A Semantic Criterion for Proving Infeasibility in Conditional Rewriting

Salvador Lucas Raúl Gutiérrez

DSIC, Universitat Politècnica de València, Spain

6^{th} INTERNATIONAL WORKSHOP ON CONFLUENCE, IWC 2017

Conditional Term Rewriting Systems (CTRSs) consist of rules of the following general form:

$$\ell \to r \Leftarrow c$$

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

Infeasible rules

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

$$a \to b \Leftarrow a \to c$$

In this case, the *test* specified by the conditional part is a *reachability test* $a \to^* c$ (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 $\ell \to r \Leftarrow d$ and $\ell' \to r' \Leftarrow d'$ in $\mathcal{R}$.
2. Operational termination is analyzed by means of *conditional dependency pairs* $u \to v \Leftarrow c$ where c is obtained from the conditional part d of a rule $\ell \to r \Leftarrow d$ in $\mathcal{R}$.

Feasibility tests

In all these cases, the following question is relevant:

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

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

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

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

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

$$\ell \to r \Leftarrow s_1 \to t_1, \ldots, s_n \to t_n$$

whose operational semantics is given by the following inference system:

(Rf) $$\frac{}{x \to^* x}$$

(C) $$\frac{x_i \to y_i}{f(x_1, \ldots, x_i, \ldots, x_k) \to f(x_1, \ldots, y_i, \ldots, x_k)}$$
for all $f \in \mathcal{F}$ and $1 \leq i \leq k = arity(f)$

(T) $$\frac{x \to z \qquad z \to^* y}{x \to^* y}$$

(Rp) $$\frac{s_1 \to^* t_1 \quad \ldots \quad s_n \to^* t_n}{\ell \to r}$$
for all $\ell \to r \Leftarrow s_1 \to t_1 \cdots s_n \to t_n \in \mathcal{R}$

Definition

Let $\mathcal{R}$ be a CTRS. A sequence $s_1 \to^* t_1, \ldots, s_n \to^* t_n$, 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 σ such that for all $1 \leq i \leq n$, $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$. 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 $\frac{B_1 \ \cdots \ B_n}{A}$ become universally quantified *implications* $B_1 \wedge \cdots \wedge B_n \Rightarrow A$.

Example

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

$$\begin{array}{rcl} \mathsf{a} & \to & \mathsf{b} \\ \mathsf{f(a)} & \to & \mathsf{b} \end{array} \qquad \begin{array}{rcl} \mathsf{g}(x) & \to & \mathsf{g(a)} \Leftarrow \mathsf{f}(x) \to x \end{array}$$

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

$$\begin{array}{rl} (\forall x) & x \to^* x \\ (\forall x, y, z) & x \to y \wedge y \to^* z \Rightarrow x \to^* z \\ (\forall x, y) & x \to y \Rightarrow \mathsf{f}(x) \to \mathsf{f}(y) \\ (\forall x, y) & x \to y \Rightarrow \mathsf{g}(x) \to \mathsf{g}(y) \end{array} \qquad \begin{array}{rl} & \mathsf{a} \to \mathsf{b} \\ & \mathsf{f(a)} \to \mathsf{b} \\ (\forall x) & \mathsf{f}(x) \to^* x \Rightarrow \mathsf{g}(x) \to \mathsf{g(a)} \end{array}$$

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

Proposition

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

$$\mathcal{A} \models \overline{\mathcal{R}} \cup \{\neg(\exists\vec{x})\,(s_1 \to^* t_1 \wedge \cdots \wedge s_n \to^* t_n)\}$$

holds.

Proof by contradiction

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

Proof by contradiction

1. $\mathcal{R}$-feasiblity implies that $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\overline{\mathcal{R}} \vdash \sigma(s_i) \to^* \sigma(t_i)$) for some substitution σ and all $1 \leq i \leq n$.
2. Since $\mathcal{A} \models \overline{\mathcal{R}}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i)\ \sigma(s_i) \to^* \sigma(t_i)$ for all $1 \leq i \leq n$, where $\vec{y}_i$ are the variables in $\mathcal{V}ar(\sigma(s_i)) \cup \mathcal{V}ar(\sigma(t_i))$.

Proof by contradiction

1. $\mathcal{R}$-feasiblity implies that $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\overline{\mathcal{R}} \vdash \sigma(s_i) \to^* \sigma(t_i)$) for some substitution σ and all $1 \leq i \leq n$.
2. Since $\mathcal{A} \models \overline{\mathcal{R}}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i)\ \sigma(s_i) \to^* \sigma(t_i)$ 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, $\mathcal{A} \models (\forall \vec{y})\ (\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n))$.

Proof by contradiction

1. $\mathcal{R}$-feasiblity implies that $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\overline{\mathcal{R}} \vdash \sigma(s_i) \to^* \sigma(t_i)$) for some substitution σ and all $1 \leq i \leq n$.
2. Since $\mathcal{A} \models \overline{\mathcal{R}}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i)\ \sigma(s_i) \to^* \sigma(t_i)$ 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, $\mathcal{A} \models (\forall \vec{y})\ (\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n))$.
4. For all $\nu : \vec{y} \to \mathcal{A}$, $[\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n)]^{\mathcal{A}}_{\nu}$, is *true*.

Proof by contradiction

1. $\mathcal{R}$-feasiblity implies that $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\overline{\mathcal{R}} \vdash \sigma(s_i) \to^* \sigma(t_i)$) for some substitution σ and all $1 \leq i \leq n$.
2. Since $\mathcal{A} \models \overline{\mathcal{R}}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i)\ \sigma(s_i) \to^* \sigma(t_i)$ 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, $\mathcal{A} \models (\forall \vec{y})\ (\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n))$.
4. For all $\nu : \vec{y} \to \mathcal{A}$, $[\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n)]^{\mathcal{A}}_{\nu}$, is *true*.
5. Since $\mathcal{A}$ has a nonempty domain, there is a valuation $\nu' : \vec{x} \to \mathcal{A}$ given by $\nu'(x) = [\sigma(x)]^{\mathcal{A}}_{\nu}$ for all variable x in $\vec{x} = \bigcup_{i=1}^{n} \mathcal{V}ar(s_i) \cup \mathcal{V}ar(t_i)$, such that $[s_1 \to^* t_1 \wedge \cdots \wedge s_n \to^* t_n]^{\mathcal{A}}_{\nu'}$ is true.

Proof by contradiction

1. $\mathcal{R}$-feasiblity implies that $\sigma(s_i) \to^*_{\mathcal{R}} \sigma(t_i)$ (i.e., $\overline{\mathcal{R}} \vdash \sigma(s_i) \to^* \sigma(t_i)$) for some substitution σ and all $1 \leq i \leq n$.
2. Since $\mathcal{A} \models \overline{\mathcal{R}}$, by correctness, $\mathcal{A} \models (\forall \vec{y}_i)\ \sigma(s_i) \to^* \sigma(t_i)$ 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, $\mathcal{A} \models (\forall \vec{y})\ (\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n))$.
4. For all $\nu : \vec{y} \to \mathcal{A}$, $[\sigma(s_1) \to^* \sigma(t_1) \wedge \cdots \wedge \sigma(s_n) \to^* \sigma(t_n)]^{\mathcal{A}}_{\nu}$, is *true*.
5. Since $\mathcal{A}$ has a nonempty domain, there is a valuation $\nu' : \vec{x} \to \mathcal{A}$ given by $\nu'(x) = [\sigma(x)]^{\mathcal{A}}_{\nu}$ for all variable x in $\vec{x} = \bigcup_{i=1}^{n} \mathcal{V}ar(s_i) \cup \mathcal{V}ar(t_i)$, such that $[s_1 \to^* t_1 \wedge \cdots \wedge s_n \to^* t_n]^{\mathcal{A}}_{\nu'}$ is true.
6. This contradicts $\mathcal{A} \models \neg(\exists \vec{x})\ (s_1 \to^* t_1 \wedge \cdots \wedge s_n \to^* t_n)$.

8

The following structure $\mathcal{A}$ over $\mathbb{N} - \{0\}$:

$$\mathsf{a}^{\mathcal{A}} = 1 \qquad \mathsf{b}^{\mathcal{A}} = 2 \qquad \mathsf{f}^{\mathcal{A}}(x) = x + 1 \qquad \mathsf{g}^{\mathcal{A}}(x) = 1$$
$$x \to^{\mathcal{A}} y \Leftrightarrow y \geq x \qquad x\,(\to^*)^{\mathcal{A}}\, y \Leftrightarrow y \geq x$$

is a model of $\overline{\mathcal{R}} \cup \{\neg(\exists x)\,\mathsf{f}(x) \to^* x\}$ 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

$$\mathsf{g}(x) \to \mathsf{g}(\mathsf{a}) \Leftarrow \mathsf{f}(x) \to x$$

is proved $\mathcal{R}$-infeasible.

The following CTRS $\mathcal{R}$ (Example 23 in [Sternagel & Sternagel, FSCD'16])

$$\begin{array}{rcll} \mathsf{g}(x) & \to & \mathsf{f}(x,x) & (1) \\ \mathsf{g}(x) & \to & \mathsf{g}(x) \Leftarrow \mathsf{g}(x) \to \mathsf{f}(\mathsf{a},\mathsf{b}) & (2) \end{array}$$

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

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

$$\mathsf{g}^{\mathcal{A}}(x) = 1 \qquad x \to^{\mathcal{A}} y \Leftrightarrow x = y \qquad x\,(\to^{*})^{\mathcal{A}}\, y \Leftrightarrow x \geq y$$

is a model $\overline{\mathcal{R}} \cup \{\neg(\exists x)\ \mathsf{g}(x) \to^{*} \mathsf{f}(\mathsf{a},\mathsf{b})\}$. 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 $\mathcal{R}$ 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 $s \to^* x, t \to^* x$ is $\mathcal{R}$-feasible. If s and t are ground and $s \to^* x, t \to^* x$ is $\mathcal{R}$-feasible, then s and t are joinable.

Consider the following CTRS $\mathcal{R}$ (Example 7.3.3 in Ohlebusch's book):

$$\begin{array}{rcl} \mathsf{a} & \to & \mathsf{b} \qquad (3) \\ \mathsf{f}(x) & \to & \mathsf{c} \Leftarrow x \to \mathsf{a} \qquad (4) \end{array}$$

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

$$\begin{array}{cccc} \mathsf{a}^{\mathcal{A}} = 0 & \mathsf{b}^{\mathcal{A}} = -1 & \mathsf{c}^{\mathcal{A}} = 1 & \mathsf{f}^{\mathcal{A}}(x) = x + 1 \\ x \to^{\mathcal{A}} y \Leftrightarrow x = y & x\,(\to^*)^{\mathcal{A}}\, y \Leftrightarrow x = y & & \end{array}$$

is a model of $\overline{\mathcal{U}(\mathcal{R}, \mathsf{f}(\mathsf{b}), \mathsf{c})} \cup \{\neg(\exists x)\,(\mathsf{f}(\mathsf{b}) \to^* x \wedge \mathsf{c} \to^* x)\}$, where $\mathcal{U}(\mathcal{R}, \mathsf{f}(\mathsf{b}), \mathsf{c}) = \{(4)\}$ is the set of usable rules (from $\mathcal{R}$) for $\mathsf{f}(\mathsf{b})$ and c. Therefore, $\mathsf{f}(\mathsf{b})$ 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!