A note on guarded recursion

Eric Badouel, Philippe Darondeau

UNITÉ DE RECHERCHE
INRIA-RENNES

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

N° 1249

Programme 1
Programmation, Calcul Symbolique et Intelligence Artificielle

A NOTE ON GUARDED RECURSION

Eric BADOUEL
Philippe DARONDEAU

Juin 1990

RR.1249

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE
ET SYSTÈMES ALÉATOIRES

Campus Universitaire de Beaulieu
35042 - RENNES CÉDEX
FRANCE
Téléphone : 99 36 20 00
Télex : UNIRISA 950 473 F
Télécopie : 99 38 38 32

A note on guarded recursion

Eric Badouel - Philippe Darondeau
Irisa, Campus de Beaulieu , 35042 Rennes Cedex

Publication Interne n° 535 - Mai 1990 - 10 Pages

Abstract : *We introduce a logical notion of well-guardedness for recursive terms on arbitrary signatures defined in Plotkin's framework of structural operational specifications, restricted by de Simone's realizability requirements. We then suggest a simpler form for the logical rule that gives the behaviour of a recursively defined expression in terms of the behaviour of its unfoldings. For well-guarded terms, the simplified rule is logically equivalent to the general rule, but is has not the draw-back to ask for premises more complex that consequences.*

Une note sur la récursion gardée

Résumé : *Nous définissons sur des bases logiques la notion d'expression récursive bien gardée. Celle-ci est paramétrée par un ensemble de spécifications opérationnelles respectant le format introduit par Robert de Simone. Nous suggérons alors une forme simplifiée pour la règle qui identifie le comportement d'une expression récursive aux comportements de ses dépliages syntaxiques. Lorsqu'on se restreint aux expressions bien-gardées, la règle simplifiée s'avère logiquement équivalente à la règle générale, mais a l'avantage de ne pas introduire de prémisses structurellement plus complexes que les conclusions.*

CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE
(L. A. 227)
UNIVERSITÉ DE RENNES 1 I. N. S. A. DE RENNES

INSTITUT NATIONAL DE RECHERCHE
EN INFORMATIQUE ET EN AUTOMATIQUE
(LABORATOIRE DE RENNES)

A note on guarded recursion

Eric Badouel - Philippe Darondeau
Irisa, Campus de Beaulieu , 35042 Rennes Cedex - France

Abstract

We introduce a logical notion of well-guardedness for recursive terms on arbitrary signatures defined in Plotkin's framework of structural operational specifications, restricted by de Simone's realizability requirements. We then suggest a simpler form for the logical rule that gives the behaviour of a recursively defined expression in terms of the behaviour of its unfoldings. For well-guarded terms, the simplified rule is logically equivalent to the general rule, but is has not the draw-back to ask for premises more complex that consequences.

Let there be given a signature Σ and a denumerable set of variables X, and let $T(\Sigma, X)$ be the set of 'recursive' terms over X defined by the BNF syntax

$$t ::= x \mid f(t_1, \ldots, t_n) \mid \mathbf{rec}\ x.t$$

where $x \in X$, $f \in \Sigma_n$ is an operator of arity n and $\mathbf{rec}\ x$ binds recursively x to the operand t, entailing the usual notions of free and bound variable. As usual, substitution operations bear upon free variable : $t[u/x]$ stands for the term t in which each free occurrence of x has been replaced by u (after possible renaming of bound variables in t). When closed terms are considered as programs, their behaviour is specified by the elementary transitions $t \xrightarrow{\lambda} u$ a program can perform (the transition $t \xrightarrow{\lambda} u$ reads as "t may perform the action λ and then behave as u"). Following the method known as Structural Operational Semantics (SOS in short) advocated by Plotkin [Plo81], those transitions are defined as the formulae provable in a deductive system. As a concrete example, let us consider the restriction of CCS [Mil80] defined as follows. There is a set of actions $A = \Lambda \cup \{\tau\}$ where τ is the so-called *invisible action* and Λ is equipped with an involutive mapping of synchronization $\overline{()} : \Lambda \to \Lambda$; the complementary actions λ and $\overline{\lambda}$ are those taking place in a communication (delivering and reception of a message). The signature $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ offers a constant for inaction : $\Sigma_0 = \{nil\}$, together with operators for prefixing by an action in Λ : $\Sigma_1 = \{\lambda.\text{-}\ ;\ \lambda \in \Lambda\}$, and with two binary operators for non deterministic choice and parallel composition : $\Sigma_2 = \{+; \|\}$. The behaviour of programs is specified by the rules :

$$
\begin{array}{lll}
nil & : & \\
\lambda.- & : & \lambda.t \xrightarrow{\lambda} t \\
+ & : & \dfrac{t \xrightarrow{\lambda} t'}{t + u \xrightarrow{\lambda} t'} \qquad \dfrac{u \xrightarrow{\lambda} u'}{t + u \xrightarrow{\lambda} u'} \\
\| & : & \dfrac{t \xrightarrow{\lambda} t'}{t\|u \xrightarrow{\lambda} t'\|u} \qquad \dfrac{u \xrightarrow{\lambda} u'}{t\|u \xrightarrow{\lambda} t\|u'} \qquad \dfrac{t \xrightarrow{\lambda} t', u \xrightarrow{\bar{\lambda}} u'}{t\|u \xrightarrow{\tau} t'\|u'}
\end{array}
$$

Those rules are used in conjunction with a specific rule for recursion, stating that a term behaves as any of its syntactical unfoldings. The CCS-rule is the following :

$$\frac{u[\mathbf{rec}\ x.u/x] \xrightarrow{\lambda} v}{\mathbf{rec}\ x.u \xrightarrow{\lambda} v} \quad \mathbf{rec_1}$$

Generalizing on the above example, we will consider the rule $\mathbf{rec_1}$ in conjunction with arbitrary rules in de Simone's format [dS84]. A set of SOS rules is in *de Simone's format* when there is, for each operator $f \in \Sigma_n$, a finite set of schemes of rules which conform to the following pattern :

$$\frac{u_{i_1} \xrightarrow{\lambda_1} v_{i_1}, \ldots, u_{i_k} \xrightarrow{\lambda_k} v_{i_k}}{f(u_1, \ldots, u_n) \xrightarrow{\lambda_0} C[v_1, \ldots, v_n]} \quad \mathrm{R}_{(\lambda_0, \lambda_1, \ldots, \lambda_k)}$$

where :

- $u_1, \ldots u_n, v_{i_1}, \ldots, v_{i_k}$ are different meta-variables taken in a new alphabet, and $u_{i_1}, \ldots, u_{i_k}$ are all distinct ($\{u_{i_1}, \ldots, u_{i_k}\} \subseteq \{u_1, \ldots, u_n\}$),
- for $j \notin \{i_1, \ldots, i_k\}$, $v_j = u_j$,
- $t = C[v_1, \ldots, v_n]$ is a term over $\{v_1, \ldots, v_n\}$ in which each meta-variable v_j occurs at most once and that include no recursive operator $\mathbf{rec}\ x$ nor variable $x \in X$, and
- the side condition $R(\lambda, \lambda_1, \ldots, \lambda_k)$ is a recursive predicate of $n+1$ variables, where λ_i ranges over a recursive set of actions Λ.

The satisfaction of those conditions ensures the 'relizability' of the specified operators in Meije [Bou85]. De Simone's realizability requirements allow neither duplication of arguments $u_1, \ldots, u_n$ by the specified operator f, nor lookahead in their structure. Hence the specified operators f may just synchronize the transitions of their operands. Relying on that structural property, we will prove that the rule $\mathbf{rec_1}$ may be replaced by a simpler rule, in the absence of unguarded recursion, namely by the following :

$$\frac{u \xrightarrow{\lambda} v}{\mathbf{rec}\ x.u \xrightarrow{\lambda} v[\mathbf{rec}\ x.u/x]}\ \mathbf{rec_2}$$

In CCS the prefixing operators $\lambda.-$ are *guards* because their parameter is preserved until the guard-action λ has been performed. This is reflected by the fact that the only rules concerning guard operators are axioms (i.e. rules with empty premises : $k = 0$). The following definition is a straightforward generalization of the notion of guards to arbitrary signatures.

Definition 1 (guard operators and well-guardedness)

Let there be given a signature Σ and a fixed set of SOS rules for the operators in Σ, then

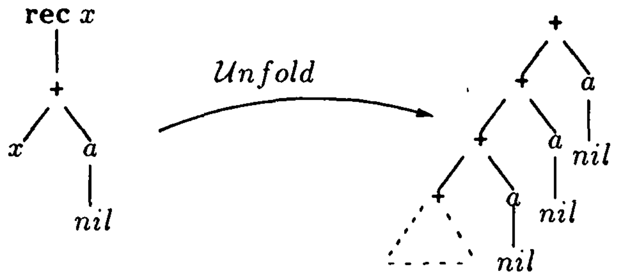
1. *$f \in \Sigma_n$ is a* **guard-operator** *if the only rules concerning that operator are axioms.*

2. *A variable $x \in X$ is* **guarded** *in a term $t \in T(\Sigma, X)$ if every free occurrence of x in t is under the scope of at least one guard.*

3. *A recursion* **rec** *$x.t$ is* **well-guarded** *if x is guarded in t.*

4. *And, a term u is well-guarded is every subterm of u which is a recursion is well-guarded.*

It may be shown that an expression is guarded if, and only if, there is at least one guard (and thus an infinity of guards) on every infinite path of the rational tree produced by unfolding ad infinitum the recursive definition of that term. For instance, the expression **rec** $x.\ (x + a.nil)$ is not well-guarded, and this is reflected by the presence in the corresponding rational tree of an infinite path all nodes of which are labelled with the operator $+$, which is not a guard. Using the general form $(\mathbf{rec_1})$



of the recursion rule, an immediate transition from an unguarded expression (e.g. **rec** $x.\ (x + a.nil)$) may be induced from operators (e.g. the prefixing $a.-$) occurring at an arbitrary depth in the corresponding rational tree. This cannot happen for well-guarded expressions, since transitions from $f(t_1, \ldots, t_n)$ never modify subterms under the scope of guards in the t_i, but by possibly suppressing them. That is the gist of the following lemma which states that if t is under the scope of a guard in $T = u[t/x]$ then any transition from T leaves t unchanged (rewriting and substitution commute) :

Lemma 2 (commutation lemma) *Let there be given a signature Σ and a fixed set of SOS rules for operators in Σ, obeying de Simone's requirements and used in conjunction with the general form* $\mathbf{rec}_1$ *of the recursion rule. If $u[t/x] \xrightarrow{\lambda} w$ for some variable x guarded in u then $u \xrightarrow{\lambda} v$ for some expression v such that $w = v[t/x]$.*

Proof :
The proof proceeds by induction on the structure of the deduction for the transition $u[t/x] \xrightarrow{\lambda} w$:

- $u = f(u_1, \ldots, u_n)$.
  Then $u[t/x] = f(u_1', \ldots, u_n')$ where $u_i' = u_i[t/x]$.

  - The deduction may be the mere application of an axiom :

    $$u[t/x] = f(u_1', \ldots, u_n') \xrightarrow{\lambda} C[u_1', \ldots, u_n'] = w$$

    By the same axiom we get $u \xrightarrow{\lambda} C[u_1, \ldots, u_n] = v$, and thus $w = v[t/x]$.

  - The deduction may be governed by the application of some rule

    $$\frac{u_{i_1}' \xrightarrow{\lambda_1} w_{i_1}, \ldots, u_{i_k}' \xrightarrow{\lambda_k} w_{i_k}}{u[t/x] = f(u_1', \ldots, u_n') \xrightarrow{\lambda_0} w = C[w_1, \ldots, w_n]} \quad \mathrm{R}(\lambda_0, \lambda_1, \ldots, \lambda_k),$$

    where $k \neq 0$ (thus f is not a guard), and $w_j = u_j'$ for $j \notin \{i_1, \ldots, i_k\}$. As x is guarded in $f(u_1, \ldots, u_n)$ and f is not a guard, x is guarded in u_{i_m}. Hence,by induction hypothesis $u_{i_m} \xrightarrow{\lambda_m} v_{i_m}$ for some v_{i_m} such that $w_{i_m} = v_{i_m}[t/x]$. We apply the same rule to deduce the transition

    $$u = f(u_1, \ldots, u_n) \xrightarrow{\lambda} C[v_1, \ldots, v_n] = v \quad \text{where} \quad v_j = u_j \quad \text{for} \quad j \notin \{i_1, \ldots, i_k\}.$$

    $$\text{And then} \quad v[t/x] = C[v_1[t/x], \ldots, v_n[t/x]] = C[w_1, \ldots, w_n] = w$$

    $$\text{because} \begin{cases} w_{i_m} = v_{i_m}[t/x] & \text{for } m \in \{1, \ldots, k\} \\ w_j = u_j' = u_j[t/x] = v_j[t/x] & \text{for } j \notin \{i_1, \ldots, i_k\} \end{cases}$$

- $u = \mathbf{rec}\ y.u'$.

  - $x = y$
    Then $u[t/x] = u$ and we get the result with $v = w$ (there is no free occurrence of x in w).

- $x \neq y$
  Then $u[t/x] = \mathbf{rec}\ y.u'[t/x]$ and the transition $u[t/x] \xrightarrow{\lambda} w$ has been inferred from the premise $(u'[t/x])[\mathbf{rec}\ y.u'[t/x]/y] \xrightarrow{\lambda} w$. We can assume without loss of generality that y does not occur free in t (otherwise, we rename the bound variable y in u in order to avoid the capture of the corresponding free variable of t in the substitution $u[t/x]$). Thus the cause of $u[t/x] \xrightarrow{\lambda} w$ is the transition $(u'[\mathbf{rec}\ y.u'/y])[t/x] \xrightarrow{\lambda} w$ and the induction on the structure of the deduction applies, yielding v such that $w = v[t/x]$ and $u'[\mathbf{rec}\ y.u'/y] \xrightarrow{\lambda} v$, wherefrom $u = \mathbf{rec}\ y.u' \xrightarrow{\lambda} v$ may be infer by the recursion rule $\mathbf{rec}_1$.

□

We are ready to prove that the alternative rules $\mathbf{rec}_1$ and $\mathbf{rec}_2$ are equivalent in the absence of unguarded recursion. Given a fixed set of SOS rules obeying de Simone's requirements, let $\mathbf{SOS}_1$ and $\mathbf{SOS}_2$ be the respective sets of transitions provable from that system from $\mathbf{rec}_1$ resp. $\mathbf{rec}_2$.

Proposition 3 *If u is a well-guarded term then $u \xrightarrow{\lambda} v \in \mathbf{SOS}_1$ if, and only if $u \xrightarrow{\lambda} v \in \mathbf{SOS}_2$ and v is then a well-guarded term.*

Proof :

- *We show that $(u \xrightarrow{\lambda} v \in \mathbf{SOS}_1)$ entails $(u \xrightarrow{\lambda} v \in \mathbf{SOS}_2)$*
  We proceed by induction on the structure of the recursive term u. The only non-trivial case is when $u = \mathbf{rec}\ x.u'$ and the transition in $\mathbf{SOS}_1$ has been inferred from the premise $u'[\mathbf{rec}\ x.u'/x] \xrightarrow{\lambda} v$. The commutation lemma shows the existence of a term v' such that $v = v'[\mathbf{rec}\ x.u'/x]$ and $u' \xrightarrow{\lambda} v' \in \mathbf{SOS}_1$. Then $u' \xrightarrow{\lambda} v' \in \mathbf{SOS}_2$ by induction hypothesis, wherefrom $u = \mathbf{rec}\ x.u' \xrightarrow{\lambda} v'[\mathbf{rec}\ x.u'/x] = v$ may be inferred by the simplified rule for recursion ($\mathbf{rec}_2$).

- *We show that $(u \xrightarrow{\lambda} v \in \mathbf{SOS}_2)$ entails $(u \xrightarrow{\lambda} v \in \mathbf{SOS}_1)$*
  We proceed by induction on the structure of the derivation in $\mathbf{SOS}_2$. The only non-trivial case is when $u = \mathbf{rec}\ x.u'$, thus $v = v'[\mathbf{rec}\ x.u'/x]$ for some v' such that $u' \xrightarrow{\lambda} v' \in \mathbf{SOS}_2$. In that case, $u' \xrightarrow{\lambda} v' \in \mathbf{SOS}_1$ by induction hypothesis, and therefore also $u'[\mathbf{rec}\ x.u'/x] \xrightarrow{\lambda} v'[\mathbf{rec}\ x.u'/x] \in \mathbf{SOS}_1$, wherefrom $u = \mathbf{rec}\ x.u' \xrightarrow{\lambda} v'[\mathbf{rec}\ x.u'/x] = v$ may be inferred by the general rule for recursion ($\mathbf{rec}_1$).

By induction on the deduction we readily verify that if u is a guarded expression, so is v whenever $u \xrightarrow{\lambda} v$ may be infer in either $\mathbf{SOS}_1$ or $\mathbf{SOS}_2$.
□

A property shared by the rule $\mathbf{rec}_2$ and by any rule in de Simone's format is that the terms appearing in the premises are always simpler than the terms appearing in the consequence. That property which doesn't hold for the rule $\mathbf{rec}_1$ allows us to reason upon the deductions by the usual induction on the terms. If we suppose that for each term t, the set of actions labelling transitions from t is a finite subset of Λ, defined effectively from t, then the resulting transition system is finitely branching and furthermore recursive : for any term t, the set $\{<\lambda, t'> \; ; \; t \xrightarrow{\lambda} t'\}$ is finite, and the set $\{<t, \lambda, t'> \; ; \; t \xrightarrow{\lambda} t'\}$ is recursive (as a subset of ω^3 modulo an encoding of Λ and of the set of well-guarded terms -which is readily seen to be recursive-). In fact, for a fixed consequence, there is only a finite number of possible applications of rules deriving that conclusion, and for each application, the left members of the premises are strict subterms of the left member of the consequence. In contrast, when used in conjunction with $\mathbf{rec}_1$, de Simone's rules allow us to realize, up to strong bisimulation, any recursively enumerable system of transitions $T \subset Q \times \Lambda \times Q$ (where Λ and Q are identified with recursive sets of numbers), see [dS84] and [Bou85]). Any recusively enumerable transition system may in fact be realized in Meije which is defined operationally by rules in de Simone's format in conjunction with $\mathbf{rec}_1$ (although the exact syntax for recursive terms is different from the one presented here).

References

[Bou85] Gérard Boudol. Notes on algebraic calculi of processes. *in Logics and models of concurrent systems (K. Apt, Ed) Nato Asi Series F13*, 1985.

[dS84] Robert de Simone. *Calculabilité et expressivité dans l'algèbre de processus MEIJE.* Thèse de 3eme Cycle, Paris VII, 1984.

[Mil80] Robin Milner. *A calculus of communicating systems.* Springer Verlag LNCS, n 92, 1980.

[Plo81] Gordon Plotkin. *A structural approach to operational semantics.* Rep Daimi FN 19, Aarhus Univ., 1981.

LISTE DES DERNIERES PUBLICATIONS INTERNES

PI 528 **CONDITIONAL REWRITE RULES AS AN ALGEBRAIC SEMANTICS OF PROCESSES**
Eric BADOUEL
Mars 1990, 46 Pages.

PI 529 **RESEAUX SYSTOLIQUES SPECIFIQUES A BASE DU PROCESSEUR API15C**
Patrice FRISON, Eric GAUTRIN, Dominique LAVENIER, Jean-Luc SCHARBARG
Mars 1990, 26 Pages.

PI 530 **SEMI-GRANULES AND SCHIELDING FOR OFF-LINE SCHEDULING**
Bernard LE GOFF, Paul LE GUERNIC, Julian ARAOZ DURAND
Avril 1990, 46 Pages.

PI 531 **DATA-FLOW TO VON NEUMANN : THE SIGNAL APPROACH**
Paul LE GUERNIC, Thierry GAUTIER
Avril 1990, 22 Pages.

PI 532 **OPERATIONAL SEMANTICS OF A DISTRIBUTED OBJECT-ORIENTED LANGUAGE AND ITS Z FORMAL SPECIFICATION**
Marc BENVENISTE
Avril 1990, 100 Pages.

PI 533 **ADAPTATION DE LA METHODE DE DAVIDSON A LA RESOLUTION DE SYSTEMES LINEAIRES : IMPLEMENTATION D'UNE VERSION PAR BLOCS SUR UN MULTIPROCESSEUR**
Miloud SADKANE, Brigitte VITAL
Avril 1990, 34 Pages.

PI 534 **DIFFUSE INTERREFLECTIONS. TECHNIQUES FOR FORM-FACTOR COMPUTATION**
Xavier PUEYO
Mai 1990, 28 Pages.

PI 535 **A NOTE ON GUARDED RECURSION**
Eric BADOUEL, Philippe DARONDEAU
Mai 1990, 10 Pages.

PI 536 **TOWARDS DOCUMENT ENGINEERING**
Vincent QUINT, Marc NANARD, Jacques ANDRE
Mai 1990, 20 Pages.

Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique

ISSN 0249-6399