On the (Non-)Existence of Least Fixed Points in Conditional Equational Logic and Conditional Rewriting

Bernhard Gramlich*
Inst. f. Computersprachen
Technische Universität Wien
Favoritenstr. 9, A-1040 Wien, Austria

Abstract

We investigate and review problems with basic definitions in conditional equational logic and conditional term rewriting that are closely related to (definitions via) least fixed point constructions. In particular, we discuss problems with non-well-defined (equality / rewrite) relations that are due to the non-existence of certain least fixed points, and present modified approaches to guarantee the existence of such fixed points.

1 Introduction, Motivation and Overwiew

Conditional equation systems (CES's) and conditional term rewriting systems (CTR's) are important specification, modelling and computation formalisms in many areas of mathematics and computer science. However, as compared to unconditional equation systems and (unconditional) term rewriting systems (TRS's), the logical and computational theory underlying CES's and CTRS's is well-known to be considerably more involved (cf. e.g. [17], [1] for surveys on unconditional and conditional equational logic and term rewriting).

Here we are concerned with conditional equations and rewrite rules of the form $l = r \Longleftarrow c$ and $l \to r \Longleftarrow c$, respectively, with left-hand side l, right-hand side r and condition c (where l, r are terms built from some given signature $\mathcal{F}$ of function symbols and some disjoint set $\mathcal{V}$ of variables).[1] Now, if the equality (or rewrite relation) induced by a system of such equations (rules) is to be formally defined, one needs to be more specific about the form of allowed conditions c. In the simplest case, c is just a (possibly empty) conjunction (denoted as list) of equations between terms: $s_1 = t_1, \ldots, s_n = t_n$. For being able to inductively define the rewrite relation induced by a set $\mathcal{R}$ of such rules of the form

$$l \to r \Longleftarrow s_1 = t_1, \ldots, s_n = t_n$$

we also have to be more specific about how to interpret "=" in the conditions $s_i = t_i$. Some common choices for this interpretation of equality in the conditions are (cf. e.g. [5], [9, 10, 8], [17]):

- logical equality, i.e., the equality induced by $\mathcal{R}$ considered as a conditional equation system
- semi-equational semantics, where "=" in the conditions is interpreted as convertibility $\leftrightarrow^*$ (where $\to$ is the rewrite relation to be defined)
- join semantics, where "=" is joinability $\downarrow$ (i.e.: $\to^* \circ {}^*\!\leftarrow$)
- oriented semantics (cf. e.g. [22], [25]) where "=" is reachability $\to^*$ (leading to conditions of the form $s_i \to^* t_i$).

*E-mail: gramlich@logic.at www: http://www.logic.at/staff/gramlich

[1] Of course, more general forms of conditions are also conceivable, and are sometimes considered.

With any of the above interpretations of equality in the conditions it is fairly straightforward to give an inductive definition of the (equality $=_{\mathcal{R}}$ in case of logical semantics of CES's or) rewrite relation $\to_{\mathcal{R}}$ induced by a set $\mathcal{R}$ of conditional rewrite rules of the corresponding type (semi-equational, join, or oriented CTRS's). Such inductive definitions correspond to the construction of the least fixed point of some monotone operator. The crucial ingredient that makes this kind of describing $\to_{\mathcal{R}}$ well-defined (i.e., that ensures the existence of the corresponding least fixed point) consists in the fact that the conditions only contain *positive* information. As soon as *negative* information in c is allowed, things get immediately much more involved and the above approach does in general not lead to a well-defined rewrite relation anymore (technically, this is due to the fact that the operator in the fixed point construction is no longer montonic in such cases). This basic problem in conditional equational logic and conditional term rewriting has appeared in many approaches in the literature and in various settings, and has sometimes caused considerable confusion. In certain cases the latter effect is mainly due to the fact that negation need not always occur in explicit form but may also be hidden somewhere (i.e., appear implicitly) in some assumptions or conditions.

Explicit uses of negation appear for instance when we allow negative equality literals in c, i.e., literals of the form $s_i \neq t_i$ (abbreviating $\neg(s_i = t_i)$). More generally, when considering equational clauses $L_1 \vee \ldots \vee L_k$ (where the L_i are positive or negative equality literals, i.e., of the form $u = v$ or $u \neq v$, respectively) with more than one positive literal[2] instead of conditional equations or rewrite rules, we also get this (kind of explicit) negation problem. Allowing negative equality literals in (the condition) c (of a conditional equation / rewrite rule) leads to what has sometimes been called *positive/negative conditional* equation (rewrite) systems (as coined by [16]). For further works on this type of systems we refer to e.g. [19], [23, 24] (in algebraic specification and inductive theorem proving), to [2], [13], [3] in (ordering-based) logic programming with equality and (inductive) theorem proving in first-order equational logic, and to [15] [6], [14], [7], [12, 11] in *transition system specifications with negative premises*, *operational semantics*, *process algebras* and related fields.

Impicit uses of negation are more subtle and often disguised. We distinguish here three types of implicit negation:

- *Normal* (join or oriented) CTRS, where rewrite rules have the form $l \to r \Longrightarrow s_1 \to^* t_1, \ldots, ..., s_n \to^* t_n$ with all t_i being $\mathcal{R}_u$-irreducible ground terms. Here, $\mathcal{R}_u$ is the "unconditional part" of the CTRS $\mathcal{R}$ under consideration, i.e., $\mathcal{R}_u := \{l \to r \mid l \to r \Longleftarrow c \in \mathcal{R}\}$. If only $\mathcal{R}$-irreducibility of the t_i is required, then $\to_{\mathcal{R}}$ cannot be (well) defined in the usual way (cf. e.g. [5], [17])!

- Conditional rewriting under *(position based) strategies*: One may be tempted to define *innermost* rewriting in $\mathcal{R}$ ($\to_{\mathcal{R},i}$), *outermost* rewriting in $\mathcal{R}$ ($\to_{\mathcal{R},o}$) (and similar restrictions of rewriting under position-based strategies), directly by some inductive construction of $\to_{\mathcal{R},i}$ or $\to_{\mathcal{R},o}$, respectively, which again leads to inconsistent results. The problem here is due to the fact that the strategy restriction implicitly incorporates negative information. For instance, innermost rewriting a term $f(a)$ at root position with some rule of the form $f(x) \to r \Longleftarrow c$ requires verifying that its proper subterm a is **not** reducible w.r.t. the rewrite relation being defined(!).

- *Priority rewrite systems* (cf. [4], [18], [21], [20]): In such systems the idea is that rewrite rules are equipped with a priority, and before rewriting a term with some rule, every other rule with higher priority must have already been (negatively) tested for applicability. In other words, this amounts to some *irreducibility* assumptions (i.e., a term must **not** be reducible by a certain subset of rewrite rules) in the definition of the rewrite relation itself and entails severe semantical problems.[3]

In the talk we shall review the mentioned problems (and approaches) from a systematic point of view, relating the problems with explicit and, in particular, implicit negation to a more abstract setting of (least) fixed point construction. Especially, we shall be concerned with the non-existence of least fixed points in certain cases and with sufficient conditions (within modified approaches) guaranteeing their existence.

[2] This generalizes conditional equations $l = r \Longleftarrow s_1 = t_1, \ldots, s_n = t_n$ which in clausal form $l = r \vee s_1 \neq t_1 \ldots \vee s_n \neq t_n$ have exactly one positive literal.

[3] In fact, priority rewrite systems are already fully problematic in the unconditional case, i.e., for unconditional TRS's!

References

[1] F. Baader and T. Nipkow. *Term rewriting and All That.* Cambridge University Press, 1998.

[2] L. Bachmair and H. Ganzinger. Perfect model semantics for logic programs with equality. In K. Furukawa, ed., *Proc. 8th Int. Conf. on Logic Programming*, pages 645–659, Paris, France, June 1991. MIT Press.

[3] L. Bachmair and H. Ganzinger. Rewrite-based equational theorem proving with selection and simplification. *Journal of Logic and Computation*, 4(3):217–247, 1994.

[4] J. Baeten, J. Bergstra, J. Klop, and P. Weijland. Term-rewriting systems with rule priorities. *Theoretical Computer Science*, 67(2/3):283–301, 1989.

[5] J. Bergstra and J. Klop. Conditional rewrite rules: Confluence and termination. *Journal of Computer and System Sciences*, 32(3):323–362, 1986.

[6] R. Bol and J. Groote. The meaning of negative premises in transition system specifications. *Journal of the ACM*, 43(5):863–914, 1996.

[7] P. D'Argenio and C. Verhoef. A general conservative extension theorem in process algebras with inequalities. *Theoretical Computer Science*, 177(2):351–380, 1997.

[8] N. Dershowitz and M. Okada. A rationale for conditional equational programming. *Theoretical Computer Science*, 75:111–138, 1990.

[9] N. Dershowitz, M. Okada, and G. Sivakumar. Canonical conditional rewrite systems. In E. Lusk and R. Overbeek, eds., *Proc. 9th Int. Conf. on Automated Deduction (CADE'88)*, LNCS, pages 538–549. Springer-Verlag, 1988.

[10] N. Dershowitz, M. Okada, and G. Sivakumar. Confluence of conditional rewrite systems. In S. Kaplan and J.-P. Jouannaud, eds., *Proc. 1st Int. Workshop on Conditional Term Rewriting Systems (CTRS'87)*, LNCS 308, pages 31–44. Springer-Verlag, 1988.

[11] W. Fokking and C. Verhoef. Conservative extension in positive/negative conditional term rewriting with applications to software renovation factories. In J.-P. Finance, ed., *Proc. 2nd Conf. on Fundamental Approaches to Software Engineering (FASE'99)*, LNCS 1577, pages 98–113, Amsterdam, The Netherlands, Mar. 1999. Springer-Verlag.

[12] W. Fokkink and C. Verhoef. A conservative look at operational semantics with variable binding. *Information and Computation*, 146(1):24–54, 1998.

[13] H. Ganzinger and J. Stuber. Inductive theorem proving by consistency for first-order clauses. In *Informatik – Festschrift zum 60. Geburtstag von Günter Hotz.* Teubner Verlag, 1992.

[14] R. Glabbeek. The meaning of negative premises in transition system specifications II. In F. Meyer auf der Heide and B. Monien, eds., *Proc. 23rd Int. Conf. on Automata, Languages and Programming (ICALP'96)*, LNCS 1099, pages 502–513, Paderborn, Germany, 1996. Springer-Verlag.

[15] J. Groote. Transition system specifications with negative premises. *Theoretical Computer Science*, 118(2):263–299, 1993.

[16] S. Kaplan. Positive/negative conditional rewriting. In S. Kaplan and J.-P. Jouannaud, eds., *Proc. 1st Int. Workshop on Conditional Term Rewriting Systems (CTRS'87)*, LNCS 308, pages 129–143. Springer-Verlag, 1987.

[17] J. W. Klop. Term rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, eds., *Handbook of Logic in Computer Science*, volume 2, chapter 1, pages 2–117. Clarendon Press, Oxford, 1992.

[18] C. Mohan. Priority rewriting: Semantics, confluence, and conditional. In N. Dershowitz, ed., *Proc. 3rd Int. Conf. on Rewriting Techniques and Applications (RTA'89)*, LNCS 355, pages 278–291, Chapel Hill, North Carolina, USA, Apr. 1989. Springer-Verlag.

[19] C. Mohan and M. Srivas. Conditional specifications with inequational assumptions. In S. Kaplan and J.-P. Jouannaud, eds., *Proc. 1st Int. Workshop on Conditional Term Rewriting Systems (CTRS'87)*, LNCS 308, pages 161–178, Orsay, France, 1988. Springer-Verlag.

[20] J. Pol. Operational semantics of rewriting with priorities. *Theoretical Computer Science*, 200(1/2):289–312, 1998.

[21] M. Sakai and Y. Toyama. Semantics and strong sequentiality of priority term rewriting systems. In H. Ganzinger, ed., *Proc. 7th Int. Conf. on Rewriting Techniques and Applications (RTA'96)*, LNCS 1103, pages 377–391, New Brunswick, NJ, USA, July 1996. Springer-Verlag.

[22] T. Suzuki, A. Middeldorp, and T. Ida. Level-confluence of conditional rewrite systems with extra variables in right-hand sides. In J. Hsiang, ed., *Proc. 6th Int. Conf. on Rewriting Techniques and Applications (RTA'95)*, LNCS 914, pages 179–193, Kaiserslautern, Germany, Apr. 1995. Springer-Verlag.

[23] C.-P. Wirth and B. Gramlich. A constructor-based approach for positive/negative conditional equational specifications. *Journal of Symbolic Computation*, 17:51–90, 1994.

[24] C.-P. Wirth and B. Gramlich. On notions of inductive validity for first-order equational clauses. In A. Bundy, ed., *Proc. 12th Int. Conf. on Automated Deduction (CADE'94)*, LNAI 814, pages 162–176, Nancy, France, 1994. Springer-Verlag.

[25] T. Yamada, J. Avenhaus, C. Loría-Sáenz, and A. Middeldorp. Logicality of conditional rewrite systems. *Theoretical Computer Science*, 236(1,2):209–232, 2000.