Fitting Semantics for Conditional Term Rewriting

Chilukuri K. Mohan
School of Computer and Information Science
Syracuse University, Syracuse, NY 13244-4100

Abstract

This paper investigates the semantics of conditional term rewriting systems with negation, which may not satisfy desirable properties like termination. It is shown that the approach used by Fitting [5] for Prolog-style logic programs is applicable in this context. A monotone operator is developed, whose fixpoints describe the semantics of conditional rewriting. Several examples illustrate this semantics for non-terminating rewrite systems which could not be easily handled by previous approaches.

1 Introduction

Conditional term rewriting systems (CTRS) have attracted much attention in the recent past as a useful generalization of the simpler formalism of term rewriting systems (TRS). But CTRS have not been unconditionally accepted, due to the absence of well defined semantics for conditional rewriting mechanisms. This paper suggests one remedy, following the approach of Melvin Fitting, who suggested similar semantics for Prolog-style logic programs [5].

Past work on the semantics of conditional term rewriting has followed three directions:

1. Impose restrictions on the syntax of the CTRS formalism to ensure termination and the existence of a unique precongruence which is considered to describe the meaning of the rewrite relation [8]. This approach does not define the meaning of rewriting when the CTRS does not satisfy the relevant termination criterion. Also, the termination criterion itself is undecidable, and is not a necessary condition for each rewrite step and all rewrite sequences to terminate finitely.
2. Give logical semantics for a CTRS *R* as a set of conditional equations $\mathcal{L}(R)$ together with a set of "default" negative equality literals [13]. This approach is useful if all rewrite sequences terminate or if the CTRS is intended to describe a specification based on a set of free constructor functions.
3. Transform CTRS into "equivalent" TRS, and identify the semantics of the CTRS with that of the transformed systems [1]. Assign an "initial algebra" semantics for TRS. The drawback of this approach is that it does not adequately describe the operational use of CTRS with negative literals in the antecedents of rules.

This paper attempts to fill the lacuna using an elegant approach of Fitting, following Kripke[10] who brought together Kleene's multivalued logics [9], and Tarski's lattice-theoretical fixpoint theorem [16]. Fitting [5] uses this approach to present an alternative to the semantics of logic programming given by Apt and Van Emden [2]. The main contribution of this paper is to show that this approach can also successfully explain the meaning of conditional rewriting systems with negation, including the problematic CTRS whose semantics have eluded the grasp of previous approaches
(e.g., $p \neq q \Rightarrow p \rightarrow q, p = q \Rightarrow p \rightarrow q$).

In the next section, we introduce CTRS and point out the deficiencies of a two-valued fixpoint semantics. In section 3, following some mathematical preliminaries, we describe the new semantics for conditional rewriting. Several examples are then given in section 4 to illustrate the semantics. References follow concluding remarks.

2 Preliminaries

2.1 Conditional Rewriting

We define the formalism and operational use of a language for expressing data type and function specifications [13, 8].

Definition 1 Equational-Inequational-Conditional Term Rewriting Systems (EI-CTRS) are finite sets of *rules* of the general form

$$[s_1 = t_1 \wedge \ldots \wedge s_n = t_n \wedge p_1 \neq q_1 \wedge \ldots \wedge p_m \neq q_m] \Rightarrow lhs \rightarrow rhs,$$

where *lhs* and *rhs* are two terms, and the antecedent is a conjunction of zero or more equations $s_i = t_i$ and negated equality literals $p_j \neq q_j$. Every variable occurring in each s_i, t_i, p_j, q_j and *rhs* must also occur in *lhs.*

Following the notation of [4], 'p/j' refers to the subterm of *p* at position j, and '$p[q]_j$' refers to the result of replacing p/j by *q* in *p.* For instance, when positions are described in Dewey decimal notation, $f(g(a, h(b, c)), d)/1.2$ is $h(b, c)$, and $f(g(a, h(b, c)), d)[m]_{1.2}$ is $f(g(a, m), d)$.

Besides matching and replacement, conditional rewriting requires checking that the antecedent of a rule holds. The basic idea underlying the definition of El-rewriting is to conclude that two terms are equal if they have converging reduction sequences, and not equal otherwise (for ground terms). Not surprisingly, the attempt to find a valley-proof for an equality does not always terminate. But if reduction sequences from two ground terms p, q do terminate without converging, then we can assert that $p = q$ has no 'valley-proof using the given rewriting system.

Variables in different rules are first renamed to be distinct from each other and from those in the term to be El-reduced.

Definition 2 : A term m <u>El-Reduces</u> (or *El-Rewrites)* to n using an El-CTRS *R* (written $m \rightarrow_R n$) if R contains a rule $cond \Rightarrow lhs \rightarrow rhs$ and there is a substitution *a* matching *lhs* with a subter m/j of m, such that each of the following conditions hold:

1. Match and Replace: $(lhs)\sigma \cong m/j$ and $n \equiv m[(rhs)\sigma]_j$.
2. Demonstrable Convergence: For each equality $s_i = t_i \in$ *cond,* there is a common term to which $s_i\sigma$ and $t_i\sigma$ can be El-reduced in a finite number of steps.
3. Demonstrable Non-Convergence: For every negated equality literal $s_i \neq t_i \in$ *cond,* it can be demonstrated by finite El-rewriting sequences that $s_i\sigma$ and $t_i\sigma$ are ground terms with no common reduct. More precisely, the following conditions must hold:
   (i) $s_i\sigma$ and $t_i\sigma$ are ground terms;
   (ii) all El-rewriting sequences from $s_i\sigma$ and $t_i\sigma$ terminate; and
   (iii) the set of all reducts of $s_i\sigma$ is disjoint from that of $t_i\sigma$.

For a more detailed description of El-rewriting with examples, see [14].

2.2 Fixpoint Semantics

Definition 3 If / is a function (of one argument) whose domain and range are the same, then *S* is a fixpoint (or fixed point) of / whenever f(S) = 5. If the domain elements are partially ordered, / may have zero or more partially ordered fixpoints:

- a fixpoint *S* is minimal if there is no other fixpoint *T* such $T < S$ in the ordering;
- a minimal fixpoint *S* is the least fixpoint if $S \leq T$ for every fixpoint;
- a fixpoint *S* is maximal if there is no other fixpoint *T* such that $T > S$ in the ordering;
- two fixpoints S, *T* are compatible if they have a common upper bound which is a fixpoint,
$$\exists S'.(S \leq S') \wedge (T \leq S') \wedge (f(S') = S');$$
- a fixpoint is intrinsic [10] (or optimal [12]) iff it is compatible with every fixpoint of /.

In the fixpoint semantics approach, the 'meaning* of a program is considered to be the least fixpoint of a function/relation which represents the behavior of the program on some input. The following fixpoint semantics can be suggested for conditional rewriting, as in [8]. A function ΨR is associated with each CTRS *R* such that if *S* is a binary relation, and S^* is its reflexive transitive closure, then $\Psi_R(S)$ is a binary relation which is the set of all two-tuples $\langle p, q\rangle$ such that for some rule $[s_1 = t_1 \wedge \ldots \wedge s_n = t_n] \wedge [p_1 \neq q_1 \wedge \ldots \wedge p_m \neq q_m] \Rightarrow l \rightarrow r$ in R, we have $\exists k, \exists \sigma.\ p/k \equiv l\sigma,\ q \equiv p[r\sigma]_k,\ \forall i, \exists r_i.\langle s_i\sigma, r_i\rangle \in S^*$ and $\langle t_i\sigma, r_i\rangle \in S^*$, <u>and</u> $\forall j$, there is no r_j such that $\langle p_j\sigma, r_j\rangle \in S^*$, $\langle q_j\sigma, r_j\rangle \in S^*$. In this approach, the 'meaning' of rewriting with R is identified with the least fixed point of Ψ_R, <u>if</u> it exists.

But such a least fixed point exists only when the CTRS satisfies certain stringent (and undecidable) conditions that ensure the decidability and finite termination of all rewrite sequences. The following is an example of a CTRS R such that Ψ_R has no fixpoint according to the above definition: $\{a \neq b \Rightarrow a \rightarrow b\}$. Note that $\{\langle a, b\rangle\}$ is not a fixpoint because the antecedent $a \neq b$ of the conditional rule is such that $\exists x.\langle a, x\rangle$ and $\langle b, x\rangle$ are members of $\{\langle a, a\rangle, \langle a, b\rangle, \langle b, b\rangle\}$, the reflexive symmetric closure of $\{\langle a, b\rangle\}$.

The problems with the above approach can be pinpointed to the following reasons:

1. The use of a two-valued logic precludes distinguishing between cases when we know that a rewrite doesn't occur, and cases when we do not know whether a rewrite can occur, particularly cases involving non-terminating reductions arising from terms in the antecedent of an invoked rewrite rule.
2. The relation Ψ_R is not 'monotone'.

Definition 4 A mapping Φ is monotone iff for all the elements in the (partially ordered) domain, $S \leq T$ implies $\Phi(S) \leq \Phi(T)$.

2.3 Kleene's 3-valued logic

Kleene [9] presented a three-valued logic, partly motivated by the desire to give truth-value meanings to partial recursive functions. The logic lends itself easily to explain non-deterministic and infinite computations. Kleene's third truth value represents the ***indeterminate*** or ***unknown*** nature of statements. The truth, falsehood, or indeterminacy of statements may be captured by using Smullyan's notation of 'signed statements':

Definition 5 If ψ is a statement, $T\psi$ is a signed statement which is to be understood as asserting "ψ is true". Similarly, $F\psi$ is a signed statement which is to be understood as asserting "ψ is false", which is contradictory to $T\psi$. A set of signed statements is consistent if it does not contain the pair $T\psi, F\psi$ for any statement ψ.

Implicitly, if a set of signed statements contains neither $T\psi$ nor $F\psi$ for some formula ψ, it is understood that ψ is indeterminate or has Kleene's third truth value.

'Saturated' consistent sets of signed statements are intended to serve as models for logic programs. To summarize a lengthy definition in [5], a set S of signed statements is saturated iff it contains the intuitive consequences of the members of S; *e.g.*,
- $T(X \wedge Y) \in S$ implies $T(X) \in S$ and $T(Y) \in S$,
- $F(\forall x.P(x)) \in S$ implies $F(P(t)) \in S$ for some closed term t,
- $FX \in S$ implies $T(\neg X) \in S$,
- $FX \in S$ and $FY \in S$ imply $F(X \vee Y) \in S$,

et cetera.

3 New Semantics

Unlike Prolog-style logic programs, the operational use of CTRS involves iterated rewrites ensuing from the antecedents of conditional rules. So the definition of conditional rewriting recursively involves that of iterated rewriting. A careful definition of what it means for "$p \rightarrow_R^* q$" to be "*false*" is needed, since this is needed when evaluating negated equality literals in the antecedents of rules.

Let S be a consistent set of signed two-tuples; intuitively, S is a potential description of a rewrite relation. $T\langle p,q\rangle \in S$ is an abbreviation for $T(Rewrites(p,q))$, intended to mean that p rewrites to q; and similarly $F\langle p,q\rangle \in S$ means p is known not to rewrite to q. If neither of these is present in S, that means the reduction from p to q is not known to be true or false, i.e., "$p \rightarrow q$" has the third truth value.

A new set of signed statements S^*, describing the iteration of rewrites in S, is defined as follows. For convenience, let $S^* = S_T^* \uplus S_F^*$, distinguishing sets of statements with prefixes T and F, respectively. S_T^* is just the reflexive transitive closure of the true statements in S, which is the least set satisfying the following three conditions.

1. $\forall a.\ T\langle a,a\rangle \in S_T^*$,
2. $\forall a,b$, if $T\langle a,b\rangle \in S$, then $T\langle a,b\rangle \in S_T^*$, and
3. $\forall a,b,c$, if $T\langle a,b\rangle \in S_T^*$ and $T\langle b,c\rangle \in S_T^*$, then $T\langle a,c\rangle \in S_T^*$.

The construction of S_F^* is slightly more complicated; note that $F\langle a,b\rangle \in S$ does not necessarily imply that $F\langle a,b\rangle \in S^*$: a rewrite sequence from a to b might exist through some other terms, e.g., if $\{F\langle a,b\rangle,\ T\langle a,c\rangle,\ T\langle c,b\rangle\} \subseteq S$, then $T\langle a,b\rangle \in S^*$.

To conclude the absence of rewrite sequences, there should be no intermediate term from which a reduction sequence can occur. It is safe to assert that $(p \rightarrow_R^* q)$ is false iff it can be determined that there is no rewrite sequence of finite length from p to q. This aspect can be described by an iterative construction as follows, where S_F^i represents 2 tuples among which there is no reduction sequence of length $\leq i$.

$S_F^0 = \{F\langle a,b\rangle | a \not\equiv b\}$
$S_F^1 = \{F\langle a,b\rangle | F\langle a,b\rangle \in S_F^0 \wedge F\langle a,b\rangle \in S\}$
$S_F^{i+1} = \{F\langle a,b\rangle | F\langle a,b\rangle \in S_F^i \wedge$
$\qquad\qquad \forall y[F\langle y,b\rangle \in S_F^i \vee F\langle a,y\rangle \in S]\}$
$S_F^* = \lim_{i\rightarrow\infty} S_F^i$

Note: $\forall i, S_F^{i+1} \subseteq S_F^i$, thus S_F^* is the greatest lower bound of a chain of sets in the subset ordering.

Example 1 Let the language contain only the constants a,b,c,d.

$$\text{Let } R = \left\{ \begin{array}{l} a \rightarrow b \\ b \rightarrow c \\ c \rightarrow d \end{array} \right\}.$$

$$\begin{aligned} \text{Let } S = \ & \{F\langle a,a\rangle, T\langle a,b\rangle, , F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,b\rangle, T\langle b,c\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, T\langle c,d\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\} \end{aligned}$$

$$\begin{aligned} \text{Then } S_T^* = \ & \{T\langle a,a\rangle, T\langle a,b\rangle, T\langle a,c\rangle, T\langle a,d\rangle, \\ & T\langle b,b\rangle, T\langle b,c\rangle, T\langle b,d\rangle, \\ & T\langle c,c\rangle, T\langle c,d\rangle, \\ & T\langle d,d\rangle\} \end{aligned}$$

$$\begin{aligned} S_F^0 = \ & \{F\langle a,b\rangle, F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,c\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,d\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \\ S_F^1 = \ & \{F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \\ S_F^2 = \ & \{F\langle a,d\rangle, F\langle b,a\rangle, F\langle c,a\rangle, F\langle c,b\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \end{aligned}$$

$S_F^* = S_F^3 = \{F\langle b,a\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\}$

Example 2 Let the language contain only the constants a,b,c,d.

$$\text{Let } R = \left\{ \begin{array}{l} a \rightarrow b \\ b \rightarrow c \\ c = d \Rightarrow c \rightarrow d \end{array} \right\}.$$

$$\begin{aligned} \text{Let } S = \ & \{F\langle a,a\rangle, T\langle a,b\rangle, F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,b\rangle, T\langle b,c\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\} \end{aligned}$$

$$\begin{aligned} \text{Then } S_T^* = \ & \{T\langle a,a\rangle, T\langle a,b\rangle, T\langle a,c\rangle, \\ & T\langle b,b\rangle, T\langle b,c\rangle, T\langle c,c\rangle, T\langle d,d\rangle\} \end{aligned}$$

$$\begin{aligned} S_F^0 = \ & \{F\langle a,b\rangle, F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,c\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,d\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \end{aligned}$$

$$\begin{aligned} S_F^1 = \ & \{F\langle a,c\rangle, F\langle a,d\rangle, \\ & F\langle b,a\rangle, F\langle b,d\rangle, \\ & F\langle c,a\rangle, F\langle c,b\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \end{aligned}$$

$$\begin{aligned} S_F^2 = \ & \{F\langle a,d\rangle, F\langle b,a\rangle, F\langle c,a\rangle, F\langle c,b\rangle, \\ & F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\} \end{aligned}$$

$S_F^* = S_F^3 = \{F\langle b,a\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle\}$

In this case, $S^* = S_T^* \cup S_F^*$ contains neither $T\langle c,d\rangle$ nor $F\langle c,d\rangle$.

Lemma 1 If S is consistent, then S^* is also consistent.

Lemma 2 The '$*$' operator is monotone, i.e., if $S_1 \subseteq S_2$, then $S_1^* \subseteq S_2^*$.

Definition 6 When R is an EI-CTRS, a mapping Φ_R from sets of signed two-tuples to sets of signed two-tuples is defined as follows:
$\Phi_R(S)$ is the smallest relation such that

- $T\langle a, b\rangle \in \Phi_R(S)$ if R contains a rule
$$\bigwedge_{i=1}^{n} s_i = t_i \bigwedge_{j=1}^{m} p_j \neq q_j \Rightarrow l \rightarrow r$$
such that for some position k and some substitution σ, we have
$$\left[\begin{array}{l} a/k \equiv l\sigma \ \wedge \ b \equiv a[r\sigma]_k \wedge \\ \ [\forall i \cdot \exists r_i \cdot T\langle s_i\sigma, r_i\rangle \in S^* \wedge T\langle t_i\sigma, r_i\rangle \in S^*] \wedge \\ \ [\forall j \cdot \forall r_j \cdot F\langle p_j\sigma, r_j\rangle \in S^* \vee F\langle q_j\sigma, r_j\rangle \in S^*] \end{array}\right]$$

- $F\langle a, b\rangle \in \Phi_R(S)$ if for every rule $[\bigwedge_i s_i = t_i \bigwedge_j p_j \neq q_j \Rightarrow l \rightarrow r]$ in R, and for every k, σ such that $(a/k \equiv l\sigma \wedge b \equiv a[r\sigma]_k)$, we have
$$\left[\begin{array}{l} (\exists i \cdot \forall r_i \cdot F\langle s_i\sigma, r_i\rangle \in S^* \vee F\langle t_i\sigma, r_i\rangle \in S^*) \\ \vee(\exists j \exists r_j \cdot T\langle p_j\sigma, r_j\rangle \in S^* \wedge T\langle q_j\sigma, r_j\rangle \in S^*) \end{array}\right].$$

Lemma 3 If S is consistent, then $\Phi_R(S)$ is consistent.

Theorem 1 Φ_R is monotone, i.e.,
$$S_1 \subseteq S_2 \Rightarrow \Phi_R(S_1) \subseteq \Phi_R(S_2).$$

Proof: Assume $S_1 \subseteq S_2$.
If $T\langle a, b\rangle \in \Phi_R(S_1)$, then R contains a rule
$$\bigwedge_{i=1}^{n} s_i = t_i \bigwedge_{j=1}^{m} p_j \neq q_j \Rightarrow l \rightarrow r$$
such that for some k, σ, we have
$$\left[\begin{array}{ll} & a/k \equiv l\sigma \ \wedge \ b \equiv a[r\sigma]_k \\ \wedge & [\forall i \cdot \exists r_i \cdot T\langle s_i\sigma, r_i\rangle \in S_1^* \wedge T\langle t_i\sigma, r_i\rangle \in S_1^*] \\ \wedge & [\forall j \cdot \forall r_j \cdot F\langle p_j\sigma, r_j\rangle \in S_1^* \vee F\langle q_j\sigma, r_j\rangle \in S_1^*] \end{array}\right].$$

Since $S_1^* \subseteq S_2^*$ (by lemma 2), we have
$$\begin{array}{ll} & \forall i \cdot \exists r_i \cdot T\langle s_i\sigma, r_i\rangle \in S_2^* \wedge T\langle t_i\sigma, r_i\rangle \in S_2^*, \\ \text{and} & \forall j \forall r_j \cdot F\langle p_j\sigma, r_j\rangle \in S_2^* \vee F\langle q_j\sigma, r_j\rangle \in S_2^*. \end{array}$$

Therefore, $T\langle a, b\rangle \in \Phi_R(S_2)$, if $T\langle a, b\rangle \in \Phi_R(S_1)$.

Similarly, it can be shown that if $F\langle a, b\rangle \in \Phi_R(S_1)$, then $F\langle a, b\rangle \in \Phi_R(S_2)$.

Hence $\Phi_R(S_1) \subseteq \Phi_R(S_2)$. □

Key Observation: The fixpoints of Φ_R describe the semantics of conditional rewriting with an EI-CTRS R; particularly important are the least fixpoint and the largest intrinsic fixpoint.

We now investigate the fixpoints of the monotone relation Φ_R corresponding to each rewrite system R.

Theorem 2 Let R be any EI-CTRS, and Φ_R be as defined above. Then:

1. Φ_R has maximal fixpoints.
2. Φ_R has a smallest fixpoint.
3. Φ_R has a largest intrinsic fixpoint, which is a subset of $\bigcap\{$maximal fixpoints of $\Phi_R\}$.
4. The smallest fixpoint of Φ_R above the empty set is intrinsic.

Proof: Let D be the collection of all consistent sets of signed statements, ordered by the subset relation. D has a smallest member $\{\,\}$, since the empty set is consistent. Every chain $S_1 \subseteq S_2 \subseteq \cdots$ of elements of D has an upper bound $\bigcup_i \lim S_i$. Also, every nonempty set having an upper bound has a least upper bound. Since Φ_R is monotone on D, all the premises of theorem 2.2 in [6] are satisfied, and the conclusions stated in this theorem directly follow. □

A smallest fixpoint can be constructed using the following transfinite sequence of consistent sets of signed statements.

Definition 7 Let Φ_R be as defined earlier, for any EI-CTRS R.
$$\begin{array}{rl} A_0 & = \{\,\} \\ A_{i+1} & = \Phi_R(A_i) \\ A_\lambda & = \bigcup\{A_\alpha | \alpha < \lambda\} \ \text{ for limit ordinals } \lambda \end{array}$$

By transfinite induction, it can be shown that this is a weakly increasing sequence, i.e., $A_\alpha \leq A_{\alpha+1}$ for all α. But the sequence cannot be strongly increasing, i.e., it is not possible that $\forall\alpha \cdot A_\alpha < A_{\alpha+1}$, since a sequence of consistent sets cannot have as many members as there are ordinals. Hence, for some α, we must have $A_\alpha = \Phi_R(A_\alpha)$, i.e., some member of the sequence is a fixpoint of Φ_R.

4 Examples

The following are some examples which illustrate the application of the new fixpoint semantics. Some of the CTRS's considered here cannot be handled adequately by previously given semantics, because they do not satisfy the conditions for termination, and are not constructor-based specifications. In each case, we assume that the only symbols in the language are those that appear in the rules of the rewriting system being considered. For each CTRS R, we begin with candidates for fixpoints which are supersets of 'Z', defined as
$$\begin{array}{l} \{F\langle p, q\rangle | \text{ there is no rule } (C \Rightarrow l \rightarrow r) \in R \\ \qquad\qquad \text{such that } \exists k, \sigma . l\sigma \equiv p/k \wedge q \equiv p[r\sigma]_k\}. \end{array}$$

The rationale is that for any S, $\Phi_R(S)$ will always contain $F\langle p, q\rangle$, for those pairs of terms p, q such that no rule in R can possibly reduce p to q.

Example 3 Let R_1 be $\{a = b \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a, a\rangle, F\langle b, a\rangle, F\langle b, b\rangle\}$. Candidates for the fixpoints of Φ_{R_1} are: Z, $Z \cup \{T\langle a, b\rangle\}$, and $Z \cup \{F\langle a, b\rangle\}$. Each of these is a fixpoint; Z itself is the least fixpoint, whereas the other two are maximal fixpoints, with each of which Z is compatible. Hence Z is also the greatest intrinsic fixpoint of Φ_{R_1}.

Example 4 Let R_2 be $\{a \neq b \Rightarrow a \rightarrow b\}$.

Again, let $Z = \{F\langle a,a\rangle, F\langle b,a\rangle, F\langle b,b\rangle\}$. Candidates for the fixpoints of Φ_{R_2} are also again Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. Of these, only Z is a fixpoint: note that $\Phi_{R_2}(Z \cup \{T\langle a,b\rangle\})$ will contain $F\langle a,b\rangle$, and $\Phi_{R_2}(Z \cup \{F\langle a,b\rangle\})$ will contain $T\langle a,b\rangle$. Hence the others are not fixpoints of Φ_{R_2}.

Example 5 Let R_3 be $\{a \neq c \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle\}$. Candidates for the fixpoints of Φ_{R_3} are Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. Of these, Z is a fixpoint, because the absence of $T\langle a,b\rangle$ and $F\langle a,b\rangle$ from Z implies that $\Phi_{R_3}(Z)$ will contain neither $T\langle a,b\rangle$ nor $F\langle a,b\rangle$. Also, $Z \cup \{T\langle a,b\rangle\}$ is a fixpoint since $\Phi_{R_2}(Z \cup \{T\langle a,b\rangle\})$ will contain $T\langle a,b\rangle$. But $\Phi_{R_2}(Z \cup \{F\langle a,b\rangle\})$ will contain $T\langle a,b\rangle$, hence the third candidate is not a fixpoint. Hence Z is the least fixpoint, whereas $Z \cup \{T\langle a,b\rangle\}$ is the only maximal fixpoint, which is hence the greatest intrinsic fixpoint.

Example 6 Let R_4 be $\{a = c \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle\}$ again. Candidates for the fixpoints of Φ_{R_4} are Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. Of these, Z is again a (least) fixpoint. $Z \cup \{F\langle a,b\rangle\}$ is also a fixpoint, but not $Z \cup \{T\langle a,b\rangle\}$. Thus $Z \cup \{F\langle a,b\rangle\}$ is the greatest intrinsic fixpoint.

Example 7 Let R_5 be $\{c \neq d \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle a,d\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle b,d\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, F\langle c,d\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\}$. Candidates for the fixpoints of Φ_{R_5} are Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. This time, Z is <u>not</u> a fixpoint because $\Phi_{R_5}(Z)$ contains $T\langle a,b\rangle$. For the same reason, $Z \cup \{F\langle a,b\rangle\}$ is also not a fixpoint. The only fixpoint is $Z \cup \{T\langle a,b\rangle\}$.

Example 8

Let R_6 be $\{c \neq d \Rightarrow a \rightarrow b,\ a \neq b \Rightarrow c \rightarrow d\}$. Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle a,d\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle b,d\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\}$. Candidates for the fixpoints of Φ_{R_6} are those supersets of Z which contain at most one of $T\langle a,b\rangle$, $F\langle a,b\rangle$, and also at most one of $T\langle c,d\rangle$, $F\langle c,d\rangle$. Of these, Z itself is a (least) fixpoint since it does not contain signed tuples which would enable either of the rewrite rules in R_6 to be activated. If a fixpoint contains $F\langle a,b\rangle$, then it must also contain $T\langle c,d\rangle$ since the second rewrite rule is activated; indeed, $Z \cup \{F\langle a,b\rangle, T\langle c,d\rangle\}$ is a (maximal) fixpoint. Similarly, $Z \cup \{F\langle c,d\rangle, T\langle a,b\rangle\}$ is also a (maximal) fixpoint. These maximal fixpoints are not mutually compatible, and Z is a fixpoint compatible with each of these maximal fixpoints. Hence Z is the least as well as the largest intrinsic fixpoint.

It can be shown that the other candidates are not fixpoints. For instance, $\Phi_{R_6}(Z \cup \{T\langle a,b\rangle\})$ does not contain $T\langle a,b\rangle$, since the antecedent of the first rewrite rule is not enabled: $F\langle c,d\rangle \notin Z \cup \{T\langle a,b\rangle\}$

Example 9

Let R_7 be $\{a = b \Rightarrow a \rightarrow b,\quad a \neq b \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a,a\rangle, F\langle b,a\rangle, F\langle b,b\rangle\}$. Candidates for the fixpoints of Φ_{R_7} are Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. Of these, Z is a (least) fixpoint. Also $Z \cup \{T\langle a,b\rangle\}$ is a fixpoint, but not $Z \cup \{F\langle a,b\rangle\}$. Hence $Z \cup \{T\langle a,b\rangle\}$ is the greatest intrinsic fixpoint. Note that a 'disjunctive' conditional rewriting mechanism which examines the antecedents of multiple rules achieves computation of the greatest intrinsic fixpoint in this case.

Example 10 Let R_8 be $\{c \neq d \Rightarrow a \rightarrow b,$
$a \neq b \Rightarrow a \rightarrow b\}$.

Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle a,d\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle b,d\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, F\langle c,d\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\}$. Candidates for the fixpoints of Φ_{R_8} are Z, $Z \cup \{T\langle a,b\rangle\}$, and $Z \cup \{F\langle a,b\rangle\}$. Since Z contains $F\langle c,x\rangle$ and $F\langle d,x\rangle$ for every x, every superset of Z which is a fixpoint must contain $T\langle a,b\rangle$ since the first rewrite rule is enabled. Hence $Z \cup \{T\langle a,b\rangle\}$ is the only fixpoint.

Example 11 Let R_9 be $\{c \neq d \Rightarrow a \rightarrow b,$ $a \neq b \Rightarrow c \rightarrow d,\quad a = b \Rightarrow c \rightarrow d\}$. Let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle a,d\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle b,d\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\}$. Candidates for the fixpoints of Φ_{R_9} are those supersets of Z which contain at most one of $T\langle a,b\rangle$, $F\langle a,b\rangle$, and also at most one of $T\langle c,d\rangle$, $F\langle c,d\rangle$. Of these, Z itself is a (least) fixpoint since it does not contain signed tuples which would enable either of the rewrite rules in R_9. If a fixpoint contains $F\langle a,b\rangle$, then it must also contain $T\langle c,d\rangle$ since the second rewrite rule is activated; indeed, $Z \cup \{F\langle a,b\rangle, T\langle c,d\rangle\}$ is a (maximal) fixpoint.

There are no other fixpoints: every superset of Z which contains $F\langle a,b\rangle$ or $T\langle a,b\rangle$ must also contain $T\langle c,d\rangle$ if it is a fixpoint, since the second or third rewrite rule is enabled. But every superset of Z which contains $T\langle c,d\rangle$ must also contain $F\langle a,b\rangle$ if it is a fixpoint, since the antecedent of the first rewrite rule is falsified. And every superset of Z which contains $F\langle c,d\rangle$ must also contain $T\langle a,b\rangle$ if it is a fixpoint, since the first rewrite rule is enabled.

Example 12

Let R_{10} be $\{c \neq d \Rightarrow a \rightarrow b,\quad a \neq b \Rightarrow c \rightarrow d,$ $a = b \Rightarrow c \rightarrow d,\quad c = d \Rightarrow a \rightarrow b\}$. Again, let $Z = \{F\langle a,a\rangle, F\langle a,c\rangle, F\langle a,d\rangle, F\langle b,a\rangle, F\langle b,b\rangle, F\langle b,c\rangle, F\langle b,d\rangle, F\langle c,a\rangle, F\langle c,b\rangle, F\langle c,c\rangle, F\langle d,a\rangle, F\langle d,b\rangle, F\langle d,c\rangle, F\langle d,d\rangle\}$. We again examine various consistent supersets of Z which are potential candidates for the fixpoints of $\Phi_{R_{10}}$. Z itself is a (least) fixpoint, as in the previous example. If one of the other candidates contains $T\langle a,b\rangle$, then the third rewrite rule dictates that it must also contain $T\langle c,d\rangle$ if it is a fixpoint (and conversely). Indeed, $Z \cup \{T\langle a,b\rangle, T\langle c,d\rangle\}$ is a (maximal) fixpoint. Candidates which contain $F\langle a,b\rangle$ (or $F\langle c,d\rangle$) are not fixpoints, because the second (or first) rewrite rule is then enabled, generating $T\langle c,d\rangle$ (or $T\langle a,b\rangle$, respectively), which in turn implies that the candidate must contain $T\langle a,b\rangle$ (or $T\langle c,d\rangle$, respectively) making it inconsistent. Hence there are no other fixpoints.

5 Conclusions

We have investigated the fixpoint semantics of conditional term rewriting systems with negation. Two-valued semantics does not ascribe a meaning to CTRS's that do not satisfy useful properties such as termination. We have shown that a three-valued approach used by Fitting [5] for Prolog-style logic programs is applicable in this context. A monotone operator is developed, whose fixpoints describe the semantics of conditional rewriting. Several examples illustrate this semantics for 'troublesome' rewrite systems which could not be handled easily by previous approaches. This work supports the contention that results achieved in research on Prolog-style logic programming can be useful in the context of conditional term rewriting.

We have hesitated to say whether it is the least fixpoint or the greatest intrinsic fixpoint which better describes the semantics of the EI-CTRS. The examples may motivate a preference for one or the other. The operational mechanism described in [13] and [8] computed members of the least fixpoint. To compute the greatest intrinsic fixpoint, we need a different operational mechanism which uses "disjunctive rewriting" [15] (cf. example 9) as well as a mechanism which returns failure in some cases when naive evaluation of the antecedent leads to non-termination (cf. example 5). The formulation of such a rewriting mechanism, which computes precisely the greatest intrinsic fixpoint, is an issue for future work. In non-controversial cases, when termination requirements are satisfied, the least fixpoint and the greatest intrinsic fixpoint coincide (cf. examples 3, 4, 7). and are essentially equivalent to the semantics given in previous work for such well-behaved rewrite systems.

Acknowledgements

The author thanks Melvin Fitting and anonymous referees for their comments on the paper.

References

[1] H.Aida and J.Meseguer. "Getting Rid of Conditional Equations". Proc. Second Int'l. Workshop on Conditional and Typed Term Rewriting Systems, Montreal, June 1990.

[2] K.R.Apt and M.H.Van Emden. "Contributions to the Theory of Logic Programming". *Journal of the Assoc. Computing Mach.,* vol.29, pp. 841-862, 1982.

[3] K.L.Clark. "Negation as Failure" in H.Gallaire and J.Minker (eds): "Logic and Data Bases". Plenum Press, N.Y., 1978.

[4] N.Dershowitz and J.-P.Jouannaud. "Rewrite Systems" in "Handbook of Theoretical Comp. Science". Academic Press, 1989.

[5] M.Fitting. "A Kripke-Kleene Semantics for Logic Programs". *Journal of Logic Programming,* vol.4, pp. 295-312, 1985.

[6] M.Fitting. "Notes on the Mathematical Aspects of Kripke's Theory of Truth". *Notre Dame Journal of Formal Logic,* vol.27, no.l, pp.75-88, Jan. 1986.

[7] S. Kaplan. "Simplifying Conditional Term Rewriting Systems: Unification, Termination and Confluence". Rapport de Recherche no. 194, Universite de Paris-Sud, Nov. 1984.

[8] S.Kaplan. "Positive/Negative Conditional Rewriting". Proc. First Int'l. Workshop on Conditional Term Rewriting Systems, Paris, Springer-Verlag LNCS 308, 1987.

[9] S.C.Kleene. "Introduction to Metamathematics". Van Nostrand, New York, 1952.

[10] S.Kripke. "Outline of a Theory of Truth". *J. Philosophy* vol.72, pp.690-716, 1975.

[11] R.Kowalski. "Predicate Logic as a Programming Language". 1FIP Info. Processing, North-Holland, pp.569-574, 1974.

[12] Z.Manna and A.Shamir. "The Theoretical Aspect of the Optimal Fixed Point". *SIAM J. of Computing,* vol.5, pp.414-426, 1976.

[13] C.K.Mohan and M.K.Srivas. "Conditional Specifications with Inequational Assumptions". Proc. First Int'l. Workshop on Conditional Term Rewriting Systems, Paris, Springer-Verlag LNCS 308, 1987.

[14] C.K.Mohan and M.K.Srivas. "Negation in Conditional Term Rewriting", in R.Wilkerson (ed.): "Advances in Logic Programming and Automated Reasoning" (to appear), Ablex Press, 1991.

[15] D.Plaisted. "Confluence and Reduction Properties of Conditional Term Rewriting Systems". Manuscript, 1985.

[16] A.Tarski. "A lattice-theoretical fixpoint theorem and its applications". *Pacific J. Math.,* vol.5, pp.285-309, 1955.