Some results on confluence: decision and what to do without

Florent Jacquemard

Some results on confluence: decision and what to do without.

Florent Jacquemard[1*]

INRIA – Sorbonne Universités
STMS (IRCAM-CNRS-UPMC), Paris
florent.jacquemard@inria.fr

Abstract

We recall first some decidability results on the confluence of TRS, and related properties about unicity of normal forms. In particular we put it in perspective old proofs of undecidability of confluence for the class of flat systems with more recent results, in order to discuss the importance of linearity wrt these decision problems.

Second, we describe a case study on musical rhythm notation involving modeling rewrite systems which are not confluent. In this case, instead of applying rewrite rules directly, we enumerate the equivalence class of a given term using automata-based representations and dynamic programming.

1 Confluence (un)decidability

When term rewriting systems (TRS) are used as models in fields such as functional programming languages semantics, automated deduction or system or program verification, the application of rewrite rules can be highly non-deterministic. Confluence permits to relax from this problem by guaranteeing that divergent reduction will eventually converge to a canonical form, in case of termination. It is therefore an crucial property to decide for TRS.

Decidability of confluence for linear TRS. Confluence of TRS is undecidable in general, even for linear systems (every variable can occur at most once in every left- or right-hand-side of rules) [28]. It has been shown decidable for ground TRS (rewrite rules without variables) [18, 3] and for left-linear right-ground TRS [2]. Polynomial time decision procedures have been proposed years later for ground TRS [1, 22], for left-shallow-linear and right-ground TRS (every variable can occur at most once and at depth at most one in every left-hand-side of rule) [22], for linear-shallow TRS (every variable occurs at most once in each rule and at depth at most one) [22, 10], and for linear and shallow TRS (every variable occurs at most once and at depth at most one in each side rule but can occur twice in a rule) [7].

Uniqueness of Normal Forms. The decidability of several alternatives to confluence has been studied. A first alternative, *uniqueness of normal forms* ($\mathrm{UN}^=$), implied by confluence, expresses that no two distinct normal forms (irreducible terms) can be equivalent modulo the rewrite system considered. $\mathrm{UN}^=$ has been shown decidable for ground TRS [28], and for shallow TRS (without the restriction of linearity) [19]. It is also polynomial time decidable for shallow and linear TRS [24]. It is undecidable for right ground TRS [26], for linear, non-collapsing (the right-hand-side of rules cannot be a variable), variable-preserving, and depth-two TRS [25], for

*co-authors for the results mentioned in §1: Ichiro Mitsuhashi, Michio Oyamaguchi, Guillem Godoy, and in S2: Jean Bresson, Masahiko Sakai, Adrien Ycart, Adrien Maire, Pierre Donat Bouillud, Slawek Staworko.

left-linear and left-flat TRS with with depth-two right-hand sides of rules [19] as well as for right-ground, right-flat TRS [25].

A second alternative, *unique normalization* (UN) expresses that every term can reach at most one normal form using the TRS considered. $\text{UN}^=$ implies UN but the converse is not true. UN is decidable in polynomial time for ground TRS [27], and also for for shallow and linear TRS [9]. On the negative side, UN is undecidable for right-ground TRS [23], for flat TRS (left- and right- hand side of rules have depth at most one) [8], for linear and right-flat TRS [11] and for flat and right-linear TRS [9].

Decidability of confluence for non-linear TRS The linearity is often considered as a yardstick when considering decision of properties of TRS such as confluence, reachability or joinability. For instance, tree automata based methods sometimes used in this context [18, 3, 2, 9] need, in case of non-linear TRS, generalized models with difficult decision problems.

Confluence is shown undecidable for flat (non-linear) TRS [14, 17] by reduction of reachability, also shown undecidable in this case (note that this is in contrast with $\text{UN}^=$ [19]). The latter proofs have been simplified drastically in [8]. However, confluence has been shown decidable for some classes of TRS allowing non-linear rules, like right-ground TRS (without restriction on the left-hand-sides of rules) [16], and shallow and right-linear TRS [12].

The latter proof uses decidability of reachability and joinability, both implied by regularity preservation result. To our knowledge, it is an open question whether confluence is decidable for other classes of TRS preserving regularity such as right-linear and finite-path-overlapping TRS [21] (shallow right-linear TRS are a particular case) or Layer Transducing TRS [20]. It is also interesting to consider the decision of confluence for particular rewriting strategies *e.g.* bottom-up [5, 6]. Finally, it can be observed that collapsing (right-variable) rules are essential in shifted pairing like constructions for undecidability proofs [14, 17, 8]. It is also unknown whether confluence is decidable for shallow and non-collapsing TRS.

2 What to do when there is no confluence

Traditional music notation is since centuries the standard format for the communication, exchange, and preservation of musical works in Western musical practice. We have been working recently on modeling the notation of rhythm (durations), following an approach based on formal languages and term rewriting.

In common western music notation, durations values are expressed proportionally, by recursive subdivisions of a unit (*beat*). This hierarchical definition induces naturally tree-structured representations called *rhythm trees* (RT). Every position in a RT is associated to a duration value. In a simple variant (see Figure 1), the root position is associated a fixed duration value and every non-root position is associated the duration of its parent p_0 divided by the number of edges outgoing from p_0. Moreover, if a leaf position p labeled by $\circ$, the the duration of p is added to the duration of the next leaf p' in depth-first-traversal (if it exists). The other leafs may be labeled by symbols giving information on notes, rests *etc*, and the labels of inner positions are not significant (here we use named after their arity $2, 3, 4 \ldots$). To a RT, we associate the sequence of durations of the non-o leaves (in *dfs*). To capture more complex rhythm notations, we use a dag representations not described here.

The RT representations are used in a new tool for the transcription of timestamped event sequences into a music notation [29]. It is implemented as a library of the algorithmic composition framework OpenMusic (Figure 2). We are also developing Music Information Retrieval

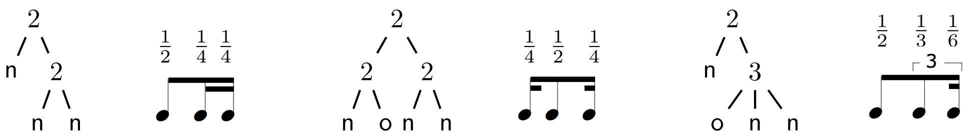

Figure 1: Rhythm Trees with associated duration sequences (symbol n represents a note).

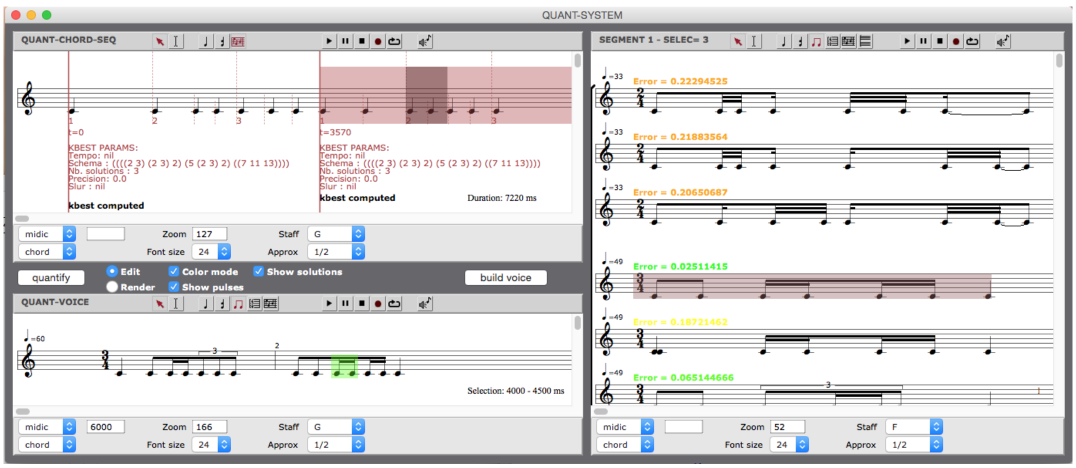

Figure 2: Transcription librairy for OpenMusic (Ircam). http://repmus.ircam.fr/cao/rq

tasks based on RT representations, in particular for querying bases of digital music scores (*e.g.* by *query by tapping*) and for musicologist research, using similarity measures and tree edit distances.

Structural theory of RT. For reasoning about rhythm notations in the above tasks, we define an equivalence between RT with term rewriting rules [15, 4]. For instance, the rules $2(\mathsf{o},\mathsf{n}) \to \mathsf{n}$, $3(\mathsf{o},\mathsf{o},\mathsf{n}) \to \mathsf{n}$,... and $2(\mathsf{o},\mathsf{o}) \to \mathsf{o}$, ...comply with the semantics of o presented above, and rules of the form $3\big(2(x_1,x_2), 2(x_3,x_4), 2(x_5,x_6)\big) \to 2\big(3(x_1,x_2,x_3), 3(x_4,x_5,x_6)\big)$ can be used in order to simplify RT. The TRS containing these simple rules is not confluent. For instance, starting from $t = 3\big(2(\mathsf{o},\mathsf{o}), 2(\mathsf{n},\mathsf{o}), 2(\mathsf{o},\mathsf{n})\big)$, we have the following non-joinable critical peak:

$$3(\mathsf{o}, 2(\mathsf{n},\mathsf{o}), \mathsf{n}) \xleftarrow{*} t \to 2\big(3(\mathsf{o},\mathsf{o},\mathsf{n}), 3(\mathsf{o},\mathsf{o},\mathsf{n})\big) \xrightarrow{*} 2(\mathsf{n},\mathsf{n}).$$

Exploring sets of equivalent terms. Therefore, in order to to reason about sets of equivalent terms (in particular the set $[\![t]\!]$ of terms equivalent to a given RT t), instead of applying rewriting to reach a canonical normal form that does not exist, we use automata-based characterizations. Some techniques like *tree automata completion*, can be used to compute a tree automaton recognizing the rewrite closure of a given regular tree set (in particular recognizing $[\![t]\!]$ given $\{t\}$), by superposition of rewrite rules into tree automata transition rules. Such techniques have been used for verify safety properties of program or systems modeled as TRS (possibly not confluent) by reduction to the problem of emtiness of tree automata intersection (*regular tree model checking*).

With rewrite rules like the above ones, it is not easy to establish the termination of standard tree automata completion procedures. Even though in our case in practice we only need to consider terms of a bounded depth, hence finite set of terms, it is neither easy to reasonably bound the size of the automaton obtained this way. As an alternative, we have developed an ad hoc construction using the duration sequence associated to a given RT, and a tree automaton representing the family of RT that we want to consider. Once an automaton recognizing $[\![t]\!]$ is constructed, we use dynamic programming for the lazy enumeration of this set, according to a measure of tree complexity, following techniques of *k-best parsing* [13]. This way, we can enumerate efficiently the rhythms equivalent to a given rhythm, by increasing complexity.

References

[1] Hubert Comon, Guillem Godoy, and Robert Nieuwenhuis. The confluence of ground term rewrite systems is decidable in polynomial time. In *Proceedings of the 42nd IEEE symposium on Foundations of Computer Science* (FOCS), pages 298–307. IEEE Computer Society, 2001.

[2] Max Dauchet, Thierry Heuillard, Pierre Lescanne, and Sophie Tison. Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems. *Inf. Comput.*, 88(2):187–201, 1990.

[3] Max Dauchet and Sophie Tison. The theory of ground rewrite systems is decidable. In *Proceedings Fifth Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 242–248. IEEE Computer Society, 1990.

[4] Pierre Donat-Bouillud, Florent Jacquemard, and Masahiko Sakai. Towards an equational theory of rhythm notation. In *Music Encoding Conference*, 2015.

[5] Irène Durand, Géraud Sénizergues, and Marc Sylvestre. Termination of linear bounded term rewriting systems. In *Proceedings of the 21st Int. Conf. on Rewriting Techniques and Applications* (RTA), vol.6, pages 341–356 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 2010.

[6] Irène Durand and Marc Sylvestre. Left-linear Bounded TRSs are Inverse Recognizability Preserving. In *22nd RTA*, volume 10 of LIPIcs, pages 361–376, 2011.

[7] G. Godoy, A. Tiwari, and R. Verma. On the confluence of linear shallow term rewrite systems. In *20th Intl. Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 2607 of *Lecture Notes in Computer Science*, pages 85–96. Springer, 2003.

[8] Guillem Godoy and Hugo Hernández. Undecidable properties for flat term rewrite systems. *Applicable Algebra in Engineering, Communication and Computing*, 20(2):187–205, 2009.

[9] Guillem Godoy and Florent Jacquemard. Unique normalization for shallow trs. In *20th International Conference on Rewriting Techniques and Applications (RTA)*, volume 5595 of *Lecture Notes in Computer Science*, pages 63–77. Springer, 2009.

[10] Guillem Godoy, Robert Nieuwenhuis, and Ashish Tiwari. Classes of term rewrite systems with polynomial confluence problems. *ACM Trans. Comput. Logic*, 5(2):321–331, 2004.

[11] Guillem Godoy and Sophie Tison. On the normalization and unique normalization properties of term rewrite systems. In *Proc. 21st International Conference on Automated Deduction (CADE)*, volume 4603 of *Lecture Notes in Computer Science*, pages 247–262. Springer, 2007.

[12] Guillem Godoy and Ashish Tiwari. Confluence of shallow right-linear rewrite systems. In *19th International Workshop of Computer Science Logic, CSL*, volume 3634 of *Lecture Notes in Computer Science*, pages 541–556. Springer, 2005.

[13] Liang Huang and David Chiang. Better k-best parsing. In *Proceedings of the Ninth International Workshop on Parsing Technology*, Parsing '05, pages 53–64, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[14] Florent Jacquemard. Reachability and confluence are undecidable for flat term rewriting systems. *Information Processing Letters*, 87(5):265–270, 2003.

[15] Florent Jacquemard, Pierre Donat-Bouillud, and Jean Bresson. A Structural Theory of Rhythm Notation based on Tree Representations and Term Rewriting. In *5th International Conference on Mathematics and Computation in Music (MCM)*, volume 9110 of *Lecture Notes in Artificial Intelligence*. Springer, 2015.

[16] Lukasz Kaiser. Confluence of right ground term rewriting systems is decidable. In *International Conference on Foundations of Software Science and Computation Structures (Fossacs)*, pages 470–489. Springer, 2005.

[17] I. Mitsuhashi, M. Oyamaguchi, and F. Jacquemard. The confluence problem for flat TRSs. In *Proc. 8th Intl. Conf. on Artificial Intelligence and Symbolic Computation (AISC'06)*, volume 4120 of *LNAI*, pages 68–81. Springer, 2006.

[18] M. Oyamaguchi. The Church-Rosser property for ground term rewriting systems is decidable. *Theoretical Computer Science*, 49:43–79, 1987.

[19] Nicholas Radcliffe and Rakesh M. Verma. Uniqueness of normal forms is decidable for shallow term rewrite systems. In *Conference on Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2010.

[20] H. Seki, T. Takai, Y. Fujinaka, and Y. Kaji. Layered Transducing Term Rewriting System and Its Recognizability Preserving Property. In *Int. Conf. on Rewriting Techniques and Applications (RTA)*, volume 2378 of *Lecture Notes in Computer Science*, pages 98–113. Springer, 2002.

[21] T. Takai, Y. Kaji, and H. Seki. Right-linear finite path overlapping term rewriting systems effectively preserve recognizability. In *11th Int. Conf. on Rewriting Techniques and Applications (RTA)*, volume 1833 of *Lecture Notes in Computer Science*, pages 246–260. Springer, 2000.

[22] A. Tiwari. Deciding confluence of certain term rewriting systems in polynomial time. In *IEEE Symposium on Logic in Computer Science (LICS)*, pages 447–456. IEEE Society, 2002.

[23] R. Verma. Complexity of normal form properties and reductions for rewriting problems. *Fundamenta Informaticae*, 2008.

[24] R. Verma and J. Zinn. A polynomial-time algorithm for uniqueness of normal forms of linear shallow term rewrite systems. In *Symposium on Logic in Computer Science LICS (short presentation)*, 2006.

[25] Rakesh Verma. New undecidability results for properties of term rewrite systems. *Electronic Notes in Theoretical Computer Science*, 290:69–85, 2008.

[26] Rakesh Verma. Complexity of normal form properties and reductions for term rewriting problems complexity of normal form properties and reductions for term rewriting problems. *Fundamenta Informaticae*, 92(1-2):145–168, 2009.

[27] Rakesh Verma and Ara Hayrapetyan. A new decidability technique for ground term rewriting systems with applications. *ACM Trans. Comput. Logic*, 6(1):102–123, January 2005.

[28] Rakesh M. Verma, Michael Rusinowitch, and Denis Lugiez. Algorithms and reductions for rewriting problems. *Fundam. Inf.*, 46:257–276, August 2001.

[29] Adrien Ycart, Florent Jacquemard, Jean Bresson, and Slawomir Staworko. A Supervised Approach for Rhythm Transcription Based on Tree Series Enumeration. In *Proceedings of the 42nd International Computer Music Conference (ICMC)*. ICMA, 2016.