Completion of operadic rewriting systems by Gaussian elimination

Benjamin Dupont, Philippe Malbos, Isaac Ren

Completion of operadic rewriting systems by Gaussian elimination

Benjamin Dupont[1], Philippe Malbos[2], and Isaac Ren[3]

[1,2] Université de Lyon, France, {bdupont,malbos}@math.univ-lyon1.fr
[3] École Normale Supérieure de Lyon, France, isaac.ren@ens-lyon.fr

Abstract

We study the confluence properties of non-symmetric operadic rewriting systems using linear algebra methods. We extend the completion procedure F4, known for commutative and non-commutative algebras, to operads. This procedure allows us to parallelize completion by applying a Gaussian elimination process in order to treat multiple critical branchings simultaneously. We discuss heuristics and strategies to optimize this procedure in the operadic context: first to reduce the set of critical branchings to be examined and then to parallelize the elimination.

1 Introduction

Algebraic rewriting theory aims at studying rewriting relations in algebraic and categorical structures such as monoids, categories, equational theories, linear algebras, operads and higher-dimensional algebras, and categories. Proofs of confluence of an algebraic rewriting system (AlgRS) are mainly based on the critical branching lemma (CBL) that proves local confluence from confluence of a set of *critical branchings*, which correspond to confluence obstructions induced by minimal overlappings of rules. The CBL approach is used various contexts, including automated theorem proofs, word problems in universal algebras, and polynomial ideal membership. CBL's were proved for numerous AlgRS's: rewriting on strings [24], terms [18], and higher-dimensional categories [14, 15]. CBL's also have various formulations in linear structures, for commutative algebras [4], associative algebras [1, 2, 23], non-symmetric and shuffle operads [7, 20], and higher-dimensional linear categories [10]. In algebraic rewriting, the CBL constitutes the first step in the construction of cofibrant replacements of algebraic and categorical structures [13, 15, 19].

The principle of the critical branching completion procedure (CBCP) on an AlgRS can be formulated as follows:

Input: A set R of rules of an algebraic rewriting system.

$\mathsf{R}' := \mathsf{R}$;
$\mathcal{C} :=$ critical branchings of R';
while $\mathcal{C} \neq \varnothing$ **do**
  Select a subset B of branchings in $\mathcal{C}$, and remove them from $\mathcal{C}$;
  Add rewriting rules to R' to make the non-confluent branchings of B confluent;
  Update $\mathcal{C}$ with branchings induced by the new rules;
return R';

If the additional rewriting rules are oriented with respect to a termination order, such as a monomial order, the procedure returns a terminating rewriting system. If moreover the procedure terminates, then the result is a convergent rewriting system. Otherwise, it builds an increasing sequence of rewriting systems, whose limit is convergent. The resulting rewriting system is finite if and only if the input is finite and the procedure terminates.

Concrete implementations of CBCP are based on the preparation of the AlgRS (autoreductions and adding new generators), the type of critical branchings considered, a filtration on the critical branchings, and the parallelization of the computation of their confluence at each step. The choice of branchings to consider can depend on a study of overlapping patterns (Buchberger's criterion [5], Triangle Lemma [3]), or relations between critical branchings and Gebauer-Möller criteria [12, 20, 22]. The filtration on critical branchings gives the order in which to examine the critical branchings and depends on the shape of the rules (reduced, homogeneous...). Finally, several methods can be used to compute confluence in parallel wrt the filtration. One approach is based on Gaussian elimination, mainly developed for the computation of commutative [11] and non-commutative Gröbner bases [6, 25], see also [16]. This principle also appears in [3] for the study of non-symmetric operads.

However, these optimizations were not fully developed in the case of operadic rewriting. Indeed, this algebraic paradigm is complex due to the linear context, the problem of managing symmetric actions, and the complexity of operadic patterns. Rewriting systems for non-symmetric operads were studied in [3, 8, 20], and the question of management of the action of symmetries on terms was addressed in [7], which introduces a notions of Gröbner bases for shuffle operads, and implemented in [9].

In this work, we study the optimization of completion procedures for operadic rewriting systems (ORS). We define a completion algorithm for ORS resolving non-confluence by Gaussian elimination with respect to a chosen confluence obstruction strategy. This work is part of a general program that aims to define computational tools for mathematicians studying higher algebras and higher categories. Indeed, novel higher structures appear in numerous fields such as geometry, physical mathematics, representation theory and quantum topology. Higher structures are generally defined by complex presentations by generators and relations, so there is real need for efficient completion procedures in algebraic contexts.

This abstract is organised as follows. In Section 2, we recall the notion of rewriting systems for non-symmetric operads and we explain some strategies for the implementation of CBCP. Section 3 presents the completion algorithm for ORS's by Gaussian elimination.

2 Confluence of operadic rewriting systems

In this section we recall the notion of operadic rewriting systems on a ground field $\mathbb{K}$ of zero characteristic and the different approaches to obtaining a CBL for these systems.

2.1. Operadic rewriting systems. A *collection* is a sequence $(V(n))_{n\in\mathbb{N}}$ of vector spaces indexed by *arities* $n \geqslant 0$. A *(non-symmetric) operad* is a collection P with an *identity* element $\varepsilon \in P(1)$, and equipped with composition maps $\circ : P(k)\otimes P(n_1)\otimes\ldots\otimes P(n_k) \to P(n_1+\ldots+n_k)$ satisfying identity and associativity conditions. The set of *monomials* $\mathcal{T}(\Sigma)$ is the term algebra on a graded set $\Sigma = (\Sigma(n))_{n>0}$. As for the free algebra generated by a family of indeterminates, we define the free operad $\mathcal{F}(\Sigma)$ on Σ, where, for $n > 0$, $\mathcal{F}(\Sigma)(n)$ is the vector space spanned by monomials of arity n, called *(homogeneous) polynomials*. The *support* of $f = \sum_{i\in I} \lambda_i u_i$ is the set of monomials $\mathrm{Supp}(f) := \{u_i \mid i \in I\}$ that appear in its decomposition. A *context of* $\mathcal{F}(\Sigma)$ *of inner arity* k is a term C of $\mathcal{T}(\Sigma \cup \{\Box_k\})$, where $\Box_k$ is a symbol of arity k that appears exactly once in C. For a monomial u of arity k, we denote by $C[u]$ the monomial C where we replace $\Box_k$ by u; we extend this notation to polynomials by linearity.

An *operadic rewriting system* (ORS) is the data $X = (\Sigma, R)$ made of a graded set Σ and a relation $R \subset \mathcal{T}(\Sigma) \times \mathcal{F}(\Sigma)$, whose elements are *rewriting rules* $\alpha : s(\alpha) \to t(\alpha)$. We define the graph $\mathcal{R}_X$, whose vertices are the elements of $\mathcal{F}(\Sigma)$ and whose edges are the $\lambda C[\alpha] + 1_b$:

$\lambda C[s(\alpha)] + b \to_R \lambda C[t(\alpha)] + b$, where $\alpha \in R$, C is a context, $\lambda \in \mathbb{K} \setminus \{0\}$, and b is a polynomial of $\mathcal{F}(\Sigma)$. An edge of $\mathcal{R}_X$ is a *rewriting monomial* when $\lambda = 1$ and $b = 0$, and a *rewriting step* when $C[s(\alpha)] \notin \mathrm{Supp}(b)$. Denote by $\mathcal{R}_X^m$ the set of rewriting monomials of X and by $\cdot$ the composition of paths in $\mathcal{R}_X$. The paths in $\mathcal{R}_X$ made of rewriting steps are called *rewriting paths of* X. A polynomial a in $\mathcal{F}(\Sigma)$ is in *normal form wrt* X if there is no rewriting step with source a. A *reduction strategy* is a map σ, which to any monomial u associates an identity if u is reduced, and a rewriting monomial $\sigma(u)$ of source u otherwise. The ORS X is *terminating* if there does not exist an infinite rewriting path.

A *monomial order* on $\mathcal{T}(\Sigma)$ is a total order $\prec$ stable by product, that is, for all $u, u' \in \mathcal{T}(\Sigma)(k)$, $v, v' \in \mathcal{T}(\Sigma)(\ell)$, and $1 \leqslant i \leqslant k$, $(u \prec u', v \prec v')$ implies $u \circ_i v \prec u' \circ_i v'$. An ORS X is *compatible with* $\prec$ if, for every rewriting rule $\alpha \in R$ and every monomial $v \in \mathrm{Supp}(t(\alpha))$, $v \prec s(\alpha)$. Note that if $\prec$ is well-founded, then X is terminating.

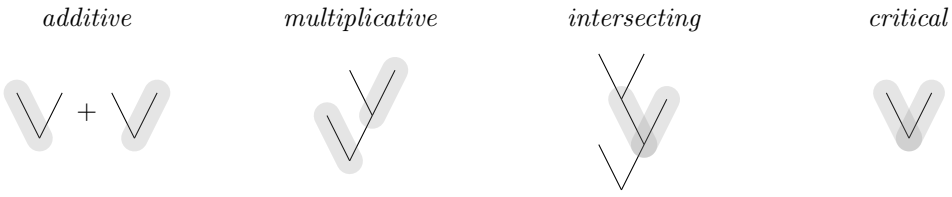
A *branching* (resp. *local branching*) is a pair (f, g) of rewriting paths (resp. rewriting steps) such that $f \neq g$ and $s(f) = s(g)$. The local branchings of X are classified as follows:

i) *additive branchings*: $(\lambda f + \mu 1_v + 1_c, \lambda 1_u + \mu g + 1_c)$, where $f : u \to a, g : v \to b \in \mathcal{R}_X^m$, $\lambda, \mu \in \mathbb{K} \setminus \{0\}$, c is a 0-cell, $u \neq v$, and $u, v \notin \mathrm{Supp}(c)$.

ii) *multiplicative branchings*: $(\lambda C[f, 1_v] + 1_c, \lambda C[1_u, g] + 1_c)$, where C is a two-hole context, $f : u \to a, g : v \to b \in \mathcal{R}_X^m$, $\lambda \in \mathbb{K} \setminus \{0\}$, c is a 0-cell, and $C[u, v] \notin \mathrm{Supp}(c)$.

iii) *intersecting branchings*: the rest of the local branchings. A *critical branching* is an intersecting branching that is minimal for the order induced by $(f, g) \subseteq (C[f] + 1_c, C[g] + 1_c)$ for a context C and a polynomial c of $\mathcal{F}(\Sigma)$.

In a schematic way, we can illustrate local branchings for an ORS as follows, where the highlighted parts of tree monomials indicate the sources of the rewriting rules:



A branching (f, g) is *confluent* if there exist rewriting paths h and k such that $t(f \cdot h) = t(g \cdot k)$. Given a set B of branchings of X, X is B-*confluent* if every $b \in B$ is confluent. If B is the set of all branchings, then we say that X is confluent. We say that X is *convergent* if it is terminating and confluent.

2.2. Strategies for completion procedures. A completion procedure wrt a given monomial order $\prec$ transforms an ORS into a convergent one by adding rules, oriented wrt the order $\prec$, to amend non-confluent branchings. Such a procedure is based on a map that selects a type of branching whose confluence implies the confluence of all branchings, defined as follows.

A map $\mathcal{CO}$ that associates to every ORS X a set of branchings $\mathcal{CO}(X)$ of X is a *confluence obstruction map* when every terminating ORS X is confluent iff it is $\mathcal{CO}(X)$-confluent. For example, there exists a minimal confluence obstruction map $\mathcal{M}$ defined as $\mathcal{M}(X) = \varnothing$ if X is confluent, and $\mathcal{M}(X) = \{b\}$ if X is non-confluent and b is a non-confluent branching. However, it is impracticable to write a completion procedure wrt $\mathcal{M}$, as it would imply being able to determine confluence and compute a non-confluent branching in the first place.

Another approach is to consider confluence-generating sets of branchings. A set B of branchings of an ORS X is *confluence-generating* if, for any branching (f, g) of X, there exist branchings

$(f_1, g_1), \dots, (f_n, g_n)$, which are additive, multiplicative, or in B, rewriting paths f' and g', and contexts $C_1, \dots, C_n$ such that $f = C_1[f_1] \cdot f'$, $g = C_n[g_n] \cdot g'$, and for all $1 \leqslant i \leqslant n-1$, $C_i[g_i] = C_{i+1}[f_{i+1}]$. We get the following lemma:

2.3. Lemma. *A map $\mathcal{B}$ that associates to every ORS X a confluence-generating set of branchings $\mathcal{B}(X)$ is a confluence obstruction map.*

The converse is not true, however: consider $\mathcal{M}(X)$, which is not confluence-generating as soon as X is confluent with a branching.

There are several examples confluence-generating sets in the literature. The classical one is the set of critical branchings, in which case Lemma 2.3 is the CBL, also called Buchberger's criterion for linear rewriting systems [4, 23]. Smaller confluence-generating sets were developed to take into account additional relations between critical branchings. These sets, along with the corresponding proofs of Lemma 2.3, were defined for commutative algebras [5, 22], non-commutative algebras [16, 17], and non-symmetric operads [20].

Small confluence-generating sets appear to be a good compromise between minimizing the size of a confluence obstruction map and minimizing the number of times the confluence obstruction map is called. The question is then to find a minimal confluence-generating set. In certain cases, the answer is known: for instance, for quadratic ORS's, critical branchings form a minimal confluence-generating set.

3 Confluence by elimination

Linear rewriting can be done without a monomial order [13, 19], but in most applications the rewriting rules are compatible with a monomial order. In this case convergent AlgRS's are *Gröbner bases*, and rewriting properties are formulated algebraically. Branchings are described by *S-polynomials* and confluence means that every S-polynomial reduces to zero. Finally, the elimination of critical branchings is encoded by relations among relations (syzygies). In this section we give an implementation of the CBCP for ORS using Gaussian elimination inspired by the F4 algorithm [16].

Fix an ORS $X = (\Sigma, \prec, R)$ compatible with a monomial order $\prec$. Let $P = \{f_1, \dots, f_n\}$ be a set of rewriting monomials on Σ, and consider the totally ordered set $\mathrm{Supp}(P) := \cup_{f \in P} \mathrm{Supp}(s(f) - t(f)) = \{u_1 \prec \cdots \prec u_k\}$. We define the matrix $M_P \in \mathcal{M}_{n,k}(\mathbb{K})$ where $(M_P)_{i,j}$ is the coefficient of u_j in $s(p_i) - t(p_i)$. Thus we can read the elements of P as the rows of M_P, where the largest nonzero coefficient is the source monomial and the other coefficients correspond to the target polynomial. For examples, see the matrices in the appendix.

The first step of completion is as follows. We fix a reduction strategy σ. For each rewriting monomial p of P, we calculate a reduction path, starting with p, from $s(p)$ to a normal form, which follows σ after the first step. We then re-

```
GetRM(σ)(X, P)
Input: An ORS X = (Σ, ≺, R),
A list of rewriting monomials P.
Output: A list of rewriting monomials R′.
R′ := P;
T := ∪_{f∈P} Supp(t(f));
treated := lm(P);
while T ≠ ∅ do
    select u ∈ T;
    T := T \ {u};
    treated := treated ∪ {u};
    if σ(u) not an identity then
        R′ := R′ ∪ {σ(u)};
        T := T ∪ {Supp(t(σ(u))) \ treated};
return R′;
```

turn the set $\mathsf{R}' := \mathsf{GetRM}(\sigma)(\mathsf{X}, \mathsf{P})$ of rewriting monomials wrt R that appear in these paths. As for the case of non-commutative algebras [16, Prop. 4.21], if P is finite, then $\mathsf{GetRM}(\mathsf{X}, \mathsf{P})$ terminates.

The next step is to reduce the matrix $M_{\mathsf{R}'}$ to its row reduced echelon form, $\mathsf{RowReduce}(M_{\mathsf{R}'})$, by Gaussian elimination. The resulting rows whose largest monomials are not sources of rewriting monomials in R' form a set of new rewriting rules P', which is the result of $\mathsf{Reduction}(\mathsf{X}, \mathsf{P})$.

$\mathsf{Reduction}(\mathsf{X}, \mathsf{P})$
Input: An ORS $\mathsf{X} = (\Sigma, \prec, \mathsf{R})$,
A list of rewriting monomials P.
Output: A list of rewriting rules P'.

$\mathsf{R}' := \mathsf{GetRM}(\sigma)(\mathsf{X}, \mathsf{P})$;
$M' := \mathsf{RowReduce}(M_{\mathsf{R}'})$;
$\mathsf{P}' := \{\alpha$ row of M'
$\quad \mid \alpha \neq 0$ and $s(\alpha) \notin s(\mathsf{R}')\}$;
return P';

Finally, we choose a confluence obstruction map $\mathcal{CO}$ and a *selection strategy* $\mathcal{S}$, that returns a subset of branchings, in order to parallelize the completion procedure. The selection strategy in the procedure $\mathsf{F4}$ is equivalently a filtration on `Branchings`. For instance, the *normal selection strategy* consists in filtering branchings by weight of the source, and starting with those of minimal leading weight [11]. For homogeneous presentations, this appears to works well.

3.1. Theorem. *Let X be an ORS, $\mathcal{CO}$ a confluence obstruction map and $\mathcal{S}$ a selection strategy. If the procedure $\mathsf{F4}(\mathcal{C}, \mathcal{S})$ terminates on X, then the ORS $\mathsf{F4}(\mathcal{CO}, \mathcal{S})(\mathsf{X})$ is convergent.*

The proof works as follows. $\mathsf{F4}(\mathcal{CO}, \mathcal{S})(\mathsf{X})$ terminates only if, at some iteration of the first while loop, P' is an empty set for every iteration of the second while loop. This only happens if X' is $\mathcal{CO}(\mathsf{X}')$-convergent, which is equivalent to convergence of X'.

Note that an associative algebra can be seen as a non-symmetric operad concentrated in arity 1. By specifying $\mathcal{CO}$ and $\mathcal{S}$ and restricting $\mathsf{F4}$ to associative algebras, we recover some previously published procedures. If $\mathcal{CO}$ returns the set of critical branchings, we get the non-commutative F4 procedure introduced in [25]. If $\mathcal{S}$ selects a single branching and $\mathcal{CO}$ returns the set of critical branchings, we get the non-commutative Buchberger procedure [1, 2]. If $\mathcal{CO}$ eliminates critical branchings following the optimizations of [17, 25] (interreduction and chain criterion), then we recover their procedures.

$\mathsf{F4}(\mathcal{CO}, \mathcal{S})(\mathsf{X})$
Input: An ORS $\mathsf{X} = (\Sigma, \prec, \mathsf{R})$.
Output: A convergent ORS
$\mathsf{X}' = (\Sigma, \prec, \mathsf{R}')$.

$\mathsf{R}' := \mathsf{R}$;
AddedRules := true;
while AddedRules **do**
  AddedRules := false;
  Branchings := $\mathcal{CO}(\Sigma, \mathsf{R}')$;
  while Branchings $\neq \varnothing$ **do**
    $\mathsf{B} :=$ $\mathcal{S}$(Branchings);
    Branchings := Branchings $\setminus$ B;
    $\mathsf{P} := \cup_{\{\mathsf{f},\mathsf{g}\}\in\mathsf{B}}\{\mathsf{f}, \mathsf{g}\}$;
    $\mathsf{P}' := \mathsf{Reduction}((\Sigma, \prec, \mathsf{R}'), \mathsf{P})$;
    if $\mathsf{P}' \neq \varnothing$ **then**
      $\mathsf{R}' := \mathsf{R}' \cup \mathsf{P}'$;
      AddedRules := true;
return $(\Sigma, \prec, \mathsf{R}')$;

References

[1] George M. Bergman. The diamond lemma for ring theory. *Adv. in Math.*, 29(2):178–218, 1978.

[2] Leonid A. Bokut. Imbeddings into simple associative algebras. *Algebra i Logika*, 15(2):117–142, 245, 1976.

[3] Murray R. Bremner and Vladimir Dotsenko. *Algebraic operads*. CRC Press, Boca Raton, FL, 2016. An algorithmic companion.

[4] Bruno Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal (An Algorithm for Finding the Basis Elements in the Residue Class Ring Modulo a Zero Dimensional Polynomial Ideal)*. PhD thesis, Mathematical Institute, University of Innsbruck, Austria, 1965. English translation in J. of Symbolic Computation, Special Issue on Logic, Mathematics, and Computer Science: Interactions. Vol. 41, Number 3-4, Pages 475–511, 2006.

[5] Bruno Buchberger. A criterion for detecting unnecessary reductions in the construction of gröbner bases. In *Symbolic and Algebraic Computation*, pages 3–21, 01 1979.

[6] Cyrille Chenavier. A lattice formulation of the noncommutative F_4 procedure. *Internat. J. Algebra Comput.*, 29(1):23–40, 2019.

[7] Vladimir Dotsenko and Anton Khoroshkin. Gröbner bases for operads. *Duke Math. J.*, 153(2):363–396, 2010.

[8] Vladimir Dotsenko and Bruno Vallette. Higher Koszul duality for associative algebras. *Glasg. Math. J.*, 55(A):55–74, 2013.

[9] Vladimir Dotsenko and Mikael Vejdemo-Johansson. Implementing Gröbner bases for operads. In *OPERADS 2009*, volume 26 of *Sémin. Congr.*, pages 77–98. Soc. Math. France, Paris, 2013.

[10] Benjamin Dupont. Rewriting modulo isotopies in pivotal linear (2,2)-categories. submitted preprint, arXiv:1906.03904, June 2019.

[11] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1-3):61–88, June 1999.

[12] Rüdiger Gebauer and H. Michael Möller. On an installation of Buchberger's algorithm. *J. Symbolic Comput.*, 6(2-3):275–286, 1988. Computational aspects of commutative algebra.

[13] Yves Guiraud, Eric Hoffbeck, and Philippe Malbos. Convergent presentations and polygraphic resolutions of associative algebras. *Math. Z.*, 293(1-2):113–179, 2019.

[14] Yves Guiraud and Philippe Malbos. Higher-dimensional categories with finite derivation type. *Theory Appl. Categ.*, 22:No. 18, 420–478, 2009.

[15] Yves Guiraud and Philippe Malbos. Higher-dimensional normalisation strategies for acyclicity. *Adv. Math.*, 231(3-4):2294–2351, 2012.

[16] Clemens Hofstadler. Certifying operator identities and ideal membership of noncommutative polynomials, March 2020. Masterarbeit, 2020.

[17] Jamal Hossein Poor, Clemens G. Raab, and Georg Regensburger. Algorithmic operator algebras via normal forms in tensor rings. *Journal of Symbolic Computation*, 85:247–274, 2018. 41th International Symposium on Symbolic and Alge-braic Computation (ISSAC'16).

[18] Donald Knuth and Peter Bendix. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra (Proc. Conf., Oxford, 1967)*, pages 263–297. Pergamon, Oxford, 1970.

[19] Philippe Malbos and Isaac Ren. Shuffle polygraphic resolutions for operads. submitted preprint, arXiv:2012.15718, December 2020.

[20] Philippe Malbos and Isaac Ren. Completion in operads via essential syzygies. In *Proceedings of the 46th International Symposium on Symbolic and Algebraic Computation*, ISSAC '21, New York, NY, USA, 2021. Association for Computing Machinery.

[21] Martin Markl and Elisabeth Remm. (Non-)Koszulness of operads for $\mathfrak{n}$-ary algebras, galgalim and other curiosities. *J. Homotopy Relat. Struct.*, 10(4):939–969, 2015.

[22] H. Michael Möller, Teo Mora, and Carlo Traverso. Gröbner bases computation using syzygies. In *Papers from the International Symposium on Symbolic and Algebraic Computation*, ISSAC '92, page 320–328, New York, NY, USA, 1992. Association for Computing Machinery.

[23] Teo Mora. An introduction to commutative and noncommutative Gröbner bases. *Theoret. Comput. Sci.*, 134(1):131–173, 1994. Second International Colloquium on Words, Languages and Combinatorics (Kyoto, 1992).

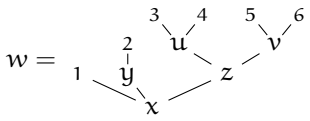
[24] Maurice Nivat. Congruences parfaites et quasi-parfaites. In *Séminaire P. Dubreil, 25e année (1971/72), Algèbre, Fasc. 1, Exp. No. 7*, page 9. Secrétariat Mathématique, Paris, 1973.

[25] Xingqiang Xiu. *Non-commutative Gröbner Bases and Applications*. PhD thesis, Universität Passau, 2012.

Appendix

As an illustration, we execute algorithm F4 on an ORS presenting the anti-associative operad. First, we introduce some notations.

Preliminaries. We represent monomials by planar trees with numbered inputs. For instance,



is a monomial where the arities are $\mathrm{AR}(x) = 3$, $\mathrm{AR}(y) = 1$, and $\mathrm{AR}(z) = \mathrm{AR}(u) = \mathrm{AR}(v) = 2$. The *weight* of a monomial u is the number of its inner vertices. For instance, $|w| = 5$.

Let P be a collection. For $x \in P(k)$, $y \in P(n)$, and $1 \leqslant i \leqslant k$, denote by

$$x \circ_i y := x \circ (\varepsilon, \dots, \varepsilon, \underset{i}{y}, \varepsilon, \dots, \varepsilon)$$

the *elementary composition* of x and y.

Example. Consider the following ORS that presents the *anti-associative operad* [21]

$$X := \langle x \in X(2) \mid f : x \circ_1 x \to -x \circ_2 x \rangle.$$

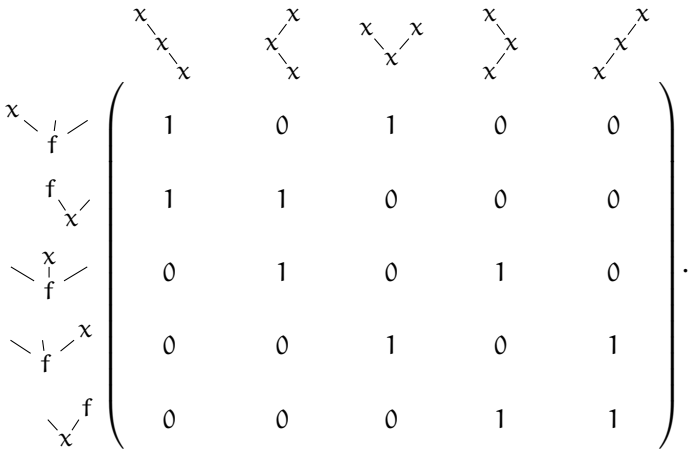
Let us study the execution of algorithm F4 with:

1. the confluence obstruction map that selects essential branchings, [20],
2. the selection strategy that selects the branchings of lowest weight,
3. the reverse path-lexicographic monomial order $\prec$, [3],
4. the reduction strategy σ given by taking the smallest rewriting monomial for the context path-lexicographic order defined in [20].

At the first iteration of the algorithm F4, there is one essential branching $(f \circ_1 x, x \circ_1 f)$. The algorithm GetRM applied to $(X, \{f \circ_1 x, x \circ_1 f\})$ returns the set

$$R' = \{x \circ_1 f, f \circ_2 x, x \circ_2 f, f \circ_1 x, f \circ_3 x\}.$$

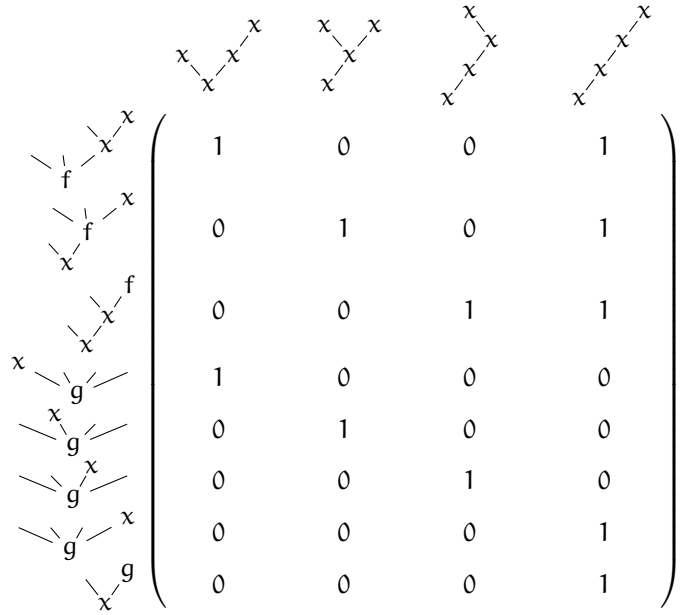
Then the matrix $M_{R'}$ is of the following form



where the columns are ordered by reverse path-lexicographic order. We check that $\mathsf{RowReduce}(M_{R'})$ is the 5×5 identity matrix, and that $\mathsf{Reduction}(X, \{f \circ_1 x, x \circ_1 f\})$ returns one rewriting rule, $g : x \circ_2 (x \circ_2 x) \to 0$, which we add to the ORS. At the next iteration, there are four essential branchings:

$$P := \{(f \circ_3 (x \circ_2 x), g \circ_1 x), (x \circ (f \circ_3 x), g \circ_2 x), (x \circ_2 (x \circ_2 f), g \circ_3 x), (x \circ_2 g, g \circ_4 x)\}.$$

Using the selection strategy, we once again select all branchings. The matrix $M_{\mathsf{GetRM}(\sigma)(X,P)}$ is



Since each column corresponds to the source of a rewriting monomial in R', the algorithm $\mathsf{Reduction}$ cannot produce new rewriting rules. Thus, the procedure $\mathsf{F4}$ terminates and the final convergent presentation is

$$\langle\, x \in X(2) \mid f : x \circ_1 x \to -x \circ_2 x,\ g : x \circ_2 (x \circ_2 x) \to 0 \,\rangle.$$