Transformations of Conditional Rewrite Systems Revisited (Extended Abstract)

Karl Gmeiner and Bernhard Gramlich

TU Wien, Austria, {gmeiner,gramlich}@logic.at

We revisit known transformations of conditional rewrite systems to unconditional ones in a systematic way. We present a unified framework for describing and classifying such transformations, discuss the major problems arising, provide simplified (old) and new counterexamples to certain (desirable) properties of specific transformations, and finally present a new transformation which has some advantages as compared to a quite recent approach, namely the one of [1].[1] In this abstract, due to lack of space we focus on the latter contribution, after briefly discussing major general issues with such transformation approaches.

Conditional term rewrite systems (CTRSs) and conditional equational specifications are very important in algebraic specification, prototyping, implementation and programming. They naturally occur in most practical applications. Yet, compared to unconditional term rewrite systems (TRSs), CTRSs are much more complicated, both in theory (especially concerning criteria and proof techniques for major properties of such systems like confluence and termination) and practice (implementing conditional rewriting in a clever way is far from being obvious, due to the inherent recursion when evaluating conditions). For these (theoretical and practical) reasons, transforming CTRSs into (unconditional) TRSs in an adequate way has been studied for a long time cf. e.g. [3], [4], [5], [6], [7], [8], [9], [1], [10], [11], [12]. The motivations for these transformations were manifold, depending on the overall goal of the analysis (see below).

Roughly, all transformations work by translating the original syntax (signature and terms) into an extended or modified one using auxiliary function symbols, and by translating the rules in a corresponding way such that the evaluation of conditions and some control structure is (appropriately) encoded within the resulting unconditional TRS.

Already from this abstract point of view, the main questions and problems of such transformation approaches can be inferred:

- How are the relevant (syntactical and semantic) properties of a given CTRS $\mathcal{R}$ and its transformed TRS $\mathcal{R}'$ related?
- Is it possible to infer a property $P(\mathcal{R})$ from $P(\mathcal{R}')$ (*soundness*) and vice versa (*completeness*). Typically, and unlike in many other settings, here *completeness* properties are less difficult to show/obtain than *soundness* properties. The intuitive reason is that the transformations are designed such that every reduction step that was possible in the original system can be simulated

[1] The work presented here is partially based on [2].

by a reduction (sequence) in the transformed system. On the other hand, since the evaluation of conditions in the transformed system is done using unconditional rewrite rules, there is no obvious encapsulation any more as in the conditional case. And this entails the danger of enabling reductions that were originally impossible (because a failed attempt to verify a condition in the conditional system has no further consequences). Hence, for instance soundness usually is a problem.
- From a theoretical point of view: Is a given transformation useful for analyzing a given CTRS via its transformed unconditional version?
- From a practical point of view: Does a given transformation yield an operational / executable specification / high-level implementation with good properties, e.g. in terms of the input/output behaviour, of efficiency, of comprehensibility, ...? Is (explicit meta-level) backtracking, corresponding to failed attempts of verifying conditions, in the transformed system avoided?

One property of transformations which is particularly important from a practical point of view, is the following: If we start a simulation (a reduction in the transformed TRS) from an transformed initial term and obtain a normal form in the transformed system, then the latter should correspond to a normal form of the initial term in the original CTRS (this property, together with a few other requirements, is called *computational equivalence* in [1]). Otherwise, some form of backtracking would be needed, because then we are stuck with a failed attempt of verifying conditions. As an example consider the (*oriented normal*) CTRS $\mathcal{R}$ (cf. [8, 1]) consisting of the two rules

$$f(g(x)) \to 0 \quad \Leftarrow \quad x \to 0 \qquad\qquad g(g(x)) \to g(x)$$

and the initial term $t = f(g(g(0)))$. *Unravelings* following the approach of [6, 7] use new function symbols to encode conditions and store the variable bindings until finally – if the conditions are verified – the right-hand side may be produced:

$$f(g(x)) \to U_1(x, x) \quad U_1(0, x) \to 0 \quad g(g(x)) \to g(x)$$

Here, t reduces to normal forms 0 and $U_1(g(0), g(0))$. However, the latter term does not correspond to a normal form in the original CTRS.

Transformations like the one of [8] increase the arity of some function symbols and encode the conditions in these new "conditional" arguments. For notational simplicity we will collect them in lists denoted by $[\ldots]$ where $[\,]$ represents the empty list. Using this approach, we get:

$$f'(g(x), [\,]) \to f'(g(x), [x]) \quad f'(g(x), [0]) \to 0 \quad g(g(x)) \to g(x)\,.$$

For *constructor–based* CTRSs, this approach of [8] yields good results, but in our example, which is not a constructor system, we still obtain – from the transformed initial term $t' = f'(g(g(0)), [\,])$ – an undesired normal form $f'(g(0), [g(0)])$ that does not correspond to a normal form of the initial term t in the original CTRS.

To solve this problem, [1] proposed an additional unary operand $\{.\}$ for "resetting" conditional arguments whenever an "inner" rewrite step occurs:

$$f'(g(x),[\,]) \to f'(g(x),\{x\}) \quad f'(g(x),\{0\}) \to \{0\} \qquad g(g(x)) \to \{g(x)\}$$
$$f'(\{x\},z) \to \{f'(x,[\,])\} \qquad g(\{x\}) \to \{g(x)\} \quad \{\{x\}\} \to \{x\}$$

Here, the only normal form of the transformed initial term $t' = f'(g(g(0)),[\,])$ is $\{0\}$ which corresponds to 0 as desired. Yet, in general this transformation destroys some syntactical properties of the original CTRS like being a constructor system, reinforces sequential processing of conditions and "(too) often" resets encoded conditions.

In our approach we encode conditions at "appropriate more inner" positions such that propagation of "reset information" is earlier possible. This approach also works for CTRSs with deterministic extra variables (DCTRSs), is a proper extension of the transformation of [8] and has better support for parallel rewriting. In the example, our new transformation yields the TRS

$$f(g'(x,[\,])) \to f(g'(x,[x])) \quad f(g'(x,[0])) \to 0 \quad g'(g'(x,z_2),z_1) \to g'(x,[\,])\,.$$

Now, starting from the transformed initial term $t' = f(g'(g'(0,[\,]),[\,]))$ which corresponds to t above, there is only one normal form as desired, namely 0.

References

1. Serbanuta, T.F., Rosu, G.: Computationally equivalent elimination of conditions. *Proc. 17th RTA*, LNCS 4098, Springer (2006) 19–34
2. Gmeiner, K.: Transformations of conditional term rewriting systems. Master's thesis, TU Wien (2007)
3. Bergstra, J., Klop, J.: Conditional rewrite rules: Confluence and termination. Journal of Computer and System Sciences **32**(3) (1986) 323–362
4. Giovanetti, E., Moiso, C.: Notes on the elimination of conditions. *Proc. 1st CTRS, Orsay, France, 2007*, LNCS 308, Springer (1988) 91–97
5. Viry, P.: Elimination of conditions. J. Symb. Comput. **28**(3) (1999) 381–401
6. Marchiori, M.: Unravelings and ultra-properties. In Hanus, M., Rodríguez-Artalejo, M.M., eds., Proc. 5th ALP, LNCS 1139, Springer (September 1996) 107–121
7. Ohlebusch, E.: Advanced Topics in Term Rewriting. Springer (2002)
8. Antoy, S., Brassel, B., Hanus, M.: Conditional narrowing without conditions. In: *Proc. 5th PPDP*, ACM Press (2003) 20–31
9. Rosu, G.: From conditional to unconditional rewriting. In: *Proc. 17th WADT*, LNCS 3423, Springer (2004) 218–233
10. Nishida, N., Mizutani, T., Sakai, M.: Transformation for refining unraveled conditional term rewriting systems. *ENTCS* 174(10), 2007.
11. Schernhammer, F., Gramlich, B.: On proving and characterizing operational termination of deterministic conditional rewrite systems. In: *Proc. 9th WST*, Paris, France. (2007) 82–85
12. Lucas, S., Meseguer, J., Marché, C., Urbain, X.: Proving operational termination of membership equational programs. Higher-Order and Symbolic Computation **21**(1–2) (1998) 59–88