Reasoning over Existential Rules with Acyclicity Notions and the Datalog-first Restricted Chase

David Carral

TECHNISCHE UNIVERSITÄT DRESDEN

Slides available at https://iccl.inf.tu-dresden.de/web/Existential-rules-acyclicity

Preliminaries

Existential Rules

$$\forall x, y, z \,.\, \Big(\mathsf{HasParent}(x, y) \wedge \mathsf{HasSister}(y, z) \rightarrow \mathsf{HasAunt}(x, z)\Big)$$

$$\forall x \,.\, \Big(\mathsf{Human}(x) \rightarrow \exists y \,.\, \big(\mathsf{HasParent}(x, y) \wedge \mathsf{Human}(y)\big)\Big)$$

$$\forall x, y, w \,.\, \Big(\mathsf{P}(x, \mathsf{a}, y) \wedge \mathsf{R}(y, w) \wedge \mathsf{S}(w, x) \rightarrow \exists v \,.\, \big(\mathsf{R}(w, v) \wedge \mathsf{A}(v)\big)\Big)$$

Existential Rules

$$\mathsf{HasParent}(x, y) \wedge \mathsf{HasSister}(y, z) \rightarrow \mathsf{HasAunt}(x, z)$$

$$\mathsf{Human}(x) \rightarrow \exists y\,.\,\mathsf{HasParent}(x, y) \wedge \mathsf{Human}(y)$$

$$\mathsf{P}(x, \mathsf{a}, y) \wedge \mathsf{R}(y, w) \wedge \mathsf{S}(w, x) \rightarrow \exists v\,.\,\mathsf{R}(w, v) \wedge \mathsf{A}(v)$$

Existential Rules

$$\mathsf{HasParent}(x, y) \wedge \mathsf{HasSister}(y, z) \rightarrow \mathsf{HasAunt}(x, z)$$

$$\mathsf{Human}(x) \rightarrow \exists y \,.\, \mathsf{HasParent}(x, y) \wedge \mathsf{Human}(y)$$

$$\mathsf{P}(x, \mathsf{a}, y) \wedge \mathsf{R}(y, w) \wedge \mathsf{S}(w, x) \rightarrow \exists v \,.\, \mathsf{R}(w, v) \wedge \mathsf{A}(v)$$

Facts

$$\mathsf{HasFriend}(\mathsf{stan}, \mathsf{kyle})$$

$$\mathsf{P}(\mathsf{a}, \mathsf{c}, \mathsf{d})$$

Existential Rules

$$\mathsf{HasParent}(x, y) \land \mathsf{HasSister}(y, z) \rightarrow \mathsf{HasAunt}(x, z)$$

$$\mathsf{Human}(x) \rightarrow \exists y \,.\, \mathsf{HasParent}(x, y) \land \mathsf{Human}(y)$$

$$\mathsf{P}(x, \mathsf{a}, y) \land \mathsf{R}(y, w) \land \mathsf{S}(w, x) \rightarrow \exists v \,.\, \mathsf{R}(w, v) \land \mathsf{A}(v)$$

Facts

$$\mathsf{HasFriend}(\mathsf{stan}, \mathsf{kyle})$$

$$\mathsf{P}(\mathsf{a}, \mathsf{c}, \mathsf{d})$$

BCQs

$$\exists x, y \,.\, \mathsf{HasConflictOfInterest}(x, y)$$

$$\exists x, y, z, w \,.\, \mathsf{P}(x, y, z) \land \mathsf{R}(x, w) \land \mathsf{A}(w)$$

The Chase Algorithm

$\mathrm{Features}(x, y) \longrightarrow \mathrm{Actor}(y)$

$\mathrm{ActsIn}(x, y) \longrightarrow \mathrm{Features}(y, x)$

$\mathrm{DirectedBy}(x, y) \longrightarrow \mathrm{Directs}(y, x)$

$\mathrm{Directs}(x, y) \wedge \mathrm{Features}(y, z) \longrightarrow \mathrm{DirectsActor}(x, z)$

$\mathrm{Director}(\mathrm{spielberg})$

$\mathrm{ActsIn}(\mathrm{judeLaw}, \mathrm{ai})$

$\mathrm{DirectedBy}(\mathrm{ai}, \mathrm{spielberg})$

The Chase Algorithm

Features(x, y) ⟶ Actor(y)

DirectedBy(x, y) ⟶ Directs(y, x)

ActsIn(x, y) ⟶ Features(y, x)

Directs(x, y) ∧ Features(y, z) ⟶ DirectsActor(x, z)

judeLaw

spielberg

Director(spielberg)

ActsIn(judeLaw, ai)

ai

DirectedBy(ai, spielberg)

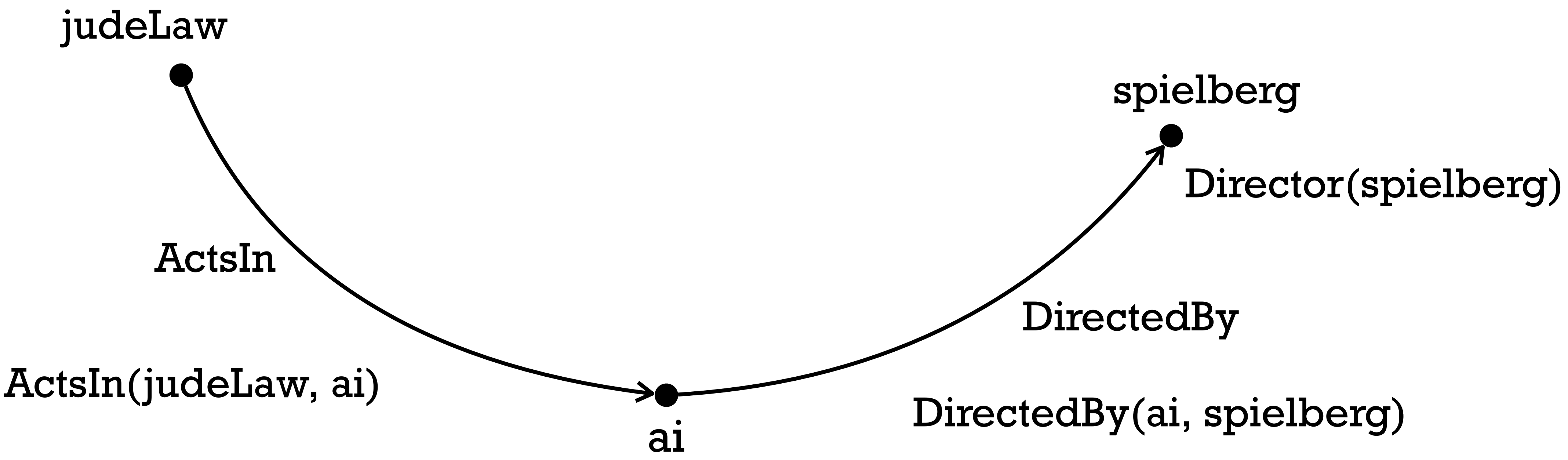
The Chase Algorithm

$\text{Features}(x, y) \longrightarrow \text{Actor}(y)$

$\text{ActsIn}(x, y) \longrightarrow \text{Features}(y, x)$

$\text{DirectedBy}(x, y) \longrightarrow \text{Directs}(y, x)$

$\text{Directs}(x, y) \wedge \text{Features}(y, z) \longrightarrow \text{DirectsActor}(x, z)$



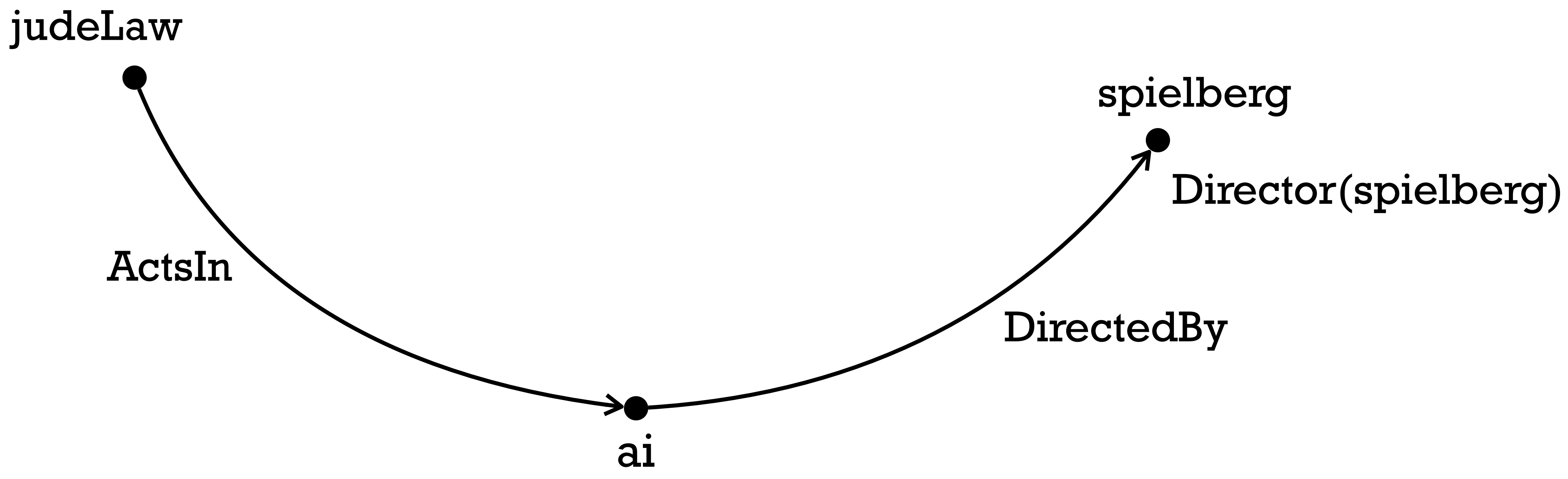
The Chase Algorithm

$\mathsf{Features}(x, y) \longrightarrow \mathsf{Actor}(y)$

$\mathsf{ActsIn}(x, y) \longrightarrow \mathsf{Features}(y, x)$

$\mathsf{DirectedBy}(x, y) \longrightarrow \mathsf{Directs}(y, x)$

$\mathsf{Directs}(x, y) \wedge \mathsf{Features}(y, z) \longrightarrow \mathsf{DirectsActor}(x, z)$



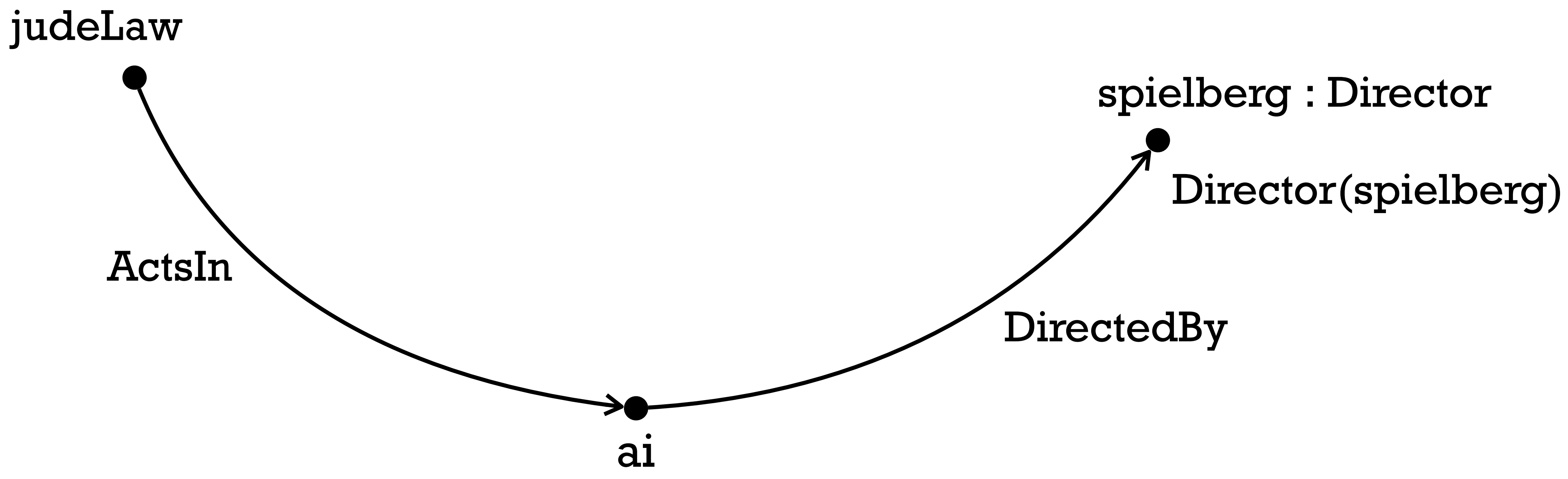
The Chase Algorithm

$\mathsf{Features}(x, y) \longrightarrow \mathsf{Actor}(y)$

$\mathsf{ActsIn}(x, y) \longrightarrow \mathsf{Features}(y, x)$

$\mathsf{DirectedBy}(x, y) \longrightarrow \mathsf{Directs}(y, x)$

$\mathsf{Directs}(x, y) \wedge \mathsf{Features}(y, z) \longrightarrow \mathsf{DirectsActor}(x, z)$



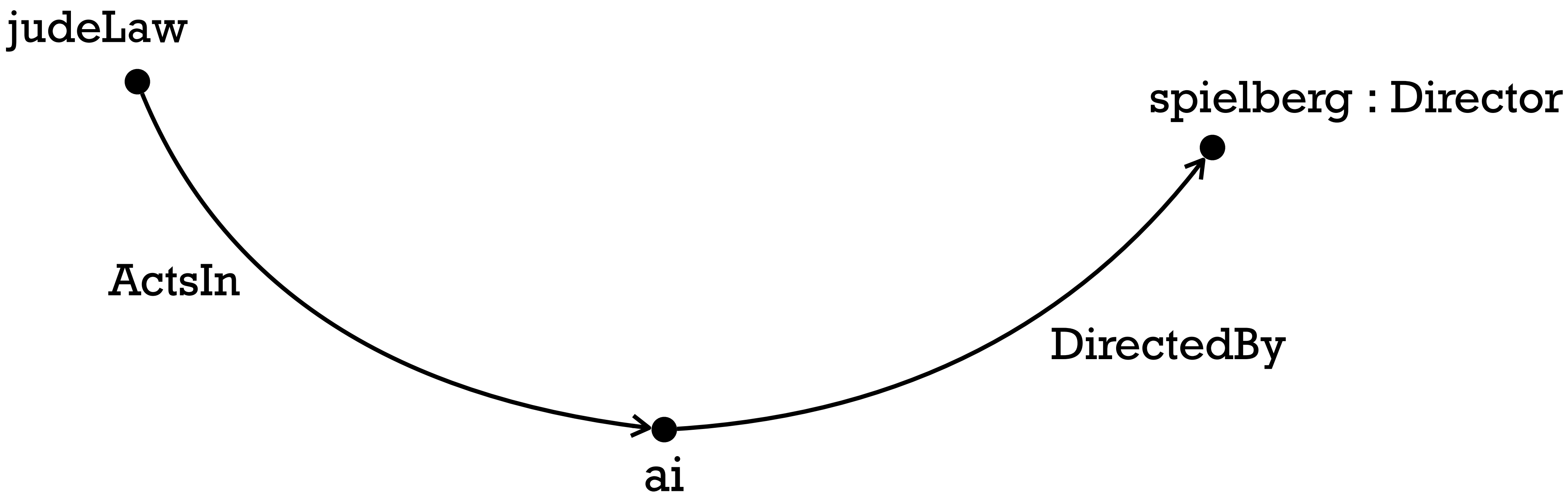
The Chase Algorithm

$\mathrm{Features}(x, y) \longrightarrow \mathrm{Actor}(y)$

$\mathrm{ActsIn}(x, y) \longrightarrow \mathrm{Features}(y, x)$

$\mathrm{DirectedBy}(x, y) \longrightarrow \mathrm{Directs}(y, x)$

$\mathrm{Directs}(x, y) \wedge \mathrm{Features}(y, z) \longrightarrow \mathrm{DirectsActor}(x, z)$



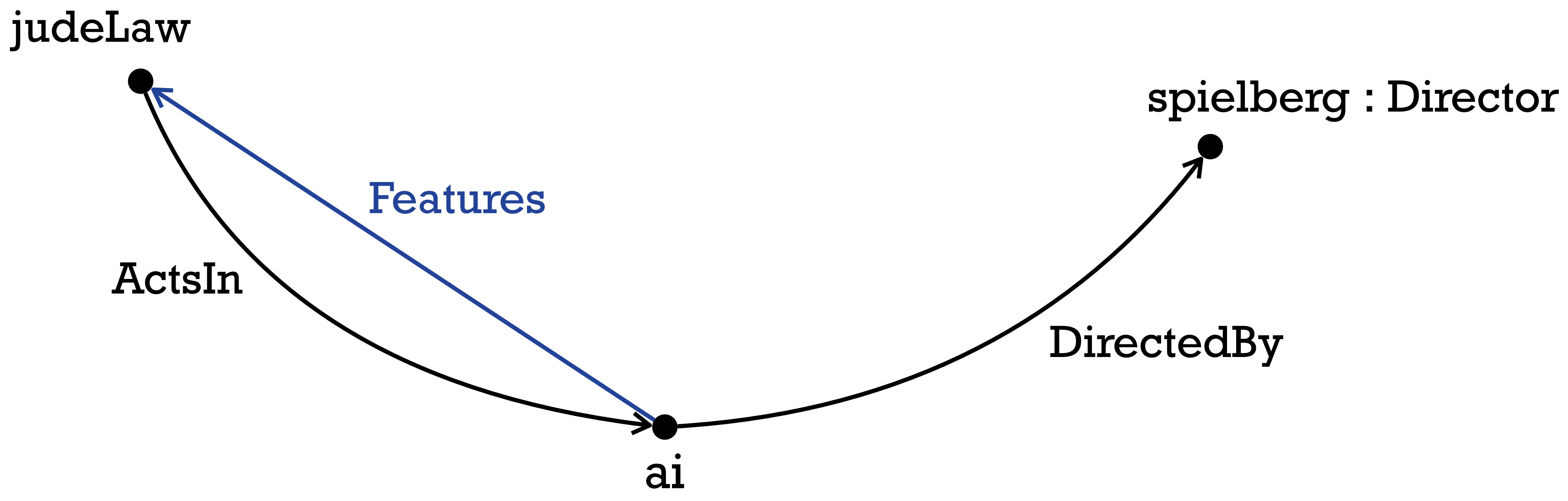
The Chase Algorithm

$\mathrm{Features}(x, y) \longrightarrow \mathrm{Actor}(y)$

$\mathrm{ActsIn}(x, y) \longrightarrow \mathrm{Features}(y, x)$

$\mathrm{DirectedBy}(x, y) \longrightarrow \mathrm{Directs}(y, x)$

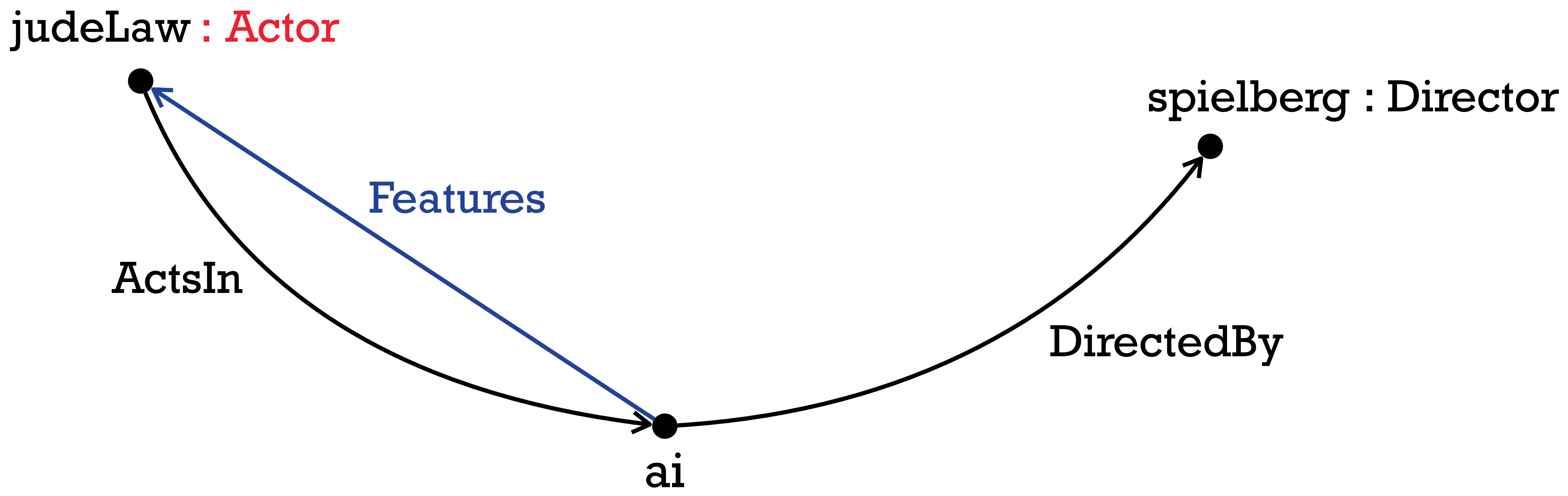
$\mathrm{Directs}(x, y) \wedge \mathrm{Features}(y, z) \longrightarrow \mathrm{DirectsActor}(x, z)$



The Chase Algorithm

$\mathrm{Features}(x, y) \longrightarrow \mathrm{Actor}(y)$ $\quad$ $\mathrm{DirectedBy}(x, y) \longrightarrow \mathrm{Directs}(y, x)$

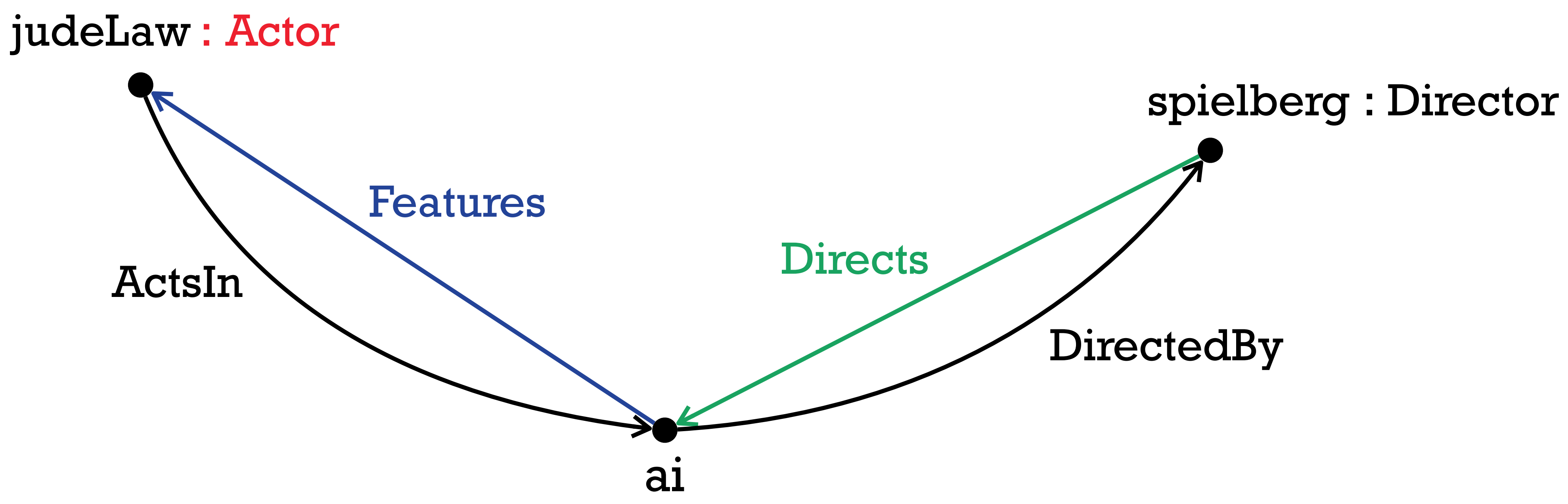
$\mathrm{ActsIn}(x, y) \longrightarrow \mathrm{Features}(y, x)$ $\quad$ $\mathrm{Directs}(x, y) \wedge \mathrm{Features}(y, z) \longrightarrow \mathrm{DirectsActor}(x, z)$



The Chase Algorithm

$\mathrm{Features}(x, y) \longrightarrow \mathrm{Actor}(y)$ $\quad$ $\mathrm{DirectedBy}(x, y) \longrightarrow \mathrm{Directs}(y, x)$

$\mathrm{ActsIn}(x, y) \longrightarrow \mathrm{Features}(y, x)$ $\quad$ $\mathrm{Directs}(x, y) \wedge \mathrm{Features}(y, z) \longrightarrow \mathrm{DirectsActor}(x, z)$



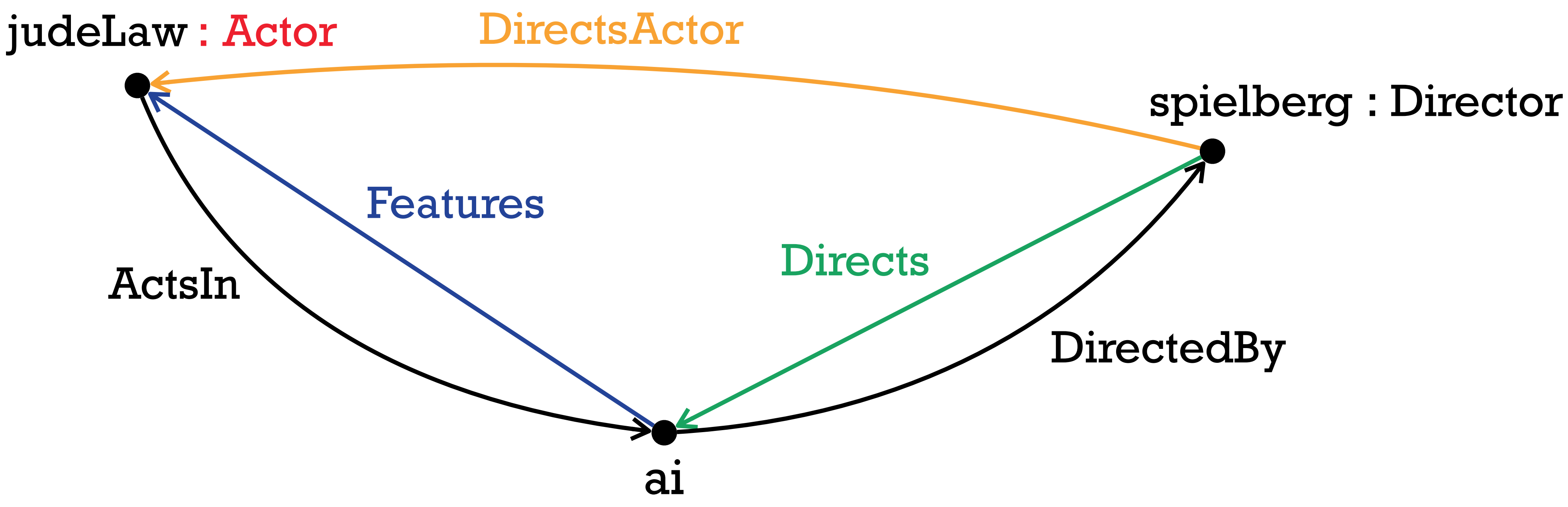
The Chase Algorithm

$\text{Features}(x, y) \longrightarrow \text{Actor}(y)$

$\text{ActsIn}(x, y) \longrightarrow \text{Features}(y, x)$

$\text{DirectedBy}(x, y) \longrightarrow \text{Directs}(y, x)$

$\text{Directs}(x, y) \wedge \text{Features}(y, z) \longrightarrow \text{DirectsActor}(x, z)$



The Skolem Chase

$\text{Bicycle}(x) \longrightarrow \exists v\ .\ \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w\ .\ \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$

b : Bicycle

The Skolem Chase

$$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, f_v(x)) \wedge \mathrm{Wheel}(f_v(x)) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, f_w(x)) \wedge \mathrm{Bicycle}(f_w(x)) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

b : Bicycle

The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

b : Bicycle

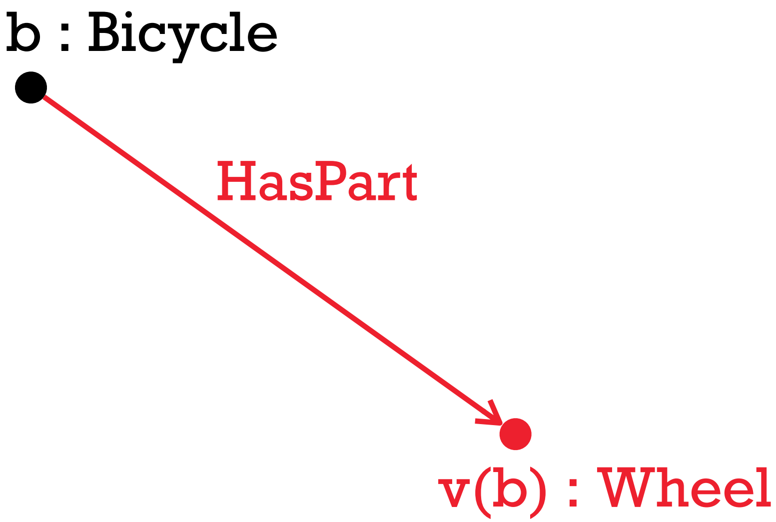
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



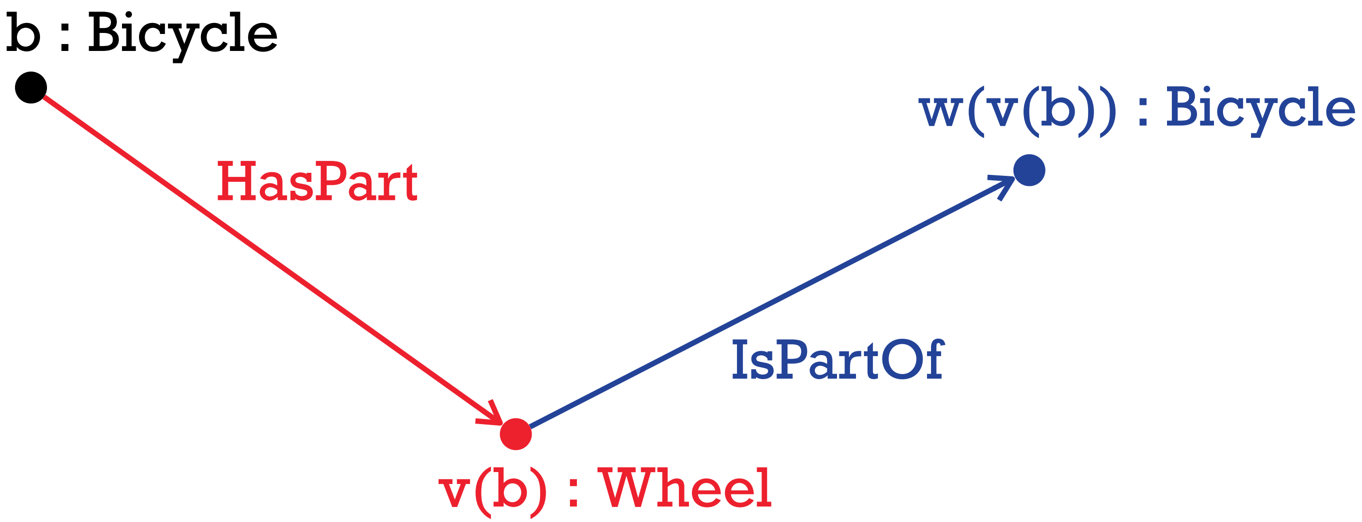
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



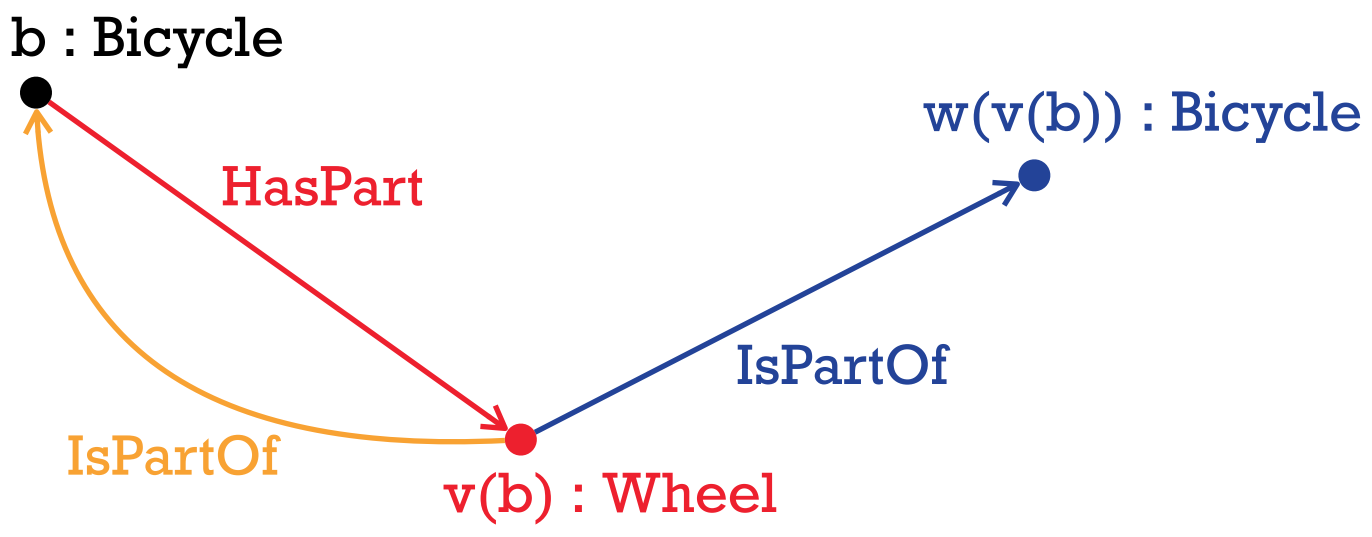
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



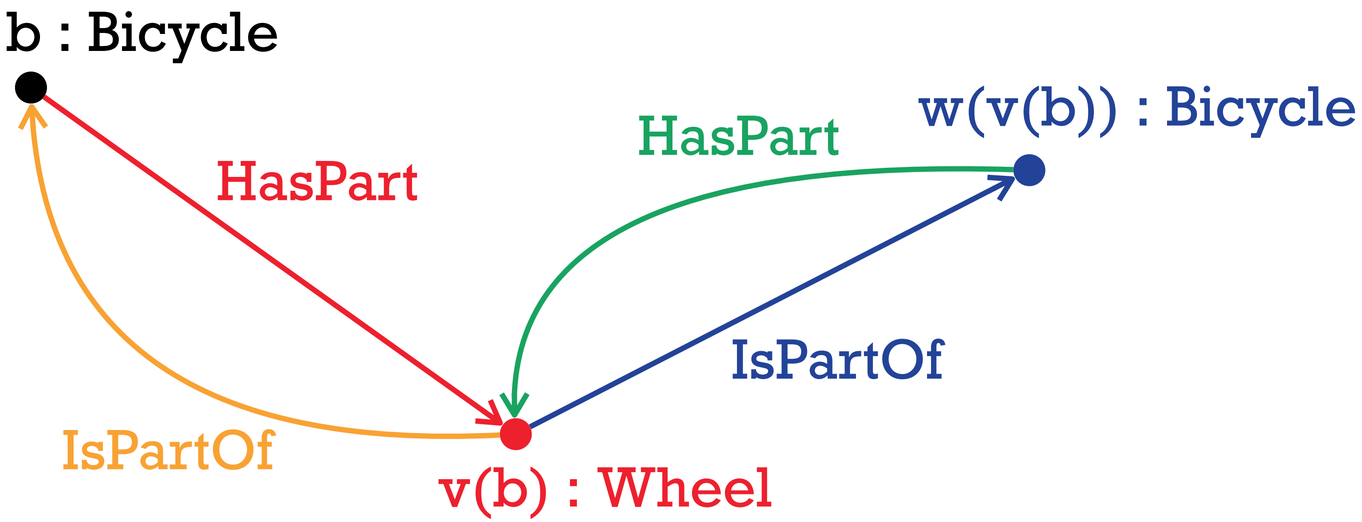
The Skolem Chase

$\text{Bicycle}(x) \longrightarrow \text{HasPart}(x, v(x)) \wedge \text{Wheel}(v(x))$

$\text{Wheel}(x) \longrightarrow \text{IsPartOf}(x, w(x)) \wedge \text{Bicycle}(w(x))$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



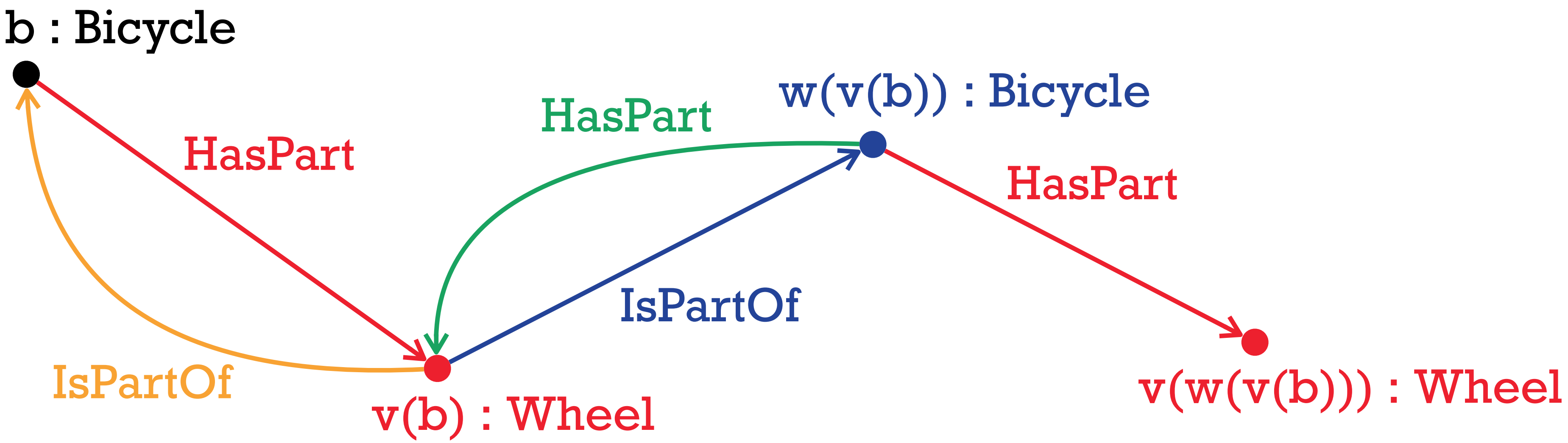
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



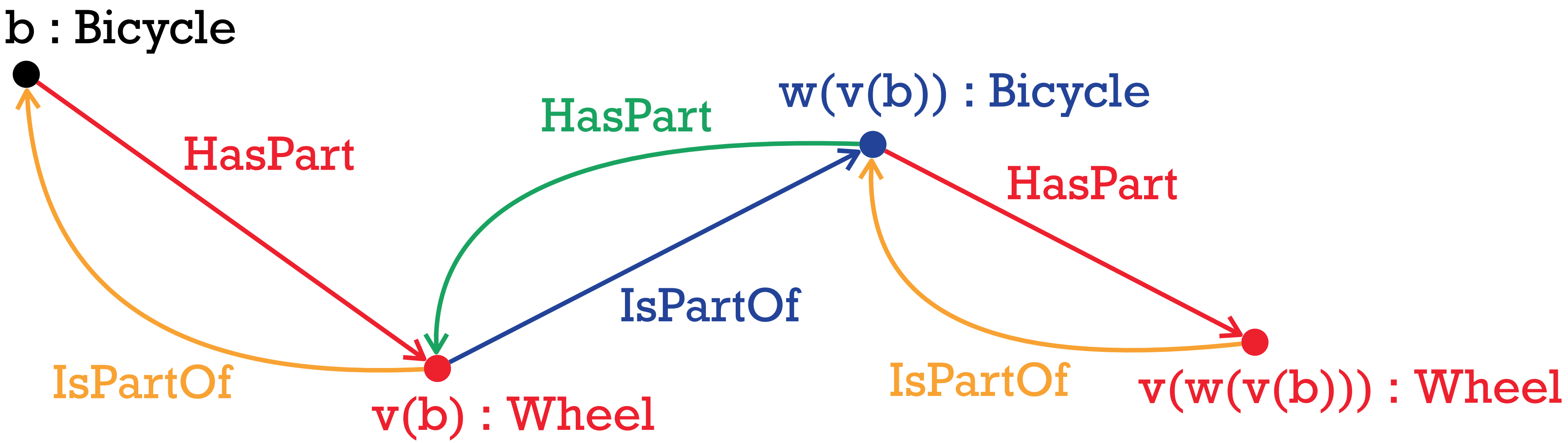
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

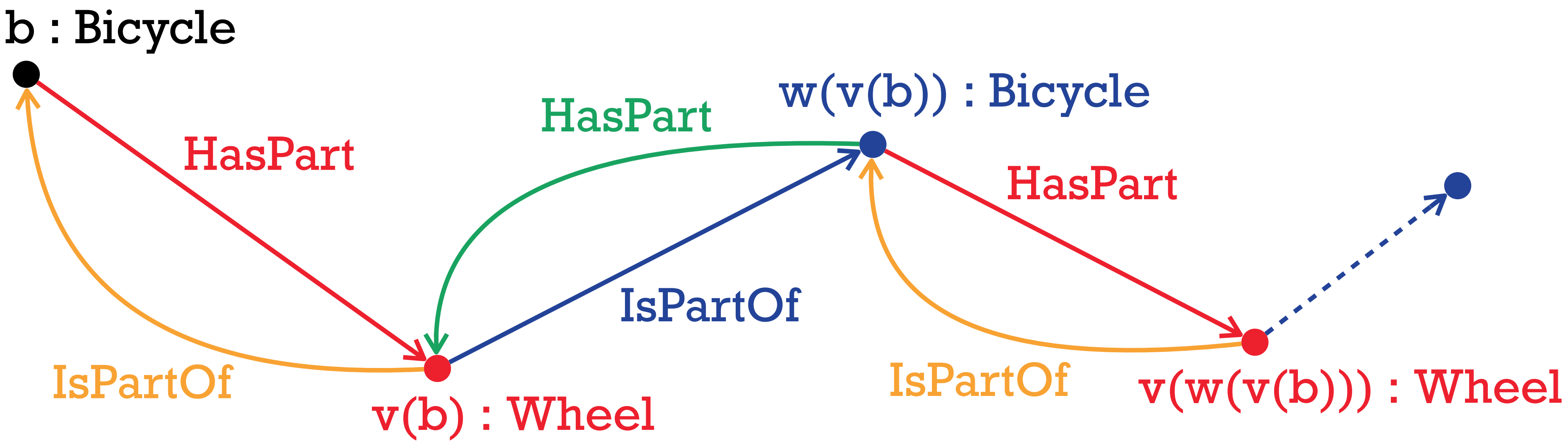
$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The Skolem Chase

$$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x)) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

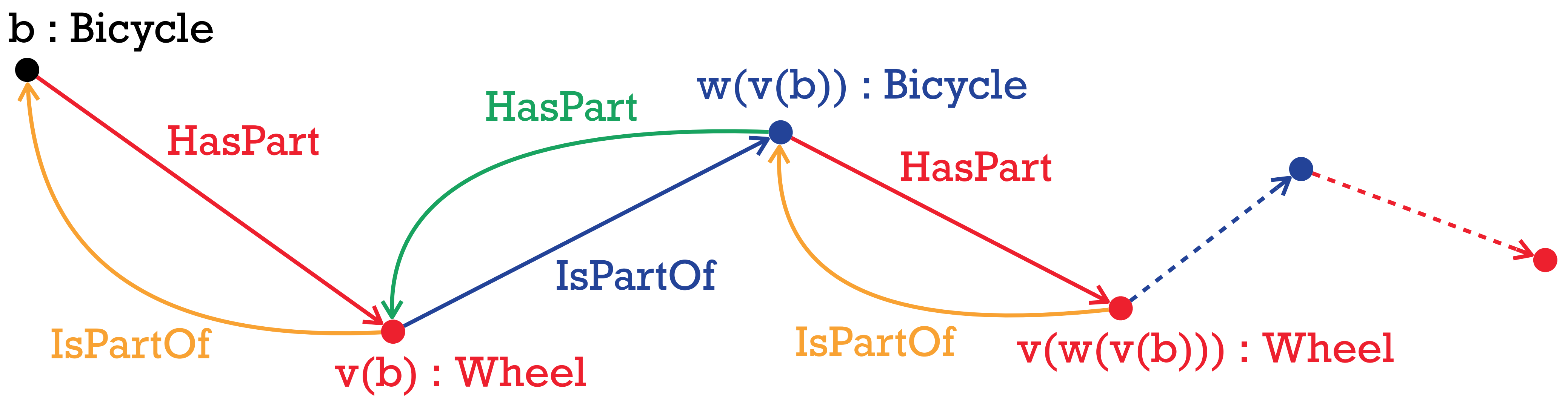
$$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x)) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$ $\qquad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$ $\qquad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



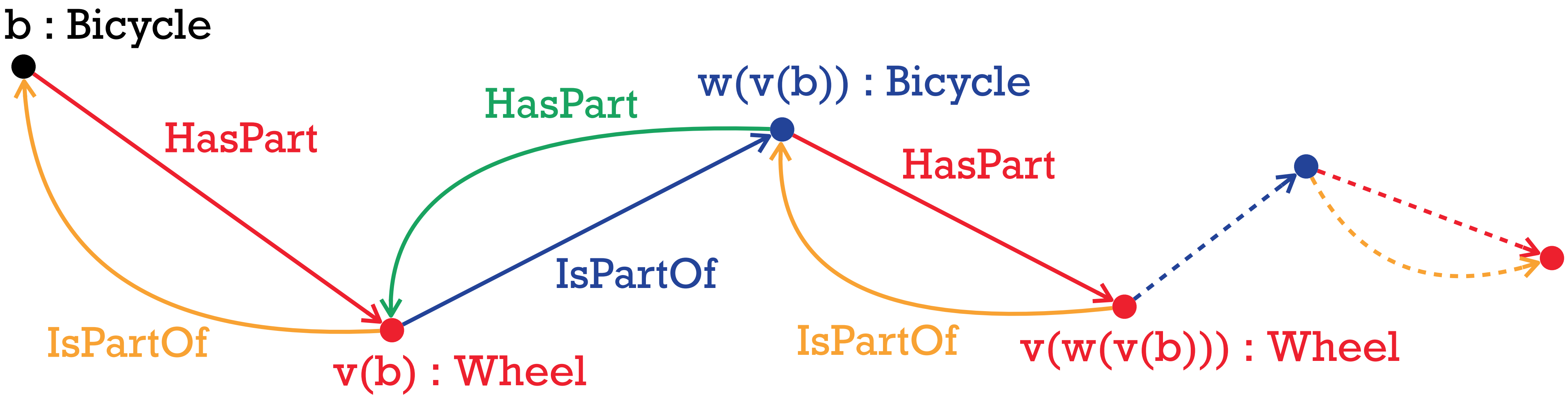
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



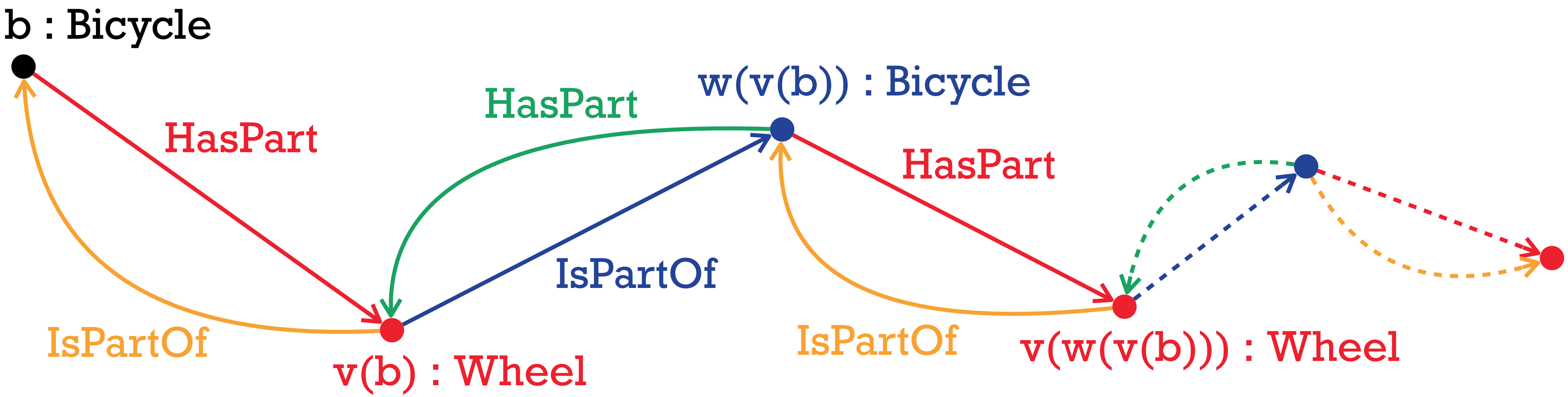
The Skolem Chase

$\mathrm{Bicycle}(x) \longrightarrow \mathrm{HasPart}(x, v(x)) \wedge \mathrm{Wheel}(v(x))$

$\mathrm{Wheel}(x) \longrightarrow \mathrm{IsPartOf}(x, w(x)) \wedge \mathrm{Bicycle}(w(x))$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

The Restricted Chase

$$\mathrm{Bicycle}(x) \longrightarrow \exists v \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists w \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

$$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

b : Bicycle

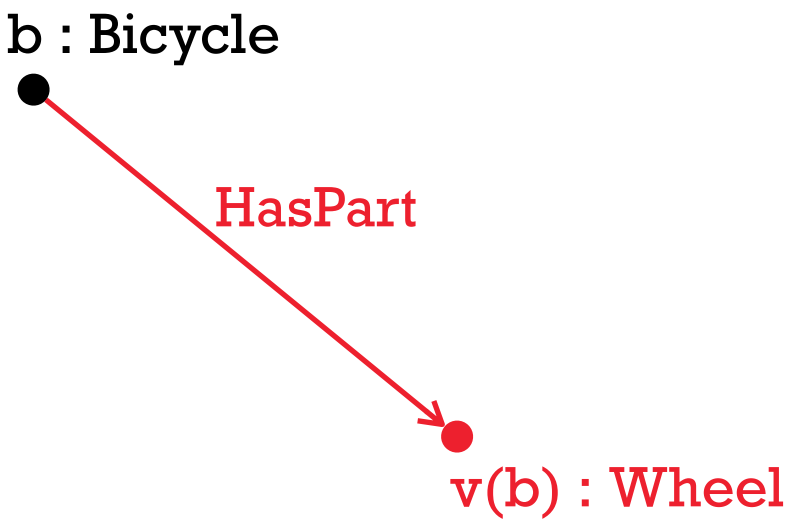
The Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



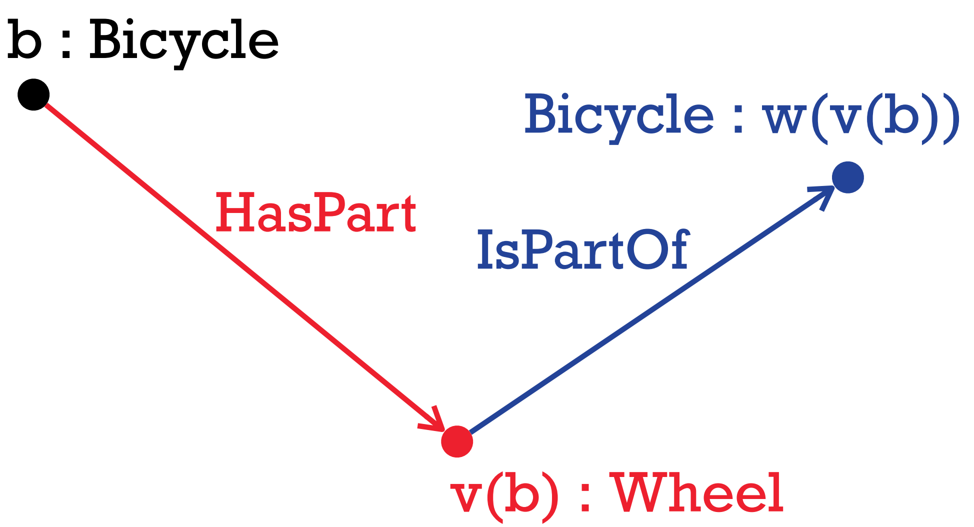
The Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



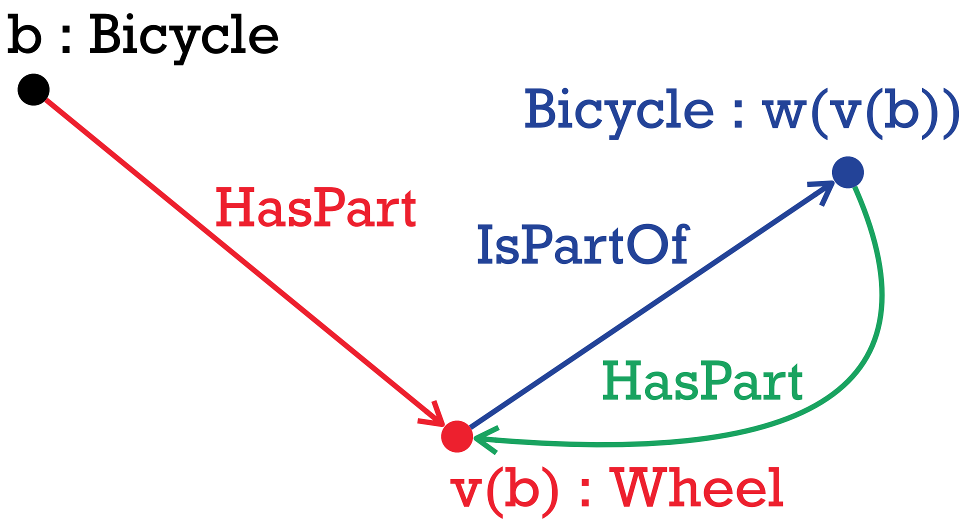
The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\ .\ \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



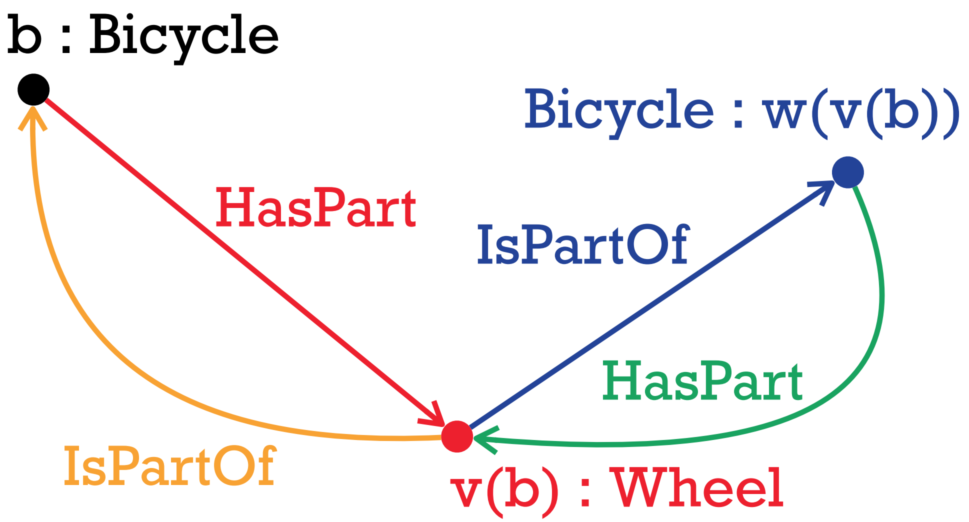
The Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v\, .\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w\, .\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



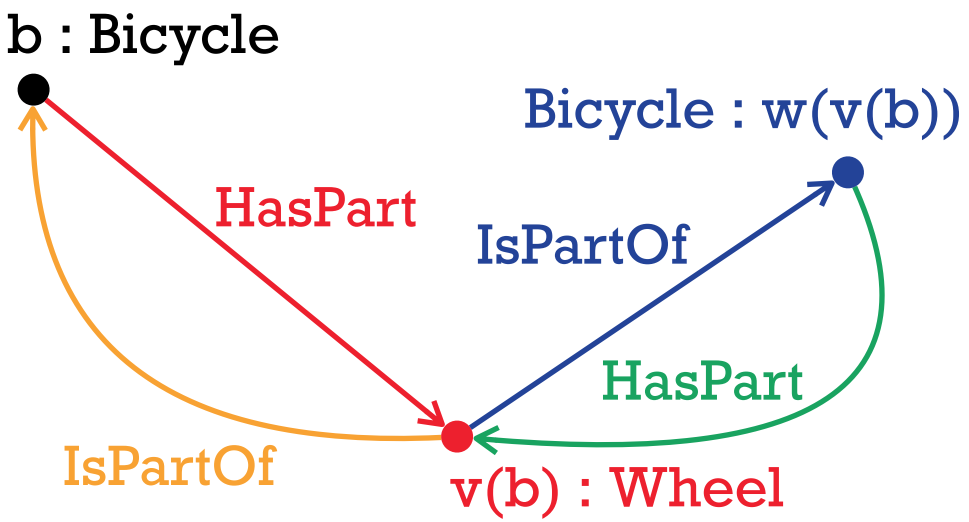
The Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v \, . \, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w \, . \, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



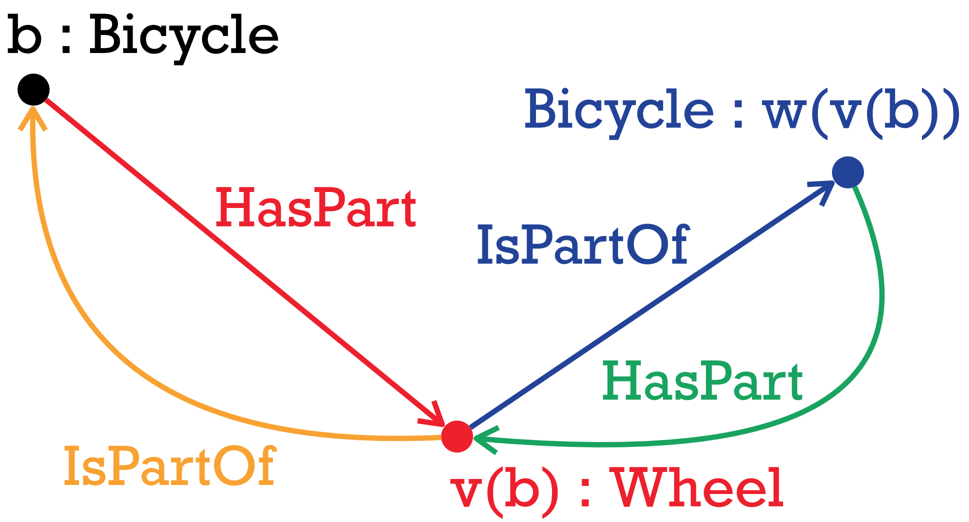
The Restricted Chase

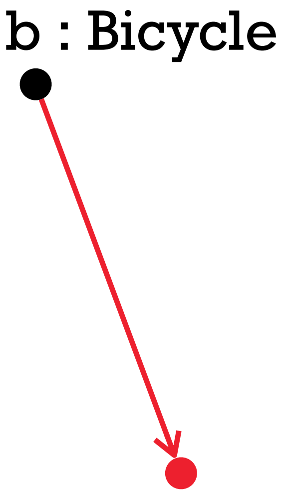
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





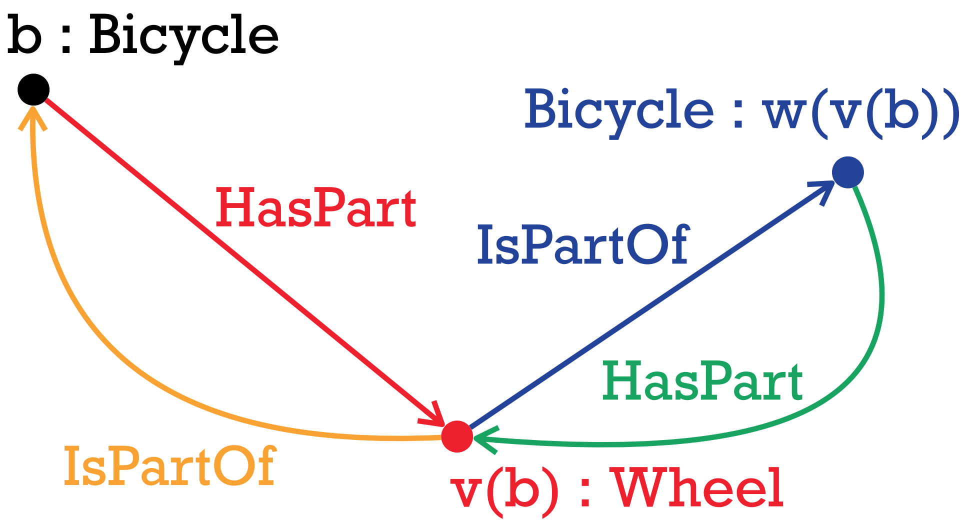
The Restricted Chase

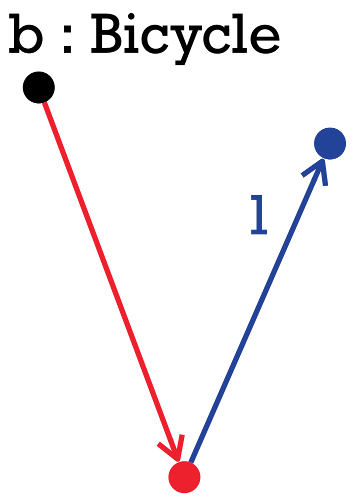
$\mathrm{Bicycle}(x) \longrightarrow \exists v \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

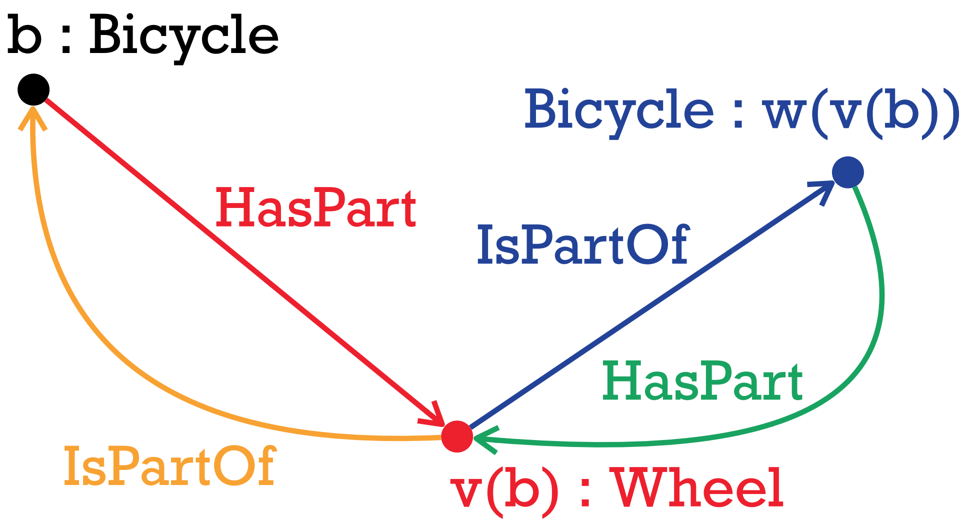


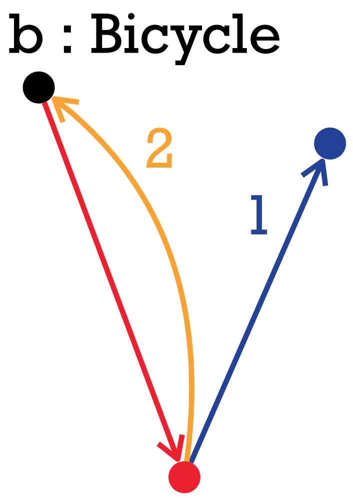


The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$ $\qquad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \exists w \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$ $\qquad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





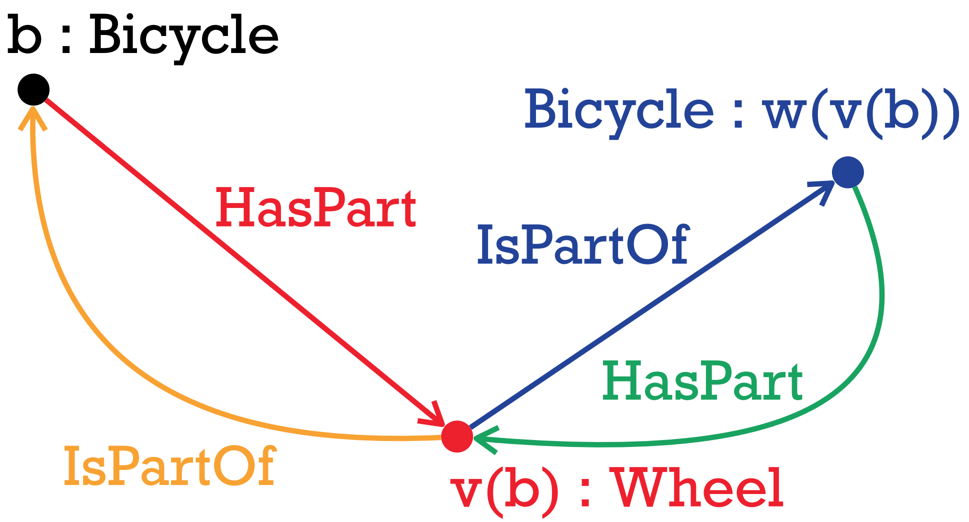
The Restricted Chase

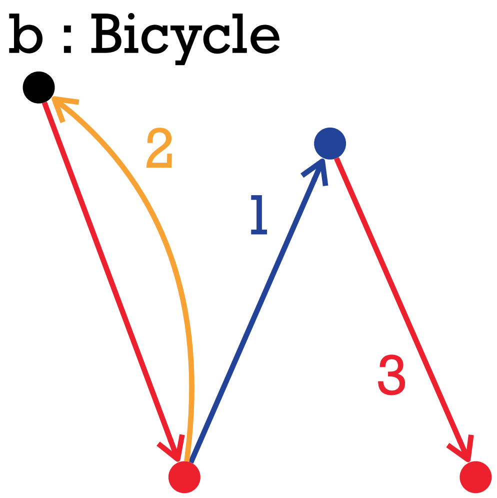
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





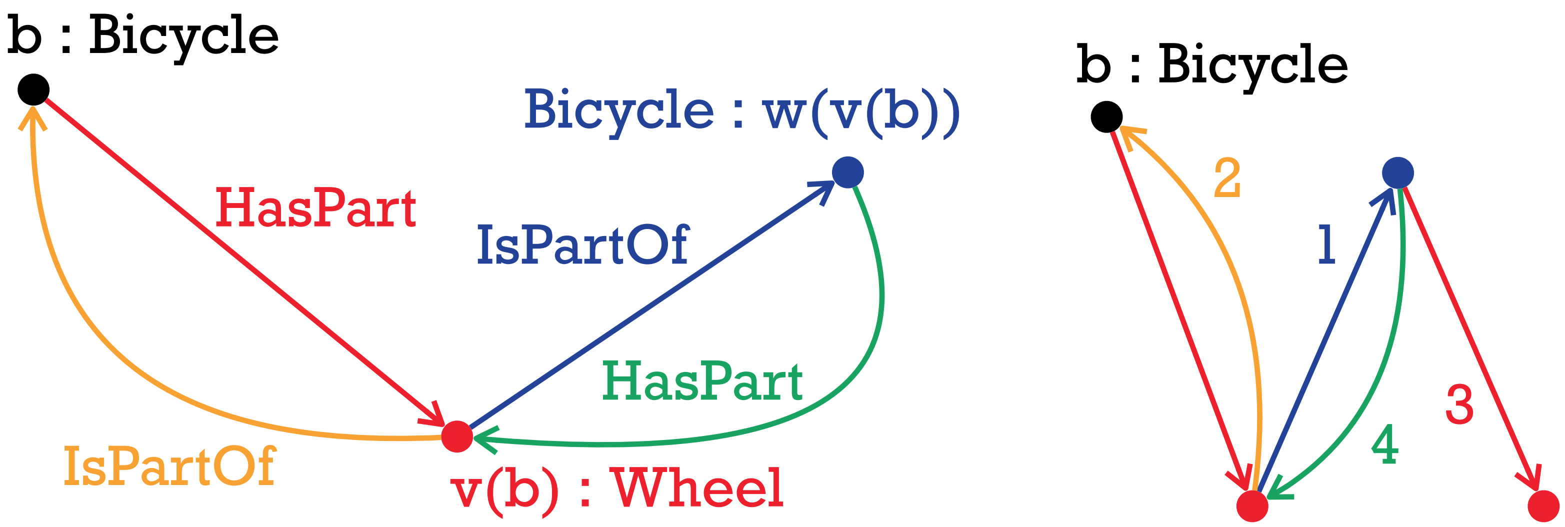
The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



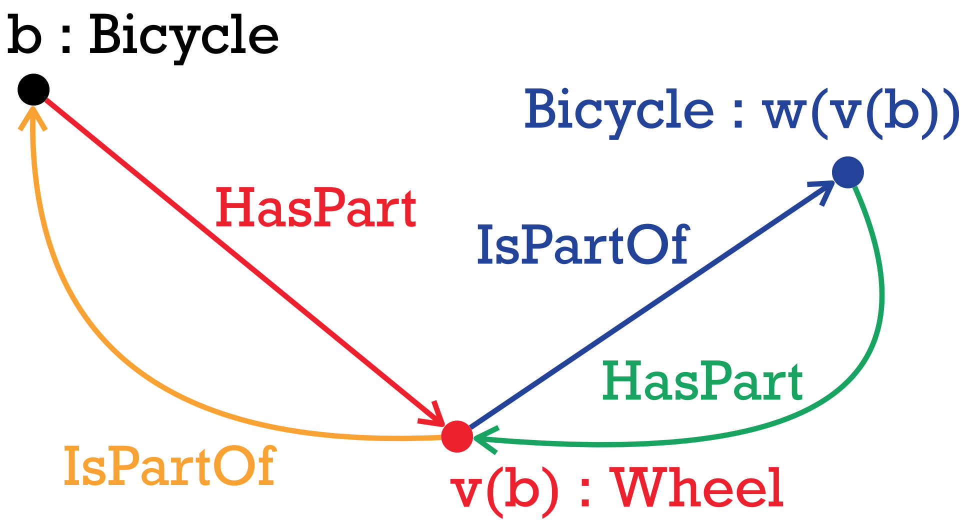
The Restricted Chase

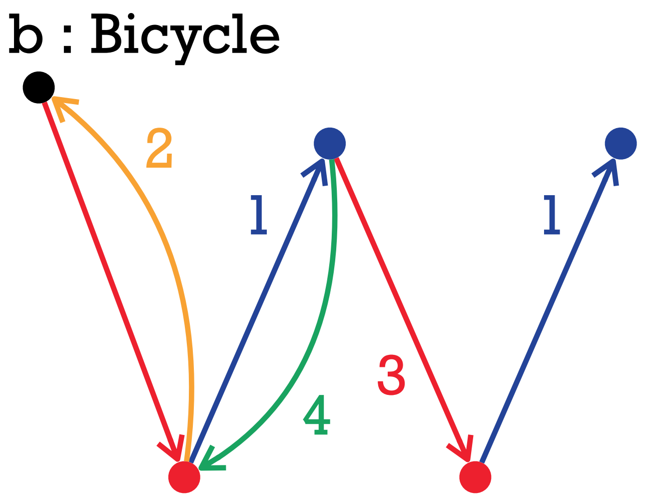
$\text{Bicycle}(x) \longrightarrow \exists v \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$

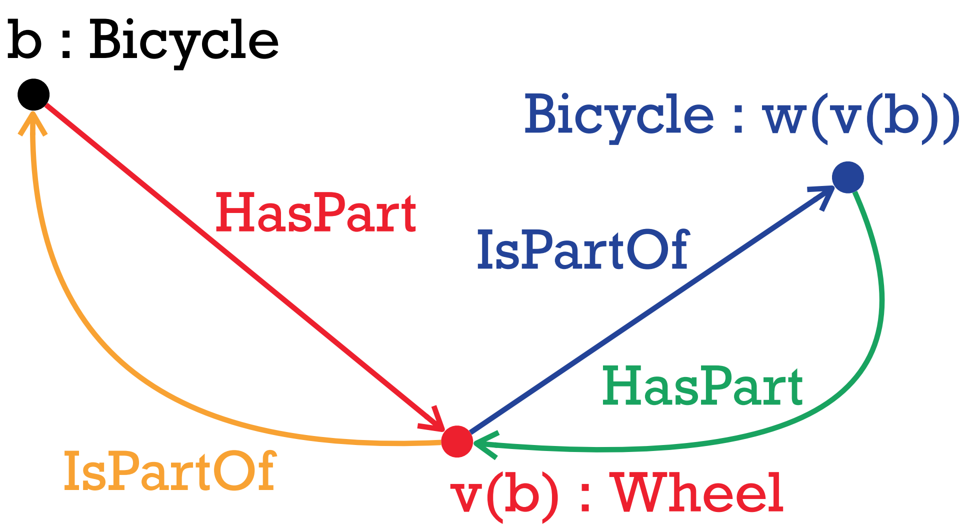


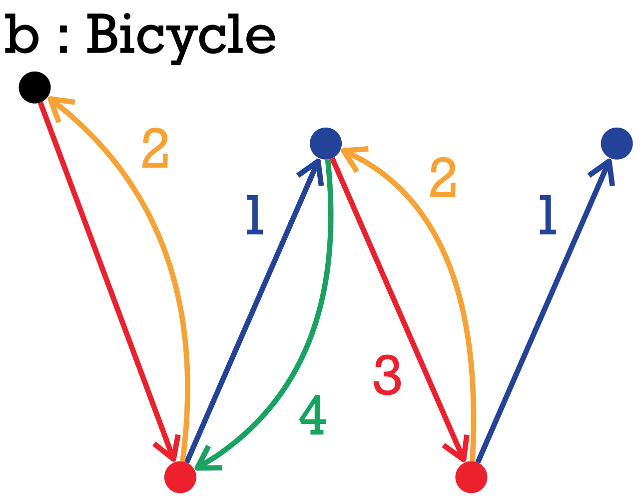


The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$ $\quad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$ $\quad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





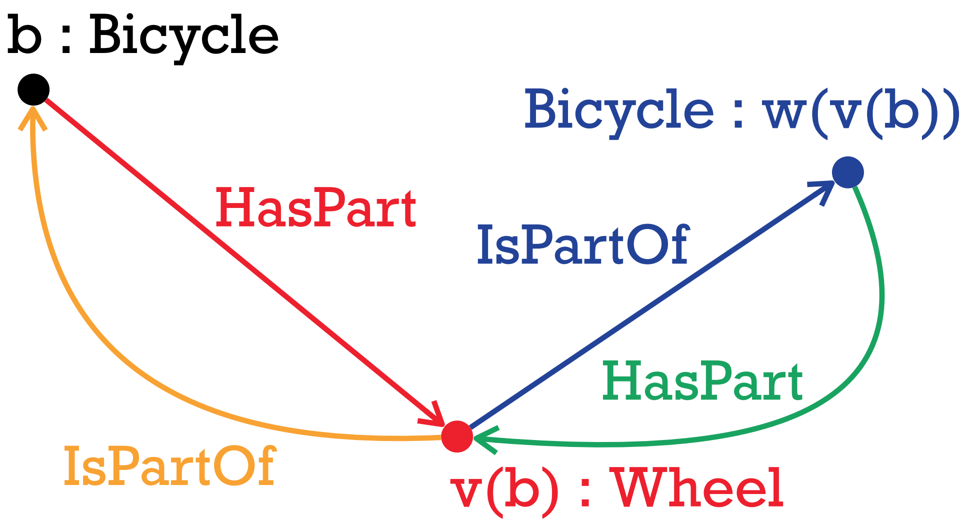
The Restricted Chase

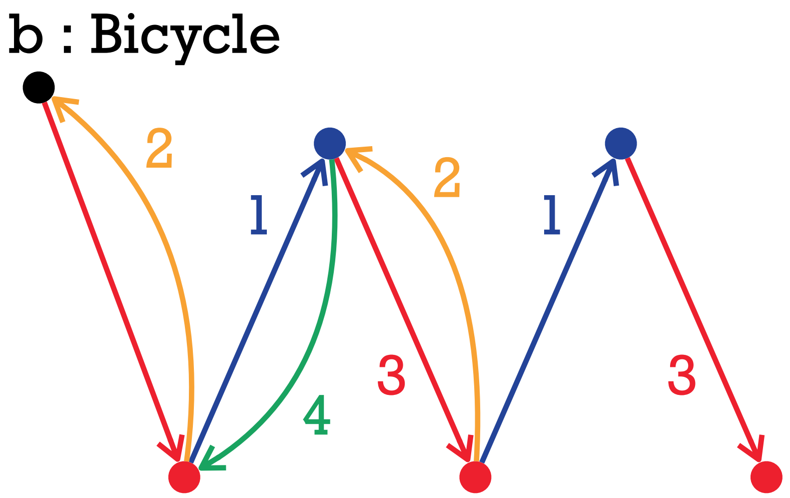
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





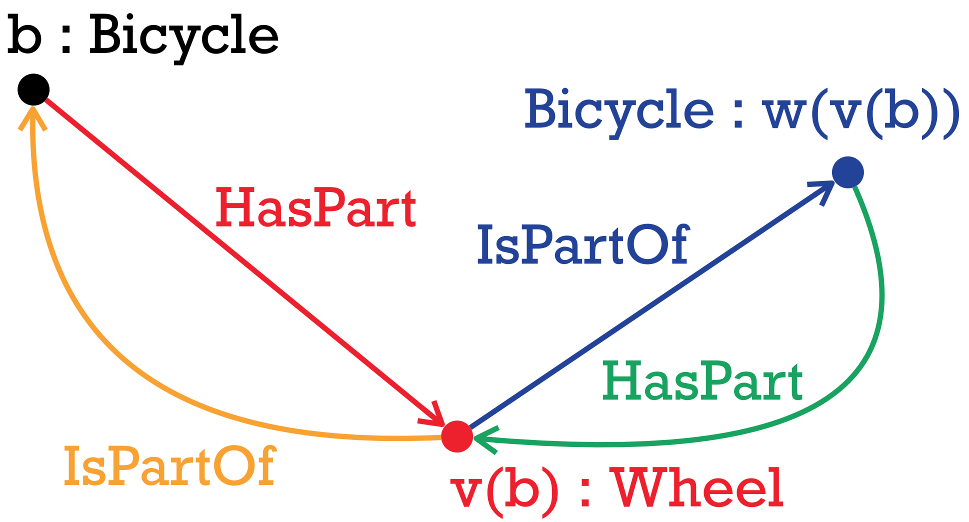
The Restricted Chase

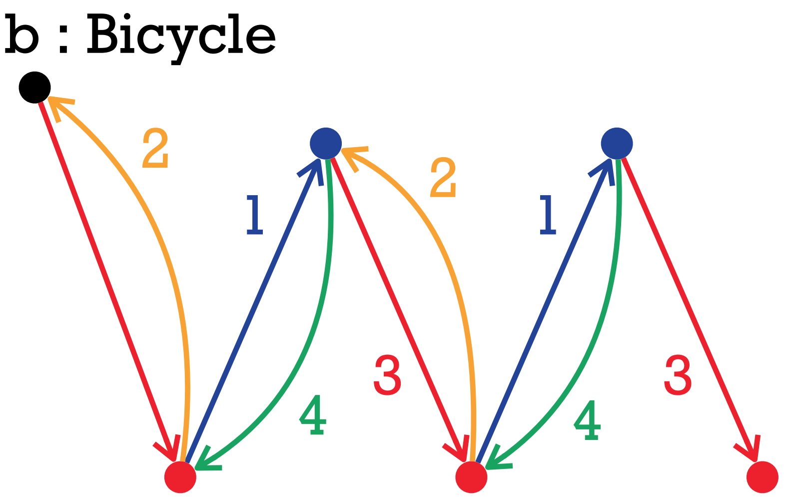
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

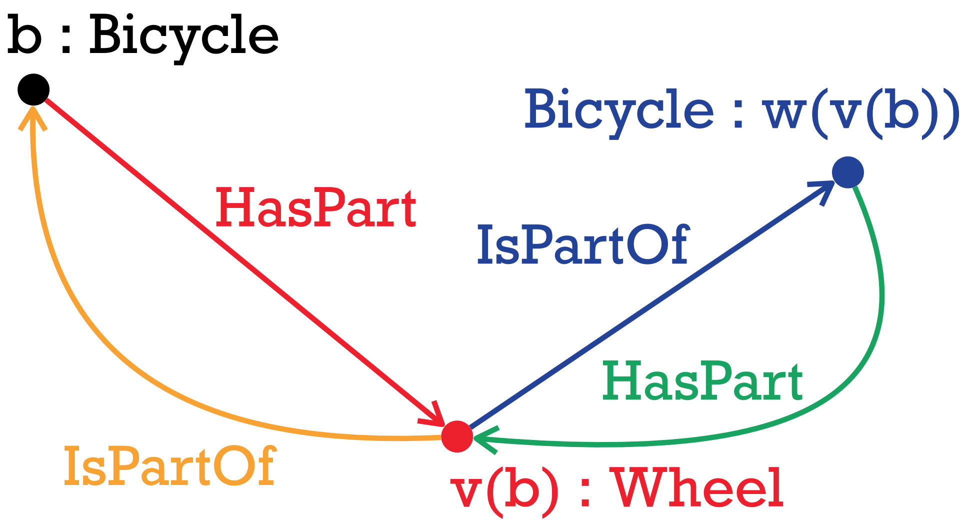


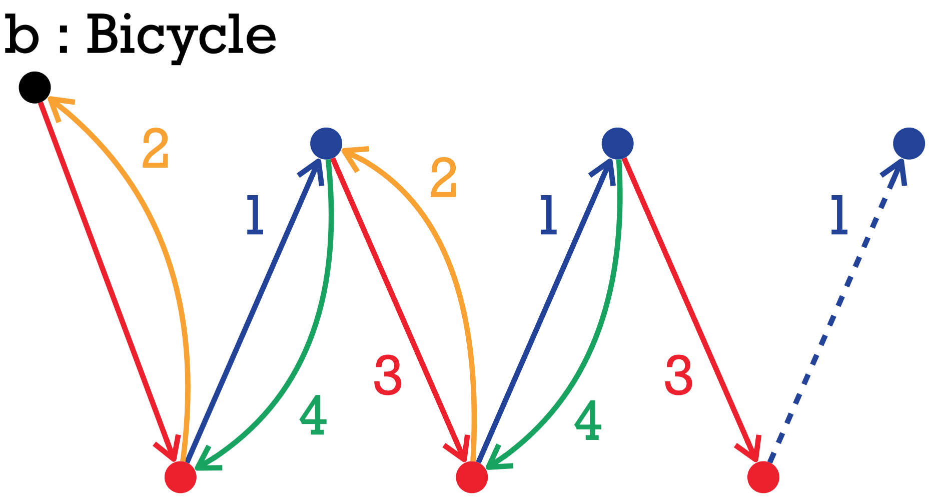


The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$ $\qquad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$ $\qquad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

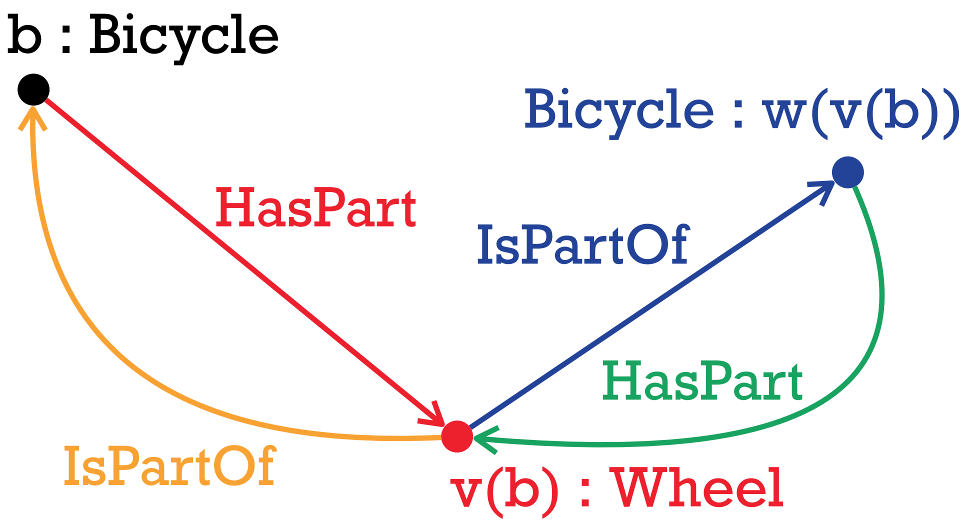


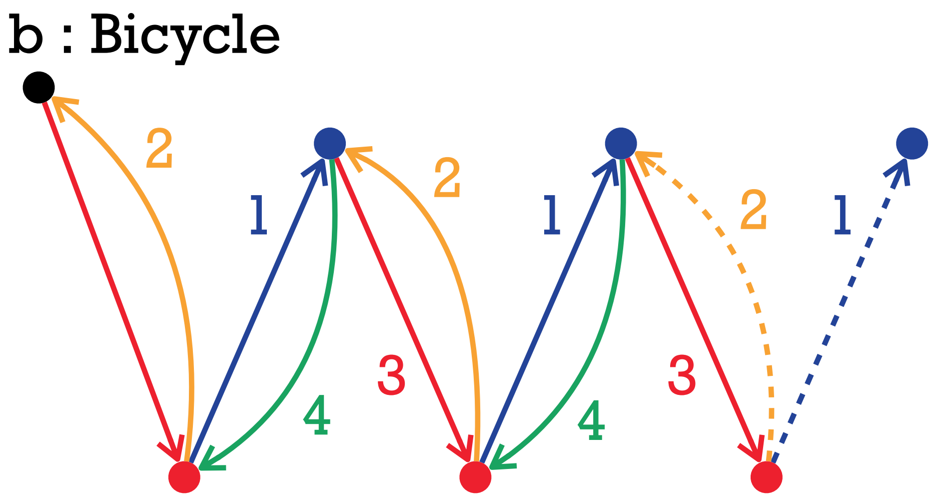


The Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$ $\qquad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$ $\qquad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





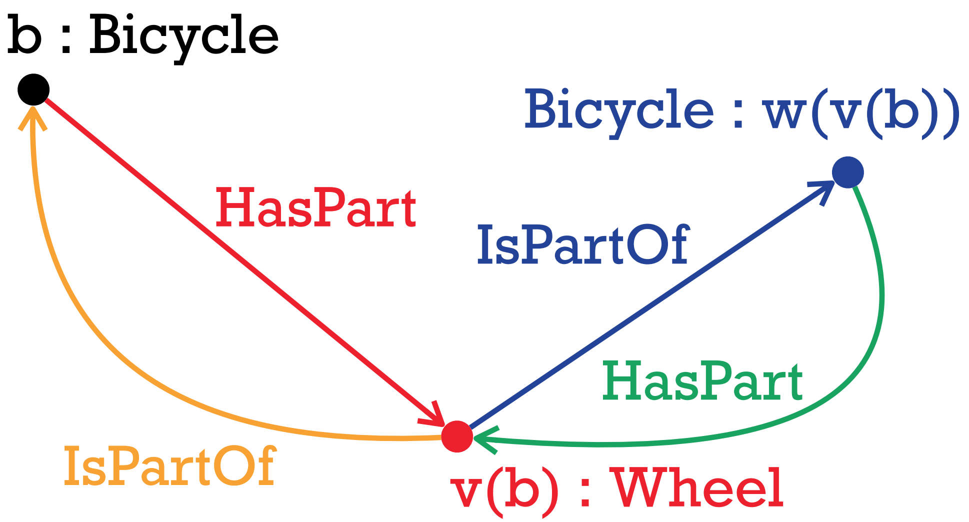
The Restricted Chase

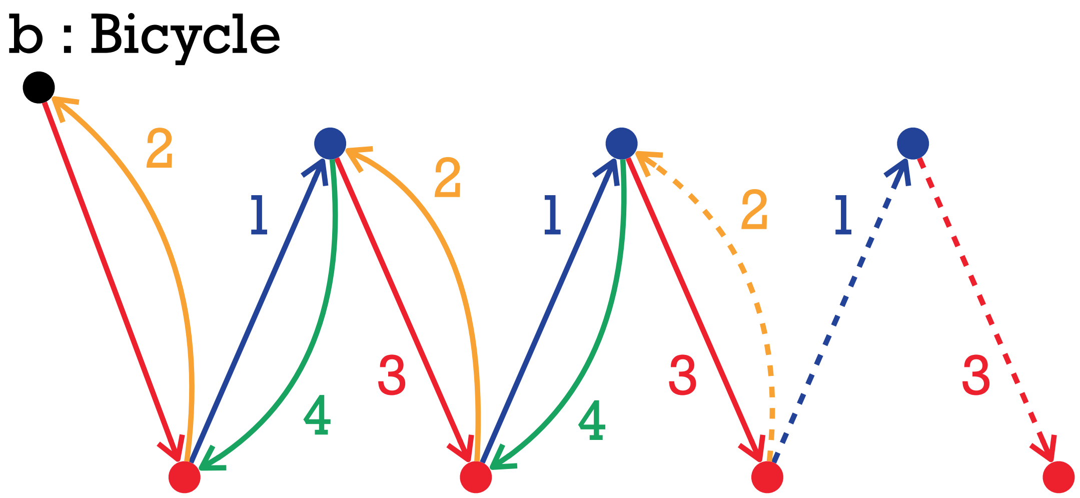
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





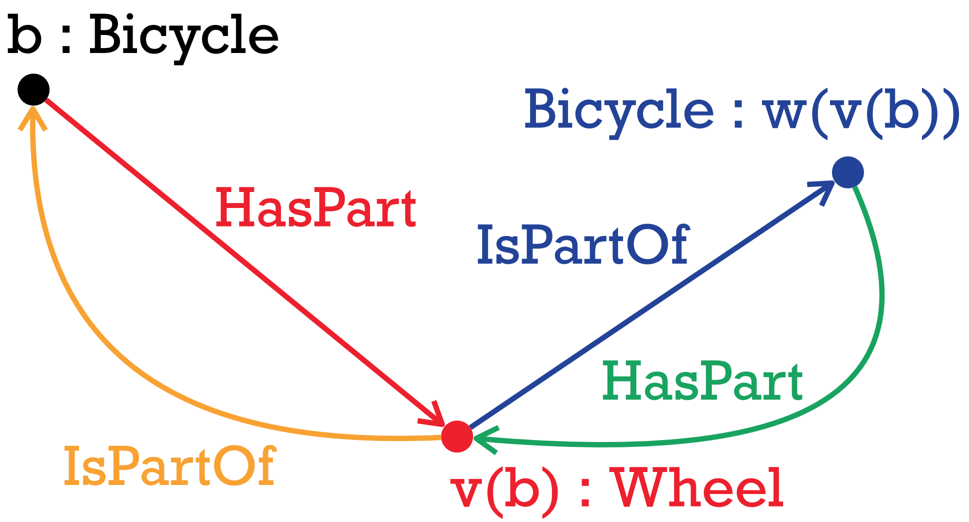
The Restricted Chase

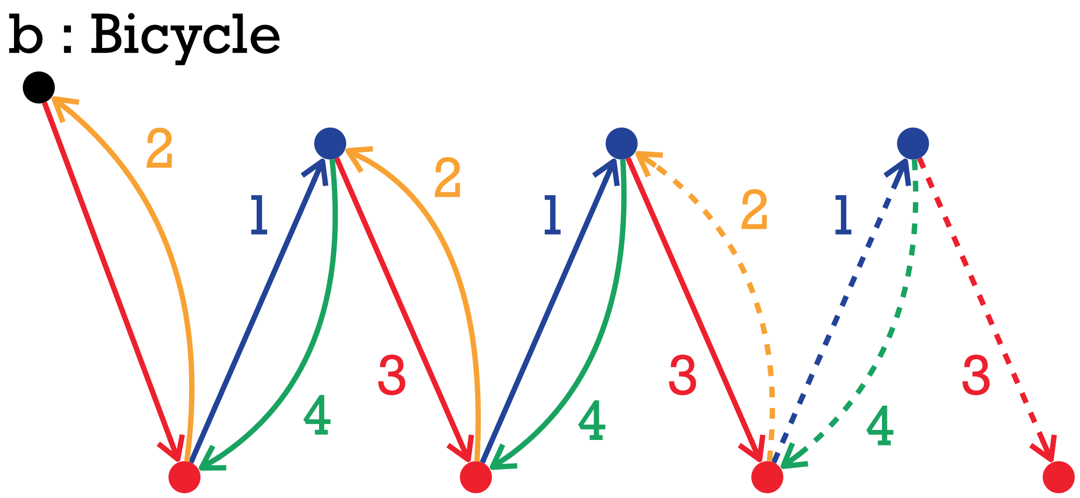
$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$





The Datalog-First Restricted Chase

$\mathrm{Bicycle}(x) \longrightarrow \exists v\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists w\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

b : Bicycle

The Datalog-First Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v\,.\,\text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w\,.\,\text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$

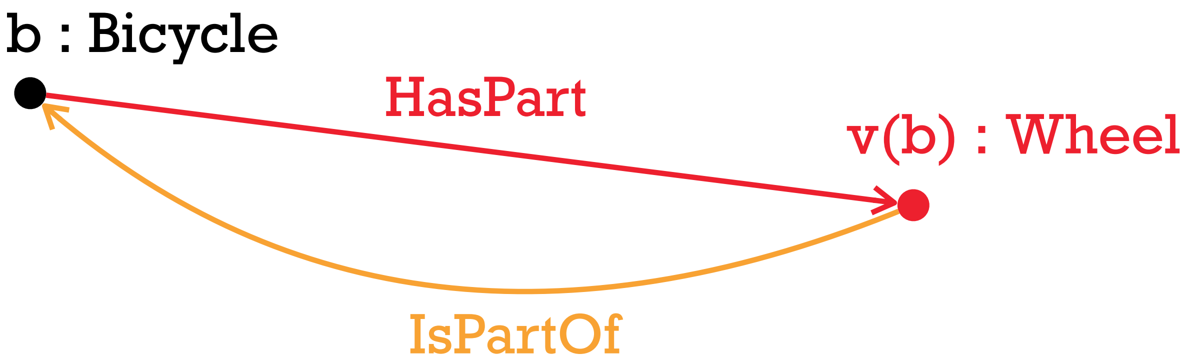
The Datalog-First Restricted Chase

$\text{Bicycle}(x) \longrightarrow \exists v \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists w \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



Acyclicity Notions

Acyclicity Notions

Restricted Chase (Non)Termination for Existential Rules with Disjunctions

David Carral, Irina Dragoste, and Markus Krötzsch

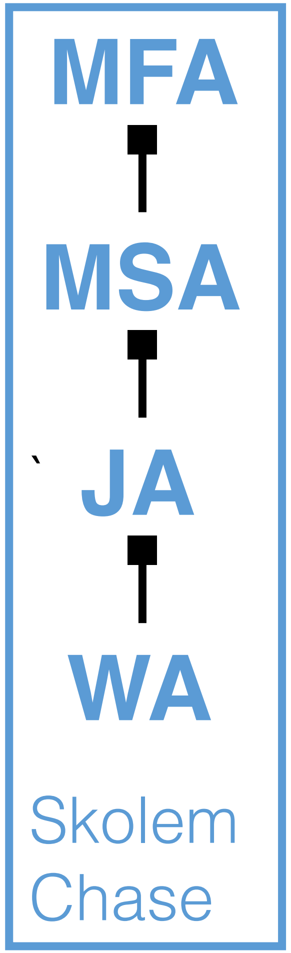
[IJCAI 2017]

Acyclicity Notions for Universal Termination

* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]

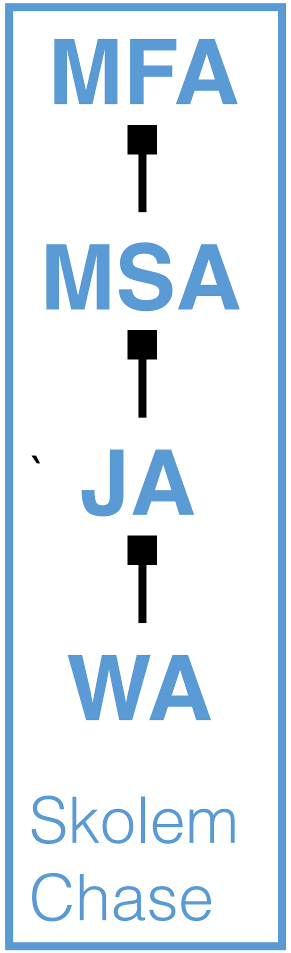
Acyclicity Notions for Universal Termination

* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]



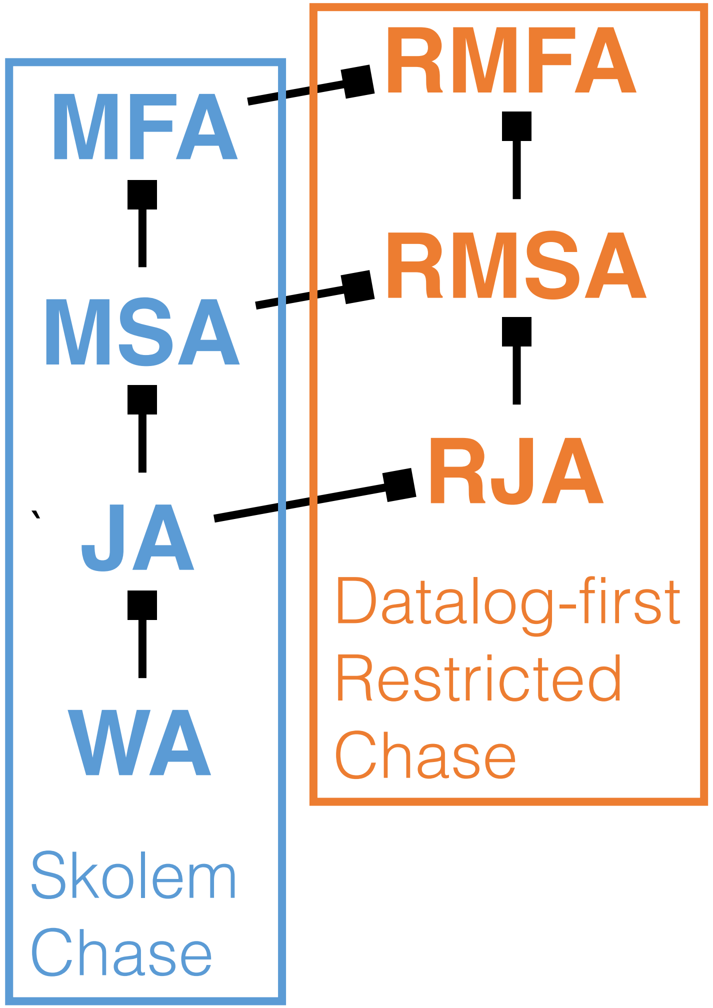
Acyclicity Notions for Universal Termination

* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]
* Restricted Joint Acyclicity (**RJA**),
  Restricted Model-Summarising Acyclicity (**RMSA**), and
  Restricted Model-Faithful Acyclicity (**RMFA**) [IJCAI 2017]



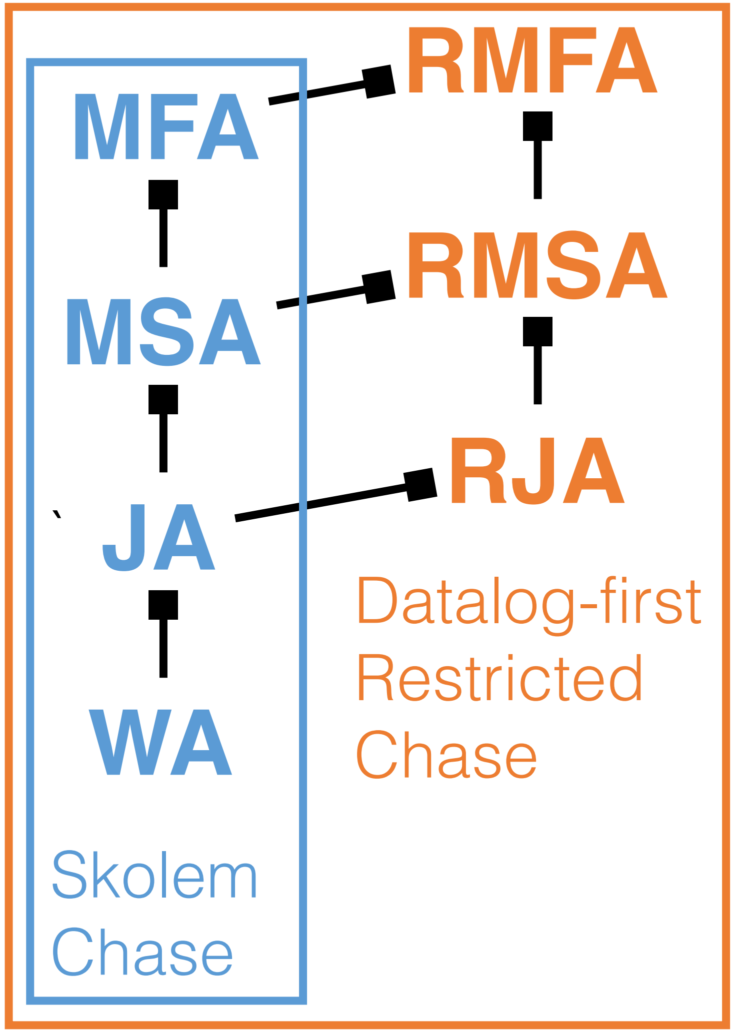
Acyclicity Notions for Universal Termination

* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]
* Restricted Joint Acyclicity (**RJA**),
  Restricted Model-Summarising Acyclicity (**RMSA**), and
  Restricted Model-Faithful Acyclicity (**RMFA**) [IJCAI 2017]



Acyclicity Notions for Universal Termination

* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]
* Restricted Joint Acyclicity (**RJA**),
  Restricted Model-Summarising Acyclicity (**RMSA**), and
  Restricted Model-Faithful Acyclicity (**RMFA**) [IJCAI 2017]



Acyclicity Notions for Universal Termination

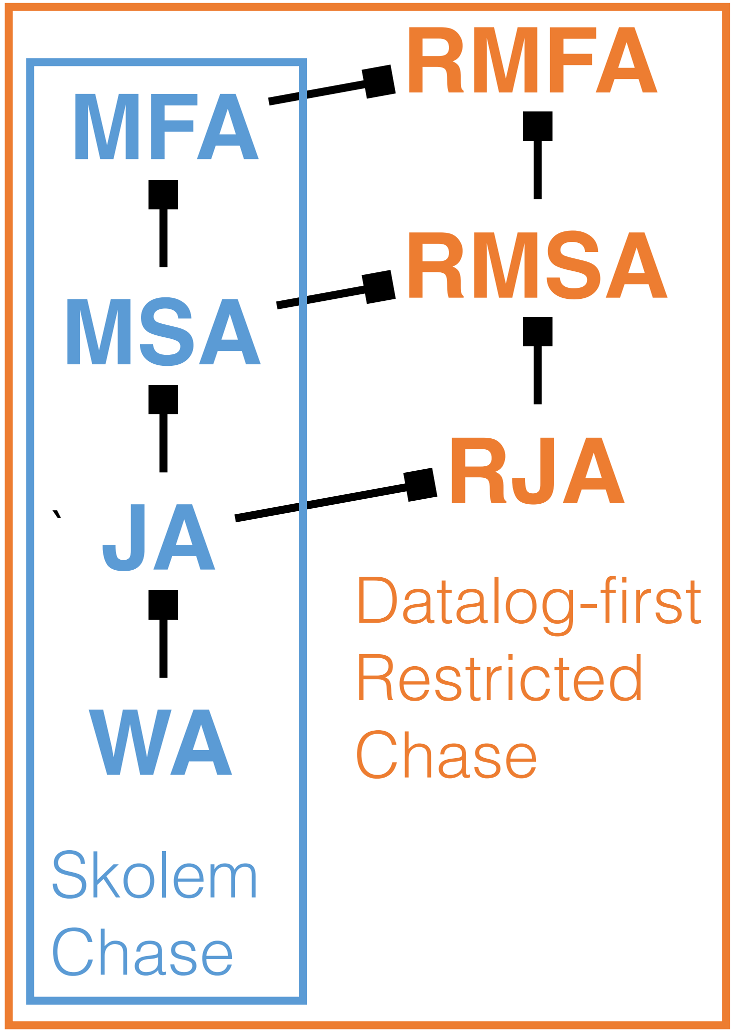
* Weak Acyclicity (**WA**) [Theor. Comput. Sci. 2005]
* Joint Acyclicity (**JA**) [IJCAI 2011]
* Model-Summarising Acyclicity (**MSA**) and
  Model-Faithful Acyclicity (**MFA**) [J. Artif. Intell. Res. 2013]
* Restricted Joint Acyclicity (**RJA**),
  Restricted Model-Summarising Acyclicity (**RMSA**), and
  Restricted Model-Faithful Acyclicity (**RMFA**) [IJCAI 2017]

$$\mathrm{Bicycle}(x) \longrightarrow \exists v \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists w \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

$$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.

The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant “★”), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

The MFA Check

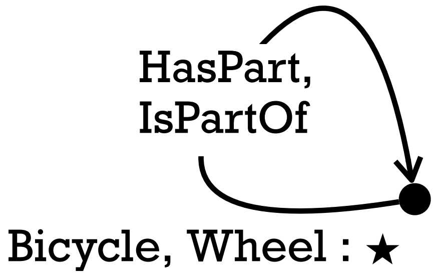
* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

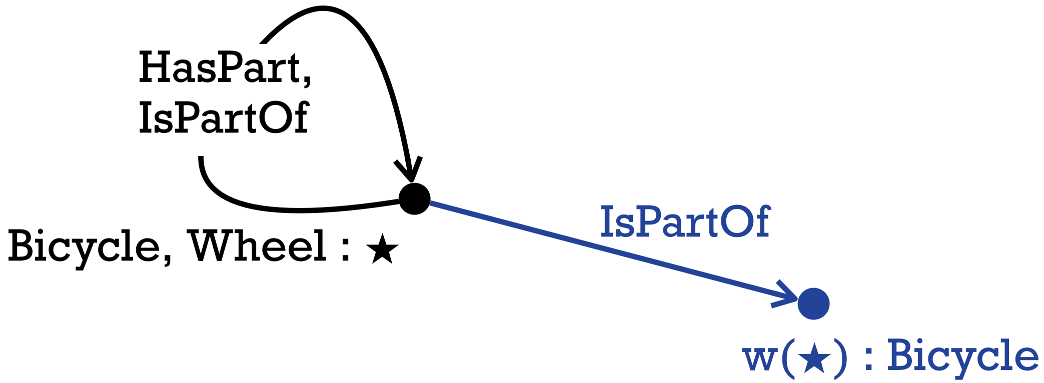


The MFA Check

- **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant “★”), then it terminates on all sets of facts.
- **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

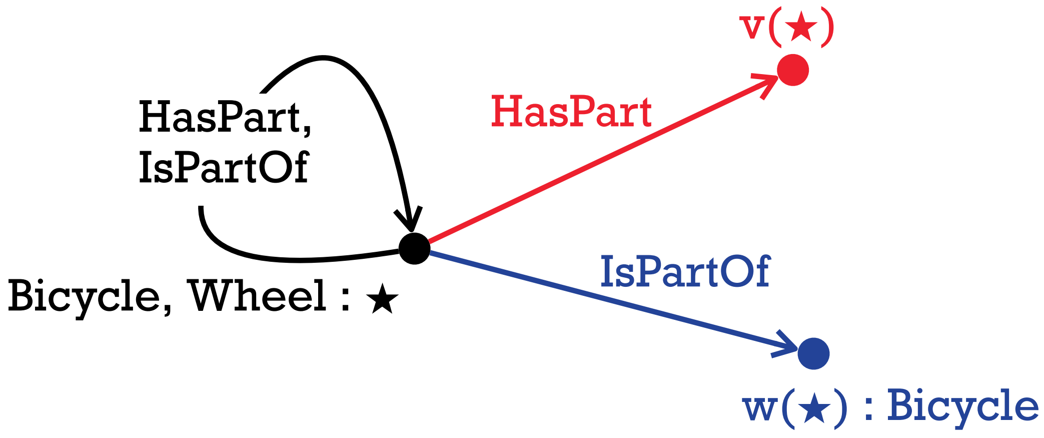
* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

$$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

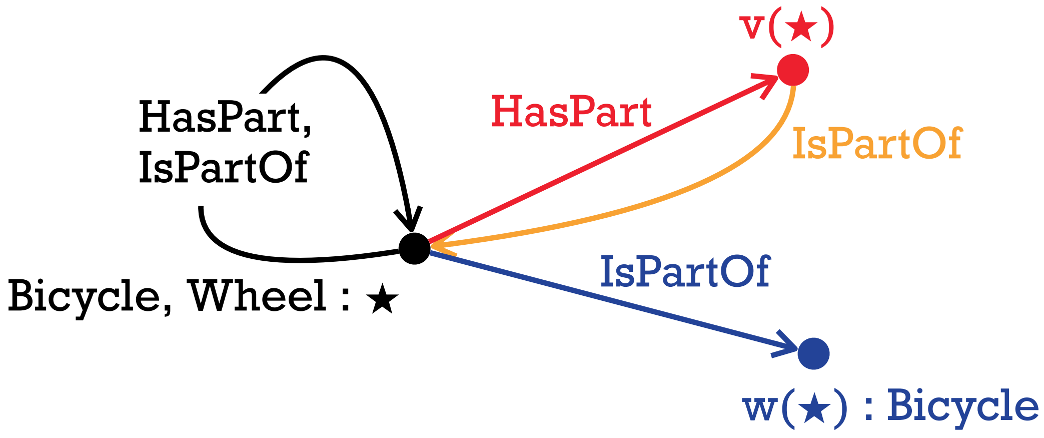
* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

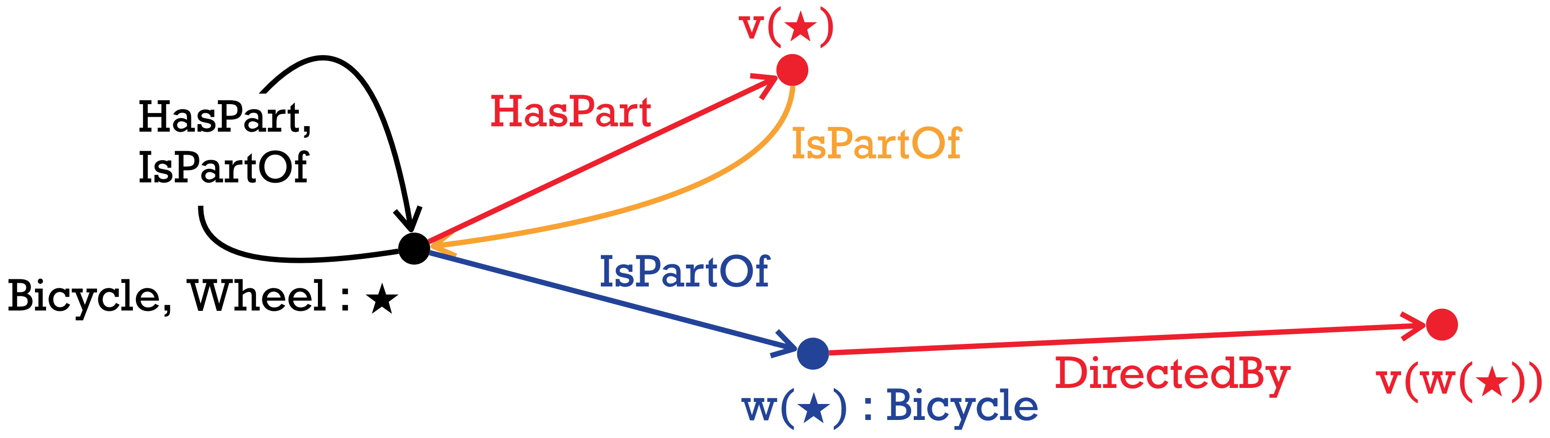


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

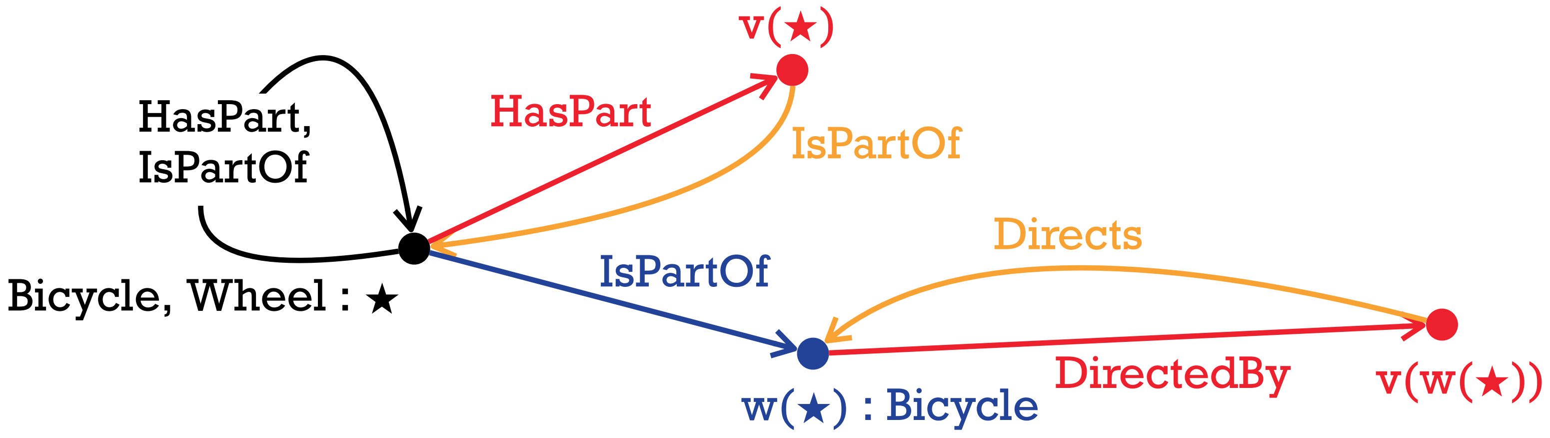


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v)$ $\qquad$ $\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$ $\qquad$ $\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

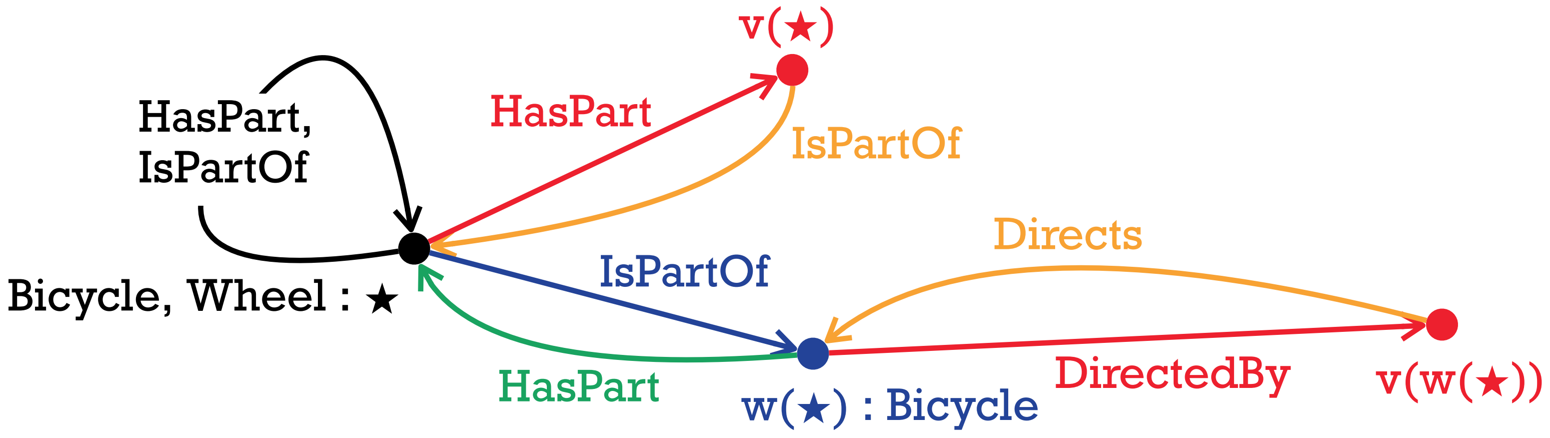


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

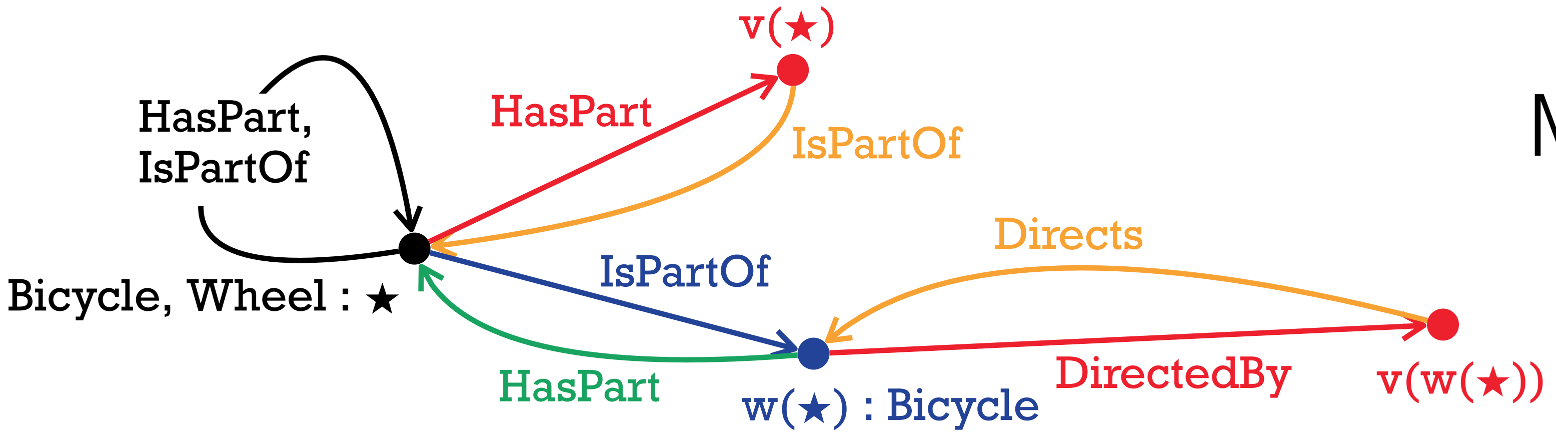
* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



MFA

The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

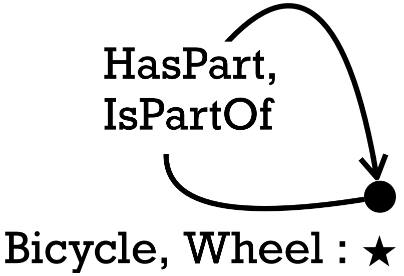
$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

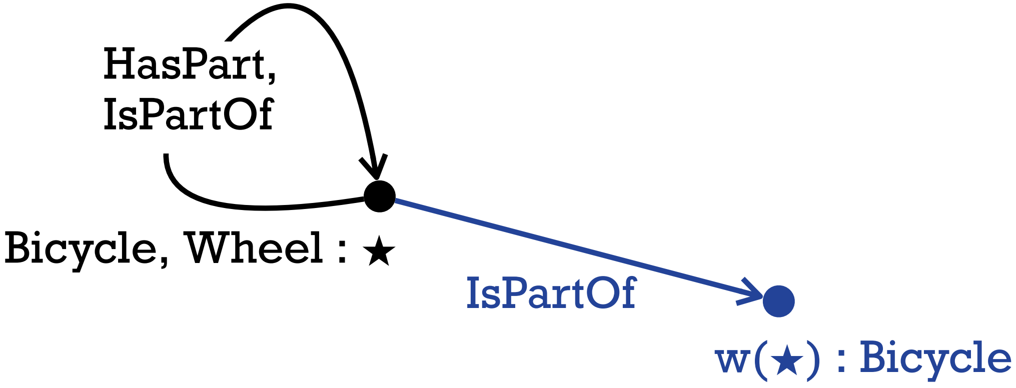


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant “★”), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

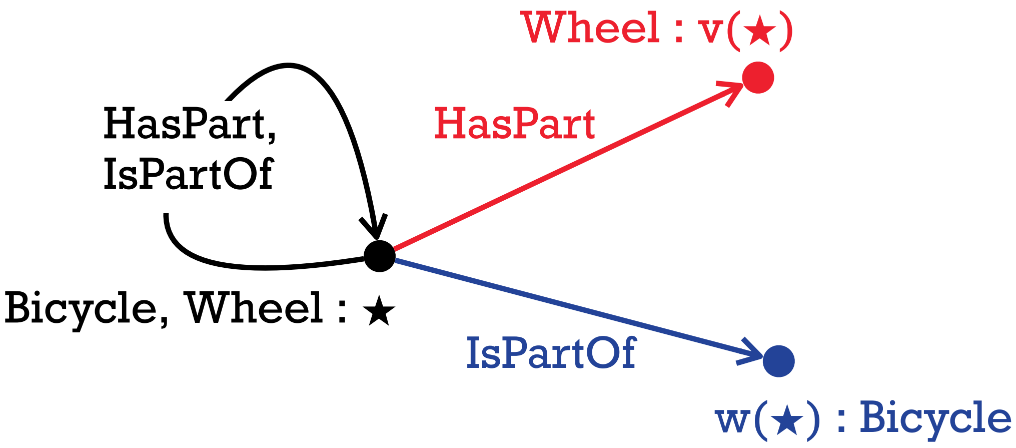
- **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
- **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

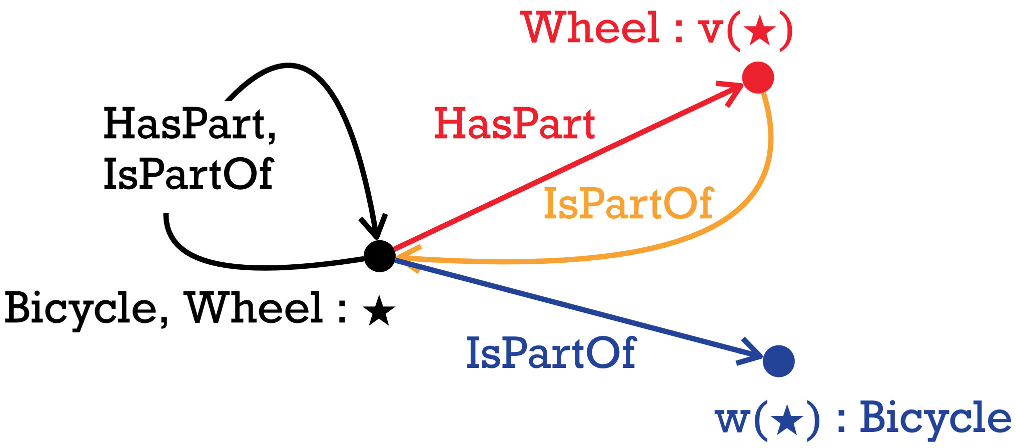


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

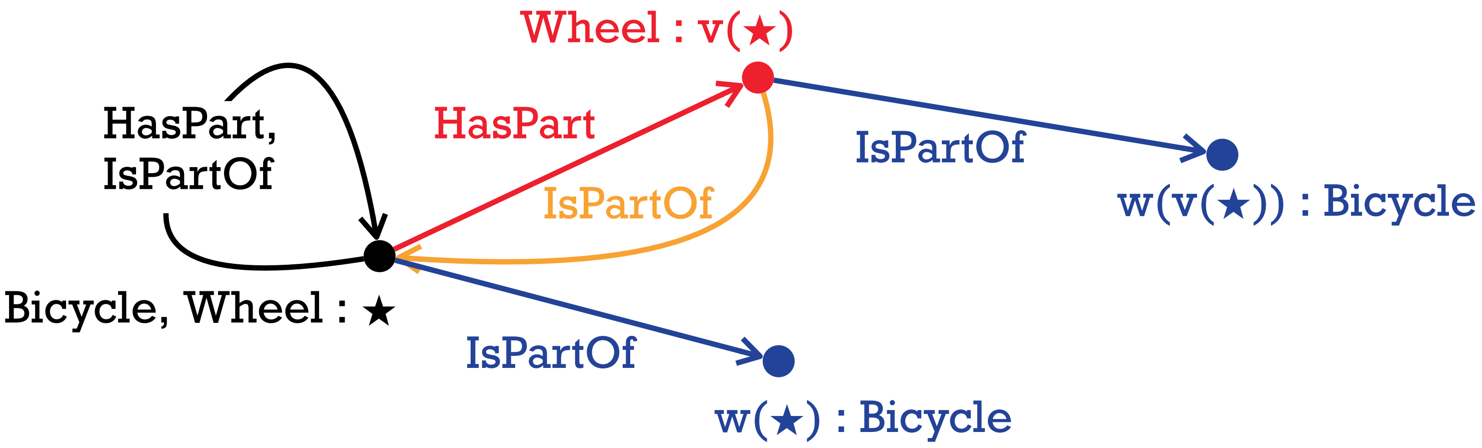


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

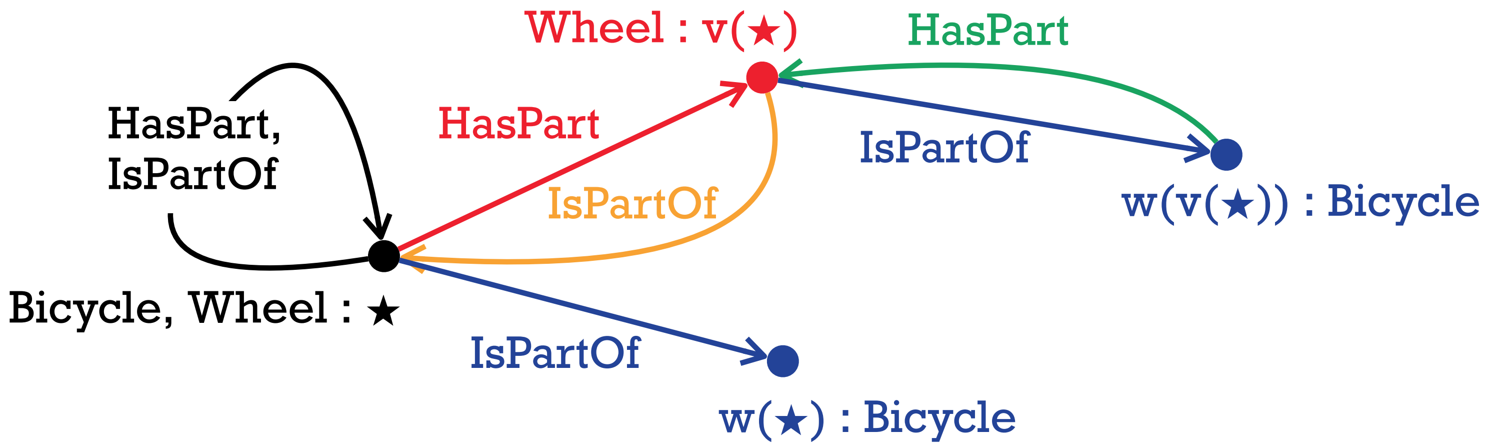


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The MFA Check

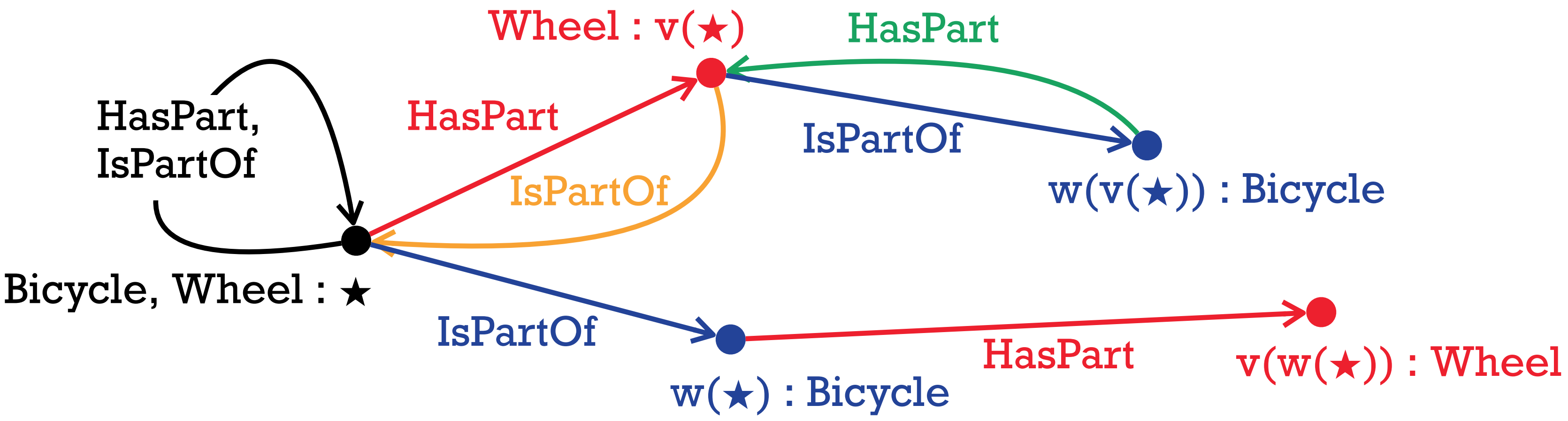
* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

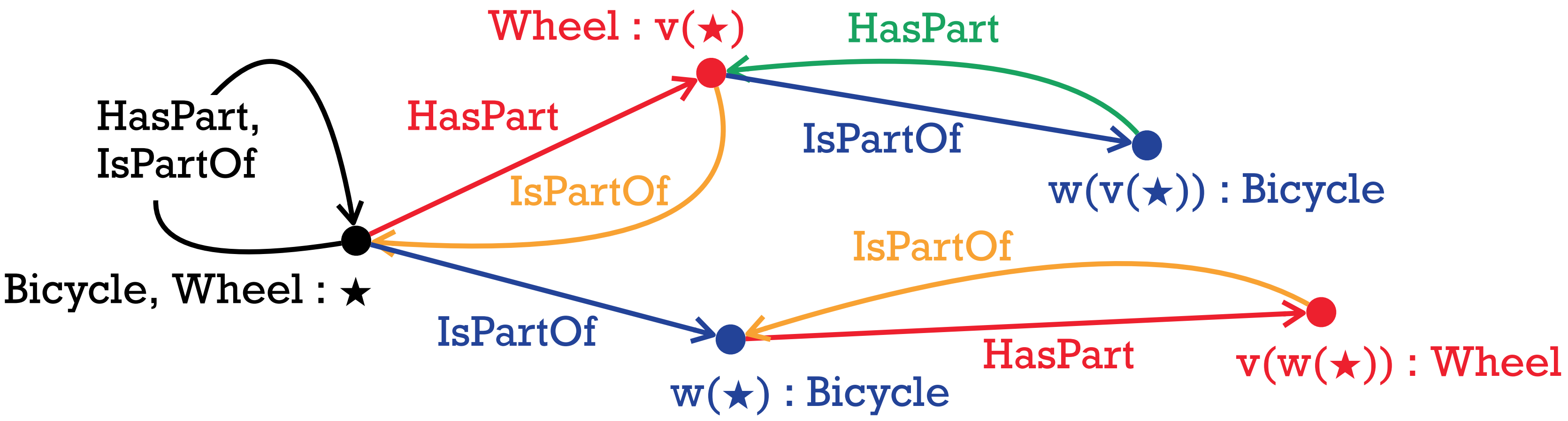


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

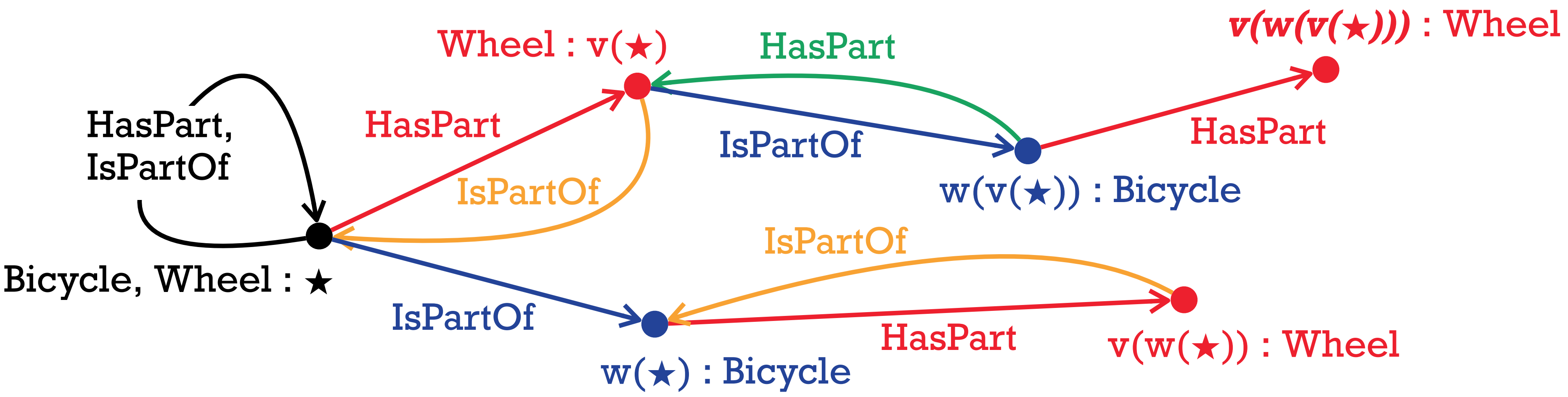


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant “★”), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

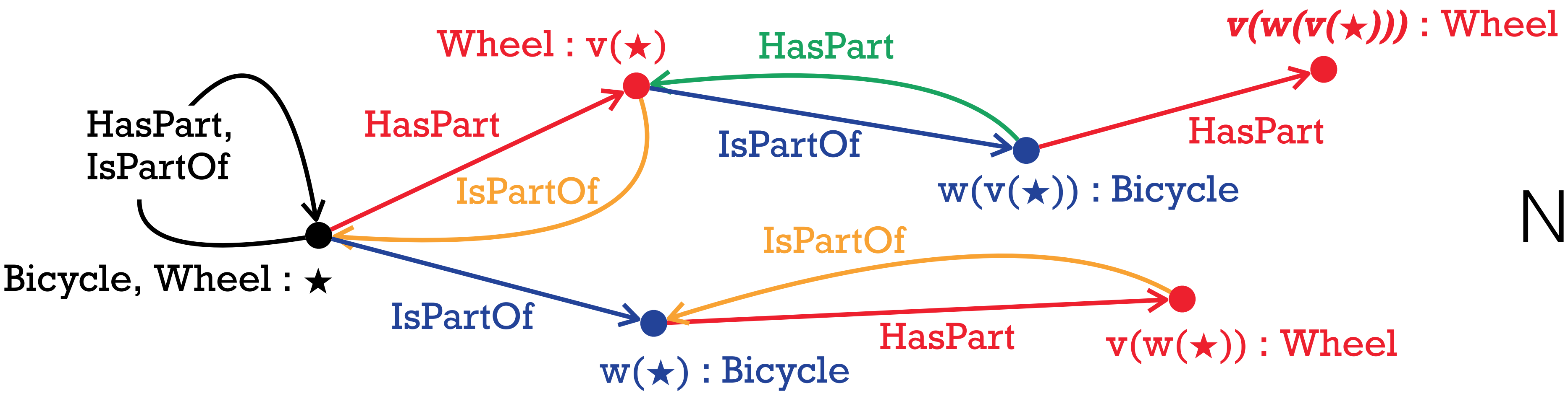


The MFA Check

* **Fact:** If the Skolem chase terminates on the critical instance (the set of all possible facts containing a single constant "★"), then it terminates on all sets of facts.
* **MFA Check:** Perform chase with the critical instance, check if it stops; give up if a **cyclic skolem term** (with a repeated function symbol) appears.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



Not MFA!

The RMFA Check: Blocked Checks

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

$$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

Example: Suppose for a contradiction that the fact $\mathrm{Wheel}(v(w(t)))$ with **t** some term is derived during the computation of a chase sequence.

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

Example: Suppose for a contradiction that the fact $\mathrm{Wheel}(v(w(t)))$ with **t** some term is derived during the computation of a chase sequence.

$v(w(t)) : \mathit{Wheel}$

The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

$v(w(t)) : Wheel$

Example: Suppose for a contradiction that the fact $\mathrm{Wheel}(v(w(t)))$ with $\mathbf{t}$ some term is derived during the computation of a chase sequence.

* Such a fact may only be derived via application of the **red rule** on $\mathrm{Bicycle}(w(t))$ which in turn may only be derived if the **blue rule** is applied. Hence, $\mathbf{Wheel(t)}$ and $\mathrm{IsPartOf}(t, w(t))$ and are also part of the chase before $\mathrm{Wheel}(v(w(t)))$ is derived.

The RMFA Check: Blocked Checks

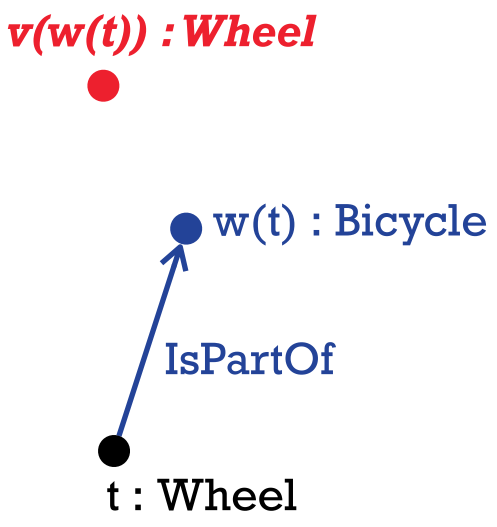
* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

Example: Suppose for a contradiction that the fact $\mathrm{Wheel}(v(w(t)))$ with **t** some term is derived during the computation of a chase sequence.

* Such a fact may only be derived via application of the **red rule** on $\mathrm{Bicycle}(w(t))$ which in turn may only be derived if the **blue rule** is applied. Hence, $\mathrm{Wheel}(t)$ and $\mathrm{IsPartOf}(t, w(t))$ and are also part of the chase before $\mathrm{Wheel}(v(w(t)))$ is derived.



The RMFA Check: Blocked Checks

* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$\text{Bicycle}(x) \longrightarrow \exists y\ .\ \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

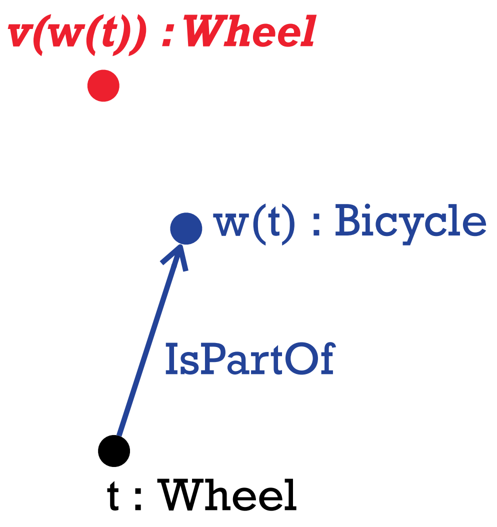
$\text{Wheel}(x) \longrightarrow \exists y\ .\ \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$

Example: Suppose for a contradiction that the fact Wheel(v(w(t))) with **t** some term is derived during the computation of a chase sequence.

* Such a fact may only be derived via application of the red rule on Bicycle(w(t)) which in turn may only be derived if the blue rule is applied. Hence, Wheel(t) and IsPartOf(t, w(t)) and are also part of the chase before Wheel(v(w(t))) is derived.
* Because the green rule is Datalog, DirectedBy(v(t), t) is also part of the chase.



The RMFA Check: Blocked Checks

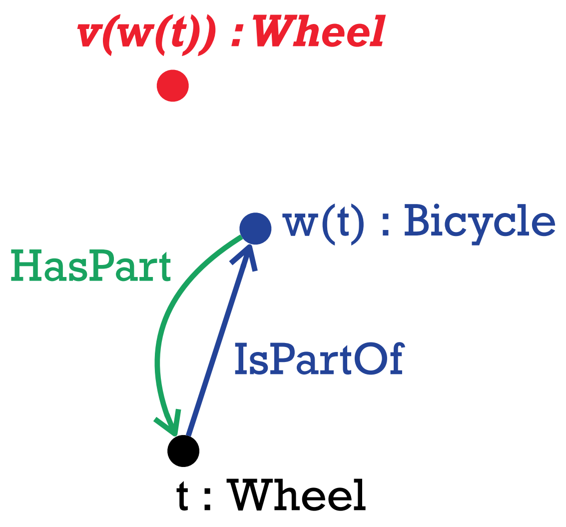
* Problem: Datalog-first restricted chase termination is not monotone!
* In particular, it always terminates on the critical instance.
* Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

Example: Suppose for a contradiction that the fact Wheel(v(w(t))) with **t** some term is derived during the computation of a chase sequence.

* Such a fact may only be derived via application of the red rule on Bicycle(w(t)) which in turn may only be derived if the blue rule is applied. Hence, Wheel(t) and IsPartOf(t, w(t)) and are also part of the chase before Wheel(v(w(t))) is derived.
* Because the green rule is Datalog, DirectedBy(v(t), t) is also part of the chase.



The RMFA Check: Blocked Checks

- Problem: Datalog-first restricted chase termination is not monotone!
- In particular, it always terminates on the critical instance.
- Idea: for each fact that occurs in the chase sequence, we can re-trace a necessary fact set the must have been derived to derive this fact. By checking these facts we can in some cases determine that the application of the rule and substitution that generates this fact is **blocked**.

$\mathrm{Bicycle}(x) \longrightarrow \exists y\ .\ \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

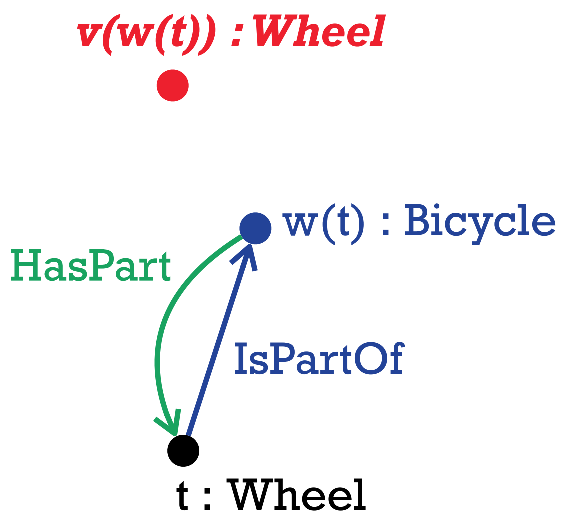
$\mathrm{Wheel}(x) \longrightarrow \exists y\ .\ \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

Example: Suppose for a contradiction that the fact $\mathrm{Wheel}(v(w(t)))$ with **t** some term is derived during the computation of a chase sequence.

- Such a fact may only be derived via application of the **red rule** on $\mathrm{Bicycle}(w(t))$ which in turn may only be derived if the **blue rule** is applied. Hence, $\mathrm{Wheel}(t)$ and $\mathrm{IsPartOf}(t, w(t))$ and are also part of the chase before $\mathrm{Wheel}(v(w(t)))$ is derived.
- Because the **green rule** is Datalog, $\mathrm{DirectedBy}(v(t), t)$ is also part of the chase.
- The **red rule** may not be applied to introduce $\mathrm{Director}(v(w(t)))$ since its application with respect to the substitution $\{x\ /\ w(t)\}$ is restricted.



The RMFA Check

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$$

$$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

The RMFA Check

* Perform a chase like construction on the critical instance.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

The RMFA Check

* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

The RMFA Check

* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\text{Bicycle}(x) \longrightarrow \exists y \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists y \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$

The RMFA Check

* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

The RMFA Check

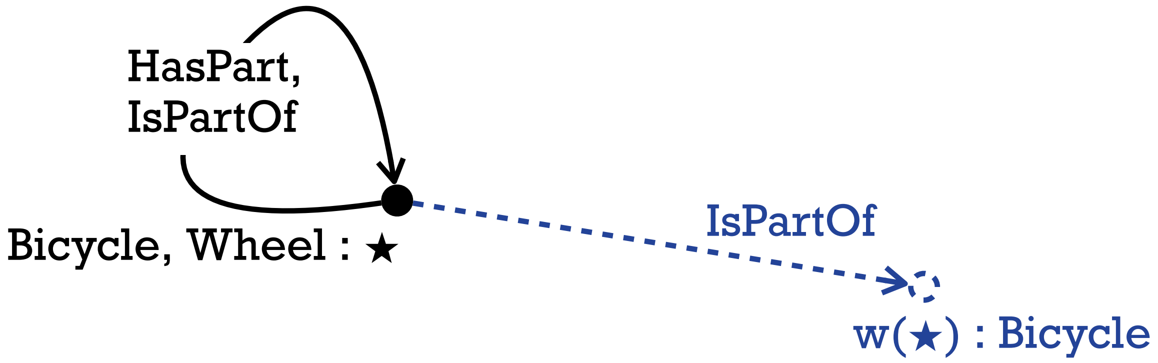
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The RMFA Check

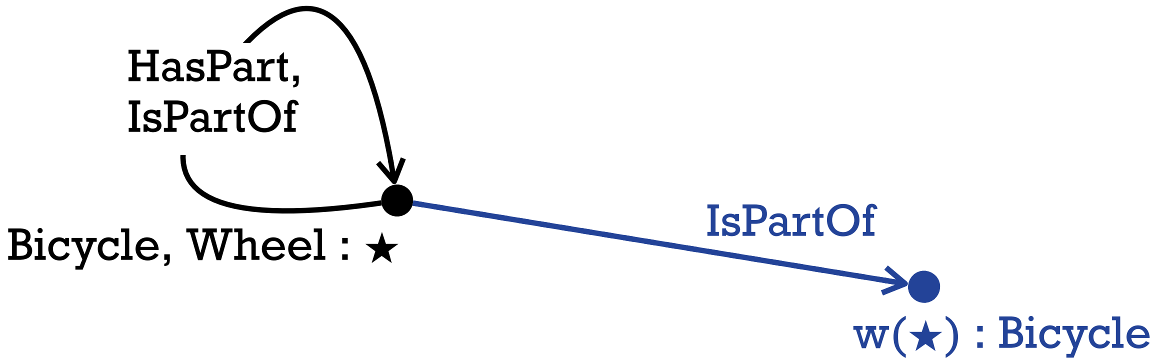
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \land \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \land \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The RMFA Check

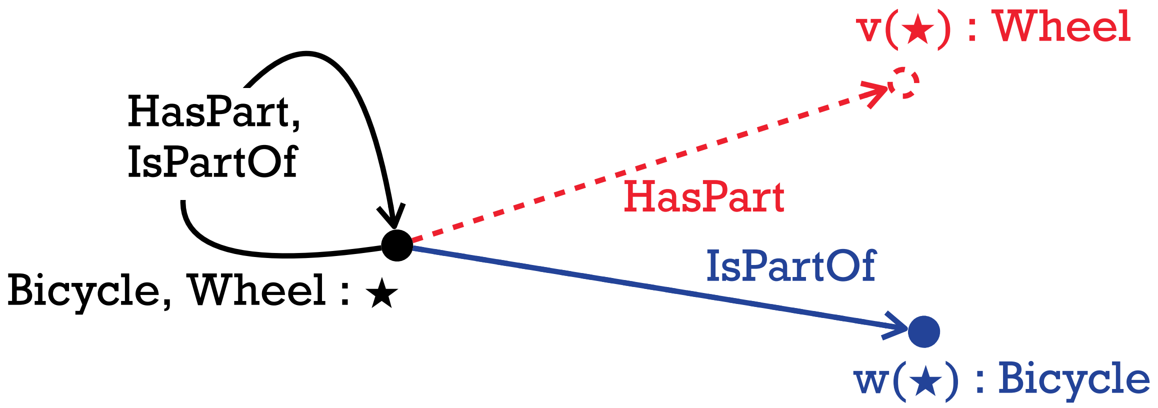
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\text{Bicycle}(x) \longrightarrow \exists y \,.\, \text{HasPart}(x, v) \wedge \text{Wheel}(v)$

$\text{Wheel}(x) \longrightarrow \exists y \,.\, \text{IsPartOf}(x, w) \wedge \text{Bicycle}(w)$

$\text{HasPart}(x, y) \longrightarrow \text{IsPartOf}(y, x)$

$\text{IsPartOf}(x, y) \longrightarrow \text{HasPart}(y, x)$



The RMFA Check

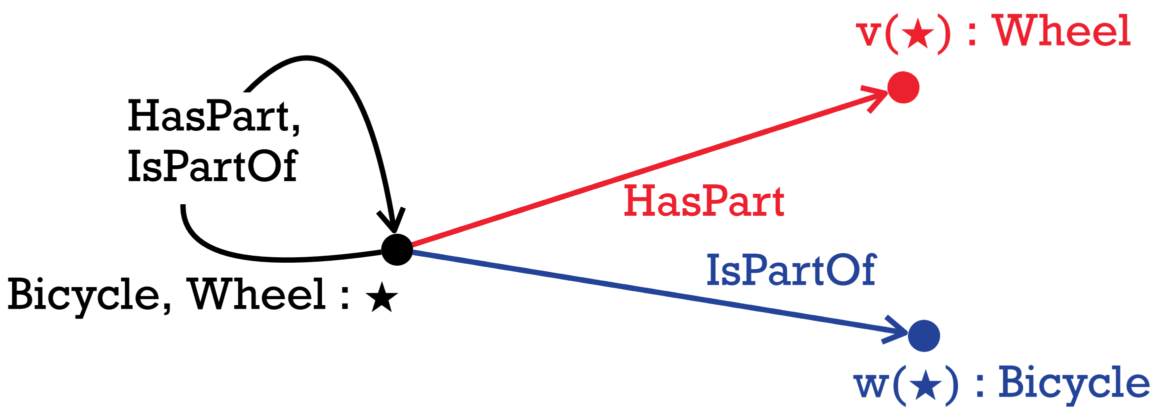
- Perform a chase like construction on the critical instance.
- Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
- Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y\, .\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y\, .\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$

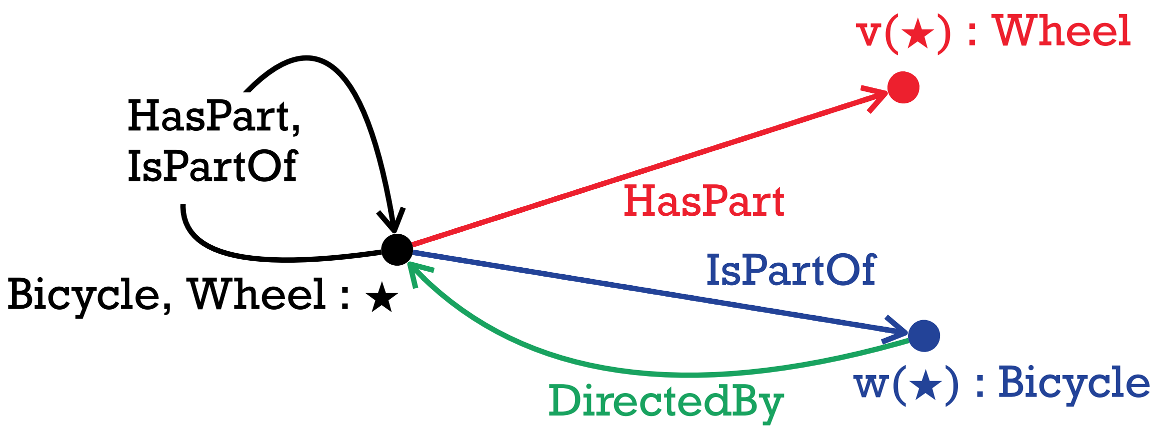


The RMFA Check

* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

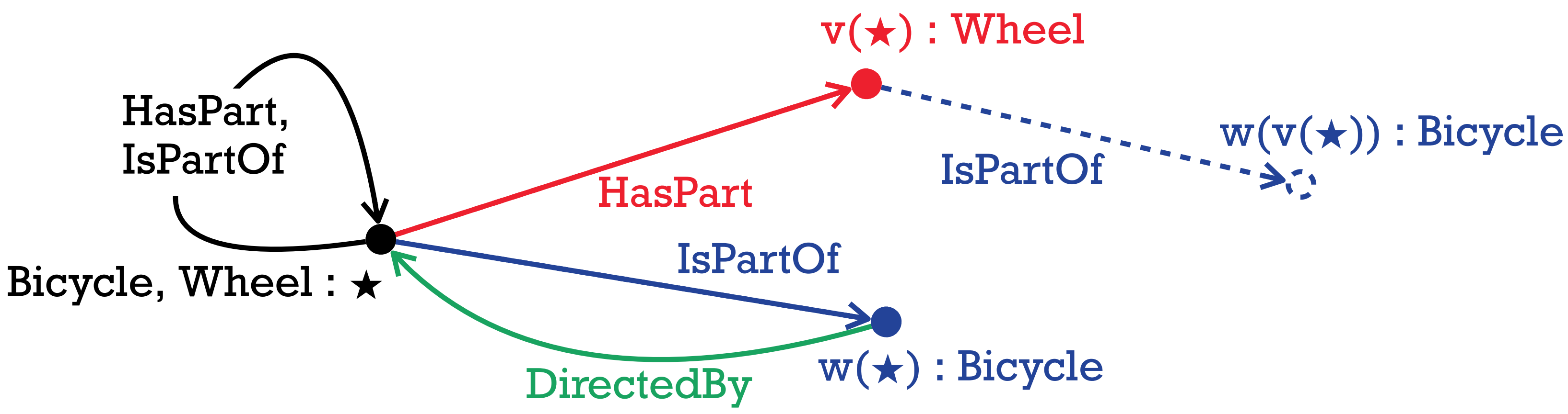


The RMFA Check

- Perform a chase like construction on the critical instance.
- Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
- Give up if the procedure does not stop before the occurrence of a cyclic term.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$

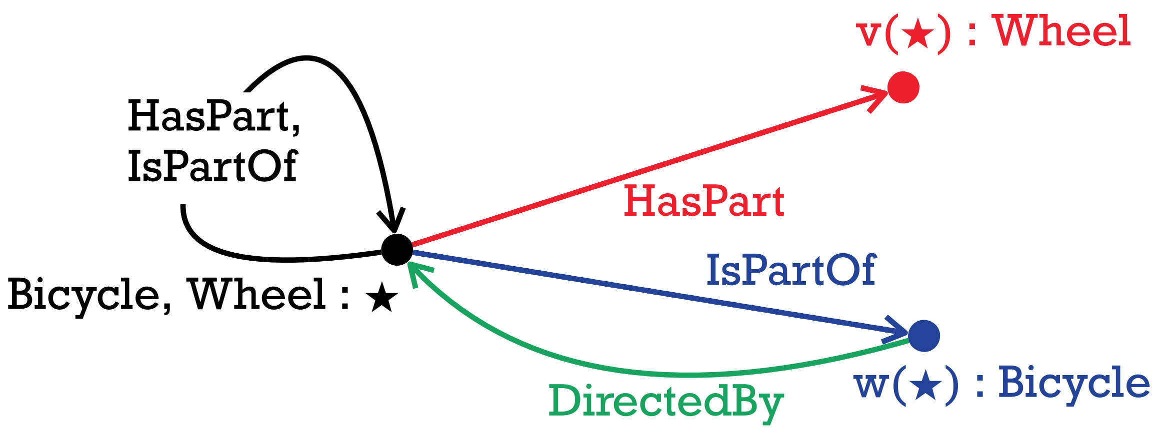


The RMFA Check

* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v) \qquad \mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$$

$$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w) \qquad \mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$$



The RMFA Check

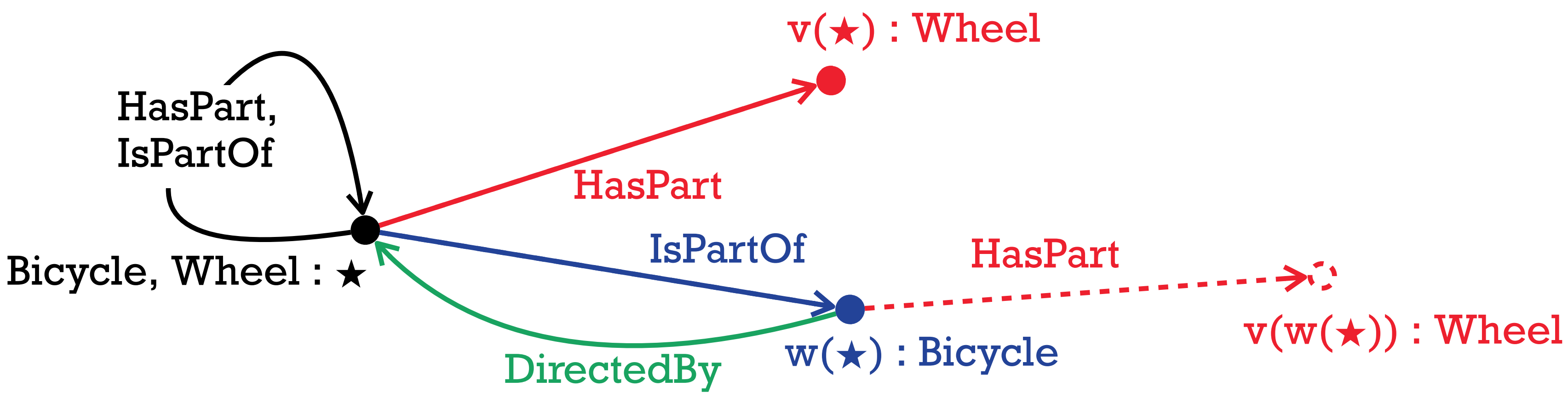
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \, . \, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \, . \, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The RMFA Check

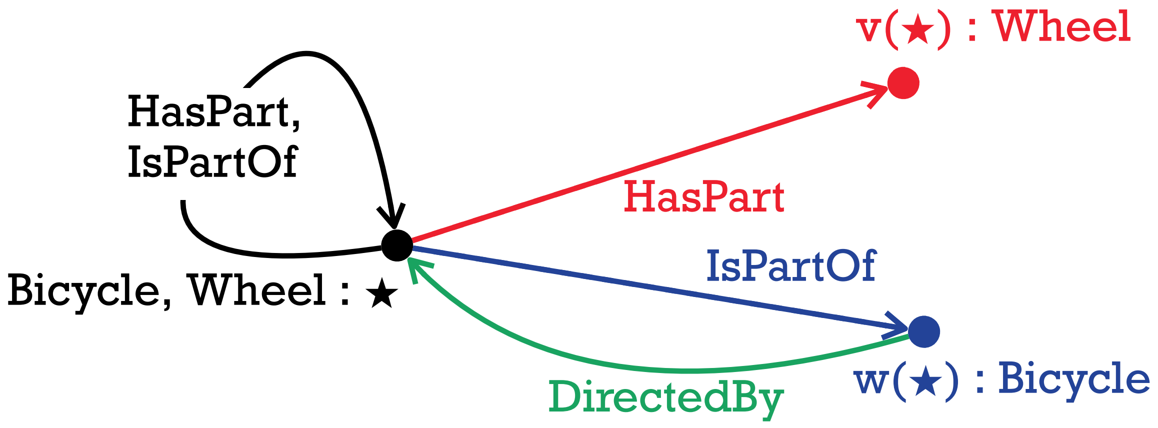
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y\,.\,\mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y\,.\,\mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The RMFA Check

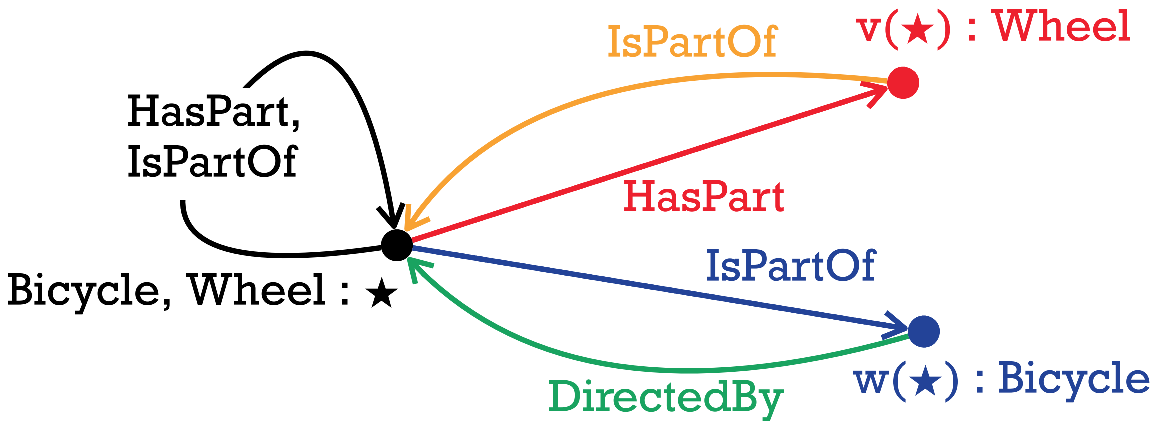
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



The RMFA Check

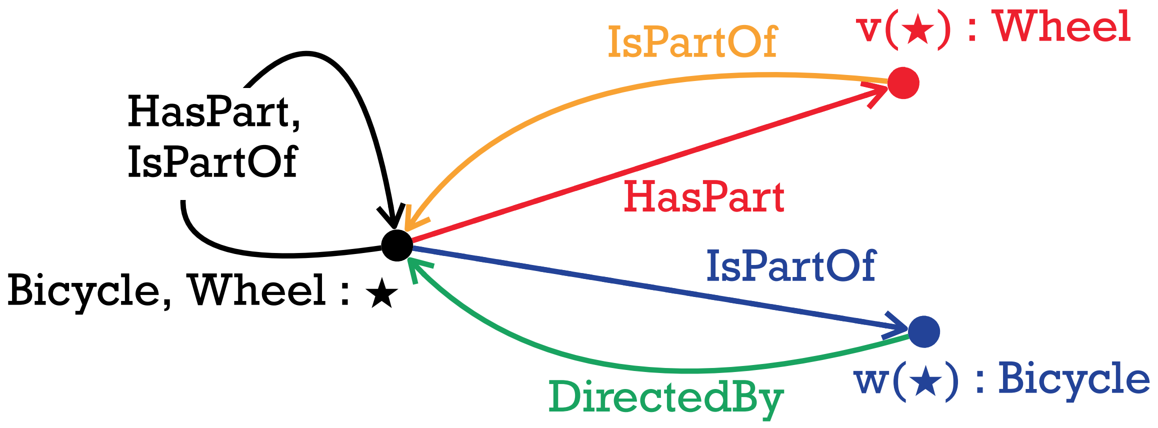
* Perform a chase like construction on the critical instance.
* Only apply an existential rule with respect to a substitution if this pair is not **blocked**.
* Give up if the procedure does not stop before the occurrence of a cyclic term.

$\mathrm{Bicycle}(x) \longrightarrow \exists y \,.\, \mathrm{HasPart}(x, v) \wedge \mathrm{Wheel}(v)$

$\mathrm{Wheel}(x) \longrightarrow \exists y \,.\, \mathrm{IsPartOf}(x, w) \wedge \mathrm{Bicycle}(w)$

$\mathrm{HasPart}(x, y) \longrightarrow \mathrm{IsPartOf}(y, x)$

$\mathrm{IsPartOf}(x, y) \longrightarrow \mathrm{HasPart}(y, x)$



RMFA

Real-world Coverage

Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y . R(x, y) \wedge B(y)$

Real-world Coverage

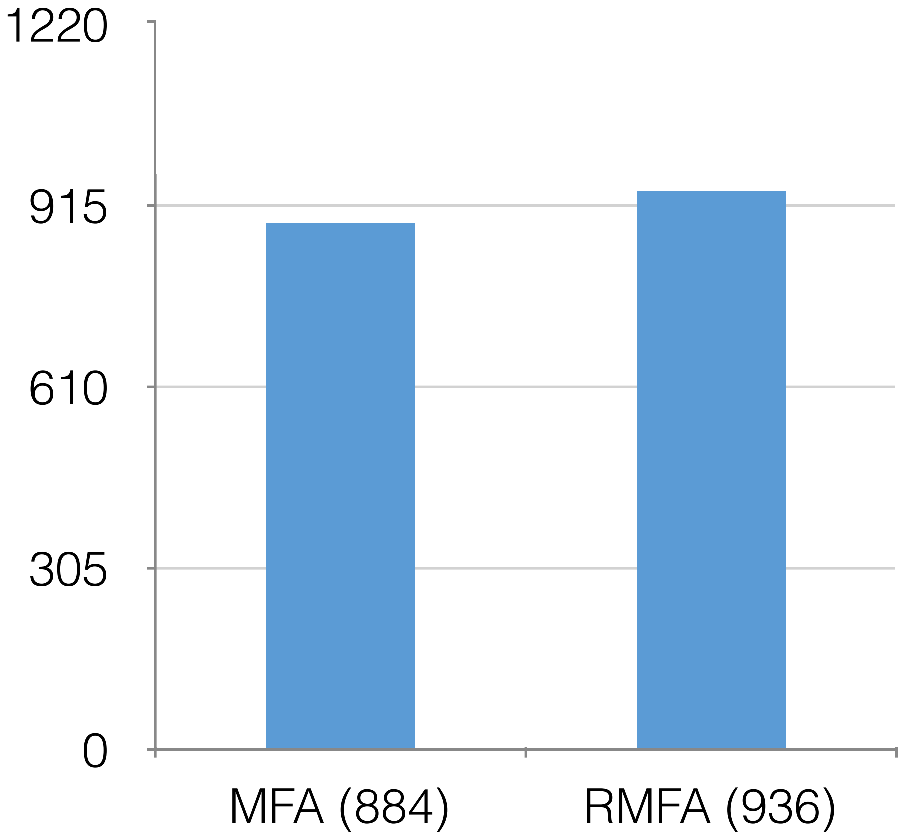
* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \to \exists y\,.\,R(x,y) \wedge B(y)$

* We also considered (all) the rule sets in the Oxford Ontology Library.

Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y \,.\, R(x, y) \wedge B(y)$

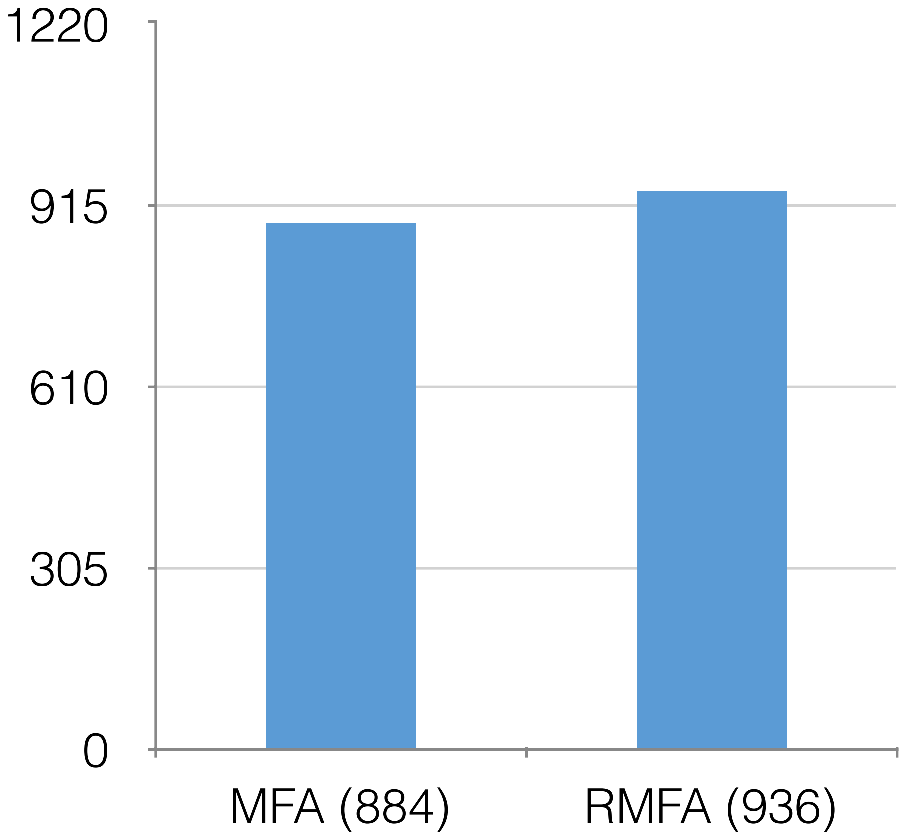
* We also considered (all) the rule sets in the Oxford Ontology Library.



Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y \, . \, R(x, y) \wedge B(y)$

* We also considered (all) the rule sets in the Oxford Ontology Library.



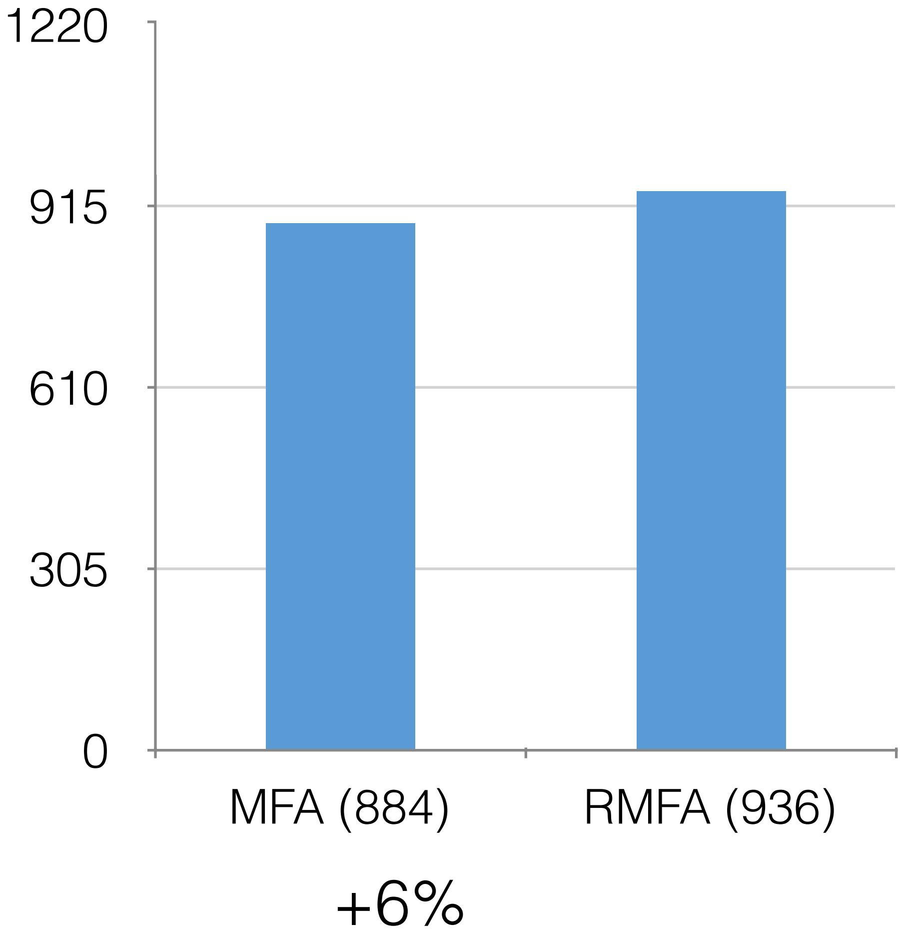
+6%

Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y . R(x, y) \wedge B(y)$

* We also considered (all) the rule sets in the Oxford Ontology Library.

* We developed a cyclicity notion, i.e., sufficient condition for chase non-termination: Restricted Model-Faithful Cyclicity (**RMFC**)

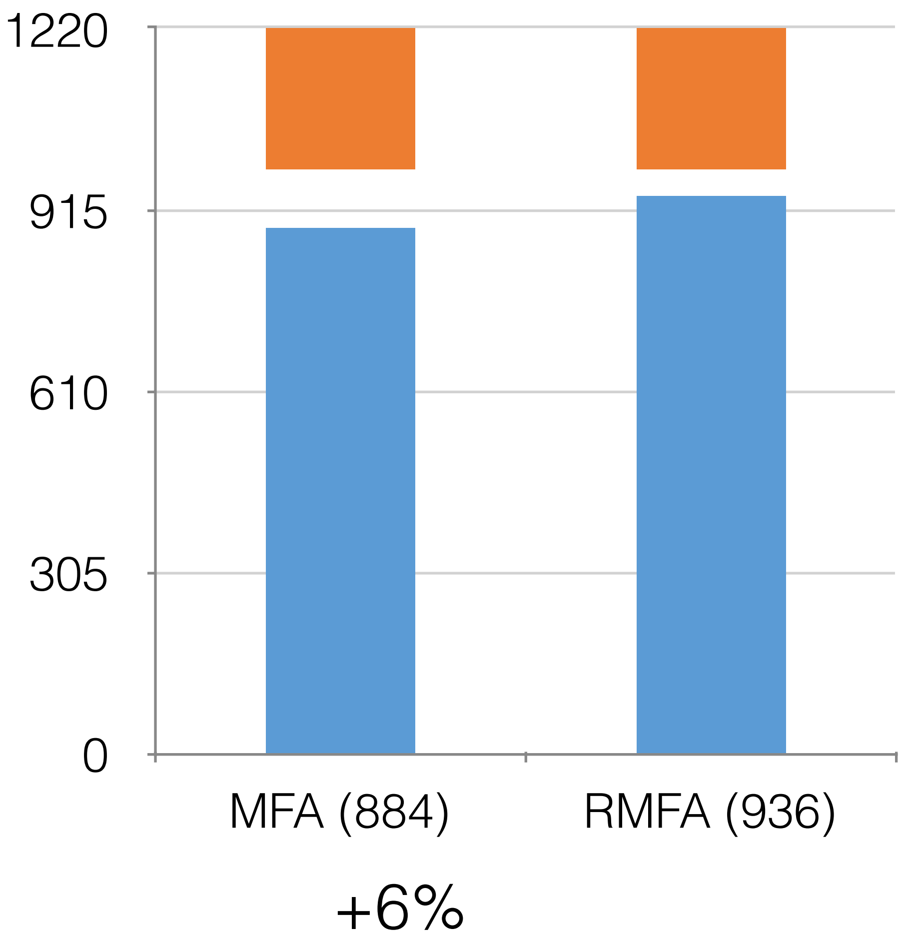


Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y \,.\, R(x,y) \wedge B(y)$

* We also considered (all) the rule sets in the Oxford Ontology Library.

* We developed a cyclicity notion, i.e., sufficient condition for chase non-termination: Restricted Model-Faithful Cyclicity (**RMFC**)

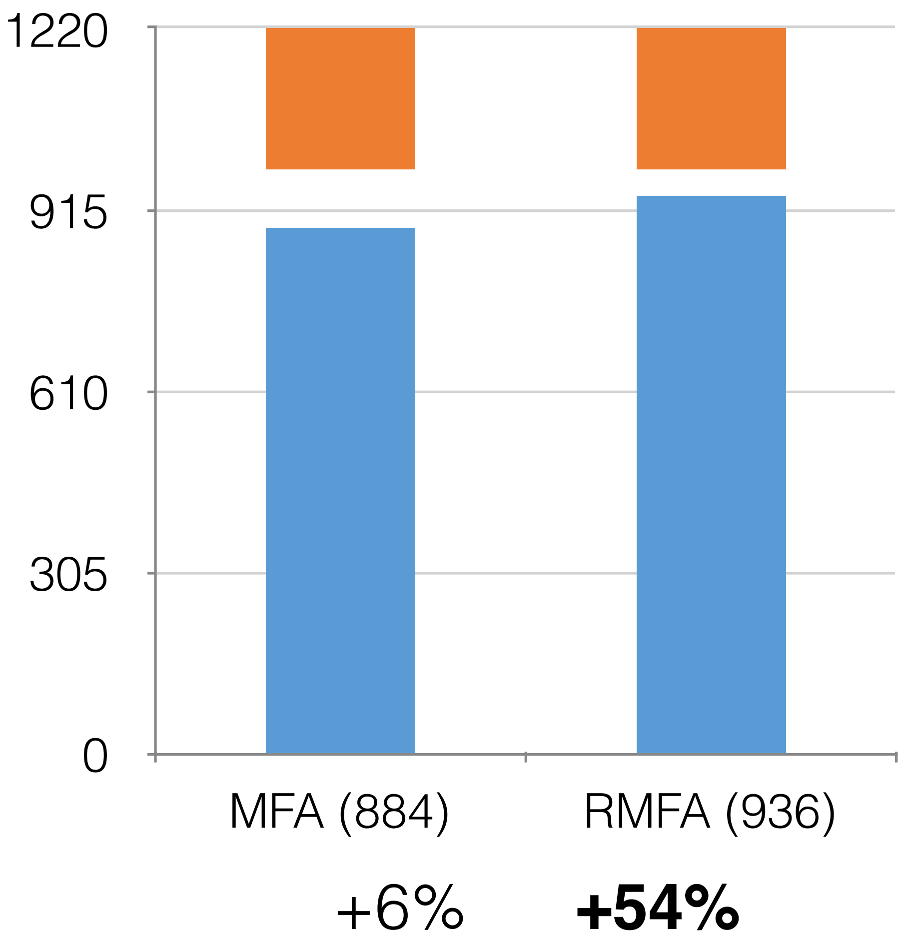


Real-world Coverage

* We selected all rule sets from the MOWLCorp with less than 1000 rules of the form $A(x) \rightarrow \exists y \, . \, R(x, y) \wedge B(y)$

* We also considered (all) the rule sets in the Oxford Ontology Library.

* We developed a cyclicity notion, i.e., sufficient condition for chase non-termination: Restricted Model-Faithful Cyclicity (**RMFC**)



Membership Checks

	RJA	RMSA	RMFA
No restrictions	ExpTime	ExpTime	2-ExpTime
Variables per rule bounded	P	P	2-ExpTime

Membership Checks

	RJA	RMSA	RMFA
No restrictions	ExpTime	ExpTime	2-ExpTime
Variables per rule bounded	P	P	2-ExpTime

BCQ entailment: 2-ExpTime

Ensuring Tractability of the Chase

Ensuring Tractability of the Chase

Tractable Query Answering for Expressive Ontologies and Existential Rules

David Carral, Irina Dragoste, and Markus Krötzsch

[ISWC 2017]

Existential Dependency Graph

$$\mathsf{A}(x) \to \exists y \,.\, \mathsf{S}(x, y) \wedge \mathsf{B}(y)$$

$$\mathsf{B}(x) \to \exists z \,.\, \mathsf{R}(x, z) \wedge \mathsf{D}(z)$$

$$\mathsf{D}(x) \to \mathsf{E}(x)$$

$$\mathsf{E}(x) \to \exists w \,.\, \mathsf{R}(x, w)$$

$$\mathsf{B}(x) \wedge \mathsf{C}(x) \to \mathsf{E}(x)$$

$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$

Existential Dependency Graph

$$\mathsf{A}(x) \to \mathsf{S}(x, y(x)) \wedge \mathsf{B}(y(x))$$

$$\mathsf{B}(x) \to \mathsf{R}(x, z(x)) \wedge \mathsf{D}(z(x))$$

$$\mathsf{D}(x) \to \mathsf{E}(x)$$

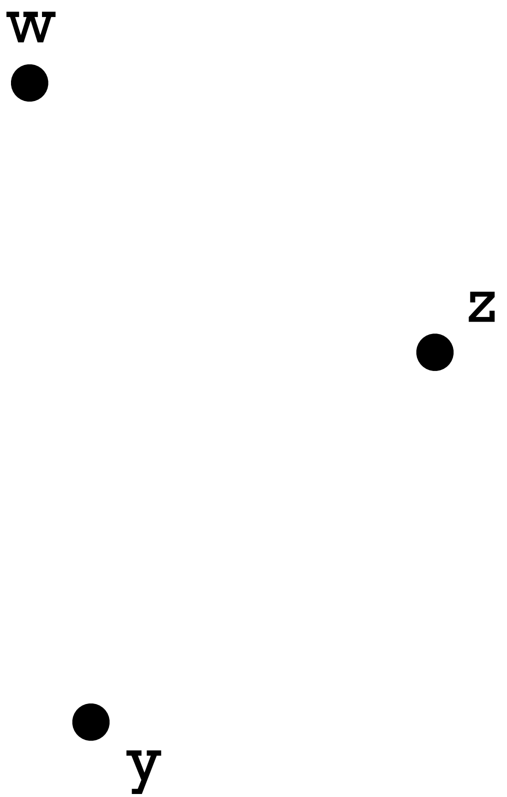
$$\mathsf{E}(x) \to \mathsf{R}(x, w(x))$$

$$\mathsf{B}(x) \wedge \mathsf{C}(x) \to \mathsf{E}(x)$$

$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$

Existential Dependency Graph

$$\mathsf{A}(x) \to \mathsf{S}(x, y(x)) \land \mathsf{B}(y(x))$$
$$\mathsf{B}(x) \to \mathsf{R}(x, z(x)) \land \mathsf{D}(z(x))$$
$$\mathsf{D}(x) \to \mathsf{E}(x)$$
$$\mathsf{E}(x) \to \mathsf{R}(x, w(x))$$
$$\mathsf{B}(x) \land \mathsf{C}(x) \to \mathsf{E}(x)$$
$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$



Existential Dependency Graph

$$A(x) \rightarrow S(x, y(x)) \wedge B(y(x))$$

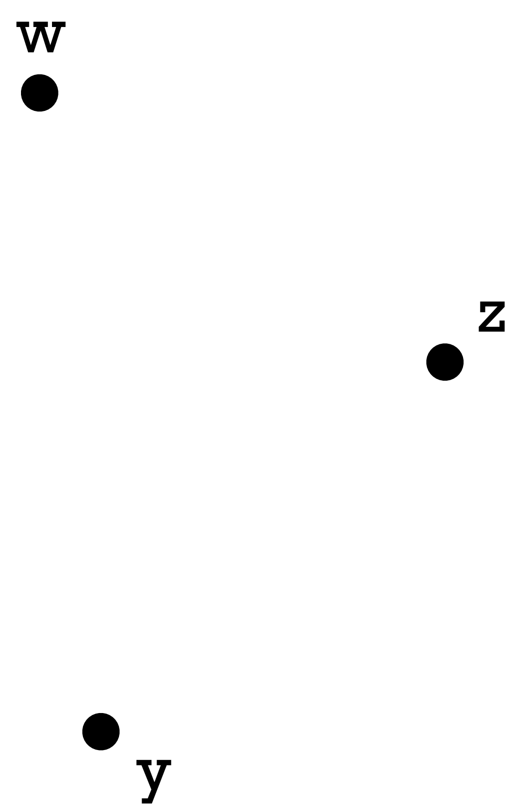
$$B(x) \rightarrow R(x, z(x)) \wedge D(z(x))$$

$$D(x) \rightarrow E(x)$$

$$E(x) \rightarrow R(x, w(x))$$

$$B(x) \wedge C(x) \rightarrow E(x)$$

$$S(x, y) \rightarrow C(x)$$



Existential Dependency Graph

$$\boxed{\begin{aligned} \mathsf{A}(x) &\to \mathsf{S}(x, y(x)) \land \mathsf{B}(y(x)) \\ \mathsf{B}(x) &\to \mathsf{R}(x, z(x)) \land \mathsf{D}(z(x)) \end{aligned}}$$

$$\mathsf{D}(x) \to \mathsf{E}(x)$$

$$\mathsf{E}(x) \to \mathsf{R}(x, w(x))$$

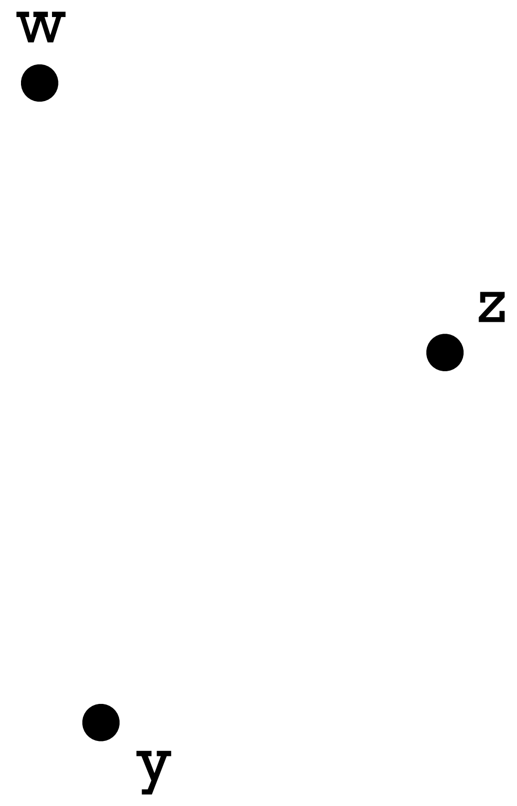
$$\mathsf{B}(x) \land \mathsf{C}(x) \to \mathsf{E}(x)$$

$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$

$\mathsf{A}(c)$

$\mathsf{S}(c, y(c)), B(y(c))$

$\mathsf{R}(y(c), z(y(c))), D(z(y(c)))$



Existential Dependency Graph

$$\boxed{\begin{array}{l}\mathsf{A}(x) \to \mathsf{S}(x, y(x)) \wedge \mathsf{B}(y(x)) \\ \mathsf{B}(x) \to \mathsf{R}(x, z(x)) \wedge \mathsf{D}(z(x))\end{array}}$$

$$\mathsf{D}(x) \to \mathsf{E}(x)$$

$$\mathsf{E}(x) \to \mathsf{R}(x, w(x))$$

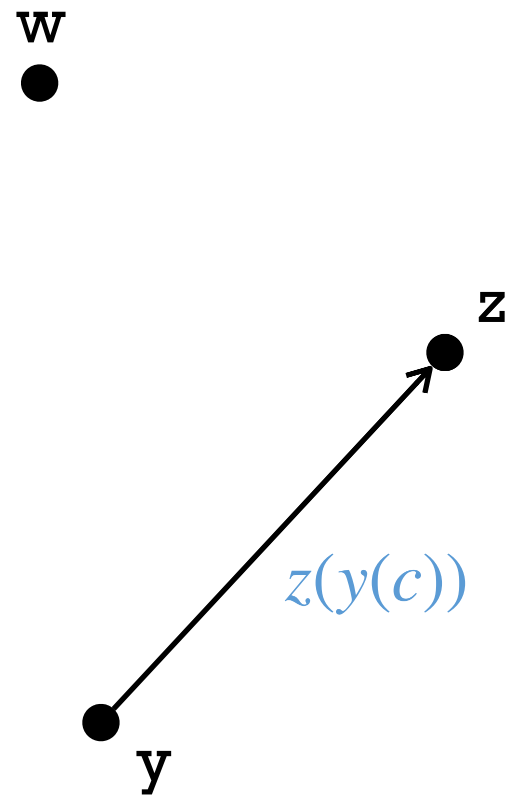
$$\mathsf{B}(x) \wedge \mathsf{C}(x) \to \mathsf{E}(x)$$

$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$

$\mathsf{A}(c)$

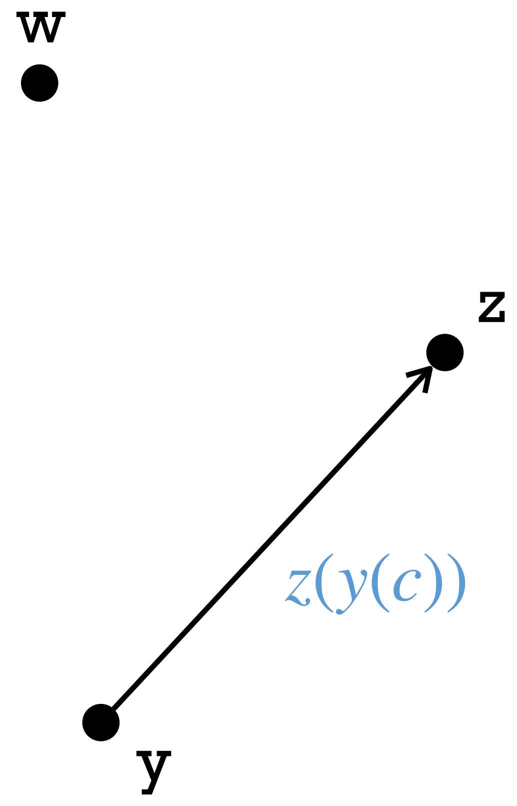
$\mathsf{S}(c, y(c)), B(y(c))$

$\mathsf{R}(y(c), z(y(c))), D(z(y(c)))$



Existential Dependency Graph

$$\mathsf{A}(x) \to \mathsf{S}(x, y(x)) \land \mathsf{B}(y(x))$$
$$\mathsf{B}(x) \to \mathsf{R}(x, z(x)) \land \mathsf{D}(z(x))$$
$$\mathsf{D}(x) \to \mathsf{E}(x)$$
$$\mathsf{E}(x) \to \mathsf{R}(x, w(x))$$
$$\mathsf{B}(x) \land \mathsf{C}(x) \to \mathsf{E}(x)$$
$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$



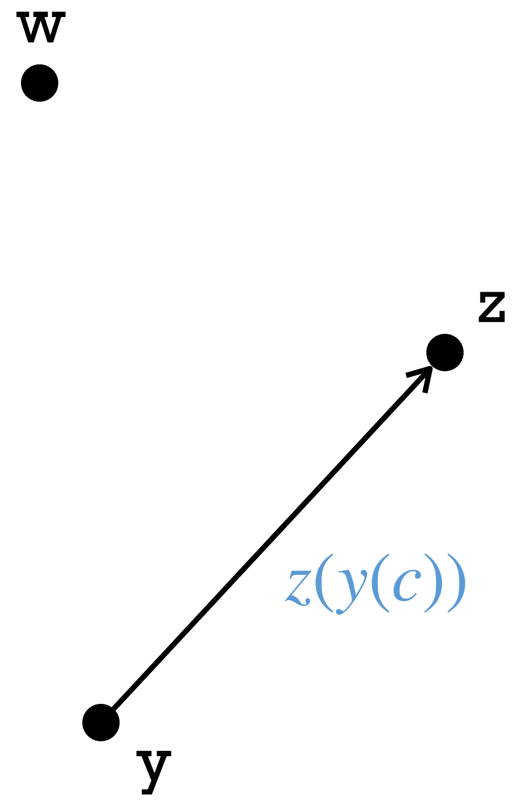
Existential Dependency Graph

$$A(x) \to S(x, y(x)) \wedge B(y(x))$$

$$\boxed{\begin{array}{l} B(x) \to R(x, z(x)) \wedge D(z(x)) \\ D(x) \to E(x) \\ E(x) \to R(x, w(x)) \end{array}}$$

$$B(x) \wedge C(x) \to E(x)$$

$$S(x, y) \to C(x)$$



Existential Dependency Graph

$$A(x) \to S(x, y(x)) \wedge B(y(x))$$

$$B(x) \to R(x, z(x)) \wedge D(z(x))$$

$$D(x) \to E(x)$$

$$E(x) \to R(x, w(x))$$

$$B(x) \wedge C(x) \to E(x)$$

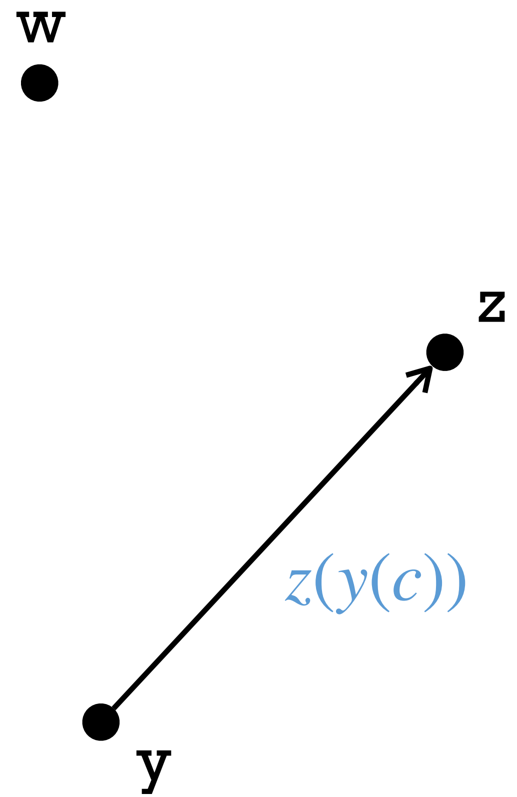
$$S(x, y) \to C(x)$$

$B(c)$

$R(c, z(c)), D(z(c)),$

$E(z(c)),$

$R(z(c), w(z(c)))$



Existential Dependency Graph

$$\mathsf{A}(x) \to \mathsf{S}(x, y(x)) \wedge \mathsf{B}(y(x))$$

$$\boxed{\begin{array}{l}\mathsf{B}(x) \to \mathsf{R}(x, z(x)) \wedge \mathsf{D}(z(x)) \\ \mathsf{D}(x) \to \mathsf{E}(x) \\ \mathsf{E}(x) \to \mathsf{R}(x, w(x))\end{array}}$$

$$\mathsf{B}(x) \wedge \mathsf{C}(x) \to \mathsf{E}(x)$$

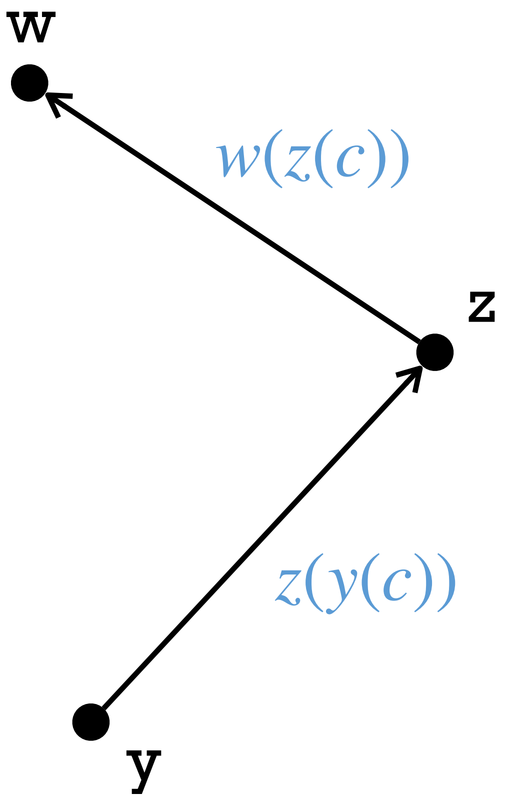
$$\mathsf{S}(x, y) \to \mathsf{C}(x)$$

$\mathsf{B}(c)$

$\mathsf{R}(c, z(c)), D(z(c)),$

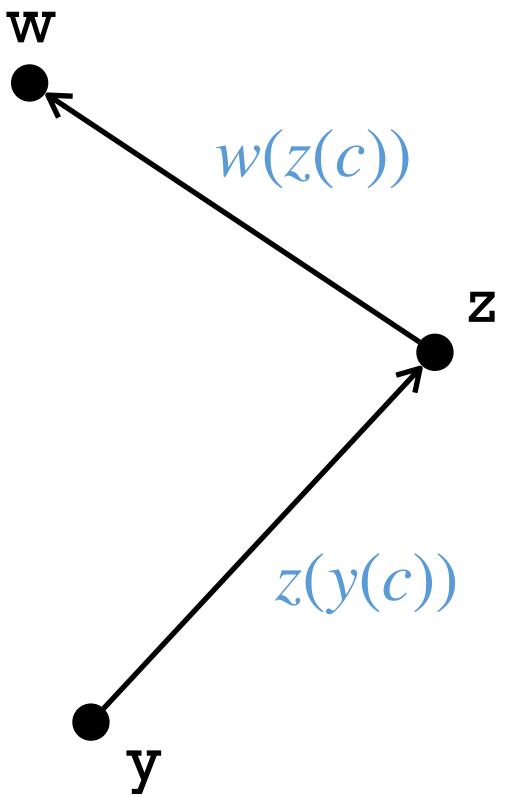
$\mathsf{E}(z(c)),$

$\mathsf{R}(z(c), w(z(c)))$



Existential Dependency Graph

$$A(x) \rightarrow S(x, y(x)) \wedge B(y(x))$$
$$B(x) \rightarrow R(x, z(x)) \wedge D(z(x))$$
$$D(x) \rightarrow E(x)$$
$$E(x) \rightarrow R(x, w(x))$$
$$B(x) \wedge C(x) \rightarrow E(x)$$
$$S(x, y) \rightarrow C(x)$$



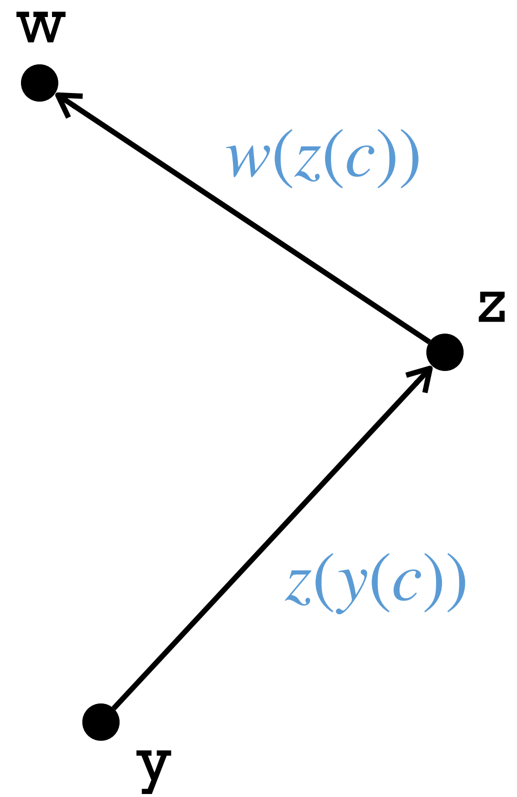
Existential Dependency Graph

$$\boxed{A(x) \rightarrow S(x, y(x)) \wedge B(y(x))}$$

$$B(x) \rightarrow R(x, z(x)) \wedge D(z(x))$$

$$D(x) \rightarrow E(x)$$

$$\boxed{\begin{array}{rcl} E(x) & \rightarrow & R(x, w(x)) \\ B(x) \wedge C(x) & \rightarrow & E(x) \\ S(x, y) & \rightarrow & C(x) \end{array}}$$



Existential Dependency Graph

$$\boxed{\mathsf{A}(x) \rightarrow \mathsf{S}(x, y(x)) \wedge \mathsf{B}(y(x))}$$

$$\mathsf{B}(x) \rightarrow \mathsf{R}(x, z(x)) \wedge \mathsf{D}(z(x))$$

$$\mathsf{D}(x) \rightarrow \mathsf{E}(x)$$

$$\boxed{\begin{array}{r}
\mathsf{E}(x) \rightarrow \mathsf{R}(x, w(x)) \\
\mathsf{B}(x) \wedge \mathsf{C}(x) \rightarrow \mathsf{E}(x) \\
\mathsf{S}(x, y) \rightarrow \mathsf{C}(x)
\end{array}}$$

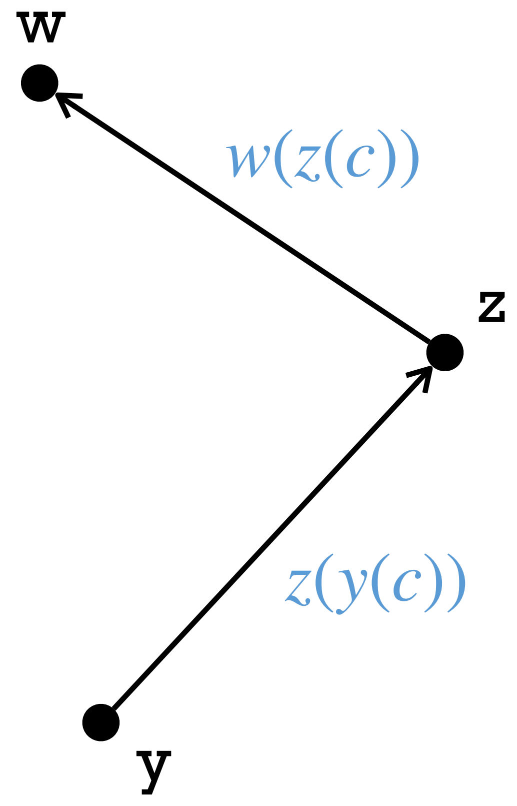
$\mathsf{A}(c)$

$\mathsf{S}(c, y(c)), \mathsf{B}(y(c)),$

$\mathsf{C}(y(c)),$

$\mathsf{E}(y(c)),$

$\mathsf{R}(y(c), w(y(c)))$



Existential Dependency Graph

$$\boxed{\mathsf{A}(x) \rightarrow \mathsf{S}(x, y(x)) \wedge \mathsf{B}(y(x))}$$

$$\mathsf{B}(x) \rightarrow \mathsf{R}(x, z(x)) \wedge \mathsf{D}(z(x))$$

$$\mathsf{D}(x) \rightarrow \mathsf{E}(x)$$

$$\boxed{\begin{aligned} \mathsf{E}(x) &\rightarrow \mathsf{R}(x, w(x)) \\ \mathsf{B}(x) \wedge \mathsf{C}(x) &\rightarrow \mathsf{E}(x) \\ \mathsf{S}(x, y) &\rightarrow \mathsf{C}(x) \end{aligned}}$$

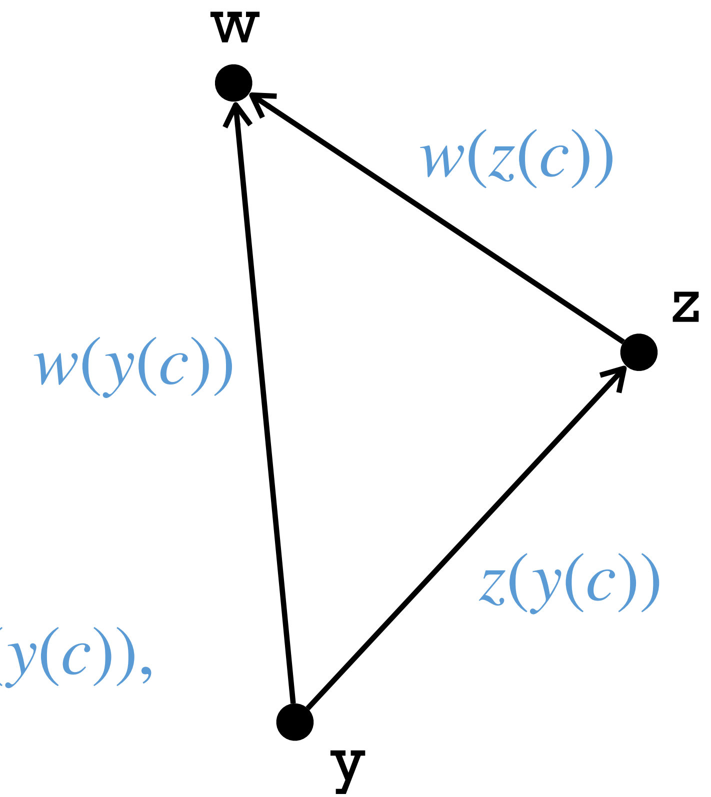
$\mathsf{A}(c)$

$\mathsf{S}(c, y(c)), \mathsf{B}(y(c)),$

$\mathsf{C}(y(c)),$

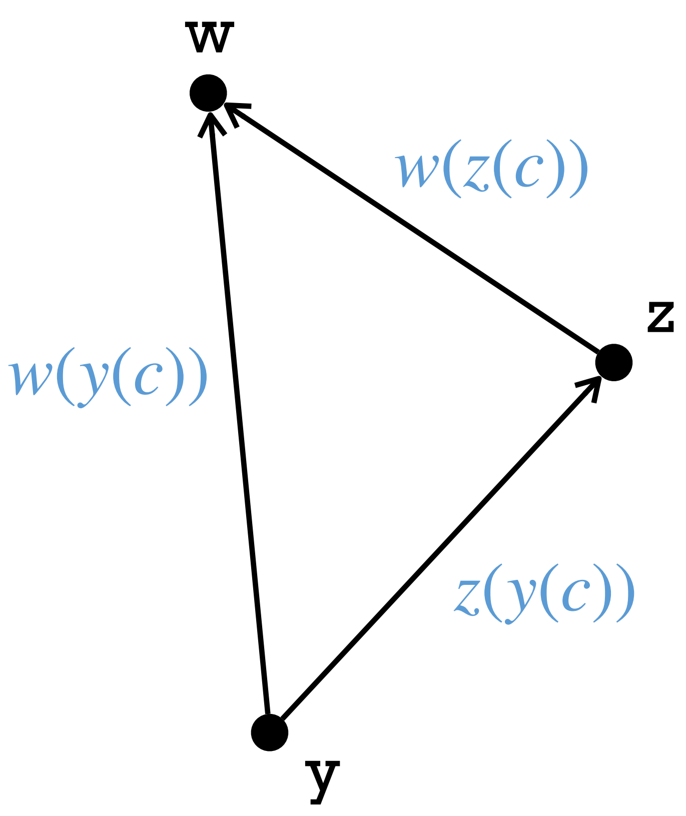
$\mathsf{E}(y(c)),$

$\mathsf{R}(y(c), w(y(c)))$



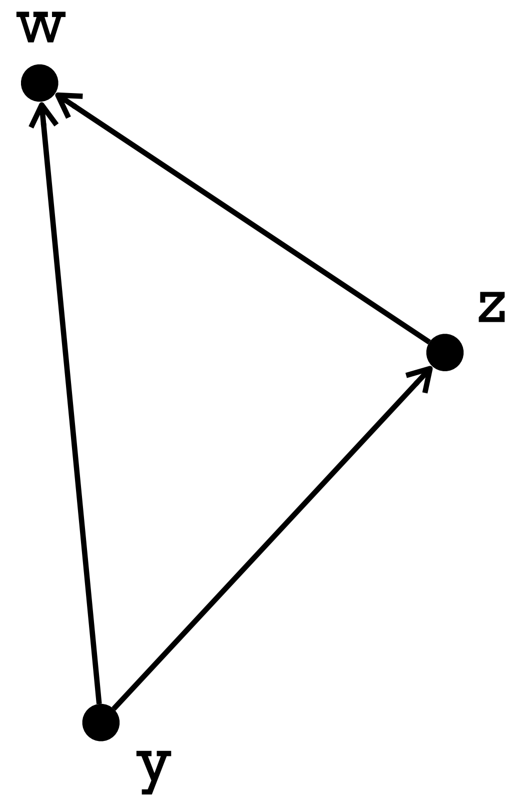
Existential Dependency Graph

$$A(x) \to S(x, y(x)) \wedge B(y(x))$$
$$B(x) \to R(x, z(x)) \wedge D(z(x))$$
$$D(x) \to E(x)$$
$$E(x) \to R(x, w(x))$$
$$B(x) \wedge C(x) \to E(x)$$
$$S(x, y) \to C(x)$$

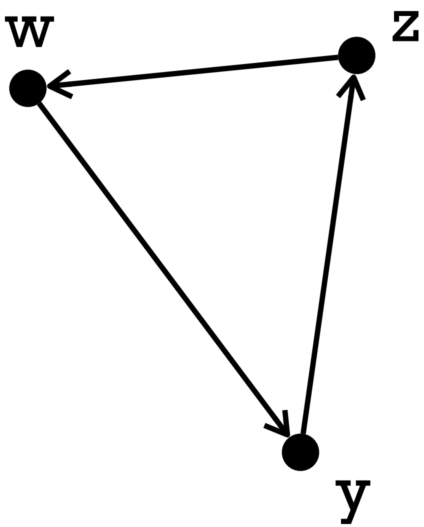


Existential Dependency Graph

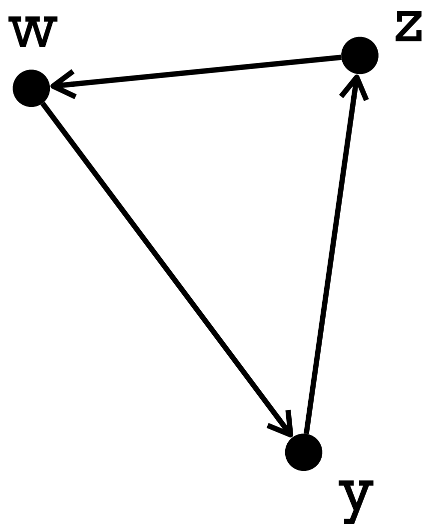
$$A(x) \to S(x, y(x)) \wedge B(y(x))$$
$$B(x) \to R(x, z(x)) \wedge D(z(x))$$
$$D(x) \to E(x)$$
$$E(x) \to R(x, w(x))$$
$$B(x) \wedge C(x) \to E(x)$$
$$S(x, y) \to C(x)$$



(a) Acyclicity

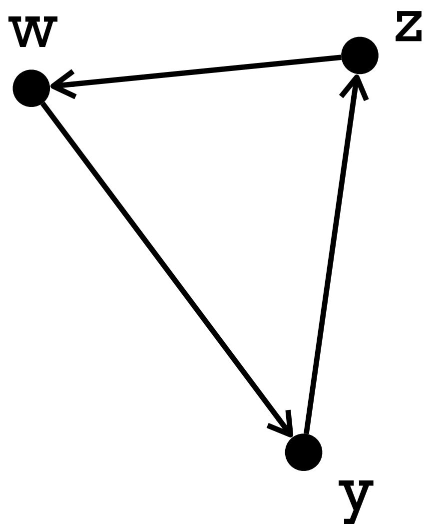


(a) Acyclicity



$z(c)$

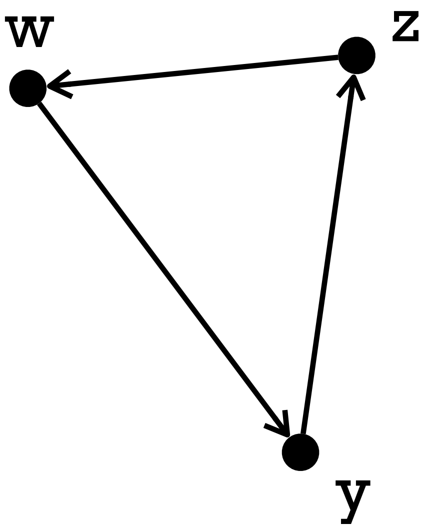
(a) Acyclicity



$z(c)$

$w(z(c))$

(a) Acyclicity

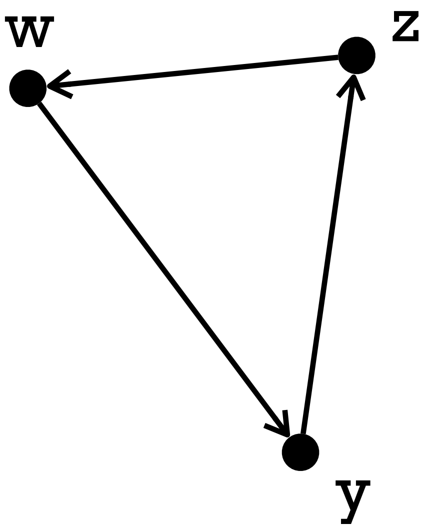


$z(c)$

$y(w(z(c)))$

$w(z(c))$

(a) Acyclicity



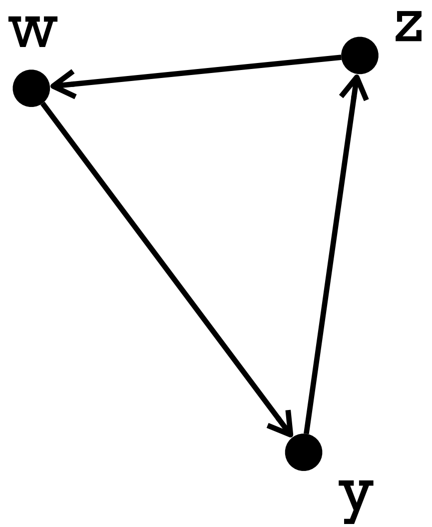
$z(c)$

$y(w(z(c)))$

$w(z(c))$

$z(y(w(z(c))))$

(a) Acyclicity



$z(c)$

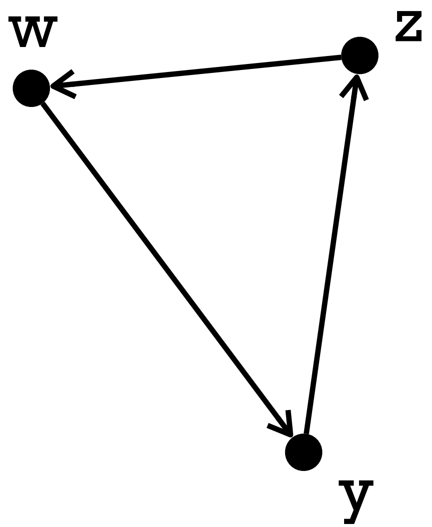
$y(w(z(c)))$

$w(z(c))$

$w(z(y(w(z(c)))))$

$z(y(w(z(c))))$

(a) Acyclicity



$z(c)$

$y(w(z(c)))$

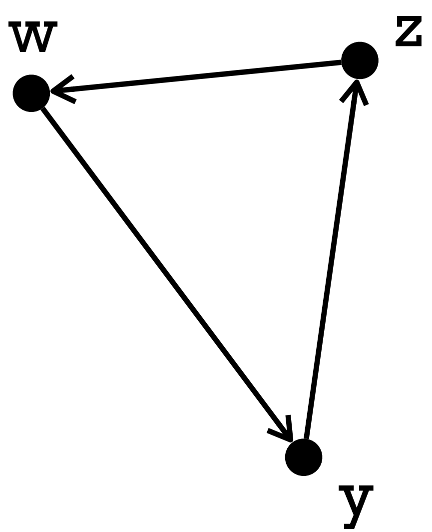
$w(z(c))$

$w(z(y(w(z(c)))))$

$z(y(w(z(c))))$

$\ldots\ldots\ldots$

(a) Acyclicity



Remark. If the existential dependency graph of a given set of rules is acyclic, then the set of terms introduced during the computation of the chase is finite.

(f) Arity at Most 1

$$\mathsf{Film}(x) \rightarrow \exists y\,.\,\mathsf{IsFilmDirectedBy}(x, y) \wedge \mathsf{Director}(y)$$

$$\mathsf{A}(x) \wedge \mathsf{B}(x, w) \wedge \mathsf{C}(x, z) \rightarrow \exists z\,.\,\mathsf{R}(x, w, z)$$

(f) Arity at Most 1

$$\mathsf{Film}(x) \rightarrow \exists y\,.\,\mathsf{IsFilmDirectedBy}(x, y) \wedge \mathsf{Director}(y)$$

$$\mathsf{Film}(x) \rightarrow \mathsf{IsFilmDirectedBy}(x, y(x)) \wedge \mathsf{Director}(y(x))$$

$$\mathsf{A}(x) \wedge \mathsf{B}(x, y) \wedge \mathsf{C}(x, z) \rightarrow \exists z\,.\,\mathsf{R}(x, y, z)$$

$$\mathsf{A}(x) \wedge \mathsf{B}(x, w) \wedge \mathsf{C}(x, z) \rightarrow \mathsf{R}(x, w, z(x, w))$$

(f) Arity at Most 1

$$\mathsf{Film}(x) \to \exists y\,.\,\mathsf{IsFilmDirectedBy}(x, y) \wedge \mathsf{Director}(y)$$

$$\mathsf{Film}(x) \to \mathsf{IsFilmDirectedBy}(x, y(x)) \wedge \mathsf{Director}(y(x))$$

$$\mathsf{A}(x) \wedge \mathsf{B}(x, y) \wedge \mathsf{C}(x, z) \to \exists z\,.\,\mathsf{R}(x, y, z)$$

$$\mathsf{A}(x) \wedge \mathsf{B}(x, w) \wedge \mathsf{C}(x, z) \to \mathsf{R}(x, w, z(x, w))$$

Remark. *If the arity of every function symbol in the skolemisation of a program is at most 1, then every term in the chase is of the form* **`x1(...xn(c)...)`** *with* **`c`** *constant.*

(f) Arity at Most 1

Remark. If the arity of every function symbol in the skolemisation of a program is at most 1, then every term in the chase is of the form **`x1(...xn(c)...)`** *with* **`c`** *constant.*

(f) Arity at Most 1

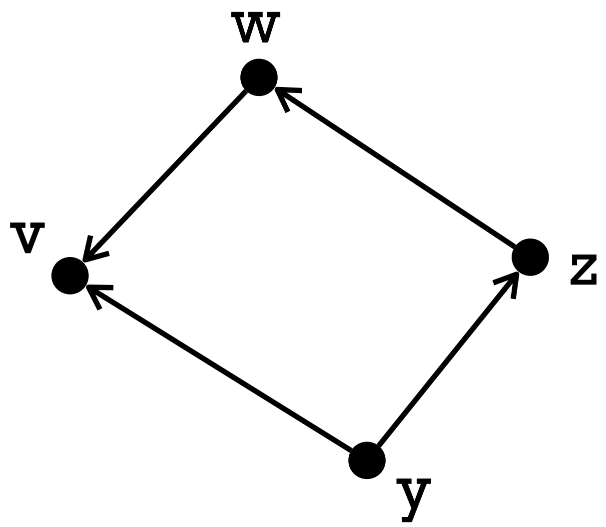
Remark. If the arity of every function symbol in the skolemisation of a program is at most 1, then every term in the chase is of the form **`x1(...xn(c)...)`** *with* **`c`** *constant.*

Corollary. Every term occurring in the chase corresponds to a path in the dependency graph and a constant.

(f) Arity at Most 1

Remark. If the arity of every function symbol in the skolemisation of a program is at most 1, then every term in the chase is of the form **`x1(...xn(c)...)`** *with* **`c`** *constant.*

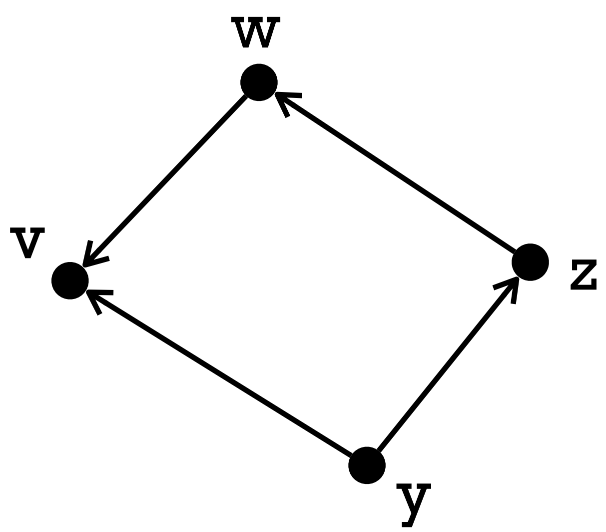
Corollary. Every term occurring in the chase corresponds to a path in the dependency graph and a constant.



(f) Arity at Most 1

Remark. *If the arity of every function symbol in the skolemisation of a program is at most 1, then every term in the chase is of the form* **x1(...xn(c)...)** *with* **c** *constant.*

Corollary. *Every term occurring in the chase corresponds to a path in the dependency graph and a constant.*



v(w(z(y(c))))
y(v(c))

Ensuring Tractability

Ensuring Tractability

(a) The dependency graph
is acyclic.

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

(w) The number of variables per rule is bounded.

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

(w) The number of variables per rule is bounded.

Rules can be applied in polynomial time

The number of facts is polynomial in the number of terms

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

(w) The number of variables per rule is bounded.

Rules can be applied in polynomial time

The number of facts is polynomial in the number of terms

The number of paths in the dependency graph is polynomial

Ensuring Tractability

(a) The dependency graph is acyclic.

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

(w) The number of variables per rule is bounded.

All skolem terms correspond to some path in the dependency graph and some constant

Rules can be applied in polynomial time

The number of facts is polynomial in the number of terms

The number of paths in the dependency graph is polynomial

Polynomiality

Ensuring Tractability

(a) The dependency graph is acyclic.

Polynomiality

All skolem terms correspond to some path in the dependency graph and some constant

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

Rules can be applied in polynomial time

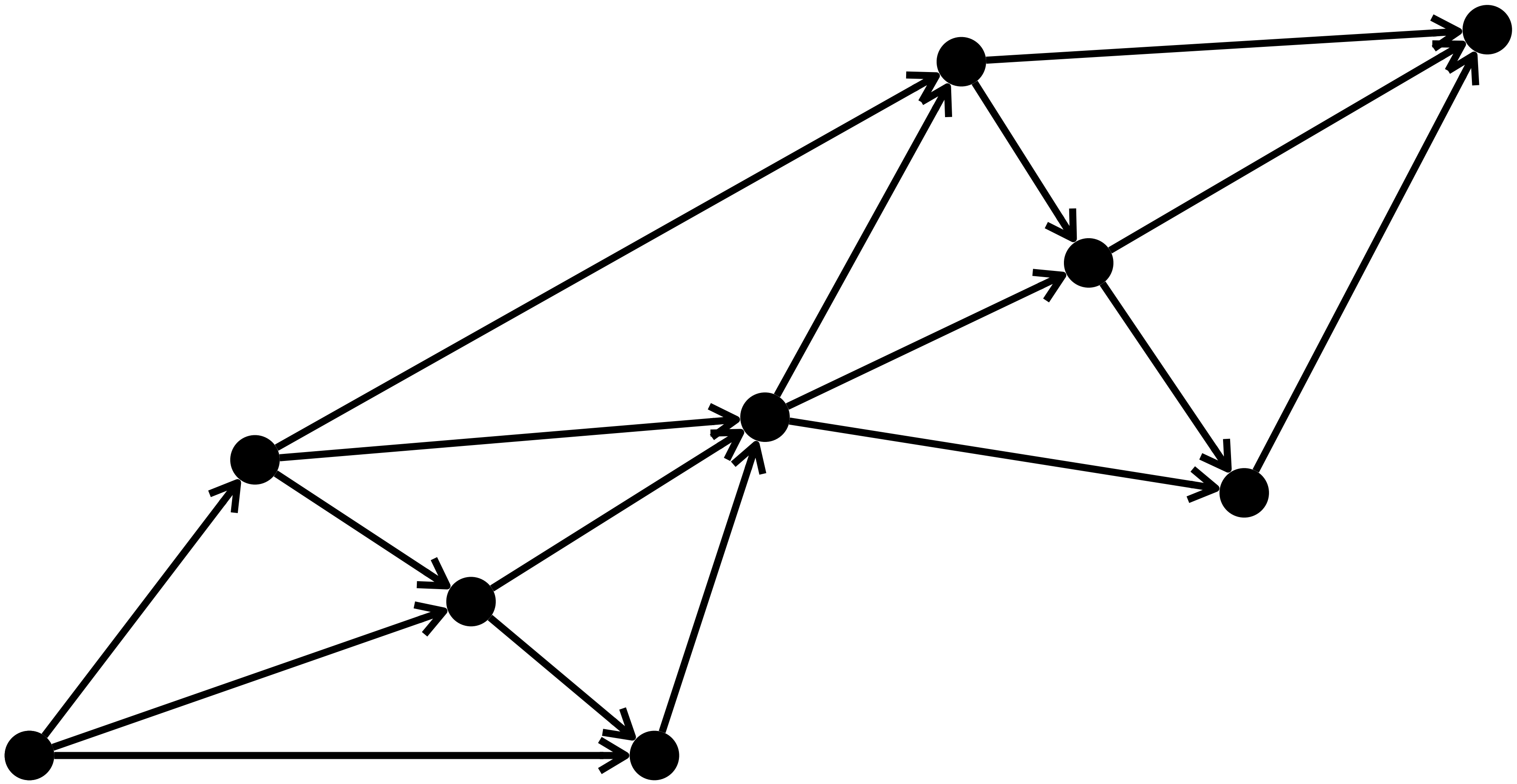
The number of paths in the dependency graph is polynomial

(w) The number of variables per rule is bounded.

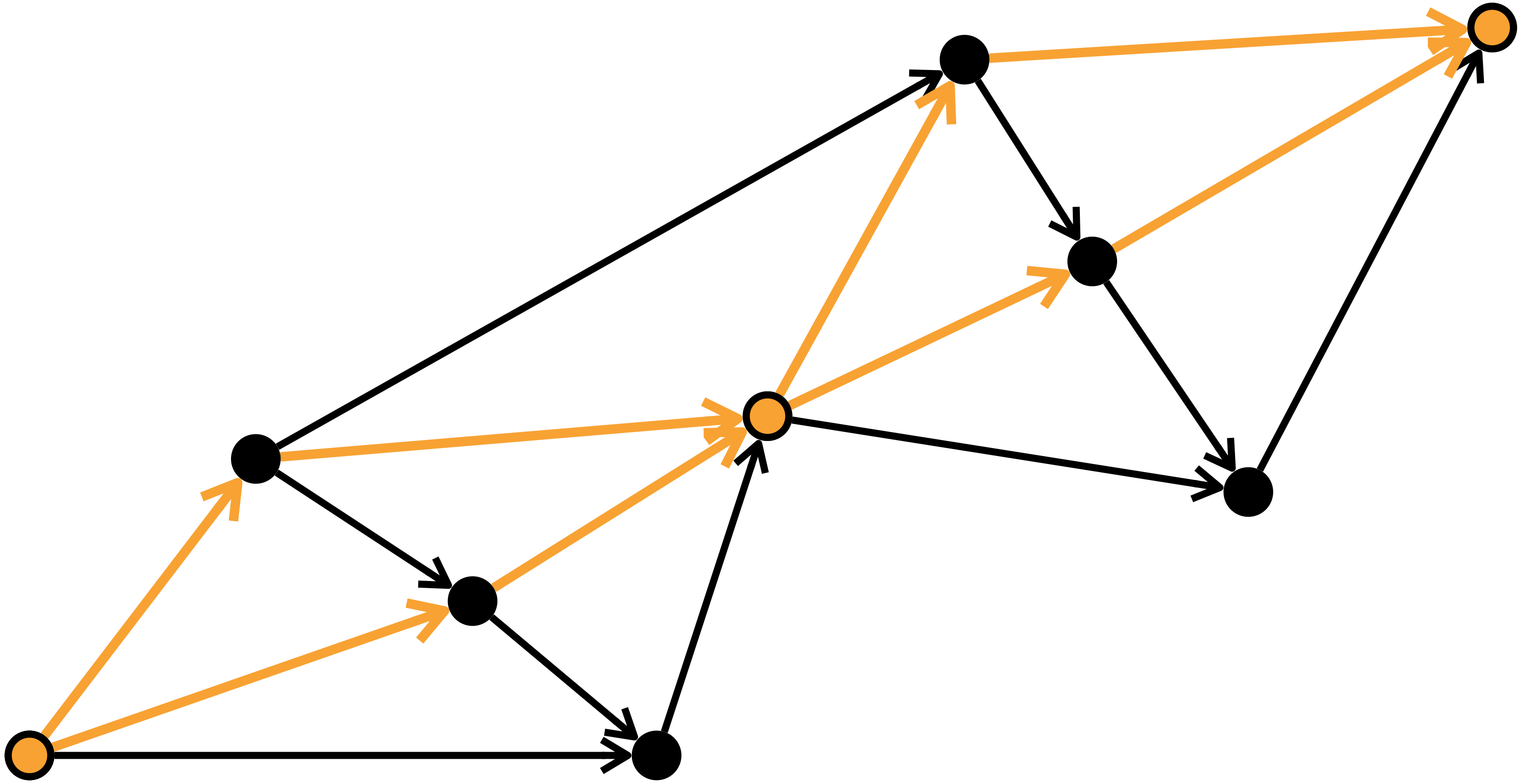
The number of facts is polynomial in the number of terms

(?)

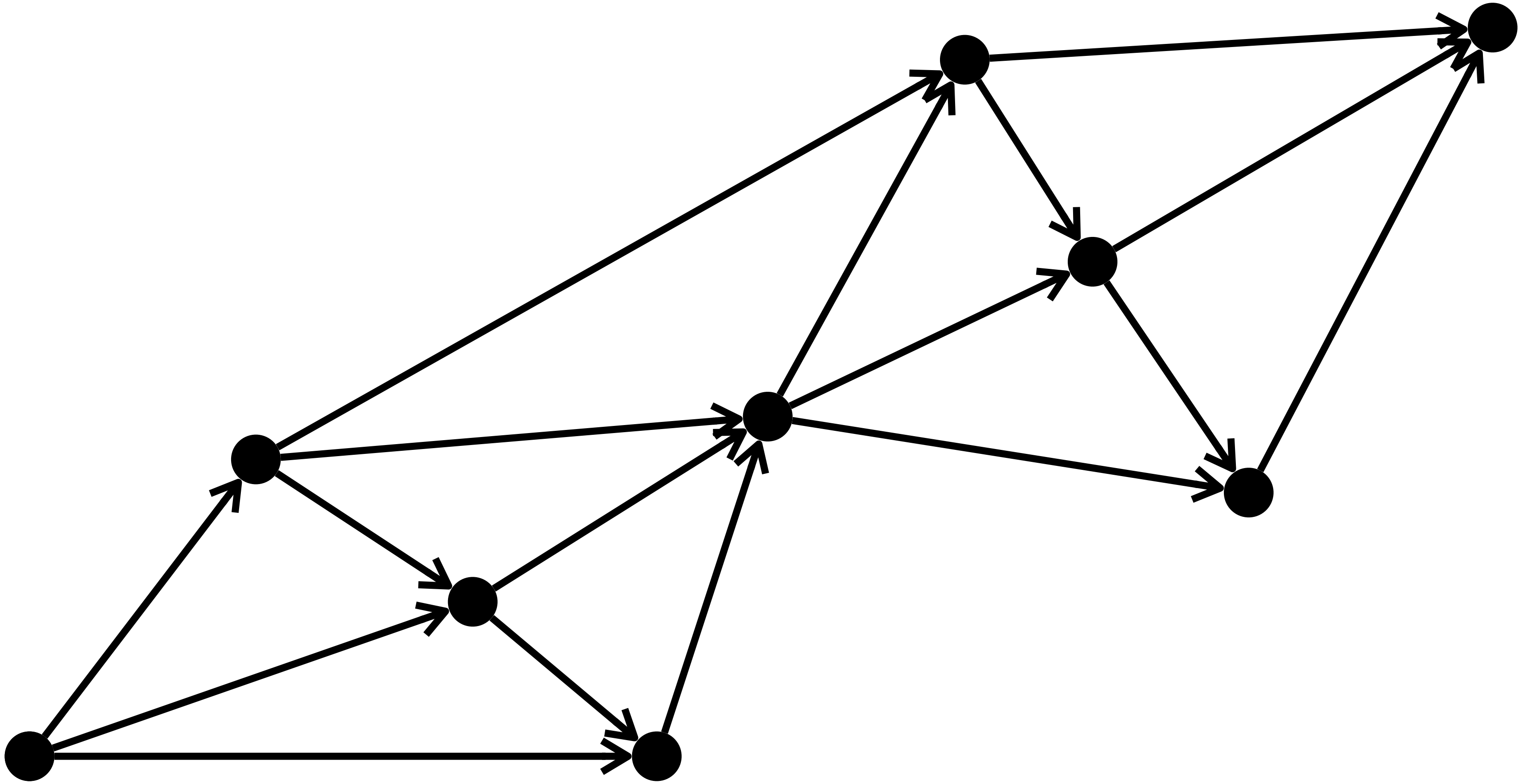
Braids



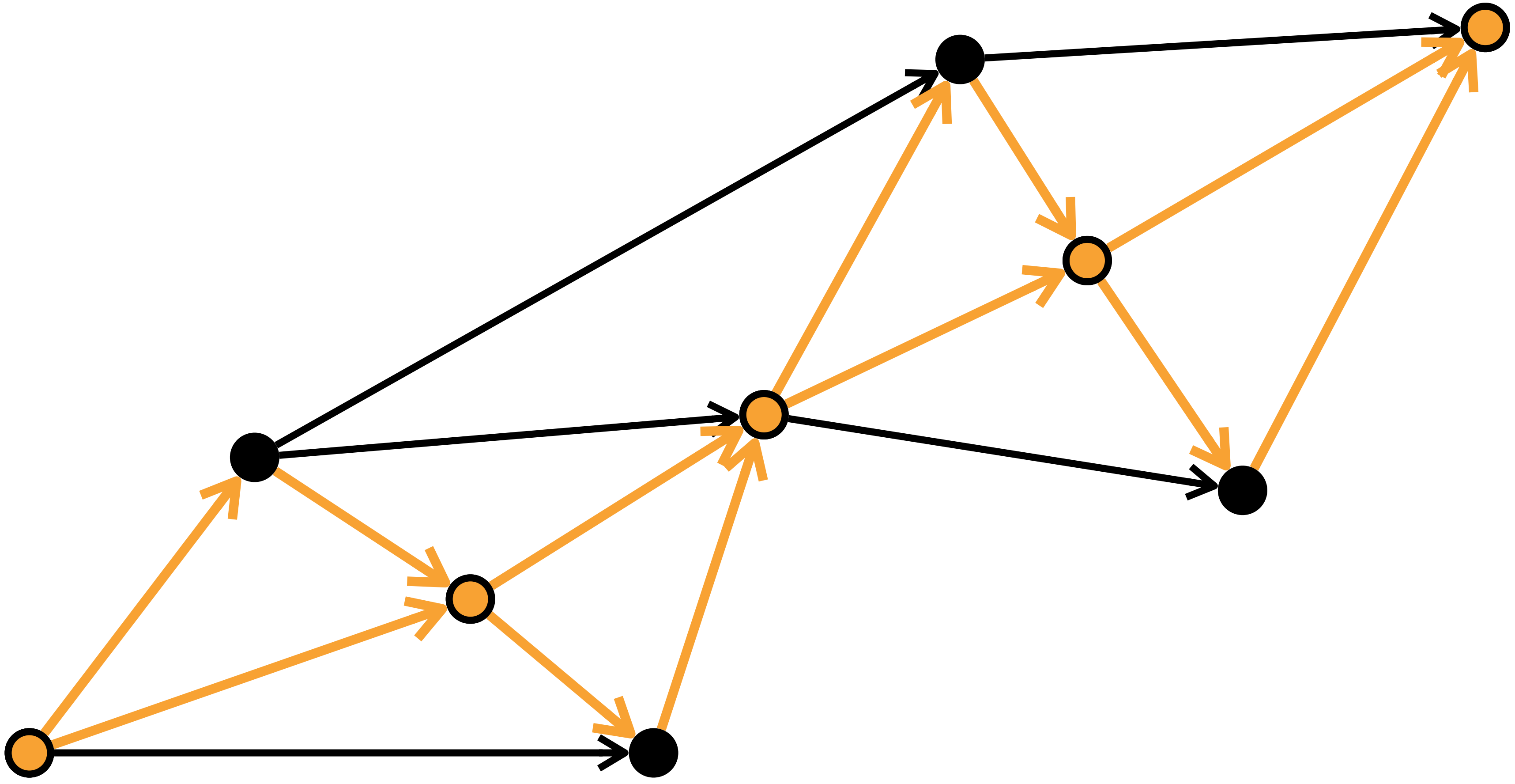
Braids



Braids



Braids



Ensuring Tractability

(a) The dependency graph is acyclic.

Polynomiality

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

(w) The number of variables per rule is bounded.

Rules can be applied in polynomial time

The number of facts is polynomial in the number of terms

The number of paths in the dependency graph is polynomial

(?)

Ensuring Tractability

(a) The dependency graph is acyclic.

Polynomiality

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

Rules can be applied in polynomial time

The number of paths in the dependency graph is polynomial

(w) The number of variables per rule is bounded.

The number of facts is polynomial in the number of terms

(b) The length of the braids in the dependency graph is bounded.

Ensuring Tractability

Caveats.
1. Fixed query size.
2. Horn rule set.

(a) The dependency graph is acyclic.

Polynomiality

(f) The arity of all function symbols in the skolemisation of the program is at most 1.

All skolem terms correspond to some path in the dependency graph and some constant

(w) The number of variables per rule is bounded.

Rules can be applied in polynomial time

The number of facts is polynomial in the number of terms

The number of paths in the dependency graph is polynomial

(b) The length of the braids in the dependency graph is bounded.

Real-world coverage: SRI Ontologies

$$
\begin{aligned}
A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B &\mapsto \bigwedge_{i=1}^{n} A_i(x) \to B(x) \\
A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n &\mapsto A(x) \to \bigvee_{i=1}^{n} B_i(x) \\
A \sqsubseteq \forall R . B &\mapsto A(y) \wedge R(x,y) \to B(x) \\
A \sqsubseteq \exists R . B &\mapsto A(x) \to \exists y . R(x,y) \wedge B(y) \\
R \sqsubseteq S &\mapsto R(x,y) \to S(x,y) \\
R \circ S \sqsubseteq V &\mapsto R(x,y) \wedge S(y,z) \to S(x,z) \\
R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S &\mapsto \bigwedge_{i=1}^{n} R_i(x,y) \to S(x,y) \\
A(a) &\mapsto A(a) \\
R(a,b) &\mapsto R(a,b)
\end{aligned}
$$

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \quad \mapsto \quad \bigwedge_{i=1}^{n} A_i(x) \to B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \quad \mapsto \quad A(x) \to \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R \,.\, B \quad \mapsto \quad A(y) \wedge R(x, y) \to B(x)$$

$$A \sqsubseteq \exists R \,.\, B \quad \mapsto \quad A(x) \to \exists y \,.\, R(x, y) \wedge B(y)$$

$$R \sqsubseteq S \quad \mapsto \quad R(x, y) \to S(x, y)$$

$$R \circ S \sqsubseteq V \quad \mapsto \quad R(x, y) \wedge S(y, z) \to S(x, z)$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \quad \mapsto \quad \bigwedge_{i=1}^{n} R_i(x, y) \to S(x, y)$$

$$A(a) \quad \mapsto \quad A(a)$$

$$R(a, b) \quad \mapsto \quad R(a, b)$$

Remark 1. Deciding CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \quad \mapsto \quad \bigwedge_{i=1}^{n} A_i(x) \rightarrow B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \quad \mapsto \quad A(x) \rightarrow \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R\,.\,B \quad \mapsto \quad A(y) \wedge R(x,y) \rightarrow B(x)$$

$$A \sqsubseteq \exists R\,.\,B \quad \mapsto \quad A(x) \rightarrow \exists y\,.\,R(x,y) \wedge B(y)$$

$$R \sqsubseteq S \quad \mapsto \quad R(x,y) \rightarrow S(x,y)$$

$$R \circ S \sqsubseteq V \quad \mapsto \quad R(x,y) \wedge S(y,z) \rightarrow S(x,z)$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \quad \mapsto \quad \bigwedge_{i=1}^{n} R_i(x,y) \rightarrow S(x,y)$$

$$A(a) \quad \mapsto \quad A(a)$$

$$R(a,b) \quad \mapsto \quad R(a,b)$$

Remark 1. Deciding CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Remark 2.
1. SRI rules feature at most 3 variables.
2. Every function symbol in the skolemisation of a SRI ontology is of arity one.

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \quad \mapsto \quad \bigwedge_{i=1}^{n} A_i(x) \rightarrow B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \quad \mapsto \quad A(x) \rightarrow \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R \,.\, B \quad \mapsto \quad A(y) \wedge R(x,y) \rightarrow B(x)$$

$$A \sqsubseteq \exists R \,.\, B \quad \mapsto \quad A(x) \rightarrow \exists y \,.\, R(x,y) \wedge B(y)$$

$$R \sqsubseteq S \quad \mapsto \quad R(x,y) \rightarrow S(x,y)$$

$$\boxed{R \circ S \sqsubseteq V \quad \mapsto \quad R(x,y) \wedge S(y,z) \rightarrow S(x,z)}$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \quad \mapsto \quad \bigwedge_{i=1}^{n} R_i(x,y) \rightarrow S(x,y)$$

$$A(a) \quad \mapsto \quad A(a)$$

$$R(a,b) \quad \mapsto \quad R(a,b)$$

Remark 1. Deciding CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Remark 2.
1. SRI rules feature at most 3 variables.
2. Every function symbol in the skolemisation of a SRI ontology is of arity one.

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \mapsto \bigwedge_{i=1}^{n} A_i(x) \rightarrow B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \mapsto A(x) \rightarrow \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R \,.\, B \mapsto A(y) \wedge R(x,y) \rightarrow B(x)$$

$$A \sqsubseteq \exists R \,.\, B \mapsto A(x) \rightarrow \exists y \,.\, R(x,y) \wedge B(y)$$

$$R \sqsubseteq S \mapsto R(x,y) \rightarrow S(x,y)$$

$$R \circ S \sqsubseteq V \mapsto R(x,y) \wedge S(y,z) \rightarrow S(x,z)$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \mapsto \bigwedge_{i=1}^{n} R_i(x,y) \rightarrow S(x,y)$$

$$A(a) \mapsto A(a)$$

$$R(a,b) \mapsto R(a,b)$$

Remark 1. *Deciding* CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Remark 2.
1. SRI rules feature at most 3 variables.
2. Every function symbol in the skolemisation of a SRI ontology is of arity one.

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \mapsto \bigwedge_{i=1}^{n} A_i(x) \to B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \mapsto A(x) \to \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R . B \mapsto A(y) \land R(x,y) \to B(x)$$

$$\boxed{A \sqsubseteq \exists R . B \mapsto A(x) \to \exists y . R(x,y) \land B(y)}$$

$$R \sqsubseteq S \mapsto R(x,y) \to S(x,y)$$

$$R \circ S \sqsubseteq V \mapsto R(x,y) \land S(y,z) \to S(x,z)$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \mapsto \bigwedge_{i=1}^{n} R_i(x,y) \to S(x,y)$$

$$A(a) \mapsto A(a)$$

$$R(a,b) \mapsto R(a,b)$$

Remark 1. Deciding CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Remark 2.
1. SRI rules feature at most 3 variables.
2. Every function symbol in the skolemisation of a SRI ontology is of arity one.

Real-world coverage: SRI Ontologies

$$A_1 \sqcap \ldots \sqcap A_n \sqsubseteq B \mapsto \bigwedge_{i=1}^{n} A_i(x) \to B(x)$$

$$A \sqsubseteq B_1 \sqcup \ldots \sqcup B_n \mapsto A(x) \to \bigvee_{i=1}^{n} B_i(x)$$

$$A \sqsubseteq \forall R . B \mapsto A(y) \wedge R(x,y) \to B(x)$$

$$A \sqsubseteq \exists R . B \mapsto A(x) \to \exists y . R(x,y) \wedge B(y)$$

$$R \sqsubseteq S \mapsto R(x,y) \to S(x,y)$$

$$R \circ S \sqsubseteq V \mapsto R(x,y) \wedge S(y,z) \to S(x,z)$$

$$R_1 \sqcap \ldots \sqcap R_n \sqsubseteq S \mapsto \bigwedge_{i=1}^{n} R_i(x,y) \to S(x,y)$$

$$A(a) \mapsto A(a)$$

$$R(a,b) \mapsto R(a,b)$$

Remark 1. Deciding CQ entailment for SRI ontologies is 2ExpTime-Hard and in 3ExpTime.

Remark 2.
1. SRI rules feature at most 3 variables.
2. Every function symbol in the skolemisation of a SRI ontology is of arity one.

SRI Axioms

Remark 2.
1. *Every rule in an SRI ontology has at most 3 variables.*
2. Every function symbol in the skolemisation of a SRI ontology has arity one

SRI Axioms

Remark 2.
1. *Every rule in an SRI ontology has at most 3 variables.*
2. Every function symbol in the skolemisation of a SRI ontology has arity one

Corollary. To guarantee that tractable CQ entailment over a SRI ontology is possible we only need to verify the following:
1. Acyclicity.
2. Braid length in the dependency graph is bounded.

Evaluation Results

Evaluation Results

Acyclicity

	MOWL Corpus	Oxford Ontology Repo
Ontologies	1576	225
Acyclic	974 (61.8%)	170 (75.6%)

Evaluation Results

Acyclicity

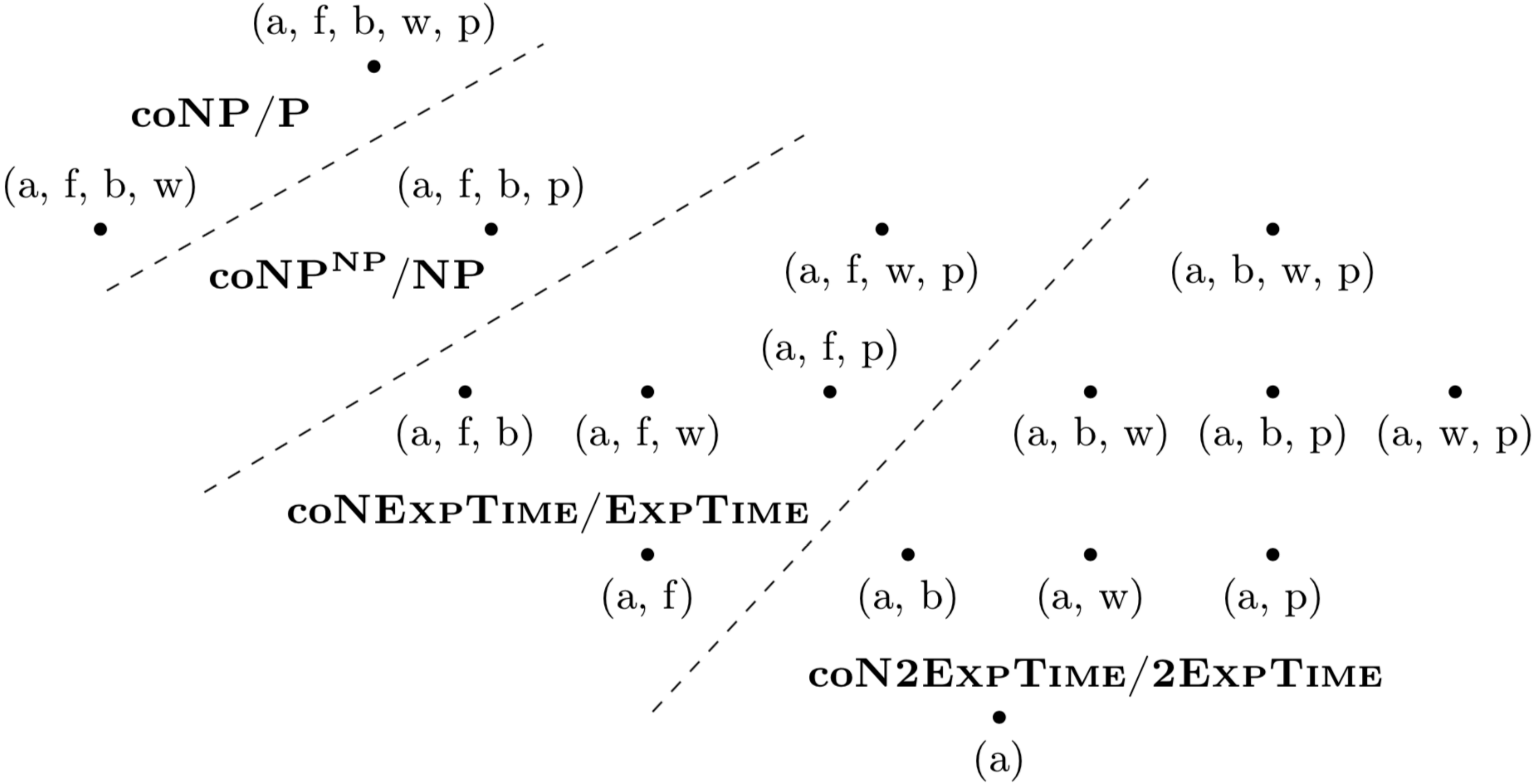
	MOWL Corpus	Oxford Ontology Repo
Ontologies	1576	225
Acyclic	974 (61.8%)	170 (75.6%)

Braid Length

MOWL Corpus + Oxford

(max. length of a braid)	1	2	3	4	5	6	11	22	23	25	Total
(count)	851	153	56	61	11	1	1	2	7	1	1144
	74	88	93	98	99	99	99	99.1	99.3	99.9	100

More Results!



VLog

VLog

Efficient Model Construction for Horn Logic with VLog — System Description

Jacopo Urbani, Markus Krötzsch, Ceriel J. H. Jacobs,
Irina Dragoste, and David Carral

[IJCAR 2018]

An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.

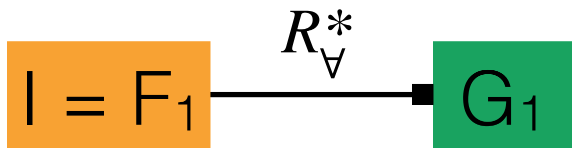
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.

$I = F_1$

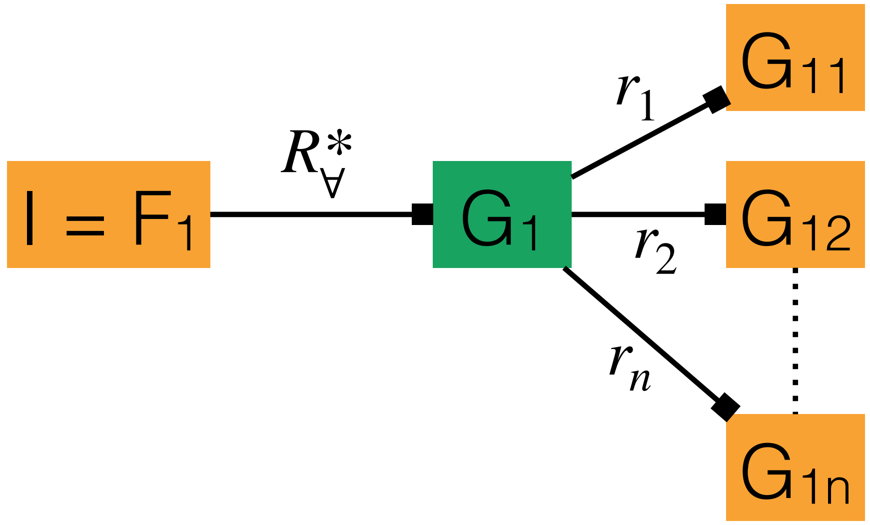
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.



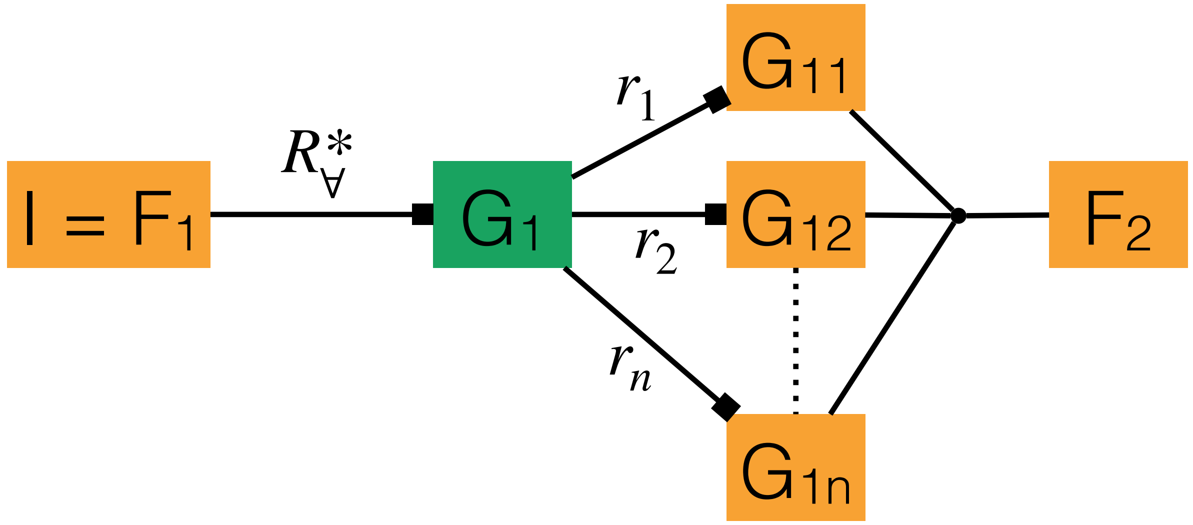
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.



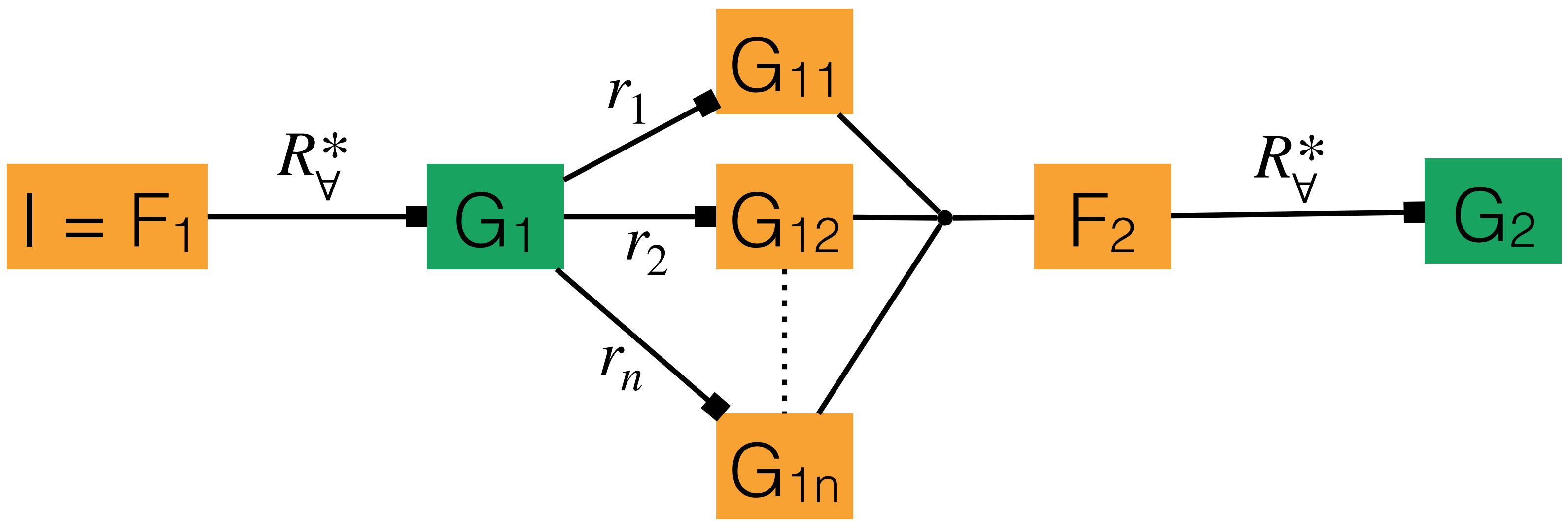
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.



An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.

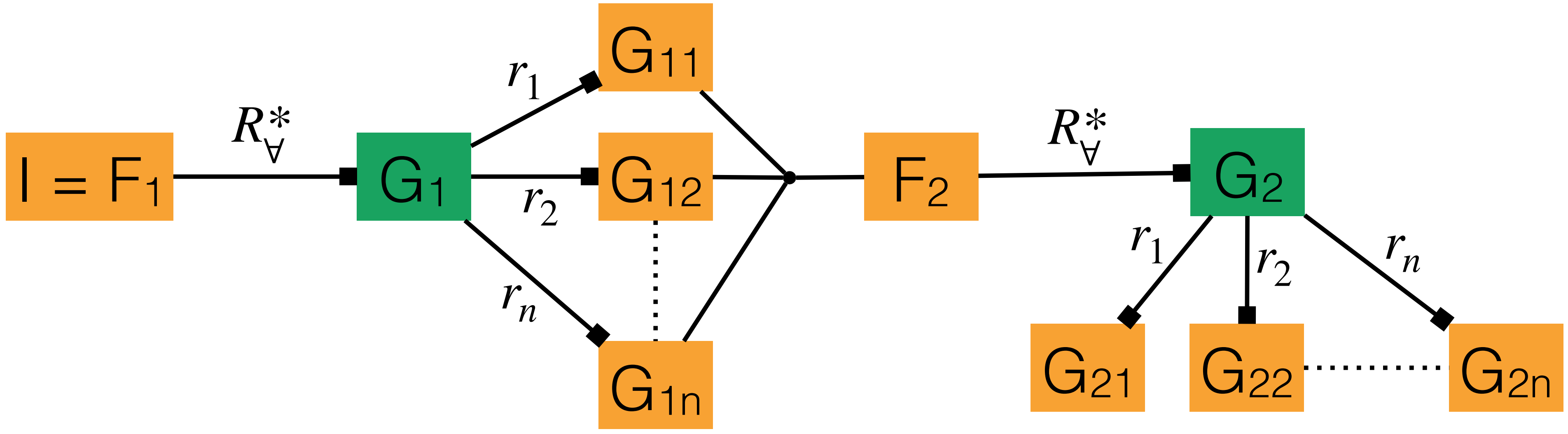


An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.

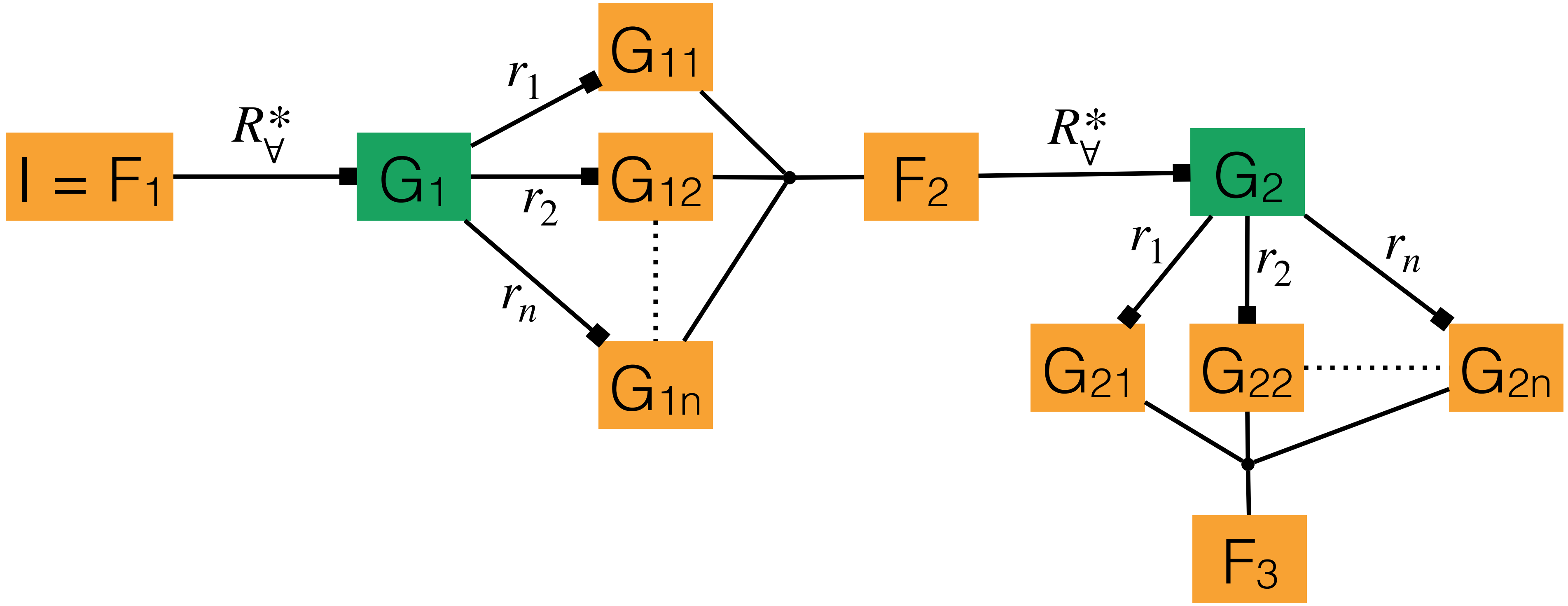
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.

The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.



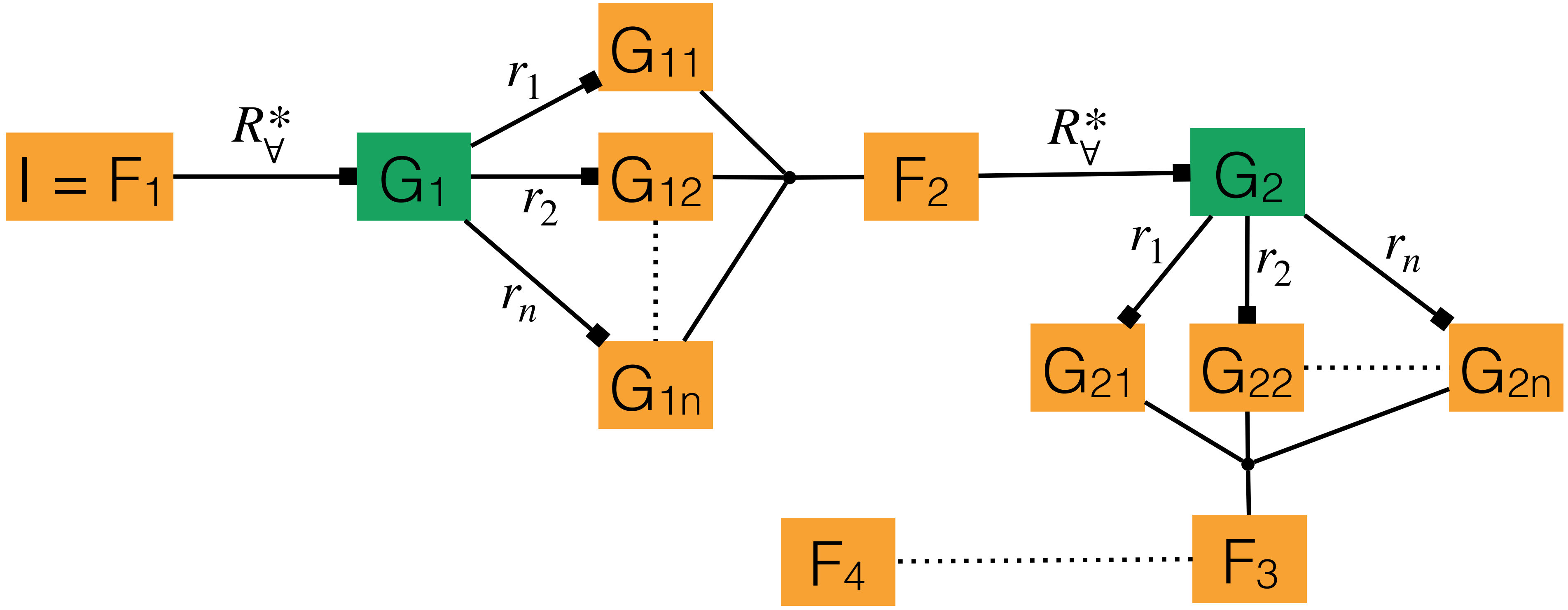
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with Ch(R, I), is computed as follows.



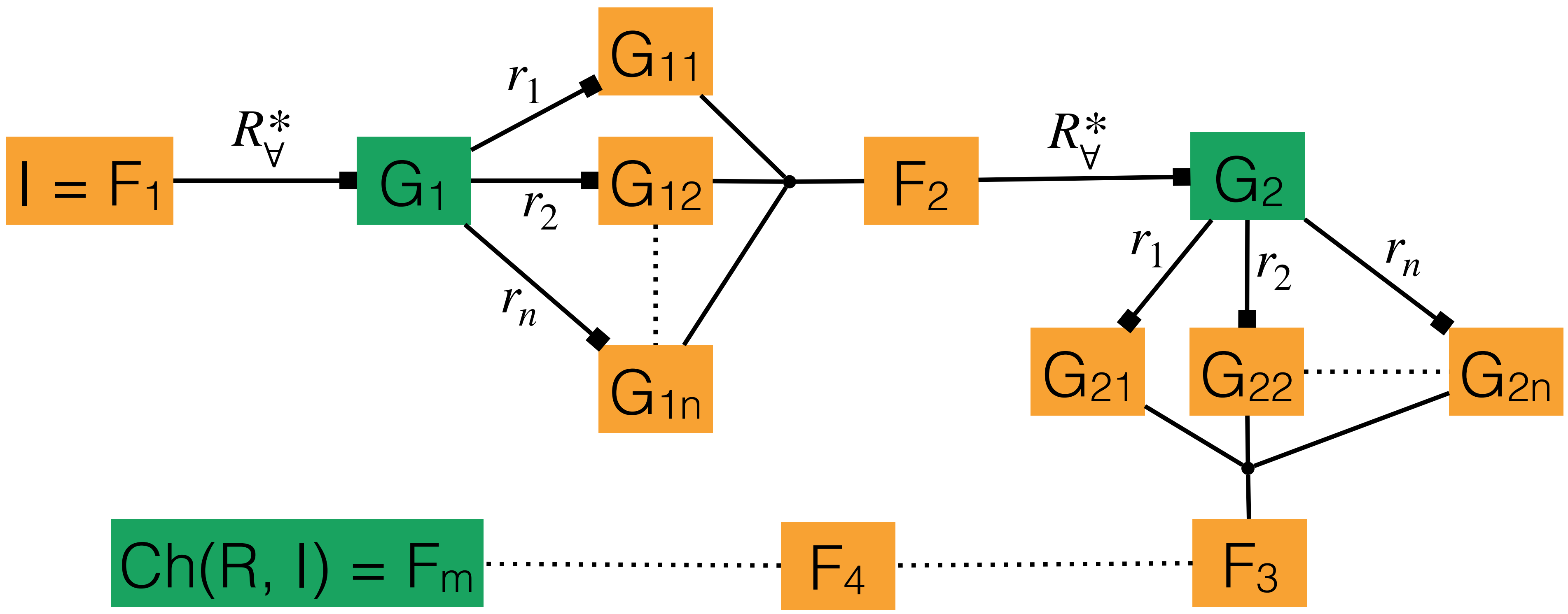
An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with $\mathsf{Ch}(R, I)$, is computed as follows.



An Implementation for the DF Restricted Chase

Consider a rule set R and an instance I.
Let $R_\forall$ and $R_\exists = \{r_1, \ldots, r_n\}$ be the sets of all Datalog and non-Datalog rules in R, respectively.
The Datalog-first restricted chase of R and I, denoted with Ch(R, I), is computed as follows.



Performance: VLog vs RDFox

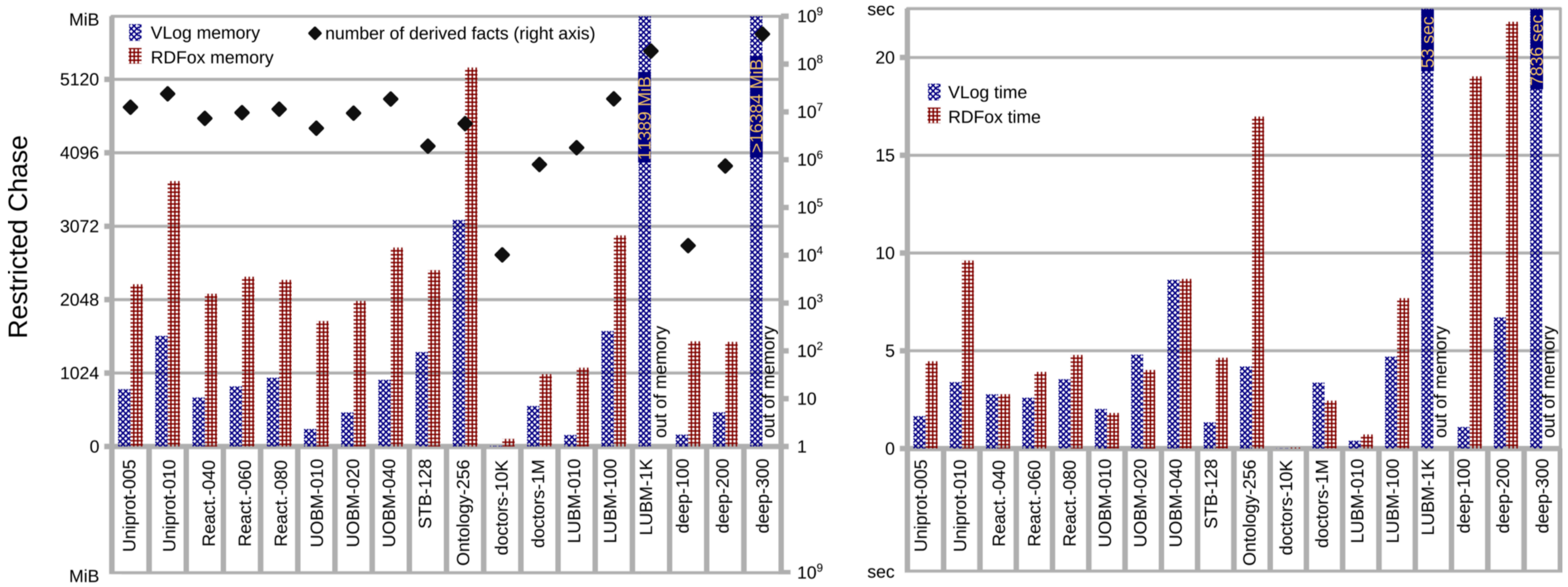


Fig. 1. Memory usage (left) and materialisation time (right) for VLog and RDFox

Conclusions

Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!

Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets

Problem Solved?

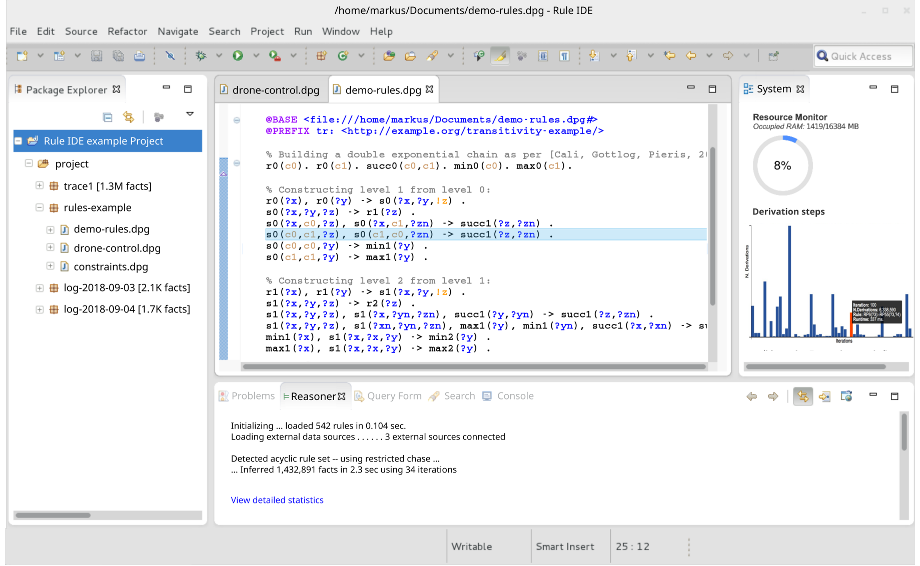
This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in "repairing" non-acyclic rule sets

Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

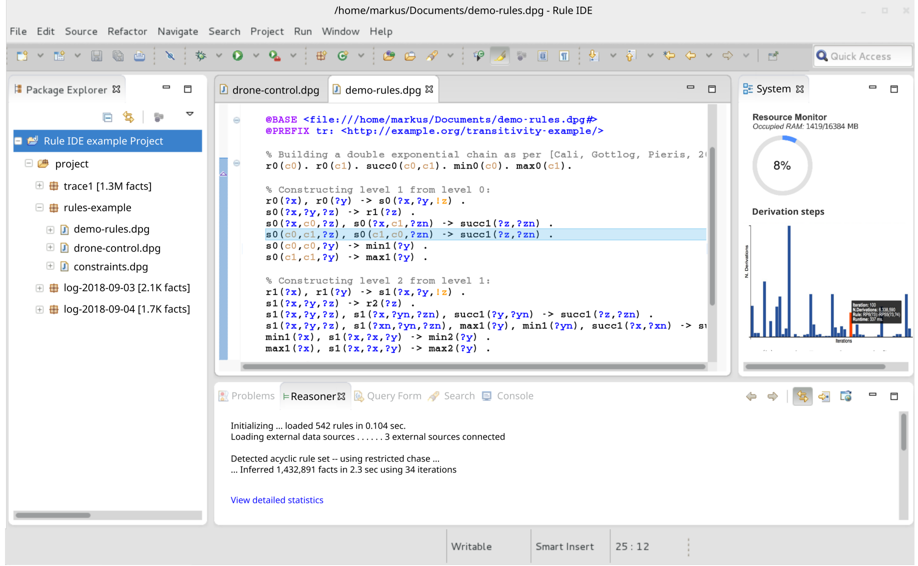
* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets

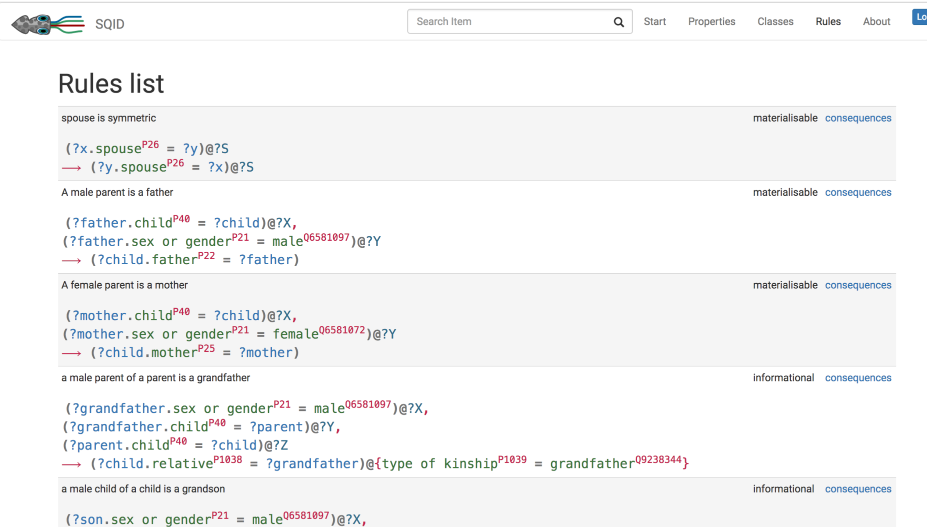


Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets

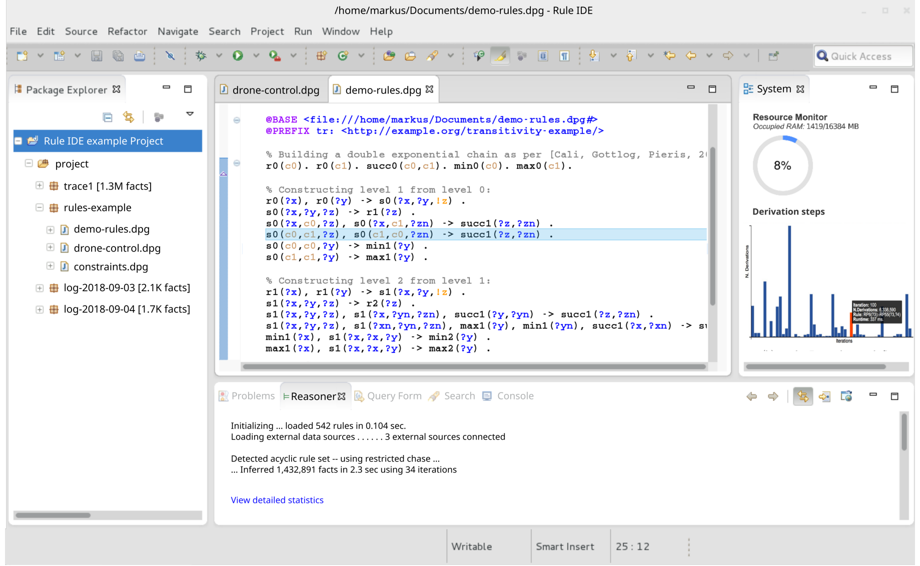


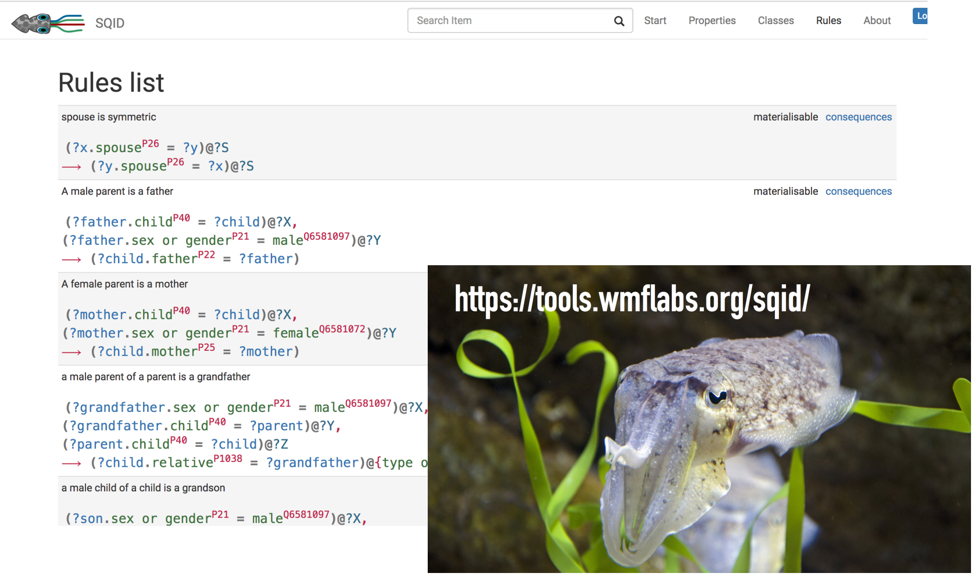


Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets

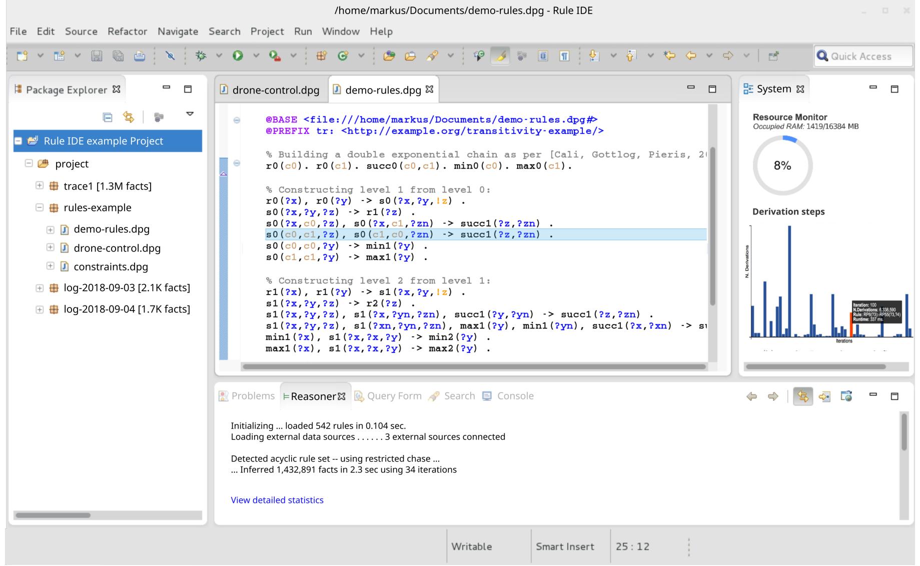


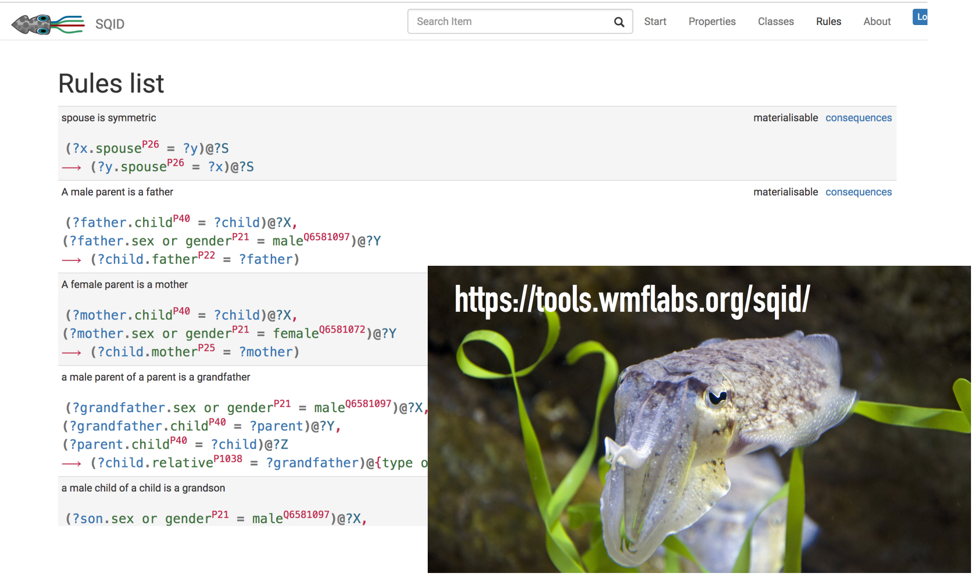


Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets
* Efficient implementation for acyclicity checks

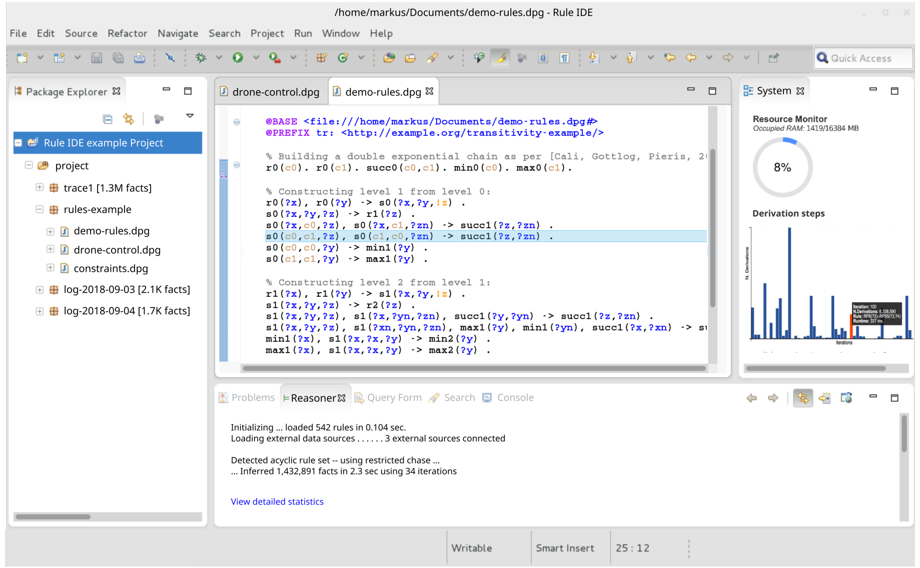


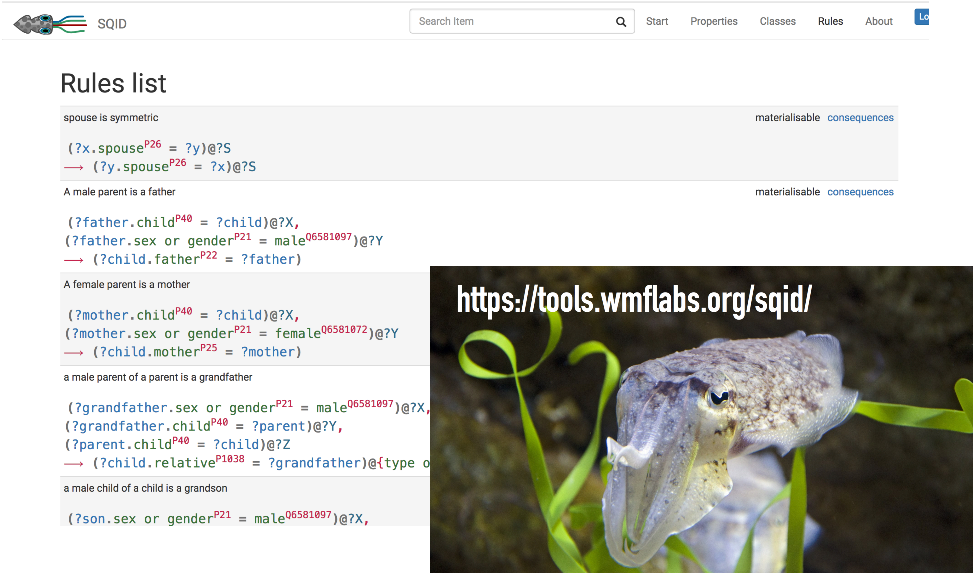


Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets
* Efficient implementation for acyclicity checks
* Optimise chase reasoners

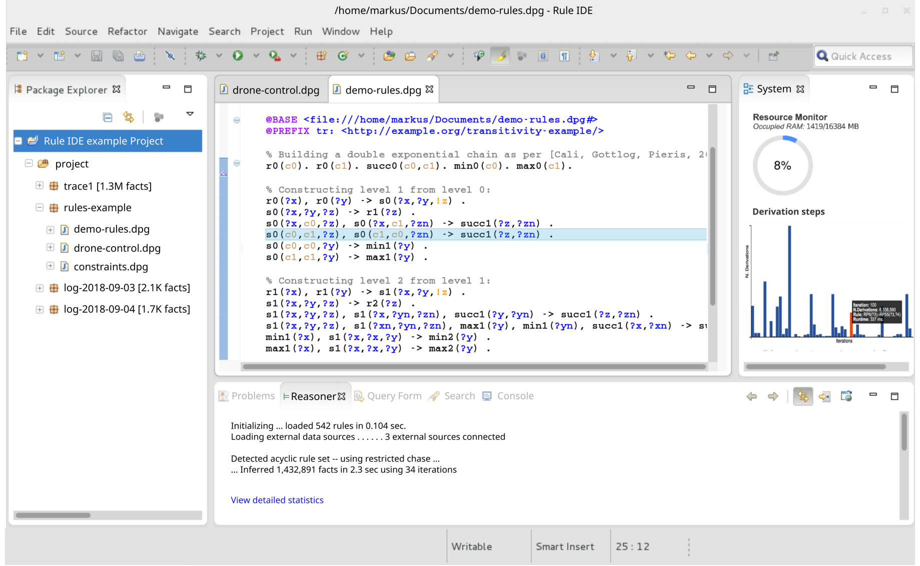


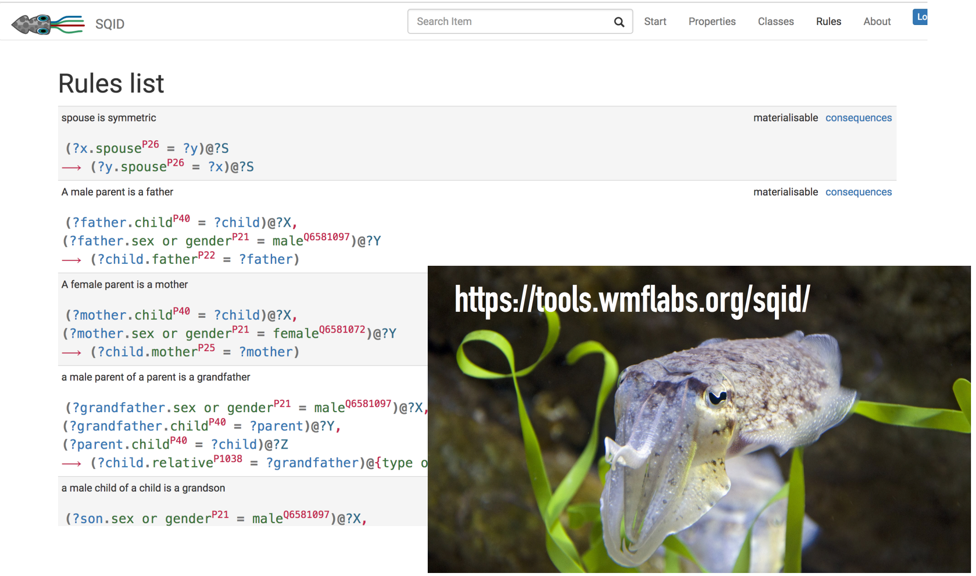


Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets
* Efficient implementation for acyclicity checks
* Optimise chase reasoners
* The disjunctive case

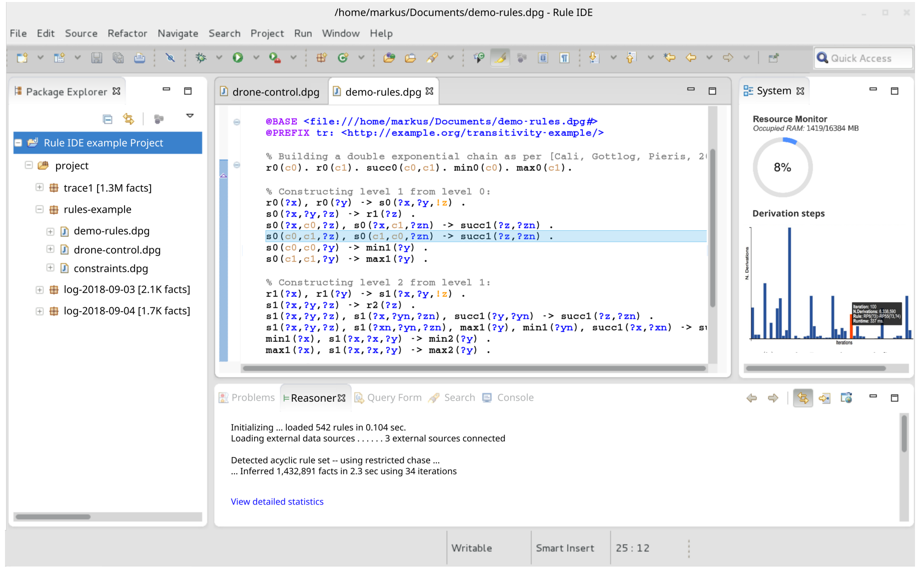


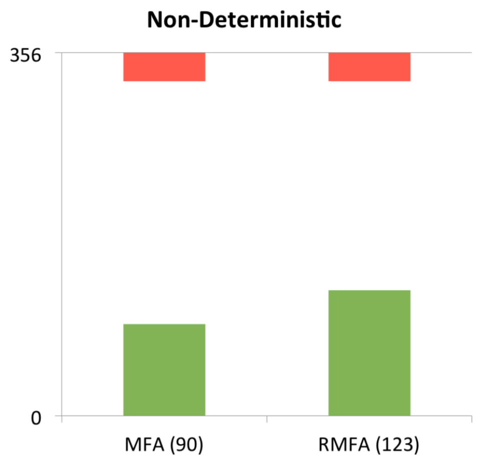


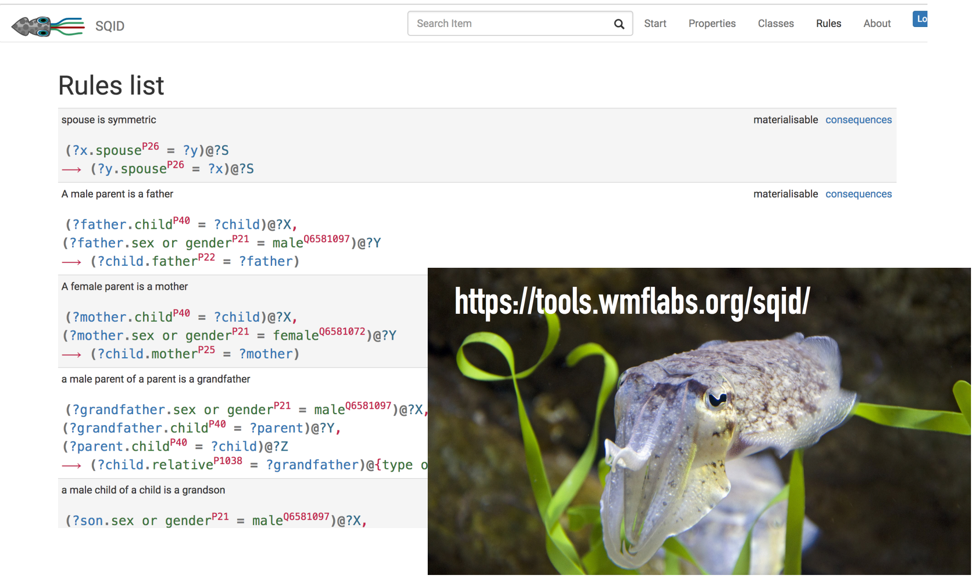
Problem Solved?

This is it, everybody should use existential rules + acyclicity notions!
Surely not, a lot of work still needs to be done:

* Experiments with non-DL existential rule sets
* Develop tools that can assist knowledge engineers in “repairing” non-acyclic rule sets
* Efficient implementation for acyclicity checks
* Optimise chase reasoners
* The disjunctive case







Bibliography

Reasoning over Existential Rules with Acyclicity Notions and the Datalog-first Restricted Chase

David Carral

TECHNISCHE UNIVERSITÄT DRESDEN

Slides available at https://iccl.inf.tu-dresden.de/web/Existential-rules-acyclicity

Bibliography: Sections

* **First section**
  Restricted Chase (Non)Termination for Existential Rules with Disjunctions [IJCAI 2017]
  https://iccl.inf.tu-dresden.de/web/Inproceedings3140/en

* **Second section**
  Tractable Query Answering for Expressive Ontologies and Rules [ISWC 2017]
  https://iccl.inf.tu-dresden.de/web/Inproceedings3163/en

* **Third section**
  Efficient Model Construction for Horn Logic with VLog [IJCAR 2018]
  https://iccl.inf.tu-dresden.de/web/Article3046/en

Bibliography: Rule Engines

* **VLog**
  Efficient Model Construction for Horn Logic with VLog [IJCAR 2018]
  https://iccl.inf.tu-dresden.de/web/Article3046/en
  Column-Oriented Datalog Materialization for Large Knowledge Graphs [AAAI 2016]
  http://korrekt.org/papers/Urbani-Jacobs-Kroetzsch_Vlog-datalog-materialization-AAAI2016.pdf

* **RDFox**
  Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems [AAAI 2014]
  http://www.cs.ox.ac.uk/ian.horrocks/Publications/download/2014/MNPHO14a.pdf

Bibliography: Acyclicity Notions

* **Restricted Model-Faithful Acyclicity (RMFA)**
  Restricted Chase (Non)Termination for Existential Rules with Disjunctions. [IJCAI 2017]
  https://iccl.inf.tu-dresden.de/web/Inproceedings3140/en

* **Model-Faithful Acyclicity (MFA)**
  Acyclicity Notions for Existential Rules and Their Application to QA in Ontologies [J. Artif. Intell. Res. 47]
  https://iccl.inf.tu-dresden.de/web/Article4005/en

* **Joint Acyclicity (JA)**
  Extending Decidable Existential Rules by Joining Acyclicity and Guardedness [IJACI 2011]
  https://iccl.inf.tu-dresden.de/web/Inproceedings3149/en

* **Weak Acyclicity (WA)**
  Data Exchange: Semantics and Query Answering [Theor. Comput. Sci. 336]
  https://www.sciencedirect.com/science/article/pii/S030439750400725X