Fundamentals of context-sensitive rewriting ⋆

Salvador Lucas

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
Camino de Vera s/n, E-46071 Valencia, Spain.
e.mail: slucas@dsic.upv.es

Abstract. Recently we have introduced a novel class of rewrite relations we call *context-sensitive rewriting* that proves useful to mechanize a particular class of inference systems. In this paper, we give a more accurate characterization of some properties of context-sensitive rewriting. We also show that several reduction strategies used in λ-calculus namely weak reduction and lazy (*call by name*) reduction, can be viewed as an instance of the context-sensitive rewriting.

Keywords: term rewriting systems, regularity, functional programming.

1 Introduction

The theory of Term Rewriting Systems (TRSs) has proven useful to mechanize the execution of functional programs [3] by imposing directionality on the use of equations in proofs [1, 2]. Directed equations, called 'rewrite rules', are used to replace equals by equals, but only in the indicated direction. In this scheme, computation consists of rewriting terms to a normal form, i.e. an expression that cannot be rewritten any further [1]. The notion of sequence of rewrites traditionally defines an operational view of computation. The equational origin of term rewriting techniques leads to consider an expression $t \to s$ as 't *equals* s' rather than 't *becomes* s' [8]. As a consequence, the property of 'being closed under context application' (a particular form of the algebraic property of regularity) is essential in the definition of the rewriting relation.

In [6] we have introduced a novel class of rewrite relations, we call *context-sensitive* rewriting, whose definition is based on regularity. Roughly speaking, given a term t, the subterm of t at the occurrence u is a redex of the context-sensitive rewrite relation iff the symbols in t which label the occurrences above u satisfy a particular condition we call *replacement condition*.

In this paper, we introduce some properties of the replacement condition, which give a more powerful characterization of the context-sensitive rewriting process. In particular, we show that the evaluation of the replacement condition can be made compositionally. We also show that context-sensitive rewriting satisfies some desirable properties such as stability which allows us to ensure that a (context-sensitive) reducible term also rewrites to another when we instantiate its variable occurrences. This property plays an important role in studying the computational properties of rewriting such as confluence and termination. We also show that several common reduction strategies used in λ-calculus [4, 5], such as weak reduction and lazy reduction, can be viewed as an instance of the context-sensitive rewriting.

The paper is organized as follows. In Section 2, we briefly recall the technical concepts and results used in the remainder of the paper. In Section 3, we shows the basic properties of the context-sensitive rewrite relation. At the end of this section, we illustrate the application of this technique to weak reduction and lazy λ-calculus. We conclude by pointing to some further research.

⋆ This work has been partially supported by CICYT under grant TIC 95-0433-C03-03.

2 Preliminaries

Let us first introduce the main notations used in the paper. For full definitions we refer to [2, 5]. Throughout the paper, V denotes a countable set of variables and Σ denotes a set of function symbols {f, g, ...}, each with a fixed arity given by a function $ar : \Sigma \rightarrow \mathcal{N}$. By $\mathcal{T}(\Sigma, V)$ we denote the set of terms. Given a term t, $Var(t)$ is the set of variable symbols in t.

Terms are viewed as labelled trees in the usual way. Occurrences $u, v, \ldots$ are represented by chains of positive natural numbers used to address subterms of t, and they are ordered by the prefix ordering: $u \leq v$ if there exists a w such that $u.w = v$. By $u \parallel v$ we denote that the occurrences u, v are disjoint, i.e. $u \not\leq v$ and $v \not\leq u$. $O(t)$ denotes the set of occurrences of a term t. $t|_u$ is the subterm at occurrence u of t. $t[s]_u$ is the term t with the subterm at the occurrence u replaced with s. We refer to any term C that is the same as t everywhere except below u, i.e. there exists a term s such that $C[s]_u = t$, as the *context* within the replacement occurs.

A rewrite rule is an ordered pair (l, r), written $l \rightarrow r$, with $l, r \in \mathcal{T}(\Sigma, V)$, $l \notin V$ and $Var(r) \subseteq Var(l)$. l is said the left-hand side (*lhs*) of the rule and r is the right-hand side (*rhs*). A TRS is a pair $\mathcal{R} = (\Sigma, R)$ where R is a set of rewrite rules. For a given TRS $\mathcal{R} = (\Sigma, R)$, a term t rewrites to a term s (at the occurrence u), written $t \rightarrow_{\mathcal{R}} s$, if $t|_u = \sigma(l)$ and $s = t[\sigma(r)]_u$, for some rule $l \rightarrow r$ in R, occurrence u in t and substitution σ. $\rightarrow_{\mathcal{R}}$ is the one-step rewrite relation for $\mathcal{R}$.

$\mathcal{N}_k^+$ is an initial segment $\{1, 2, \ldots k\}$ of the set of positive natural numbers $\mathcal{N}^+$ where $\mathcal{N}_0^+ = \emptyset$.

A homogeneous algebra $\mathcal{A}$ is a pair (A, F) where A is a set (the carrier) and F is a set of mappings $\{f, g, \ldots\}$. The mapping $ar : F \rightarrow \mathcal{N}$ associates with each function f in F its arity, i.e. $ar(f) = k$ iff f is a mapping from $A^k \rightarrow A$.

3 Context-sensitive rewriting

3.1 Regularity

First we summarize the main algebraic concepts that support the context-sensitive rewriting. More details and missing proofs can be found in [6, 7].

Definition 3.1 (Regularity) *Let* R *be a relation on* A *and* $f : A^k \rightarrow A$ *be a mapping. We say that* R *is* regular *wrt* f *with index* j, $1 \leq j \leq k$ *(we also say that* R *is* j-regular *wrt* f*) iff for all* $a, b, a_1, \ldots, a_k \in A$. $a \mathrel{\mathrm{R}} b \Rightarrow f(a_1, \ldots, a_{j-1}, a, \ldots, a_k) \mathrel{\mathrm{R}} f(a_1, \ldots, a_{j-1}, b, \ldots, a_k)$. *The set* $m_{\mathrm{R}}(f) \subseteq \mathcal{N}_k^+$ *of all indexes* j *such that* R *is* j*-regular wrt* f *is called the* regularity map *of* R *wrt* f.

The concept of regularity can be isolated of considering any particular relation by just taking a set of indices for each function in the algebra. This leads to the relation-independent concept of *algebraic regularity map*.

Definition 3.2 (Algebraic regularity map) *Let* $\mathcal{A} = (A, F)$ *be an algebra. A mapping* $m : F \rightarrow 2^{\mathcal{N}}$ *is an* algebraic regularity map *(r-map) for* $\mathcal{A}$ *iff for all* $f \in F$. $m(f) \subseteq \mathcal{N}_{ar(f)}^+$. *The set of all r-maps of* $\mathcal{A}$ *is denoted by* $M_{\mathcal{A}}$.

We express in a compact mode the regularity property of a relation R wrt each function in a given algebra $\mathcal{A} = (A, F)$, according to definition 3.1. The map $m_R : F \rightarrow 2^{\mathcal{N}}$ is called the *regularity map* (or simply the r-map) of R wrt $\mathcal{A}$ iff for all $f \in F$. $m_R(f)$ is the regularity map of R wrt f. We also say that R is m_R-regular wrt $\mathcal{A}$.

The following example shows the regularity maps of a given relation with respect to several algebras. Let us consider the relation R of 'being multiple of' in the set $\mathcal{Q}^+$ of positive rational numbers, i.e. $\forall\, a, b \in \mathcal{Q}^+.\ a \mathrel{\mathrm{R}} b \Leftrightarrow \exists c \in \mathcal{N}.\ a = b \times c$. Then the r-map of R wrt $(\mathcal{Q}^+, \{+\})$ verifies $m_R(+) = \emptyset$, the r-map of R wrt $(\mathcal{Q}^+, \{/\})$ verifies $m_R(/) = \{1\}$ and the r-map of R wrt $(\mathcal{Q}^+, \{\times\})$ verifies $m_R(\times) = \{1, 2\}$.

Now we introduce a partial order on r-maps.

Definition 3.3 (Partial order on regularity maps) *Let $\mathcal{A} = (A, F)$ be an algebra. Let $M_{\mathcal{A}}$ be the set of r-maps for $\mathcal{A}$ and let $m, m' \in M_{\mathcal{A}}$. We define a partial order $\sqsubseteq$ on $M_{\mathcal{A}}$ as follows: $m \sqsubseteq m' \Leftrightarrow \forall f \in F.\ m(f) \subseteq m'(f)$.*

It is immediate to see that $(M_{\mathcal{A}}, \sqsubseteq)$ is a complete lattice. The map $m_{\top}$ which assigns to each function $f \in F$ the whole segment $\mathcal{N}^{+}_{ar(f)}$ is the top element and the map $m_{\perp}$ which assigns to each function $f \in F$ the empty set $\emptyset$ is the bottom element.

3.2 Basic properties of context-sensitive rewriting

In this section, we show some powerful properties of context-sensitive rewriting. The idea behind the concept of context-sensitive rewriting is to impose a syntactic replacement condition which prevents one from performing some reduction steps. This is achieved by using a map on the signature we call replacement map. A mapping $\mu : \Sigma \to 2^{\mathcal{N}}$ is a *replacement map* (or Σ-map) for the signature Σ iff for all $\mathrm{f} \in \Sigma$. $\mu(\mathrm{f}) \subseteq \mathcal{N}^{+}_{ar(\mathrm{f})}$.

We use the replacement map to give a condition on the occurrences of a given term where the rewrite rules can be applied.

Definition 3.4 (Replacement condition) *Let Σ be a signature and μ be a Σ-map. Let $t \in \mathcal{T}(\Sigma, V)$ be a term. The replacement condition is a relation $\gamma_{\mu,t}$ defined on the set of occurrences $O(t)$ as follows:*

$\gamma_{\mu,t}(\varepsilon)$.

$\gamma_{\mu,\mathrm{f}(t_1,\ldots,t_i,\ldots,t_k)}(i.u) \Leftrightarrow (i \in \mu(\mathrm{f})) \wedge \gamma_{\mu,t_i}(u)$.

We say that the occurrence u of a term t satisfies the replacement condition $\gamma_{\mu,t}$ iff $\gamma_{\mu,t}(u)$. We also say that u is a replacing occurrence of t or that the subterm $t|_u$ is a replacing subterm.

In the following, we write $\gamma_t(u)$ when the replacement map is clear from the context. Also, let us denote as $O^{\mu}(t)$ the set of *replacing* occurrences of a term t: $u \in O^{\mu}(t) \Leftrightarrow u \in O(t) \wedge \gamma_t(u)$.

Roughly speaking, the replacement condition for a term t wrt μ imposes a restriction based on the function symbols which label the path between the root of t and the subterm at occurrence u. To make clear the meaning of this restriction, we present some basic properties of the replacement condition we defined above.

Proposition 3.5 (Compositionality in evaluating the replacement condition) *Let Σ be a signature, $t \in \mathcal{T}(\Sigma, V)$ and $u \in O(t)$. Then the following statement holds: $u = u_1.u_2 \Rightarrow (\gamma_t(u) \Leftrightarrow (\gamma_t(u_1) \wedge \gamma_{t|_{u_1}}(u_2)))$.*

This Proposition allows us to realize the evaluation of replacement conditions in a compositionally mode. As an immediate consequence of this result, we have the following corollaries.

Corollary 3.6 *Let Σ be a signature, $t \in \mathcal{T}(\Sigma, V)$ and $u, v \in O(t)$. The following statements hold:*

1. $(u \leq v \wedge \gamma_t(v)) \Rightarrow \gamma_t(u)$,
2. $(u \leq v \wedge \neg\gamma_t(u)) \Rightarrow \neg\gamma_t(v)$.

Corollary 3.7 *Let Σ be a signature, $t \in \mathcal{T}(\Sigma, V)$ and $u \in O(t)$. The following statement holds: $\gamma_t(u) \Rightarrow (u = u_1.u_2 \Rightarrow \gamma_{t|_{u_1}}(u_2))$*

We can demonstrate some useful properties related with the persistence of the evaluation of the replacement condition with respect to substitution application and subterm replacement.

Proposition 3.8 (Persistence with respect to substitution application) *Let Σ be a signature, $t \in \mathcal{T}(\Sigma, V)$ and $u \in O(t)$. Let $\sigma : V \to \mathcal{T}(\Sigma, V)$ be a substitution. Then, $\gamma_t(u) \Leftrightarrow \gamma_{\sigma(t)}(u)$.*

Proposition 3.9 (Persistence with respect to subterm replacements) *Let Σ be a signature, $t, t' \in \mathcal{T}(\Sigma, V)$ and $u, v \in O(t)$. Then,*

1. $u \leq v \Rightarrow (\gamma_t(u) \Leftrightarrow \gamma_{t[t']_v}(u))$ *and*
2. $u \parallel v \Rightarrow (\gamma_t(u) \Leftrightarrow \gamma_{t[t']_v}(u))$.

The above results allow us to devise more efficient methods for the evaluation of the replacement condition in context-sensitive systems.

The order $\sqsubseteq$ on algebraic regularity maps can easily be translated to compare *replacement* maps for a given signature Σ. By abuse of notation, we will use the same symbol $\sqsubseteq$ to denote both orderings. The next result establishes that a replacement condition that holds for a given replacement map is also verified wrt a greater replacement map.

Proposition 3.10 *Let Σ be a signature, $t \in \mathcal{T}(\Sigma, V)$ and $u \in O(t)$. Let μ, μ' be two Σ-maps such that $\mu \sqsubseteq \mu'$. Then we have $\gamma_{\mu,t}(u) \Rightarrow \gamma_{\mu',t}(u)$*

Now we introduce the context-sensitive rewriting by means of their one-step reduction relation.

Definition 3.11 (One-step context-sensitive rewrite relation) *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and μ be a Σ-map. A term t μ-rewrites to a term s, written $t \hookrightarrow_{\mathcal{R}(\mu)} s$, if $t \rightarrow_{\mathcal{R}} s$ reducing the subterm of t at the* replacing *occurrence $u \in O^{\mu}(t)$. $\hookrightarrow_{\mathcal{R}(\mu)}$ is the one-step context-sensitive rewrite relation of $\mathcal{R}$ wrt μ. $\hookrightarrow^*_{\mathcal{R}(\mu)}$ is the context-sensitive rewrite relation of $\mathcal{R}$ wrt μ.*

In the following, when no necessary, we assume that the TRS $\mathcal{R} = (\Sigma, R)$ and Σ-map μ are given, and we drop references to them by solely writing $\hookrightarrow$ instead of $\hookrightarrow_{\mathcal{R}(\mu)}$ to denote the one-step context-sensitive rewrite relation.

Let us now consider a first example of context-sensitive rewriting. Let $\alpha : \mathrm{f}(x) \rightarrow x$ be a rule and $t = \mathrm{g}(\mathrm{f}(a), \mathrm{f}(b))$ be a term. a) If $\mu(\mathrm{g}) = \emptyset$ then t cannot be reduced, because neither the occurrence $u = 1$ nor $u = 2$ satisfies the replacement condition wrt μ. b) If $\mu(\mathrm{g}) = \{1\}$, then we can reduce $\mathrm{f}(a)$ but we cannot reduce $\mathrm{f}(b)$. c) If $\mu(\mathrm{g}) = \{2\}$, then we can reduce $\mathrm{f}(b)$ but we cannot reduce $\mathrm{f}(a)$. d) If $\mu(\mathrm{g}) = \{1, 2\}$, then we can be reduce both $\mathrm{f}(a)$ and $\mathrm{f}(b)$.

The following proposition illustrates the (one-step) context-sensitive version of the 'being closed under context application' property. The context-sensitive rewriting is 'closed under *replacing* context application'.

Proposition 3.12 (Restricted context replacement) *Let $C, t, s \in \mathcal{T}(\Sigma, V)$. Let $u \in O^{\mu}(C)$. If $t \hookrightarrow s$ then $C[t]_u \hookrightarrow C[s]_u$.*

Note that for the *top* Σ-map $\mu_{\top}$ the replacement condition is satisfied for every term t and occurrence u. This is to say that, given a TRS $\mathcal{R}$, the one-step context-sensitive rewrite relation for $\mu_{\top}$ and the standard one-step rewrite relation coincide, i.e. $\hookrightarrow_{\mathcal{R}(\mu_{\top})} = \rightarrow_{\mathcal{R}}$. Now we show that the context-sensitive rewriting is stable under substitution.

Proposition 3.13 (Stability of $\hookrightarrow$) *Let $\mathcal{R} = (\Sigma, R)$ be a TRS, μ be a Σ-map and $\sigma : V \rightarrow \mathcal{T}(\Sigma, V)$ be a substitution. Then $t \hookrightarrow s \Rightarrow \sigma(t) \hookrightarrow \sigma(s)$.*

The following proposition establishes that, given a TRS, the one-step context-sensitive rewrite relation induced by a replacement map includes the relation induced by a lesser replacement map.

Proposition 3.14 (Monotonicity of $\hookrightarrow$ with respect to $\sqsubseteq$) *Let $\mathcal{R} = (\Sigma, R)$ be a TRS and μ, μ' be Σ-maps. Then $\mu \sqsubseteq \mu' \Rightarrow \hookrightarrow_{\mathcal{R}(\mu)} \subseteq \hookrightarrow_{\mathcal{R}(\mu')}$.*

3.3 Context-sensitive rewriting and Inference Systems

In [7] we have proven that a TRS $\mathcal{R} = (\Sigma, R)$ and a Σ-map μ generate an inference system (IS) which transition relation and the context-sensitive rewrite relation $\hookrightarrow_{\mathcal{R}(\mu)}$ coincide. Roughly speaking, the axioms of the IS are instances of the rewrite rules and a *contextual* inference rule is defined for each k-ary symbol $\mathrm{f} \in \Sigma$ and each argument $i \in \mu(\mathrm{f})$. The *basic inference system* associated to $\mathcal{R}$ and μ, denoted by $I(\mathcal{R}, \mu)$, is defined as follows:

1. $\alpha : l \to r \in R$ iff
   $(\mathsf{a}_\alpha)\dfrac{}{l \diamond r}$ is an (scheme of) axiom of $I(\mathcal{R}, \mu)$
2. For each $\mathrm{f} \in \Sigma$, $k = ar(\mathrm{f})$, $i \in \mu(\mathrm{f})$ and terms $t_1, \ldots, t_k, s_i$
   $(\mathsf{r}[\mathrm{f}, i])\dfrac{t_i \diamond s_i}{\mathrm{f}(t_1, \ldots, t_i, \ldots, t_k) \diamond \mathrm{f}(t_1, \ldots, s_i, \ldots, t_k)}$ is a rule of $I(\mathcal{R}, \mu)$

The relation defined by $I(\mathcal{R}, \mu)$ on $\mathcal{T}(\Sigma, V)$ (that is, its *deductive closure*) is denoted by $\to_{I(\mathcal{R},\mu)}$.

This is the class of ISs that context-sensitive rewriting mechanizes, i.e. those ISs such that $\to_{I(\mathcal{R},\mu)} = \hookrightarrow_{\mathcal{R}(\mu)}$. Conversely, given an IS such that every rule fits the pattern $\mathsf{r}[\mathrm{f}, i]$, we define a TRS by taking the axioms of the IS as rewrite rules and we attach a replacement map according to the existing contextual inference rules $\mathsf{r}[\mathrm{f}, i]$.

Now, if we want to define an abstract machine that computes the transition relation of such an IS we can modify a standard rewriting machine (such as the graph reduction machine [10]) to deal with the restrictions imposed by the replacement condition. Roughly speaking, in order to execute a program with respect to a set of rewrite rules, the *evaluator* of a rewriting machine must follows the following algorithm (until a normal form is reached) [10]:

1. selecting the next redex to be reduced and
2. reducing it.

To accomodate this to a context-sensitive rewriting system we must to take into account the replacement condition in the first step.

3.4 Examples and applications

Let us consider the λ-calculus. A λ-term $M ::= x \mid \lambda x.M \mid MN$ is represented as a tree by considering each λx as an unary symbol, application as a binary symbol @ and each variable as a symbol of arity zero [5]. This is expressed by means of the signature $\Sigma_\lambda = C_V \cup \Lambda_x \cup \{@\}$, where $\Lambda_x = \{\lambda_x \mid x \in V\}$ and C_V is a set of arbitrary constants (one for each variable symbol). The immediate β-reduction of λ-calculus $\to_\beta$, can be obtained from the following inference system [4, 5], which we refer to as IS_β (in spite of the introduction of Σ_λ, we express IS_β in the standard λ-notation):

$$(red)\ \frac{}{(\lambda x.P)Q \rhd P[x := Q]}$$

We can see that any instance of the axiom scheme (red) can be taken as a rewrite rule $(\lambda x.P)Q \to P[x := Q]$. In the sequel, R_λ is the set of rules which are instances of the (red) axiom scheme.

$$(r\lambda)\ \frac{M \rhd M'}{\lambda x.M \rhd \lambda x.M'} \qquad (r1)\ \frac{M \rhd M'}{MN \rhd M'N} \qquad (r2)\ \frac{M \rhd M'}{NM \rhd NM'}$$

Note that the *context passing* rules $(r\lambda)$, $(r1)$ and $(r2)$ fit the pattern $\mathsf{r}[\mathrm{f,i}]$.

Example 3.15 (Weak β-reduction) *Let us consider the weak β-reduction, a restriction on the β-reduction of λ-calculus. For this reduction, an abstraction $\lambda x.M$ is irreducible, even if M contains a redex. This corresponds to drop the rule $(r\lambda)$ in IS_β, thus obtaining IS_w whose only rules are $r1$ and $r2$. We can simulate this behavior with context-sensitive rewriting by considering the following TRS $\mathcal{R} = (\Sigma_\lambda, R_\lambda)$ and the Σ_λ-map μ_w defined as follows: $\mu_w(@) = \{1, 2\}$, $\mu_w(f) = \varnothing$ for all $f \in \Sigma_\lambda - \{@\}$.*

Example 3.16 (Lazy λ-calculus) *Now consider the lazy λ-calculus, another restriction on the β-reduction used by the majority of functional languages to avoid the 'eagerness' in evaluating functional applications. To delay the evaluation of an application MN we need only one context passing rule, namely (r1) to constitute IS_z, the inference system for lazy λ-calculus. We can simulate this behavior with context-sensitive rewriting by considering the following TRS $\mathcal{R} = (\Sigma_\lambda, R_\lambda)$ and the Σ_λ-map μ_z defined as follows: $\mu_z(@) = \{1\}$, $\mu_z(f) = \emptyset$ for all $f \in \Sigma_\lambda - \{@\}$.*

The R. Milner's π-calculus [9] is a computational scheme which models concurrency. In [7] we define a TRS and a replacement map whose context-sensitive rewriting can be seen as a mechanization of the π-calculus process execution.

4 Further research

We have introduced a powerful characterization of the context-sensitive rewriting process. We have examined several computational properties of context-sensitive relations and demonstrated that they are stable under substitution and closed under replacing context application.

A first approach to the characterization of confluence of context-sensitive rewriting can be found in [6]. A desirable direction for further research will be the definition of syntactic criteria to ensure the confluence and termination of induced context-sensitive rewriting when we attach a replacement map to a given term rewriting system.

5 Acknowledgements

I would like to thank María Alpuente and Germán Vidal for their encouragement and support. Also, I am grateful to Asunción Casanova, Javier Oliver and Javier Piris for their helpful comments and corrections.

References

1. N. Dershowitz. A Taste of Rewrite Systems. In P.E. Lauer, editor, *Proc of Functional Programming, Concurrency, Simulation and Automated Reasoning*, volume 693 of *Lecture Notes in Computer Science*, pages 199-228, Berlin, 1993.
2. N. Dershowitz and J.P. Jouannaud. Rewrite Systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 243-320. Elsevier, Amsterdam and The MIT Press, Cambridge, MA, 1990.
3. H. Ehrig and B. Mahr. Fundamentals of Algebraic Specification. volume 6 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, Berlin, 1985.
4. J.R. Hindley and J.P. Seldin. Introduction to Combinators and λ-Calculus. Cambridge University Press, 1986.
5. R. Lalement. Computation as Logic. Masson-Prentice Hall International, 1993.
6. S. Lucas. Computational properties in context-sensitive rewriting. In *Proc. of 1995 Joint Conference on Declarative Programming, GULP-PRODE'95*. To appear.
7. S. Lucas and J. Oliver. Context-sensitive rewriting. Technical report DSIC-II/23/94.
8. J. Meseguer. Conditional Rewriting Logic as an Unified Model of Concurrency. *Journal of Theoretical Computer Science*, 96:73-155, 1992.
9. R. Milner. The polyadic π-calculus: A tutorial. In F.L. Brauer, W Bauer and H Schwintenberg, editors, *Logic and Algebra of Specifications*, Springer-Verlag, Berlin, 1993.
10. S. L. Peyton-Jones. The Implementation of Functional Programming Languages. Prentice-Hall, 1987.
11. D. Plaisted. Equational Reasoning and Term Rewriting Systems. In D. Gabbay and J. Siekmann, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, volume 1, 1994.

This article was processed using the LaTeX macro package with LLNCS style