<}{\longmapsto} L : (\forall \beta < \alpha : X(\beta) \in P) \Rightarrow (\bigsqcup_{\beta<\alpha} X(\beta) \in P)]) \Rightarrow (\mathit{lfp}_{\perp}^{\sqsubseteq} F \in P)$. □

A last example is the inductive definitions of functions $f \in |\Im| \mapsto D$:

Proposition 36 (Inductive function definition) Let $\Im = \langle U, \Phi, \bot, \sqcup\rangle$ be a well-formed inductive definition, D be a set and $\tau_i \in \wp(P_i \times D) \mapsto D$ for all $\frac{P_i}{c_i} \in \Phi$. There exists a unique total function $f \in |\Im| \mapsto D$ such that for all $\frac{P_i}{c_i} \in \Phi : f(c_i) = \tau_i(\{\langle x, f(x)\rangle \mid x \in P_i\})$. $\square$

When specialized to positive inductive definitions, propositions (30) and (33) amount to (for short, similar corollaries holding for the other varieties of inductive definitions are not stated):

Corollary 37 (Fixpoint induction for positive inductive definitions) Let $\Im^+ = \langle U, \Phi\rangle^+$ be a positive inductive definition and $R \subseteq U$. Then $[|\Im^+| \cap Q \subseteq R] \Leftrightarrow [\exists I \in U \cup Q \mapsto \{tt, ff\} : (\forall \frac{P}{c} \in \Phi : [\forall x \in P : I(x)] \Rightarrow I(c)) \wedge (\forall x \in Q : I(x) \Rightarrow x \in R)]$. $\square$

Corollary 38 (Iteration induction for positive inductive definitions) Let $\Im^+ = \langle U, \Phi\rangle^+$ be a positive inductive definition and $Q \subseteq U$. We have: $[Q \subseteq |\Im^+|] \Leftrightarrow [\exists W : \exists \prec \subseteq W \times W : \mathit{WellFounded}(W, \prec) \wedge \exists I \in W \mapsto \wp(U) : [\forall u \in Q : \exists x \in W : u \in I(x))] \wedge [\forall x \in W : \forall c \in I(x) : \exists \frac{P}{c} \in \Phi : \forall c' \in P : \exists x' \prec x : c' \in I(x')]]$. $\square$

Example 39 (Well-founded part of a relation) Let S be a set of states and $t \subseteq S \times S$ be a relation on S. We write $s \xrightarrow{t} s'$ for $\langle s, s'\rangle \in t$. The well-founded part $\mathit{Wf}(t)$ of t is the set of $\sigma_0 \in S$ such that there is no infinite sequence $\sigma_0 \xrightarrow{t} \sigma_1 \xrightarrow{t} \sigma_2 \xrightarrow{t} \ldots$ that is:

$$\mathit{Wf}(t) = \{s \in S \mid \neg(\exists \sigma \in \omega \mapsto S : s = \sigma_0 \wedge \forall i \in \omega : \sigma_i \xrightarrow{t} \sigma_{i+1})\} \quad (19)$$

The well-founded part $\mathit{Wf}(t)$ of t is specified by the inductive definition $\Im^+ = \langle S, \Phi\rangle^+$ where Φ consists of the rules instances [1]:

$$\frac{\{s' \in S \mid s \xrightarrow{t} s'\}}{s} \quad (20)$$

for all $s \in S$.

- To show that $|\Im^+| \subseteq \mathit{Wf}(t)$, we use proposition 37 with $I(x) = [x \in \mathit{Wf}(t)]$ so that $(\forall x \in S : I(x) \Rightarrow x \in \mathit{Wf}(t))]$ is obvious. We must also show that $\forall s \in S : [\forall s' \in S : s \xrightarrow{t} s' \Rightarrow I(s')] \Rightarrow I(s)$ which holds since $\neg I(s) = [\exists \sigma' \in \omega \mapsto S : s = \sigma'_0 \wedge \forall i \in \omega : \sigma'_i \xrightarrow{t} \sigma'_{i+1}]$ implies $[\exists s' \in S : s \xrightarrow{t} s' \wedge (\exists \sigma \in \omega \mapsto S : s' = \sigma_0 \wedge \forall i \in \omega : \sigma_i \xrightarrow{t} \sigma_{i+1})] = [\exists s' \in S : s \xrightarrow{t} s' \wedge \neg I(s')]$ by choosing $s' = \sigma_0 = \sigma'_1$ and $\sigma_i = \sigma'_{i+1}$ for all $i \in \omega$.
- To show that $\mathit{Wf}(t) \subseteq |\Im^+|$, we use proposition 38 with $W = \mathit{Wf}(t)$ and $s' \prec s = s \xrightarrow{t} s'$ so that obviously $\mathit{WellFounded}(W, \prec)$ holds and $I(x) = \{x\}$ so that $[\forall u \in \mathit{Wf}(t) : \exists x \in W : u \in I(x))]$ is true when choosing $x = u$. Moreover $s \xrightarrow{t} s'$ implies $s' \prec s$ so that $[\forall x \in W : \forall c \in I(x) : \exists \frac{P}{c} \in \Phi : \forall c' \in P : \exists x' \prec x : c' \in I(x')]$ holds. $\square$

Example 40 (Floyd's partial correctness and termination proof methods) Let $T = \langle \mathrm{S}, \mathit{Act}, \{\xrightarrow{\alpha} \mid \alpha \in \mathit{Act}\}, \mathit{Init}\rangle$ be a labelled transition system specified by a program P. Let $\Psi \in S \times S \mapsto \{tt, ff\}$ be an input-output specification of program P. P is partially correct with respect to Ψ if and only if $\forall s, s' \in S : s \xrightarrow{*} s' \Rightarrow \Psi(s, s')$ (where $\xrightarrow{*}$ is the reflexive transitive closure of $\rightarrow$). According to proposition 37, the partial correctness proof can be done by discovering an invariant $I \subseteq U = S \times S$ satisfying the following verification conditions (as given in [11]):

$$\forall s \in S : \langle s, s\rangle \in I \quad (21)$$
$$\forall s, s', s'' \in S : \forall \alpha \in \mathit{Act} : [\langle s, s'\rangle \in I \wedge s' \xrightarrow{\alpha} s''] \Rightarrow \langle s, s''\rangle \in I \quad (22)$$
$$\forall s, s' \in S : [\langle s, s'\rangle \in I] \Rightarrow \Psi(s, s') \quad (23)$$

Let $\varepsilon \in S \mapsto \{tt, ff\}$ be a specification of the initial states of program P. Program P terminates if and only if there is no infinite execution trace $\sigma_0 \rightarrow \sigma_1 \rightarrow \sigma_2 \rightarrow \ldots$, such that $\varepsilon(\sigma_0)$. According to example 39, we must show that: $\forall s \in S : \varepsilon(s) \Rightarrow s \in \mathit{Wf}(\rightarrow)$. By proposition 38 applied to the rule-based inductive definition:

$$\frac{\{s' \in S \mid s \rightarrow s'\}}{s} \quad \forall s \in S \quad (24)$$

of $\mathit{Wf}(\rightarrow)$, this can be done by finding a well-founded set $\langle W, \prec\rangle$ and an invariant $I \in W \mapsto \wp(S)$ satisfying the following verification conditions (as given in [12]):

$$\forall \eta \in W : \forall s \in I^\eta : \forall s' \in S : [s \rightarrow s'] \Rightarrow [\exists \xi \prec \eta : s' \in I^\xi] \quad (25)$$
$$\forall s \in S : \varepsilon(s) \Rightarrow [\exists \eta \in W : s \in I^\eta] \quad (26)$$

$\square$

1.6 Well-founded system of inductive definitions

To get more expressive power, we introduce systems of inductive definitions in order to define subsets of a cartesian product $\prod_{i \in \Delta} U[i]$ of sets $U[i], i \in \Delta$. We mix this notion with transfinitely iterated definitions by induction on a well-founded set $\langle W, \multimap\!\!\prec\rangle$. Such well-founded (also called iterated) inductive definitions were first introduced by G. Kreisel and then further developped by S. Fefermann [15].

Definition 41 (Well-founded system of inductive definitions) A *well-founded system of inductive definitions* $\Im$ is a tuple $\langle W, \multimap\!\!\prec, \Delta, U, \Phi, \bot, \sqcup\rangle$ such that:

- $\multimap\!\!\prec$ is a well-founded binary relation on the set W,
- The *index* Δ is such that for all $w \in W$, $\Delta[\![w]\!]$ is a set,
- For all $w \in W$ and all $i \in \Delta[\![w]\!]$, the *universe* $U[\![w]\!][i]$ is a set,

- Φ is a set of *rules instances* $\frac{P \;\; \pi \;\; \nu}{c[\![w]\!][i]}$ where $w \in W$, $i \in \Delta[\![w]\!]$, $P \in \prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i])$, $\pi, \nu \in \{\langle w', S\rangle \mid w' \multimap\!\!\prec w \wedge S \in \prod_{i \in \Delta[\![w']\!]} \wp(U[\![w']\!][i])\}$ and $c \in U[\![w]\!][i]$,
- For all $w \in W$, $\bot[\![w]\!] \in \prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i])$ is the *basis*,
- For all $w \in W$, $\sqcup[\![w]\!] \in \wp(\prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i])) \hookrightarrow \prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i])$ is the *join*.

$\square$

Definition 42 (Operator induced by a well-founded system of inductive definitions) For each $w \in W$, the *operator* $\overline{\Phi}[\![w]\!]$ induced by $\Im = \langle W, \multimap\!\!\prec, \Delta, U, \Phi, \bot, \sqcup\rangle$ is $\overline{\Phi}[\![w]\!] \in \prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i]) \hookrightarrow \prod_{i \in \Delta[\![w]\!]} \wp(U[\![w]\!][i])$ such that:

$$\begin{aligned}\overline{\Phi}[\![w]\!](X) = \textstyle\prod_{i \in \Delta[\![w]\!]} \{\, & c \in U[\![w]\!][i] \mid \exists P \subseteq X : \\ & \exists \pi \subseteq \{\langle w', S\rangle \mid w' \multimap\!\!\prec w \wedge S \subseteq |\Im|[\![w']\!]\} : \\ & \quad \exists \nu \subseteq \{\langle w', S\rangle \mid w' \multimap\!\!\prec w \wedge S \not\subseteq |\Im|[\![w']\!]\} : \\ & \qquad \tfrac{P \;\; \pi \;\; \nu}{c[\![w]\!][i]} \in \Phi \,\}\end{aligned}$$

where $\forall i \in \Delta[\![w']\!] : |\Im|[\![w']\!][i] \overset{\text{def}}{=} \overline{\Phi}[\![w']\!]^{\infty}_{\sqcup[\![w']\!]}(\bot[\![w']\!])[i]$.
$\square$

Definition 43 (Well-formed well-founded system of inductive definitions) The well-founded system of inductive definitions $\Im = \langle W, \multimap\!\!\prec, \Delta, U, \Phi, \bot, \sqcup\rangle$ is *well-formed* if and only if $\overline{\Phi}[\![w]\!]^{\infty}_{\sqcup[\![w]\!]}(\bot[\![w]\!])$ exists for all $w \in W$, in which case it is the *cartesian product* $|\Im|[\![w]\!]$ *of rank* w *of sets* $|\Im|[\![w]\!][i]$ *of index* $i \in \Delta[\![w]\!]$ *inductively defined by* $\Im$. $\square$

If each $\wp(U[\![w]\!][i])\langle\sqsubseteq[\![w]\!][i], \bot[\![w]\!][i], \sqcup[\![w]\!][i]\rangle$ is a cpo (or a complete lattice) then propositions 9, 11, 13 and the above proof methods are easily extended to well-founded systems of inductive definitions by componentwise induction on the well-founded relation $\langle W, \multimap\!\!\prec\rangle$.

When defining subsets of a cartesian product of sets, the well-founded set $\langle W, \multimap\!\!\prec\rangle$ can be omitted since it is reduced to a singleton. This would be the case for example when understanding a context-free grammar as a system of inductive definitions. The next example is a well-founded inductive definition (the index set Δ being omitted since it is reduced to a singleton).

Example 44 (Weak fairness) Let A be an alphabet and X be a finite subset of A. The set $F[\![X]\!]$ of finite words containing at least one occurrence of each $x \in X$ is defined by:

$$\frac{y \in A}{y \in F[\![\emptyset]\!]}\,{}^{+} \qquad\qquad \frac{x \in X}{x \in F[\![\{x\}]\!]}\,{}^{+} \tag{27}$$

$$\frac{y \in A \quad \wedge \quad Y \subseteq X \quad \wedge \quad \sigma \in F[\![Y]\!]}{y\sigma \in F[\![Y]\!]}\,{}^{+} \tag{28}$$

$$\frac{x \in X \quad \wedge \quad Y \subseteq X \quad \wedge \quad \sigma \in F[\![Y]\!]}{x\sigma \in F[\![Y \cup \{x\}]\!]}\,{}^{+} \tag{29}$$

The set $I[\![X]\!]$ of infinite words with infinitely many occurrences of each $x \in X$ is defined by:

$$\frac{\sigma \in F[\![X]\!] \quad \wedge \quad \varsigma \in I[\![X]\!]}{\sigma\varsigma \in I[\![X]\!]}\,{}^{-} \tag{30}$$

This is a very simple example with $W = \{F[\![Y]\!] \mid Y \subseteq X\} \cup \{I[\![X]\!]\}$, $F[\![Y]\!] \multimap\!\!\prec F[\![Z]\!]$ when $Y \subset Z \subseteq X$ and $F[\![Y]\!] \multimap\!\!\prec I[\![X]\!]$ when $Y \subseteq X$. $\square$

Negation can be used in the premises of the rules of iterated definitions: a set $X[\![\lambda]\!]$ can be inductively defined in terms of $X[\![\lambda]\!]$ as well as $X[\![\alpha]\!]$ and $\neg X[\![\alpha]\!]$ for $\alpha \multimap\!\!\prec \lambda$. In particular this generalizes J. Groote's transition system specifications with negative premises [16] without resorting to 3-valued minimal model or stable model approaches [6] originating from logic programming, for which the logical meaning is not always simple and clear [19].

Combining systems and well-founded inductive definitions, we get well-founded systems of inductive definitions as illustrated by the following example:

Example 45 (Maximal execution traces III) Let $T = \langle \mathrm{S}, Act, \{\xrightarrow{\alpha} \mid \alpha \in Act\}, Init\rangle$ be a labelled transition system.

The set $T[\![\lambda]\!]$ ($T[\![{<}\lambda]\!]$ and $T[\![{\le}\lambda]\!]$), $\lambda \le \omega$, of maximal execution traces of length λ (respectively strictly less than λ and less than or equal to λ) can be defined as follows ($s, s', s'' \in S$, $0 \le n < \omega$, $\lambda \le \omega$):

$$\frac{s \not\longrightarrow}{s \in T[\![\cdot 0 \cdot]\!][s, s]}\,{}^{+} \tag{31}$$

$$\frac{s \xrightarrow{\alpha} s' \quad \wedge \quad \sigma \in T[\![\cdot n \cdot]\!][s', s'']}{s \xrightarrow{\alpha} \sigma \in T[\![\cdot n+1 \cdot]\!][s, s'']}\,{}^{+} \tag{32}$$

$$\frac{s \xrightarrow{\alpha} s' \quad \wedge \quad \sigma \in T[\![\cdot \omega]\!][s'] \quad \wedge \quad \overline{\sigma}_0 = s'}{s \xrightarrow{\alpha} \sigma \in T[\![\cdot \omega]\!][s]}\,{}^{-} \tag{33}$$

$$\frac{s \in Init \quad \wedge \quad \sigma \in T[\![\cdot n \cdot]\!][s, s']}{\sigma \in T[\![n]\!]}\,{}^{+} \tag{34}$$

$$\frac{s \in Init \quad \wedge \quad \sigma \in T[\![\cdot \omega]\!][s]}{\sigma \in T[\![\omega]\!]}\,{}^{+} \tag{35}$$

$$\frac{n < \lambda \quad \wedge \quad \sigma \in T[\![n]\!]}{\sigma \in T[\![{<}\lambda]\!]}\,{}^{+} \tag{36}$$

$$\frac{\sigma \in T[\![{<}\lambda]\!]}{\sigma \in T[\![{\le}\lambda]\!]}\,{}^{+} \qquad\qquad \frac{\sigma \in T[\![\lambda]\!]}{\sigma \in T[\![{\le}\lambda]\!]}\,{}^{+} \tag{37}$$

The partial ordering $\multimap\!\!\prec$ such that for all $0 \le n < \lambda \le \omega$, $\cdot n \cdot \multimap\!\!\prec \cdot n+1 \cdot$, $\cdot n \cdot \multimap\!\!\prec n$, $\cdot \omega \multimap\!\!\prec \omega$, $n \multimap\!\!\prec {<}\lambda$, $\lambda \multimap\!\!\prec {\le}\lambda$, ${<}\lambda \multimap\!\!\prec {\le}\lambda$, is well-founded. $\square$

Such iterated inductive definitions are very powerful since they have an expressive power greater than D. Park's μ-calculus [26] or E. Emerson's CTL* [14].

These iterated inductive definitions are used in G$^\infty$SOS to define the semantics of programming languages by induction on the syntax of programs. In this case, W is the set of all program components while $\multimap\!\!\prec$ means "is a sub-component of".

2 Abstract interpretation of inductive definitions

The quest for a unique general-purpose semantics for programming languages has failed. A better approach is to establish correspondences between various semantics at different levels of abstraction. As noticed by E. Astesiano [4] and G. Reggio [29], rule-based inductive definitions should form a unifying framework for expressing these semantics. A first step toward this goal is to describe finite and infinite program behaviors in the same way. A second step consists in adopting an abstract interpretation approach [8] [10] in order to relate inductive definitions. Abstract interpretation can be used, as follows, to justify and even to formally construct abstract rules in terms of a concrete *ground* (named *static* in [8] and renamed *collecting* in [24]) *semantics* (which could involve e.g. execution traces as it is the case in our examples):

Definition 46 (Operator abstraction) $\prec$ standing either for $=$ or $\sqsubseteq'$, we write:

$$[S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,F]\ \xrightarrow{\alpha[\prec]}\ [S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,F'] \tag{38}$$

to mean that:

- $S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle$ is a cpo such that $\bot \in S$,
- $S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle$ is a partial order such that $\bot' \in S'$,
- $F \in S \overset{mo\,\sqsubseteq}{\longmapsto} S$, $F' \in S' \overset{mo\,\sqsubseteq'}{\longmapsto} S'$,
- $\alpha \in S \mapsto S'$ and
- $\forall\lambda \leq clo^{\sqcup}_{\bot}(F) : \alpha(F^{\lambda}) \prec F'^{\lambda}$ (iteration from $F^0 = \bot$ and $F'^0 = \bot'$).

□

The above condition $\forall\lambda \leq clo^{\sqcup}_{\bot}(F) : \alpha(F^{\lambda}) \prec F'^{\lambda}$ is implied by the following ones:

- $\alpha(\bot) \prec \bot'$,
- $\forall\lambda \leq clo^{\sqcup}_{\bot}(F) : \alpha \circ F(F^{\lambda}) \prec F' \circ \alpha(F^{\lambda})$ and
- for all limit ordinals $\lambda \in \mathit{Limit}$ such that $\lambda \leq clo^{\sqcup}_{\bot}(F)$, we have $\alpha(\sqcup_{\delta<\lambda}F^{\delta}) \prec \sqcup'_{\delta<\lambda}\alpha(F^{\delta})$.

and by the strongest ones:

- $\alpha(\bot) \prec \bot'$,
- $\alpha \circ F \prec F' \circ \alpha$ and
- $\alpha(\sqcup_{\delta<\lambda}x^{\delta}) \prec \sqcup'_{\delta<\lambda}\alpha(x^{\delta})$ for all increasing chains $\{x^{\delta} \mid \delta \leq clo^{\sqcup}_{\bot}(F)\}$.

This last condition is implied by the fact that α is a complete join morphism, which is the case when it is the upper adjoint of a Galois connection:

Definition 47 (Galois connection) We write $P\,\langle\leq\rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} Q\,\langle\preceq\rangle$ to mean that:

- $P\,\langle\leq\rangle$ and $Q\,\langle\preceq\rangle$ are posets,
- $\alpha \in P \mapsto Q$,
- $\gamma \in Q \mapsto P$ and
- $\forall x \in P : \forall y \in Q : [\alpha(x) \preceq y] \Leftrightarrow [x \leq \gamma(y)]$.

□

In this case, we write $[S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,F] \underset{\gamma}{\overset{\alpha[\prec]}{\Longleftrightarrow}} [S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,F']$.

Proposition 48 If $[S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,F] \xrightarrow{\alpha[\prec]} [S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,F']$ then $\alpha(\mathit{lfp}^{\sqsubseteq}_{\bot}\,F) \prec \mathit{lfp}^{\sqsubseteq'}_{\bot'}\,F'$ where $\prec$ stands either for $=$ or $\sqsubseteq'$. □

Abstractions are usually specified by successive compositions:

Proposition 49 If $[S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,F] \xrightarrow{\alpha_1[\prec_1]} [S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,F']$ and $[S'\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,F'] \xrightarrow{\alpha_2[\prec_2]} [S''\,\langle\sqsubseteq'',\,\bot'',\,\sqcup''\rangle,\,F'']$ then $[S\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,F] \xrightarrow{\alpha_2\circ\alpha_1[\prec_2]} [S''\,\langle\sqsubseteq'',\,\bot'',\,\sqcup''\rangle,\,F'']$ where $\prec_1$ (respectively $\prec_2$) stands either for $=$ (resp. $=$) or $\sqsubseteq'$ (resp. $\sqsubseteq''$). □

Definition 50 (Inductive definition abstraction) If $\Im = \langle U,\,\overline{\Phi},\,\bot,\,\sqcup\rangle$ and $\Im' = \langle U',\,\overline{\Phi}',\,\bot',\,\sqcup'\rangle$ then we write $\Im \xrightarrow{\alpha[\prec]} \Im'$ for $[\wp(U)\,\langle\sqsubseteq,\,\bot,\,\sqcup\rangle,\,\overline{\Phi}] \xrightarrow{\alpha[\prec]} [\wp(U')\,\langle\sqsubseteq',\,\bot',\,\sqcup'\rangle,\,\overline{\Phi'}]$ where $\sqsubseteq$ and $\sqsubseteq'$ are the partial orderings respectively induced by $\sqcup$ and $\sqcup'$ and $\prec$ stands either for $=$ or $\sqsubseteq'$.

If moreover $\wp(U)\,\langle\sqsubseteq\rangle \underset{\gamma}{\overset{\alpha}{\rightleftarrows}} \wp(U')\,\langle\sqsubseteq'\rangle$ then we write $\Im \underset{\gamma}{\overset{\alpha[\prec]}{\Longleftrightarrow}} \Im'$. □

Corollary 51 If $\Im \xrightarrow{\alpha[\prec]} \Im'$ then $\alpha(|\Im|) \prec |\Im'|$ where $\prec$ stands either for $=$ or $\sqsubseteq'$. □

Example 52 (Disjoined bi-inductive definition) A positive inductive definition $\Im^+ = \langle U^+,\,\Phi^+\rangle^+$ and a negative one $\Im^- = \langle U^-,\,\Phi^-\rangle^-$ such that $U^+ \cap U^- = \emptyset$ can be combined into a system of inductive definitions $\Im = \langle\Delta,\,U,\,\Phi,\,\bot,\,\sqcup\rangle$ where $\Delta = \{1,\,2\}$, $U = U^+ \times U^-$, $\Phi = \left\{\frac{\langle P,\,\emptyset\rangle}{c[1]}\,\middle|\,\frac{P}{c} \in \Phi^+\right\} \cup \left\{\frac{\langle\emptyset,\,P\rangle}{c[2]}\,\middle|\,\frac{P}{c} \in \Phi^-\right\}$, $\bot = \langle\emptyset,\,U^-\rangle$ and $\sqcup_{i\in S}\langle X_i,\,Y_i\rangle = \langle\cup_{i\in S}X_i,\,\cap_{i\in S}Y_i\rangle$. They can also be combined into a disjoined bi-inductive definition $\Im^{\pm} = \langle U^+,\,U^-,\,\ \Phi^- \cup \Phi^+\rangle$. The bijection is established by $\alpha(\langle X,\,Y\rangle) = X \cup Y$ and $\gamma(X) = \langle X \cap U^+,\,X \cap U^-\rangle$, so that by propositions 27 and 51, we have $|\Im^+| \cup |\Im^-| = |\Im^{\pm}| = \alpha(|\Im|)$ □

A simple way to abstract inductive definitions is to use an abstraction of subsets of the universe:

Proposition 53 Let $\Im = \langle U,\,\Phi,\,\bot,\,\sqcup\rangle$ be a monotonic inductive definition on a cpo $\wp(U)\langle\sqsubseteq,\,\bot,\,\sqcup\rangle$, $\wp(U')\langle\sqsubseteq',\,\sqcup'\rangle$ be a partial order and $\alpha \in \wp(U) \mapsto \wp(U')$ be a complete $\sqcup$-morphism (for all sets $\{\overline{\Phi}^{\beta} \mid \beta < \lambda\}$, $\lambda \in \mathit{Ord}$) and a complete $\cup$-morphism such that:

$$\begin{aligned}\forall\tfrac{P}{c} \in \Phi : \forall X \subseteq U : \alpha(P) \subseteq \alpha(X) \Rightarrow\\ \exists\tfrac{P'}{c'} \in \Phi : P' \subseteq X \wedge \alpha(\{c\}) \subseteq \alpha(\{c'\})\end{aligned} \tag{39}$$

Define $\Phi' \overset{\text{def}}{=} \left\{\frac{\alpha(P)}{c'}\,\middle|\,\frac{P}{c} \in \Phi\ \wedge\ c' \in \alpha(\{c\})\right\}$ and $\bot' \overset{\text{def}}{=} \alpha(\bot)$. Then $\Im' \overset{\text{def}}{=} \langle\alpha(U),\,\Phi',\,\bot',\,\sqcup'\rangle$ is a well formed inductive definition such that $\Im \xrightarrow{\alpha[=]} \Im'$. If moreover $\wp(U)\langle\sqsubseteq,\,\bot,\,\top,\,\sqcup,\,\sqcap\rangle$ is a complete lattice then $\Im \underset{\gamma}{\overset{\alpha[=]}{\Longleftrightarrow}} \Im'$ where $\gamma \in \wp(U') \mapsto \wp(U)$ is $\gamma(Y) = \sqcup\{X \in U \mid \alpha(X) \sqsubseteq' Y\}$. □

This notion of abstraction can be used to show the equivalence of inductive definitions.

Example 54 (Maximal execution traces IV) In example 21 we have characterized the execution traces $T^{\leq\omega}$ of a transition system $T = \langle S,\ Act,\ \{\xrightarrow{\alpha} \mid \alpha \in Act\},\ Init\rangle$ by a negative inductive definition $\Im^- = \langle \Sigma^{\leq\omega},\ \Phi^-,\ \Sigma^{\leq\omega},\ \cap\rangle$ where Φ^- is specified by schema (12) and $\wp(\Sigma^{\leq\omega})\langle\supseteq,\ \Sigma^{\leq\omega},\ \cap\rangle$ is the induced cpo. In example 28 we have characterized $T^{\leq\omega}$ by a disjoined inductive definition $\Im^\pm = \langle\Sigma^{<\omega}, \Sigma^\omega, \Phi^\pm\rangle^\pm$ where $\Phi^\pm$ is specified by schemata (14) and (15) and $\wp(\Sigma^{\leq\omega})\langle\sqsubseteq^\pm,\ \bot^\pm,\ \sqcup^\pm\rangle$ is the induced cpo.

In order to relate them, we observe that they are abstractions of a common ground semantics:

$$[\Sigma^{\leq\omega} \times (\omega + 1)\langle\sqsubseteq,\ \bot,\ \sqcup\rangle,\ \phi] \tag{40}$$

such that $\langle X,\ \lambda\rangle \sqsubseteq \langle X',\ \lambda'\rangle \stackrel{\text{def}}{=} (X \sqsubseteq^\pm X') \wedge (\lambda \leq \lambda')$, $\bot \stackrel{\text{def}}{=} \langle\bot^\pm,\ 0\rangle$, $\sqcup_i\langle X_i,\ \lambda_i\rangle \stackrel{\text{def}}{=} \langle\sqcup^\pm_i X_i,\ \ max_i\lambda_i\rangle$ where the set ω of natural numbers is the supremum of the set $\omega + 1 = \omega \cup \{\omega\}$ and $\phi(\langle X,\ \lambda\rangle) = \langle\overline{\Phi^\pm}(X),\ 1 + \lambda\rangle$ with $1 + \omega = \omega$.

We have $[\Sigma^{\leq\omega} \times (\omega + 1)\langle\sqsubseteq,\ \bot,\ \sqcup\rangle,\ \phi] \overset{\alpha^\pm[=]}{\underset{\gamma^\pm}{\Longleftrightarrow}} [\Sigma^{\leq\omega}\langle\sqsubseteq^\pm,\ \bot^\pm,\ \sqcup^\pm\rangle,\ \overline{\phi^\pm}]$ where $\alpha^\pm(\langle X,\ \lambda\rangle) = X$ and $\gamma^\pm(X) = \langle X,\ \omega\rangle$. By proposition 48, it follows that $\alpha^\pm(lfp^{\sqsubseteq}_{\bot}\,\phi) = lfp^{\sqsubseteq^\pm}_{\bot^\pm}\,\overline{\phi^\pm} = |\Im^\pm|$.

We also have the correspondence $[\Sigma^{\leq\omega} \times (\omega + 1)\langle\sqsubseteq,\ \bot,\ \sqcup\rangle,\ \phi] \xrightarrow{\alpha^-[=]} [\Sigma^{\leq\omega}\langle\supseteq,\ \Sigma^{\leq\omega},\ \cap\rangle,\ \overline{\phi^-}]$ where $\alpha^-(\langle X, \lambda\rangle) = X \cup (\cup_{m>\lambda} T^{\lambda m})$ and $\cup_{m>\lambda} T^{\lambda m}$ has been defined at example 21 for $\lambda \leq \omega$ and is equal to $\emptyset$ when $\lambda = \omega$. By proposition 48, $\alpha^-(lfp^{\sqsubseteq}_{\bot}\,\phi) = lfp^{\supseteq}_{\Sigma^{\leq\omega}}\,\overline{\phi^-} = |\Im^-|$. $\square$

Abstraction can also be used to relate definitions of the semantics of languages described at different levels of details.

Example 55 (Erratic, demoniac and angelic relational semantics of transition systems)
• The *erratic relational semantics* is obtained by abstraction of $T^{\leq\omega}$ as characterized by the disjoined inductive definition $\Im^\pm = \langle\Sigma^{<\omega},\ \Sigma^\omega,\ \ \Phi^\pm\rangle^\pm$ of example 28, where $\Phi^\pm$ is specified by schemata (14) and (15) and $\wp(U)\langle\sqsubseteq^\pm,\ \bot^\pm,\ \sqcup^\pm\rangle$ (where $U = \Sigma^{\leq\omega}$) is the induced cpo. The finite traces are approximated by the pair of their initial and final states and the infinite ones by their initial state together with $\bot$ denoting non-termination. Let us define $U^{\natural+} \stackrel{\text{def}}{=} S \times S$, $U^{\natural-} \stackrel{\text{def}}{=} S \times \{\bot\}$ where $\bot \notin S$ and $U^\natural \stackrel{\text{def}}{=} U^{\natural+} \cup U^{\natural-}$. By propositions 25.1 and 27, $\wp(U^\natural)\langle\sqsubseteq^\natural,\ \bot^\natural,\ \top^\natural,\ \sqcup^\natural,\ \sqcap^\natural\rangle$ is a complete lattice, where:

$$\pi^{\natural+}(r) \quad \stackrel{\text{def}}{=} \quad r \cap (S \times S) \tag{41}$$

$$\pi^{\natural-}(r) \quad \stackrel{\text{def}}{=} \quad r \cap (S \times \{\bot\}) \tag{42}$$

$$t \sqsubseteq^\natural r \quad \stackrel{\text{def}}{=} \quad \begin{array}{l}[\pi^{\natural+}(t) \subseteq \pi^{\natural+}(r)] \wedge \\ [\pi^{\natural-}(t) \supseteq \pi^{\natural-}(r)]\end{array} \tag{43}$$

$$\bot^\natural \quad \stackrel{\text{def}}{=} \quad S \times \{\bot\} \tag{44}$$

$$\sqcup^\natural_i r_i \quad \stackrel{\text{def}}{=} \quad \cup_i \pi^{\natural+}(r_i)\ \cup\ \cap_i \pi^{\natural-}(r_i) \tag{45}$$

The approximation can be formalized by the erratic abstraction $\alpha^\natural \in \wp(U)\langle\sqsubseteq^\pm\rangle \longmapsto \wp(U^\natural)\langle\sqsubseteq^\natural\rangle$ such that:

$$\begin{array}{rcl}\alpha^\natural(X) & = & \{\langle\overline{\sigma}_0,\ \overline{\sigma}_{|\sigma|-1}\rangle \mid \sigma \in X \cap \Sigma^{<\omega}\} \\ & & \cup\ \{\langle\overline{\sigma}_0,\ \bot\rangle \mid \sigma \in X \cap \Sigma^\omega\}\end{array} \tag{46}$$

Let $\Im^\natural = \langle U^\natural,\ \Phi^\natural,\ \bot^\natural,\ \sqcup^\natural\rangle$ be the inductive definition such that the abstract rules instances $\Phi^\natural$ are given by the following schemata (where $s, s' \in S$ and $x \in S \cup \{\bot\}$):

$$\frac{s \not\longrightarrow}{\langle s,\ s\rangle \in T^\natural}^{+} \qquad \frac{s \xrightarrow{\alpha} s' \ \wedge\ \langle s',\ x\rangle \in T^\natural}{\langle s,\ x\rangle \in T^\natural}^{\pm} \tag{47}$$

Then proposition 53 implies $\Im^\pm \overset{\alpha^\natural[=]}{\underset{\gamma^\natural}{\Longleftrightarrow}} \Im^\natural$ since $\alpha^\natural(\cup^\pm_i X_i) = \cup^\natural_i \alpha^\natural(X_i)$, $\alpha^\natural(\sqcup^\pm_i X_i) = \sqcup^\natural_i \alpha^\natural(X_i)$, property (39) holds, $\alpha^\natural(\bot^\pm) = \bot^\natural$ and $\alpha^\natural(U) = U^\natural$.
• The *angelic relational semantics* is obtained by abstraction of finite traces by the pair of their initial and final states while infinite ones are simply ignored. This can be formalized by $U^\flat = S \times S$ and the angelic abstraction:

$$\wp(U^\natural)\langle\sqsubseteq^\natural\rangle \overset{\alpha^\flat}{\underset{\gamma^\flat}{\longleftrightarrow}} \wp(U^\flat)\langle\subseteq\rangle \tag{48}$$

such that:

$$\alpha^\flat(X) \quad = \quad X \cap U^\flat \tag{49}$$

$$\gamma^\flat(X) \quad = \quad X \tag{50}$$

Let $\Im^\flat = \langle U^\flat,\ \Phi^\flat,\ \emptyset,\ \cup\rangle$ be the inductive definition such that the abstract rules instances $\Phi^\flat$ are given by the following schemata (where $s, s', s'' \in S$):

$$\frac{s \not\longrightarrow}{\langle s,\ s\rangle \in T^\flat}^{+} \qquad \frac{s \xrightarrow{\alpha} s' \ \wedge\ \langle s',\ s''\rangle \in T^\flat}{\langle s,\ s''\rangle \in T^\flat}^{+} \tag{51}$$

Then proposition 53 implies $\Im^\natural \overset{\alpha^\flat[=]}{\underset{\gamma^\flat}{\Longleftrightarrow}} \Im^\flat$ since α is a complete $\cup$-morphism, property (39) holds, $\alpha^\flat(U^\natural) = U^\flat$ and $\alpha^\flat(\bot^\natural) = \bot^\flat = \emptyset$. By proposition 49, we conclude $\Im^\pm \overset{\alpha^\flat\circ\alpha^\natural[=]}{\underset{\gamma^\natural\circ\gamma^\flat}{\Longleftrightarrow}} \Im^\flat$.
• The *demoniac relational semantics* is obtained by abstraction of $T^{\leq\omega}$ as characterized by the negative inductive definition $\Im^- = \langle\Sigma^{\leq\omega},\ \Phi^-,\ \Sigma^{\leq\omega},\ \cap\rangle$ of example 21 where Φ^- is specified by schema (12) and $\wp(\Sigma^{\leq\omega})\langle\supseteq,\ \Sigma^{\leq\omega},\ \cap\rangle$ is the induced cpo. Finite traces are approximated by the pair of their initial and final states and the infinite ones by their initial state together with $\bot$ denoting non-termination as well as any state so as to represent demoniac termination. Let us define the demoniac abstraction $\alpha^\sharp \in \wp(\Sigma^{\leq\omega})\langle\supseteq\rangle \longmapsto \wp(U^\sharp)\langle\supseteq\rangle$ where $U^\sharp = S \times (S \cup \{\bot\})$ such that:

$$\begin{array}{rl}\alpha^\sharp(X) = & \{\langle\overline{\sigma}_0,\ \overline{\sigma}_{|\sigma|-1}\rangle \mid \sigma \in X \cap \Sigma^{<\omega}\} \cup \\ & \{\langle\overline{\sigma}_0,\ x\rangle \mid \sigma \in X \cap \Sigma^\omega \ \wedge\ x \in S \cup \{\bot\}\}\end{array} \tag{52}$$

Let $\Im^\sharp = \langle U^\sharp,\ \Phi^\sharp,\ U^\sharp,\ \cap\rangle$ be the negative inductive definition such that the abstract rules instances $\Phi^\sharp$ are given by the following schemata (where $s, s', s'' \in S$ and $x \in S \cup \{\bot\}$):

$$\frac{s \not\longrightarrow}{\langle s,\ s\rangle \in T^\sharp}\,- \qquad \frac{s \xrightarrow{\alpha} s' \ \wedge\ \langle s',\ s''\rangle \in T^\sharp}{\langle s,\ s''\rangle \in T^\sharp}\,- \tag{53}$$

$$\frac{s \xrightarrow{\alpha} s' \ \wedge\ \forall y \in S \cup \{\bot\} : \langle s',\ y\rangle \in T^\sharp}{\langle s,\ x\rangle \in T^\sharp}\,- \tag{54}$$

Then proposition 53 implies $\Im^- \xrightarrow{\alpha^\sharp[=]} \Im^\sharp$. Then rule schema (54) can be simplified into:

$$\frac{s \xrightarrow{\alpha} s' \ \wedge\ \langle s',\ \bot\rangle \in T^\sharp}{\langle s,\ x\rangle \in T^\sharp}\,- \tag{55}$$

since $\bot \in T^\sharp$ implies $\forall y \in S \cup \{\bot\} : y \in T^\sharp$. □

By further abstractions one can derive powerdomains based state transformation semantics [3].

3 G$^\infty$SOS

We now introduce G$^\infty$SOS, a generalization of SOS (G. Plotkin's structured operational semantics [28]) using the above rule-based systems of transfinitely iterated inductive definitions. G$^\infty$SOS enables us to describe the finite, as well as the infinite executions of programs. The nature of the finitary or infinitary objects representing terminating and non-terminating program executions is not fixed and depends upon the considered language. For example we have defined the G$^\infty$SOS semantics of λ-calculi [27] using judgements $\rho \vdash E \rightsquigarrow \nu$ (expression E evaluates to ν in environment ρ) and $\rho \vdash E \rightsquigarrow \bot$ (evaluation of expression E does not terminate in environment ρ); K. Apt and G. Plotkin's nondeterministic language [3] using maximal finite and infinite execution traces and R. Milner's CCS using infinite synchronization trees [21] and partial orders [13] with infinite chains[2].

Example 56 (Trace semantics of while loops) In order to define the operational trace semantics of an imperative language, let $\rho \in \Gamma$ be an environment (recording the values of identifiers), $c \in C$ be a command, $s \in S = (\Gamma \times C) \cup \Gamma$ be a state written $\rho \vdash c$ or ρ, $U^+ = \Sigma^{<\omega}$ be the set of finite traces, $U^- = \Sigma^\omega$ be the set of infinite traces and $U = \Sigma^{\leq\omega} = \Sigma^{<\omega} \cup \Sigma^\omega$ be the set of traces over states S and actions C. For short assume that the evaluation of a boolean expression $b \in B$ in environment ρ always terminates without error and yields a boolean value $\mathcal{B}[\![b]\!]\rho$. If $\sigma \in \Sigma^{\leq\omega}$ and $c \in C$, define $\sigma \mathbin{\text{\textcircled{;}}} c$ by $\rho \mathbin{\text{\textcircled{;}}} c = \rho \vdash c$, $(\rho \vdash c) \mathbin{\text{\textcircled{;}}} c' = \rho \vdash (c; c')$ and $(\sigma \xrightarrow{\alpha} \sigma') \mathbin{\text{\textcircled{;}}} c = (\sigma \mathbin{\text{\textcircled{;}}} c) \xrightarrow{\alpha} (\sigma' \mathbin{\text{\textcircled{;}}} c)$. The trace semantics of the while loop $w \equiv$ **while** b **do** c is the set $\mathcal{T}[\![w]\!][\rho] \subseteq \Sigma^{\leq\omega}$ defined by the following rule schemata:

• Execution of the while loop w terminates immediately when the test b evaluates to false:

$$\frac{\neg\mathcal{B}[\![b]\!]\rho}{\rho \vdash w \xrightarrow{b} \rho \ \in\ \mathcal{T}[\![w]\!][\rho]}\,+ \tag{56}$$

• A terminating or non-terminating execution of the while loop starts with a first step $\rho \vdash w \xrightarrow{b} \rho \vdash c; w$ to evaluate the test b to true followed by an execution $\rho \vdash c \xrightarrow{\alpha} \sigma \xrightarrow{\alpha'} \rho'$ of the loop body c (where control states memorize the fact that further iterations of w may be necessary: $(\rho \vdash c \xrightarrow{\alpha} \sigma \xrightarrow{\alpha'} \rho') \mathbin{\text{\textcircled{;}}} w)$ followed by the remaining iterations $\rho' \vdash w \xrightarrow{\alpha''} \sigma'$:

$$\frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \ \wedge\ \rho \vdash c \xrightarrow{\alpha} \sigma \xrightarrow{\alpha'} \rho' \ \in\ \mathcal{T}[\![c]\!][\rho] \\ \wedge\ \rho' \vdash w \xrightarrow{\alpha''} \sigma' \ \in\ \mathcal{T}[\![w]\!][\rho']\end{array}}{\begin{array}{c}\rho \vdash w \xrightarrow{\mathrm{b}} \rho \vdash c; w \xrightarrow{\alpha} (\sigma \mathbin{\text{\textcircled{;}}} w) \xrightarrow{\alpha'} \rho' \vdash w \xrightarrow{\alpha''} \sigma' \\ \in \mathcal{T}[\![w]\!][\rho]\end{array}}\,\pm \tag{57}$$

• Execution of the while loop may also not terminate when the test b is true and execution of the body c never terminates:

$$\frac{\mathcal{B}[\![b]\!]\rho \ \wedge\ \rho \vdash c \xrightarrow{\alpha} \sigma \ \in\ \mathcal{T}[\![c]\!][\rho]}{\rho \vdash w \xrightarrow{\mathrm{b}} \rho \vdash c; w \xrightarrow{\alpha} (\sigma \mathbin{\text{\textcircled{;}}} w) \ \in\ \mathcal{T}[\![w]\!][\rho]}\,- \tag{58}$$

Observe that we proceed by *syntactic induction* on the well-founded ordering $c \multimap\!\!\!< c'$ iff c is a syntactic component of c'. For example in rule schema (57), $c \multimap\!\!\!< w \equiv$ **while** b **do** c. In the instances of rule (57) corresponding to finite execution traces, we use induction on the length of execution traces: execution of w in environment ρ' takes strictly less steps than its execution in environment ρ. This can also be understood as *action induction* [21], an induction upon the depth of the inference by which $\rho \vdash w \xrightarrow{\mathrm{b}} \rho \vdash c; w \xrightarrow{\alpha} (\sigma \mathbin{\text{\textcircled{;}}} w) \xrightarrow{\alpha'} \rho' \vdash w \xrightarrow{\alpha''} \sigma'$ is inferred. It is a sound principle just because lengths of finite executions ordered by $<$ are well-founded. This argument is no longer valid in the instance of rule (57) for infinite execution traces since non-terminating executions of w in environments ρ' and ρ both take infinitely many steps. This shows that negative rules are useful to provide a direct description of non-termination. □

Denotational semantics [23] has several important advantages over traditional operational semantics:

(1) The semantics of programs is given in terms of mathematical models (domain theory [17]);
(2) Denotations are defined by induction on the abstract syntax of programs;
(3) Finite and infinite program behaviors are handled in the same way (using fixpoints).

SOS copes with point (2) but not (3). One can define a big-steps SOS semantics where only error-free, terminating behaviors of programs are described. Alternatively, one can define a small-steps semantics where

[2]These examples, as others, are omitted for lack of space.

non-terminating executions have to be described using another formalism such as execution traces (see [3]). Both approaches are incomplete since the first describes the "good" cases and leaves out the "bad" ones while the second provides a microscopic view of a macroscopic process. Using both approaches simultaneously implies a lot of work to relate them, and this has to be done again and again for each language [4]. G$^\infty$SOS can cope with terminating and non-terminating executions whence the prefix G$^\infty$ added to SOS indicating the generalization to infinite behaviors. The correct handling of non-termination is necessary, for example to define fair executions of parallel processes or to serve as a ground semantics for the inference of liveness properties of programs by abstract interpretation such as strictness analysis. This abstraction process can also be used to define more abstract semantics as shown by the following:

Example 57 (Relational semantics of while loops) To obtain the big-step semantics of commands, we define an abstraction $\alpha \in \wp(\Sigma^{\leq\omega}) \mapsto \wp(\Gamma \times \Gamma_\perp)$ by $\alpha(\{\sigma_i \mid i \in \Delta\}) \stackrel{\text{def}}{=} \{\alpha(\sigma_i) \mid i \in \Delta\}$ where $\Gamma_\perp \stackrel{\text{def}}{=} \Gamma \cup \{\perp\}$, $\alpha \in \Sigma^{\leq\omega} \mapsto \Gamma \times \Gamma_\perp$ is given by $\alpha(\rho \vdash c \xrightarrow{\alpha'} \sigma \xrightarrow{\alpha''} \rho') \stackrel{\text{def}}{=} \langle\rho,\, \rho'\rangle$ for finite traces and by $\alpha(\rho \vdash c \xrightarrow{\alpha'} \sigma)$ $\stackrel{\text{def}}{=} \langle\rho,\, \perp\rangle$ for infinite traces. The *relational semantics* of $c \in C$ is then $\mathcal{R}[\![c]\!] = \alpha(\{\mathcal{T}[\![c]\!][\rho] \mid \rho \in \Gamma\})$. We write $\rho \vdash c \rightsquigarrow \rho'$ for $\langle\rho,\, \rho'\rangle \in \mathcal{R}[\![c]\!]$. The *natural semantics* [18] of commands is $\mathcal{R}[\![c]\!] \cap \Sigma^{<\omega}$ since it only deals with terminating executions. By proposition 53 applied to (56), (57), (58), it is defined for the while loop $w \equiv$ **while** b **do** c by the following rule schemata:

$$\frac{\neg\mathcal{B}[\![b]\!]\rho}{\rho \vdash w \rightsquigarrow \rho}^{+} \qquad \frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \wedge \\ \rho \vdash c \rightsquigarrow \rho' \wedge \rho' \vdash w \rightsquigarrow \rho''\end{array}}{\rho \vdash w \rightsquigarrow \rho''}^{+} \tag{59}$$

Using G$^\infty$SOS, non-termination can be expressed directly:

$$\frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \wedge \\ \rho \vdash c \rightsquigarrow \perp\end{array}}{\rho \vdash w \rightsquigarrow \perp}^{-} \qquad \frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \wedge \\ \rho \vdash c \rightsquigarrow \rho' \wedge \rho' \vdash w \rightsquigarrow \perp\end{array}}{\rho \vdash w \rightsquigarrow \perp}^{-} \tag{60}$$

□

Example 58 (G$^\infty$SOS semantics of the nondeterministic choice) To illustrate fairness, let us consider the simple case of the nondeterministic choice operator $[c_1 \Box c_2]$.

• With Plotkin's *erratic semantics*, termination or non-termination of $[c_1 \boxed{\scriptstyle E} c_2]$ is possible whenever that of c_1 or c_2 is possible:

$$\frac{\rho \vdash c_1 \rightsquigarrow \rho'}{\rho \vdash [c_1 \boxed{\scriptstyle E} c_2] \rightsquigarrow \rho'}^{+} \qquad \frac{\rho \vdash c_2 \rightsquigarrow \rho''}{\rho \vdash [c_1 \boxed{\scriptstyle E} c_2] \rightsquigarrow \rho''}^{+} \tag{61}$$

$$\frac{\rho \vdash c_1 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle E} c_2] \rightsquigarrow \perp}^{-} \qquad \frac{\rho \vdash c_2 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle E} c_2] \rightsquigarrow \perp}^{-} \tag{62}$$

• With Hoare's fair *angelic semantics*, termination is possible when execution of c_1 or that of c_2 may terminate whereas non-termination of $[c_1 \boxed{\scriptstyle A} c_2]$ requires that both c_1 and c_2 cannot terminate:

$$\frac{\rho \vdash c_1 \rightsquigarrow \rho'}{\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow \rho'}^{+} \qquad \frac{\rho \vdash c_2 \rightsquigarrow \rho''}{\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow \rho''}^{+} \tag{63}$$

$$\frac{\begin{array}{c}\rho \vdash c_1 \rightsquigarrow \perp \ \wedge\ \rho \vdash c_1 \not\rightsquigarrow \rho' \\ \wedge\ \rho \vdash c_2 \rightsquigarrow \perp \ \wedge\ \rho \vdash c_2 \not\rightsquigarrow \rho''\end{array}}{\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow \perp}^{-} \tag{64}$$

(As pointed out by M. Broy, if $\rho \vdash c_1 \rightsquigarrow 1$, $\rho \vdash c_1 \rightsquigarrow \perp$, $\rho \vdash c_2 \rightsquigarrow 2$ and $\rho \vdash c_1 \rightsquigarrow \perp$ then $\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow 1$, $\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow 2$ but $\rho \vdash [c_1 \boxed{\scriptstyle A} c_2] \rightsquigarrow \perp$ is not true, a miracle since $[c_1 \boxed{\scriptstyle A} c_2]$ must be able to avoid non-termination in the erratic behavior of both c_1 and c_2!)

• With McCarthy's fair *parallel semantics*, non-termination is possible when both executions of c_1 and c_2 may not terminate:

$$\frac{\rho \vdash c_1 \rightsquigarrow \rho'}{\rho \vdash [c_1 \boxed{\scriptstyle P} c_2] \rightsquigarrow \rho'}^{+} \qquad \frac{\rho \vdash c_2 \rightsquigarrow \rho''}{\rho \vdash [c_1 \boxed{\scriptstyle P} c_2] \rightsquigarrow \rho''}^{+} \tag{65}$$

$$\frac{\rho \vdash c_1 \rightsquigarrow \perp \quad \wedge \quad \rho \vdash c_2 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle P} c_2] \rightsquigarrow \perp}^{-}$$

• With Smyth's unfair *demoniac semantics*, termination of $[c_1 \boxed{\scriptstyle D} c_2]$ is possible only when both that of c_1 and c_2 are guaranteed whereas non-termination is possible whenever that of c_1 or c_2 is possible:

$$\frac{\rho \vdash c_1 \rightsquigarrow \rho' \ \wedge\ \rho \vdash c_1 \not\rightsquigarrow \perp \ \wedge\ \rho \vdash c_2 \not\rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle D} c_2] \rightsquigarrow \rho'}^{+} \tag{66}$$

$$\frac{\rho \vdash c_2 \rightsquigarrow \rho'' \ \wedge\ \rho \vdash c_1 \not\rightsquigarrow \perp \ \wedge\ \rho \vdash c_2 \not\rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle D} c_2] \rightsquigarrow \rho''}^{+} \tag{67}$$

$$\frac{\rho \vdash c_1 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle D} c_2] \rightsquigarrow \perp}^{-} \qquad \frac{\rho \vdash c_2 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle D} c_2] \rightsquigarrow \perp}^{-} \tag{68}$$

• With the unfair, à la Prolog, *left to right semantics*, termination of $[c_1 \boxed{\scriptstyle lr} c_2]$ is possible when that of c_1 is possible or when c_1 cannot diverge and c_2 may terminate. Non-termination of $[c_1 \boxed{\scriptstyle lr} c_2]$ is possible when that of c_1 is possible or when c_1 cannot diverge but c_2 can:

$$\frac{\rho \vdash c_1 \rightsquigarrow \rho'}{\rho \vdash [c_1 \boxed{\scriptstyle lr} c_2] \rightsquigarrow \rho'}^{+} \qquad \frac{\rho \vdash c_1 \not\rightsquigarrow \perp \ \wedge\ \rho \vdash c_2 \rightsquigarrow \rho''}{\rho \vdash [c_1 \boxed{\scriptstyle lr} c_2] \rightsquigarrow \rho''}^{+} \tag{69}$$

$$\frac{\rho \vdash c_1 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle lr} c_2] \rightsquigarrow \perp}^{-} \qquad \frac{\rho \vdash c_1 \not\rightsquigarrow \perp \ \wedge\ \rho \vdash c_2 \rightsquigarrow \perp}{\rho \vdash [c_1 \boxed{\scriptstyle lr} c_2] \rightsquigarrow \perp}^{-} \tag{70}$$

In rule schemata (63) to (70), the use of negations is sound since $c_1 \prec [c_1 \Box c_2]$ and $c_2 \prec [c_1 \Box c_2]$. □

This last example illustrates the *semantical compositionality* property: we can change the semantics of the nondeterministic choice command without having to change the semantics of other (while, if, ...)

(sub)commands. In denotational semantics, one will have to use Plotkin, Hoare, Smyth, ... powerdomains [17] to describe the behavior of the choice command. By doing so one will have to change the ordering used for computing fixpoints, hence potentially the semantics of other commands (such as while loops or recursion). More generally, semantical compositionality requires that the specification of the whole should be done without interfering with the specification of the components.

As a last abstraction, let us consider predicate transformers:

Example 59 (Predicate transformers) The predicate transformer of command $c \in C$ is $wp[\![c]\!] = \alpha(\mathcal{R}[\![c]\!])$ where the abstraction $\alpha \in \wp(\Gamma \times \Gamma_\perp) \mapsto (\wp(\Gamma) \mapsto \wp(\Gamma))$ is defined by $\alpha(R) = \lambda P.\{\rho \mid (\exists \rho' \in P : \langle \rho, \rho' \rangle \in R) \wedge (\forall \rho' : \langle \rho, \rho' \rangle \in R \Rightarrow \rho' \in P) \wedge (\langle \rho, \perp \rangle \notin R)\}$. Since $\mathcal{R}[\![c]\!] \neq \emptyset$, proposition 53 applied to (59) and (60) yields the following definition of $wp[\![w]\!]$ where $w \equiv$ **while** b **do** c:

$$\frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \wedge \\ \rho \in P\end{array}}{\rho \in wp[\![w]\!](P)} - \qquad \frac{\begin{array}{c}\mathcal{B}[\![b]\!]\rho \wedge \rho \in wp[\![c]\!](Q) \\ \wedge Q \subseteq wp[\![w]\!](P)\end{array}}{\rho \in wp[\![w]\!](P)} - \qquad (71)$$

□

Further abstractions would perfect the lattice of abstract interpretations considered in [10]. Abstract interpretation was first introduced using transition systems [7] that is an operational semantics. Mycroft [24], followed by Nielson [25], advocated using denotational instead of operational base semantics. We think that G$^\infty$SOS is better suited for designing ground semantics from which other, more abstract or approximate, semantics can be derived. In particular denotational semantics, which are abstract interpretations of operational behaviors, can be understood as intermediate steps in the approximation process.

Acknowledgements We would like to thank the members of IFIP WG 2.3 present at Santa Catalina and Pouilly en Auxois meetings for discussions.

References

[1] P. Aczel. An introduction to inductive definitions. In J. Barwise, ed., *Handbook of Mathematical Logic*, Elsevier, 739–782, 1977.

[2] K.R. Apt. Logic programming. In [31], 493–574.

[3] K.R. Apt & G.D. Plotkin. Countable nondeterminism and random assignment, *JACM* 33(4):724–767, 1986.

[4] E. Astesiano. Inductive semantics. In E.J. Neuhold & M. Paul, eds., *Formal Description of Programming Concepts*, Springer, 1990.

[5] J. Barwise. Mixed fixed points. In vol. 17 of *CSLI lecture notes*, 285–287, CSLI/Stanford, 1989.

[6] R.N. Bol & J.F. Groote. The meaning of negative premises in transition system specification. *LNCS* 510, 481–494, Springer, 1991.

[7] P. Cousot. Semantic foundations of program analysis. In S.S. Muchnick & N.D. Jones, eds., *Program Flow Analysis: Theory and Applications*, 303–342, Prentice-Hall, 1981.

[8] P. Cousot & R. Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *4th POPL*, 238–252, 1977.

[9] P. Cousot & R. Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific J. of Math.*, 82(1):43–57, 1979.

[10] P. Cousot & R. Cousot. Systematic design of program analysis frameworks. In *6th POPL*, 269–282, 1979.

[11] P. Cousot & R. Cousot. Induction principles for proving invariance properties of programs, In E. Neel, ed., *Tools and Notions for Program Construction*, 43–119, Cambridge Univ. Press, 1982.

[12] P. Cousot & R. Cousot. 'À la Floyd' induction principles for proving inevitability properties of programs. In M. Nivat & J. Reynolds, eds., *Algebraic methods in semantics*, 277–312, Cambridge Univ. Press, 1985.

[13] P. Degano, R. de Nicola & U. Montanari. A partial ordering semantics for CCS, *TCS* 75(3):223–262, 1990.

[14] E.A. Emerson. Temporal and modal logic. In [31], 995–1072.

[15] S. Feferman. Formal theories for transfinite iterations of generalized inductive definitions and some subsystems of analysis, In A. Kino, J. Myhill & R.E. Vesley, eds., *Intuitionism and Proof Theory*, 303–326, North Holland, 1970.

[16] J.F. Groote. Transition System Specification with Negative Premises. *LNCS* 458, 332–341 1990.

[17] C.A. Gunter & D.S. Scott. Semantic domains, In [31], 633–674.

[18] G. Kahn. Natural semantics, In K. Fuchi & M. Nivat, eds., *Programming of Future Generation Computers*, 237–258, Elsevier, 1988.

[19] K. Kunen. Declarative semantics of logic programming. *Bull. EATCS* 44:147-167, 1991.

[20] J.-L. Lassez & M.J. Maher. Closure and fairness in the semantics of programming logic. *TCS* 29:167–184, 1984.

[21] R. Milner. Operational and algebraic semantics of concurrent processes, In [31], 1201–1242.

[22] R. Milner & M. Tofte. Co-induction in relational semantics, *TCS* 87:209–220, 1991.

[23] P.D. Mosses. Denotational semantics, In [31], 575–631.

[24] A. Mycroft. *Abstract interpretation and optimising transformations for applicative programs.* PhD Thesis, CST-15-81, Univ. of Edinburgh, 1981.

[25] F. Nielson. A denotational framework for data flow analysis. *Acta Informatica*, 18:265–287, 1982.

[26] D. Park. On the semantics of fair parallelism. *LNCS* 86, 504–526, 1980.

[27] G.D. Plotkin. Call-by-name, call-by-value and the λ-calculus. *TCS* 1:125–159, 1975.

[28] G.D. Plotkin. A structural approach to operational semantics. Tech. Rep. DAIMI FN-19, Aarhus Univ., 1981.

[29] G. Reggio, A non-standard inductive semantics, *LNCS* 472, 362–372, 1990.

[30] M.H. van Emden & R.A. Kowalski, The semantics of predicate logic as a programming language, *JACM* 23(4):733–742, 1976.

[31] J. van Leeuwen (ed.). Formal Models and Semantics. vol. B of *Handbook of Theoretical Computer Science*, Elsevier, 1990.