Nils Küchenmeister
TU Dresden, Institute of Theoretical Computer Science, Knowledge-Based Systems Group

Reliance-based Static Analysis of Existential Rules

Diploma Thesis (defense) // Dresden, September 18, 2025

ρ_1 : $\mathit{student}(x) \longrightarrow \exists y.\ \mathit{author}(x, y) , \mathit{thesis}(y)$

Reliances [Krötzsch, KR'22]

- When applying ρ_1, a new unsatisfied match for ρ_2 may occur
- When applying ρ_2 before ρ_1, redundant nulls may be introduced

$\rho_1 : \mathit{student}(x) \longrightarrow \exists y.\, \mathit{author}(x, y),\ \mathit{thesis}(y)$

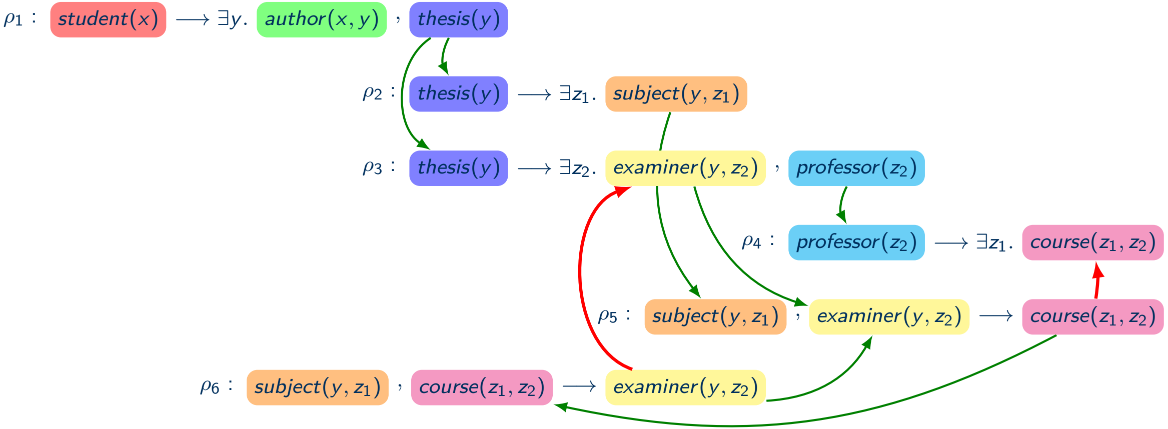
$\rho_2 : \mathit{thesis}(y) \longrightarrow \exists z_1.\, \mathit{subject}(y, z_1)$

$\rho_3 : \mathit{thesis}(y) \longrightarrow \exists z_2.\, \mathit{examiner}(y, z_2),\ \mathit{professor}(z_2)$

$\rho_4 : \mathit{professor}(z_2) \longrightarrow \exists z_1.\, \mathit{course}(z_1, z_2)$

Reliances [Krötzsch, KR'22]

- When applying ρ_1, a new unsatisfied match for ρ_2 may occur $\Rightarrow$ **positive reliance** $\rho_1 \overset{+}{\prec} \rho_2$
- When applying ρ_2 before ρ_1, redundant nulls may be introduced



Reliances [Krötzsch, KR'22]

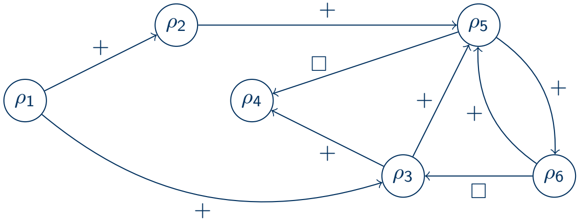
- When applying ρ_1, a new unsatisfied match for ρ_2 may occur $\Rightarrow$ **positive reliance** $\rho_1 \overset{+}{\prec} \rho_2$
- When applying ρ_2 before ρ_1, redundant nulls may be introduced $\Rightarrow$ **restraint reliance** $\rho_1 \overset{\square}{\prec} \rho_2$

Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

- vertices $\triangleq$ rules
- labelled edges $\triangleq$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

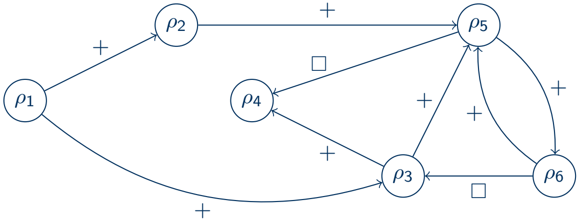
- vertices $\triangleq$ rules
- labelled edges $\triangleq$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**



Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

- vertices $\hat{=}$ rules
- labelled edges $\hat{=}$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

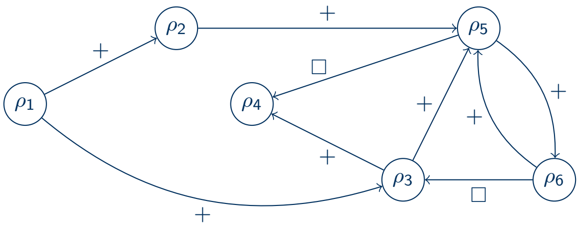
- A **core** is a minimal model (w/o redundancy).



Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

- vertices $\triangleq$ rules
- labelled edges $\triangleq$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

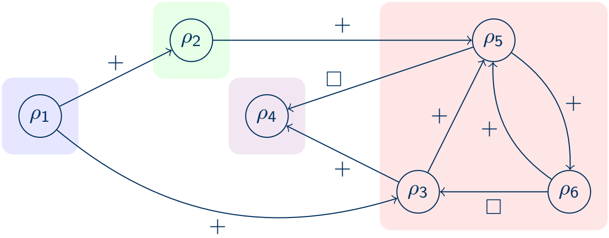
- A **core** is a minimal model (w/o redundancy).
- The **restricted chase** will yield a universal core, when rule selection never violates restraints.



Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

- vertices $\triangleq$ rules
- labelled edges $\triangleq$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

- A **core** is a minimal model (w/o redundancy).
- The **restricted chase** will yield a universal core, when rule selection never violates restraints.



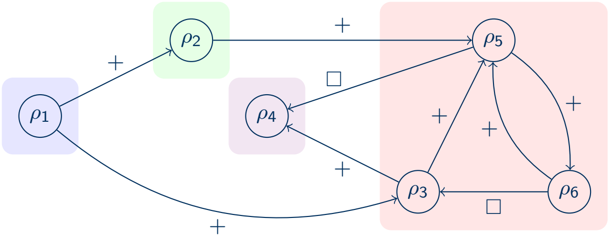
- There is an SCC $\{\rho_3, \rho_5, \rho_6\}$ with restraint edge in the reliance graph.

Reliance Graph $\mathcal{G}(R)$

[Krötzsch, KR'22]

- vertices ≙ rules
- labelled edges ≙ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

- A **core** is a minimal model (w/o redundancy).
- The **restricted chase** will yield a universal core, when rule selection never violates restraints.

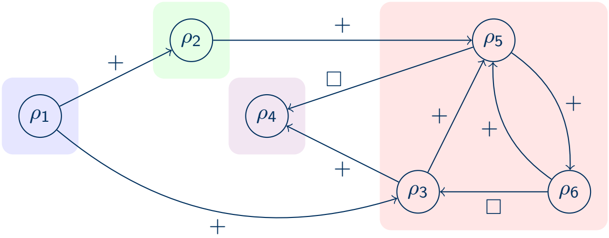


- There is an SCC $\{\rho_3, \rho_5, \rho_6\}$ with restraint edge in the reliance graph.
- The ruleset is not **core-stratified** !

Reliance Graph $\mathcal{G}(R)$ [Krötzsch, KR'22]

- vertices $\triangleq$ rules
- labelled edges $\triangleq$ reliance relations, with $+$ for **positive reliances** and $\square$ for **restraints**

- A **core** is a minimal model (w/o redundancy).
- The **restricted chase** will yield a universal core, when rule selection never violates restraints.



- There is an SCC $\{\rho_3, \rho_5, \rho_6\}$ with restraint edge in the reliance graph.
- The ruleset is not **core-stratified** ! $\rightarrow$ But a core can still be computed with the restricted chase !

Contractions

Definition

Let $\rho_1 : \phi_1 \to \psi_1$ and $\rho_2 : \phi_2 \to \psi_2$ be two rules with distinct variables.

$$\rho_3 : \mathit{thesis}(y) \longrightarrow \exists z_2.\ \mathit{examiner}(y, z_2), \mathit{professor}(z_2)$$

$$\rho_5 : \mathit{subject}(y, z_1), \mathit{examiner}(y, z_2) \longrightarrow \mathit{course}(z_1, z_2)$$

Contractions

Definition

Let $\rho_1 : \phi_1 \to \psi_1$ and $\rho_2 : \phi_2 \to \psi_2$ be two rules with distinct variables.

The *contracted rule* of ρ_1 and ρ_2 for a variable substitution $\eta : \boldsymbol{V} \to (\boldsymbol{V} \cup \boldsymbol{C})$ is

$$\rho^{\oplus}_{\rho_1,\rho_2,\eta} : \underbrace{\phi_1\eta \cup (\phi_2\eta \setminus \psi_1\eta)}_{=\phi^{\oplus}} \longrightarrow \underbrace{\psi_1\eta \cup \psi_2\eta \setminus \phi^{\oplus}}_{=\psi^{\oplus}}$$

$$\rho_3 : \mathit{thesis}(y) \longrightarrow \exists z_2.\ \mathit{examiner}(y, z_2),\ \mathit{professor}(z_2)$$

$\oplus$ $\qquad\qquad \eta = \mathrm{id}$

$$\rho_5 : \mathit{subject}(y, z_1),\ \mathit{examiner}(y, z_2) \longrightarrow \mathit{course}(z_1, z_2)$$

$$\rho^{\oplus}_{\rho_3,\rho_5,\eta} : \mathit{thesis}(y),\ \mathit{subject}(y, z_1) \longrightarrow \exists z_2.\ \mathit{examiner}(y, z_2),\ \mathit{professor}(z_2),\ \mathit{course}(z_1, z_2)$$

Contractions

Definition

We call $\rho_1 \oplus \rho_2$ their *contractions* with

$$\rho_1 \oplus \rho_2 := \{\, \rho^{\oplus}_{\rho_1,\rho_2,\eta} \mid \chi \subseteq \phi_2,\ \mu : \chi \to \psi_1,\ \eta : \boldsymbol{V} \to (\boldsymbol{V} \cup \boldsymbol{C}) \text{ leftmost MGU of } \chi\mu \text{ and } \chi,\ \mathit{var}_{\exists}(\psi_1\eta) \cap \mathit{var}(\phi_2\eta \setminus \psi_1\eta) = \varnothing \,\}$$

$$\rho_3 : \mathit{thesis}(y) \longrightarrow \exists z_2.\ \mathit{examiner}(y, z_2),\ \mathit{professor}(z_2)$$

$$\oplus$$

$$\rho_5 : \mathit{subject}(y, z_1),\ \mathit{examiner}(y, z_2) \longrightarrow \mathit{course}(z_1, z_2)$$

$$\rho_3 \oplus \rho_5 = \{\ \mathit{thesis}(y),\ \mathit{subject}(y, z_1) \longrightarrow \exists z_2.\ \mathit{examiner}(y, z_2),\ \mathit{professor}(z_2),\ \mathit{course}(z_1, z_2)\ \}$$

Enriching the ruleset with contractions

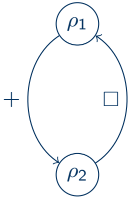
Definition

Let the contracted rules completion Comp(R) (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \text{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \text{Comp}(R) \times R$ with minimal renaming ζ, s.t. $\mathit{var}(\rho_1) \cap \mathit{var}(\rho_2\zeta) = \emptyset$, their contractions $\rho_1 \oplus \rho_2\zeta \subset \text{Comp}(R)$.

Symmetry

$$\rho_1 : \quad a(x) \longrightarrow \exists v.r(x, v), b(v)$$
$$\rho_2 : \quad r(x, y) \longrightarrow r(y, x)$$



reliance graph $\mathcal{G}(R)$

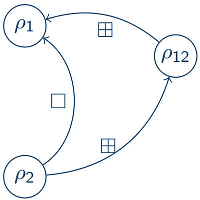
Definition

Let the contracted rules completion Comp(R) (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \text{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \text{Comp}(R) \times R$ with minimal renaming ζ, s.t. $var(\rho_1) \cap var(\rho_2 \zeta) = \emptyset$, their contractions $\rho_1 \oplus \rho_2 \zeta \subset \text{Comp}(R)$.

Symmetry

$$\rho_1 : \quad a(x) \longrightarrow \exists v.r(x, v), b(v)$$
$$\rho_2 : \quad r(x, y) \longrightarrow r(y, x)$$
$$\rho_{12} : \quad a(x) \longrightarrow \exists v.r(x, v), b(v), r(v, x)$$



contracted reliance graph $\mathcal{G}^{\oplus}(R)$

- There is no restraint cycle in $\mathcal{G}^{\oplus}(R)$. $\rightarrow$ We call R **contraction core-stratified** !

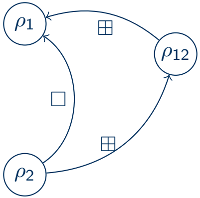
Definition

Let the contracted rules completion $\mathrm{Comp}(R)$ (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \mathrm{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \mathrm{Comp}(R) \times R$ with minimal renaming ζ, s.t. $var(\rho_1) \cap var(\rho_2\zeta) = \emptyset$, their contractions $\rho_1 \oplus \rho_2\zeta \subset \mathrm{Comp}(R)$.

Symmetry

$$\begin{aligned}
\rho_1 &: \quad a(x) \longrightarrow \exists v. r(x, v), b(v) \\
\rho_2 &: \quad r(x, y) \longrightarrow r(y, x) \\
\rho_{12} &: \quad a(x) \longrightarrow \exists v. r(x, v), b(v), r(v, x)
\end{aligned}$$



contracted reliance graph $\mathcal{G}^{\oplus}(R)$

- There is no restraint cycle in $\mathcal{G}^{\oplus}(R)$. $\to$ We call R **contraction core-stratified** !
- Here, $\mathrm{Comp}(R)$ is sure to be *finite*, as there is no positive reliance cycle in $\mathcal{G}(R)$.
  $\to$ But this is not generally the case. . .

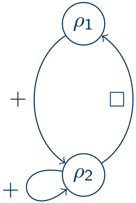
Definition

Let the contracted rules completion $\mathrm{Comp}(R)$ (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \mathrm{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \mathrm{Comp}(R) \times R$ with minimal renaming ζ, s.t. $var(\rho_1) \cap var(\rho_2\zeta) = \varnothing$, their contractions $\rho_1 \oplus \rho_2\zeta \subset \mathrm{Comp}(R)$.

Transitivity

$$\rho_1 : \quad a(x) \longrightarrow \exists v.r(x, v), b(v)$$
$$\rho_2 : \quad r(x, y), r(y, z) \longrightarrow r(x, z)$$



reliance graph $\mathcal{G}(R)$

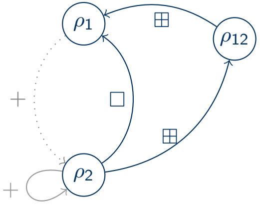
Definition

Let the contracted rules completion Comp(R) (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \mathrm{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \mathrm{Comp}(R) \times R$ with minimal renaming ζ, s.t. $var(\rho_1) \cap var(\rho_2\zeta) = \varnothing$, their contractions $\rho_1 \oplus \rho_2\zeta \subset \mathrm{Comp}(R)$.

Transitivity

$$\rho_1 : \quad a(x) \longrightarrow \exists v.r(x,v), b(v)$$
$$\rho_2 : \quad r(x,y), r(y,z) \longrightarrow r(x,z)$$
$$\rho_{12} : \quad a(x), r(x',x) \longrightarrow \exists v.r(x,v), b(v), r(x',v)$$



contracted reliance graph $\mathcal{G}^{\oplus}(R)$

- There is no restraint cycle in $\mathcal{G}^{\oplus}(R)$. → We call R **contraction core-stratified** !

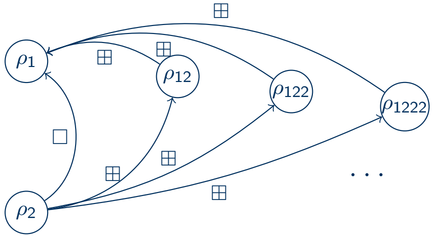
Definition

Let the contracted rules completion $\mathrm{Comp}(R)$ (for a ruleset R) be the smallest rulesets, s.t.

1. $R \subseteq \mathrm{Comp}(R)$, and
2. for any pair $\langle \rho_1, \rho_2 \rangle \in \overset{+}{\prec} \cap \mathrm{Comp}(R) \times R$ with minimal renaming ζ, s.t. $var(\rho_1) \cap var(\rho_2\zeta) = \emptyset$, their contractions $\rho_1 \oplus \rho_2\zeta \subset \mathrm{Comp}(R)$.

Transitivity

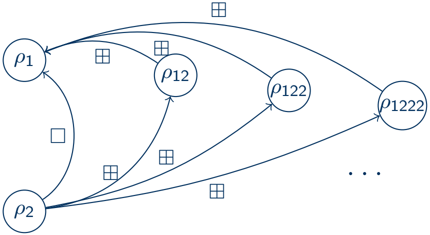
$$\rho_1 : \quad a(x) \longrightarrow \exists v.r(x,v), b(v)$$
$$\rho_2 : \quad r(x,y), r(y,z) \longrightarrow r(x,z)$$
$$\rho_{12} : \quad a(x), r(x',x) \longrightarrow \exists v.r(x,v), b(v), r(x',v)$$
$$\ldots$$



- There is no restraint cycle in $\mathcal{G}^{\oplus}(R)$. $\rightarrow$ We call R **contraction core-stratified** !
- In this case, there is a positive reliance cycle in $\mathcal{G}(R)$ and $\mathrm{Comp}(R)$ is *infinite*.

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$

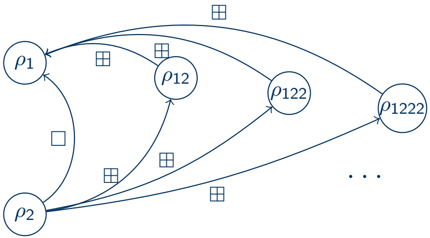
(I)



- Structure of $\mathcal{G}^{\oplus}(R)$ seams regular

$\rightarrow$ Is it sufficient to consider a finite subgraph?

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$



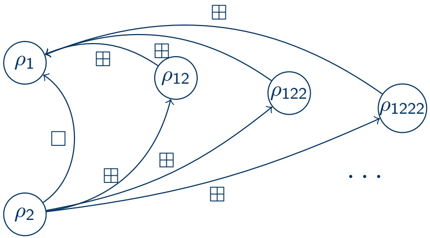
- Structure of $\mathcal{G}^{\oplus}(R)$ seams regular

→ Is it sufficient to consider a finite subgraph?

Finite contracted reliance graph $\mathcal{G}^{[\oplus]}(R)$

- Assign each (contracted) rule a set orig($\rho^{\oplus}$) of sequences over alphabet $\boldsymbol{\Sigma}$ to track information regarding their formation
- **Idea:** Stop "lengthening" $\rho^{\oplus}$ when all words in orig($\rho^{\oplus}$) are of the form $uwqwv$

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$



- Structure of $\mathcal{G}^{\oplus}(R)$ seams regular

$\rightarrow$ Is it sufficient to consider a finite subgraph?

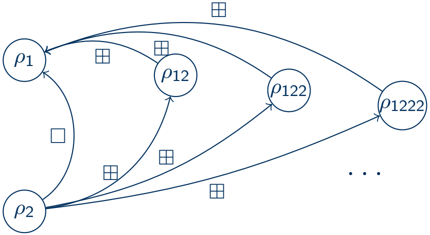
What do we need to show?

- If there is a restraint cycle in $\mathcal{G}^{\oplus}(R)$, then there is a restraint cycle in $\mathcal{G}^{[\oplus]}(R)$
- Rules labeled *uwqw* and *uw* can be continued in the same way
- Restraints depend only on first/last letter of labels

Finite contracted reliance graph $\mathcal{G}^{[\oplus]}(R)$

- Assign each (contracted) rule a set orig$(\rho^{\oplus})$ of sequences over alphabet Σ to track information regarding their formation
- **Idea:** Stop "lengthening" $\rho^{\oplus}$ when all words in orig$(\rho^{\oplus})$ are of the form *uwqwv*

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$



- Structure of $\mathcal{G}^{\oplus}(R)$ seams regular

$\rightarrow$ Is it sufficient to consider a finite subgraph?

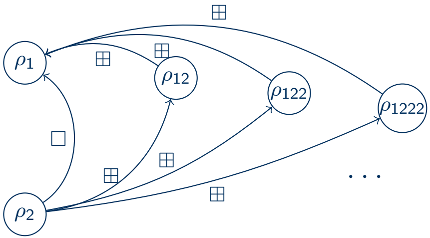
What do we need to show?

- If there is a restraint cycle in $\mathcal{G}^{\oplus}(R)$, then there is a restraint cycle in $\mathcal{G}^{[\oplus]}(R)$
- Rules labeled *uwqw* and *uw* can be continued in the same way
- Restraints depend only on first/last letter of labels

Finite contracted reliance graph $\mathcal{G}^{[\oplus]}(R)$

- Assign each (contracted) rule a set orig$(\rho^{\oplus})$ of sequences over alphabet $\boldsymbol{\Sigma}$ to track information regarding their formation
- **Idea:** Stop “lengthening” $\rho^{\oplus}$ when all words in orig$(\rho^{\oplus})$ are of the form *uwqwv*
- **Question:** Which information must be tracked in letters of $\boldsymbol{\Sigma}$?

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$



- Structure of $\mathcal{G}^{\oplus}(R)$ seams regular

$\rightarrow$ Is it sufficient to consider a finite subgraph?

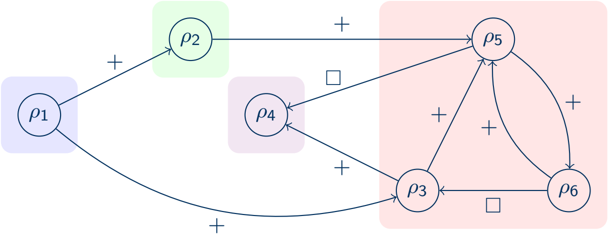
Approaches?

- $\boldsymbol{\Sigma} = R$ does not work, as the unifier can change internal structure in contractions
- Additionally tracking change of internal structure of component rules is not enough, as "surrounding" atoms in the contraction may prevent positive reliance to other rules
- $\boldsymbol{\Sigma} = 2^{\mathcal{H}} \times 2^{\mathcal{H}}$

Finite contracted reliance graph $\mathcal{G}^{[\oplus]}(R)$

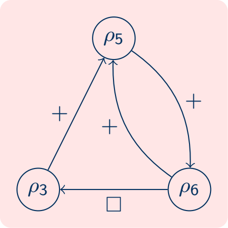
- Assign each (contracted) rule a set $\mathrm{orig}(\rho^{\oplus})$ of sequences over alphabet $\boldsymbol{\Sigma}$ to track information regarding their formation
- **Idea:** Stop "lengthening" $\rho^{\oplus}$ when all words in $\mathrm{orig}(\rho^{\oplus})$ are of the form $uwqwv$
- **Question:** Which information must be tracked in letters of $\boldsymbol{\Sigma}$?

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)



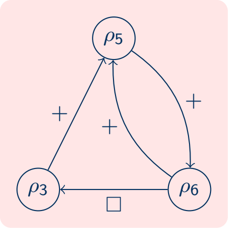
Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)

- The head pieces $\tilde{\mathcal{H}}$ are
  - $\exists z_2.\ \mathit{examiner}(y, z_2)\ ,\ \mathit{professor}(z_2)$
  - $\mathit{course}(z_1, z_2)$
  - $\mathit{examiner}(y, z_2)$



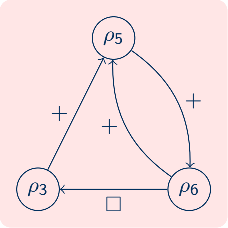
Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)

- The head pieces $\tilde{\mathcal{H}}$ are
  - $\exists z_2.$ $examiner(y, z_2)$, $professor(z_2)$
  - $course(z_1, z_2)$
  - $examiner(y, z_2)$
- Variables are $\{y, z_1, z_2\}$
  - Replace the existentials in $\tilde{\mathcal{H}}$ with fresh variables
  - Consider all mappings of frontier variables in $\tilde{\mathcal{H}}$ to other variables



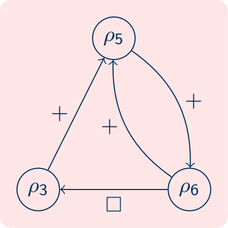
Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)

- The head pieces $\tilde{\mathcal{H}}$ are
  - $\exists z_2.\ examiner(y, z_2),\ professor(z_2)$
  - $course(z_1, z_2)$
  - $examiner(y, z_2)$
- Variables are $\{y, z_1, z_2\}$
  - Replace the existentials in $\tilde{\mathcal{H}}$ with fresh variables
  - Consider all mappings of frontier variables in $\tilde{\mathcal{H}}$ to other variables
- The head images $\mathcal{H}$ are
  - $examiner(y, \tilde{z}_2), professor(\tilde{z}_2)$
  - $examiner(z_1, \tilde{z}_2), professor(\tilde{z}_2)$
  - $examiner(z_2, \tilde{z}_2), professor(\tilde{z}_2)$
  - 9 options for $course(z_1, z_2)$
  - 9 options for $examiner(y, z_2)$



Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)

- The head pieces $\tilde{\mathcal{H}}$ are
  - $\exists z_2.$ $\mathit{examiner}(y, z_2)$, $\mathit{professor}(z_2)$
  - $\mathit{course}(z_1, z_2)$
  - $\mathit{examiner}(y, z_2)$
- Variables are $\{y, z_1, z_2\}$
  - Replace the existentials in $\tilde{\mathcal{H}}$ with fresh variables
  - Consider all mappings of frontier variables in $\tilde{\mathcal{H}}$ to other variables
- The head images $\mathcal{H}$ are
  - $\mathit{examiner}(y, \tilde{z}_2), \mathit{professor}(\tilde{z}_2)$
  - $\mathit{examiner}(z_1, \tilde{z}_2), \mathit{professor}(\tilde{z}_2)$
  - $\mathit{examiner}(z_2, \tilde{z}_2), \mathit{professor}(\tilde{z}_2)$
  - 9 options for $\mathit{course}(z_1, z_2)$
  - 9 options for $\mathit{examiner}(y, z_2)$



$\rightarrow$ The alphabet $\Sigma = 2^{\mathcal{H}} \times 2^{\mathcal{H}}$ tracks

. . . the head images contributed to the head of the contracted rule by the second rule

. . . the head images that are necessarily satisfied

Deciding acyclicity of $\mathcal{G}^{\oplus}(R)$ (III)

- The head pieces $\tilde{\mathcal{H}}$ are
  - $\exists z_2.$ $\mathit{examiner}(y, z_2)$, $\mathit{professor}(z_2)$
  - $\mathit{course}(z_1, z_2)$
  - $\mathit{examiner}(y, z_2)$

Repetition

$$\mathrm{orig}(\rho_{3565}) = \{\langle\{\{\exists\tilde{z}_2.\mathit{examiner}(y,\tilde{z}_2), \mathit{professor}(\tilde{z}_2)\}\}, \{\{\exists\tilde{z}_2.\mathit{examiner}(y,\tilde{z}_2), \mathit{professor}(\tilde{z}_2)\}, \{\mathit{examiner}(y, z_2)\}\}\rangle$$
$$\langle\{\mathit{course}(z_1, z_2)\}, \{\mathit{course}(z_1, z_2)\}\rangle$$
$$\langle\{\{\mathit{examiner}(y, z_2)\}\}, \{\{\exists\tilde{z}_2.\mathit{examiner}(y,\tilde{z}_2), \mathit{professor}(\tilde{z}_2)\}, \{\mathit{examiner}(y, z_2)\}\}\rangle$$
$$\langle\{\mathit{course}(z_1, z_2)\}, \{\mathit{course}(z_1, z_2)\}\rangle\ \}$$

$\rightarrow$ The alphabet $\boldsymbol{\Sigma} = 2^{\mathcal{H}} \times 2^{\mathcal{H}}$ tracks

. . . the head images contributed to the head of the contracted rule by the second rule
. . . the head images that are necessarily satisfied

The selection strategy

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Strategy:

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i

Strategy:

Assumptions:

R is . . .

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i

$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\mathrm{Comp}(R)$ in chase step i

$\rightarrow$ fixed-point computation

Strategy:

Assumptions:

R is . . .

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i

$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\mathrm{Comp}(R)$ in chase step i

$\rightarrow$ fixed-point computation

Strategy:

1. Apply triggers for datalog rules as early as possible
   $\rightarrow$ This cannot cause (additional) alternative matches!

Assumptions:

R is . . .

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i
$\Lambda_i^{\oplus}$ is the set of all triggers for rules from Comp(R) in chase step i
$\rightarrow$ fixed-point computation

Strategy:

1. Apply triggers for datalog rules as early as possible
   $\rightarrow$ This cannot cause (additional) alternative matches!

- Let $R_1, \ldots, R_n$ be a topological sorting of the SCCs of the reliance graph $\mathcal{G}(R)$

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i

$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\mathrm{Comp}(R)$ in chase step i

$\rightarrow$ fixed-point computation

Strategy:

1. Apply triggers for datalog rules as early as possible

   $\rightarrow$ This cannot cause (additional) alternative matches!

- Let $R_1, \ldots, R_n$ be a topological sorting of the SCCs of the reliance graph $\mathcal{G}(R)$
- $\tilde{R}$ is the union of SCCs without restraint edges

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i

$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\text{Comp}(R)$ in chase step i

$\rightarrow$ fixed-point computation

Strategy:

1. Apply triggers for datalog rules as early as possible
   $\rightarrow$ This cannot cause (additional) alternative matches!

- Let $R_1, \ldots, R_n$ be a topological sorting of the SCCs of the reliance graph $\mathcal{G}(R)$
- $\tilde{R}$ is the union of SCCs without restraint edges
- Only consider triggers for rules that belong to R_i, if there is no $j < i$, s.t. there are triggers for R_j

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i
$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\mathrm{Comp}(R)$ in chase step i
$\rightarrow$ fixed-point computation

Strategy:

1. Apply triggers for datalog rules as early as possible
   $\rightarrow$ This cannot cause (additional) alternative matches!

- Let $R_1, \ldots, R_n$ be a topological sorting of the SCCs of the reliance graph $\mathcal{G}(R)$
- $\tilde{R}$ is the union of SCCs without restraint edges
- Only consider triggers for rules that belong to R_i, if there is no $j < i$, s.t. there are triggers for R_j

2. In case $R_i \subseteq \tilde{R}$, select any of these triggers
   $\rightarrow$ Like with core-stratified rulesets, no restraints are violated!

Assumptions:

R is ...

- universally terminating and
- SCC-contraction core-stratified

Λ_i is the set of all triggers for rules from R in chase step i
$\Lambda_i^{\oplus}$ is the set of all triggers for rules from $\mathrm{Comp}(R)$ in chase step i
$\rightarrow$ fixed-point computation

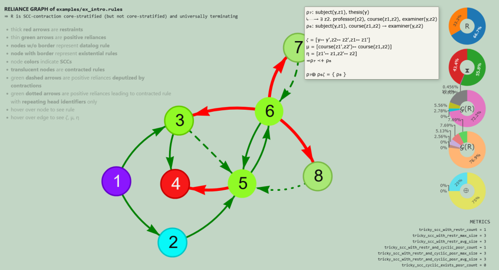
Strategy:

1. Apply triggers for datalog rules as early as possible
   $\rightarrow$ This cannot cause (additional) alternative matches!

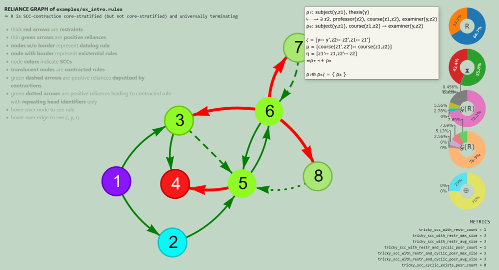
- Let $R_1, \ldots, R_n$ be a topological sorting of the SCCs of the reliance graph $\mathcal{G}(R)$
- $\tilde{R}$ is the union of SCCs without restraint edges
- Only consider triggers for rules that belong to R_i, if there is no $j < i$, s.t. there are triggers for R_j

2. In case $R_i \subseteq \tilde{R}$, select any of these triggers
   $\rightarrow$ Like with core-stratified rulesets, no restraints are violated!
3. Otherwise, select only the triggers of R_i that are leader of contracted triggers whose rules are not restrained by other contracted triggers
   $\rightarrow$ Applying such a trigger will not lead to future appearance of new triggers that restrain the selected trigger!
   - As the restraint relation of $\mathcal{G}^{\oplus}(R)$ is acyclic by assumption, at least one selection is always possible!

Prototype implementation

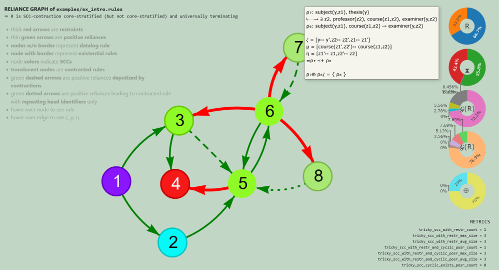


- 1,913 lines of Python code
- Prototypical
  (Not fully optimized for efficiency yet)
- Interactive visualization of the **finite contracted reliance graph**



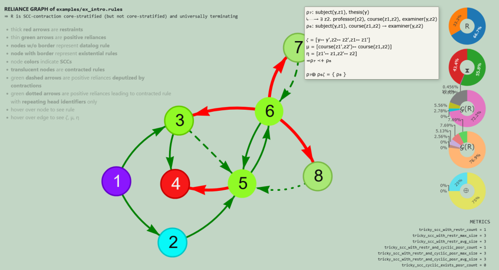
- 1,913 lines of Python code
- Prototypical
  (Not fully optimized for efficiency yet)
- Interactive visualization of the **finite contracted reliance graph**

1. Compute $\mathcal{G}(R)$ like the prior VLog implementation [Ivliev, 21]
   - local optimization: depth-first exploration of atom mappings with early backtracking,
   - global optimization: rule pairs are discarded when head/body don't share predicates, and reliances are memoized (disregarding concrete predicate and variable names)



- 1,913 lines of Python code
- Prototypical
  (Not fully optimized for efficiency yet)
- Interactive visualization of the **finite contracted reliance graph**

1. Compute $\mathcal{G}(R)$ like the prior VLog implementation [Ivliev, 21]
   - local optimization: depth-first exploration of atom mappings with early backtracking,
   - global optimization: rule pairs are discarded when head/body don't share predicates, and reliances are memoized (disregarding concrete predicate and variable names)
2. Tarjan's linear-time SCC decomposition



- 1,913 lines of Python code
- Prototypical
  (Not fully optimized for efficiency yet)
- Interactive visualization of the **finite contracted reliance graph**

1. Compute $\mathcal{G}(R)$ like the prior VLog implementation [Ivliev, 21]
   - local optimization: depth-first exploration of atom mappings with early backtracking,
   - global optimization: rule pairs are discarded when head/body don't share predicates, and reliances are memoized (disregarding concrete predicate and variable names)
2. Tarjan's linear-time SCC decomposition
3. Add contractions for. . . restrained rules → **contraction core-stratified** ?
   . . . existential cyclic positive reliances
   → detect absence of **unbounded contractions** ⇒ universal termination

Evaluation

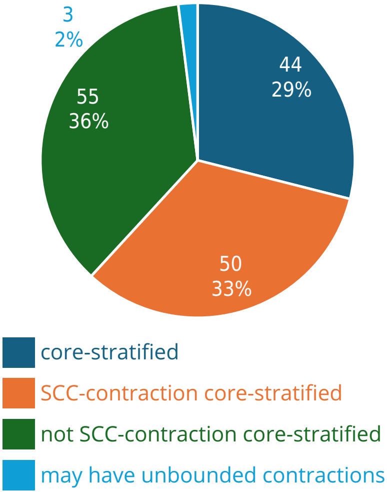
Evaluation

- *Prototypical* Python implementation to perform static analysis
  1) Build the initial reliance graph and decompose it into SCCs
  2) For each SCC with restraint, add appropriate contracted rules and check for restraint cycles
  3) For each positive-reliance-SCC with existentials, add contractions until either we are sure that the completion is finite or a full “cycle” was traversed

- Dataset: *Oxford Ontology Repository*
  - `https://www.cs.ox.ac.uk/isg/ontologies/`
  - same rulesets that were previously used to evaluate core-stratification [Ivliev, 21 & González et al., ISWC’22]
  - 201 of 787 ontologies were translated into existential rules

- Experiments:
  - on Wille server
  - with 25min timeout
    - 49 timeouts (during reliance graph computation)
    - mostly for rulesets with over 12,000 rules

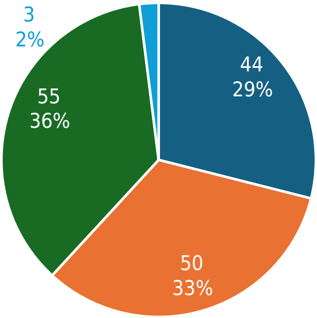
Evaluation: prevalence of contraction core-stratification (II)



- Roughly 1/3 are core-stratified.
  - This is consistent with prior observations [Ivliev, 21 & González et al., ISWC'22]
- Roughly an additional 1/3 are SCC-contraction core-stratified.
  - New static analysis covers a significant portion of the evaluated real-world rulesets!
- Of the 55 not SCC-contraction core-stratified rulesets,
  - 87% have restraint cycles and
  - 11% have self-restraining rules.

  → Identifiable before starting to add contractions!

Evaluation: prevalence of contraction core-stratification (II)

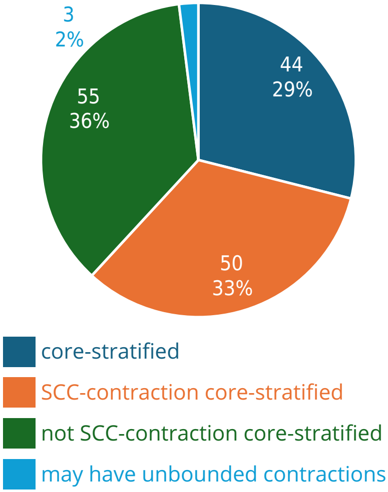


- core-stratified
- SCC-contraction core-stratified
- not SCC-contraction core-stratified
- may have unbounded contractions

- Roughly 1/3 are core-stratified.
  - This is consistent with prior observations [Ivliev, 21 & González et al., ISWC'22]
- Roughly an additional 1/3 are SCC-contraction core-stratified.
  - New static analysis covers a significant portion of the evaluated real-world rulesets!
- Of the 55 not SCC-contraction core-stratified rulesets,
  - 87% have restraint cycles and
  - 11% have self-restraining rules.

  → Identifiable before starting to add contractions!

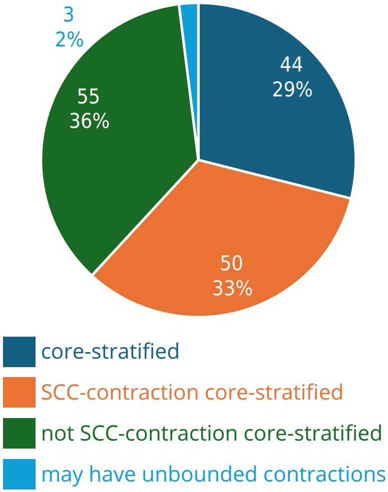
Evaluation: prevalence of contraction core-stratification (II)



- Roughly 1/3 are core-stratified.
  - This is consistent with prior observations [Ivliev, 21 & González et al., ISWC'22]
- Roughly an additional 1/3 are SCC-contraction core-stratified.
  - New static analysis covers a significant portion of the evaluated real-world rulesets!
- Of the 55 not SCC-contraction core-stratified rulesets,
  - 87% have restraint cycles and
  - 11% have self-restraining rules.

  $\rightarrow$ Identifiable before starting to add contractions!

Evaluation: prevalence of contraction core-stratification (II)

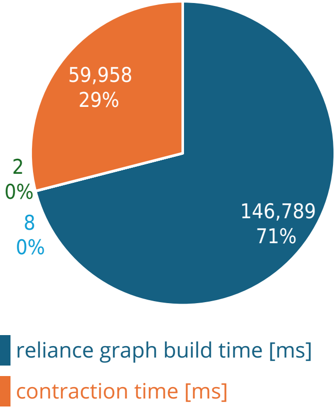


- Roughly 1/3 are core-stratified.
  - This is consistent with prior observations [Ivliev, 21 & González et al., ISWC'22]
- Roughly an additional 1/3 are SCC-contraction core-stratified.
  - New static analysis covers a significant portion of the evaluated real-world rulesets!
- Of the 55 not SCC-contraction core-stratified rulesets,
  - 87% have restraint cycles and
  - 11% have self-restraining rules.

  → Identifiable before starting to add contractions!

Evaluation: extra analysis cost

- (Only considering rulesets requiring ≥ 1 contracted rule.)

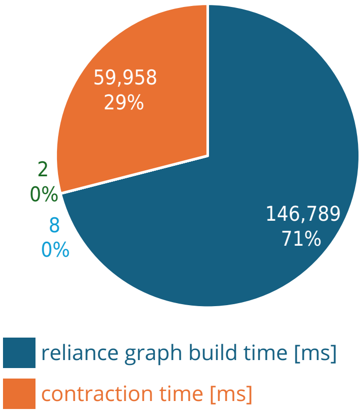


- In average, most of the analysis time is needed to build the initial reliance graph.
- Trend is not uniform: some rulesets require significantly more contractions:

- Additional time to check whether R may have unbounded contractions is negligible.

Evaluation: extra analysis cost (III)

- (Only considering rulesets requiring ≥ 1 contracted rule.)

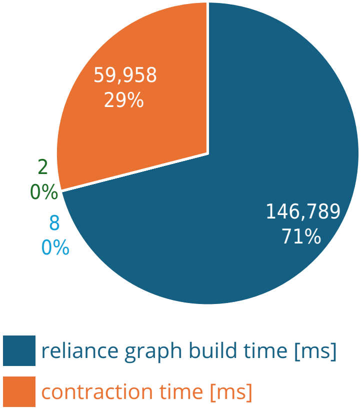


- In average, most of the analysis time is needed to build the initial reliance graph.
- Trend is not uniform: some rulesets require significantly more contractions:

- Additional time to check whether R may have unbounded contractions is negligible.

Evaluation: extra analysis cost (III)

- (Only considering rulesets requiring ≥ 1 contracted rule.)

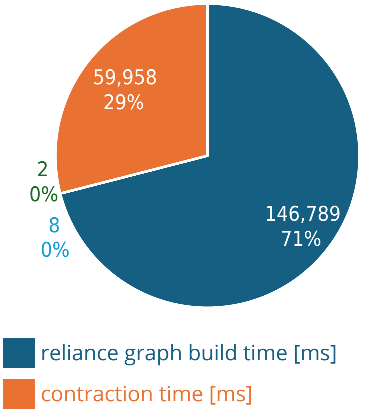


- In average, most of the analysis time is needed to build the initial reliance graph.
- Trend is not uniform: some rulesets require significantly more contractions:

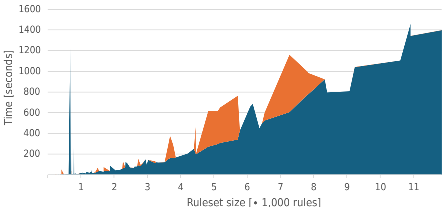
- Additional time to check whether R may have unbounded contractions is negligible.

Evaluation: extra analysis cost

- (Only considering rulesets requiring ≥ 1 contracted rule.)



- In average, most of the analysis time is needed to build the initial reliance graph.
- Trend is not uniform: some rulesets require significantly more contractions:



- Additional time to check whether *R* may have unbounded contractions is negligible.

Conclusion

Conclusion

Summary

- SCCs of the reliance graph with restraints prevent core-stratification
- Contractions deputize the combined effect of applying positively relying rules
- Look ahead by adding contractions $\Rightarrow$ absence of restraint cycles

Future work

- Continue working out proof details for $\mathcal{G}^{[\oplus]}(R)$
- Study use of contractions to detect universal termination in detail
- *Evaluation:*
  - conduct more targeted experiments to clarify runtime behavior
- *Implementation:*
  - Improve efficiency
  - Possible Nemo integration

Pro

+ Generality
  - not restricted to special cases
+ Covers 2/3 of evaluated rulesets

  $\rightarrow$ find core with restricted chase

Con

— Additional analysis effort
  - worst-case: *2Exp*

— Complicated selection strategy
  - fixed-point computation

Conclusion

Summary

- SCCs of the reliance graph with restraints prevent core-stratification
- Contractions deputize the combined effect of applying positively relying rules
- Look ahead by adding contractions $\Rightarrow$ absence of restraint cycles

Future work

- Continue working out proof details for $\mathcal{G}^{[\oplus]}(R)$
- Study use of contractions to detect universal termination in detail
- *Evaluation:*
  - conduct more targeted experiments to clarify runtime behavior
- *Implementation:*
  - Improve efficiency
  - Possible Nemo integration

Pro

+ Generality
  - not restricted to special cases
+ Covers $^2/_3$ of evaluated rulesets

  $\rightarrow$ find core with restricted chase

Con

— Additional analysis effort
  - worst-case: *2Exp*

— Complicated selection strategy
  - fixed-point computation

Conclusion

Summary

- SCCs of the reliance graph with restraints prevent core-stratification
- Contractions deputize the combined effect of applying positively relying rules
- Look ahead by adding contractions $\Rightarrow$ absence of restraint cycles

Future work

- Continue working out proof details for $\mathcal{G}^{[\oplus]}(R)$
- Study use of contractions to detect universal termination in detail
- *Evaluation:*
  - conduct more targeted experiments to clarify runtime behavior
- *Implementation:*
  - Improve efficiency
  - Possible Nemo integration

Pro

\+ Generality
  - not restricted to special cases

\+ Covers 2/3 of evaluated rulesets
  - $\rightarrow$ find core with restricted chase

Con

— Additional analysis effort
  - worst-case: *2Exp*

— Complicated selection strategy
  - fixed-point computation

Thank you for your attention!

Errata

- There is a mistake in the implementation of the $\textsc{check}^{\square}_{self}$ function, that causes certain self-restraints of rules with multiple pieces to be missed
  - like: $a(x) \longrightarrow \exists v.b(x), r(x, v), c(v)$