Journal of Symbolic Computation 84 (2018) 57–83

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Reduction operators and completion of rewriting systems

Cyrille Chenavier

INRIA, équipe πr^2, Laboratoire IRIF, CNRS UMR 8243, Université Paris-Diderot, Case 7014, 75205 Paris Cedex 13, France

ARTICLE INFO

Article history:
Received 3 June 2016
Accepted 12 February 2017
Available online 23 March 2017

Keywords:
Reduction operators
Completion
Noncommutative Groebner bases

ABSTRACT

We propose a functional description of rewriting systems where reduction rules are represented by linear maps called reduction operators. We show that reduction operators admit a lattice structure. Using this structure we define the notions of confluence and of Church–Rosser property. We show that these notions are equivalent. We give an algebraic formulation of completion and show that such a completion exists using the lattice structure. We interpret the confluence for reduction operators in terms of Gröbner bases. Finally, we introduce generalised reduction operators relative to non totally ordered sets.

1. Introduction

Convergent rewriting systems are confluent and terminating rewriting systems. They appear in rewriting theory to solve decision problems such as the word problem or the ideal membership problem. Completion algorithms were introduced to compute convergent rewriting systems: the Knuth–Bendix completion algorithm (Knuth and Bendix, 1970) for term rewriting (Baader and Nipkow, 1998) and string rewriting (Book and Otto, 1993) or the Buchberger algorithm for Gröbner bases (Buchberger, 1965; Shirshov, 2009; Bokut', 1976; Bergman, 1978) of commutative algebras (Buchberger 1965, 1987) or associative algebras (Teo, 1994). In this paper, we propose an algebraic approach to completion: we formulate it algebraically and show that it can be obtained with an algebraic construction.

E-mail address: chenavier@pps.univ-paris-diderot.fr.

http://dx.doi.org/10.1016/j.jsc.2017.03.002

We use the functional point of view considered by Berger (1998a) for rewriting on non-commutative polynomials. The latter are linear combinations of words. In this introduction, we first explain how the functional approach to string rewriting systems works. In the second part, we introduce *reduction operators* and formulate the confluence and the completion with those. We also make explicit the link between reduction operators and rewriting on non-commutative polynomials, which gives us our algebraic constructions.

1.1. A functional approach to string rewriting and Gröbner bases

Confluence for string rewriting systems For string rewriting systems, the method consists in considering an idempotent application modelling the rewrite rules. This method works for *semi-reduced* string rewriting systems, that is the systems such that

1. the left-hand sides of its rewrite rules are pairwise distinct,
2. no right-hand side of its rules is the left-hand side of another one.

For instance, the string rewriting system with alphabet $\{x, y\}$ and with one rewrite rule $yy \longrightarrow yx$ is semi-reduced.

Given a string rewriting system $\langle X \mid R\rangle$ with alphabet X and set of rewrite rules R, we denote by X^* the set of words over X. Our algebraic constructions require that $\langle X \mid R\rangle$ is equipped with a total termination order $<$, that is, a terminating order on words such that every left-hand side of a rewrite rule is greater than the corresponding right-hand side. In Theorem 2.1.13, we show, using this order, that $\langle X \mid R\rangle$ can be transformed into a unique semi-reduced string rewriting system, so that we may assume that it has this property. The application modelling its rules is the map $S : X^* \longrightarrow X^*$ defined by

1. $S(l(\alpha)) = r(\alpha)$ for every $\alpha \in R$ with left-hand side $l(\alpha)$ and right-hand side $r(\alpha)$,
2. $S(w) = w$ if no element of R has left-hand side w.

The application associated to our example maps yy to yx and fixes all other words.

The order $<$ guarantees that $\langle X \mid R\rangle$ terminates. Thus, it is sufficient to study whether it is confluent or not to know if it is convergent. In order to obtain the functional formulation of confluence, we consider the *extensions of* S, that is, the applications $S_{p,q}$ defined for every pair of integers (p, q) by

1. $S_{p,q}(w) = w_1 r(\alpha) w_2$, if there exist words w_1, w_2 of length p and q, respectively, and $\alpha \in R$, such that w is equal to $w_1 l(\alpha) w_2$,
2. $S_{p,q}(w) = w$, otherwise.

In the previous example, $S_{0,1}$ maps yyx to yxx and yyy to yxy, and $S_{1,0}$ maps xyy to xyx and yyy to yyx. These applications enable us to characterise the normal forms for $\langle X \mid R\rangle$: a normal form is a word whose every sub-word is fixed by S, that is, the normal forms are the words fixed by all the extensions of S.

Given a word w, we denote by $[w]$ the class of w for the equivalence relation induced by R. The order $<$ being total and well-founded, $[w]$ admits a smallest element. Let M be the application from X^* to itself mapping a word to this minimum. A word w fixed by all the extensions of S but which is not fixed by M is called an *obstruction* of $\langle X \mid R\rangle$. In other words, an obstruction is a normal form which is not minimal in its equivalence class. Hence, the set of obstructions is empty if and only if each equivalence class contains exactly one normal form. Moreover, recall that a terminating rewriting system is confluent if and only if every element admit exactly one normal form (see for instance Baader and Nipkow, 1998, Section 2.1). Thus, we obtain the following functional characterisation of confluence: $\langle X \mid R\rangle$ is confluent if and only if the set of obstructions is empty. Considering our example, we deduce from the diagram