Combining Decidability Paradigms for Existential Rules

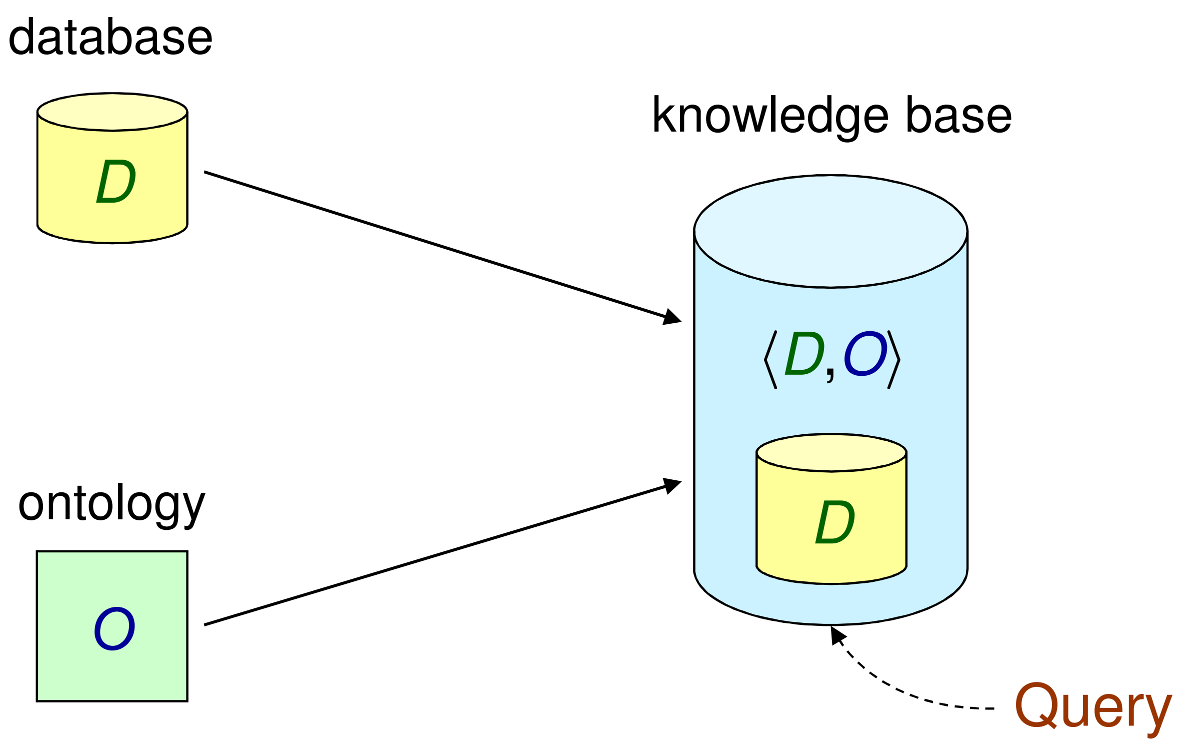
Georg Gottlob[1] Marco Manna[2] Andreas Pieris[1]

[1]Department of Computer Science, University of Oxford, UK

[2]Department of Mathematics, University of Calabria, Italy

ICLP 2013, Istanbul, August 26

Ontological Query Answering



$$\langle D, O \rangle \models \text{Query} \quad \Leftrightarrow \quad D \wedge O \models \text{Query}$$

Ontological Query Answering: Example

D = $\mathit{person}(\text{john})$

O = $\forall P\, \mathit{person}(P) \rightarrow \exists F\, \mathit{father}(F,P)$
$\forall F \forall P\, \mathit{father}(F,P) \rightarrow \mathit{person}(P)$

$\forall M \models \langle D,O \rangle$ = $\ldots\ \mathit{father}(z,\text{john})\ \ \mathit{person}(z)\ \ldots$

Ontological Query Answering: Example

D = *person*(john)

O =

$$\forall P\ person(P) \rightarrow \exists F\ father(F,P)$$
$$\forall F \forall P\ father(F,P) \rightarrow person(P)$$

$$\forall M \models \langle D, O \rangle \quad = \quad \ldots\ father(z,\text{john}) \quad person(z)\ \ldots$$

$$\exists x\ father(x,\text{john}) \wedge person(x)$$

✓

Ontological Query Answering: Example

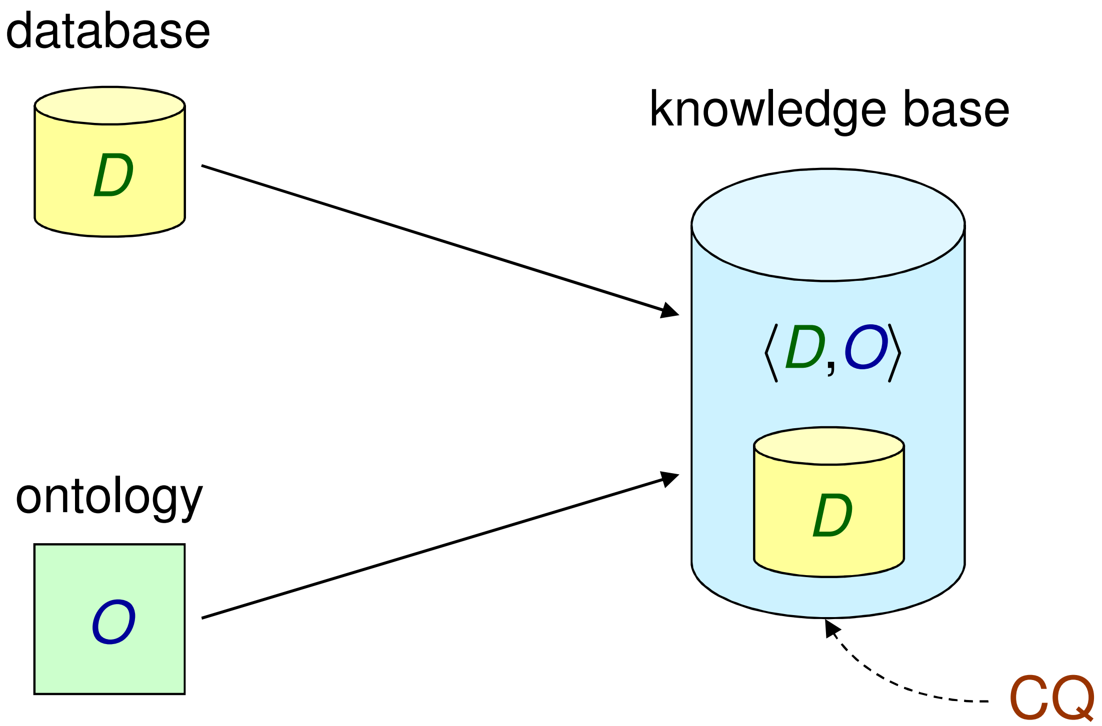
$$D = \textit{person}(\text{john}) \qquad O = \begin{array}{l} \forall P\ \textit{person}(P) \rightarrow \exists F\ \textit{father}(F,P) \\ \forall F \forall P\ \textit{father}(F,P) \rightarrow \textit{person}(P) \end{array}$$

$$\forall M \models \langle D, O \rangle \quad = \quad \ldots\ \textit{father}(z,\text{john})\ \ \textit{person}(z)\ \ldots$$

$\exists x\ \textit{father}(x,\text{john}) \wedge \textit{person}(x)$ ✓

$\exists x\ \textit{father}(\text{john},x)$ ✗

Ontology and Query Language



existential rules

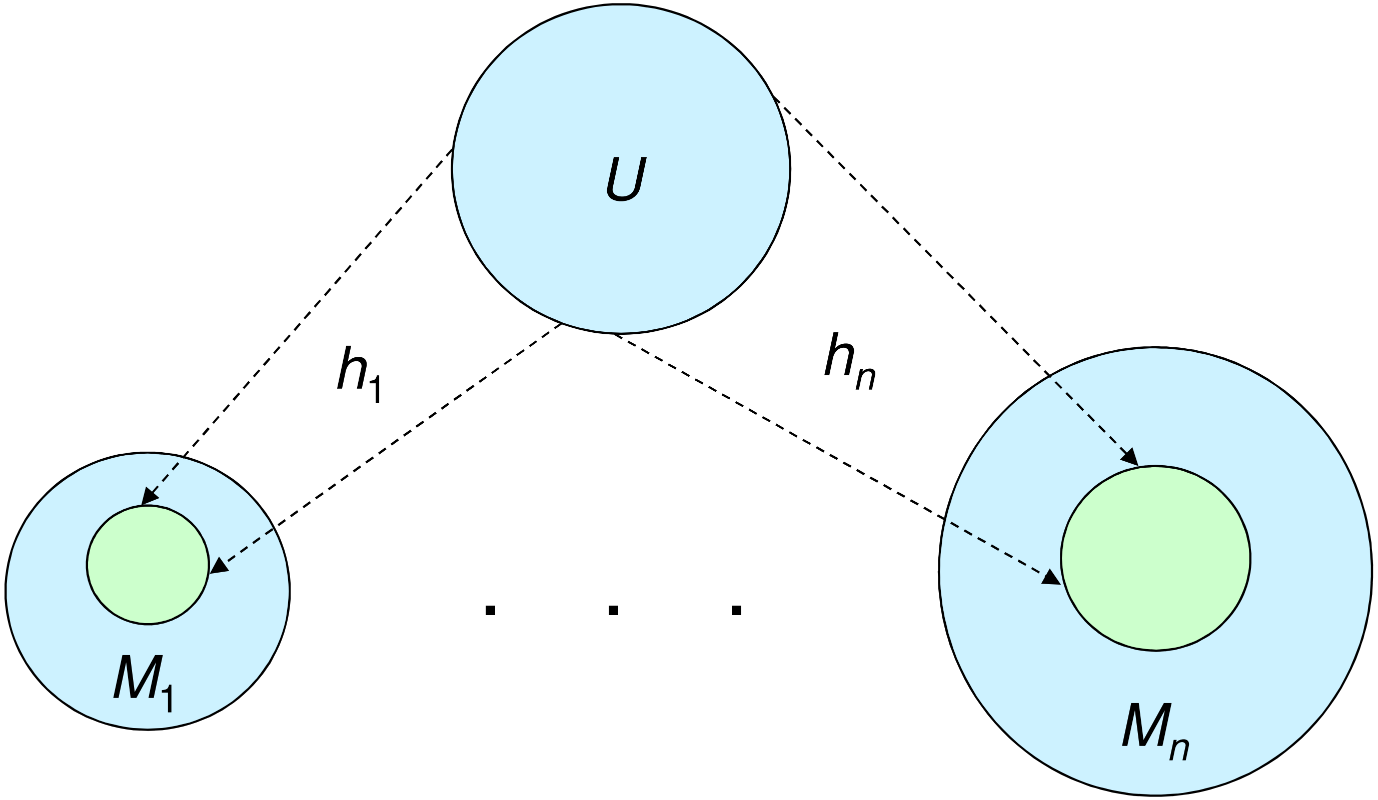
conjunctive queries

$\forall \mathbf{X}\ \varphi(\mathbf{X}) \quad \rightarrow \quad \exists \mathbf{Y}\ \psi(\mathbf{X},\mathbf{Y})$

$\exists \mathbf{X}\ \varphi(\mathbf{X})$

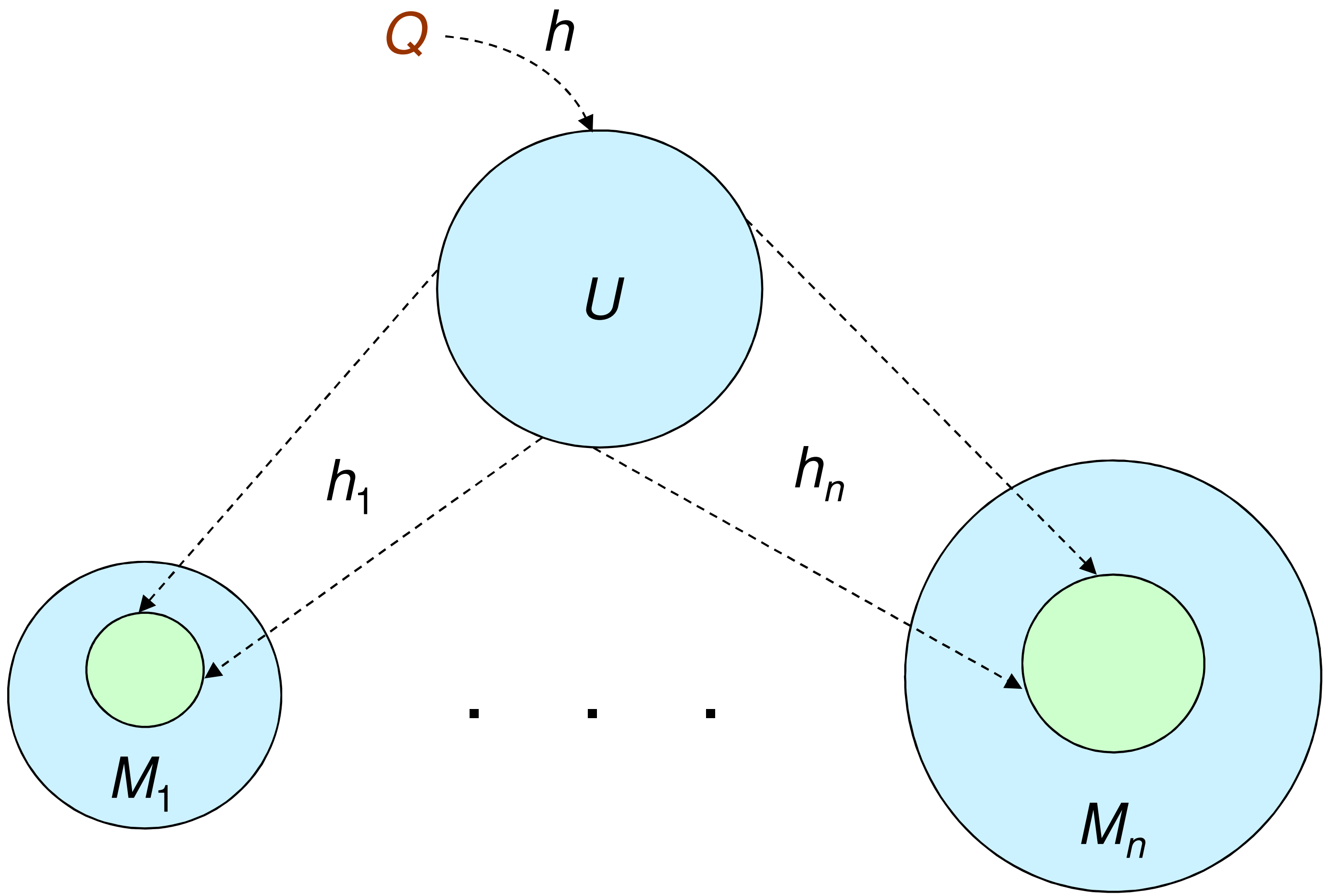
Existential Rules: The Key Property

Existence of a universal model of $\langle D,O \rangle$



$$\forall M \,(M \models \langle D,O \rangle \quad \Rightarrow \quad U \xrightarrow{\text{hom}} M)$$

Existential Rules: The Key Property



$$\langle D, O \rangle \models Q \quad \Leftrightarrow \quad U \models Q$$

The Chase Procedure

Input: Database D, ontology O

Output: A universal model of $\langle D,O \rangle$

D = $\boxed{person(\text{john})}$

O = $\boxed{person(P) \rightarrow \exists F\ father(F,P) \qquad father(F,P) \rightarrow person(F)}$

$chase(D,O) = D \cup$?

The Chase Procedure

Input: Database D, ontology O

Output: A universal model of $\langle D, O \rangle$

$$D = \textit{person}(\text{john})$$

$$O = \textit{person}(P) \rightarrow \exists F\, \textit{father}(F,P) \qquad \textit{father}(F,P) \rightarrow \textit{person}(F)$$

$$\textit{chase}(D,O) = D \cup \{\textit{father}(z_1,\text{john})$$

The Chase Procedure

Input: Database D, ontology O

Output: A universal model of $\langle D, O \rangle$

$$D = \textit{person}(\text{john})$$

$$O = \textit{person}(P) \rightarrow \exists F\ \textit{father}(F,P) \qquad \textit{father}(F,P) \rightarrow \textit{person}(F)$$

$$\textit{chase}(D,O) = D \cup \{\textit{father}(z_1,\text{john}), \textit{person}(z_1)$$

The Chase Procedure

Input: Database D, ontology O

Output: A universal model of $\langle D,O \rangle$

$$D = \boxed{person(\text{john})}$$

$$O = \boxed{person(P) \rightarrow \exists F\, father(F,P) \qquad father(F,P) \rightarrow person(F)}$$

$$chase(D,O) = D \cup \{father(z_1,\text{john}),\ person(z_1),\ father(z_2,z_1)$$

The Chase Procedure

Input: Database D, ontology O

Output: A universal model of $\langle D,O \rangle$

D = *person*(john)

O = $person(P) \rightarrow \exists F\ father(F,P) \qquad father(F,P) \rightarrow person(F)$

$chase(D,O) = D \cup \{father(z_1,\text{john}),\ person(z_1),\ father(z_2,z_1),\ \ldots$

least fixpoint

Chase: The Challenge of Infinity

- In general is infinite

$$D = \{p(\mathrm{a},\mathrm{b})\} \qquad\qquad p(X,Y) \rightarrow \exists Z\, p(Y,Z)$$

$$\text{Solution} = \{p(\mathrm{a},\mathrm{b}),p(\mathrm{b},z_1),p(z_1,z_2),p(z_2,z_3),\ldots\}$$

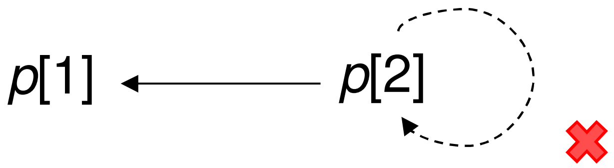
- Query answering under existential rules is undecidable
  implicit in [Beeri & Vardi, ICALP 1981]

… syntactic restrictions are needed!

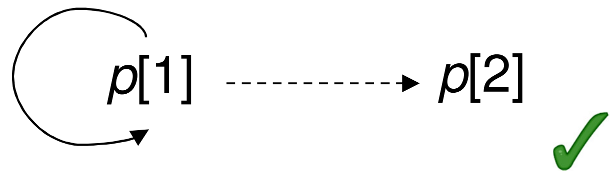
Weak-acyclicity

- Graph-based definition - dependency graph

$p(X,Y) \rightarrow \exists Z\, p(Y,Z)$



$p(X,Y) \rightarrow \exists Z\, p(X,Z)$



- Guarantees termination of the chase $\Rightarrow$ query answering is decidable

[Fagin, Kolaitis, Miller & Popa, Theoretical Computer Science 2005]

Guardedness

- There exists a body-atom that contains all the body-variables

$$\mathit{employee}(X),\ \mathit{supervisorOf}(X,Y),\ \mathit{manager}(Y)\ \rightarrow\ \mathit{manager}(X)$$

guard

- Chase has finite treewidth $\Rightarrow$ query answering is decidable

[Calì, Gottlob & Kifer, KR 2008]

Stickiness

- Join-variables stick to the inferred atoms

$$p(X,Y),p(Y,Z) \rightarrow \exists W\, t(X,Y,W)$$

$$t(X,Y,Z) \rightarrow \exists W\, s(X,W)$$

✗

$$p(X,Y),p(Y,Z) \rightarrow \exists W\, t(X,Y,W)$$

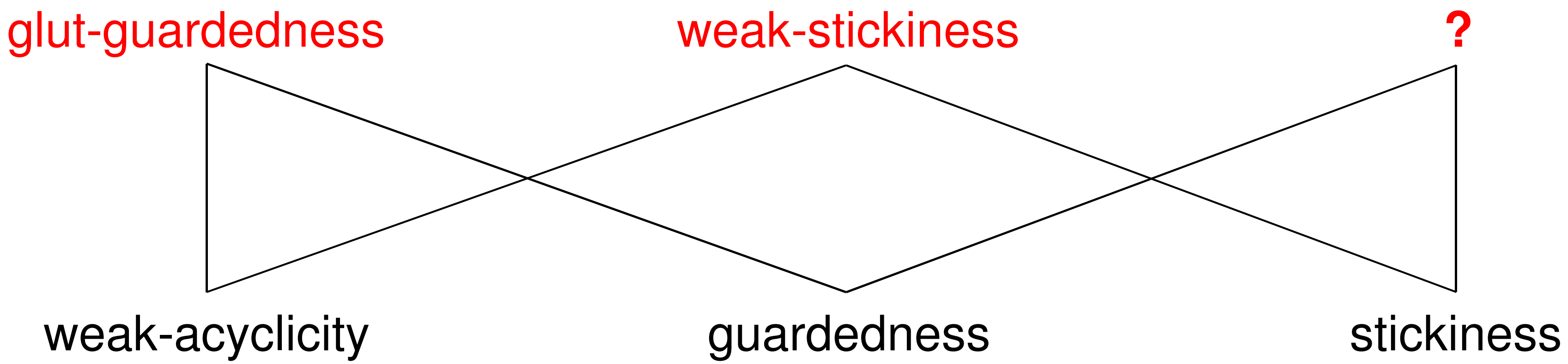
$$t(X,Y,Z) \rightarrow \exists W\, s(Y,W)$$

✓

- Proof-theoretic procedures $\Rightarrow$ query answering is decidable

[Calì, Gottlob & P., Artificial Intelligence 2010]

Combining Decidability Paradigms



- Glut-guardedness - guard only harmful variables
  [Krötzsch & Rudolph, IJCAI 2011]

- Weak-stickiness - only harmful join-variables stick to the inferred atoms
  [Calì, Gottlob & P., Artificial Intelligence 2010]

- Guardedness + stickiness - the subject of this work

Guarded ∪ Sticky (G|S)

$\mathcal{O}_g$ is guarded

$\mathcal{O}$ partition $\Rightarrow$ $\mathcal{O}$ is a G|S ontology

$\mathcal{O}_s$ is sticky

Guarded ∪ Sticky (G|S)

$$\begin{tikzcd}[column sep=large] & O_g \text{ is guarded} \\ O \arrow[ur] \arrow[dr] \arrow[r, phantom, "\text{partition}"] & & \Rightarrow & O \text{ is a G|S ontology} \\ & O_s \text{ is sticky} \end{tikzcd}$$

Theorem: Query answering under G|S is undecidable.

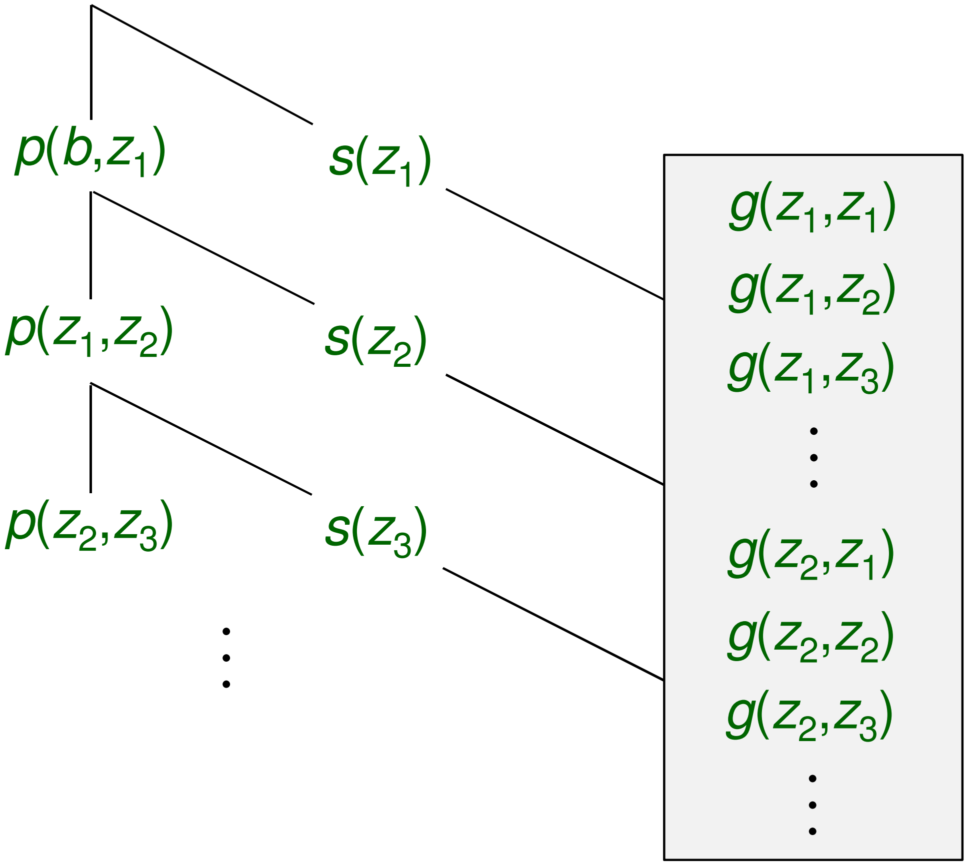
Proof: By reduction from query answering under general existential rules.

Source of Undecidability

Guard predicate may store non-treelike structures

$$O = \left\{ \begin{array}{c} g(X,Y), s(X) \rightarrow s(Y) \\ p(X,Y) \rightarrow \exists z\, p(Y,Z), s(X) \\ s(X), s(Y) \rightarrow g(X,Y) \end{array} \right\}$$

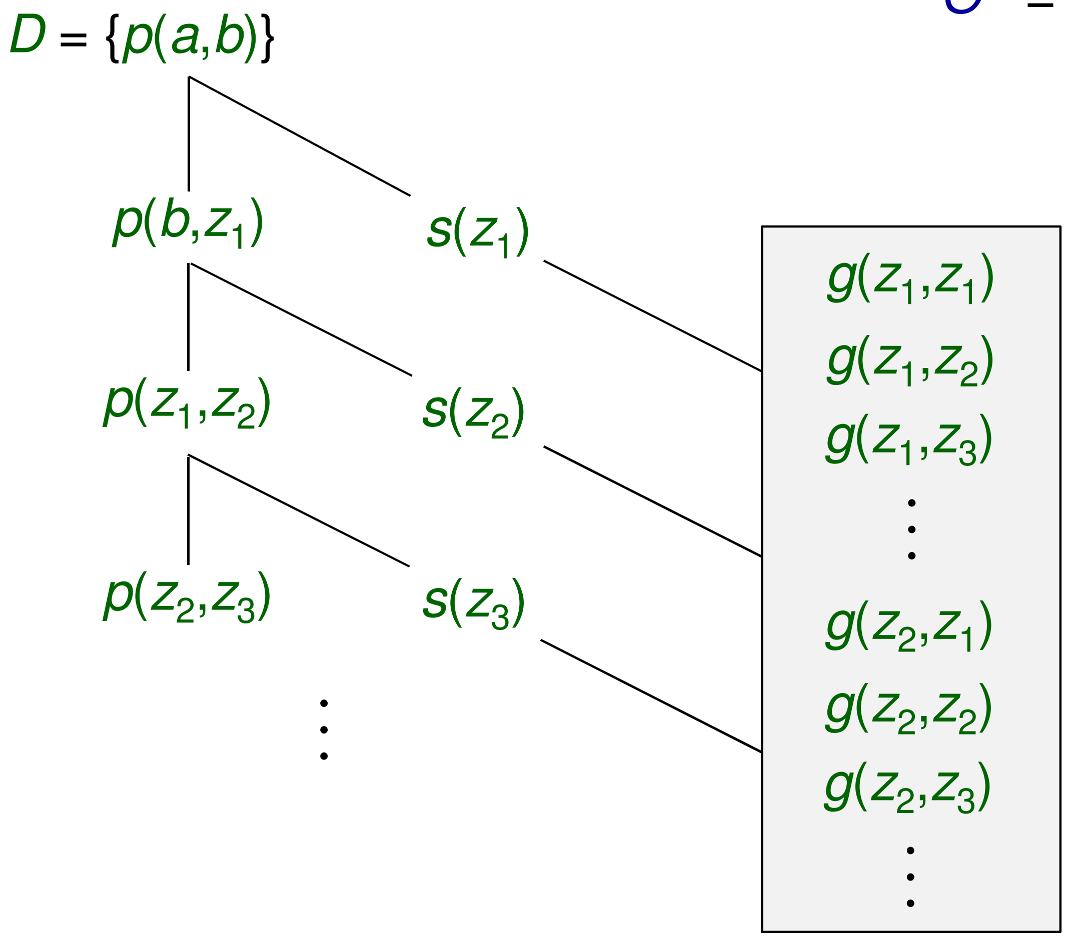
$D = \{p(a,b)\}$



Source of Undecidability

Guard predicate may store non-treelike structures

$$O = \left\{ \begin{array}{c} g(X,Y), s(X) \rightarrow s(Y) \\ p(X,Y) \rightarrow \exists z\, p(Y,Z), s(X) \\ s(X), s(Y) \rightarrow g(X,Y) \end{array} \right\}$$



$$\underbrace{|nullsOf(chase(D,O))|}_{\text{infinite}}\text{-clique}$$

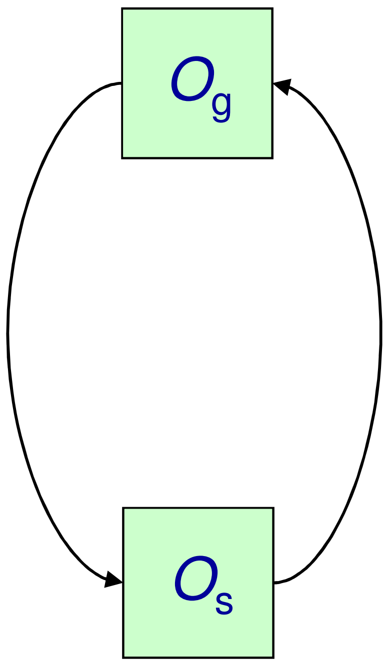
Taming the Interaction

O_g is guarded

tame G|S ontology O — partition

O_s is sticky

unrestricted interaction

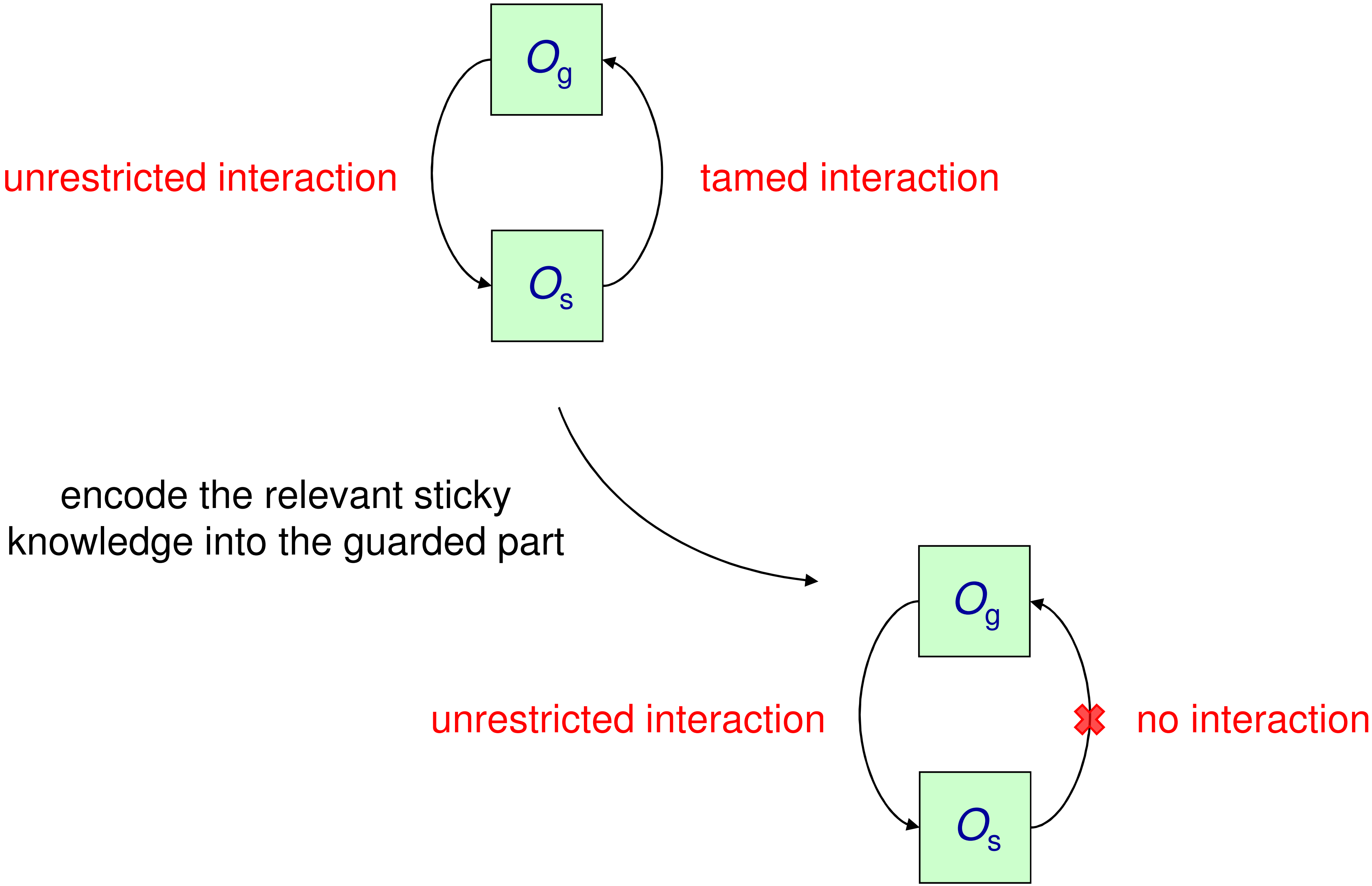


tamed interaction

- do not feed the guard atom
- may feed the non-guard atoms

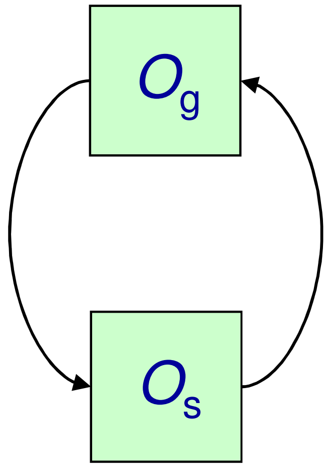
e.g., predicate-tameness

Taming the Interaction: Decidability



Taming the Interaction: Decidability

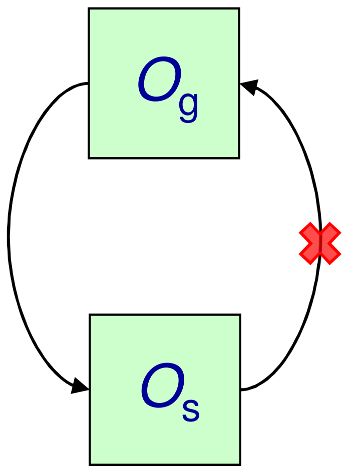
unrestricted interaction



tamed interaction

encode the relevant sticky
knowledge into the guarded part

unrestricted interaction



no interaction

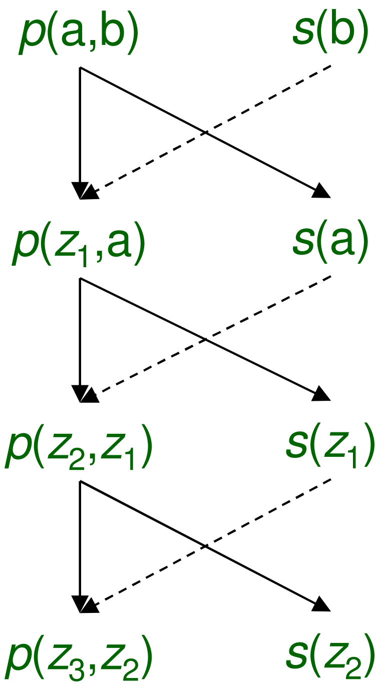
… but what about the complexity?

The Guarded Case

Guarded Chase Forest

$D = \{p(\mathrm{a},\mathrm{b}),\ s(\mathrm{b})\}$

$$O = \left\{ \begin{array}{c} p(X,Y), s(Y) \rightarrow \exists Z\, p(Z,X) \\ p(X,Y) \rightarrow s(X) \end{array} \right\}$$

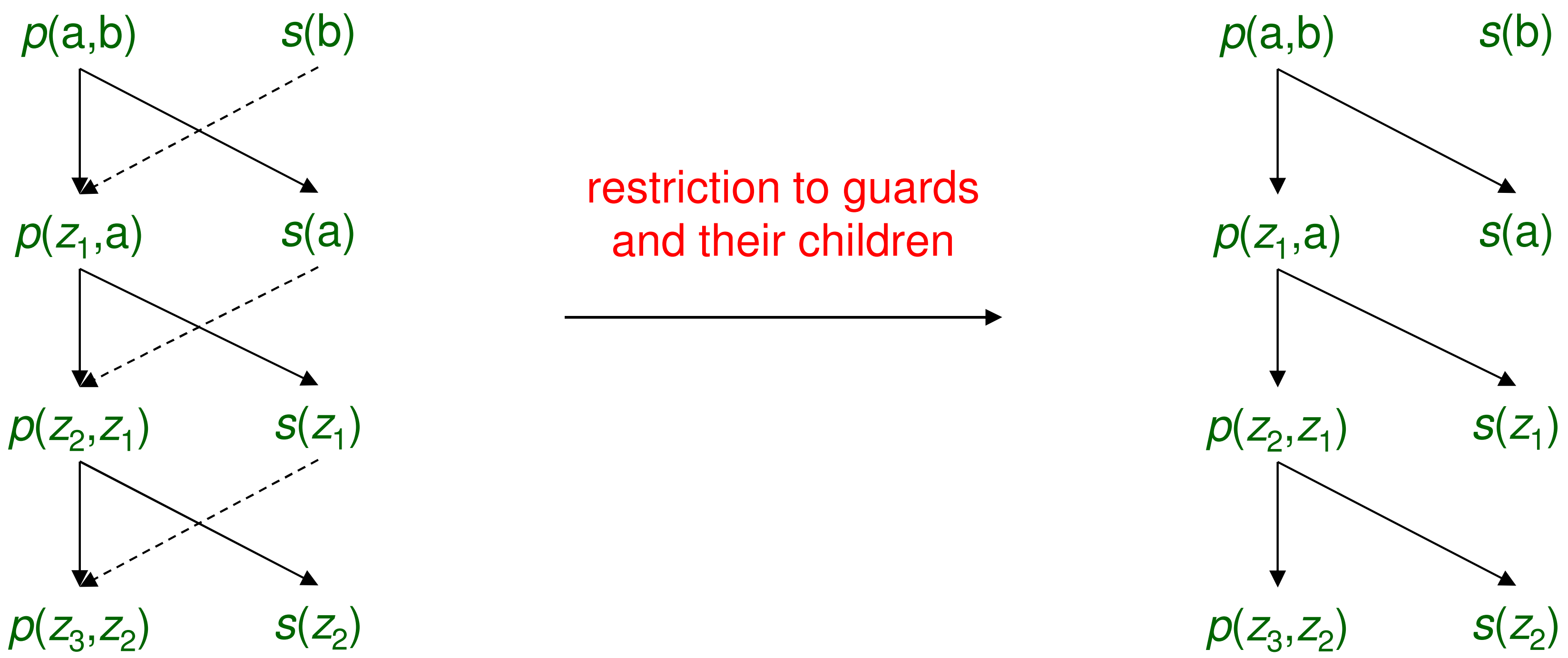


The Guarded Case

Guarded Chase Forest

$D = \{p(\mathrm{a},\mathrm{b}),\ s(\mathrm{b})\}$

$$O = \left\{ \begin{array}{c} p(X,Y), s(Y) \rightarrow \exists Z\, p(Z,X) \\ p(X,Y) \rightarrow s(X) \end{array} \right\}$$



The Guarded Case

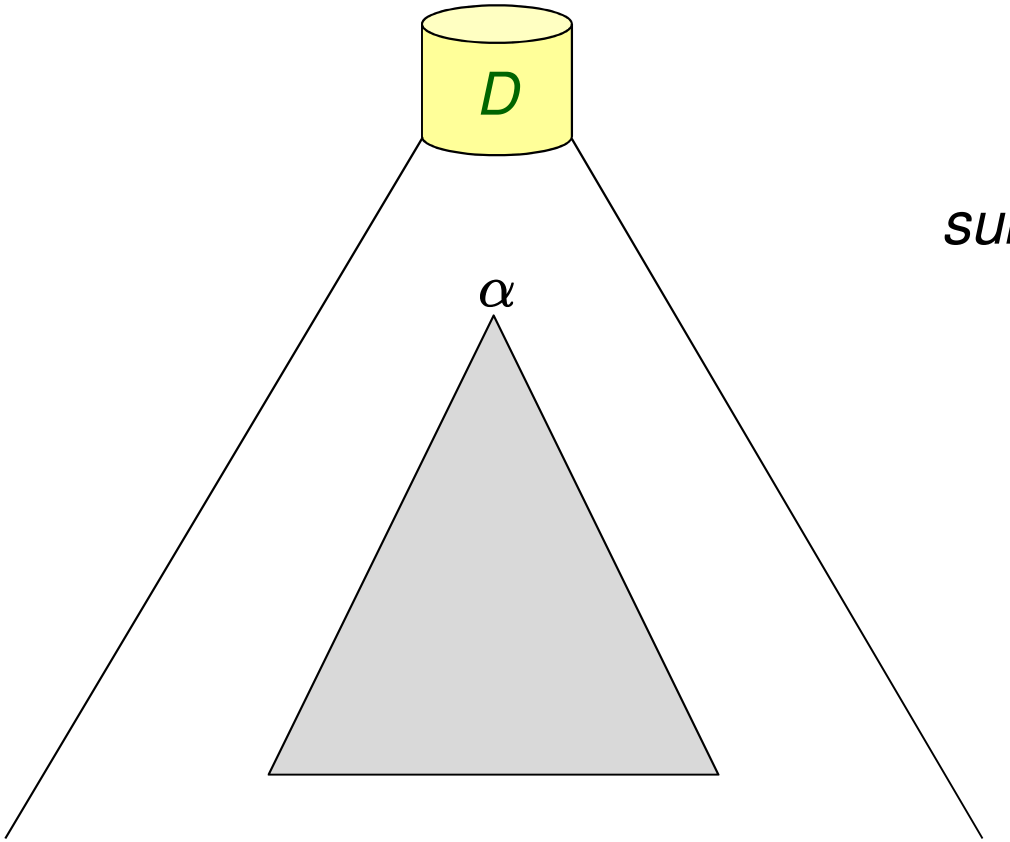
Type of an atom

$$type(\alpha, D, O) = \{\ \beta \in chase(D, O) \mid termsOf(\beta) \subseteq termsOf(\alpha)\ \}$$

The Guarded Case

Type of an atom

$$type(\alpha, D, O) = \{ \beta \in chase(D, O) \mid termsOf(\beta) \subseteq termsOf(\alpha) \}$$



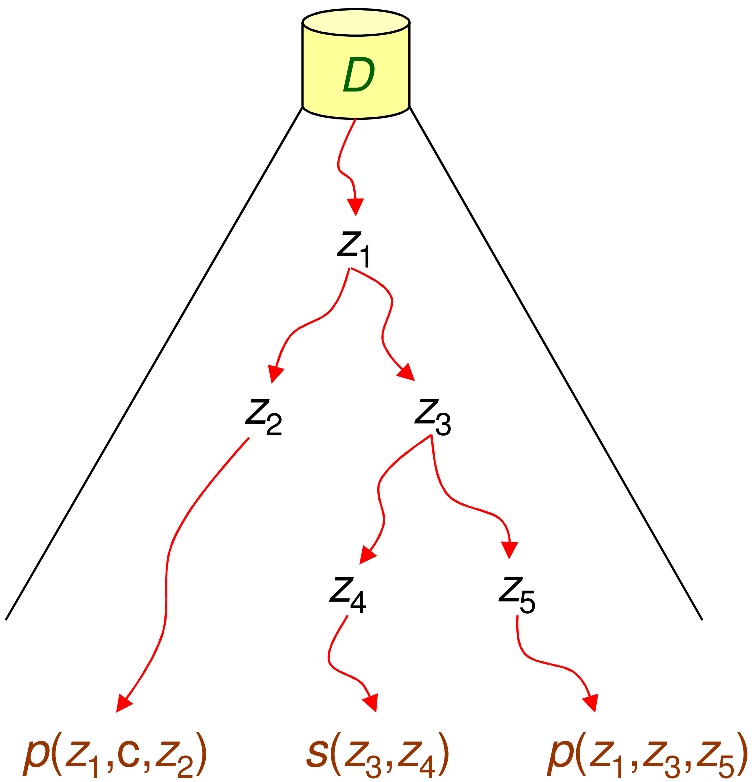
$subtreeOf(\alpha)$ is determined by

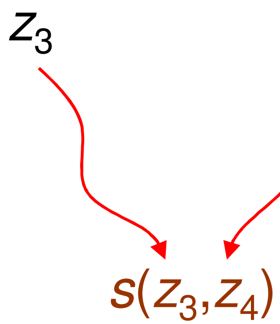
$type(\alpha, D, O)$

The Guarded Case

An alternating algorithm

- Guess the image of the given query - $p(z_1,\mathrm{c},z_2),s(z_3,z_4),p(z_1,z_3,z_5)$
- Guess the order of null generation - $z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4 \quad z_1 \prec z_3 \prec z_5$





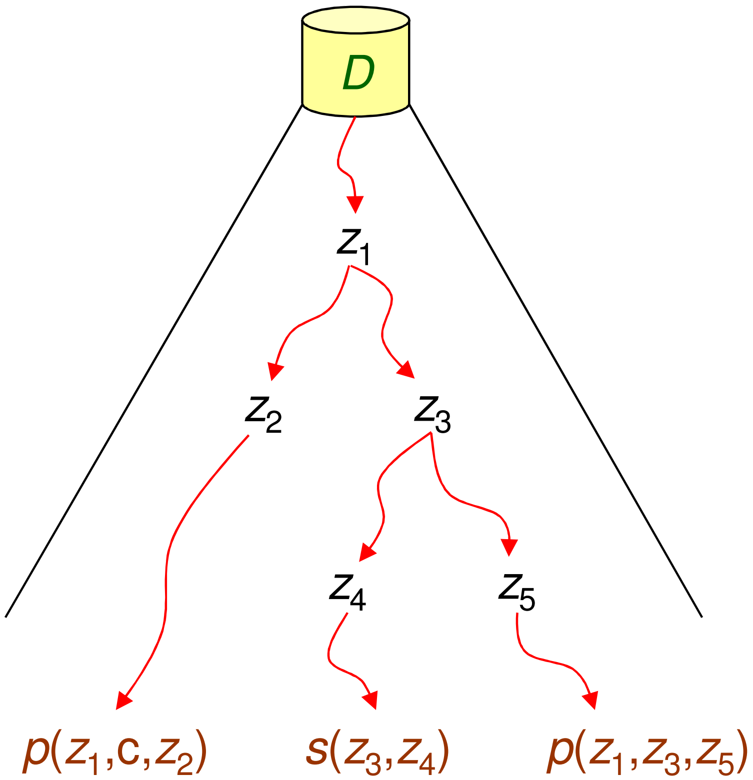
$$q(\ldots,X,\ldots),t(\ldots,Y,\ldots) \;\rightarrow\; s(X,Y)$$

non-guarded

The Guarded Case

An alternating algorithm

- Guess the image of the given query - $p(z_1,c,z_2),s(z_3,z_4),p(z_1,z_3,z_5)$
- Guess the order of null generation - $z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4 \quad z_1 \prec z_3 \prec z_5$
- Universally prove the relevant chase derivations



remember the type of an atom

$$|type(\alpha,D,O)| \leq \#pred \cdot maxarity^{maxarity}$$

$$\Downarrow$$

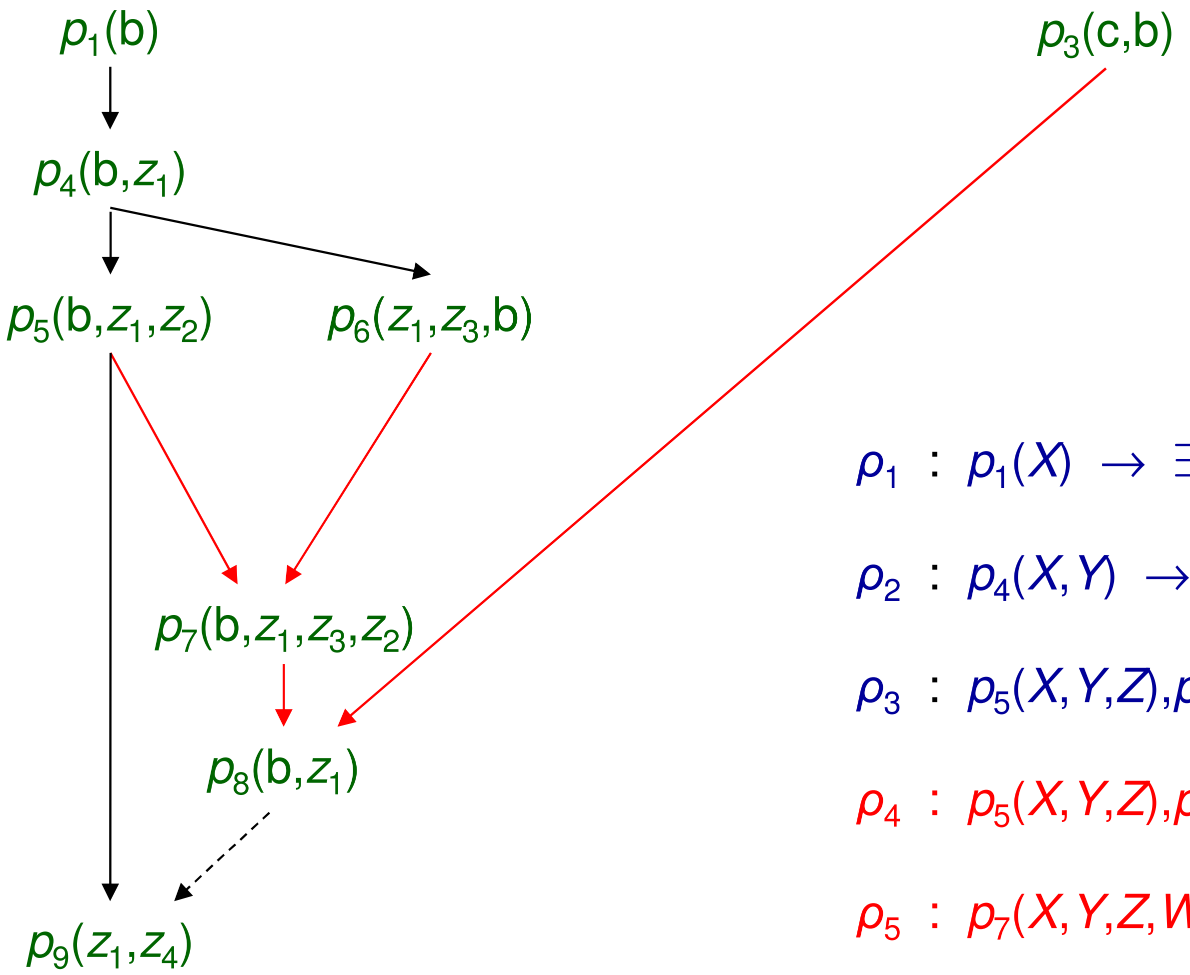
AEXPSPACE = 2EXPTIME (combined)

APSPACE = EXPTIME (bounded arity)

ALOGSPACE = PTIME (data)

The Tamed Case - Difficulty I

Determine the subtree of an atom - the type is not enough



$$\rho_1 \;:\; p_1(X) \rightarrow \exists Y\, p_4(X,Y)$$

$$\rho_2 \;:\; p_4(X,Y) \rightarrow \exists Z \exists W\, p_5(X,Y,Z), p_6(Y,W,X)$$

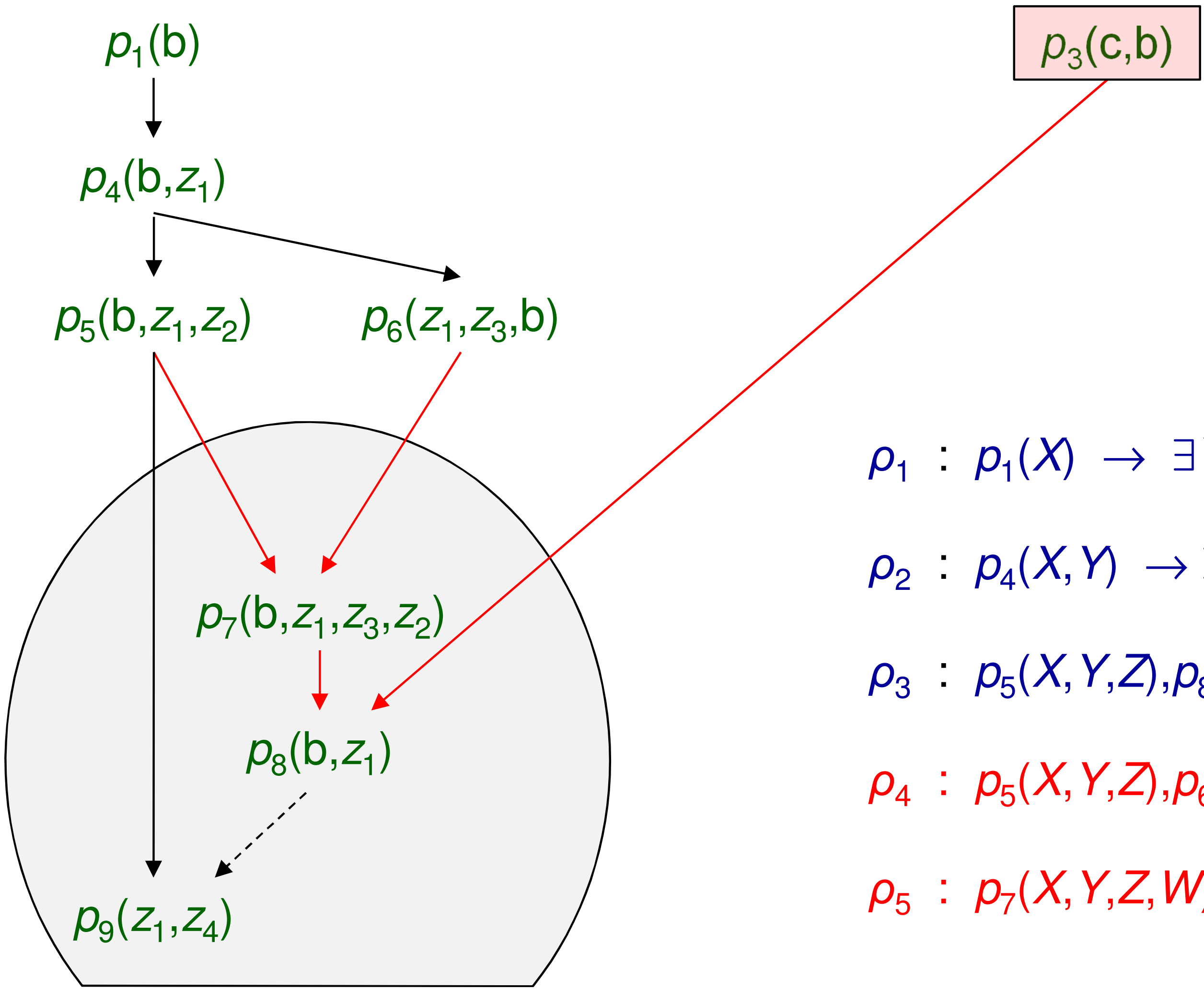
$$\rho_3 \;:\; p_5(X,Y,Z), p_8(X,Y) \rightarrow \exists W\, p_9(Y,W)$$

$$\rho_4 \;:\; p_5(X,Y,Z), p_6(Y,W,X) \rightarrow p_7(X,Y,W,Z)$$

$$\rho_5 \;:\; p_7(X,Y,Z,W), p_3(V,X) \rightarrow p_8(X,Y)$$

The Tamed Case - Difficulty I

Determine the subtree of an atom - the type is not enough



not in the type of $p_5(\mathrm{b}, z_1, z_2)$

$$\rho_1 \;:\; p_1(X) \rightarrow \exists Y\, p_4(X,Y)$$

$$\rho_2 \;:\; p_4(X,Y) \rightarrow \exists Z \exists W\, p_5(X,Y,Z), p_6(Y,W,X)$$

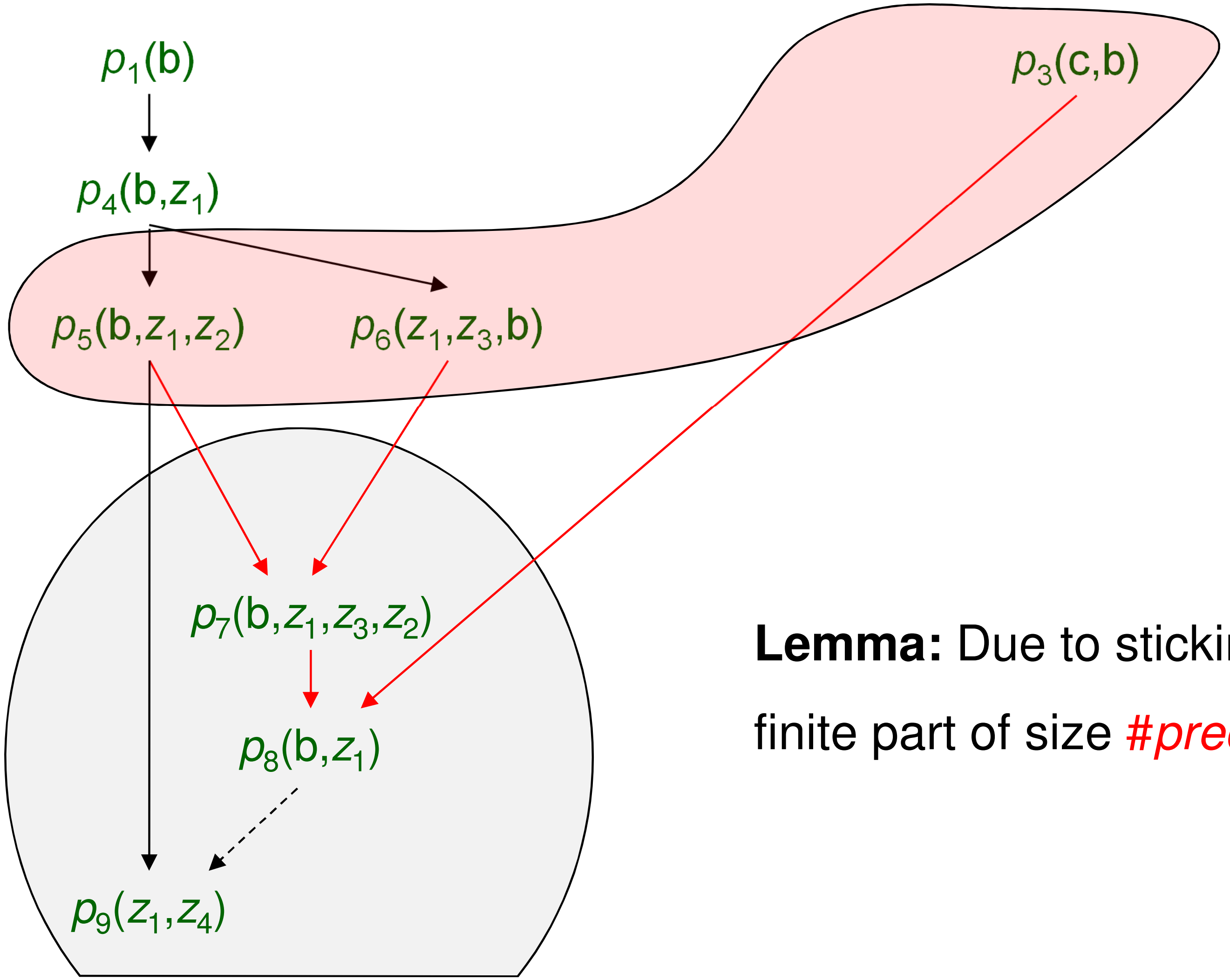
$$\rho_3 \;:\; p_5(X,Y,Z), p_8(X,Y) \rightarrow \exists W\, p_9(Y,W)$$

$$\rho_4 \;:\; p_5(X,Y,Z), p_6(Y,W,X) \rightarrow p_7(X,Y,W,Z)$$

$$\rho_5 \;:\; p_7(X,Y,Z,W), p_3(V,X) \rightarrow p_8(X,Y)$$

The Tamed Case - Difficulty I

Active type of an atom



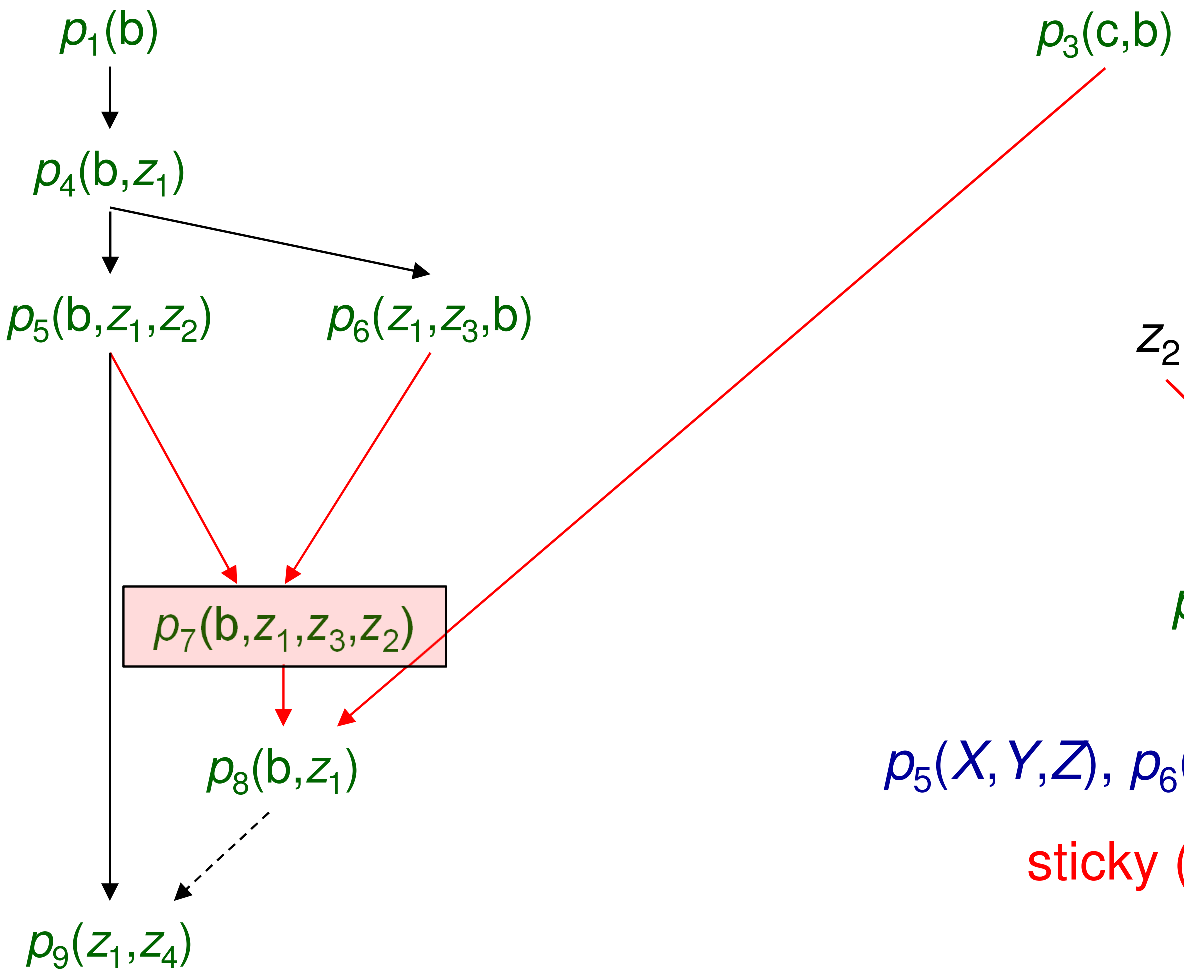
the active type is

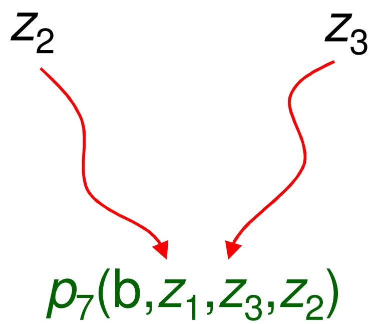
in general infinite

Lemma: Due to stickiness we can focus on a finite part of size $\#pred \cdot (maxarity + 1)^{maxarity}$.

The Tamed Case - Difficulty II

Order of null generation - mixing of incompatible guarded nulls



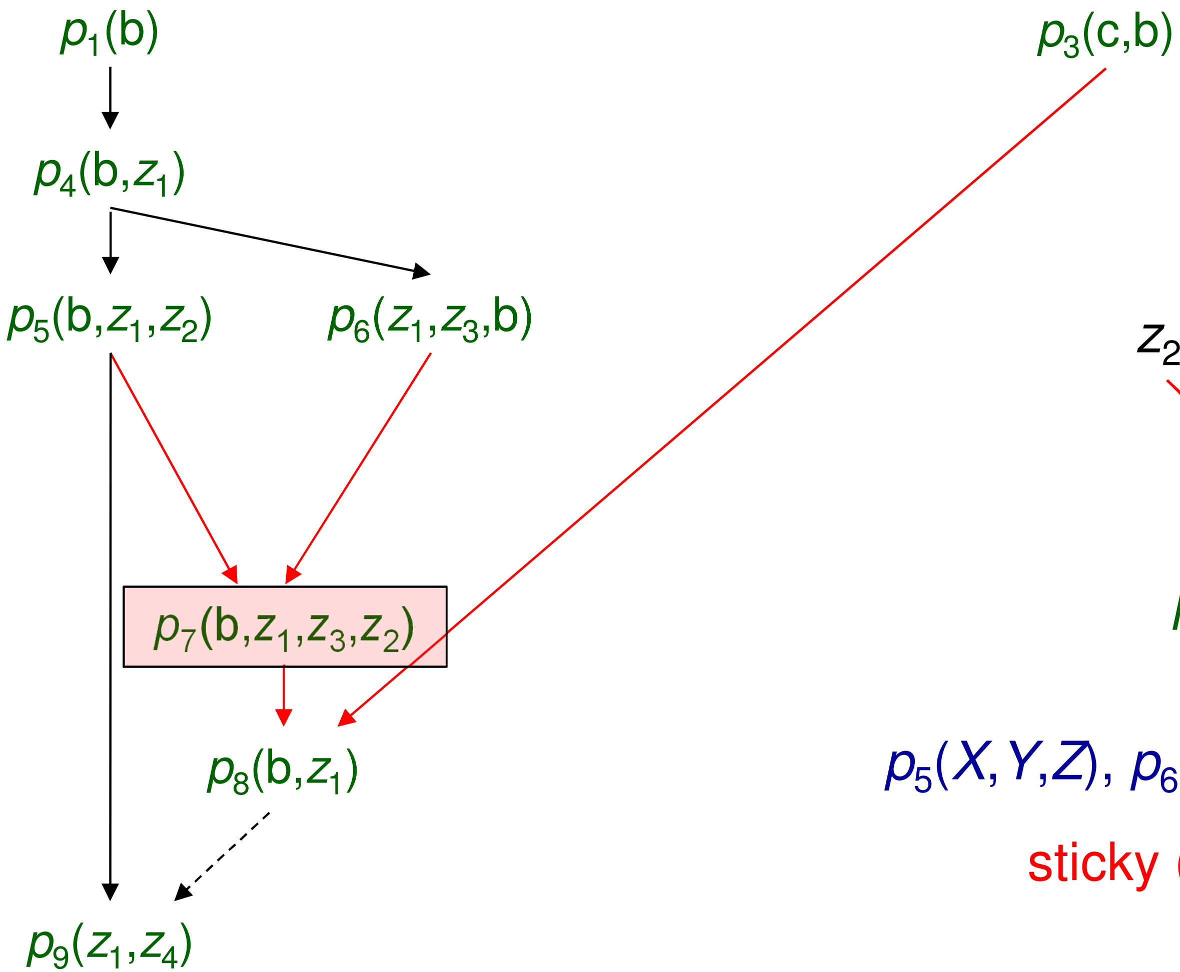


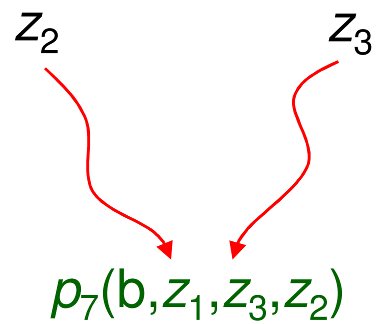
$p_5(X,Y,Z),\ p_6(Y,W,X) \;\rightarrow\; p_7(X,Y,W,Z)$

sticky (non-guarded) rule

The Tamed Case - Difficulty II

Order of null generation - mixing of incompatible guarded nulls





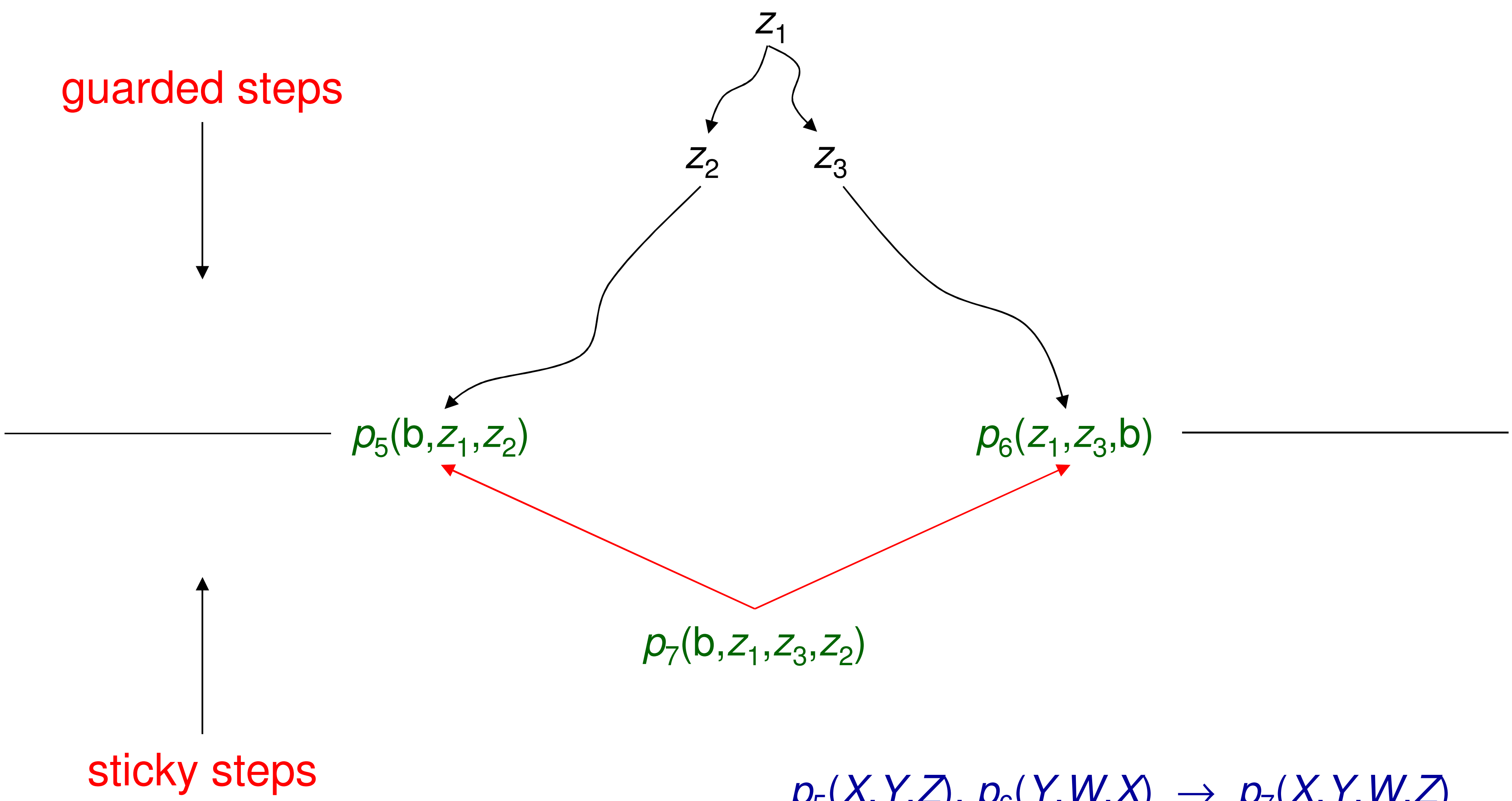
$$p_5(X,Y,Z),\ p_6(Y,W,X)\ \rightarrow\ p_7(X,Y,W,Z)$$

sticky (non-guarded) rule

…mixing of guarded and sticky nulls

The Tamed Case - Difficulty II

Backward resolution steps (sticky steps)

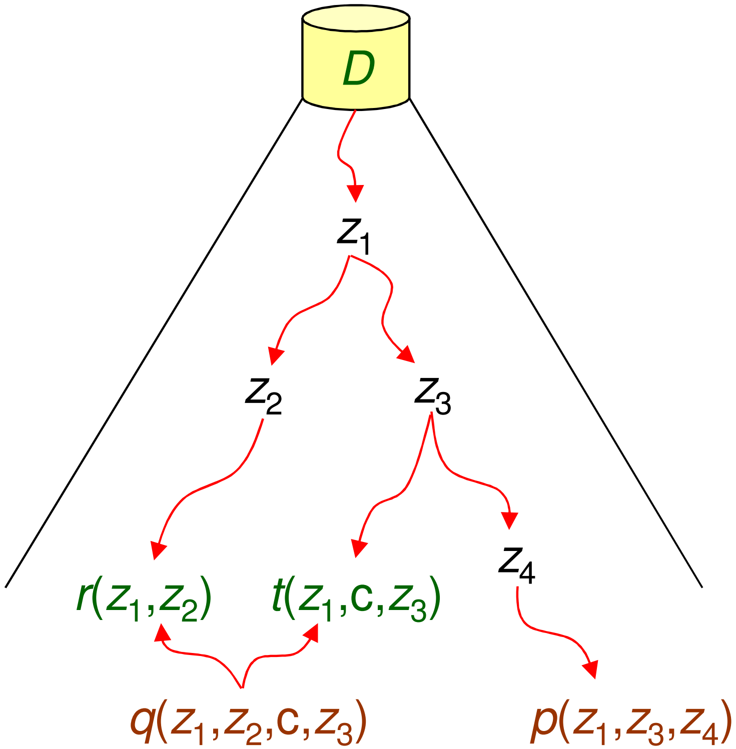


$$p_5(X,Y,Z),\ p_6(Y,W,X)\ \rightarrow\ p_7(X,Y,W,Z)$$

The Tamed Case

A (hybrid) alternating algorithm

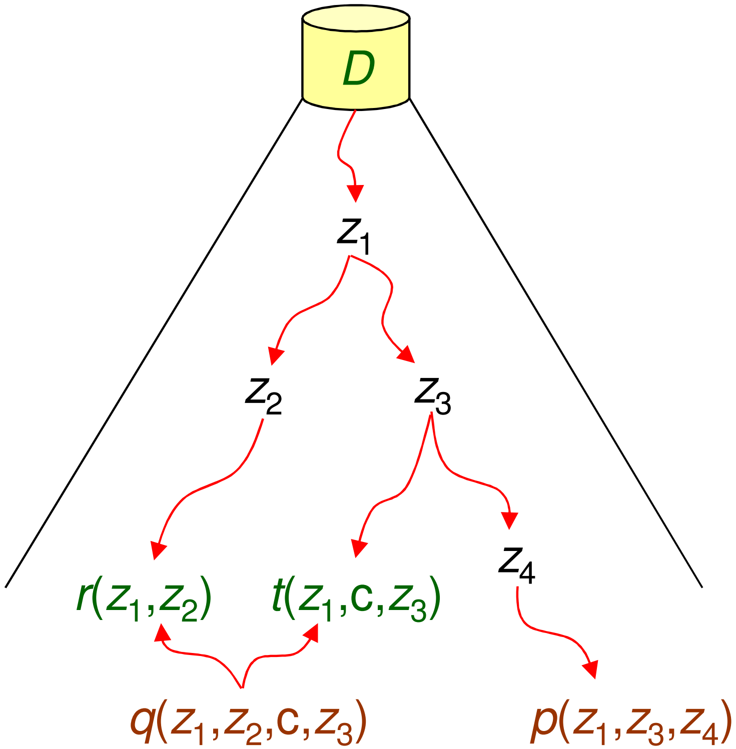
- Guess the image of the given query - $q(z_1,z_2,c,z_3),p(z_1,z_3,z_4)$
- Guess a partition of nulls into sticky and guarded
- Guess the order of guarded null generation - $z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4$



The Tamed Case

A (hybrid) alternating algorithm

- Guess the image of the given query - $q(z_1,z_2,c,z_3),p(z_1,z_3,z_4)$
- Guess a partition of nulls into sticky and guarded
- Guess the order of guarded null generation - $z_1 \prec z_2 \quad z_1 \prec z_3 \prec z_4$
- Universally prove the relevant chase derivations (guarded + sticky steps)



remember the active type of an atom

$$|atype(\alpha,D,O)| \leq \#pred \cdot (maxarity + 1)^{maxarity}$$

$$\Downarrow$$

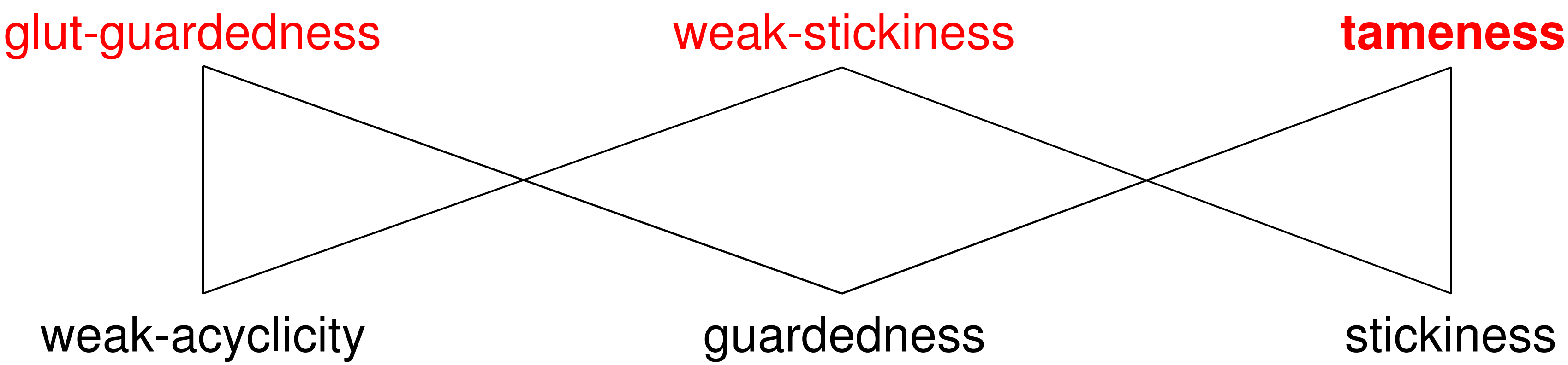
Theorem: Query answering under tame G|S is in:

AEXPSPACE = 2EXPTIME (combined)

APSPACE = EXPTIME (bounded arity)

ALOGSPACE = PTIME (data).

Overview



- Glut-guardedness - guard only harmful variables

  [Krötzsch & Rudolph, IJCAI 2011]

- Weak-stickiness - only harmful join-variables stick to the inferred atoms

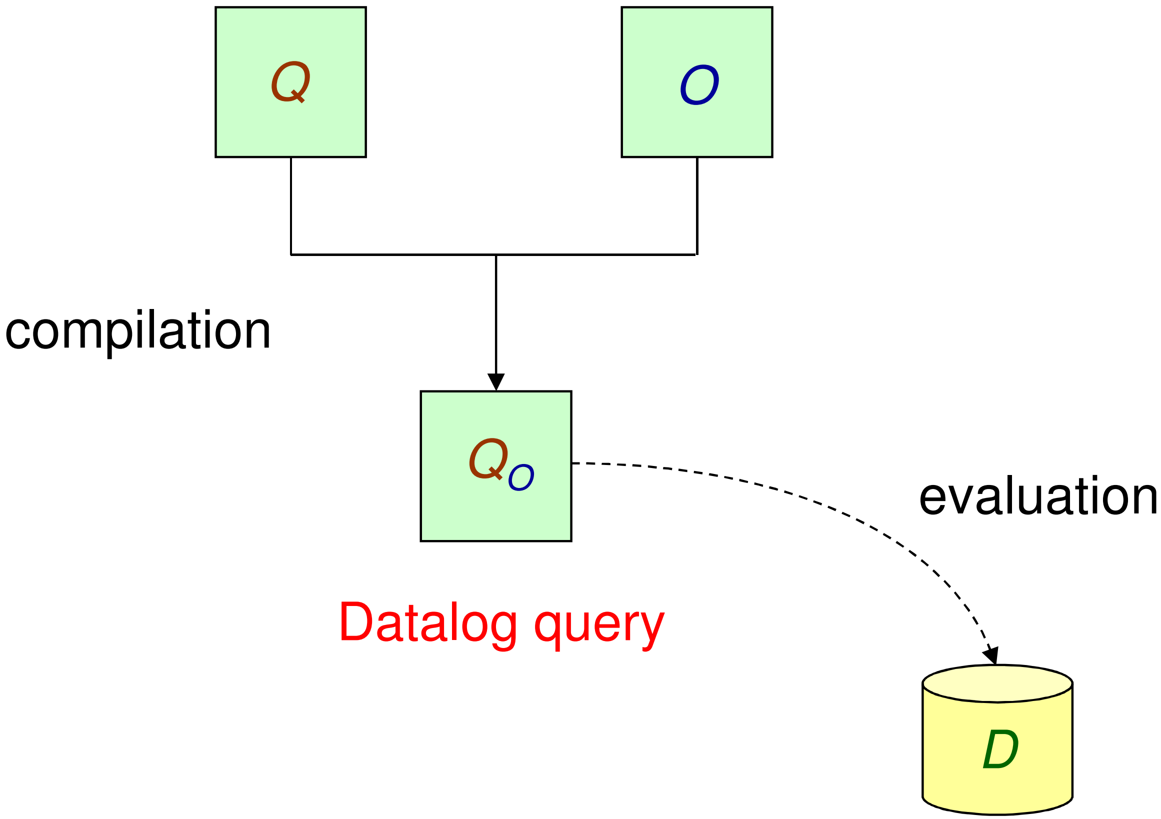
  [Calì, Gottlob & P., Artificial Intelligence 2010]

- Tameness - sticky rules do not feed the guard atom

  [this work]

Next Step: Datalog Rewriting

Towards practical query answering algorithms



Thank you!