Abstract Interpretation of Algebraic Polynomial Systems
(Extended Abstract)

Patrick Cousot[1] and Radhia Cousot[2]

[1] LIENS, École Normale Supérieure, 45 rue d'Ulm, 75230 Paris cedex 05, France
cousot@dmi.ens.fr http://www.dmi.ens.fr/~cousot
[2] LIX, CNRS & École Polytechnique, 91140 Palaiseau cedex, France
rcousot@lix.polytechnique.fr http://lix.polytechnique.fr/~rcousot

Abstract. We define a hierarchy of compositional formal semantics of algebraic polynomial systems over $\mathcal{F}$-algebras by abstract interpretation. This generalizes classical formal language theoretical results and context-free grammar flow-analysis algorithms in the same uniform framework of universal algebra and abstract interpretation.

1 Introduction

We consider algebraic polynomial systems generalizing language-theoretic context-free grammars to $\mathcal{F}$-algebras [Courcelle, 1996; Mezei & Wright, 1967]. We provide a compositional fixpoint derivation tree semantics and show that by abstract interpretation [Cousot & Cousot, 1977, 1979], we can derive a hierarchy of formal fixpoint semantics generalizing results from the theory of formal languages e.g. [Ginsburg & Rice, 1962; Schützenberger, 1962] as well as grammar analysis algorithms e.g. [Möncke & Wilhelm, 1991; Jeuring & Swierstra, 1994, 1995]. We extend the results to infinite terms.

2 Algebraic Polynomial Systems

Let $\mathcal{S}$ be a set of *sorts*. An $\mathcal{S}$*-signature* is a set $\mathcal{F}$ of *function symbols* equipped with a mapping $\mathit{type} \in \mathcal{F} \longmapsto \mathcal{S}^* \times \mathcal{S}$. We write $f^{s_1 \times \ldots \times s_n \to s}$ for $\mathit{type}(f) = \langle s_1 \ldots s_n,\ s\rangle$ and $f^{\epsilon \to s}$ when $n = 0$. An $\mathcal{F}$*-algebra* [Meinke & Tucker, 1992; Wirsing, 1990] is a pair $\mathcal{A} = \langle \{\mathcal{A}_s\}_{s \in \mathcal{S}},\ \{f_\mathcal{A}\}_{f \in \mathcal{F}}\rangle$ where the *domain* $\mathcal{A}_s$ of sort s of $\mathcal{A}$ is a non-empty set and for each $f \in \mathcal{F}$ of type $s_1 \times \ldots \times s_n \to s$, the *operation* $f_\mathcal{A}$ is a total mapping $\mathcal{A}_{s_1} \times \ldots \times \mathcal{A}_{s_n} \longmapsto \mathcal{A}_s$. Classical examples of $\mathcal{F}$-algebras are the monoids of words and traces, trees, graphs with sources, etc. [Courcelle, 1996].

Let $\mathcal{X}$ be an $\mathcal{S}$-sorted set of free *variables* x (disjoint from function symbols). We write x^s for $\mathit{type}(x) = s$. Let $\mathcal{M}^{s,\mathcal{X}}$ be the set of *monomials of type* s *over variables* $\mathcal{X}$ with the following syntax:

$$M^{s,\{x^s\}\cup Y} \to x^s, \qquad \text{monomial of type } s \text{ over variables } \{x^s\} \cup Y;$$

$$M^{s,Y} \to f^{\sigma \to s}(L^{\sigma,Y}), \quad f \in \mathcal{F},\ (c() \text{ is written } c \text{ for constants } c^{\epsilon \to s});$$

$$L^{s\times\sigma,Y} \rightarrow M^{s,Y}, L^{\sigma,Y}, \quad \text{list of monomials of type } s\sigma \text{ over variables } Y;$$
$$L^{s,Y} \rightarrow M^{s,Y}, \quad \text{unary list.}$$

Let $\mathcal{D}$ be an $\mathcal{S}$-sorted set of *derivation labels* d^s (disjoint from function symbols and variables). Let $\delta \in \mathcal{X} \longmapsto \wp(\mathcal{D})$ such that $\forall x^s \in \mathcal{X}^s : \forall d \in \delta(x^s) : d \in \mathcal{D}^s$ and $\forall x, y \in \mathcal{X} : x \neq y \Longrightarrow \delta(x) \cap \delta(y) = \emptyset$ be an *assignment of derivation labels to variables*. For each $d^s \in \mathcal{D}$, we let $\Delta(d^s) \in \mathcal{M}^{s,\mathcal{X}}$ be a monomial of type s over the variables $\mathcal{X}$ representing the *derivation* labeled $d^s \in \delta(x^s)$ for variable x^s. An $\mathcal{S}$-sorted *polynomial system* $\mathcal{P}$ is a tuple $\langle \mathcal{F}, \mathcal{X}, \mathcal{D}, \delta, \Delta \rangle$ defining the *system of equations* $\{x^s = \Delta(d^s) \mid s \in \mathcal{S} \wedge x^s \in \mathcal{X}^s \wedge d^s \in \delta(x^s)\}$.

Example 1 (Polynomial system). The polynomial system given by $\mathcal{S} = \{s\}$, $\mathcal{F} = \{b^{s\times s\rightarrow s}, a^{\epsilon\rightarrow s}\}$, $\mathcal{X} = \{A, B\}$, $\mathcal{D} = \{d_1, d_2\}$, $\delta(A) = \{d_1, d_2\}$, $\delta(B) = \emptyset$, $\Delta(d_1) = b(A, A)$ and $\Delta(d_2) = a$ with equations $\{A = b(A, A),\ A = a\}$ can be written as: $A = b(A, A)\ [d_1]\ +\ a\ [d_2]$, $B = \Omega$. □

If $\mathcal{F}$ is an $\mathcal{S}$-signature then $\mathcal{F}_+$ is $\mathcal{F}$ enlarged for each sort $s \in \mathcal{S}$ with a new symbol $+_s$ of type $s \times s \rightarrow s$ and a new constant Ω_s of type $\epsilon \rightarrow s$. The syntax of an *$\mathcal{F}_+$-polynomial system $\mathcal{P}$ over $\mathcal{X}$ and $\mathcal{D}$* [Courcelle, 1996; Mezei & Wright, 1967] of type σ is given by the following context-free attribute grammar with axiom $S^{\sigma,\mathcal{X},\mathcal{X},\mathcal{D}}$, $\sigma \in \mathcal{S}^+$ (for short we often omit the derivation labels):

$S^{s\times\sigma,X,Y,D_1\cup D_2} \rightarrow E^{s,x^s,Y,D_1}\ S^{\sigma,X\setminus\{x^s\},Y,D_2}$,

polynomial system of type $s\sigma$ ($s \notin \sigma$) with left-variables $X \subseteq \mathcal{X}$, right-variables $Y \subseteq \mathcal{X}$ and labels $D_1 \cup D_2 \subseteq \mathcal{D}$ ($D_1 \cap D_2 = \emptyset$);

$S^{s,\{x^s\},Y,D} \rightarrow E^{s,x^s,Y,D}$,		monoequational polynomial system;
$E^{s,x^s,Y,D} \rightarrow x^s = P^{s,Y,D}$,		equation of type s;
$E^{s,x^s,Y,\emptyset} \rightarrow x^s = \Omega_s$,		void equation;
$P^{s,Y,\{d^s\}\cup D} \rightarrow M^{s,Y}\ [d^s] +_s P^{s,Y,D}$,		polynomial of type s over variables Y and derivation labels $\{d^s\} \cup D$ with $d^s \notin D$;
$P^{s,Y,\{d^s\}} \rightarrow M^{s,Y}\ [d^s]$,		labeled derivation.

The corresponding label assignments and derivations are given by:

$$\begin{aligned}
\delta[\![ES]\!] &= \delta[\![E]\!] \cup \delta[\![S]\!], & \Delta[\![ES]\!] &= \Delta[\![E]\!] \cup \Delta[\![S]\!],\\
\delta[\![x = P]\!] &= \{\langle x,\ \delta[\![P]\!]\rangle\}, & \Delta[\![x = P]\!] &= \Delta[\![P]\!],\\
\delta[\![x = \Omega]\!] &= \emptyset, & \Delta[\![x = \Omega]\!] &= \emptyset,\\
\delta[\![M\ [d] + P]\!] &= \{d\} \cup \delta[\![P]\!], & \Delta[\![M\ [d] + P]\!] &= \{\langle d,\ M\rangle\} \cup \Delta[\![P]\!],\\
\delta[\![M\ [d]]\!] &= \{d\}, & \Delta[\![M\ [d]]\!] &= \{\langle d,\ M\rangle\}\ .
\end{aligned}$$

Context-free grammars can be considered as algebraic polynomial systems in several ways [Courcelle, 1996]. The meta-syntax of context-free grammars is given by the following meta-grammar[1]:

$G \rightarrow P\ G \mid P$,	Grammar;	$R \rightarrow V\ R \mid R$,	Right sides;
$P \rightarrow N$ '$\rightarrow$' A,	Production;	$A \rightarrow A$ '|' $R \mid R \mid$ 'ε',	Alternative right sides.
$V \rightarrow N \mid T$,	Vocabulary;		

[1] We omit attributes ensuring that all productions with the same left side nonterminal have their right sides grouped, with the alternative right sides separated by |.

In this meta-grammar the meta-terminals are $\{`\rightarrow', `|', `\varepsilon'\}$ and the meta-nonterminals are $\{G, P, A, R, V, N, T\}$. The meta-productions for nonterminals $N \in \mathcal{N}$ and terminals $T \in \mathcal{T}$ (not containing $\rightarrow$, $|$ and ε) have been left unspecified. The meta-vocabulary $\mathcal{V}$ is $\mathcal{T} \cup \mathcal{N}$. A grammar G is usually presented as a triple $\langle \mathcal{T}, \mathcal{N}, \mathsf{P}[\![G]\!]\rangle$ or a quadruple $\langle \mathcal{T}, \mathcal{N}, \mathtt{A}, \mathsf{P}[\![G]\!]\rangle$, where the axiom $\mathtt{A} \in \mathcal{N}$ is a distinguished nonterminal, and the productions are:

$$\mathsf{P}[\![P\,G]\!] \triangleq \mathsf{P}[\![P]\!] \cup \mathsf{P}[\![G]\!], \quad \mathsf{P}[\![N \rightarrow A]\!] \triangleq \{N \rightarrow \mathrm{r} \mid \mathrm{r} \in \mathsf{A}[\![A]\!]\},$$
$$\mathsf{A}[\![A \mid R]\!] \triangleq \mathsf{A}[\![A]\!] \cup \mathsf{A}[\![R]\!], \qquad \mathsf{A}[\![R]\!] \triangleq \{\mathsf{R}[\![R]\!]\}, \qquad \mathsf{A}[\![\varepsilon]\!] \triangleq \{\varepsilon\},$$
$$\mathsf{R}[\![V\,R]\!] \triangleq \mathsf{R}[\![V]\!]\,\mathsf{R}[\![R]\!], \qquad \mathsf{R}[\![T]\!] \triangleq T, \qquad \mathsf{R}[\![N]\!] \triangleq N\ .$$

Example 2 (Sentence generating grammars). If grammars are understood as defining a set of sentences then there is only one sort *string*. The nullary function symbols are ε (empty string) and the terminals $T \in \mathcal{T}$. The only binary function symbol is the concatenation $\bullet$ of strings also denoted by juxtaposition. $\mathcal{F}_+$ includes $+$, also denoted $|$ for alternative/language union and Ω which denotes the empty language. The nonterminals $N \in \mathcal{N}$ are the variables. For example, if $\mathcal{N} \triangleq \{x, y, z\}$ and $\mathcal{T} \triangleq \{a, b\}$ then the grammar $x \rightarrow yx \mid y$, $x \rightarrow yx \mid y$ corresponds to $x = y \bullet x\ [1] + y\ [2]$, $y = a\ [3] + b\ [4]$, $z = \Omega$. The translation is given by:

$$\mathsf{P}[\![P\,G]\!]^{\mathrm{D}_1 \cup \mathrm{D}_2} \triangleq \mathsf{P}[\![P]\!]^{\mathrm{D}_1}\ \mathsf{P}[\![G]\!]^{\mathrm{D}_2} \quad \mathsf{P}[\![N \rightarrow A]\!]^{\mathrm{D}} \triangleq N = \mathsf{A}[\![A]\!]^{\mathrm{D}}$$
$$\mathsf{A}[\![A \mid R]\!]^{\mathrm{D} \cup \{\mathrm{d}\}} \triangleq \mathsf{A}[\![A]\!]^{\mathrm{D}} + \mathsf{A}[\![R]\!]^{\mathrm{d}} \qquad \mathsf{A}[\![R]\!]^{\mathrm{d}} \triangleq \mathsf{R}[\![R]\!]\ [\mathrm{d}]$$
$$\mathsf{A}[\![\varepsilon]\!]^{\mathrm{d}} \triangleq \varepsilon\ [\mathrm{d}] \qquad \mathsf{R}[\![V\,R]\!] \triangleq V \bullet \mathsf{R}[\![R]\!]\ . \qquad \square$$

Example 3 (Parse tree generating grammars). We can also interpret grammars as generating parse trees, for example $x \rightarrow yx \mid y$, $x \rightarrow yx \mid y$ corresponds to $\bar{x} = x2(\bar{y}, \bar{x}) + x1(\bar{y})$, $\bar{y} = y1(a) + y1(b)$, $\bar{z} = \Omega$. The translation is then:

$$\mathsf{P}[\![P\,G]\!] \triangleq \mathsf{P}[\![P]\!]\ \mathsf{P}[\![G]\!], \qquad \mathsf{P}[\![N \rightarrow A]\!] \triangleq \overline{N} = \mathsf{A}[\![N, A]\!],$$
$$\mathsf{A}[\![N, A \mid R]\!] \triangleq \mathsf{A}[\![N, A]\!] + \mathsf{A}[\![N, R]\!], \qquad \mathsf{A}[\![N, R]\!] \triangleq N\ell[\![R]\!](\mathsf{R}[\![R]\!]),$$
$$\mathsf{A}[\![N, \varepsilon]\!] \triangleq N1(\varepsilon), \qquad \mathsf{R}[\![V\,R]\!] \triangleq \mathsf{R}[\![V]\!], \mathsf{R}[\![R]\!],$$
$$\mathsf{R}[\![T]\!] \triangleq T, \qquad \mathsf{R}[\![N]\!] \triangleq \overline{N},$$
$$\ell[\![V\,R]\!] \triangleq \ell[\![V]\!] + \ell[\![R]\!], \qquad \ell[\![V]\!] \triangleq 1\ . \qquad \square$$

3 Small Step Operational Semantics: States and Transitions

The transition/small-step operational semantics associates a *discrete transition system* to each algebraic polynomial system that is a pair $\langle \Sigma, \tau \rangle$ where Σ is a (non-empty) set of states, $\tau \subseteq \Sigma \times \Sigma$ is the binary transition relation between a state and its possible successors. We write $s\ \tau\ s'$ or $\tau(s, s')$ for $\langle s, s' \rangle \in \tau$ using the isomorphism $\wp(\Sigma \times \Sigma) \simeq (\Sigma \times \Sigma) \longmapsto \mathbb{B}$ where $\mathbb{B} \triangleq \{\mathrm{tt}, \mathrm{ff}\}$ is the set of booleans. $\check{\tau} \triangleq \{s \in \Sigma \mid \forall s' \in \Sigma : \neg(s\ \tau\ s')\}$ is the set of *final/blocking states*. We formally define several transition systems $\langle \Sigma[\![\mathcal{P}]\!], \tau[\![\mathcal{P}]\!]\rangle$ associated with polynomial system $\mathcal{P}$ which generalize the term rewriting system and the derivation trees of [Courcelle, 1996].

Example 4 (Protosentence transition system). In order to generalize the protosentence derivation relation for context-free grammars, the set $\tau^{s}[\![\mathcal{P}]\!]$ of states for the polynomial system $\mathcal{P}$ defined by $A = b(A, A) + a$ is the language generated by the grammar $A' \rightarrow A'A' \mid a \mid A$ with terminals $\{a, A\}$ and nonterminals $\{A'\}$. More generally states can be chosen as protosentences that is sentences containing constants $c^{\epsilon \rightarrow s} \in \mathcal{F}$ and variables $x \in \mathcal{X}$. The set $\Sigma^{s}[\![\mathcal{P}]\!]$ of states of the $\mathcal{S}$-sorted *polynomial system* $\mathcal{P} = \langle \mathcal{F}, \mathcal{X}, \mathcal{D}, \delta, \Delta \rangle$ is the language generated by the grammar $G = \langle \mathcal{V}, \{x' \mid x \in \mathcal{X}\}, P \rangle$ where vocabulary is $\mathcal{V} \triangleq \{f \mid c^{\epsilon \rightarrow s} \in \mathcal{F}\} \cup \mathcal{X}$ and the productions are $P = \{x' \rightarrow x \mid x \in \mathcal{X}\} \cup \{x' \rightarrow \varphi'(\Delta(d)) \mid x \in \mathcal{X} \wedge d \in \delta(x) \wedge \Delta(d) \neq \Omega\}$ with $\varphi'(x) \triangleq x'$, $\varphi'(c^{\epsilon \rightarrow s}) \triangleq c$, $\varphi'(f^{\sigma \rightarrow s}(L)) \triangleq \varphi'(L)$ when $\sigma \neq \epsilon$ and $\varphi'(M, L) = \varphi'(M)\varphi'(L)$. The transition relation is then:

$$\tau^{s}[\![\mathcal{P}]\!] \triangleq \{\langle pxq,\ p\varphi(\Delta(d))q \rangle \mid x \in \mathcal{X} \wedge \exists d \in \delta(x) : \Delta(d) \neq \Omega\} \qquad (1)$$

with $\varphi(x) \triangleq x$, $\varphi(c^{\epsilon \rightarrow s}) \triangleq c$, $\varphi(f^{\sigma \rightarrow s}(L)) \triangleq \varphi(L)$ when $\sigma \neq \epsilon$ and $\varphi(M, L) = \varphi(M)\varphi(L)$. This one-step derivation $\langle s,\ s' \rangle \in \tau^{s}[\![\mathcal{P}]\!]$ is written as $s \overset{\mathcal{P}}{\Longrightarrow} s'$. $\square$

Example 5 (Context transition system). For top-down analysis, it is convenient to consider the derivation sequence where all replacements are delimited by brackets. The transition system is: $\langle \Sigma^{c}[\![\mathcal{P}]\!],\ \tau^{c}[\![\mathcal{P}]\!] \rangle$ with:

$$\begin{aligned}
\Sigma^{c}[\![\mathcal{P}]\!] &\triangleq \mathcal{V}^{\star} \bullet \{[\} \bullet \mathcal{V}^{\star} \bullet \{]\} \bullet \mathcal{V}^{\star} \quad \text{where} \quad \mathcal{V} \triangleq \mathcal{X} \cup \{c^{\epsilon \rightarrow s} \mid c \in \mathcal{F}\}, \qquad (2)\\
\tau^{c}[\![\mathcal{P}]\!] &\triangleq \{\langle p[qxq']p',\ pq[\varphi(\Delta(d))]q'p' \rangle \mid x \in \mathcal{X} \wedge d \in \delta(x) \wedge \Delta(d) \neq \Omega\}.
\end{aligned}$$

For the example polynomial system $\mathcal{P}$ defined by $A = b(A, A) + c(A) + d(a, A) + a$, two possible derivations sequences with delimited contexts would be $[A] \overset{\mathcal{P}}{\Longrightarrow} [aA] \overset{\mathcal{P}}{\Longrightarrow} a[A]$ and $[A] \overset{\mathcal{P}}{\Longrightarrow} [AA] \overset{\mathcal{P}}{\Longrightarrow} [a]A$. $\square$

Example 6 (Parse tree transition system). Parse trees for context free grammar generalize to $\mathcal{F}$-algebras [Meinke & Tucker, 1992]. The states of the transition system $\langle \Sigma^{p}[\![\mathcal{P}]\!], \tau^{p}[\![\mathcal{P}]\!] \rangle$ for the polynomial system $\mathcal{P} = \langle \mathcal{F}, \mathcal{X}, \mathcal{D}, \delta, \Delta \rangle$ are parse trees in parenthesized form. They derive from the nonterminals x'', $x \in \mathcal{X}$ in the meta-grammar with productions:

$$\begin{aligned}
x'' &\rightarrow [x'], && \text{for each } x \in \mathcal{X},\\
x'' &\rightarrow x, && \text{for each } x \in \mathcal{X},\\
x' &\rightarrow x : \phi'(\Delta(d)), && \text{for each } x \in \mathcal{X}, d \in \delta(x) \text{ and } \Delta(d) \neq \Omega,
\end{aligned}$$

with $\phi'(x) \triangleq x'$, $\phi'(c^{\epsilon \rightarrow s}) \triangleq c$, $\phi'(f^{\sigma \rightarrow s}(L)) \triangleq f[\phi'(L)]$ when $\sigma \neq \epsilon$ and $\phi'(M, L) = \phi'(M), \phi'(L)$. For example the parse tree $[A : b[A : a, A]]$ is a state of the polynomial system $\mathcal{P}$ defined by $A = b(A, A) + a$. Transitions construct children of a variable node in a parse tree as given by the right-hand side of the polynomial system equations:

$$\tau^{p}[\![\mathcal{P}]\!] \triangleq \{\langle p[qxq']p',\ p[qx : \phi'(\Delta(d))q']p' \rangle \mid d \in \delta(x) \wedge \Delta(d) \neq \Omega \wedge q' \neq [q'']\}. \qquad (3)$$

For the $A = b(A, A) + a$ example, a possible transition is $[A : b[A, A]] \overset{\mathcal{P}}{\Longrightarrow} [A : b[A : a, A]]$. $\square$

Example 7 (Derivation tree transition system). More generally, we can built the parse tree and record which variables are expanded in parallel at each derivation step thus generalizing the *derivation trees* of [Courcelle, 1996]. For example, the states of the polynomial system $A = b(A, A)\ [d_1] + a\ [d_2]$ are given by $A' \rightarrow \langle A\rangle \mid A \mid A[d_1]b(A', A') \mid A[d_2]a$ where $\langle A\rangle$ is to be derived as in $\langle A\rangle \overset{\mathcal{P}}{\Longrightarrow} A[d_1]b(\langle A\rangle, \langle A\rangle) \overset{\mathcal{P}}{\Longrightarrow} A[d_1]b(A[d_1]b(\langle A\rangle, A), A[d_2]a) \overset{\mathcal{P}}{\Longrightarrow} A[d_1]b(A[d_1](A[d_2]a, \langle A\rangle), A[d_2]a) \overset{\mathcal{P}}{\Longrightarrow} A[d_1]b(A[d_1]b(A[d_2]a, A[d_2]a), A[d_2]a)$.

Given an $\mathcal{S}$-sorted polynomial system $\mathcal{P} = \langle \mathcal{F}, \mathcal{X}, \mathcal{D}, \delta, \Delta\rangle$, the set of state is the set $\Sigma^{d}[\![\mathcal{P}]\!]$ of *derivation trees* T^s of sort s defined by the following grammar:

$$
\begin{array}{rlll}
T^s & \rightarrow & x^s & \text{variable,}\\
& \mid & \langle x^s\rangle & \text{substitution variable,}\\
& \mid & f^{s_1\times\ldots\times s_n\rightarrow s}(T^{s_1}, \ldots, T^{s_n}) & \text{node,}\\
& \mid & c^{\epsilon\rightarrow s} & \text{leave,}\\
& \mid & x^s[d^s]f^{s_1\times\ldots\times s_n\rightarrow s}(T^{s_1}, \ldots, T^{s_n}) & \text{substituted node,}\\
& \mid & x^s[d^s]c^{\epsilon\rightarrow s} & \text{substituted leave.}
\end{array}
$$

A *substitution* θ maps variables $x^s \in \mathcal{X}$ to sets $\theta(x)$ of monomials m^s of type s over the variables $\{x, \langle x\rangle \mid x \in \mathcal{X}\}$. The *identity substitution* ι satisfies $\forall x \in \mathcal{X} : \iota(x) = \{x, \langle x\rangle\}$. The *application* $T[\![\theta]\!]$ *of substitution* θ *for derivation label* d *to a tree* T is defined as follows:

$$
\begin{aligned}
x^s[\![\theta]\!]d^s &\triangleq \{x^s, \langle x^s\rangle\},\\
\langle x^s\rangle[\![\theta]\!]d^s &\triangleq \{x^s[d^s]m^s \mid m^s \in \theta(x^s)\},\\
f^{s_1\times\ldots\times s_n\rightarrow s}(T^{s_1}, \ldots, T^{s_n})[\![\theta]\!]d^s &\triangleq \{f(m^{s_1}, \ldots, m^{s_n}) \mid \bigwedge_{i=1}^{n} m^{s_i} \in T^{s_i}[\![\theta]\!]d^s\},\\
c^{\epsilon\rightarrow s}[\![\theta]\!]d^s &\triangleq \{c\},\\
x^s[d_0^s]f^{s_1\times\ldots\times s_n\rightarrow s}(T^{s_1}, \ldots, T^{s_n})[\![\theta]\!]d^s &\triangleq \{x^s[d_0^s]f(m^{s_1}, \ldots, m^{s_n}) \mid \bigwedge_{i=1}^{n} m^{s_i} \in T^{s_i}[\![\theta]\!]d^s\},\\
x^s[d_0^s]c^{\epsilon\rightarrow s}[\![\theta]\!]d^s &\triangleq \{x^s[d_0^s]c\}\ .
\end{aligned}
$$

The transition relation consists in replacing all substitution variables in a derivation tree by corresponding right-hand side monomials and then in choosing substitution variables for the next step:

$$\tau^{d}[\![\mathcal{P}]\!] \triangleq \{\langle T, T'\rangle \mid \exists\theta : \forall x \in \mathcal{X} : \exists d \in \delta(x) : \theta(x) \subseteq \Delta(d)[\![\iota]\!] \wedge T' \in T[\![\theta]\!]d\}\ .$$

$\langle T, T'\rangle \in \tau^{d}[\![\mathcal{P}]\!]$ is written $T \overset{\mathcal{P}}{\Longrightarrow} T'$, $\overset{\mathcal{P}}{\underset{\star}{\Longrightarrow}}$ is the reflexive transitive closure. □

4 Fixpoint Semantics

A *fixpoint semantic specification* is a pair $\langle D, F\rangle$ where the *semantic domain* $\langle D, \sqsubseteq, \perp, \sqcup\rangle$ is a poset with partial order $\sqsubseteq$, infimum $\perp$ and partially defined least upper bound (lub) $\sqcup$ and the *semantic transformer* $F \in D \overset{m}{\longmapsto} D$ is a total monotone map from D to D assumed to be such that the transfinite *iterates of* F *from* $\perp$, that is $F^0 \triangleq \perp$, $F^{\delta+1} \triangleq F(F^\delta)$ for successor ordinals $\delta + 1$

and $F^\lambda \triangleq \bigsqcup_{\delta<\lambda} F^\delta$ for limit ordinals λ are well-defined (e.g. when $\langle D, \sqsubseteq, \perp, \sqcup\rangle$ is a directed-complete partial order or DCPO). By monotonicity, these iterates form an increasing chain, hence reach a fixpoint so that the *iteration order* can be defined as the least ordinal ϵ such that $F(F^\epsilon) = F^\epsilon$. This specifies the *fixpoint semantics* S as the $\sqsubseteq$-least fixpoint $S \triangleq \mathrm{lfp}^{\sqsubseteq} F = F^\epsilon$ of F.

5 Derivation Semantics, Its Fixpoint Characterization

The set Σ^+ of finite derivations for the transition system $\langle\Sigma, \tau\rangle$ is the set $\tau^{\vec{\star}} \triangleq \bigcup_{n>0} \tau^{\vec{n}}$ where the set of non-empty sequences of n states separated by transitions τ is $\tau^{\vec{n}} \triangleq \{\sigma_0 \ldots \sigma_{n-1} \mid \forall i \in [0, n-1[: \langle\sigma_i,\ \sigma_{i+1}\rangle \in \tau\}$.

Example 8. A possible maximal derivation for the polynomial system $A = b(A, A)\ [d_1] + a\ [d_2]$ is $\langle A\rangle \stackrel{\mathcal{P}}{\Longrightarrow} A[d_{d_1}]b(\langle A\rangle, A) \stackrel{\mathcal{P}}{\Longrightarrow} A[1]b(A[d_2]a, \langle A\rangle) \stackrel{\mathcal{P}}{\Longrightarrow} A[d_1]b(A[d_2]a, A[d_2]a)$. □

The set $\tau^{\vec{\star}}$ of finite derivations for the transition system $\langle\Sigma, \tau\rangle$ can be characterized in fixpoint top-down/forward and bottom-up/backward form:

$$\begin{aligned}
\tau^{\vec{\star}} &= \mathrm{lfp}^{\subseteq} \vec{T} \quad \text{where} \quad \vec{T}(X) \triangleq \tau^{\vec{1}} \cup (X \mathbin{;} \tau^{\vec{)}}2 \quad \text{(top-down/forward)}, && (4)\\
&= \mathrm{lfp}^{\subseteq} \vec{B} \qquad\qquad\quad \vec{B}(X) \triangleq \tau^{\vec{2}} \mathbin{;} (X \cup \tau^{\vec{1}}) \quad \text{(bottom-up/backward)}, && (5)
\end{aligned}$$

where $\tau^{\vec{1}} \cong \Sigma$ is the set of state sequences $\sigma_0 \in \Sigma$ of length one and $X \mathbin{;} Y \triangleq \{\sigma_0 \ldots \sigma_{n-1} \bullet \sigma'_1 \ldots \sigma'_{m-1} \mid \sigma_0 \ldots \sigma_{n-1} \in X \wedge \sigma_{n-1} = \sigma'_0 \wedge \sigma'_0 \ldots \sigma'_{m-1} \in Y\}$ is the sequential composition of sets of finite derivations. The transformers $\vec{T}$ and $\vec{B}$ are $\cup$-additive on the complete lattice $\langle\wp(\Sigma^+), \subseteq, \emptyset, \Sigma, \cup, \cap\rangle$ of sets of derivation sequences.

6 Fixpoint Semantics Transfer and Approximation

In abstract interpretation, the *concrete semantics* $S^\natural$ is approximated by a (usually computable) *abstract semantics* $S^\sharp$ via an abstraction function $\alpha \in D^\natural \longmapsto D^\sharp$ such that $\alpha(S^\natural) \sqsubseteq^\sharp S^\sharp$ [2]. The abstraction is *exact* if $\alpha(S^\natural) = S^\sharp$ and *approximate* if $\alpha(S^\natural) \sqsubset^\sharp S^\sharp$.

6.1 Fixpoint Semantics Approximation

To derive $S^\sharp$ from $S^\natural$ by abstraction or $S^\natural$ from $S^\sharp$ by refinement, we can use the following fixpoint approximation theorems (as usual, we call a function f *Scott-continuous*, written $f : D \stackrel{c}{\longmapsto} E$, if it is monotone and preserves the lub of any directed subset A of D):

Theorem 9 (S. Kleene fixpoint approximation). Let $\langle\langle D^\natural, \sqsubseteq^\natural, \perp^\natural, \sqcup^\natural\rangle, F^\natural\rangle$ and $\langle\langle D^\sharp, \sqsubseteq^\sharp, \perp^\sharp, \sqcup^\sharp\rangle, F^\sharp\rangle$ be concrete and abstract fixpoint semantic specifications. If the $\perp$-strict Scott-continuous abstraction function $\alpha \in D^\natural \stackrel{\perp,c}{\longmapsto} D^\sharp$ is such that for

[2] More generally, we look for an abstract semantics $S^\sharp$ such that $\alpha(S^\natural) \preceq^\sharp S^\sharp$ for the *approximation partial ordering* $\preceq^\sharp$ corresponding to logical implication which may differ from the *computational partial orderings* $\sqsubseteq$ used to define least fixpoints [Cousot & Cousot, 1994].

all $x \in D^\natural$ such that $x \sqsubseteq^\natural F^\natural(x)$ there exists $y \sqsubseteq^\natural x$ such that $\alpha(F^\natural(x)) \sqsubseteq^\sharp F^\sharp(\alpha(y))$ then $\alpha(\mathrm{lfp}^{\sqsubseteq^\natural} F^\natural) \sqsubseteq^\sharp \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$. □

Theorem 10 (A. Tarski fixpoint approximation). Let $\langle D^\natural, F^\natural\rangle$ and $\langle D^\sharp, F^\sharp\rangle$ be concrete and abstract fixpoint semantic specifications such that $\langle D^\natural, \sqsubseteq^\natural, \bot^\natural, \top^\natural, \sqcup^\natural, \sqcap^\natural\rangle$ and $\langle D^\sharp, \sqsubseteq^\sharp, \bot^\sharp, \top^\sharp, \sqcup^\sharp, \sqcap^\sharp\rangle$ are complete lattices. If the monotone abstraction function $\alpha \in D^\natural \xmapsto{m} D^\sharp$ is such that for all $y \in D^\sharp$ such that $F^\sharp(y) \sqsubseteq^\sharp y$ there exists $x \in D^\natural$ such that $\alpha(x) \sqsubseteq^\sharp y$ and $F^\natural(x) \sqsubseteq^\natural x$ then $\alpha(\mathrm{lfp}^{\sqsubseteq^\natural} F^\natural) \sqsubseteq^\sharp \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$. □

6.2 Fixpoint Semantics Transfer

When the abstraction must be exact, that is $\alpha(S^\natural) = S^\sharp$, we can use the following fixpoint transfer theorem, which provide guidelines for designing $S^\sharp$ from $S^\natural$ (or dually) in fixpoint form [Cousot & Cousot, 1979, theorem 7.1.0.4(3)]:

Theorem 11 (S. Kleene fixpoint transfer). Let $\langle D^\natural, F^\natural\rangle$ and $\langle D^\sharp, F^\sharp\rangle$ be concrete and abstract fixpoint semantic specifications. If the $\bot$-strict Scott-continuous abstraction function $\alpha \in D^\natural \xmapsto{\bot,c} D^\sharp$ satisfies the *commutation condition* $F^\sharp \circ \alpha = \alpha \circ F^\natural$ then $\alpha(\mathrm{lfp}^{\sqsubseteq^\natural} F^\natural) = \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$. Moreover the respective iterates ${F^\natural}^\delta$ and ${F^\sharp}^\delta$, $\delta \in \mathbb{O}$ of $F^\natural$ and $F^\sharp$ from $\bot^\natural$ and $\bot^\sharp$ satisfy $\forall\delta \in \mathbb{O}$: $\alpha({F^\natural}^\delta) = {F^\sharp}^\delta$ and the iteration order of $F^\sharp$ is less than or equal to that of $F^\natural$. □

Observe that in theorem Theorem 11 (as well as in theorem Theorem 9), Scott-continuity of the abstraction function α is a too strong hypothesis since we only use the fact that α preserves the lub of the iterates of $F^\natural$ starting from $\bot^\natural$. When this is not the case, but α preserves glbs, we can use:

Theorem 12 (A. Tarski fixpoint transfer). Let $\langle D^\natural, F^\natural\rangle$ and $\langle D^\sharp, F^\sharp\rangle$ be concrete and abstract fixpoint semantic specifications such that $\langle D^\natural, \sqsubseteq^\natural, \bot^\natural, \top^\natural, \sqcup^\natural, \sqcap^\natural\rangle$ and $\langle D^\sharp, \sqsubseteq^\sharp, \bot^\sharp, \top^\sharp, \sqcup^\sharp, \sqcap^\sharp\rangle$ are complete lattices. If the abstraction function $\alpha \in D^\natural \xmapsto{\sqcap} D^\sharp$ is a complete $\sqcap$-morphism satisfying the *commutation inequality* $F^\sharp \circ \alpha \sqsubseteq^\sharp \alpha \circ F^\natural$ and the *post-fixpoint correspondence* $\forall y \in D^\sharp : F^\sharp(y) \sqsubseteq^\sharp y \Longrightarrow \exists x \in D^\natural : \alpha(x) = y \wedge F^\natural(x) \sqsubseteq^\natural x$ then $\alpha(\mathrm{lfp}^{\sqsubseteq^\natural} F^\natural) = \mathrm{lfp}^{\sqsubseteq^\sharp} F^\sharp$. □

6.3 Semantics Abstraction

An important particular case of abstraction function $\alpha \in D^\natural \longmapsto D^\sharp$ is when α preserves existing lubs $\alpha\big(\bigsqcup^\natural_{i\in\Delta} x_i\big) = \bigsqcup^\sharp_{i\in\Delta} \alpha(x_i)$. In this case there exists a unique map $\gamma \in D^\sharp \longmapsto D^\natural$ (so-called the *concretization function* [Cousot & Cousot, 1977]) such that the pair $\langle\alpha, \gamma\rangle$ is a *Galois connection*, written$\langle D^\natural, \sqsubseteq^\natural\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle D^\sharp, \sqsubseteq^\sharp\rangle$, which means that $\langle D^\natural, \sqsubseteq^\natural\rangle$ and $\langle D^\sharp, \sqsubseteq^\sharp\rangle$ are posets, $\alpha \in D^\natural \longmapsto D^\sharp$, $\gamma \in D^\sharp \longmapsto D^\natural$, and $\forall x \in D^\natural : \forall y \in D^\sharp : \alpha(x) \sqsubseteq^\sharp y \Longleftrightarrow x \sqsubseteq^\natural \gamma(y)$. If α is surjective (resp. injective, bijective) then we have a *Galois insertion* written $\underset{\alpha}{\overset{\gamma}{\leftrightarrows\!\!\!\!\!\!\twoheadrightarrow}}$ (resp. *embedding*[3] written $\underset{\alpha}{\overset{\gamma}{\twoheadleftarrow\!\!\!\!\!\!\rightarrow}}$, *bijection* written $\underset{\alpha}{\overset{\gamma}{\twoheadleftarrow\!\!\!\!\!\!\twoheadrightarrow}}$). The use of

[3] If α and γ are Scott-continuous then this is an embedding-projection pair.

Galois connections in abstract interpretation was motivated by the fact that $\alpha(x)$ is the best possible approximation of $x \in D^\natural$ within $D^\sharp$ [Cousot & Cousot, 1977, 1979]. We often use the fact that Galois connections compose[4]. If $\langle D^\flat, \sqsubseteq^\flat\rangle \xleftrightarrow[\alpha_1]{\gamma_1} \langle D^\natural, \sqsubseteq^\natural\rangle$ and $\langle D^\natural, \sqsubseteq^\natural\rangle \xleftrightarrow[\alpha_2]{\gamma_2} \langle D^\sharp, \sqsubseteq^\sharp\rangle$ then $\langle D^\flat, \sqsubseteq^\flat\rangle \xleftrightarrow[\alpha_2\circ\alpha_1]{\gamma_1\circ\gamma_2} \langle D^\sharp, \sqsubseteq^\sharp\rangle$.

Example 13 (Elementwise subset abstraction). If $D^\natural$ is a set and $D^\sharp \subseteq D^\natural$ then the *subset abstraction* is $\langle \wp(D^\natural), \subseteq\rangle \xtwoheadleftrightarrow[\alpha]{\gamma} \langle \wp(D^\sharp), \subseteq\rangle$ where $\alpha(X) \triangleq X \cap D^\sharp$ and $\gamma(Y) \triangleq X \cup \neg D^\sharp$ (where the *complement* of $\mathcal{E} \subseteq \mathcal{D}$ is $\neg\mathcal{E} \triangleq \{x \in \mathcal{D} \mid x \notin \mathcal{E}\}$).

If $@ \in D^\natural \longmapsto D^\sharp$, the *elementwise abstraction* is $\langle \wp(D^\natural), \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle \wp(D^\natural), \subseteq\rangle$, where:

$$\alpha \in \wp(D^\natural) \longmapsto \wp(D^\sharp), \qquad \alpha(X) \triangleq \{@(s) \mid s \in X\}, \tag{6}$$

$$\gamma \in \wp(D^\sharp) \longmapsto \wp(D^\natural), \qquad \gamma(Y) \triangleq \{s \mid @(s) \in Y\}\ . \tag{7}$$

Moreover, if $@$ is surjective then so is α.

If $\mathcal{S} \subseteq D^\natural$ and $@ \in \mathcal{S} \longmapsto D^\sharp$ then by composition, we get the *elementwise subset abstraction* $\langle \wp(D^\natural), \subseteq\rangle \xleftrightarrow[\alpha]{\gamma} \langle \wp(D^\natural), \subseteq\rangle$ where $\alpha(X) \triangleq \{@(x) \mid x \in X \cap \mathcal{S}\}$ and $\gamma \triangleq \{x \mid @(x) \in Y\} \cup \neg\mathcal{S}$. □

Example 14 (Transitive derivation relation). In order to illustrate fixpoint transfer Theorem 11, let us consider the reflexive transitive closure $\tau^\star$ of the transition relation τ. It is the image $\alpha(\vec{\tau^\star})$ of the derivation sequences for τ by the elementwise abstraction $\langle \wp(\Sigma^+), \subseteq\rangle \xtwoheadleftrightarrow[\alpha]{\gamma} \langle \wp(\Sigma \times \Sigma), \subseteq\rangle$ defined by (6) for the abstraction mapping which records initial and final states of derivations:

$$@ \in \Sigma^+ \longmapsto \Sigma \times \Sigma, \qquad @(\sigma_0 \ldots \sigma_{n-1}) \triangleq \langle \sigma_0, \sigma_{n-1}\rangle\ . \tag{8}$$

The classical fixpoint characterization:

$$\tau^\star = \operatorname{lfp} T^\star, \qquad \text{where} \qquad T^\star(\rho) \triangleq 1_\Sigma \cup (\rho \circ \tau) \tag{9}$$

and 1_Σ is the identity relation on states Σ, can be derived from the fixpoint characterization (4) of derivation sequences by fixpoint transfer Theorem 11 since $\alpha \circ \vec{T}(X) = T^\star \circ \alpha(X)$. □

7 Examples of Abstractions

We now give a number of useful abstractions to relate different semantics of algebraic polynomial systems, at various levels of abstraction.

7.1 State Abstraction

Example 15 (Parse tree to protosentence abstraction). A protosentence is a less informative state than a parse tree since the details of the derivation process

[4] contrary to Galois's original definition corresponding to the semi-dual $\langle D^\natural, \sqsubseteq^\natural\rangle \xleftrightarrow[\alpha]{\gamma} \langle D^\sharp, \sqsupseteq^\sharp\rangle$.

are lost. The abstraction mapping $@^{ps} \in \Sigma^{p} \longmapsto \Sigma^{s}$ flatten a parse tree to its tips. For example $@^{ps}([A : b[A : a, A]]) = aA$. Formally:

$$@^{ps}([\Theta]) \triangleq @^{ps}(\Theta), \qquad @^{ps}(x) \triangleq x, \qquad @^{ps}(x : c^{\epsilon \to s}) \triangleq c, \tag{10}$$
$$@^{ps}(x : f^{s_1 \times \ldots \times s_n \to s}[\Theta_{s_1}, \ldots, \Theta_{s_n}]) \triangleq @^{ps}(\Theta_{s_1}) \bullet \ldots \bullet @^{ps}(\Theta_{s_n}) . \qquad \square$$

Example 16 (Context to protosentence abstraction). The same way, the abstraction mapping:

$$@^{cs} \in \Sigma^{c} \longmapsto \Sigma^{s}, \qquad @^{cs}(p[r]q) \triangleq p \bullet r \bullet q$$

abstracts contexts to protosentences. The $@^{cs}$ abstraction does <u>not</u> preserve blocking states (as shown by the counter example $@^{cs}(a[A]) = @^{cs}([a]A) = aA$ but $[a]A$ is a blocking state for $\tau^{c}[\![\mathcal{P}]\!]$ when $\mathcal{P}$ is $A = b(A, A) + c(A) + d(a, A) + a$ while $a[A]$ is not). It follows that in general $\{@(\sigma) \mid \sigma \in \vec{\tau^{\star}}\} \neq \vec{@(\tau)^{\star}}$ so that it is a priori not equivalent to reason on contexts and protosentences. However we observe that:

$$\langle s,\ s'\rangle \in \tau^{c}[\![\mathcal{P}]\!]^{\star} \implies \langle @^{cs}(s),\ @^{cs}(s')\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star},$$

while the inverse implication does not hold. Nevertheless:

Theorem 17. If $x \in \mathcal{X}$, $V \in \mathcal{X} \cup \{c^{\epsilon \to s} \mid c \in \mathcal{F}\}$ and $\langle x,\ p \bullet V \bullet q\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star}$ then $\langle x,\ p'[p'' \bullet V \bullet q'']q'\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star}$, $\langle p' \bullet p'',\ p\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star}$ and $\langle q'' \bullet q',\ q\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star}$.

These abstractions can be extended to sets of states by (6). $\square$

7.2 Transition Abstraction

Given an abstraction mapping $@ \in \Sigma \longmapsto \Sigma^{\sharp}$, a concrete transition system $\langle \Sigma, \tau\rangle$ can be approximated by any abstract transition system $\langle \Sigma^{\sharp},\ \tau^{\sharp}\rangle$ such that:

$$\langle s,\ s'\rangle \in \tau \implies \langle @(s),\ @(s')\rangle \in \tau^{\sharp} . \tag{11}$$

The least such $\subseteq$-upper approximation is:

$$@(\tau) \triangleq \{\langle \xi,\ \xi'\rangle \mid \exists s, s' : \xi = @(s) \wedge \xi' = @(s') \wedge \langle s,\ s'\rangle \in \tau\} . \tag{12}$$

Example 18. Applying (12) to the abstraction (10) of parse trees into protosentences, the parse tree transition system (3) is approximated by $\tau^{s}[\![\mathcal{P}]\!] = @^{ps}(\tau^{p}[\![\mathcal{P}]\!])$ as defined in (1). $\square$

7.3 Derivation Abstraction

An abstraction mapping on states $@ \in \Sigma \longmapsto \Sigma^{\sharp}$ can be extended pointwise to derivation sequences (i.e. non empty sequences of states):

$$@ \in \Sigma^{+} \longmapsto \Sigma^{\sharp^{+}}, \qquad @(\sigma_0 \ldots \sigma_{n-1}) = @(\sigma_0) \bullet \ldots \bullet @(\sigma_{n-1}) .$$

Example 19. According to (10), the parse tree derivation sequence $[A] \stackrel{\mathcal{P}}{\Longrightarrow} [A : b[A, A]] \stackrel{\mathcal{P}}{\Longrightarrow} [A : b[A : a, A]] \stackrel{\mathcal{P}}{\Longrightarrow} [A : b[A : a, A : a]]$ for the algebraic polynomial system $A = b(A, A) + a$ can be approximated by the protosentence derivation $A \stackrel{\mathcal{P}}{\Longrightarrow} AA \stackrel{\mathcal{P}}{\Longrightarrow} aA \stackrel{\mathcal{P}}{\Longrightarrow} aa$. $\square$

If $\langle \Sigma^\sharp, \tau^\sharp \rangle$ is an @-abstraction of $\langle \Sigma, \tau \rangle$ (i.e. satisfying (11)) then the abstraction process can only introduce more derivation sequences:

$$\{@(\sigma) \mid \sigma \in \tau^{\vec{\star}}\} \subseteq \tau^{\sharp^{\vec{\star}}} .$$

We say that the abstraction mapping @ preserves blocking states for τ if and only if:

$$\forall s, s' \in \Sigma : [@(s) = @(s') \wedge \exists s'' : \langle s,\ s'' \rangle \in \tau] \Longrightarrow [\exists s'' : \langle s',\ s'' \rangle \in \tau]$$

In this case, the abstraction of the concrete derivation sequences for $\langle \Sigma, \tau \rangle$ is exactly the set of abstract derivation sequences for $\langle \Sigma^\sharp, @(\tau) \rangle$:

$$\{@(\sigma) \mid \sigma \in \tau^{\vec{\star}}\} = @(\tau)^{\vec{\star}} . \tag{13}$$

8 Lattice of Semantics

A preorder can be defined on semantics $\tau^\natural \in D^\natural$ and $\tau^\sharp \in D^\sharp$ when $\tau^\sharp = \alpha^\sharp(\tau^\natural)$ and $\langle D^\natural, \leq \rangle \underset{\alpha^\sharp}{\overset{\gamma^\sharp}{\leftrightarrows}} \langle D^\sharp, \leqq \rangle$. The quotient poset is isomorphic to M. Ward lattice [Ward, 1942] of upper closure operators $\gamma^\sharp \circ \alpha^\sharp$ on $\langle \wp(\Sigma^+), \subseteq \rangle$, so that we get a lattice of semantics of algebraic polynomial systems which is part of the lattice of abstract interpretations of [Cousot & Cousot, 1977, sec. 8]. We illustrate a few abstract semantics in the lattice below.

9 Bottom-Up/Backward Abstract Semantics of Algebraic Polynomial Systems

For bottom-up analysis, we use a fixpoint semantics of the form:

$$\tau^\star = \operatorname{lfp} B^\star, \qquad \text{where} \qquad B^\star(\rho) \triangleq (\tau \circ \rho) \cup 1_\Sigma . \tag{14}$$

9.1 Compositional Bottom-Up Abstract Semantics

By refining the specification (14), the big-step semantics $\tau[\![\mathcal{P}]\!]^\star$ can be expressed in compositional form, that is by induction on the syntax of algebraic polynomial systems, as an instance of the following bottom-up abstract semantics:

$$S[\![\mathcal{P}]\!] \triangleq \operatorname{lfp}^{\sqsubseteq} B[\![\mathcal{P}]\!], \tag{15}$$

where $\langle \mathbf{L}, \sqsubseteq, \perp, \sqcup \rangle$ is a cpo, $\langle \mathbf{P}, \leq, \triangledown \rangle$ is a poset and the $\sqsubseteq$-monotonic transformer $B[\![S]\!] \in \mathbf{L} \overset{m}{\longmapsto} \mathbf{L}$ and $B[\![P]\!] \in \mathbf{L} \overset{m}{\longmapsto} \mathbf{P}$ are defined compositionally by induction on the syntax S of systems $\mathcal{P}$ and P of polynomials, as follows:

$$\begin{aligned}
B[\![ES]\!]r &\triangleq B[\![E]\!]r \sqcup B[\![S]\!]r, & B[\![x]\!]r &\triangleq x\langle r \rangle, \\
B[\![x = P]\!]r &\triangleq \langle 1 \rangle \sqcup \langle x \multimap B[\![P]\!]r \rangle r, & B[\![f^{\sigma \to s}(L)]\!]r &\triangleq f\langle r \rangle (B[\![L]\!]r), \\
B[\![\Omega]\!]r &\triangleq \langle \Omega \rangle, & B[\![M, L]\!]r &\triangleq B[\![M]\!]r \otimes B[\![L]\!]r, \\
B[\![M + P]\!]r &\triangleq B[\![M]\!]r \triangledown B[\![P]\!]r, & B[\![c^{\epsilon \to s}]\!]r &\triangleq c\langle r \rangle .
\end{aligned} \tag{16}$$

Example 20 (Big-step protosentence semantics). For $\tau^{s}[\![\mathcal{P}]\!]^{\star}$, we have (recall that $\mathcal{V} \triangleq \{c \mid c^{\epsilon \to s} \in \mathcal{F}\} \cup \mathcal{X}$):

$\mathbf{L}^{s}$	$\sqsubseteq^{s}$	$\perp^{s}$	$\sqcup^{s}$	$\mathbf{P}^{s}$	$\leq^{s}$	∇^{s}	$\langle\Omega\rangle^{s}$	$\langle 1\rangle^{s}$	$\langle x \multimap R\rangle^{s} r$	$\otimes^{s}$	$x\langle r\rangle^{s}$	$f\langle r\rangle^{s}(R)$	$c\langle r\rangle^{s}$
$\wp(\mathcal{V}^{\star} \times \mathcal{V}^{\star})$	$\subseteq$	$\emptyset$	$\cup$	$\wp(\mathcal{V}^{\star})$	$\subseteq$	$\cup$	$\emptyset$	$1_{\mathcal{V}^{\star}}$	$\dagger$	$\bullet$	$\ddagger$	R	$\{c\}$

$\dagger$ is $\{\langle pNq,\ p'mq'\rangle \mid \langle p,\ p'\rangle \in r \wedge m \in R \wedge \langle q,\ q'\rangle \in r\}$, $X \bullet Y$ is $\{xy \mid x \in X \wedge y \in Y\}$ and $\ddagger$ is $\{p \mid \langle x,\ p\rangle \in r\}$.

The proof relies on the fact that $\tau^{s}[\![\mathcal{P}]\!]^{\star}$ is a *relational morphism*, that is $r \subseteq \mathcal{V}^{\star} \times \mathcal{V}^{\star}$ such that:

$$\langle m \bullet p,\ q\rangle \in r \iff \exists m', p' : \langle m,\ m'\rangle \in r \wedge \langle p,\ p'\rangle \in r \wedge q = m' \bullet p', \quad (17)$$

which generalizes $\mathcal{F}$-homomorphisms [Meinke & Tucker, 1992]). □

Example 21 (Finitary Powerset $\mathcal{F}$-Algebra Semantics). The same way, the powerset $\mathcal{F}$-algebra semantics $S^{\star}[\![\mathcal{P}]\!]$ of [Courcelle, 1996] is given by choosing $\mathbf{L}^{\star} \triangleq \mathcal{X} \longmapsto \wp(\mathcal{M})$, $\mathbf{P}^{\star} \triangleq \wp(\mathcal{M})$ where $\mathcal{M} \triangleq \bigcup\{\mathcal{M}^{s,\emptyset} \mid s \in \mathcal{S}\}$ is the set of ground monomials, and:

$\sqsubseteq^{\star}$	$\perp^{\star}$	$\sqcup^{\star}$	$\leq^{\star}$	$\nabla^{\star}$	$\langle\Omega\rangle^{\star}$	$\langle 1\rangle^{\star} x$	$\langle x \multimap R\rangle L y$	$\otimes^{\star}$	$x\langle L\rangle^{\star}$	$f\langle L\rangle^{\star}(R)$	$c\langle L\rangle^{\star}$
$\dot{\subseteq}$	$\dot{\emptyset}$	$\dot{\cup}$	$\subseteq$	$\cup$	$\emptyset$	$\emptyset$	$(y = x\ ?\ R\ ¿\ \emptyset)$	$\times$	$L(x)$	$f[R]$	$\{c\}$

where $f^{s_1 \times \ldots \times s_n \to s}[R] \triangleq \{f(x_1, \ldots, x_n) \mid \langle x_1,\ \ldots,\ x_n\rangle \in R\}$. □

9.2 Compositional Bottom-Up Abstract Interpretations

This compositional presentation is preserved by abstract interpretation. If we have defined a concrete semantics $S^{\flat} \in \mathbf{L}^{\flat}$ (15) compositionally (16), we may want, given an abstraction $\langle \mathbf{L}^{\flat},\ \sqsubseteq^{\flat}\rangle \xleftarrow[\alpha]{\gamma} \langle \mathbf{L}^{\sharp},\ \sqsubseteq^{\sharp}\rangle$, to derive an abstract semantics $S^{\sharp} \triangleq \alpha(S^{\flat})$ which can be defined in the same compositional form (15), (16). By the fixpoint transfer theorem Theorem 11, $S^{\sharp} \triangleq \alpha(S^{\flat})$ can be defined in compositional form (15) and (16) if we can check the following conditions:

$$\begin{array}{rclrcl}
\langle \mathbf{L}^{\flat},\ \sqsubseteq^{\flat}\rangle & \xleftarrow[\alpha]{\gamma} & \langle \mathbf{L}^{\sharp},\ \sqsubseteq^{\sharp}\rangle, & \alpha(\langle x \multimap R\rangle^{\flat} r) & = & \langle x \multimap \alpha'(R)\rangle^{\sharp} \alpha(r), \\
\langle \mathbf{P}^{\flat},\ \leq^{\flat}\rangle & \xleftarrow[\alpha']{\gamma'} & \langle \mathbf{P}^{\sharp},\ \leq^{\sharp}\rangle, & \alpha'(r_1 \otimes^{\flat} r_2) & = & \alpha'(r_1) \otimes^{\sharp} \alpha'(r_2), \\
\langle\Omega\rangle^{\sharp} & = & \alpha(\langle\Omega\rangle^{\flat}), & \alpha'(f\langle r\rangle^{\flat}) & = & f\langle \alpha'(r)\rangle^{\sharp}, \\
\langle 1\rangle^{\sharp} & = & \alpha(\langle 1\rangle^{\flat}), & c\langle r\rangle^{\sharp} & = & \alpha'(c\langle r\rangle^{\sharp})\ .
\end{array} \quad (18)$$

This generalizes the homomorphism result of [Mezei & Wright, 1967] for the case of power-set $\mathcal{F}$-algebra (also [Courcelle, 1996, proposition 3.7]).

When $\langle \mathbf{L}^{\sharp},\ \sqsubseteq^{\sharp}\rangle$ satisfies the ascending chain condition, (15) and (16) can be understood as the specification of an abstract interpreter defined in terms of the abstract operations (18) and computing fixpoints iteratively. The abstract interpreter is generic and can be instantiated for particular applications as considered in the next section.

10 Examples of Bottom-Up Algebraic Polynomial Systems Abstract Interpretations

The following few examples are abstractions of bottom-up big-step polynomial system semantics and generalize classical results for context free grammars.

10.1 Examples of Bottom-Up Algebraic Polynomial Systems Semantics

Example 22 (Generated Protolanguage). The abstraction $\alpha(r) \triangleq \lambda x \centerdot \{p \mid \langle x, p\rangle \in r\}$ of the big-step semantics $S^s[\![\mathcal{P}]\!] \triangleq \tau^s[\![\mathcal{P}]\!]^\star$ provides the protolanguage $S^v[\![\mathcal{P}]\!]x$ generated for each variable $x \in \mathcal{X}$ by the polynomial system $\mathcal{P}$. α' is the identity. The corresponding compositional fixpoint definition is obtained from (15) and (16) with $\mathbf{L}^v \triangleq \mathcal{X} \longmapsto \wp(\mathcal{V}^\star)$, $\mathbf{P}^v \triangleq \wp(\mathcal{V}^\star)$ and:

$\sqsubseteq^v$	$\bot^v$	$\sqcup^v$	$\leq^v$	∇^v	$\langle\Omega\rangle^v$	$\langle 1\rangle^v x$	$\langle x \multimap R\rangle^v L y$	$\otimes^v$	$\langle x\rangle^v L$	$\langle f\rangle^v L(R)$	$\langle c\rangle^v L$
$\dot{\subseteq}$	$\dot{\emptyset}$	$\dot{\cup}$	$\subseteq$	$\cup$	$\emptyset$	$\{x\}$	$(y = x \mathrel{?} R \mathrel{¿} \emptyset)$	$\bullet$	$L(x)$	R	$\{c\}$

where the test $(b \mathrel{?} t \mathrel{¿} e)$ is t is b is true ($\mathtt{tt} \in \mathbb{B}$) and e if b is false ($\mathtt{ff} \in \mathbb{B}$) and $\langle \mathcal{X} \longmapsto \wp(\mathcal{V}^\star), \dot{\subseteq}, \dot{\emptyset}, \lambda x \centerdot \mathcal{V}^\star, \dot{\cup}, \dot{\cap}\rangle$ is the pointwise extension of the complete lattice $\langle \wp(\mathcal{V}^\star), \subseteq, \emptyset, \mathcal{V}^\star, \cup, \cap\rangle$. □

Example 23 (Generated Language). Ginsburg & Rice [1962] and Schützenberger [1962] fixpoint characterization of the language generated by a context-free grammar is easily generalized to polynomial systems by the further abstraction $S^t[\![\mathcal{P}]\!] \triangleq \alpha(S^v[\![\mathcal{P}]\!])$ which consists in ignoring nonterminal sentences: $\alpha(L) \triangleq \lambda x \centerdot \{p \mid p \in L(x) \cap \mathcal{T}^\star\}$ where $\mathcal{T} \triangleq \{c \mid c^{\epsilon \to s} \in \mathcal{F}\}$. $S^t[\![\mathcal{P}]\!]$ is defined by (15) and (16) with $\mathbf{L}^t \triangleq \mathcal{X} \longmapsto \wp(\mathcal{T}^\star)$, $\mathbf{P}^t \triangleq \wp(\mathcal{T}^\star)$ and:

$\sqsubseteq^t$	$\bot^t$	$\sqcup^t$	$\leq^t$	∇^t	$\langle\Omega\rangle^t$	$\langle 1\rangle^t x$	$\langle x \multimap R\rangle^t L y$	$\otimes^t$	$\langle x\rangle^t L$	$\langle f\rangle^t L(R)$	$\langle c\rangle^t L$
$\dot{\subseteq}$	$\dot{\emptyset}$	$\dot{\cup}$	$\subseteq$	$\cup$	$\emptyset$	$\emptyset$	$(y = x \mathrel{?} R \mathrel{¿} \emptyset)$	$\bullet$	$L(x)$	R	$\{c\}$

.

This is also an abstraction of the powerset $\mathcal{F}$-algebra semantics of Example 21 by $\alpha(L) \triangleq \lambda x \centerdot \{@(m) \mid m \in L(x)\}$ where $@(c) \triangleq c$ and $@(f^{s_1 \times \ldots \times s_n \to}(m_{s_1}, \ldots, m_{s_n})) \triangleq @(m_{s_1}) \bullet \ldots \bullet @(m_{s_n})$. □

10.2 Examples of Bottom-Up Algebraic Polynomial Systems Analysis

Other classical examples of abstraction for grammars are also easily generalizable to algebraic polynomial systems such as variable productivity, the set $\textsc{First}[\![\mathcal{P}]\!](x)$ of constants c that begin the strings derived from x, etc. [Wilhelm & Maurer, 1995]. In this last case, empty grammatical production $N \to \varepsilon$ have no direct counterpart for algebraic polynomial systems, so that we can define $\textsc{First}[\![\mathcal{P}]\!](E)x$ as the set of constants c which can start a protosentence deriving from variable x when erasing all symbols in E or ε if the string resulting from this erasure is empty. Observe that the two definitions coincide when grammars are translated into polynomial systems as specified in Example 2 and $E = \{\varepsilon\}$. Formally, we define:

$$\textsc{First}[\![\mathcal{P}]\!](x) \triangleq \alpha_E(S^t[\![\mathcal{P}]\!]), \qquad @_E(c \bullet p) \triangleq (c \in E \mathrel{?} @_E(p) \mathrel{¿} x),$$

$$\alpha_E(r)x \triangleq \{@_E(p) \mid \langle x, p\rangle \in r\}, \qquad @(\epsilon) \triangleq \varepsilon\ .$$

In compositional form, $\textsc{First}\llbracket\mathcal{P}\rrbracket(\mathrm{E})$ is defined by (15) and (16) where $\mathbf{L} \triangleq \mathcal{X} \longmapsto \wp(\mathcal{T})$, $\mathbf{P} \triangleq \wp(\mathcal{T})$ and:

$\sqsubseteq$	$\perp$	$\sqcup$	$\leq$	$\triangledown$	$\langle\Omega\rangle$	$\langle 1\rangle x$	$\langle x \multimapinv R\rangle L y$	$X \otimes Y$	$x\langle L\rangle$	$f\langle L\rangle(R)$	$c\langle L\rangle$
$\dot{\subseteq}$	$\dot{\emptyset}$	$\dot{\cup}$	$\subseteq$	$\cup$	$\emptyset$	$\emptyset$	$(\,y = x \mathrel{?} R \mathrel{\text{¿}} \emptyset\,)$	$\dagger$	$L(x)$	R	$\{c\}$

$\mathcal{T} \triangleq \{c \mid c^{\epsilon \longrightarrow s} \in \mathcal{F}\} \setminus E$ and $\dagger$ is $X \cup (\,X \cap E \neq \emptyset \mathrel{?} Y \mathrel{\text{¿}} \emptyset\,) \cup \{\varepsilon \mid X \cap E \neq \emptyset \wedge Y \cap E \neq \emptyset\}$.

11 Top-Down/Forward Compositional Abstract Semantics of Algebraic Polynomial Systems

Top-down/forward abstract semantics are abstractions of derivation sequence semantics (4), such as (9). The big-step semantics $\tau\llbracket\mathcal{P}\rrbracket^{\star}$ can be defined as an instance of the top-down abstract semantics:

$$\mathrm{S}\llbracket\mathcal{P}\rrbracket \triangleq \mathrm{lfp}^{\sqsubseteq}\,\mathrm{T}\llbracket\mathcal{P}\rrbracket,$$

where $\langle \mathbf{L}, \sqsubseteq, \perp, \sqcup\rangle$ is a cpo, $\langle \mathbf{P}, \leq, \triangledown\rangle$ is a poset, the $\sqsubseteq$-monotonic transformer $\mathrm{T}\llbracket S\rrbracket \in \mathbf{L} \overset{m}{\longmapsto} \mathbf{L}$ and $\mathrm{T}\llbracket P\rrbracket \in \mathbf{P}$ are defined compositionally by induction on the syntax S of $\mathcal{P}$ and P of polynomials, as follows:

$$\begin{aligned}
\mathrm{T}\llbracket E S\rrbracket r &\triangleq \mathrm{T}\llbracket E\rrbracket r \sqcup \mathrm{T}\llbracket S\rrbracket r, & \mathrm{T}\llbracket x\rrbracket &\triangleq \langle x\rangle,\\
\mathrm{T}\llbracket x = P\rrbracket r &\triangleq \langle 1\rangle \sqcup \langle x \multimap \mathrm{T}\llbracket P\rrbracket\rangle r, & \mathrm{T}\llbracket f^{\sigma\longrightarrow s}(L)\rrbracket &\triangleq \langle f\rangle(\mathrm{T}\llbracket L\rrbracket),\\
\mathrm{T}\llbracket \Omega\rrbracket &\triangleq \langle\Omega\rangle, & \mathrm{T}\llbracket M, L\rrbracket &\triangleq \mathrm{T}\llbracket M\rrbracket \otimes \mathrm{T}\llbracket L\rrbracket, \quad (19)\\
\mathrm{T}\llbracket M\ [d_o] + P\rrbracket &\triangleq \langle d_o\rangle \mathrm{T}\llbracket M\rrbracket \triangledown \mathrm{T}\llbracket P\rrbracket, & \mathrm{T}\llbracket c^{\epsilon\longrightarrow s}\rrbracket &\triangleq \langle c\rangle\ .\\
\mathrm{T}\llbracket M\ [d_o]\rrbracket &\triangleq \langle d_o\rangle \mathrm{T}\llbracket M\rrbracket,
\end{aligned}$$

For the big-step derivation tree semantics $\tau^{d}\llbracket\mathcal{P}\rrbracket^{\star}$, we choose $\mathbf{L}^{d} \triangleq \wp(\Sigma^{d} \times \Sigma^{d})$, $\mathbf{P}^{d} \triangleq \mathcal{D} \longmapsto \wp(\Sigma^{d})$ and:

$\sqsubseteq^{d}$	$\perp^{d}$	$\sqcup^{d}$	$\leq^{d}$	$\triangledown^{d}$	$\langle d_o\rangle M$	$\langle\Omega\rangle^{d}$	$\langle 1\rangle^{d}$	$\langle x \multimap R\rangle^{d} r$	$\otimes^{d}$	$\langle x\rangle^{d}$	$\langle f\rangle^{d} R$	$\langle c\rangle^{d}$
$\subseteq$	$\emptyset$	$\cup$	$\dot{\subseteq}$	$\dot{\cup}$	$\dagger$	$\lambda d_o \bullet \emptyset$	$1_{\Sigma^{d}}$	$\ddagger$	$\times$	$\{x, \langle x\rangle\}$	$f^{\star}(R)$	$\{c\}$

$\ddagger$ is $\{\langle T, T''\rangle \mid \langle T, T'\rangle \in r \wedge d_o \in \mathcal{D} \wedge R(d_o) \neq \emptyset \wedge T'' = T'\llbracket \lambda y \bullet (\,y = x \mathrel{?} R(d_o) \mathrel{\text{¿}} \emptyset\,)\rrbracket d\}$,

$\dagger$ is $\lambda d \bullet (\,d = d_o \mathrel{?} M \mathrel{\text{¿}} \emptyset\,)$ and $f^{\star}(M) \triangleq \{f(m_{s_1}, \ldots, m_{s_n}) \mid \langle m_{s_1}, \ldots, m_{s_n}\rangle \in M\}$.

For the big-step protosentence semantics $\tau^{s}\llbracket\mathcal{P}\rrbracket^{\star}$, we choose:

$\mathbf{L}^{s}$	$\sqsubseteq^{s}$	$\perp^{s}$	$\sqcup^{s}$	$\mathbf{P}^{s}$	$\leq^{s}$	$\triangledown^{s}$	$\langle d_o\rangle R$	$\langle\Omega\rangle^{s}$	$\langle 1\rangle^{s}$	$\langle x \multimap R\rangle^{s} r$	$\otimes^{s}$	$\langle x\rangle^{s}$	$\langle f\rangle^{s} R$	$\langle c\rangle^{s}$
$\wp(\mathcal{V}^{\star} \times \mathcal{V}^{\star})$	$\subseteq$	$\emptyset$	$\cup$	$\wp(\mathcal{V}^{\star})$	$\subseteq$	$\cup$	R	$\emptyset$	$1_{\mathcal{V}^{\star}}$	$\ddagger$	$\bullet$	$\{x\}$	R	$\{c\}$

where $\ddagger$ is $\{\langle p,\ q m q'\rangle \mid \langle p,\ q x q'\rangle \in r \wedge m \in R\}$.

For the big-step context semantics $\tau^{c}\llbracket\mathcal{P}\rrbracket^{\star}$, we choose:

$\mathbf{L}^{s}$	$\sqsubseteq^{s}$	$\perp^{s}$	$\sqcup^{s}$	$\mathbf{P}^{s}$	$\leq^{s}$	$\triangledown^{s}$	$\langle d_o\rangle R$	$\langle\Omega\rangle^{s}$	$\langle 1\rangle^{s}$	$\langle x \multimap R\rangle^{s} r$	$\otimes^{s}$	$\langle x\rangle^{s}$	$\langle f\rangle^{s} R$	$\langle c\rangle^{s}$
$\wp(\Sigma^{c} \times \Sigma^{c})$	$\subseteq$	$\emptyset$	$\cup$	$\wp(\Sigma^{c})$	$\subseteq$	$\cup$	R	$\emptyset$	$1_{\Sigma^{c}}$	$\ddagger$	$\bullet$	$\{x\}$	R	$\{c\}$

where $\ddagger$ is $\{\langle p,\ q[q' m \ell]\ell'\rangle \mid \langle p,\ q[q' x \ell]\ell'\rangle \in r \wedge m \in R\}$.

12 Examples of Top-Down Algebraic Polynomial Systems Analysis

We consider a few examples which are abstractions of the top-down semantics and are classical applications for context grammars [Wilhelm & Maurer, 1995].

Accessible Variables: Given an axiom $A \in \mathcal{X}$, a variable x is accessible (written $\textsc{Reachable}(x)$) if and only if $\langle A,\ pxq\rangle \in \tau^{s}[\![\mathcal{P}]\!]^{\star}$. It is not possible to use Kleene fixpoint transfer Theorem 11 or Tarski's fixpoint transfer Theorem 12 with the big-step protosentence semantics $\tau^{s}[\![\mathcal{P}]\!]^{\star}$. The abstraction would be $\textsc{Reachable}(x) = \alpha(\tau^{s}[\![\mathcal{P}]\!]^{\star})$ with $\alpha(r) = \lambda x \centerdot \langle A,\ pxp'\rangle \in r$. For equation $x = P$, we would have:

$$
\begin{aligned}
& \alpha(\langle x \multimap R\rangle^{s} r) \\
= {} & \lambda x \centerdot \langle A,\ pxp'\rangle \in \{\langle p,\ qmq'\rangle \mid \langle p,\ qxq'\rangle \in r \wedge m \in R\} \qquad (\text{defs. } \alpha \ \& \ \langle x \multimap R\rangle^{s}) \\
= {} & \lambda x \centerdot pxp' = qmq' \wedge \langle A,\ qxq'\rangle \in r \wedge m \in R \ .
\end{aligned}
$$

The difficulty is now that we have to consider the cases when x occur in q or q' for which no inductive information is available. Fortunately, thanks to (17), we can avoid this phenomenon using the big-step context semantics $\tau^{c}[\![\mathcal{P}]\!]^{\star}$ in compositional form (19) where:

$\mathbf{L}$	$\sqsubseteq$	$\perp$	$\sqcup$	$\mathbf{P}$	$\leq$	∇	$\langle d_0\rangle X$	$\langle \Omega\rangle$	$\langle 1\rangle y$	$\langle x \multimap X\rangle ry$	$\otimes$	$\langle x\rangle$	$\langle f\rangle X$	$\langle c\rangle$
$\mathcal{X} \longmapsto \mathbb{B}$	$\dot{\Longrightarrow}$	$\dot{\mathrm{ff}}$	$\dot{\vee}$	$\wp(\mathcal{X})$	$\subseteq$	$\cup$	X	$\emptyset$	$y = A$	$r(x) \wedge y \in X$	$\cup$	$\{x\}$	X	$\emptyset$

.

$\textsc{Follow}$*:* The set $\textsc{Follow}[\![\mathcal{P}]\!](E)x$ of constants $c \in \mathcal{T}$ which, after erasure of the symbols belonging to E, can follow variable x in a protosentence derived from the axiom $A \in \mathcal{X}$ or $\dashv$ if x can appear at the end of such a protosentence. Again by (17), we can use the big-step context semantics $\tau^{c}[\![\mathcal{P}]\!]^{\star}$ with the abstraction.

$$
\begin{aligned}
\alpha_1(r) & \triangleq \lambda x \centerdot \{c \in \textsc{First}'(m'p') \mid \langle [A],\ p[mxm']p'\rangle \in r\}, \\
\textsc{First}'(p) & \triangleq \textsc{First}(p) - \{\varepsilon\} \cup (\ \varepsilon \in \textsc{First}(p)\ ?\ \{\dashv\}\ ¿\ \emptyset\ )\ .
\end{aligned}
$$

This does not exactly lead to the classical algorithm [Aho et al., 1986] which relies on the fact that all nonterminals are assumed to be accessible. We get $\mathbf{L} \triangleq \mathcal{X} \longmapsto \wp(\mathcal{T})$, $\mathbf{P} \triangleq \wp(\mathcal{V}^{\star})$ and:

$\sqsubseteq$	$\perp$	$\sqcup$	$\leq$	∇	$\langle d_0\rangle M$	$\langle \Omega\rangle$	$\langle 1\rangle x$	$\langle x \multimap R\rangle r$	$\otimes$	$\langle x\rangle$	$\langle f\rangle R$	$\langle c\rangle$
$\dot{\subseteq}$	$\dot{\emptyset}$	$\dot{\cup}$	$\subseteq$	$\cup$	M	$\emptyset$	$(\ x = A\ ?\ \{\dashv\}\ ¿\ \emptyset\ )$	$\ddagger$	$\bullet$	$\{x\}$	R	c

where $\ddagger$ is $\lambda y \centerdot (\ \textsc{Reachable}(x) \wedge pyq \in R\ ?$
$\textsc{First}(q) - \{\varepsilon\} \cup (\ \varepsilon \in \textsc{First}(q)\ ?\ r(x)\ ¿\ \emptyset\ )\ ¿\ \emptyset\ )\ .$

Since $\textsc{Follow}$ makes use of $\textsc{Reachable}$, we can use the reduced product of the two abstractions [Cousot & Cousot, 1979] (whereas the lattice lifting technique used in [Jeuring & Swierstra, 1995] is specific to the combination with $\textsc{Reachable}$).

13 Generalization to Infinite Terms

Let $\mathcal{M}^{\star}_{s,\mathcal{X}}$ (respectively $\mathcal{M}^{\omega}_{s,\mathcal{X}}$) be the set of finite (resp. infinite) trees of root sort $s \in \mathcal{S}$ built with the function symbols $\mathcal{F}$ and variables $\mathcal{X}$. The finite trees are isomorphic with monomials $\mathcal{M}^{s,\mathcal{X}}$. Let $\mathcal{M}^{\infty}_{s,\mathcal{X}} \triangleq \mathcal{M}^{\star}_{s,\mathcal{X}} \cup \mathcal{M}^{\omega}_{s,\mathcal{X}}$. We drop the s subscript when considering the join for all sorts $s \in \mathcal{S}$ and the $\mathcal{X}$ subscript when $\mathcal{X} = \emptyset$ that is for ground trees. The infinitary powerset $\mathcal{F}$-algebra semantics $S^{\infty}[\![\mathcal{P}]\!]$ is defined compositionally by (15) and (16) when choosing $\mathbf{L}^{\infty} \triangleq \mathcal{X} \longmapsto \wp(\mathcal{M}^{\infty})$, $\mathbf{P}^{\infty} \triangleq \wp(\mathcal{M}^{\infty})$ and:

$\sqsubseteq^{\infty}$	$\bot^{\infty}$	$\sqcup^{\infty}$	$\leq^{\infty}$	$\triangledown^{\infty}$	$\langle\Omega\rangle^{\infty}$	$\langle 1\rangle^{\infty}x$	$\langle x \multimap R\rangle ry$	$\otimes^{\infty}$	$x\langle r\rangle^{\infty}$	$f\langle r\rangle^{\infty}(R)$	$c\langle r\rangle^{\infty}$
$\dot{\supseteq}$	$\lambda x^s \bullet \mathcal{M}^s$	$\dot{\cap}$	$\subseteq$	$\cup$	$\emptyset$	$\emptyset$	$(r = x\ ?\ R\ ¿\ \emptyset)$	$\times$	$r(x)$	$f[R]$	$\{c\}$

.

Define the finite projection $X^{\star} = X \cap \mathcal{M}^{\star}$ and the infinite projection $X^{\omega} = X \cap \mathcal{M}^{\omega}$. The abstraction to finite sentences defined by the Galois connection $\langle \mathcal{X} \longmapsto \wp(\mathcal{M}^{\infty}),\ \dot{\sqsubseteq}^{\infty}\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{X} \longmapsto \wp(\mathcal{M}^{\star}),\ \dot{\subseteq}\rangle$, where $\alpha(r) \triangleq \lambda x \bullet r(x)^{\star}$ and $\gamma(r) \triangleq \lambda x \bullet r(x) \cup \mathcal{M}^{\omega}$, yields to the finitary powerset $\mathcal{F}$-algebra semantics $S^{\star}[\![\mathcal{P}]\!]$ of [Courcelle, 1996] considered at Example 21 in greatest fixpoint form[5]. The abstraction to infinite terms defined by the Galois connection $\langle \mathcal{X} \longmapsto \wp(\mathcal{M}^{\infty}),\ \dot{\sqsubseteq}^{\infty}\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{X} \longmapsto \wp(\mathcal{M}^{\omega}),\ \dot{\supseteq}\rangle$, where $\alpha(r) \triangleq \lambda x \bullet r(x)^{\omega}$ and $\gamma(r) \triangleq \lambda x \bullet r(x) \cup \mathcal{M}^{\star}$, yields to a generalization $S^{\omega}[\![\mathcal{P}]\!]$ of Nivat's [1977, 1978] greatest fixpoint characterization of the infinite language generated by grammar to algebraic polynomial systems. We get $\mathbf{L}^{\omega} \triangleq \mathcal{X} \longmapsto \wp(\mathcal{M}^{\omega})$, $\mathbf{P}^{\omega} \triangleq \wp(\mathcal{M}^{\omega})$ and:

$\sqsubseteq^{\omega}$	$\bot^{\omega}$	$\sqcup^{\omega}$	$\leq^{\omega}$	$\triangledown^{\omega}$	$\langle\Omega\rangle^{\omega}$	$\langle 1\rangle^{\omega}x$	$\langle x \multimap R\rangle ry$	$\otimes^{\omega}$	$x\langle r\rangle^{\omega}$	$f\langle r\rangle^{\omega}(R)$	$c\langle r\rangle^{\omega}$
$\dot{\supseteq}$	$\lambda x^s \bullet \mathcal{M}^{\omega}_s$	$\dot{\cap}$	$\subseteq$	$\cup$	$\emptyset$	$\emptyset$	$(r = x\ ?\ R \cap \mathcal{M}^{\omega}\ ¿\ \emptyset)$	$\times$	$r(x)$	$f[R]$	$\{c\}$

.

The least fixpoint $\mathrm{lfp}^{\dot{\supseteq}}\ B[\![\mathcal{P}]\!]$ in (15) is Nivat's greatest for $\dot{\subseteq}$.

14 Infinite Abstract Domains

As an example of infinite abstract domain, let us consider the abstraction of terminal sentences by the vector of number of instances of each terminal in the sentence: $@(\varepsilon)T = 0$, $@(T\sigma)T = 1 + @(\sigma)T$, $@(T\sigma)T' = @(\sigma)T'$ when $T \neq T'$. The extension to languages is elementwise (6). By Parikh's theorem [1966], the abstraction of the generated language is the union of a finite number

[5] The complete lattice $\langle \wp(\mathcal{M}^{\infty}), \sqsubseteq^{\infty}, \bot^{\infty}, \top^{\infty}, \sqcup^{\infty}, \sqcap^{\infty}\rangle$ with generalized Scott's ordering $X \sqsubseteq^{\infty} Y \triangleq X^{\star} \subseteq Y^{\star} \wedge X^{\omega} \supseteq Y^{\omega}$, infimum $\bot^{\infty} \triangleq \mathcal{M}^{\omega}$, supremum $\top^{\infty} \triangleq \mathcal{M}^{\star}$, lub $\bigsqcup^{\infty}_{i\in\Delta} X_i \triangleq \bigcup_{i\in\Delta} X_i{}^{\star} \cup \bigcap_{i\in\Delta} X_i{}^{\omega}$ and glb $\bigsqcap^{\infty}_{i\in\Delta} X_i \triangleq \bigcap_{i\in\Delta} X_i{}^{\star} \cup \bigcup_{i\in\Delta} X_i{}^{\omega}$ introduced in [Cousot & Cousot, 1992] to provide a fixpoint characterization of the finite and infinite execution traces of a transition system cannot be *directly* generalized. A counter example is $x = a + b(x) + c(x, x)$ since the iterates start with $X^0 = \mathcal{M}^{\omega}$, so that $X^1 = \{a, b(x), c(x, y) \mid x, y \in X^0\}$, hence the limit, does not contain the infinite tree with equation $t = c(b(a)), t)$ with finite subtree $b(a) \notin X^0$. When least and greatest fixpoints may differ, we can resort to the traditional topological notions based on the prefix ordering [see e.g. de Bakker et al., 1983] which naturally generalizes to inverse limits [Meinke & Tucker, 1992] but then their is a "discontinuity" in the passage to the limit, all prefixes being finite objects while the limit is an infinite one. This causes problems when considering abstractions (which are not admissible in the sense of Scott-induction).

of linear sets. Another abstraction consists in taking the convex-hull of the semilinear set, using a widening to speed-up convergence [Cousot & Halbwachs, 1978] that would allow us to determine relationships such as "the number of a's in a sentence is greater than twice the number of b's". This is not possible in the restricted framework of [Wilhelm & Maurer, 1995].

15 Conclusion

Traditional abstract interpretations have been mainly based on small step/big-step operational semantics or denotational semantics whereas algebraic semantics have been rather neglected, except may-be for Prolog. The difficulty is often that the proposed analyses are quite dependent on the considered programming language semantics. Since there is no universal semantics, the analyzes can be hard to generalize from one semantic framework to another. We think that the use of algebraic semantics could solve this problem since many analyzes can be expressed using the algebraic metastructure in a language independent way. As a first step, algebraic polynomial systems which generalize context-free grammars with their fixpoint semantics seem to be very well suited for expressing first-order fixpoint abstract semantics. To do this the semantics of algebraic polynomial systems must be extended to infinite terms, so as e.g. to be able to encode infinite execution paths. The program abstract semantics is then one of the semantics of the algebraic polynomial system representing the abstract equations e.g. the protosentence semantics where terminals correspond to atomic actions of the program while variables correspond to procedure calls or using the Follow abstraction, the set of atomic actions which can follow a call to a given procedure.

Algebraic polynomial systems could themselves be used as abstract values in program analysis, the abstract program analysis equations being now algebraic polynomial system transformers. This would generalize grammar-based or set-based analysis abstract interpretations [Cousot and Cousot, 1995; Heintze, 1992; Jones and Muchnick, 1981; Uhl and Horspool, 1994]. It remains to explore the idea that abstract semantics of the algebraic polynomial system resulting from the abstract equations resolution would yield a hierarchy of information on the program executions abstracted by the algebraic polynomial system.

Acknowledgments

We thank A. Venet for his comments on a preliminary version of this paper.

Bibliography

A.V. Aho, R. Sethi & J.D. Ullman. *Compilers. Principles, Technique and Tools.* Addison-Wesley, 1986.

B. Courcelle. The monadic second-order logic of graphs X: Linear orders. *TCS*, 160:87–143, 1996.

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

P. Cousot & R. Cousot. Systematic design of program analysis frameworks. *6th ACM POPL*, pp. 269–282, 1979.

P. Cousot & R. Cousot. Inductive definitions, semantics and abstract interpretation. *19th ACM POPL*, pp. 83–94, 1992.

P. Cousot & R. Cousot. Higher-order abstract interpretation (and application to comportment analysis generalizing strictness, termination, projection and PER analysis of functional languages), invited paper. *Proc. 1994 ICCL*, pp. 95–112, 1994. IEEE Comp. Soc. Press.

P. Cousot and R. Cousot. Formal language, grammar and set-constraint-based program analysis by abstract interpretation. *Proc. 7th ACM FPCA*, pp. 170–181, 1995.

P. Cousot & N. Halbwachs. Automatic discovery of linear restraints among variables of a program. *5th ACM POPL*, pp. 84–97, 1978.

J.W. de Bakker, J.-J.Ch. Meyer & J.I. Zucker. On infinite computations in denotational semantics. *TCS*, 26:53–82, 1983. (Corrigendum: TCS 29:229–230, 1984).

S. Ginsburg & G. Rice. Two families of languages related to ALGOL. *J. ACM*, 9:350–371, 1962.

N. Heintze. *Set Based Program Analysis*. PhD thesis, CMU, Pittsburgh, 1992.

J. Jeuring & D. Swierstra. Bottom-up grammar analysis — a functional formulation —. *Proc. ESOP '94*, LNCS 788, pp. 317–332, 1994. Springer-Verlag.

J. Jeuring & D. Swierstra. Constructing functional programs for grammar analysis problems. *Proc. 7th ACM FPCA*, pp. 259–269, 1995.

N.D. Jones and S.S. Muchnick. Flow-analysis and optimization of Lisp-like structures. In S.S. Muchnick & N.D. Jones, editors, *Program Flow Analysis: Theory and Applications*, pp. 102–131. Prentice-Hall, 1981.

K. Meinke & J.V. Tucker. Universal algebra. In S. Abramsky, D.M. Gabbay & T.S.E. Maibaum, editors, *Background: Mathematical Structures*, vol. 1 of *Handbook of Logic in Com . Sci.*, ch. 3, pp. 189–411. Clarendon Press, 1992.

J. Mezei & J. Wright. Algebraic automata and context-free sets. *Inf. & Cont.*, 11:3–29, 1967.

U. Möncke & R. Wilhelm. Grammar flow analysis. *Proc. Int. Summer School SAGA*, LNCS 545, pp. 151–186, 1991. Springer-Verlag.

M. Nivat. Mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique*, 11:311–327, 1977.

M. Nivat. Sur les ensembles de mots infinis engendrés par une grammaire algébrique. *RAIRO Informatique Théorique*, 12:259–278, 1978.

R.J. Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966.

M.P. Schützenberger. On a theorem of R. Jungen. *Proc. Amer. Math. Soc.*, 13:885–889, 1962.

J.S. Uhl and R.N. Horspool. Flow grammars — a flow analysis methodology. *Proc. CC '94*, LNCS 786, pp. 203–217, 1994. Springer-Verlag.

M. Ward. The closure operators of a lattice. *Ann. Math.*, 43:191–196, 1942.

R. Wilhelm & D. Maurer. *Compiler Design*. Addison-Wesley, 1995.

M. Wirsing. Algebraic specification. In J. van Leeuwen, editor, *Formal Models and Semantics*, vol. B of *Handbook of TCS*, ch. 13, pp. 675–788. Elsevier, 1990.

Published in M. Johnson, editor, *Proceedings of the Sixth International Conference on Algebraic Methodology and Software Technology, AMAST '97*, Sydney, Australia, 13–18 December 1997. Lecture Notes in Computer Science 1349, pages 138–154. Springer-Verlag, Berlin, Germany.