An Algebraic Treatment of Recursion

Rob van Glabbeek
Data61, CSIRO, Sydney, Australia
School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

Jan Bergstra has put his mark on theoretical computer science by a consistent stream of original ideas, controversial opinions, and novel approaches. He sometimes reorganised the arena, enabling others to follow. I, for one, might never have entered computer science if it wasn't for Jan's support and encouragement, and will never forget the team spirit in the early days of process algebra in his group at CWI. This paper is dedicated to Jan, at the occasion of his 65th birthday and retirement.

I review the three principal methods to assign meaning to recursion in process algebra: the denotational, the operational and the algebraic approach, and I extend the latter to unguarded recursion.

1 Process Algebra

In process algebra, processes are often modelled as closed terms of single-sorted specification languages.

Definition 1 Let *Var* be a set of variables. A *signature* is a set of pairs (f,n) of a *function symbol* $f \notin \mathit{Var}$ and an *arity* $n \in \mathbb{N}$. The set $\mathbb{T}(\Sigma)$ of *terms* over a signature Σ is generated by:

- $\mathit{Var} \subseteq \mathbb{T}(\Sigma)$,
- if $(f,n) \in \Sigma$ and $t_1,\dots,t_n \in \mathbb{T}(\Sigma)$ then $f(t_1,\dots,t_n) \in \mathbb{T}(\Sigma)$,
- If $V_\mathscr{S} \subseteq \mathit{Var}$, $\mathscr{S} : V_\mathscr{S} \to \mathbb{T}(\Sigma)$ and $X \in V_\mathscr{S}$, then $\langle X|\mathscr{S}\rangle \in \mathbb{T}(\Sigma)$.

A function $\mathscr{S}$ as appears in the last clause is called a *recursive specification*. A recursive specification $\mathscr{S}$ is often displayed as $\{X = \mathscr{S}_X \mid X \in V_\mathscr{S}\}$. An occurrence of a variable y in a term t is *free* if it does not occur in a subterm of the form $\langle X|\mathscr{S}\rangle$ with $y \in V_\mathscr{S}$. A term is *closed* if it contains no free occurrences of variables.

The *semantics* of such a language is a function $[\![\_]\!] : \mathbb{T}(\Sigma) \to (\mathbb{D}^{\mathit{Var}} \to \mathbb{D})$. It assigns to every term $t \in \mathbb{T}(\Sigma)$ its meaning $[\![t]\!] \in \mathbb{D}^{\mathit{Var}} \to \mathbb{D}$. The meaning of a closed term is a *value* chosen from a class of values $\mathbb{D}$, called a *domain*. The meaning of an open term is a *Var-ary operator* on $\mathbb{D}$: a function of type $\mathbb{D}^{\mathit{Var}} \to \mathbb{D}$. It associates a value $[\![t]\!](\rho) \in \mathbb{D}$ to t that depends on the choice of a *valuation* $\rho : \mathit{Var} \to \mathbb{D}$.

Sometimes, only a subset of $\mathbb{T}(\Sigma)$ is given a semantics, for instance by restricting to terms satisfying a syntactic criterion of *guardedness*.

Another approach lacks the recursion construct itself, but declares a single recursive specification $\mathscr{S} : V_\mathscr{S} \to \mathbb{T}(\Sigma)$ for the entire language [4]. A term t in such a language can be seen as a the term $\langle t|\mathscr{S}\rangle$, obtained from t by substituting, for each $Y \in V_\mathscr{S}$, $\langle Y|\mathscr{S}\rangle$ for each occurrence of Y. Conversely, each term in the general language of Definition 1 can be converted into the form $\langle t|\mathscr{S}\rangle$ with t and $\mathscr{S}$ recursion-free.

2 Denotational, Operational and Algebraic Semantics

The standard (denotational) semantics assigns to each function $(f,n) \in \Sigma$ an n-ary operator $f_n^{\mathbb{D}} : \mathbb{D}^n \to \mathbb{D}$. The semantics of a recursion-free expression t is then given by

- $[\![X]\!](\rho) = \rho(X)$ for $X \in \mathit{Var}$, and
- $[\![f(t_1,\dots,t_n)]\!](\rho) = f_n^{\mathbb{D}}([\![t_1]\!](\rho),\dots,[\![t_n]\!](\rho))$ for $(f,n) \in \Sigma$.

Paper dedicated to Jan Bergstra,
at the occasion of his 65th birthday and retirement.

Three approaches appear in the literature to give semantics to recursion.

The *denotational* approach [2] recognises $[\![\mathscr{S}]\!]$ as having type $\mathbb{D}^{Var\setminus V_{\mathscr{S}}} \rightarrow (\mathbb{D}^{V_{\mathscr{S}}} \rightarrow \mathbb{D}^{V_{\mathscr{S}}})$ and defines $[\![\langle X|\mathscr{S}\rangle]\!](\rho)$ for $\rho \in \mathbb{D}^{Var\setminus V_{\mathscr{S}}}$ to be the X-component of the least fixed point of $[\![\mathscr{S}]\!](\rho)$. For this least fixed point to exists, either $\mathbb{D}$, equipped with a suitable preorder $\sqsubseteq$, needs to be a complete lattice, with the operators $f^{\mathbb{D}}$ monotonic, or $(\mathbb{D},\sqsubseteq)$ be a c.p.o., with the $f^{\mathbb{D}}$ continuous, or $\mathbb{D}$ be a complete metric space, with the $f^{\mathbb{D}}$ contracting (or some variation on this theme).

The *operational* approach [4] is based on a set of inference rules that derive a collection of (labelled) transitions between closed terms. The semantic domain is now the collection $\mathbb{G}$ of *process graphs* (S,T,I), with S a set of states, T a set of transitions between states, and $I \in S$ an initial state, possibly subject to some cardinality restrictions. The operational semantics $[\![P]\!]$ of a closed term P takes S to be the set of closed terms, $I = P$, and T the derivable transitions. The semantics of open terms can be dealt with by encoding the process graphs $\rho(X)$ for $X \in Var$ as constants in an appropriate extension of the process algebra. This approach covers the meaning of recursion constructs too.

Let *guardedness* be a criterion on recursive specifications, such that if $\mathscr{S}$ is guarded then is has a unique solution, meaning that if ρ_i for $i = 1,2$ are valuations with $\rho_1(Z) = \rho_2(Z)$ for all $Z \in Var \setminus V_{\mathscr{S}}$, and $\rho_i(X) = [\![\mathscr{S}_X]\!](\rho_i)$ for all $X \in V_{\mathscr{S}}$, then $\rho_1(X) = \rho_2(X)$ for all $X \in V_{\mathscr{S}}$. The *algebraic* approach [1] yields a semantics for terms with guarded recursion only, where $[\![\langle X|\mathscr{S}\rangle]\!](\rho)$ for $\rho \in \mathbb{D}^{Var\setminus V_{\mathscr{S}}}$ is the X-component of the unique solution of $[\![\mathscr{S}]\!](\rho)$.

3 Extending the Algebraic Approach to Unguarded Recursion

In [3] I proposed an extension of the algebraic approach to unguarded recursion. An expression $\langle X|\mathscr{S}\rangle$ is seen as a kind of variable, only ranging over the solutions of $\mathscr{S}$. Taking for example ACP [1], interpreted in a domain of process graphs modulo strong bisimilarity [1], then $\langle X|X = aX\rangle$ is a case of guarded recursion and denotes a specific process, namely an a-loop. On the other hand, $\langle X|X = X\rangle$ is an unguarded recursion, and seen a variable ranging over all processes, just like X itself. In between, $\langle X|X = X + aX\rangle$ is a case of unguarded recursion, and seen as a variable ranging over all processes of the form a^*P.

To avoid ambiguity in deciding when two, almost identical, processes $\langle X|\mathscr{S}\rangle$ denote the same variable or different ones, here I formalise this approach only for terms $\langle t|\mathscr{S}\rangle$ where no further recursion occurs in t or $\mathscr{S}$, thus following the second approach of Section 1.

A valuation $\rho : Var \rightarrow \mathbb{D}$ is *compatible* with a recursive specification $\mathscr{S}$ iff $\rho(Y) = [\![\mathscr{S}_Y]\!](\rho)$ for all $Y \in V_{\mathscr{S}}$. The meaning $[\![t]\!]$ of a recursion-free term t in the context of a global recursive specification $\mathscr{S}$ is now a function into $\mathbb{D}$ from the set of compatible valuations only. It is obtained from the semantics of t from Section 2 by restricting $dom([\![t]\!])$ to the compatible valuations.

In particular, an equation $t = u$ holds under this semantics iff $[\![t]\!](\rho) = [\![u]\!](\rho)$ for all valuations ρ compatible with $\mathscr{S}$. Hence it is equivalent to the conditional equation $\left(\bigwedge_{X\in V_{\mathscr{S}}} X = \mathscr{S}_X\right) \Rightarrow t = u$.

The laws of process algebra remain valid in this approach, including the congruence property for recursion: if $[\![\mathscr{S}_X]\!](\rho) = [\![\mathscr{S}'_X]\!](\rho)$ for *all* valuations ρ, and all $X \in V_{\mathscr{S}} = V_{\mathscr{S}'}$ then $[\![\langle t|\mathscr{S}\rangle]\!] = [\![\langle t|\mathscr{S}'\rangle]\!]$.

References

[1] J.A. Bergstra & J.W. Klop (1986): *Algebra of Communicating Processes*. In de Bakker, Hazewinkel & Lenstra, editors: *Mathematics & Computer Science I*, *CWI Monograph* 1, North-Holland, pp. 89–138.

[2] S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A theory of communicating sequential processes*. *Journal of the ACM* 31(3), pp. 560–599, doi:10.1145/828.833.

[3] R.J. van Glabbeek (1987): *Bounded nondeterminism and the approximation induction principle in process algebra*. In F. Brandenburg, G. Vidal-Naquet & M. Wirsing, editors: Proc. STACS'87, LNCS 247, Springer, pp. 336–347, doi:10.1007/BFb0039617.

[4] R. Milner (1989): *Communication and Concurrency*. Prentice Hall, Englewood Cliffs.