Duplicate Check

Nao Hirokawa

JAIST

February 20, 2024

About This Talk

1. SMT solving for $\mathcal{R} \approx \mathcal{S}$ (based on Thiemann's approach)

About This Talk

1. SMT solving for $\mathcal{R} \approx \mathcal{S}$ (based on Thiemann's approach)

2. fingerprints (variant of Felgenhauer's approach)

About This Talk

1. SMT solving for $\mathcal{R} \approx \mathcal{S}$ (based on Thiemann's approach)

2. fingerprints (variant of Felgenhauer's approach)

3. experiments

About This Talk

1. SMT solving for $\mathcal{R} \approx \mathcal{S}$ (based on Thiemann's approach)
2. fingerprints (variant of Felgenhauer's approach)
3. experiments
4. future work

Example (partition)

consider collection of TRSs $\mathfrak{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_6\}$:

$$\mathcal{R}_1 = \left\{\begin{matrix} \mathsf{+}(\mathsf{0}, x) \to x \\ \mathsf{+}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{+}(x, y)) \\ \mathtt{p}(\mathsf{s}(x)) \to x \end{matrix}\right\}$$

$$\mathcal{R}_2 = \left\{\begin{matrix} \mathsf{plus}(\mathsf{zero}, n) \to n \\ \mathsf{plus}(\mathsf{succ}(m), n) \to \mathsf{succ}(\mathsf{plus}(m, n)) \\ \mathtt{pred}(\mathsf{succ}(n)) \to n \end{matrix}\right\}$$

$$\mathcal{R}_3 = \{\mathsf{a} \to \mathsf{b}\}$$

$$\mathcal{R}_4 = \{\mathsf{d} \to \mathsf{c}\}$$

$$\mathcal{R}_5 = \{\mathsf{c} \to \mathsf{d}\}$$

$$\mathcal{R}_6 = \{\mathsf{f}(\mathsf{f}(x)) \to x\}$$

Example (partition)

consider collection of TRSs $\mathfrak{R} = \{\mathcal{R}_1, \ldots, \mathcal{R}_6\}$:

$$\mathcal{R}_1 = \left\{ \begin{array}{c} \mathsf{+}(\mathsf{0}, x) \to x \\ \mathsf{+}(\mathsf{s}(x), y) \to \mathsf{s}(\mathsf{+}(x, y)) \\ \mathsf{p}(\mathsf{s}(x)) \to x \end{array} \right\}$$

$$\mathcal{R}_2 = \left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, n) \to n \\ \mathsf{plus}(\mathsf{succ}(m), n) \to \mathsf{succ}(\mathsf{plus}(m, n)) \\ \mathtt{pred}(\mathsf{succ}(n)) \to n \end{array} \right\}$$

$$\mathcal{R}_3 = \{\mathsf{a} \to \mathsf{b}\}$$

$$\mathcal{R}_4 = \{\mathsf{d} \to \mathsf{c}\}$$

$$\mathcal{R}_5 = \{\mathsf{c} \to \mathsf{d}\}$$

$$\mathcal{R}_6 = \{\mathsf{f}(\mathsf{f}(x)) \to x\}$$

duplicate checker is expected to compute partition by $\approx$:

$$\mathfrak{R}/{\approx} \;=\; \{\{\mathcal{R}_1, \mathcal{R}_2\}, \{\mathcal{R}_3, \mathcal{R}_4, \mathcal{R}_5\}, \{\mathcal{R}_6\}\}$$

Definition (definition of duplicates)

Definition (definition of duplicates)

- $\mathcal{R} \doteq \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are same modulo variable renaming

Definition (definition of duplicates)

- $\mathcal{R} \doteq \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are same modulo variable renaming
- $\mathcal{R} \approx \mathcal{S}$ if $\mathcal{R}^\theta \doteq \mathcal{S}$ for some renaming θ for function symbols

Definition (definition of duplicates)

- $\mathcal{R} \doteq \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are same modulo variable renaming
- $\mathcal{R} \approx \mathcal{S}$ if $\mathcal{R}^\theta \doteq \mathcal{S}$ for some renaming θ for function symbols

Example

$$\left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, n) \to n \\ \mathsf{plus}(\mathsf{succ}(m), n) \to \mathsf{succ}(\mathsf{plus}(m, n)) \\ \mathtt{pred}(\mathsf{succ}(n)) \to n \end{array} \right\} \approx \left\{ \begin{array}{c} +(0, x) \to x \\ +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) \\ \mathsf{p}(\mathsf{s}(x)) \to x \end{array} \right\}$$

$$\text{because of renaming } \theta\text{:} \quad \left\{ \begin{array}{l} \mathsf{plus}^\theta = + \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{pred}^\theta = \mathsf{p} \end{array} \right\}$$

Definition (definition of duplicates)

- $\mathcal{R} \doteq \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are same modulo variable renaming
- $\mathcal{R} \approx \mathcal{S}$ if $\mathcal{R}^\theta \doteq \mathcal{S}$ for some renaming θ for function symbols

Example

$$\left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, n) \to n \\ \mathsf{plus}(\mathsf{succ}(m), n) \to \mathsf{succ}(\mathsf{plus}(m, n)) \\ \mathtt{pred}(\mathsf{succ}(n)) \to n \end{array} \right\} \approx \left\{ \begin{array}{c} +(0, x) \to x \\ +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) \\ \mathtt{p}(\mathsf{s}(x)) \to x \end{array} \right\}$$

$$\text{because of renaming } \theta\text{:} \quad \left\{ \begin{array}{l} \mathsf{plus}^\theta = + \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{pred}^\theta = \mathsf{p} \end{array} \right\}$$

Q. how to find it?

Definition (definition of duplicates)

- $\mathcal{R} \doteq \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are same modulo variable renaming
- $\mathcal{R} \approx \mathcal{S}$ if $\mathcal{R}^\theta \doteq \mathcal{S}$ for some renaming θ for function symbols

Example

$$\left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, n) \to n \\ \mathsf{plus}(\mathsf{succ}(m), n) \to \mathsf{succ}(\mathsf{plus}(m, n)) \\ \mathtt{pred}(\mathsf{succ}(n)) \to n \end{array} \right\} \approx \left\{ \begin{array}{c} +(0, x) \to x \\ +(\mathsf{s}(x), y) \to \mathsf{s}(+(x, y)) \\ \mathtt{p}(\mathsf{s}(x)) \to x \end{array} \right\}$$

$$\text{because of renaming } \theta\text{:} \quad \left\{ \begin{array}{l} \mathsf{plus}^\theta = + \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{pred}^\theta = \mathsf{p} \end{array} \right\}$$

Q. how to find it? A. reducing problem to equality constraint

Encoding Equivalence of Rules

Example

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array}$$

Encoding Equivalence of Rules

Example

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = \mathsf{+} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{plus}^\theta = \mathsf{+} \end{Bmatrix}$$

Encoding Equivalence of Rules

Example

$$\begin{array}{rr} & \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = & +(\quad \mathsf{s}(x_1), x_2) \to \quad \mathsf{s}(\quad +(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = + \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{plus}^\theta = + \end{Bmatrix}$$

$$\begin{array}{rr} & \mathsf{plus}^\theta(\color{magenta}{\mathsf{zero}^\theta}, x_1) \to \quad x_1 \\ = & +(\color{magenta}{\mathsf{s}(x_1)}, x_2) \to \mathsf{s}(+(x_1, x_2)) \end{array}$$

Encoding Equivalence of Rules

Example

$$\begin{array}{c} \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = \quad +(\quad \mathsf{s}(x_1), x_2) \to \quad \mathsf{s}(\quad +(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = + \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{plus}^\theta = + \end{Bmatrix}$$

$$\begin{array}{c} \mathsf{plus}^\theta(\mathsf{zero}^\theta, x_1) \to \quad x_1 \\ = \quad +(\mathsf{s}(x_1), x_2) \to \mathsf{s}(+(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = + \\ \bot \end{Bmatrix}$$

Encoding Equivalence of Rules

Example

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = \mathsf{+} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{plus}^\theta = \mathsf{+} \end{Bmatrix}$$

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{zero}^\theta, x_1) \to x_1 \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = \mathsf{+} \\ \bot \end{Bmatrix}$$

$$\begin{array}{rl} & \mathsf{f}^\theta(x_1) \to x_2 \ \textit{if} \ \mathsf{g}^\theta(x_1) \to x_2 \\ = & \mathsf{g}(x_1) \to x_2 \ \textit{if} \ \mathsf{f}(x_1) \to x_2 \end{array}$$

Encoding Equivalence of Rules

Example

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{succ}^\theta(x_1), x_2) \to \mathsf{succ}^\theta(\mathsf{plus}^\theta(x_1, x_2)) \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = \mathsf{+} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{succ}^\theta = \mathsf{s} \\ \mathsf{plus}^\theta = \mathsf{+} \end{Bmatrix}$$

$$\begin{array}{rl} & \mathsf{plus}^\theta(\mathsf{zero}^\theta, x_1) \to x_1 \\ = & \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{plus}^\theta = \mathsf{+} \\ \bot \end{Bmatrix}$$

$$\begin{array}{rl} & \mathsf{f}^\theta(x_1) \to x_2 \;\; \textit{if} \;\; \mathsf{g}^\theta(x_1) \to x_2 \\ = & \mathsf{g}(x_1) \to x_2 \;\; \textit{if} \;\; \mathsf{f}(x_1) \to x_2 \end{array} \iff \bigwedge \begin{Bmatrix} \mathsf{f}^\theta = \mathsf{g} \\ \mathsf{g}^\theta = \mathsf{f} \end{Bmatrix}$$

Fact

$$\mathcal{R} \subseteq \mathcal{S} \iff \bigwedge_{\alpha \in \mathcal{R}} \bigvee_{\beta \in \mathcal{S}} \alpha = \beta$$

Fact

$$\mathcal{R} \subseteq \mathcal{S} \iff \bigwedge_{\alpha \in \mathcal{R}} \bigvee_{\beta \in \mathcal{S}} \alpha = \beta \quad \text{and} \quad \mathcal{R}^\theta = \mathcal{S} \iff \mathcal{R}^\theta \subseteq \mathcal{S} \wedge \mathcal{R}^\theta \supseteq \mathcal{S}$$

Fact

$$\mathcal{R} \subseteq \mathcal{S} \iff \bigwedge_{\alpha \in \mathcal{R}} \bigvee_{\beta \in \mathcal{S}} \alpha = \beta \quad \text{and} \quad \mathcal{R}^\theta = \mathcal{S} \iff \mathcal{R}^\theta \subseteq \mathcal{S} \wedge \mathcal{R}^\theta \supseteq \mathcal{S}$$

Example

$$\left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, x_1) \to x_1 \\ \mathsf{plus}(\mathsf{succ}(x_1), x_2) \to \mathsf{succ}(\mathsf{plus}(x_1, x_2)) \\ \mathtt{pred}(\mathsf{succ}(x_1)) \to x_1 \end{array} \right\}^\theta \subseteq \left\{ \begin{array}{c} \mathsf{+}(\mathsf{0}, x_1) \to x_1 \\ \mathsf{+}(\mathsf{s}(x_1), x_2) \to \mathsf{s}(\mathsf{+}(x_1, x_2)) \\ \mathsf{p}(\mathsf{s}(x_1)) \to x_1 \end{array} \right\}$$

Fact

$$\mathcal{R} \subseteq \mathcal{S} \iff \bigwedge_{\alpha \in \mathcal{R}} \bigvee_{\beta \in \mathcal{S}} \alpha = \beta \quad \text{and} \quad \mathcal{R}^\theta = \mathcal{S} \iff \mathcal{R}^\theta \subseteq \mathcal{S} \wedge \mathcal{R}^\theta \supseteq \mathcal{S}$$

Example

$$\left\{ \begin{array}{c} \mathsf{plus}(\mathsf{zero}, x_1) \to x_1 \\ \mathsf{plus}(\mathsf{succ}(x_1), x_2) \to \mathsf{succ}(\mathsf{plus}(x_1, x_2)) \\ \mathtt{pred}(\mathsf{succ}(x_1)) \to x_1 \end{array} \right\}^\theta \subseteq \left\{ \begin{array}{c} +(0, x_1) \to x_1 \\ +(\mathsf{s}(x_1), x_2) \to \mathsf{s}(+(x_1, x_2)) \\ \mathsf{p}(\mathsf{s}(x_1)) \to x_1 \end{array} \right\}$$

$$\iff \bigwedge \left\{ \begin{array}{l} (\mathsf{plus}^\theta = + \ \wedge \ \mathsf{zero}^\theta = 0) \ \vee \ \bot \ \vee \ \bot \\ \bot \ \vee \ (\mathsf{plus}^\theta = + \ \wedge \ \mathsf{succ}^\theta = \mathsf{s}) \ \vee \ \bot \\ \bot \ \vee \ \bot \ \vee \ (\mathsf{pred}^\theta = \mathsf{p} \ \wedge \ \mathsf{succ}^\theta = \mathsf{s}) \end{array} \right\}$$

Encoding Equivalence of Signatures

$$\left\{ \begin{array}{c} (\mathsf{plus}, 2) \\ (\mathsf{succ}, 1) \\ (\mathsf{pred}, 1) \end{array} \right\}^{\theta} \subseteq \left\{ \begin{array}{c} (\mathsf{+}, 2) \\ (\mathsf{s}, 1) \\ (\mathsf{p}, 1) \end{array} \right\}$$

Encoding Equivalence of Signatures

$$\left\{ \begin{array}{c} (\mathsf{plus}, 2) \\ (\mathsf{succ}, 1) \\ (\mathsf{pred}, 1) \end{array} \right\}^{\theta} \subseteq \left\{ \begin{array}{c} (\mathsf{+}, 2) \\ (\mathsf{s}, 1) \\ (\mathsf{p}, 1) \end{array} \right\}$$

$$\iff \bigwedge \left\{ \begin{array}{lllll} \mathsf{plus}^{\theta} = \mathsf{+} & \vee & \bot & \vee & \bot \\ \bot & \vee & \mathsf{succ}^{\theta} = \mathsf{s} & \vee & \mathsf{succ}^{\theta} = \mathsf{p} \\ \bot & \vee & \mathsf{pred}^{\theta} = \mathsf{s} & \vee & \mathsf{pred}^{\theta} = \mathsf{p} \end{array} \right\}$$

Encoding Equivalence of Signatures

$$\left\{ \begin{array}{c} (\mathsf{plus}, 2) \\ (\mathsf{succ}, 1) \\ (\mathsf{pred}, 1) \end{array} \right\}^{\theta} \subseteq \left\{ \begin{array}{c} (\mathsf{+}, 2) \\ (\mathsf{s}, 1) \\ (\mathsf{p}, 1) \end{array} \right\}$$

$$\iff \bigwedge \left\{ \begin{array}{lllll} \mathsf{plus}^{\theta} = \mathsf{+} & \vee & \bot & \vee & \bot \\ \bot & \vee & \mathsf{succ}^{\theta} = \mathsf{s} & \vee & \mathsf{succ}^{\theta} = \mathsf{p} \\ \bot & \vee & \mathsf{pred}^{\theta} = \mathsf{s} & \vee & \mathsf{pred}^{\theta} = \mathsf{p} \end{array} \right\}$$

Note

- equivalence of $(\mathcal{F}, \mathcal{R})$ and $(\mathcal{G}, \mathcal{S})$ is decided by solving $\mathcal{F}^{\theta} = \mathcal{G} \wedge \mathcal{R}^{\theta} = \mathcal{S}$
- in same way, CS(C)TRSs and MSTRSs can be handled

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$
  if find$(i) \neq$ find(j)

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$
  if find$(i) \neq$ find(j)
    if $\mathcal{F}_i^\theta = \mathcal{F}_j$ and $\mathcal{R}_i^\theta \doteq \mathcal{R}_j$ for some θ takes 1ms

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$
  if find$(i) \neq$ find(j)
    if $\mathcal{F}_i^\theta = \mathcal{F}_j$ and $\mathcal{R}_i^\theta \doteq \mathcal{R}_j$ for some θ takes 1ms
      union(i, j)

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$
  if find$(i) \neq$ find(j)
    if $\mathcal{F}_i^\theta = \mathcal{F}_j$ and $\mathcal{R}_i^\theta \doteq \mathcal{R}_j$ for some θ takes 1ms
      union(i, j)

Estimation of Runtime

if $|\mathfrak{R}| = 1500$ then $\dfrac{1500 \times 1499}{2} \times 1$ ms ≈ 20 min

Duplicate Check I

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each (i, j) with $i < j$

  if find$(i) \neq$ find(j)

    if $\mathcal{F}_i^\theta = \mathcal{F}_j$ and $\mathcal{R}_i^\theta \doteq \mathcal{R}_j$ for some θ takes 1ms

      union(i, j)

Estimation of Runtime

if $|\mathfrak{R}| = 1500$ then $\dfrac{1500 \times 1499}{2} \times 1$ ms ≈ 20 min

Idea

use fingerprint function φ: $\varphi(\mathcal{R}) \neq \varphi(\mathcal{S}) \implies \mathcal{R} \not\approx \mathcal{S}$

$\varphi(\mathcal{R})$ in Current Implementation

$$\mathcal{R} = \left\{ \begin{array}{c} \overset{0}{\mathsf{a}}(\overset{1}{\mathsf{b}}(\overset{2}{\mathsf{c}}(\overset{0}{x}, \overset{1}{y}))) \to \overset{1}{\mathsf{b}}(\overset{0}{\mathsf{a}}(\overset{2}{\mathsf{c}}(\overset{1}{y}, \overset{1}{y}))) \\ \overset{0}{\mathsf{a}}(\overset{0}{x}) \to \overset{1}{\mathsf{b}}(\overset{0}{x}) \\ \overset{0}{\mathsf{b}}(\overset{0}{x}) \to \overset{1}{\mathsf{a}}(\overset{0}{x}) \end{array} \right\}$$

$\varphi(\mathcal{R})$ in Current Implementation

$$\mathcal{R} = \left\{ \begin{array}{c} \overset{0}{\mathsf{a}}(\overset{1}{\mathsf{b}}(\overset{2}{\mathsf{c}}(\overset{0}{x}, \overset{1}{y}))) \rightarrow \overset{1}{\mathsf{b}}(\overset{0}{\mathsf{a}}(\overset{2}{\mathsf{c}}(\overset{1}{y}, \overset{1}{y}))) \\ \overset{0}{\mathsf{a}}(\overset{0}{x}) \rightarrow \overset{1}{\mathsf{b}}(\overset{0}{x}) \\ \overset{0}{\mathsf{b}}(\overset{0}{x}) \rightarrow \overset{1}{\mathsf{a}}(\overset{0}{x}) \end{array} \right\}$$

$$\varphi(\mathcal{R}) = \mathsf{concat}(\mathsf{sort}(\begin{bmatrix} [0, 1, 2, 0, 1, 1, 0, 3, 1, 1], \\ [0, 0, 1, 0], \\ [0, 0, 1, 0] \end{bmatrix}))$$

$\varphi(\mathcal{R})$ in Current Implementation

$$\mathcal{R} = \left\{ \begin{array}{c} \overset{0}{\mathsf{a}}(\overset{1}{\mathsf{b}}(\overset{2}{\mathsf{c}}(\overset{0}{x}, \overset{1}{y}))) \rightarrow \overset{1}{\mathsf{b}}(\overset{0}{\mathsf{a}}(\overset{2}{\mathsf{c}}(\overset{1}{y}, \overset{1}{y}))) \\ \overset{0}{\mathsf{a}}(\overset{0}{x}) \rightarrow \overset{1}{\mathsf{b}}(\overset{0}{x}) \\ \overset{0}{\mathsf{b}}(\overset{0}{x}) \rightarrow \overset{1}{\mathsf{a}}(\overset{0}{x}) \end{array} \right\}$$

$$\begin{aligned} \varphi(\mathcal{R}) &= \mathsf{concat}(\mathsf{sort}(\begin{bmatrix} [0,1,2,0,1,\ 1,0,3,1,1], \\ [0,0,\ 1,0], \\ [0,0,\ 1,0] \end{bmatrix})) \\ &= [0,0,1,0,\quad 0,0,1,0,\quad 0,1,2,0,1,1,0,3,1,1] \end{aligned}$$

$\varphi(\mathcal{R})$ in Current Implementation

$$\mathcal{R} = \left\{ \begin{array}{c} \overset{0}{\mathsf{a}}(\overset{1}{\mathsf{b}}(\overset{2}{\mathsf{c}}(\overset{0}{x}, \overset{1}{y}))) \to \overset{1}{\mathsf{b}}(\overset{0}{\mathsf{a}}(\overset{2}{\mathsf{c}}(\overset{1}{y}, \overset{1}{y}))) \\ \overset{0}{\mathsf{a}}(\overset{0}{x}) \to \overset{1}{\mathsf{b}}(\overset{0}{x}) \\ \overset{0}{\mathsf{b}}(\overset{0}{x}) \to \overset{1}{\mathsf{a}}(\overset{0}{x}) \end{array} \right\}$$

$$\begin{aligned} \varphi(\mathcal{R}) &= \mathsf{concat}(\mathsf{sort}(\begin{bmatrix} [0, 1, 2, 0, 1, \ 1, 0, 3, 1, 1], \\ [0, 0, \ 1, 0], \\ [0, 0, \ 1, 0] \end{bmatrix})) \\ &= [0, 0, 1, 0, \quad 0, 0, 1, 0, \quad 0, 1, 2, 0, 1, 1, 0, 3, 1, 1] \end{aligned}$$

Theorem

$\varphi(\mathcal{R}) \neq \varphi(\mathcal{S}) \implies \mathcal{R} \not\approx \mathcal{S}$

$\varphi(\mathcal{R})$ in Current Implementation

$$\mathcal{R} = \left\{ \begin{array}{c} \overset{0}{\mathsf{a}}(\overset{1}{\mathsf{b}}(\overset{2}{\mathsf{c}}(\overset{0}{x}, \overset{1}{y}))) \to \overset{1}{\mathsf{b}}(\overset{0}{\mathsf{a}}(\overset{2}{\mathsf{c}}(\overset{1}{y}, \overset{1}{y}))) \\ \overset{0}{\mathsf{a}}(\overset{0}{x}) \to \overset{1}{\mathsf{b}}(\overset{0}{x}) \\ \overset{0}{\mathsf{b}}(\overset{0}{x}) \to \overset{1}{\mathsf{a}}(\overset{0}{x}) \end{array} \right\}$$

$$\varphi(\mathcal{R}) = \mathsf{concat}(\mathsf{sort}(\begin{bmatrix} [0,1,2,0,1,\ 1,0,3,1,1], \\ [0,0,\ 1,0], \\ [0,0,\ 1,0] \end{bmatrix}))$$

$$= [0,0,1,0,\ \ 0,0,1,0,\ \ 0,1,2,0,1,1,0,3,1,1]$$

Theorem

$\varphi(\mathcal{R}) \neq \varphi(\mathcal{S}) \implies \mathcal{R} \not\approx \mathcal{S}$ if $\mathcal{R}$ and $\mathcal{S}$ are variant-free (why?)

Duplicate Check II

given $(\mathcal{F}_1, \mathcal{R}_1), \ldots, (\mathcal{F}_n, \mathcal{R}_n)$

union-find maintains equivalence on $\{1, \ldots, n\}$

for each i

  compute variant-free version $\mathcal{R}_i'$ of $\mathcal{R}_i$

  $\vec{v}_i = \varphi(\mathcal{R}_i')$

for each (i, j) with $i < j$

  if find$(i) \neq$ find(j)

    if $\vec{v}_i = \vec{v}_j$ — takes < 0.01ms

      if $\mathcal{F}_i^\theta = \mathcal{F}_j$ and $\mathcal{R}_i^\theta \doteq \mathcal{R}_j$ for some θ — takes 1ms

        union(i, j)

Experimental Results

- Core i5-1340P @ 1.30GHz

Experimental Results

- Core i5-1340P @ 1.30GHz
- Z3 for finding renaming

Experimental Results

- Core i5-1340P @ 1.30GHz
- Z3 for finding renaming
- ARI contains 1218 (CS)(C)TRSs

Experimental Results

- Core i5-1340P @ 1.30GHz
- Z3 for finding renaming
- ARI contains 1218 (CS)(C)TRSs

DB	problems	duplicates	SMT calls	time (sec)
TRSs in COPS	577	13	16	**0.12**
ARI	1503	0	8	**0.37**

Experimental Results

- Core i5-1340P @ 1.30GHz
- Z3 for finding renaming
- ARI contains 1218 (CS)(C)TRSs

DB	problems	duplicates	SMT calls	time (sec)
TRSs in COPS	577	13	16	**0.12**
ARI	1503	0	8	**0.37**

Remark

some TRS in COPS is not variant-free; what about ARI?

Future Work I: MSTRSs

$$\{\mathsf{a} : \mathsf{A},\ \mathsf{b} : \mathsf{B}\}^{\theta} \approx \{\mathsf{0} : \mathsf{C},\ \mathsf{1} : \mathsf{D}\}$$

Future Work I: MSTRSs

$$\{\mathsf{a} : \mathsf{A},\ \mathsf{b} : \mathsf{B}\}^\theta \approx \{0 : \mathsf{C},\ 1 : \mathsf{D}\}$$

$$\iff \bigwedge \begin{Bmatrix} (\mathsf{a}^\theta = 0 \ \wedge \ \mathsf{A}^\theta = \mathsf{C}) \ \vee \ (\mathsf{a}^\theta = 1 \ \wedge \ \mathsf{A}^\theta = \mathsf{D}) \\ (\mathsf{b}^\theta = 0 \ \wedge \ \mathsf{B}^\theta = \mathsf{C}) \ \vee \ (\mathsf{b}^\theta = 1 \ \wedge \ \mathsf{B}^\theta = \mathsf{D}) \end{Bmatrix}$$

Future Work II: COM

indexes can be swapped

```
(format TRS :number 2)
(fun f 1)
(rule (f x) x     :index 1)
(rule (f (f x)) x :index 2)
```

is equivalent to

```
(format TRS :number 2)
(fun g 1)
(rule (g x) x     :index 2)
(rule (g (g x)) x :index 1)
```

Future Work III: INF

arguments can be swapped

```
(format TRS :problem infeasibility)
(fun a 0)
(fun b 0)
(infeasibility? (= a a) (= a b))
```

is equivalent to

```
(format TRS :problem infeasibility)
(fun a 0)
(fun b 0)
(infeasibility? (= a b) (= a a))
```

Future Work III: INF

arguments can be swapped

```
(format TRS :problem infeasibility)
(fun a 0)
(fun b 0)
(infeasibility? (= a a) (= a b))
```

is equivalent to

```
(format TRS :problem infeasibility)
(fun a 0)
(fun b 0)
(infeasibility? (= a b) (= a a))
```

probably infeasibility problems are generated by tools

Future Work IV

- duplicate check for LCTRSs (no idea about notion of equivalence)

Future Work IV

- duplicate check for LCTRSs (no idea about notion of equivalence)
- how does termCOMP gourp manage problems in TPDB?

Future Work IV

- duplicate check for LCTRSs (no idea about notion of equivalence)
- how does termCOMP gourp manage problems in TPDB?
- is duplicate check in P? or NP-complete?

Future Work IV

- duplicate check for LCTRSs (no idea about notion of equivalence)
- how does termCOMP gourp manage problems in TPDB?
- is duplicate check in P? or NP-complete?
- how to verify correctness of duplicate checkers?

  in COPS, some CTRS was wrongly tagged

Future Work IV

- duplicate check for LCTRSs (no idea about notion of equivalence)
- how does termCOMP gourp manage problems in TPDB?
- is duplicate check in P? or NP-complete?
- how to verify correctness of duplicate checkers?

  in COPS, some CTRS was wrongly tagged

Thanks for Your Attention!