A Semantic Criterion for Proving Infeasibility in Conditional Rewriting

Salvador Lucas  Raúl Gutiérrez

DSIC, Universitat Politècnica de València, Spain

6th International Workshop on Confluence, IWC 2017
**Conditional Term Rewriting Systems** (CTRSs) consist of rules of the following general form:

\[
\ell \rightarrow r \leftarrow c
\]

where the *conditional part* \(c\) is used as a *test* for the application of (an instance of) \(\ell\) and \(r\) in any rewriting step \(\sigma(\ell) \rightarrow \sigma(r)\), i.e., \(\sigma(c)\) must be ‘satisfied’.

**Infeasible rules**

Some conditional parts *cannot* be used with any substitution!

\[
a \rightarrow b \leftarrow a \rightarrow c
\]

In this case, the *test* specified by the conditional part is a *reachability test* \(a \rightarrow^* c\) (as for *oriented* CTRSs) which never succeeds.
In the analysis of computational properties of CTRSs $\mathcal{R}$, the conditional part of the rules is often used to investigate them.

1. Confluence is analyzed by means of *conditional critical pairs* $\langle s, t \rangle \leftarrow c$, where $c$ is obtained from two rules $\ell \rightarrow r \leftarrow d$ and $\ell' \rightarrow r' \leftarrow d'$ in $\mathcal{R}$.

2. Operational termination is analyzed by means of *conditional dependency pairs* $u \rightarrow v \leftarrow c$ where $c$ is obtained from the conditional part $d$ of a rule $\ell \rightarrow r \leftarrow d$ in $\mathcal{R}$.

**Feasibility tests**

In all these cases, the following question is relevant:

*Is there any substitution $\sigma$ such that $\sigma(c)$ holds using $\mathcal{R}$?*

Or, assuming a logic-based operational semantics for CTRSs:

*Is there any substitution $\sigma$ such that $\mathcal{R} \vdash \sigma(c)$?*
Motivation

A negative answer to this question, i.e., ensuring that $\mathcal{R} \vdash \sigma(c)$ does not hold for any substitution $\sigma$ (i.e., proving $c$ infeasible), can be useful:

- Rules $\ell \rightarrow r \iff c \in \mathcal{R}$ will never be used when rewriting with $\mathcal{R}$.
- Joinability of $s$ and $t$ in conditional critical pairs $\langle s, t \rangle \iff c$ does not matter in the analysis of confluence of $\mathcal{R}$.
- Conditional dependency pairs $u \rightarrow v \iff c$ can be safely discarded in the analysis of operational termination of $\mathcal{R}$.

Analyzing infeasibility

How to prove the conditional part of a conditional rule/critical pair/dependency pair infeasible?

This problem has been addressed before; several approaches have been proposed.

Our contribution exploits the logical approach sketched above.
In the following, we focus on oriented CTRSs $\mathcal{R}$, with rules

$$\ell \rightarrow r \iff s_1 \rightarrow t_1, \ldots, s_n \rightarrow t_n$$

whose operational semantics is given by the following inference system:

\[(\text{Rf})\quad x \rightarrow^* x\]
\[(\text{C})\quad x_i \rightarrow y_i, \quad f(x_1, \ldots, x_i, \ldots, x_k) \rightarrow f(x_1, \ldots, y_i, \ldots, x_k)\]
\[(\text{for all } f \in \mathcal{F} \text{ and } 1 \leq i \leq k = \text{arity}(f))\]

\[(\text{T})\quad x \rightarrow z, \quad z \rightarrow^* y\quad \rightarrow\]
\[
\frac{x \rightarrow z}{x \rightarrow^* y}
\]

\[(\text{Rp})\quad s_1 \rightarrow^* t_1, \ldots, s_n \rightarrow^* t_n\]
\[
\frac{\ell \rightarrow r}{\text{for all } \ell \rightarrow r \iff s_1 \rightarrow t_1 \cdots s_n \rightarrow t_n \in \mathcal{R}}
\]

**Definition**

Let $\mathcal{R}$ be a CTRS. A sequence $s_1 \rightarrow^* t_1, \ldots, s_n \rightarrow^* t_n$, where $s_i$ and $t_i$ are terms for all $1 \leq i \leq n$ is called a **feasibility sequence**. It is called $\mathcal{R}$-**feasible** if there is a substitution $\sigma$ such that for all $1 \leq i \leq n$, $\sigma(s_i) \rightarrow^*_\mathcal{R} \sigma(t_i)$. Otherwise, it is called $\mathcal{R}$-**infeasible**.
The first-order theory $\overline{\mathcal{R}}$ for a CTRS $\mathcal{R}$ is obtained by specializing $(C)$ and $(Rp)$ as above. Inference rules $\frac{B_1 \ldots B_n}{A}$ become universally quantified implications $B_1 \land \cdots \land B_n \Rightarrow A$.

**Example**

For the CTRS $\mathcal{R}$ (from [Giesl & Arts, AAECC’01])

\[
\begin{align*}
a & \rightarrow b \\
f(a) & \rightarrow b \\
g(x) & \rightarrow g(a) \iff f(x) \rightarrow x
\end{align*}
\]

its associated theory $\overline{\mathcal{R}}$ is

\[
\begin{align*}
(\forall x) \ x \rightarrow^* x \\
(\forall x, y, z) \ x \rightarrow y \land y \rightarrow^* z \Rightarrow x \rightarrow^* z \\
(\forall x, y) \ x \rightarrow y \Rightarrow f(x) \rightarrow f(y) \\
(\forall x, y) \ x \rightarrow y \Rightarrow g(x) \rightarrow g(y)
\end{align*}
\]
A structure (or interpretation) $\mathcal{A}$ for a first-order language gives meaning to function and predicate symbols as mappings and relations on a given set.

The usual interpretation of first-order formulas with respect to the structure is then considered.

A model for a set $S$ of first-order sentences (i.e., formulas without free variables) is a structure that makes them all true, written $\mathcal{A} \models S$.

**Proposition**

Let $\mathcal{R}$ be a CTRS, $s_1 \rightarrow^* t_1$, $\ldots$, $s_n \rightarrow^* t_n$ be a feasibility sequence, and $\mathcal{A}$ be a structure with nonempty domain. The sequence is $\mathcal{R}$-infeasible if

$$\mathcal{A} \models \mathcal{R} \cup \{\neg(\exists\vec{x}) \left(s_1 \rightarrow^* t_1 \land \cdots \land s_n \rightarrow^* t_n\right)\}$$

holds.
Proof by contradiction

1. $R$-feasibility implies that $\sigma(s_i) \rightarrow^*_R \sigma(t_i)$ (i.e., $\overline{R} \vdash \sigma(s_i) \rightarrow^* \sigma(t_i)$) for some substitution $\sigma$ and all $1 \leq i \leq n$. 
Proof by contradiction

1. \(\mathcal{R}\)-feasibility implies that \(\sigma(s_i) \Rightarrow^* \sigma(t_i)\) (i.e., \(\overline{\mathcal{R}} \vdash \sigma(s_i) \Rightarrow^* \sigma(t_i)\)) for some substitution \(\sigma\) and all \(1 \leq i \leq n\).

2. Since \(\mathcal{A} \models \overline{\mathcal{R}}\), by correctness, \(\mathcal{A} \models (\forall \vec{y}_i) \sigma(s_i) \Rightarrow^* \sigma(t_i)\) for all \(1 \leq i \leq n\), where \(\vec{y}_i\) are the variables in \(\text{Var}(\sigma(s_i)) \cup \text{Var}(\sigma(t_i))\).
Proof by contradiction

1. $\mathcal{R}$-feasibility implies that $\sigma(s_i) \rightarrow^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\mathcal{R} \vdash \sigma(s_i) \rightarrow^* \sigma(t_i)$) for some substitution $\sigma$ and all $1 \leq i \leq n$.

2. Since $\mathcal{A} \models \mathcal{R}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i) \; \sigma(s_i) \rightarrow^* \sigma(t_i)$ for all $1 \leq i \leq n$, where $\vec{y}_i$ are the variables in $\text{Var}(\sigma(s_i)) \cup \text{Var}(\sigma(t_i))$.

3. Thus, $\mathcal{A} \models (\forall \vec{y}) \; (\sigma(s_1) \rightarrow^* \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^* \sigma(t_n))$. 
Proof by contradiction

1. $\mathcal{R}$-feasibility implies that $\sigma(s_i) \rightarrow_{\mathcal{R}}^{*} \sigma(t_i)$ (i.e., $\mathcal{R} \vdash \sigma(s_i) \rightarrow^{*} \sigma(t_i)$) for some substitution $\sigma$ and all $1 \leq i \leq n$.

2. Since $\mathcal{A} \models \mathcal{R}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i) \sigma(s_i) \rightarrow^{*} \sigma(t_i)$ for all $1 \leq i \leq n$, where $\vec{y}_i$ are the variables in $\mathcal{V}ar(\sigma(s_i)) \cup \mathcal{V}ar(\sigma(t_i))$.

3. Thus, $\mathcal{A} \models (\forall \vec{y}) (\sigma(s_1) \rightarrow^{*} \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^{*} \sigma(t_n))$.

4. For all $\nu : \vec{y} \rightarrow \mathcal{A}$, $[\sigma(s_1) \rightarrow^{*} \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^{*} \sigma(t_n)]_{\nu}^{A}$, is true.
Proof by contradiction

1. $R$-feasibility implies that $\sigma(s_i) \rightarrow^{*}_R \sigma(t_i)$ (i.e., $\overline{R} \vdash \sigma(s_i) \rightarrow^{*} \sigma(t_i)$) for some substitution $\sigma$ and all $1 \leq i \leq n$.

2. Since $A \models \overline{R}$, by correctness, $A \models (\forall \vec{y}_i) \sigma(s_i) \rightarrow^{*} \sigma(t_i)$ for all $1 \leq i \leq n$, where $\vec{y}_i$ are the variables in $\text{Var}(\sigma(s_i)) \cup \text{Var}(\sigma(t_i))$.

3. Thus, $A \models (\forall \vec{y}) (\sigma(s_1) \rightarrow^{*} \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^{*} \sigma(t_n))$.

4. For all $\nu : \vec{y} \rightarrow A$, $[\sigma(s_1) \rightarrow^{*} \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^{*} \sigma(t_n)]^A_{\nu}$, is true.

5. Since $A$ has a nonempty domain, there is a valuation $\nu' : \vec{x} \rightarrow A$ given by $\nu'(x) = [\sigma(x)]^A_{\nu}$ for all variable $x$ in $\vec{x} = \bigcup_{i=1}^{n} \text{Var}(s_i) \cup \text{Var}(t_i)$, such that $[s_1 \rightarrow^{*} t_1 \land \cdots \land s_n \rightarrow^{*} t_n]^A_{\nu'}$ is true.
Proof by contradiction

1. \( \mathcal{R} \)-feasibility implies that \( \sigma(s_i) \rightarrow^*_\mathcal{R} \sigma(t_i) \) (i.e., \( \overline{\mathcal{R}} \vdash \sigma(s_i) \rightarrow^* \sigma(t_i) \)) for some substitution \( \sigma \) and all \( 1 \leq i \leq n \).

2. Since \( \mathcal{A} \models \overline{\mathcal{R}} \), by correctness, \( \mathcal{A} \models (\forall \vec{y}_i) \sigma(s_i) \rightarrow^* \sigma(t_i) \) for all \( 1 \leq i \leq n \), where \( \vec{y}_i \) are the variables in \( \text{Var}(\sigma(s_i)) \cup \text{Var}(\sigma(t_i)) \).

3. Thus, \( \mathcal{A} \models (\forall \vec{y}) (\sigma(s_1) \rightarrow^* \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^* \sigma(t_n)) \).

4. For all \( \nu : \vec{y} \rightarrow \mathcal{A} \), \( [\sigma(s_1) \rightarrow^* \sigma(t_1) \land \cdots \land \sigma(s_n) \rightarrow^* \sigma(t_n)]^A_{\nu} \), is true.

5. Since \( \mathcal{A} \) has a nonempty domain, there is a valuation \( \nu' : \vec{x} \rightarrow \mathcal{A} \) given by \( \nu'(x) = [\sigma(x)]^A_{\nu} \) for all variable \( x \) in \( \vec{x} = \bigcup_{i=1}^n \text{Var}(s_i) \cup \text{Var}(t_i) \), such that \( [s_1 \rightarrow^* t_1 \land \cdots \land s_n \rightarrow^* t_n]^A_{\nu'} \) is true.

6. This contradicts \( \mathcal{A} \models \neg(\exists \vec{x}) (s_1 \rightarrow^* t_1 \land \cdots \land s_n \rightarrow^* t_n) \).
The following structure $\mathcal{A}$ over $\mathbb{N} - \{0\}$:

- $a^\mathcal{A} = 1$
- $b^\mathcal{A} = 2$
- $f^\mathcal{A}(x) = x + 1$
- $g^\mathcal{A}(x) = 1$
- $x \rightarrow^\mathcal{A} y \iff y \geq x$
- $x (\rightarrow^*)^\mathcal{A} y \iff y \geq x$

is a model of $\overline{\mathcal{R}} \cup \{\neg(\exists x) f(x) \rightarrow^* x\}$ for our running CTRS $\mathcal{R}$.

**Automation**

This model has been automatically generated by using the tool AGES:

http://zenon.dsic.upv.es/ages/

Thus, rule

$$g(x) \rightarrow g(a) \iff f(x) \rightarrow x$$

is proved $\mathcal{R}$-infeasible.
The following CTRS $\mathcal{R}$ (Example 23 in [Sternagel & Sternagel, FSCD’16])

$$
g(x) \rightarrow f(x, x) \quad (1)\\
g(x) \rightarrow g(x) \iff g(x) \rightarrow f(a, b) \quad (2)
$$

has a conditional critical pair $f(x, x) \downarrow g(x) \iff g(x) \rightarrow f(a, b)$. The following structure $\mathcal{A}$ over the finite domain $\{0, 1\}$:

$$
a^\mathcal{A} = 1 \quad b^\mathcal{A} = 0 \quad f^\mathcal{A}(x, y) = \begin{cases} 
  x - y + 1 & \text{if } x \geq y \\
  y - x + 1 & \text{otherwise}
\end{cases}\\
g^\mathcal{A}(x) = 1 \quad x \rightarrow^\mathcal{A} y \iff x = y \quad x (\rightarrow^*)^\mathcal{A} y \iff x \geq y
$$

is a model $\overline{\mathcal{R}} \cup \{\neg (\exists x) \ g(x) \rightarrow^* f(a, b)\}$. The critical pair is infeasible.

In the FSCD’16 paper, this is proved by using unification tests together with a transformation. It is discussed that the alternative tree automata techniques investigated in the paper do not work for this example.
Proposition (Joinability and feasibility)

Let \( R \) be a CTRS, \( s, t \) be terms, and \( x \) be a fresh variable not occurring in \( s \) or \( t \). If \( s \) and \( t \) are joinable, then \( s \rightarrow^* x \), \( t \rightarrow^* x \) is \( R \)-feasible. If \( s \) and \( t \) are ground and \( s \rightarrow^* x \), \( t \rightarrow^* x \) is \( R \)-feasible, then \( s \) and \( t \) are joinable.

Consider the following CTRS \( R \) (Example 7.3.3 in Ohlebusch’s book):

\[
\begin{align*}
a & \rightarrow b \\
f(x) & \rightarrow c \iff x \rightarrow a
\end{align*}
\]

(3)
(4)

Although there is no critical pair, the system is not (locally) confluent because \( f(a) \rightarrow_R f(b) \) and \( f(a) \rightarrow_R c \) but \( c \) and \( f(b) \) are not joinable. The following structure \( A \) over \( \mathbb{N} \cup \{-1\} \):

\[
\begin{align*}
a^A &= 0 \\
b^A &= -1 \\
c^A &= 1 \\
f^A(x) &= x + 1
\end{align*}
\]

\[
x \rightarrow^A y \iff x = y \\
x \rightarrow^* y \iff x = y
\]

is a model of \( \mathcal{U}(R, f(b), c) \cup \{\neg(\exists x) (f(b) \rightarrow^* x \land c \rightarrow^* x)\} \), where \( \mathcal{U}(R, f(b), c) = \{(4)\} \) is the set of usable rules (from \( R \)) for \( f(b) \) and \( c \). Therefore, \( f(b) \) and \( c \) are proved non-joinable.
We have presented a semantic approach to prove infeasibility in conditional rewriting.

We could handle many examples coming from papers developing different specific techniques to deal with these problems.

We do not have a dedicated, fully automated ‘infeasibility’ checker yet.

Instead we just encode the problem we want to deal with (e.g., infeasibility of a critical pair, or rule; or non-joinability) as an specific infeasibility sequence and then use AGES to find a model.

**Future work**

Improving automation, and the connection of AGES as a backend for other (confluence) tools are interesting subjects for future work.
This paper defines feasibility problems as (instantiated) reachability problems, which corresponds to the use *oriented CTRSs*.

As shown in [Lucas, LOPSTR 2017], the current treatment can be generalized to deal with more general notions of CTRSs, with

- many-sorted signatures,
- alternative satisfiability notions for the conditions (e.g., joinability), or
- more general components there (e.g., memberships).

**Future work**

The use of our semantic techniques in proofs of confluence of Maude programs is also an interesting subject for future work.
Thanks!