

A Congruence Rule Format with Universal Quantification

Mohammad Mousavi^{1,2} and Michel Reniers²

¹Reykjavík University, Iceland

²Eindhoven University of Technology, The Netherlands

SOS'07, Wrocław, Poland

1 Introduction

2 Motivation

3 UNTYFT Format

4 Conclusions

Transition System Specification (TSS)

Example

$$(\mathbf{aaa}) \frac{}{a \xrightarrow{a} a}$$

$$(\mathbf{aab}) \frac{}{a \xrightarrow{a} b}$$

$$(\mathbf{bbb}) \frac{}{b \xrightarrow{b} b}$$

$$(\mathbf{f}) \frac{x \xrightarrow{a} y \quad y \xrightarrow{b} \neg y}{f(x) \xrightarrow{c} y}$$

Proof

Positive formula ϕ is provable from TSS R when

- there exists a rule $r = \frac{\{\phi_i \mid i \in I\}}{\psi} \in R$ and
- **there exists a substitution** σ s.t.
- $\sigma(\psi) = \phi$ and
- $\sigma(\phi_i)$'s is provable **for each** $i \in I$.

For negative formulae: show that all proof attempts for their negation is bound for failure).

Example

$$\text{(aaa)} \frac{}{a \xrightarrow{a} a}$$

$$\text{(aab)} \frac{}{a \xrightarrow{a} b}$$

$$\text{(bbb)} \frac{}{b \xrightarrow{b} b}$$

$$\text{(f)} \frac{x \xrightarrow{a} y \quad y \xrightarrow{b} z}{f(x) \xrightarrow{c} c}$$

Example

$$\text{(aaa)} \frac{}{a \xrightarrow{a} a}$$

$$\text{(aab)} \frac{}{a \xrightarrow{a} b}$$

$$\text{(bbb)} \frac{}{b \xrightarrow{b} b}$$

$$\text{(f)} \frac{x \xrightarrow{a} y \quad y \xrightarrow{b} \quad}{f(x) \xrightarrow{c} c}$$

$f(a) \xrightarrow{c} c$ and $f(b) \not\xrightarrow{c}$ are provable.

Overview

- Goal: Prove **meta-results** by restricting the **syntactic shape** of the rules
- Congruence rule formats: prove that a notion of **behavioral equivalence** is a **congruence**.

Overview

- Goal: Prove **meta-results** by restricting the **syntactic shape** of the rules
- Congruence rule formats: prove that a notion of **behavioral equivalence** is a **congruence**.

NTYFT [Groote92]

$$\frac{\{t_i \xrightarrow{l_i} y_i \mid i \in I\} \{t_j \xrightarrow{l_j} \dashv \mid j \in J\}}{f(\vec{x}) \xrightarrow{l} t}$$

Overview

- Goal: Prove **meta-results** by restricting the **syntactic shape** of the rules
- Congruence rule formats: prove that a notion of **behavioral equivalence** is a **congruence**.

NTYFT [Groote92]

$$\frac{\{t_i \xrightarrow{l_i} y_i \mid i \in I\} \{t_j \xrightarrow{l_j} z_j \mid j \in J\}}{f(\vec{x}) \xrightarrow{l} t}$$

Theorem

For a complete TSS in the NTYFT format, strong bisimilarity is a congruence.

Outline

1 Introduction

2 Motivation

3 UNTYFT Format

4 Conclusions

Safety Predicate

$$\frac{\forall_y x \xrightarrow{a} y \Rightarrow \mathit{safe}(y)}{\mathit{safe}(x)}$$

Safety Predicate

$$\frac{\forall_y x \xrightarrow{a} y \Rightarrow \text{safe}(y)}{\text{safe}(x)}$$

or

$$\frac{\forall_y x \not\xrightarrow{a} y \vee \text{safe}(y)}{\text{safe}(x)}$$

Safety Predicate (cont'd)

One may replace

$$\frac{\forall y x \xrightarrow{a} y \vee \text{safe}(y)}{\text{safe}(x)}$$

by

$$\frac{\{x \xrightarrow{a} p \vee \text{safe}(p) \mid p \in \mathcal{C}\}}{\text{safe}(x)}$$

or by infinitely many rules, such as $r_{p_0 \overline{p_1 p_2} \dots}$ s.t. $\mathcal{C} = \{p_0, p_1, \dots\}$,

$$\frac{x \xrightarrow{a} p_0 \quad \text{safe}(p_1) \quad \text{safe}(p_2) \dots}{\text{safe}(x)}$$

Motivating Examples

Safety Predicate (cont'd)

$$\frac{x \xrightarrow{a} p_0 \quad \text{safe}(p_1) \quad \text{safe}(p_2) \dots}{\text{safe}(x)}$$

Drawbacks of Eliminating Quantifiers and Disjunctions

The resulting TSS will:

- 1 have **infinitely many** more rules, and

Motivating Examples

Safety Predicate (cont'd)

$$\frac{x \xrightarrow{a} p_0 \quad \text{safe}(p_1) \quad \text{safe}(p_2) \dots}{\text{safe}(x)}$$

Drawbacks of Eliminating Quantifiers and Disjunctions

The resulting TSS will:

- 1 have **infinitely many** more rules, and
- 2 (still) be **beyond** all **congruence rule** formats.

NTYFT (recap)

$$\frac{\{t_i \xrightarrow{l_i} y_i \mid i \in I\} \{t_j \xrightarrow{l_j} \mid j \in J\}}{f(\vec{x}) \xrightarrow{l} t}$$

Other Examples

Other realistic examples include:

- 1 weak termination, semantical divergence [Aceto&Hennessy'92] and
- 2 semantics of prioritized term rewriting [van de Pol'98].

1 Introduction

2 Motivation

3 UNTYFT Format

4 Conclusions

Clause

$$\Phi ::= t \xrightarrow{I} t' \mid t \not\xrightarrow{I} t' \mid \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i$$

We assume CNF for simplicity.

Clause

$$\Phi ::= t \xrightarrow{l} t' \mid t \not\xrightarrow{l} t' \mid \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i$$

We assume CNF for simplicity.

Predicate

$$\Psi ::= \forall_{\tilde{z}_1} \Phi$$

Deduction Rule

$$\frac{\Psi}{t \xrightarrow{l} t'}$$

Clause

$$\Phi ::= t \xrightarrow{l} t' \mid t \not\xrightarrow{l} t' \mid \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i$$

We assume CNF for simplicity.

Predicate

$$\Psi ::= \exists_{\tilde{z}_1} \forall_{\tilde{z}_1} \Phi$$

Deduction Rule

$$\frac{\Psi}{t \xrightarrow{l} t'}$$

Clause

$$\Phi ::= t \xrightarrow{l} t' \mid t \not\xrightarrow{l} t' \mid \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i$$

We assume CNF for simplicity.

Predicate

$$\Psi ::= \forall_{\tilde{z}_1} \exists_{\tilde{z}_2} \Phi$$

Deduction Rule

$$\frac{\Psi}{t \xrightarrow{l} t'}$$

Clause

$$\Phi ::= t \xrightarrow{l} t' \mid t \not\xrightarrow{l} t' \mid \bigwedge_{i \in I} \Phi_i \mid \bigvee_{i \in I} \Phi_i$$

We assume CNF for simplicity.

Predicate

$$\Psi ::= \exists_{\tilde{z}_1} \forall_{\tilde{z}_1} \exists_{\tilde{z}_2} \Phi$$

Deduction Rule

$$\frac{\Psi}{t \xrightarrow{l} t'}$$

All variables are either quantified, or appear in $t \xrightarrow{l} t'$ (but not both).

Semantics

- 1 Eliminate quantification;
- 2 Eliminate disjunction;
- 3 Close the deduction rules (for remaining variables in the conclusion);
- 4 Use the traditional semantics for TSS's.

UNTYFT Rules

$$\frac{\exists \tilde{z}_0 \forall \tilde{z}_1 \exists \tilde{z}_2 \forall_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{t \xrightarrow{l} t'},$$

First Constraint

t should be of the form $f(\vec{x})$.

UNTYFT Rules

$$\frac{\exists \tilde{z}_0 \forall \tilde{z}_1 \exists \tilde{z}_2 \forall_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{t \xrightarrow{l} t'},$$

First Constraint

t should be of the form $f(\vec{x})$.

Motivating Counter-Example

$$\frac{}{f(a) \xrightarrow{a} a}$$

$a \leftrightarrow b$ but not $f(a) \leftrightarrow f(b)$.

UNTYFT Rules

$$\frac{\exists_{\tilde{z}_0} \forall_{\tilde{z}_1} \exists_{\tilde{z}_2} \bigvee_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{f(\vec{x}) \xrightarrow{l} t'},$$

Second Constraint

All x , y_i and y'_j variables should be pairwise distinct.

UNTYFT Rules

$$\frac{\exists_{\tilde{z}_0} \forall_{\tilde{z}_1} \exists_{\tilde{z}_2} \bigvee_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{f(\vec{x}) \xrightarrow{l} t'}$$

Second Constraint

All x , y_i and y'_j variables should be pairwise distinct.

Motivating Counter-Example

$$\frac{\overline{a \xrightarrow{a} a} \quad \overline{a \xrightarrow{b} a} \quad \overline{b \xrightarrow{a} a} \quad \overline{b \xrightarrow{b} b} \quad \forall_y x \xrightarrow{a} y \vee x \xrightarrow{b} y}{f(x) \xrightarrow{c} c}$$

$a \leftrightarrow b$ but not $f(a) \leftrightarrow f(b)$.

UNTYFT Rules

$$\frac{\exists \tilde{z}_0 \forall \tilde{z}_1 \exists \tilde{z}_2 \forall_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{f(\vec{x}) \xrightarrow{l} t'}$$

Third Constraint

$$\tilde{z}_1 \cap \{y_i \mid i \in I_k, k \in K\} = \emptyset$$

Motivating Counter-Example

$$\overline{a \xrightarrow{a} x} \quad \overline{b \xrightarrow{a} a} \quad \overline{b \xrightarrow{a} c} \quad \overline{b \xrightarrow{a} f(a)} \quad \overline{\forall_y x \xrightarrow{a} y} \\ f(x) \xrightarrow{c} a$$

$a \leftrightarrow b$ but not $f(a) \leftrightarrow f(b)$.

UNTYFT Rules

$$\frac{\exists \tilde{z}_0 \forall \tilde{z}_1 \exists \tilde{z}_2 \forall_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{f(\vec{x}) \xrightarrow{l} t'}$$

Fourth Constraint

$$\{y'_j \mid j \in J_k, k \in K\} \subseteq \tilde{z}_1$$

Fourth Constraint

$$\{y'_j \mid j \in J_k, k \in K\} \subseteq \tilde{z}_1$$

Motivating Counter-Example

$$\overline{a \xrightarrow{a} c} \quad \overline{a \xrightarrow{a} c'} \quad \overline{b \xrightarrow{a} c} \quad \overline{c \xrightarrow{b} c} \quad \overline{c' \xrightarrow{b} c}$$

$$\frac{\exists_{y,y',y''} x \xrightarrow{a} y \quad y \xrightarrow{b} y' \quad x \xrightarrow{a} y''}{f(x) \xrightarrow{a} c}$$

$a \leftrightarrow b$ but not $f(a) \leftrightarrow f(b)$.

UNTYFT Rules

$$\frac{\exists_{\tilde{z}_0} \forall_{\tilde{z}_1} \exists_{\tilde{z}_2} \bigvee_{k \in K} (\bigwedge_{i \in I_k} t_i \xrightarrow{l_i} y_i \wedge \bigwedge_{j \in J_k} t'_j \xrightarrow{l'_j} y'_j)}{f(\vec{x}) \xrightarrow{l} t'}$$

Last Constraint

Variables in \tilde{z}_0 should depend on variables in $\tilde{z}_1 \cup \tilde{z}_2$.

Last Constraint

Variables in \tilde{z}_0 should depend on variables in $\tilde{z}_1 \cup \tilde{z}_2$.

Motivating Counter-Example

$$\frac{}{a \xrightarrow{a} b} \quad \frac{}{b \xrightarrow{a} a}$$

$$\frac{\forall y_0 x_0 \xrightarrow{a} y_0 \quad x_1 \xrightarrow{a} y_1}{f(x_0, x_1) \xrightarrow{b} y_1}$$

$a \leftrightarrow b$ but not $f(a) \leftrightarrow f(b)$.

$$\frac{\exists z_0 \forall z_1 f(z_1, x) \xrightarrow{b} z_0}{g(x) \xrightarrow{c} c}$$

$$\frac{\exists y_0, y_1 x_0 \xrightarrow{a} y_0 \quad x_1 \xrightarrow{a} y_1}{f(x_0, x_1) \xrightarrow{b} a}$$

Theorem

For a complete extended TSS in the UNTYFT format, strong bisimilarity is a congruence.

Traditional Congruence Proof

- 1 Let R be the congruence closure of \leftrightarrow ;
- 2 Show that R is a bisimulation:

Traditional Congruence Proof

- 1 Let R be the congruence closure of \leftrightarrow ;
- 2 Show that R is a bisimulation:
 - 1 taking a pair $(f(\vec{p}), f(\vec{q}))$ s.t. $\vec{p} R \vec{q}$ and
 - 2 showing that if $f(p) \xrightarrow{!} p'$, then $f(q) \xrightarrow{!} q'$ for some q' s.t. $p' R q'$ (an induction on the structure of the proof for $f(p) \xrightarrow{!} p'$):

Traditional Congruence Proof

- 1 Let R be the congruence closure of \leftrightarrow ;
- 2 Show that R is a bisimulation:
 - 1 taking a pair $(f(\vec{p}), f(\vec{q}))$ s.t. $\vec{p} R \vec{q}$ and
 - 2 showing that if $f(p) \xrightarrow{!} p'$, then $f(q) \xrightarrow{!} q'$ for some q' s.t. $p' R q'$ (an induction on the structure of the proof for $f(p) \xrightarrow{!} p'$):
 - 1 take the last deduction rule $r = \frac{\Phi}{f(\vec{x}) \xrightarrow{!} t'}$ together with substitution σ used in the proof;
 - 2 define σ'_0 such that $\sigma'_0(\vec{x}) = \vec{q}$.
 - 3 complete the definition of σ' using a fixpoint construction by following the chains of premises and exploiting the induction hypothesis.

Challenge

- 1 We have premises of the form $t_j \xrightarrow{!} y_j$;
- 2 Assume that $p \Leftrightarrow q$; it does **not** hold that if $p \xrightarrow{!} p'$ then $q \xrightarrow{!} q'$ for some $q' \Leftrightarrow p'$.

Congruence Proof for UNTYFT

Challenge

- 1 We have premises of the form $t_j \stackrel{!}{\dashv} y_j$;
- 2 Assume that $p \Leftrightarrow q$; it does **not** hold that if $p \stackrel{!}{\dashv} p'$ then $q \stackrel{!}{\dashv} q'$ for some $q' \Leftrightarrow p'$.

UNTYFT Congruence Proof

- 1 Common to the tradition proof, we take start with σ_0 and complete its definition, but
- 2 for for premises of the form $t \stackrel{!}{\dashv} t'$, $t = y'_j$ is universally quantified; thus, we may change σ on these variables while constructing σ' .

Outline

1 Introduction

2 Motivation

3 UNTYFT Format

4 Conclusions

Done

- 1 Extended the syntax and semantics of TSS with **universal quantification** and **disjunction**.
- 2 Devised a **congruence rule format** for strong bisimilarity w.r.t. the extended TSS

To Be Done

- 1 Studying the **expressiveness** of the UNTYFT format.
- 2 Establishing a link between the UNTYFT format and the **ordered TYFT** format [MPRU, FSTTCS'06]