Datalog$^{\pm}$: A Family of Languages for Ontology Querying

Georg Gottlob

Department of Computer Science

University of Oxford

joint work with Andrea Calì,Thomas Lukasiewicz, Marco Manna, Andreas Pieris et al.

1st Motivation: Ontological Query Answering

– Input: a knowledge base $K = \langle D, \Sigma \rangle$

a Boolean conjunctive query Q

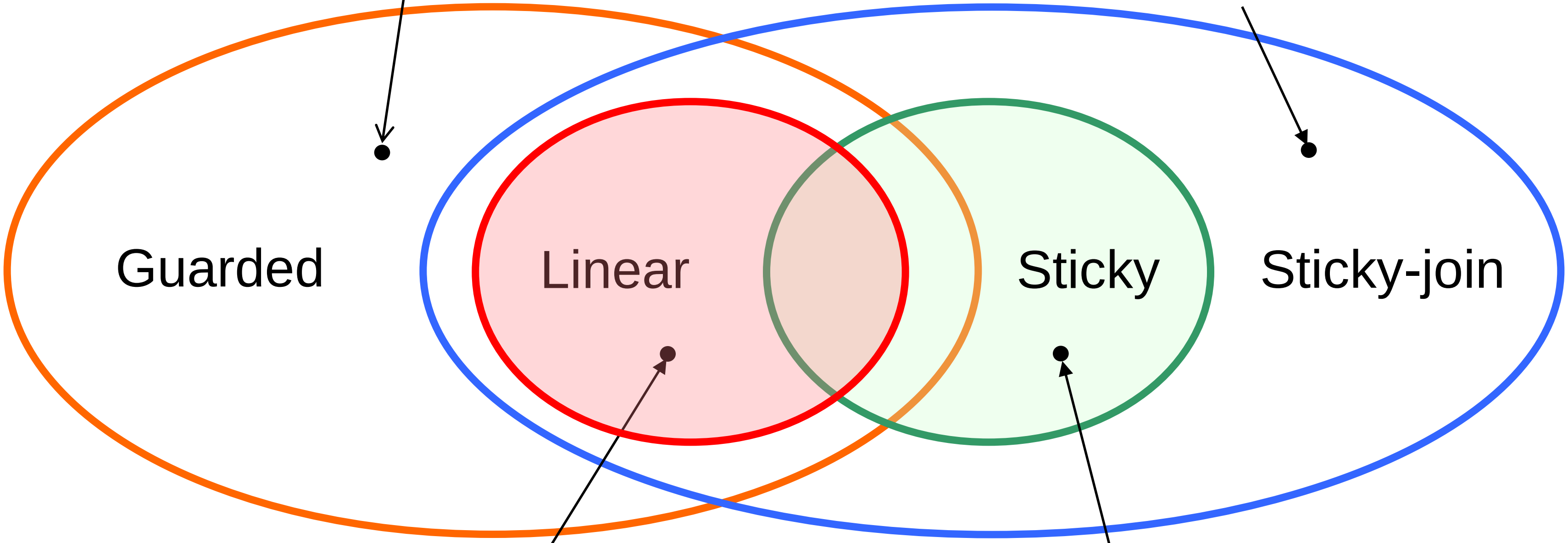
Question: $K \models Q$ (or, equivalently, $D \cup \Sigma \models Q$)

– Old database problem – CQ answering over incomplete databases

2nd Motivation: Decidable FO fragments (TGDs)
Good data complexity!

$\forall X \forall Y \forall Z\; R(X,Y,Z), S(Y), P(X,Z) \rightarrow \exists W\; Q(X,W)$

$\forall X \forall Y \forall Z\; R(X,Y), S(Y,Z,Z) \rightarrow \exists W\; P(Y,W)$



$\forall X \forall Y\; R(X,X,Y) \rightarrow \exists Z\; R(Y,Y,Z)$

$\forall X \forall Y \forall Z\; S(X,Y), R(Y,Z) \rightarrow \exists W\; P(Y,W)$

2nd Motivation: Decidable FO fragments (TGDs)

Good data complexity!

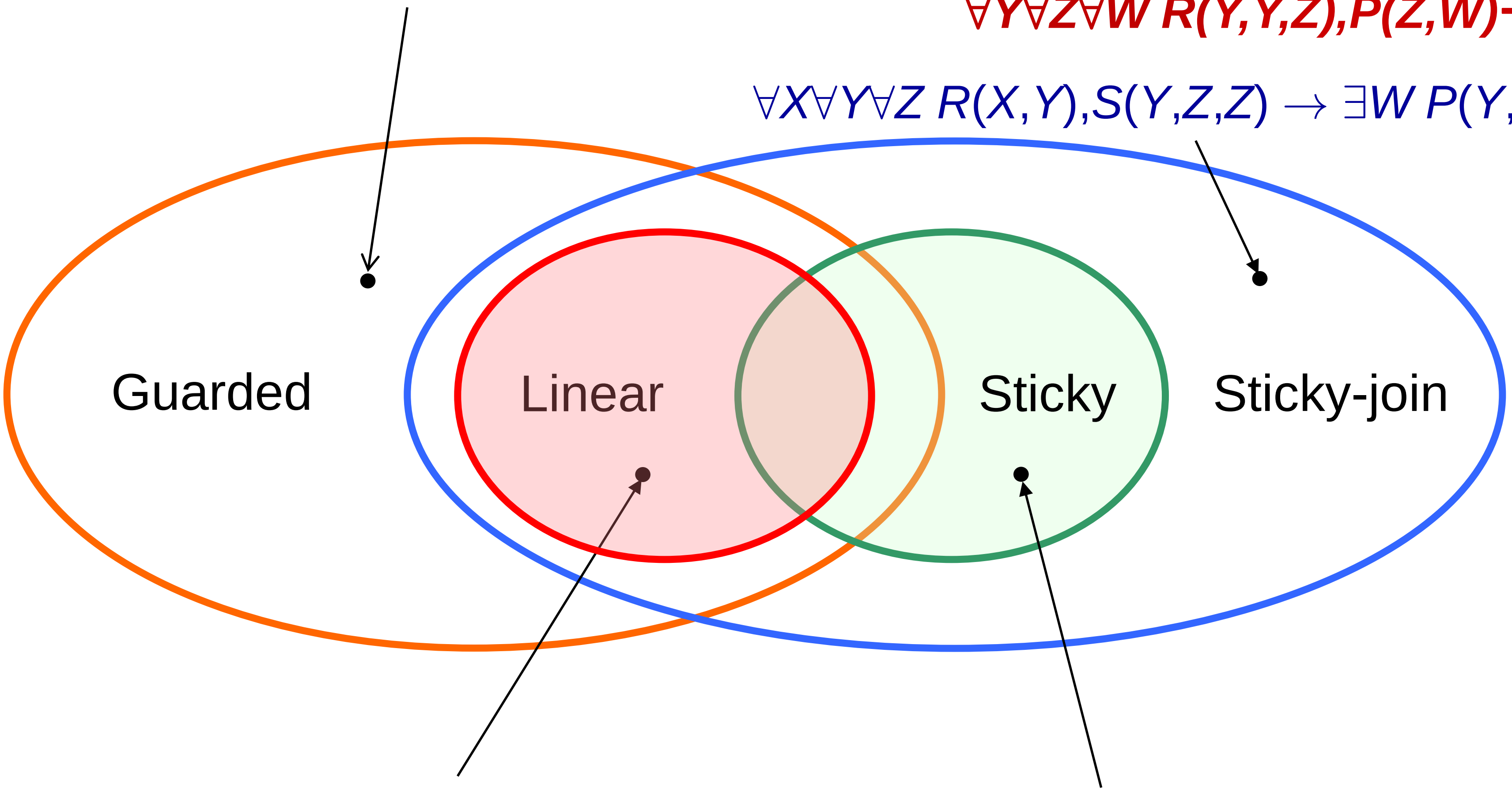
***Query:** $R(Y,Y,Z),P(Z,W)$*

Add rule:

$\forall Y \forall Z \forall W\ R(Y,Y,Z),P(Z,W) \rightarrow \bot$

$\forall X \forall Y \forall Z\ R(X,Y,Z),S(Y),P(X,Z) \rightarrow \exists W\ Q(X,W)$

$\forall X \forall Y \forall Z\ R(X,Y),S(Y,Z,Z) \rightarrow \exists W\ P(Y,W)$

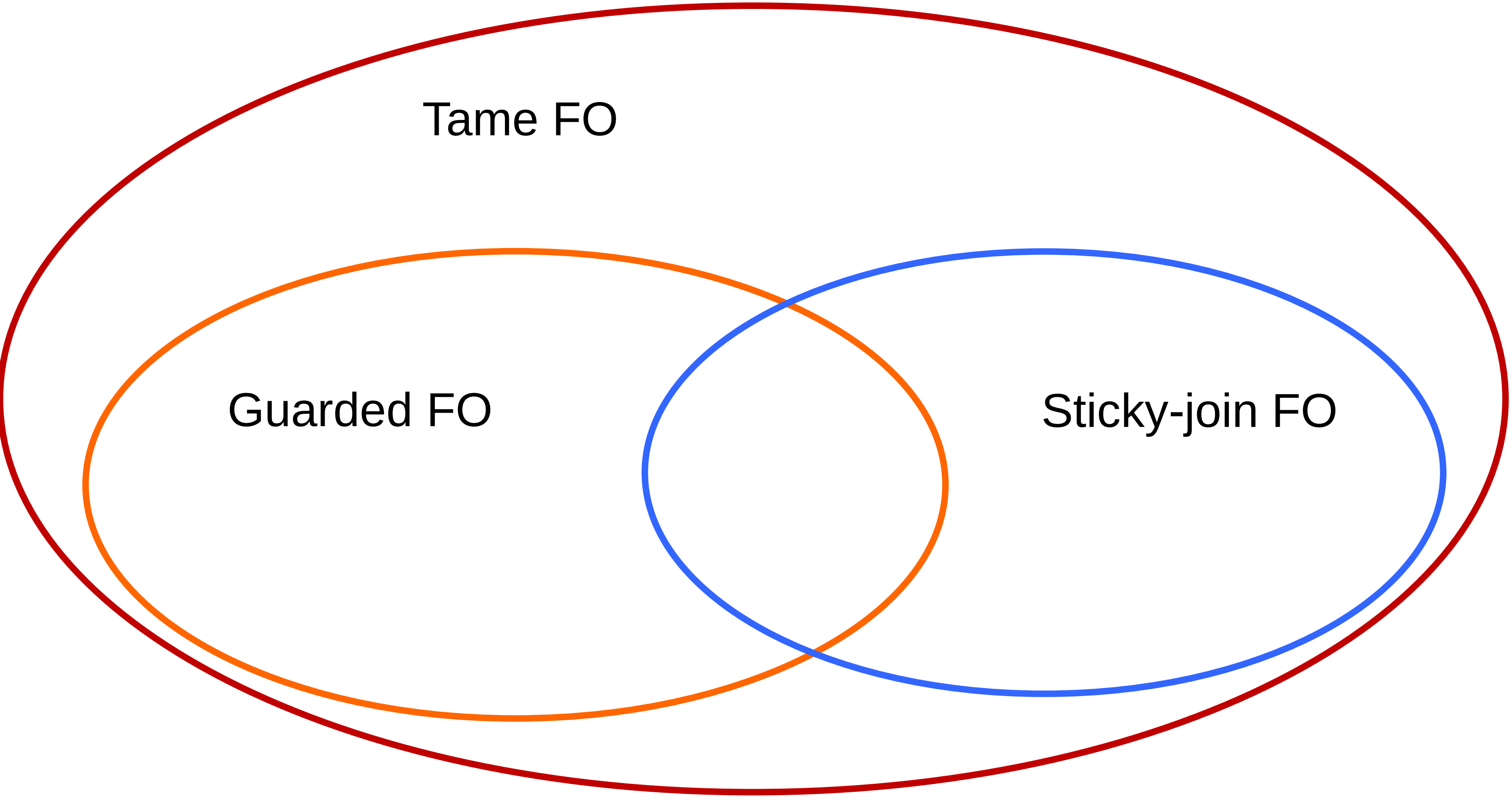


$\forall X \forall Y\ R(X,X,Y) \rightarrow \exists Z\ R(Y,Y,Z)$

$\forall X \forall Y \forall Z\ S(X,Y),R(Y,Z) \rightarrow \exists W\ P(Y,W)$

3rd Motivation: Decidable FO fragments

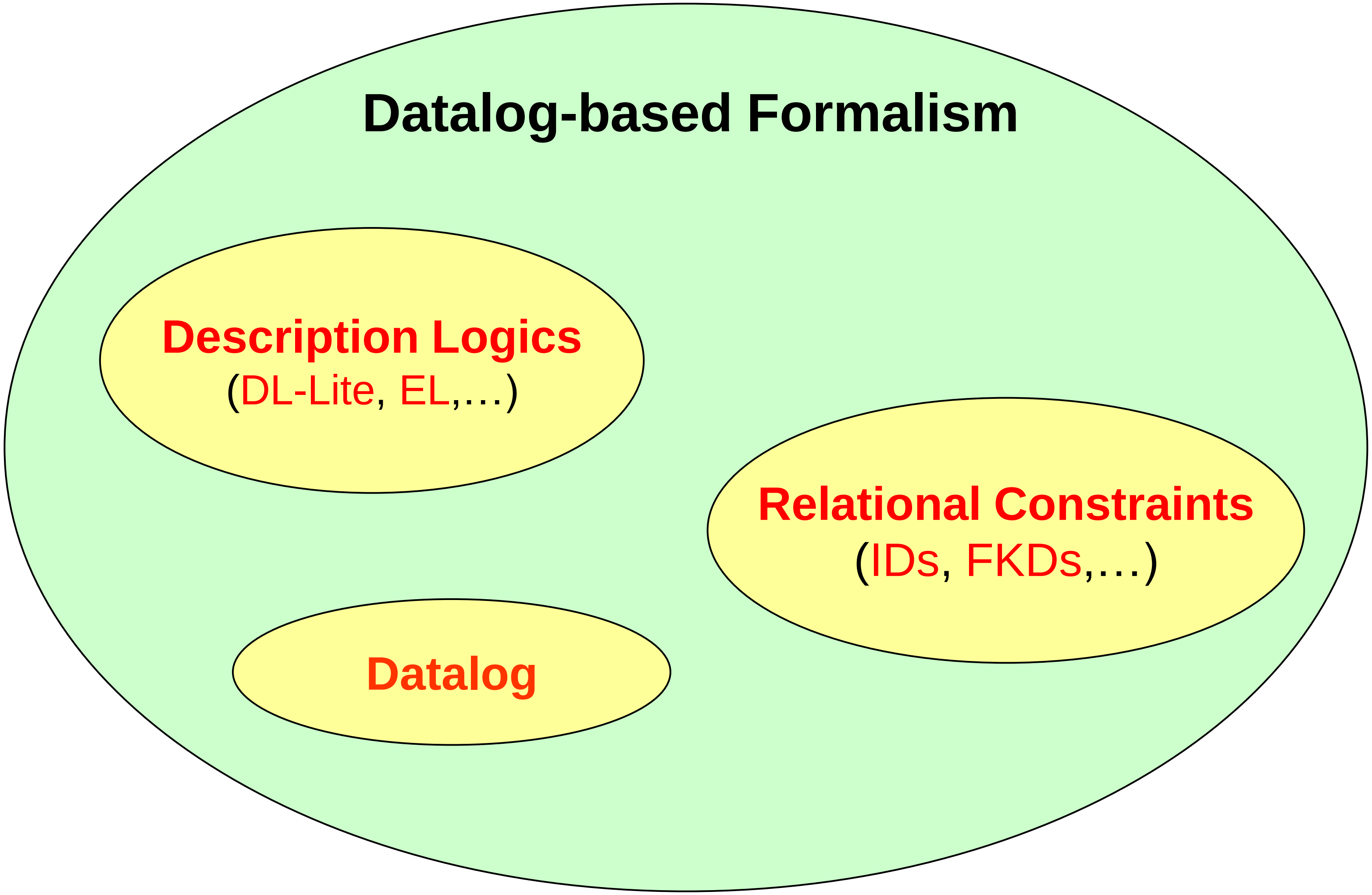
(current work)



(Some) Known Results

Σ	Data Complexity	Combined Complexity
DL-Lite	in AC_0 [Calvanese et al., JAR 07]	NP-complete [Calvanese et al., JAR 07]
EL + ELH	PTIME-complete [Rosati, DL 07]	NP-complete [Rosati, DL 07]
Horn-SHIQ	PTIME-complete [Eiter et al., JELIA 08]	EXPTIME-complete [Eiter et al., JELIA 08]
IDs + FKDs	in AC_0 [Calì et al., IJCAI 03]	PSPACE-complete [Johnson & Klug, JCSS 84]
Datalog	PTIME-complete	EXPTIME-complete

Our Goal


Datalog-based Formalism

Description Logics
(DL-Lite, EL,…)

Relational Constraints
(IDs, FKDs,…)

Datalog

… without losing tractable data complexity

Ontological Reasoning and Datalog

DL Assertion	Datalog Rule
Concept Inclusion	
$emp \sqsubseteq person$	$emp(X) \rightarrow person(X)$
Concept Product	
$sen\text{-}emp \times emp \sqsubseteq moreThan$	$sen\text{-}emp(X), emp(Y) \rightarrow moreThan(X,Y)$
(Inverse) Role Inclusion	
$reports^{-} \sqsubseteq mgr$	$reports(X,Y) \rightarrow mgr(Y,X)$
Role Transitivity	
trans(mgr)	$mgr(X,Y), mgr(Y,Z) \rightarrow mgr(X,Z)$

Ontological Reasoning and Datalog

DL Assertion	Datalog Rule
Concept Inclusion $emp \sqsubseteq person$	$emp(X) \rightarrow person(X)$
Concept Product $sen\text{-}emp \times emp \sqsubseteq moreThan$	$sen\text{-}emp(X),emp(Y) \rightarrow moreThan(X,Y)$
(Inverse) Role Inclusion $reports^{-} \sqsubseteq mgr$	$reports(X,Y) \rightarrow mgr(Y,X)$
Role Transitivity trans(mgr)	$mgr(X,Y),mgr(Y,Z) \rightarrow mgr(X,Z)$
Participation $emp \sqsubseteq \exists report$	$emp(X) \rightarrow \exists Y\ report(X,Y)$
Disjointness $emp \sqcap customer \sqsubseteq \bot$	$emp(X),customer(X) \rightarrow \bot$
Functionality funct($reports$)	$reports(X,Y),reports(X,Z) \rightarrow Y = Z$

Datalog$^{\pm}$

– Datalog: $\forall \mathbf{X} \forall \mathbf{Y}\ \Phi(\mathbf{X},\mathbf{Y}) \rightarrow R(\mathbf{X})$

– Extend Datalog by allowing in the head:

- Existential quantification ($\exists$) - TGDs: $\forall \mathbf{X} \forall \mathbf{Y}\ \Phi(\mathbf{X},\mathbf{Y}) \rightarrow \exists \mathbf{Z}\ \Psi(\mathbf{X},\mathbf{Z})$
- Equality atoms (=) - EGDs: $\forall \mathbf{X}\ \Phi(\mathbf{X}) \rightarrow X_i = X_j$
- Constant false ($\perp$) - Negative constraints: $\forall \mathbf{X}\ \Phi(\mathbf{X}) \rightarrow \perp$

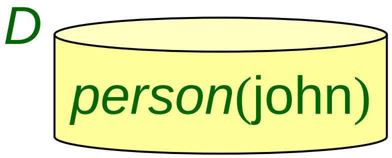
– But query answering under Datalog[$\exists$] is undecidable
[see, e.g., Beeri & Vardi, ICALP 81]

– Datalog[$\exists$,=,$\perp$] is syntactically restricted → **Datalog$^{\pm}$**

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$



D

person(john)

Σ

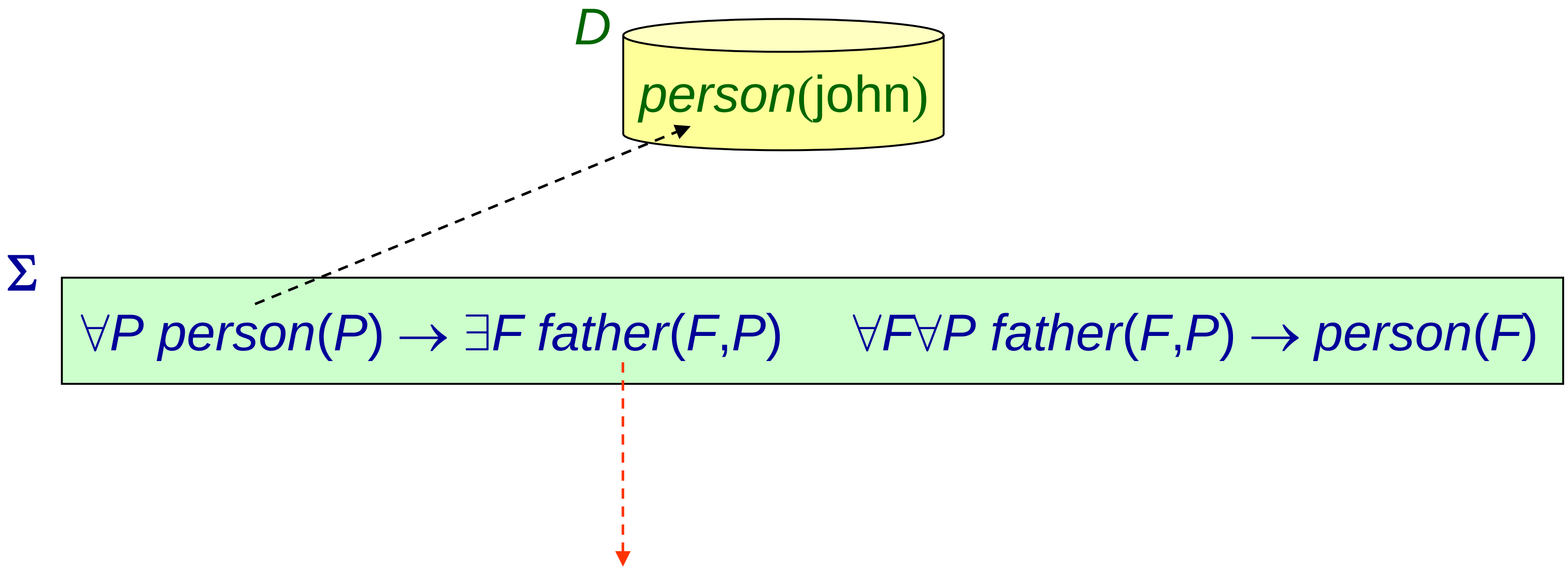
$\forall P\ person(P) \rightarrow \exists F\ father(F,P) \qquad \forall F \forall P\ father(F,P) \rightarrow person(F)$

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

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$



D

person(john)

Σ

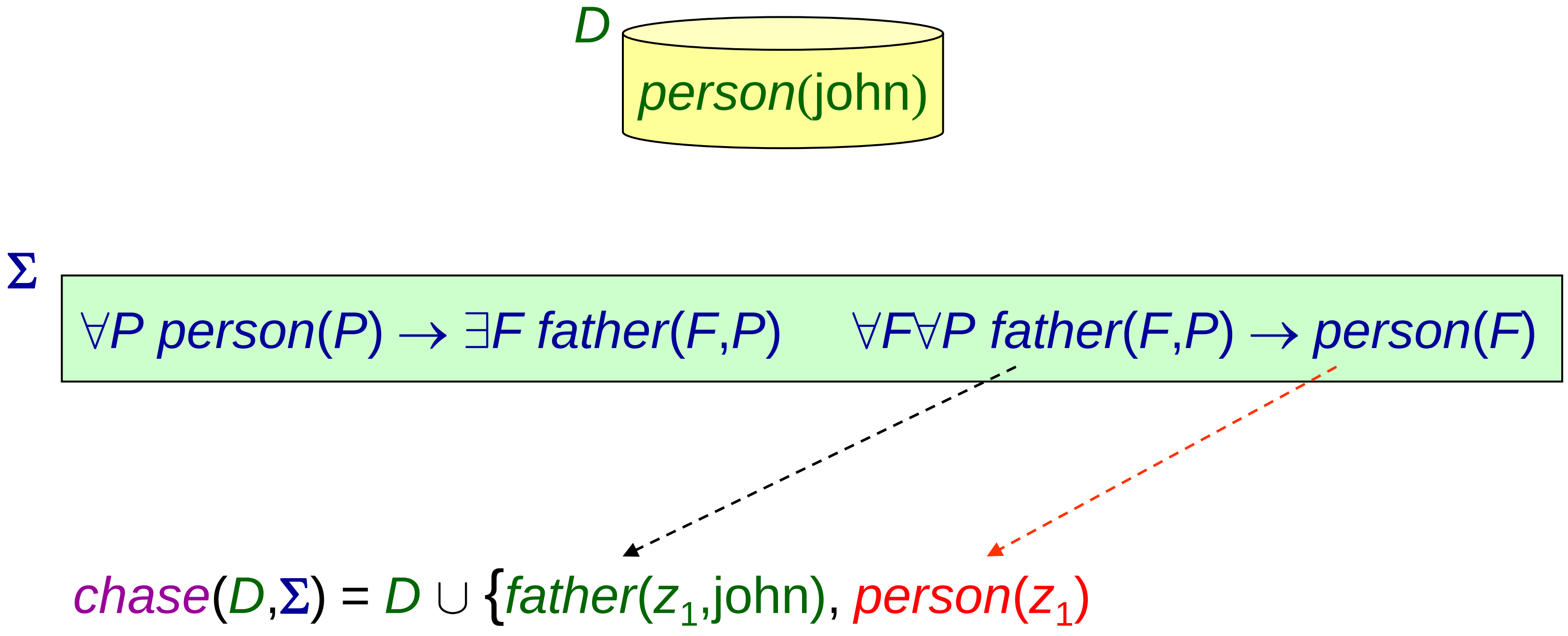
$\forall P\ person(P) \rightarrow \exists F\ father(F,P) \qquad \forall F \forall P\ father(F,P) \rightarrow person(F)$

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

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$



D

person(john)

Σ

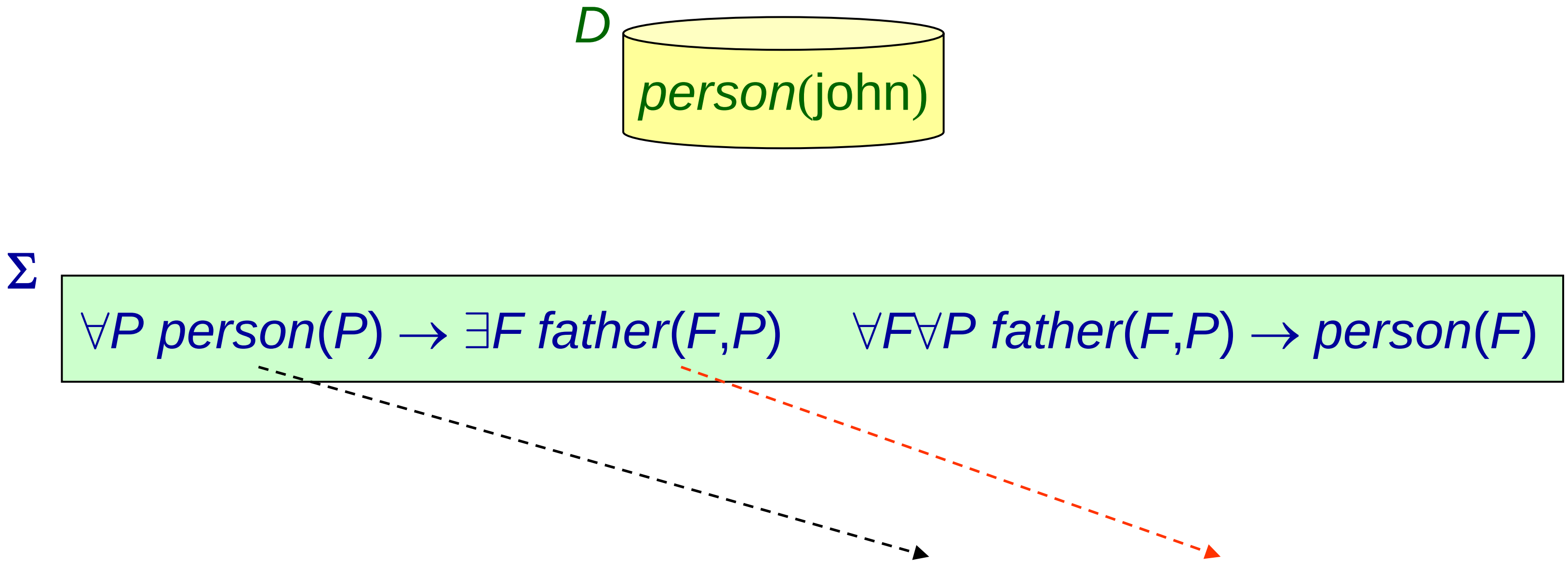
$\forall P\ person(P) \rightarrow \exists F\ father(F,P)$ $\quad$ $\forall F \forall P\ father(F,P) \rightarrow person(F)$

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

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$



D

person(john)

Σ

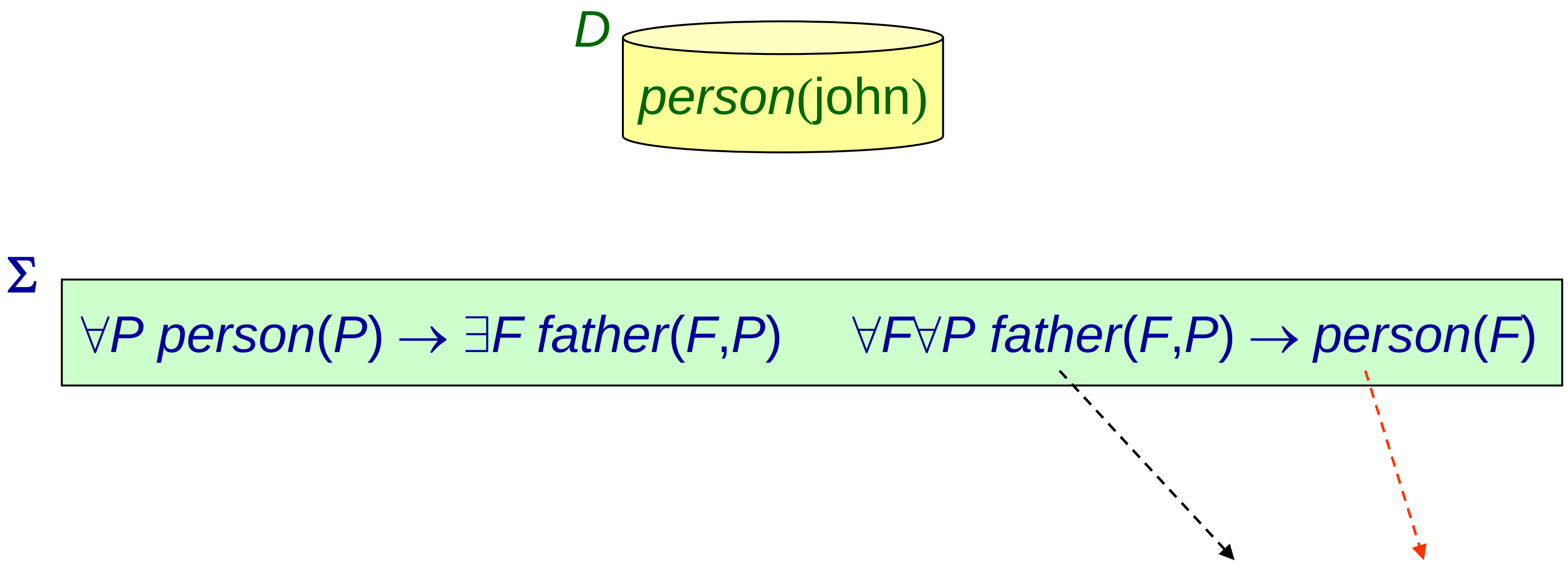
$\forall P\ person(P) \rightarrow \exists F\ father(F,P)$ $\quad$ $\forall F \forall P\ father(F,P) \rightarrow person(F)$

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

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$



D

person(john)

Σ

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

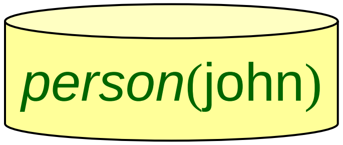
$chase(D,\Sigma) = D \cup \{father(z_1,\text{john}),\ person(z_1),\ father(z_2,z_1),\ \ldots\}$

The Chase Procedure

Input: Database D, set of TGDs Σ

Output: A model of $D \cup \Sigma$

D



$person(\text{john})$

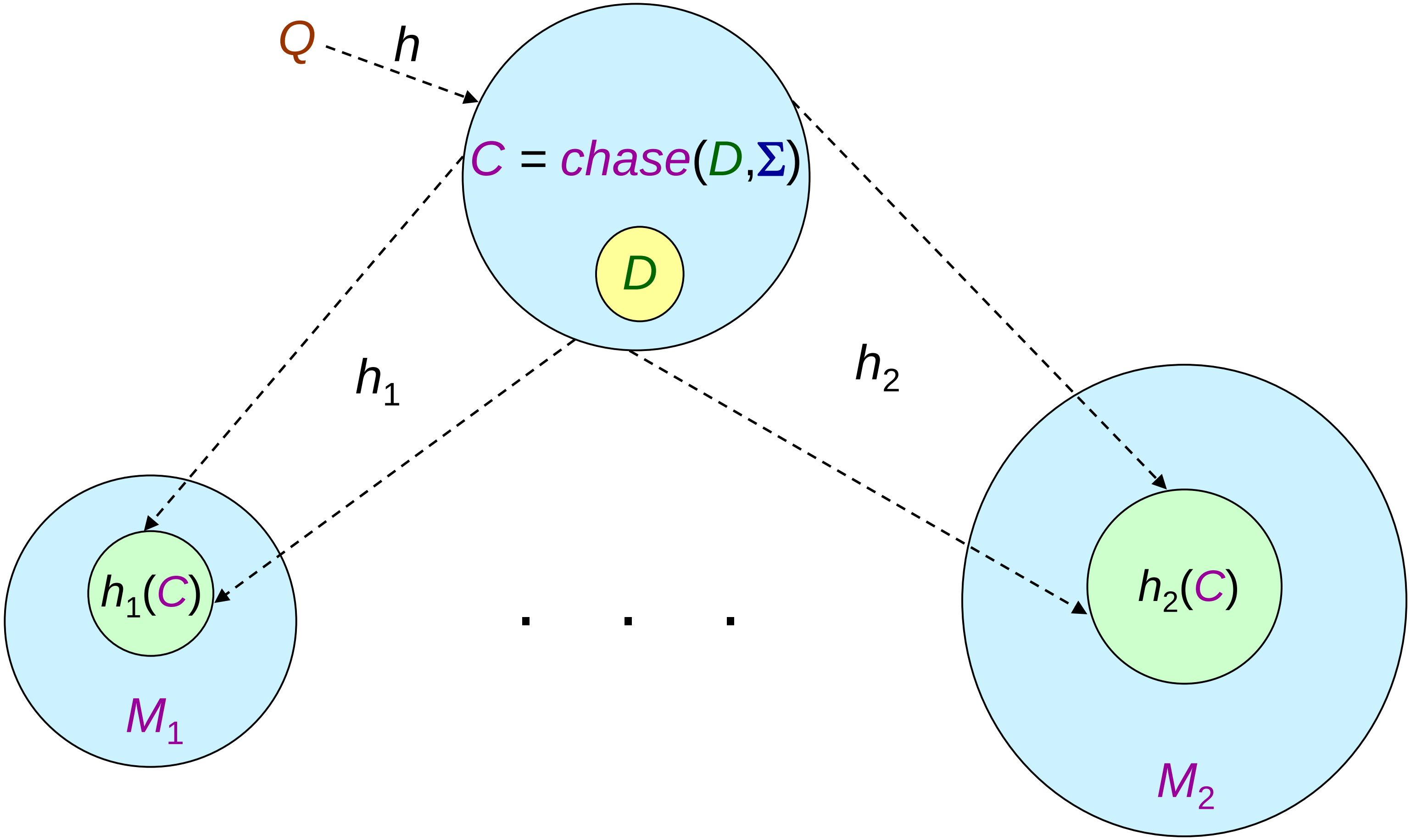
Σ

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

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

infinite instance

Query Answering via Chase



$$D \cup \Sigma \models Q \quad \Leftrightarrow \quad chase(D,\Sigma) \models Q$$

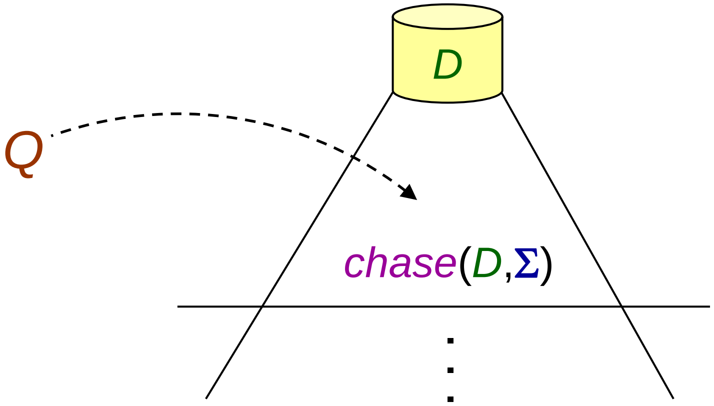
[see, e.g., Fagin, Kolaitis, Miller & Popa, TCS 05]

Early Positive Result

– Query answering under IDs is decidable P(X,Y) → ∃Z R(X,Y,Z)

- PSPACE-complete in combined complexity

- NP-complete for bounded arity



[Johnson & Klug, JCSS 84]

Guardedness

- All ∀-variables occur in one body atom - guard atom

$$\forall X \forall Y \forall Z\ \underset{\substack{\nearrow \\ \text{guard}}}{R(X,Y,Z)}, S(Y), P(X,Z) \rightarrow \exists W\ Q(X,W)$$

- Chase has finite treewidth $\Rightarrow$ decidability of query answering

  [Calì, G. & Kifer, KR 08]

- Query answering is PTIME-complete in data complexity

  [Calì, G. & Lukasiewicz, PODS 09]

- Properly extends ELH (same data complexity)

Ontology Querying

ELH: Popular DL (for biological applications) with PTIME data complexity
[Baader, IJCAI 03 and Rosati, DL 07]

ELH TBox	Datalog$^{\pm}$ Representation
$A \sqsubseteq B$	$\forall X\, A(X) \rightarrow B(X)$
$A \sqcap B \sqsubseteq C$	$\forall X\, A(X), B(X) \rightarrow C(X)$
$\exists R.A \sqsubseteq B$	$\forall X\, R(X,Y), A(Y) \rightarrow B(X)$
$A \sqsubseteq \exists R.B$	$\forall X\, A(X) \rightarrow \exists Y\, R(X,Y), B(Y)$
$R \sqsubseteq P$	$\forall X \forall Y\, R(X,Y) \rightarrow P(X,Y)$

Linearity

- Just one atom in the body $\forall \mathbf{X} \forall \mathbf{Y}\, R(\mathbf{X},\mathbf{Y}) \rightarrow \exists \mathbf{Z}\, \Psi(\mathbf{X},\mathbf{Z})$

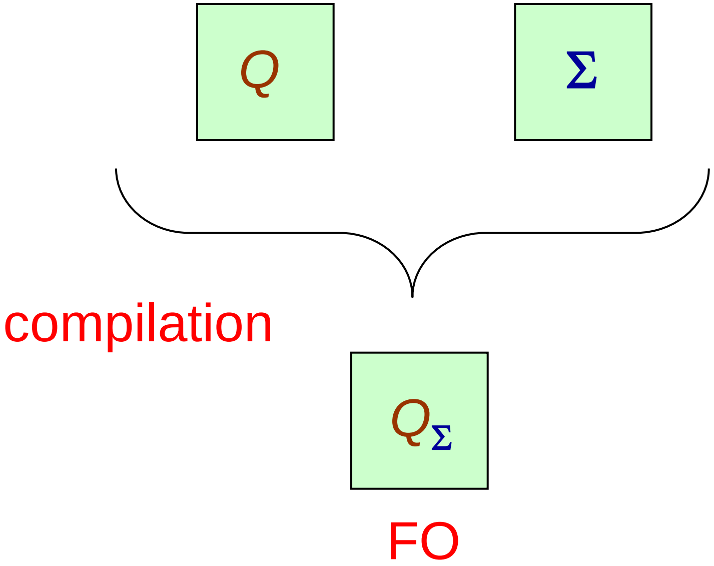
  guard

- Linear TGDs are trivially guarded

- Query answering is in AC_0 in data complexity (first-order rewritability)

  [Calì, G. & Lukasiewicz, PODS 09]

First-Order Rewritable TGDs



Q

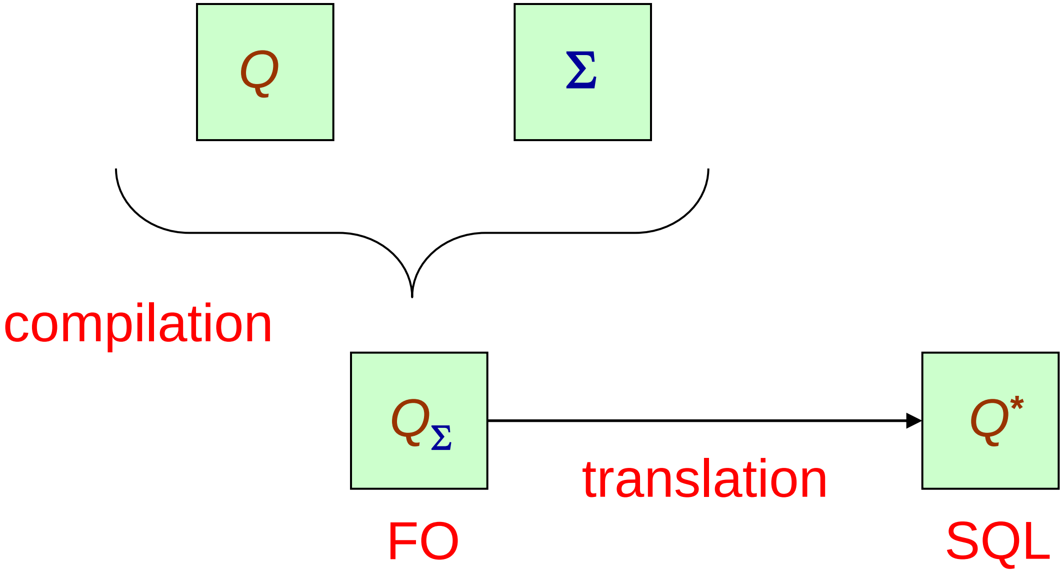
Σ

compilation

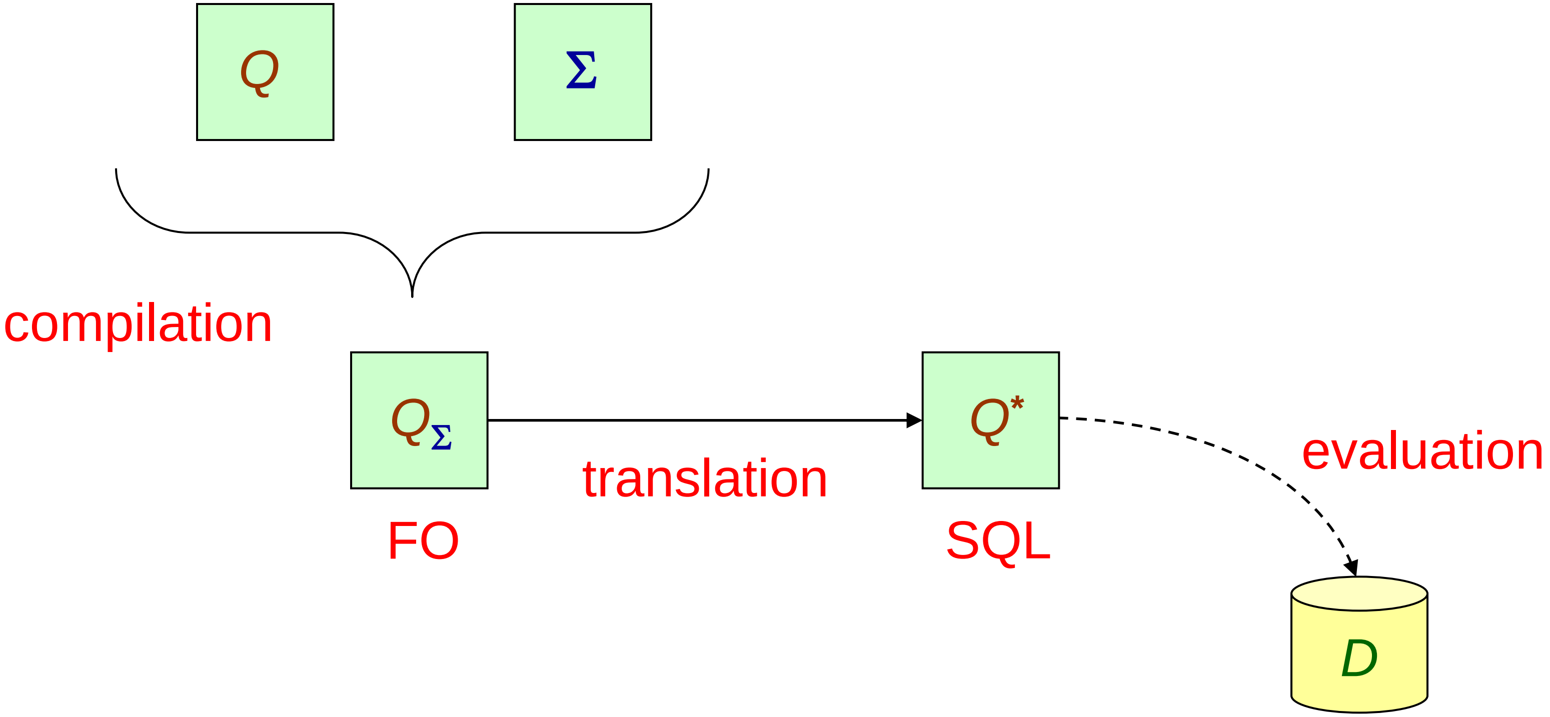
Q_Σ

FO

First-Order Rewritable TGDs



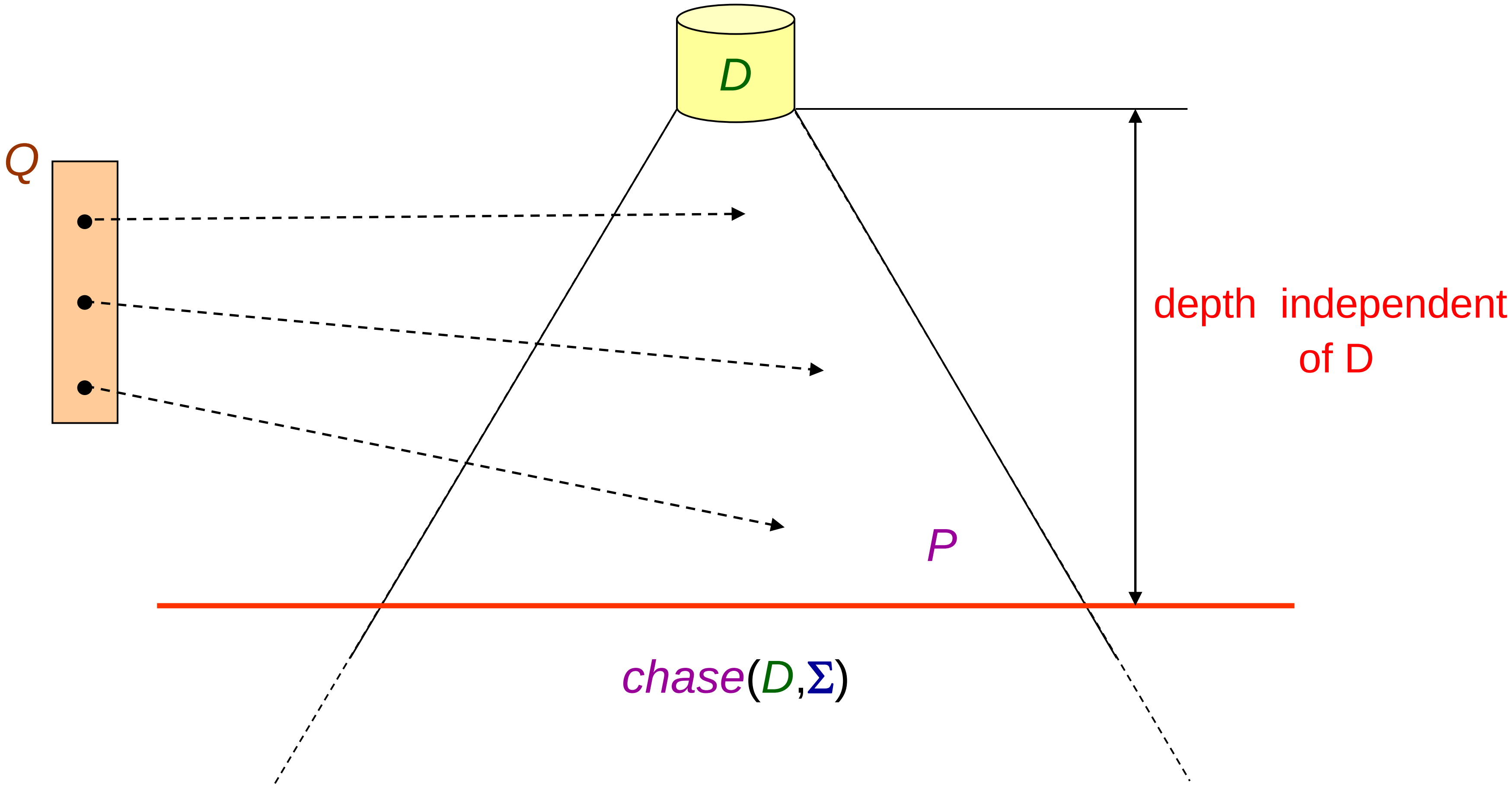
First-Order Rewritable TGDs



$$\forall D\ (D \cup \Sigma \models Q \quad \Leftrightarrow \quad D \models Q^*)$$

Query answering is in AC_0 in data complexity [Vardi, PODS 95]

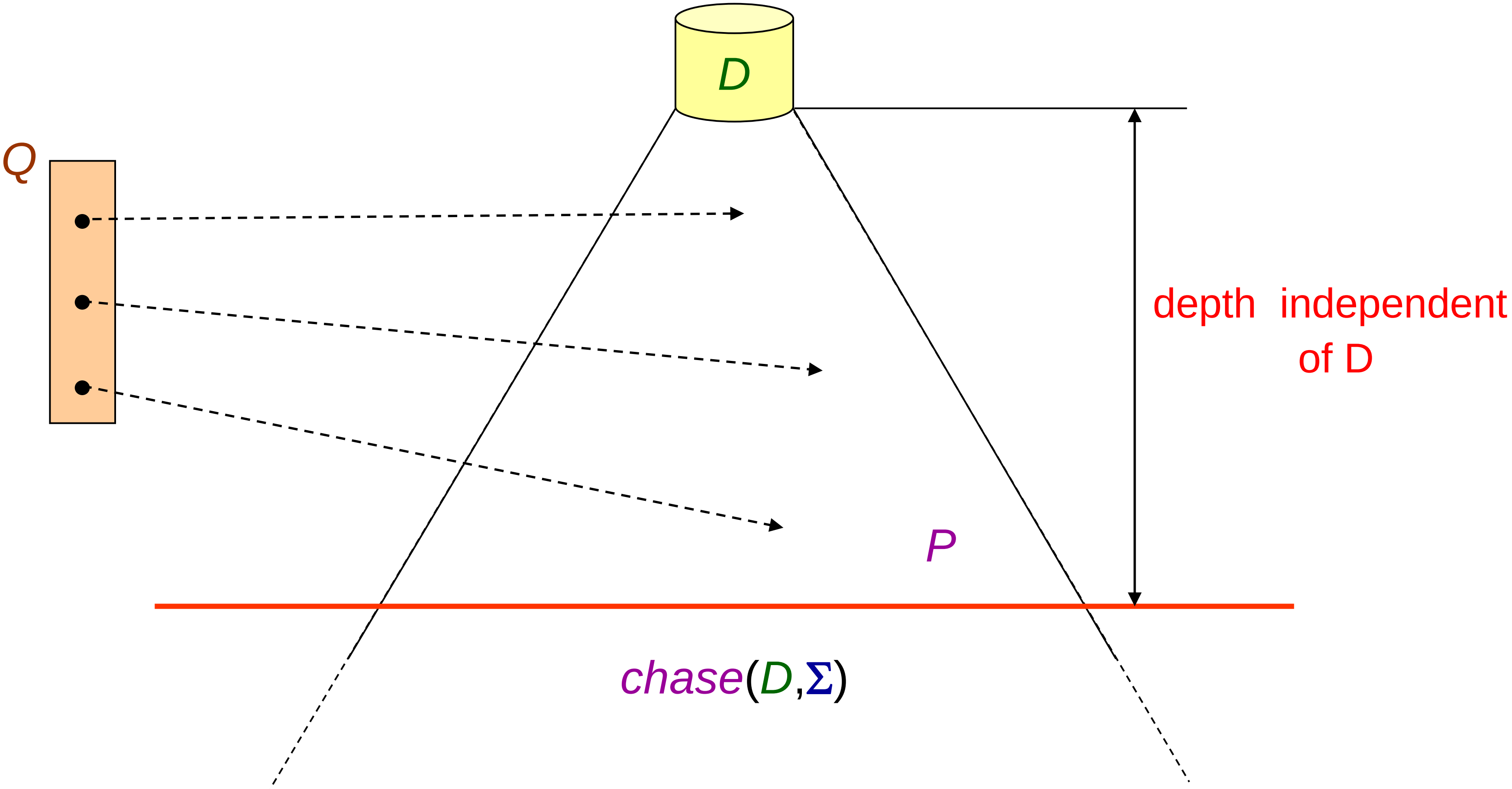
Bounded Derivation-Depth Property (BDDP)



$\mathit{chase}(D,\Sigma) \models Q \quad \Rightarrow \quad P \models Q$

[Calì, G. & Lukasiewicz, PODS 09]

Bounded Derivation-Depth Property (BDDP)



BDDP ⇒ First-Order Rewritability

[Calì, G. & Lukasiewicz, PODS 09]

Linearity

- Just one atom in the body $\forall \mathbf{X} \forall \mathbf{Y}\ \underset{\uparrow}{R(\mathbf{X},\mathbf{Y})} \rightarrow \exists \mathbf{Z}\ \Psi(\mathbf{X},\mathbf{Z})$

  trivially guard

- Linear TGDs are trivially guarded

- Query answering is in AC_0 in data complexity (first-order rewritability)

  [Calì, Gottlob & Lukasiewicz, PODS 09]

- Properly extends DL-Lite (same data complexity)

Ontology Querying

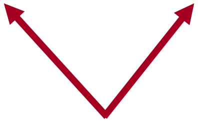
DL-Lite: Popular family of DLs with AC_0 data complexity (OWL 2 QL)
[Calvanese, De Giacomo, Lembo, Lenzerini & Rosati, JAR 07]

DL-Lite TBox	Datalog$^{\pm}$ Representation
$A \sqsubseteq B$	$\forall X\, A(X) \rightarrow B(X)$
$A \sqsubseteq \exists R$	$\forall X\, A(X) \rightarrow \exists Y\, R(X,Y)$
$\exists R \sqsubseteq A$	$\forall X \forall Y\, R(X,Y) \rightarrow A(X)$
$R \sqsubseteq P$	$\forall X \forall Y\, R(X,Y) \rightarrow P(X,Y)$

But…

- What about joins in rule bodies?

$$\forall A \forall D \forall P \; runs(D,P), area(P,A) \rightarrow \exists E \; employee(E,D,P,A)$$



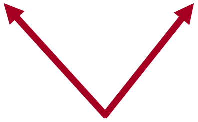
- What about the DL assertion concept product?

$$\forall E \forall M \; elephant(E), mouse(M) \rightarrow biggerThan(E,M)$$

But…

– What about joins in rule bodies?

$$\forall A \forall D \forall P\ runs(D,P), area(P,A) \rightarrow \exists E\ employee(E,D,P,A)$$



– What about the DL assertion concept product?

$$\forall E \forall M\ elephant(E), mouse(M) \rightarrow biggerThan(E,M)$$

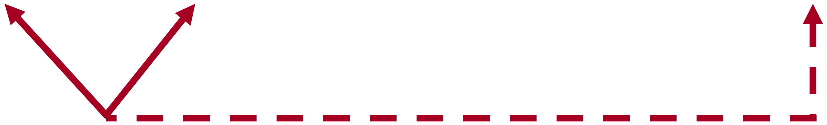
No tree-like models guaranteed

$$\left.\begin{array}{l} \forall X \forall Y\ R(X,Y) \rightarrow \exists Z\ R(Y,Z) \\ \\ \forall X \forall Y\ R(X,Y) \rightarrow S(X) \end{array}\right\} \quad \text{Infinitely many symbols in } S$$

$$\left.\forall X \forall Y\ S(X), S(Y) \rightarrow P(X,Y)\ \right\} \quad P \text{ forms an infinite clique}$$

Stickiness

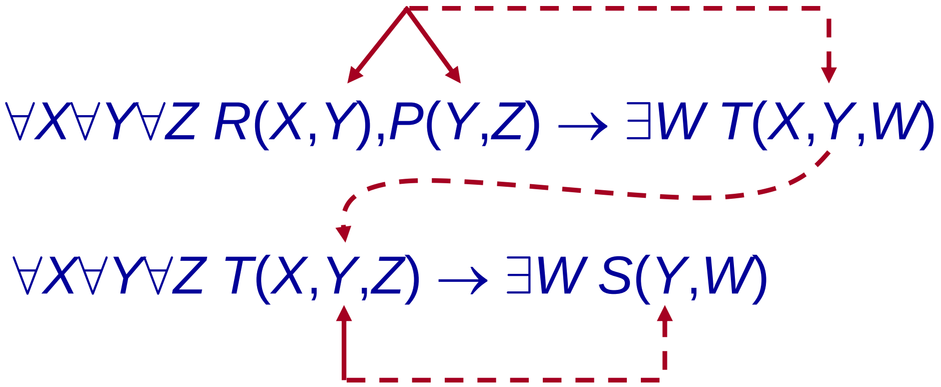
$\forall A \forall D \forall P\ runs(D,P), area(P,A) \rightarrow \exists E\ employee(E,D,P,A)$ ✓



$\forall E \forall M\ elephant(E), mouse(M) \rightarrow biggerThan(E,M)$ ✓

[Calì, G. & Pieris, VLDB 10]

Stickiness



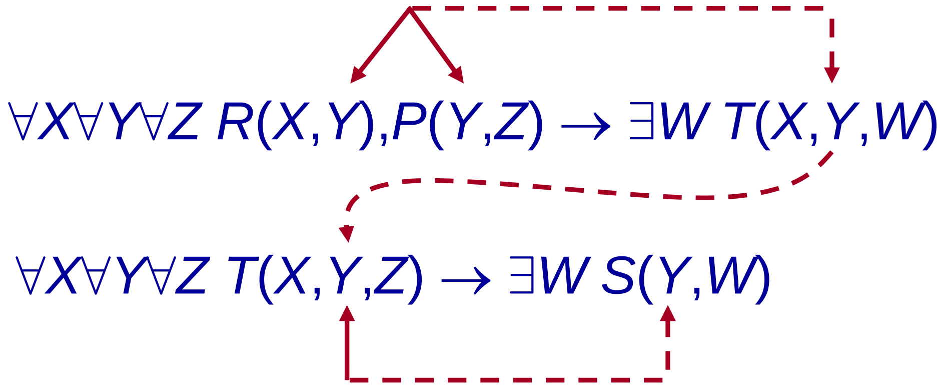
$\forall X \forall Y \forall Z\ R(X,Y), P(Y,Z) \rightarrow \exists W\ T(X,Y,W)$

$\forall X \forall Y \forall Z\ T(X,Y,Z) \rightarrow \exists W\ S(Y,W)$

✓

[Calì, G. & Pieris, VLDB 10]

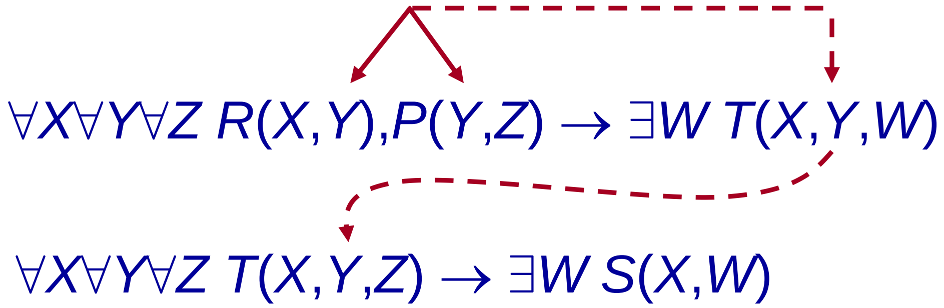
Stickiness



$\forall X \forall Y \forall Z \; R(X,Y),P(Y,Z) \rightarrow \exists W \; T(X,Y,W)$

$\forall X \forall Y \forall Z \; T(X,Y,Z) \rightarrow \exists W \; S(Y,W)$

✓



$\forall X \forall Y \forall Z \; R(X,Y),P(Y,Z) \rightarrow \exists W \; T(X,Y,W)$

$\forall X \forall Y \forall Z \; T(X,Y,Z) \rightarrow \exists W \; S(X,W)$

✗

[Calì, G. & Pieris, VLDB 10]

Stickiness: Marking Procedure

Initial Marking: mark all occurrences of body-variables that do not appear in all head-atoms

$$\forall V \forall W\, R_1(V,W) \;\rightarrow\; \exists X \exists Y \exists Z\, R_2(W,X,Y,Z)$$

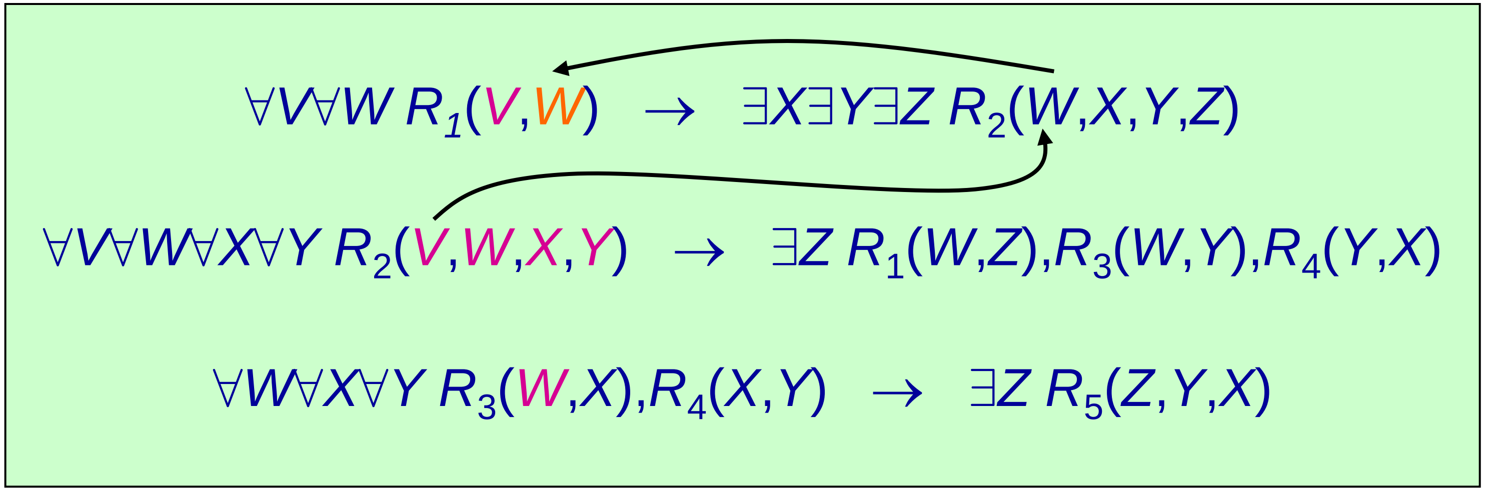
$$\forall V \forall W \forall X \forall Y\, R_2(V,W,X,Y) \;\rightarrow\; \exists Z\, R_1(W,Z),R_3(W,Y),R_4(Y,X)$$

$$\forall W \forall X \forall Y\, R_3(W,X),R_4(X,Y) \;\rightarrow\; \exists Z\, R_5(Z,Y,X)$$

Stickiness: Marking Procedure

Initial Marking: mark all occurrences of body-variables that do not appear in all head-atoms

Propagation Step: propagate the marking to body-variables via the head-atoms



$$\forall V \forall W\, R_1(V,W) \rightarrow \exists X \exists Y \exists Z\, R_2(W,X,Y,Z)$$

$$\forall V \forall W \forall X \forall Y\, R_2(V,W,X,Y) \rightarrow \exists Z\, R_1(W,Z),R_3(W,Y),R_4(Y,X)$$

$$\forall W \forall X \forall Y\, R_3(W,X),R_4(X,Y) \rightarrow \exists Z\, R_5(Z,Y,X)$$

Stickiness

- Marked variables occur just once in the body of each TGD

$$\forall V \forall W \;\; R_1(V,W) \;\rightarrow\; \exists X \exists Y \exists Z \; R_2(W,X,Y,Z)$$

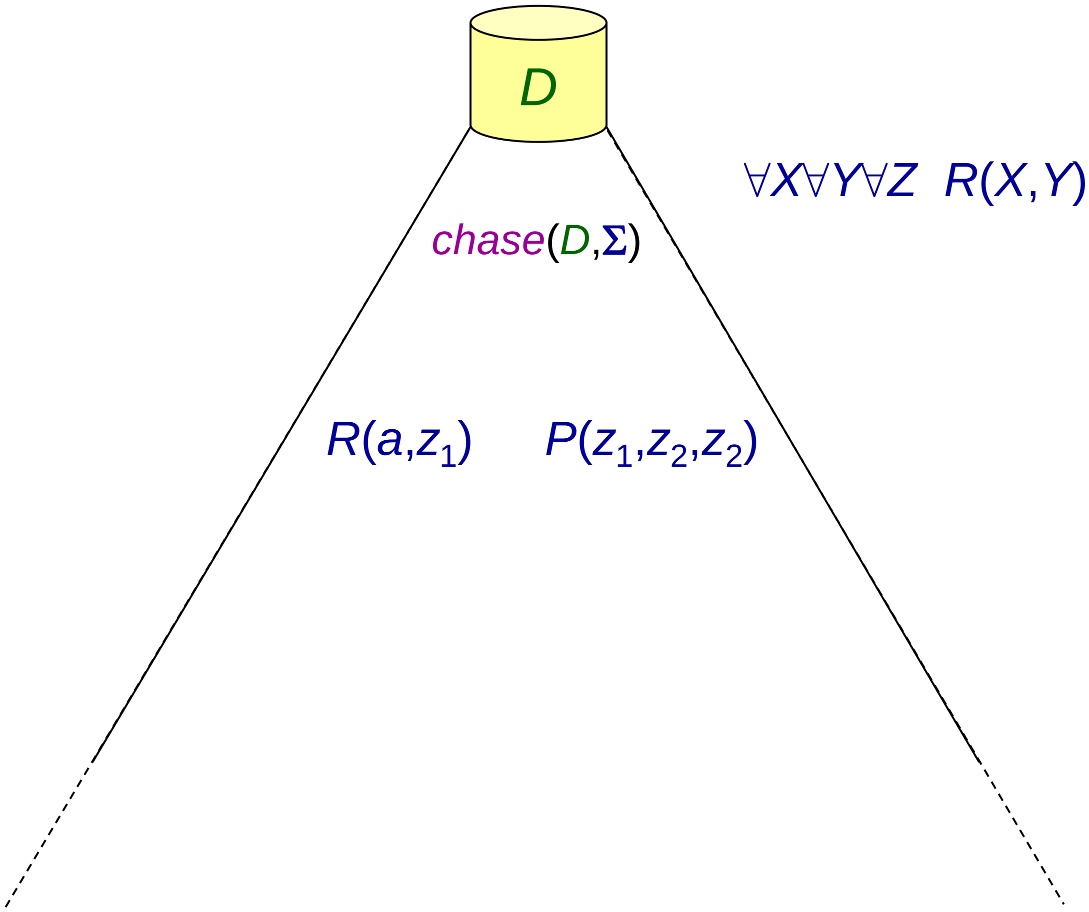
$$\forall V \forall W \forall X \forall Y \;\; R_2(V,W,X,Y) \;\rightarrow\; \exists Z \; R_1(W,Z), R_3(W,Y), R_4(Y,X)$$

$$\forall W \forall X \forall Y \;\; R_3(W,X), R_4(X,Y) \;\rightarrow\; \exists Z \; R_5(Z,Y,X)$$

- The chase has the sticky property (backward-resolution terminates)

Sticky Property

$\forall X \forall Y \forall Z \;\; R(X,Y), P(Y,Z,Z) \rightarrow \exists W \; Q(W,Y,Z)$

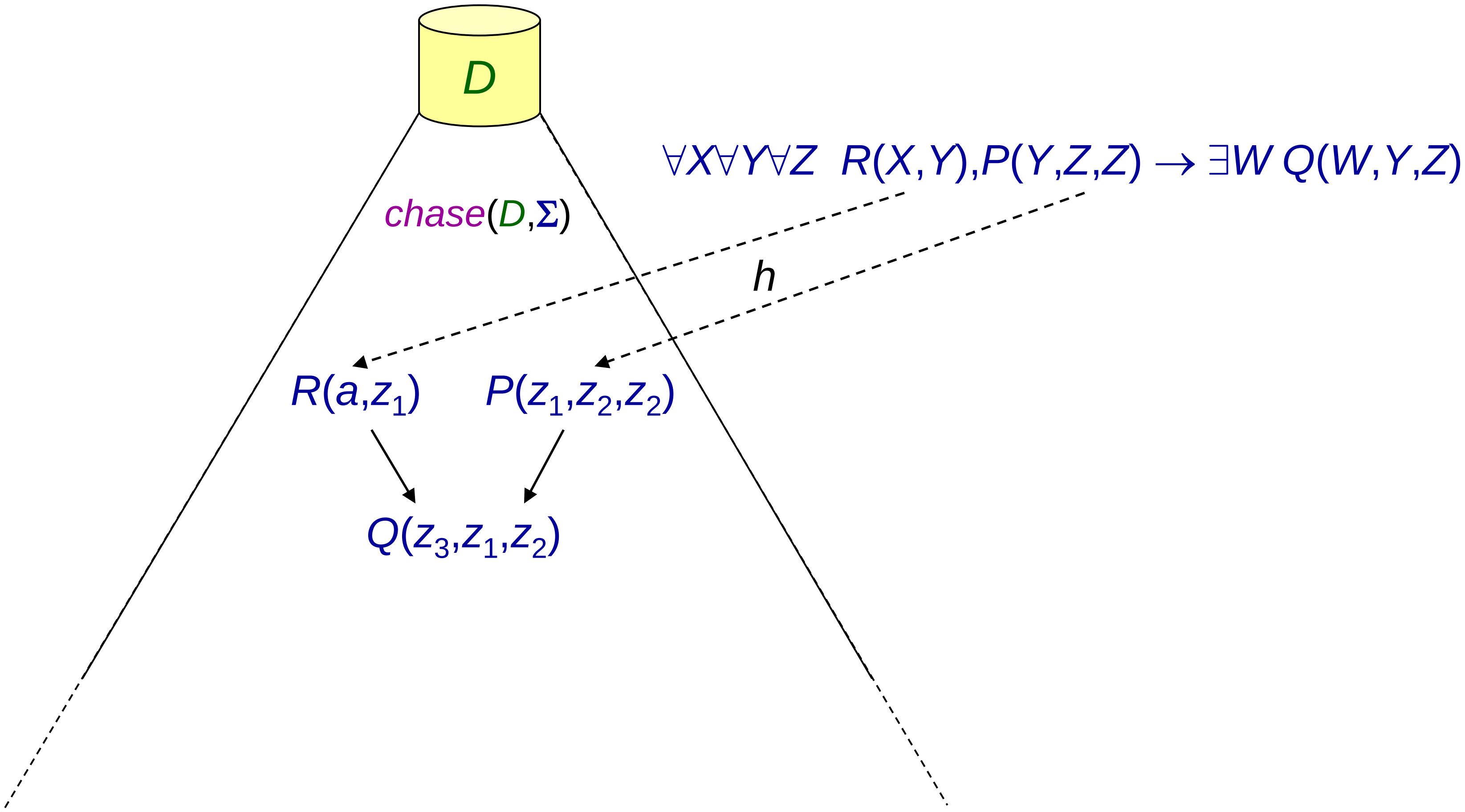


D

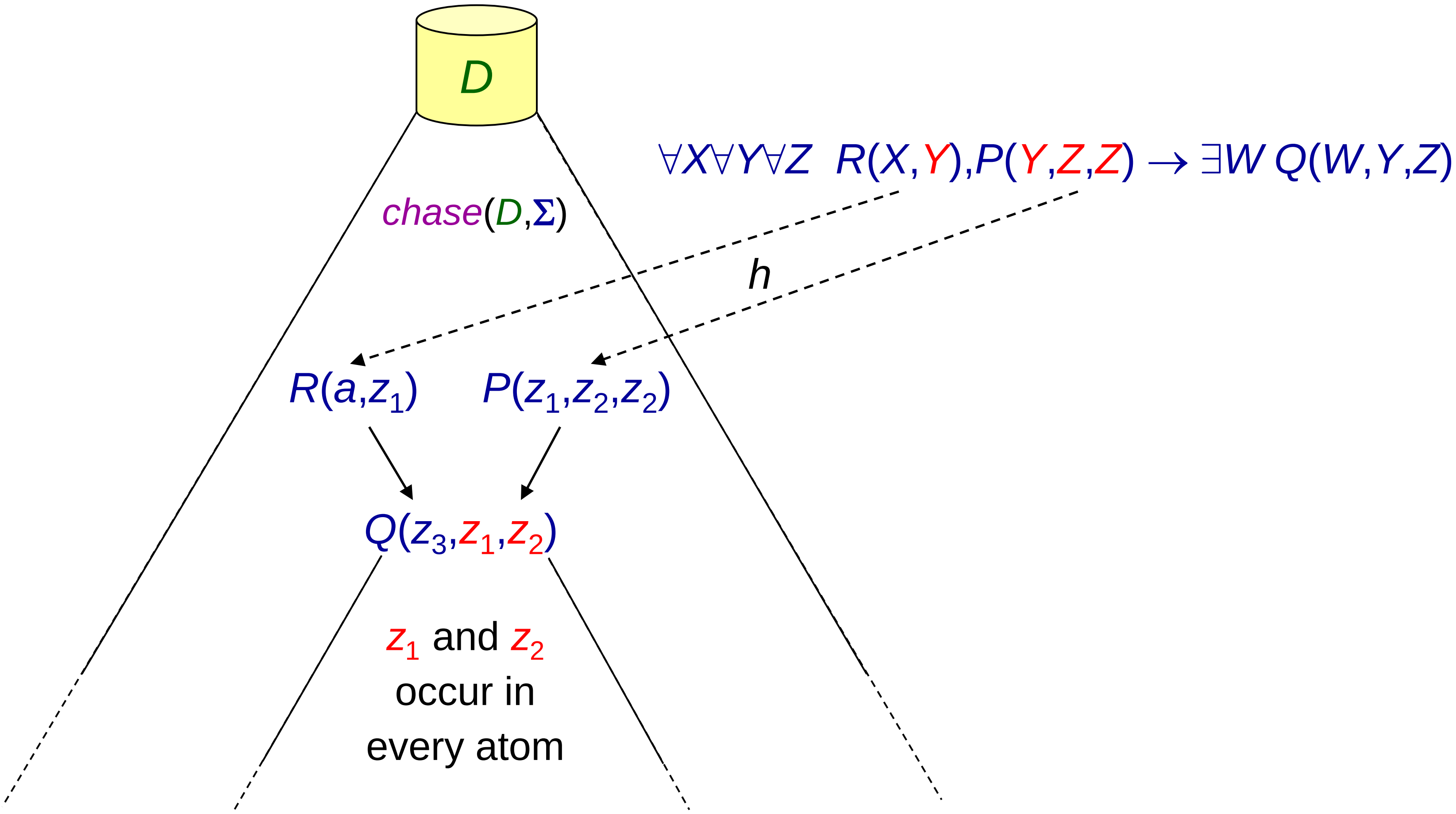
chase(D,$\boldsymbol{\Sigma}$)

$R(a,z_1)$ $\quad$ $P(z_1,z_2,z_2)$

Sticky Property


D
$chase(D,\Sigma)$
$\forall X \forall Y \forall Z \;\; R(X,Y),P(Y,Z,Z) \rightarrow \exists W\; Q(W,Y,Z)$
h
$R(a,z_1)$
$P(z_1,z_2,z_2)$
$Q(z_3,z_1,z_2)$

Sticky Property



$\forall X \forall Y \forall Z \ \ R(X,Y),P(Y,Z,Z) \rightarrow \exists W \, Q(W,Y,Z)$

D

$chase(D,\Sigma)$

h

$R(a,z_1)$

$P(z_1,z_2,z_2)$

$Q(z_3,z_1,z_2)$

z_1 and z_2
occur in
every atom

Stickiness

- Marked variables occur just once in the body of each TGD

$$\forall V \forall W \;\; R_1(V,W) \;\rightarrow\; \exists X \exists Y \exists Z \; R_2(W,X,Y,Z)$$

$$\forall V \forall W \forall X \forall Y \;\; R_2(V,W,X,Y) \;\rightarrow\; \exists Z \; R_1(W,Z), R_3(W,Y), R_4(Y,X)$$

$$\forall W \forall X \forall Y \;\; R_3(W,X), R_4(X,Y) \;\rightarrow\; \exists Z \; R_5(Z,Y,X)$$

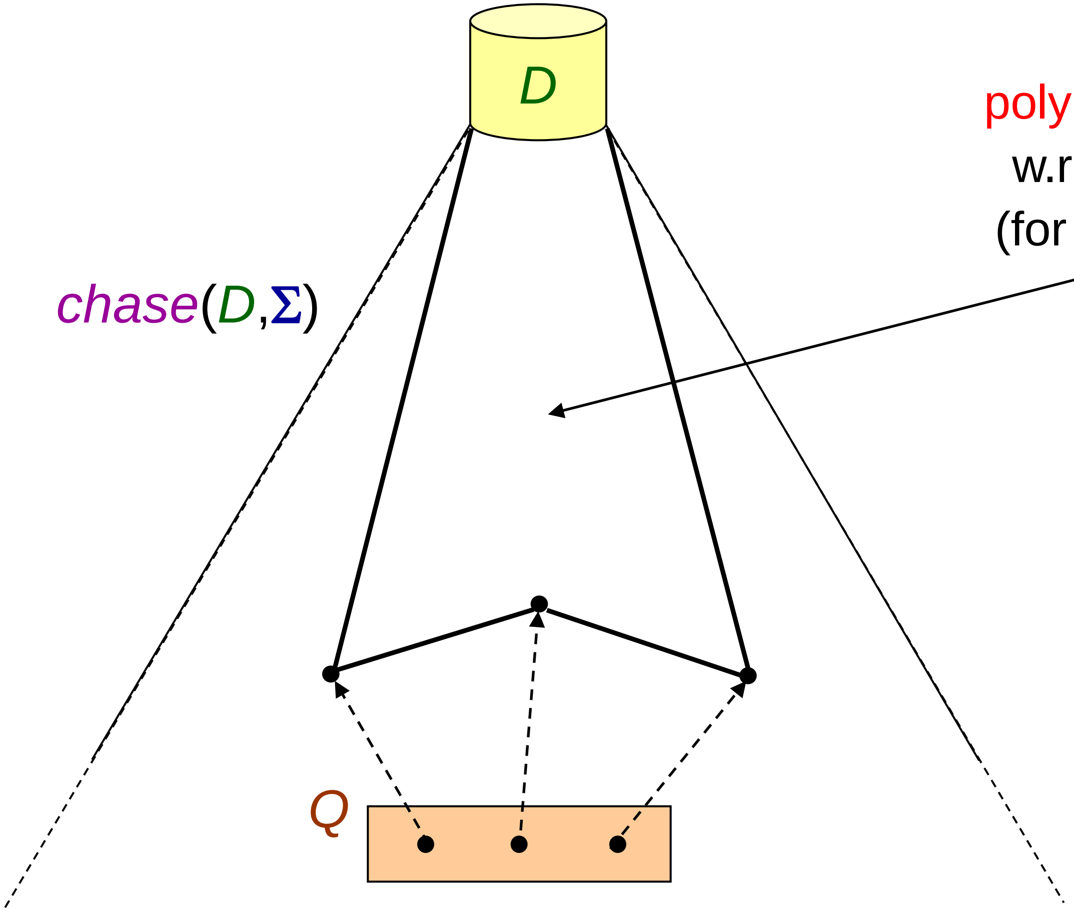
- The chase has the sticky property (backward-resolution terminates)

- Query answering is in AC_0 in data complexity (first-order rewritability)

  [Calì, G. & Pieris, VLDB 10]

- Properly extends DL-Lite (same data complexity)

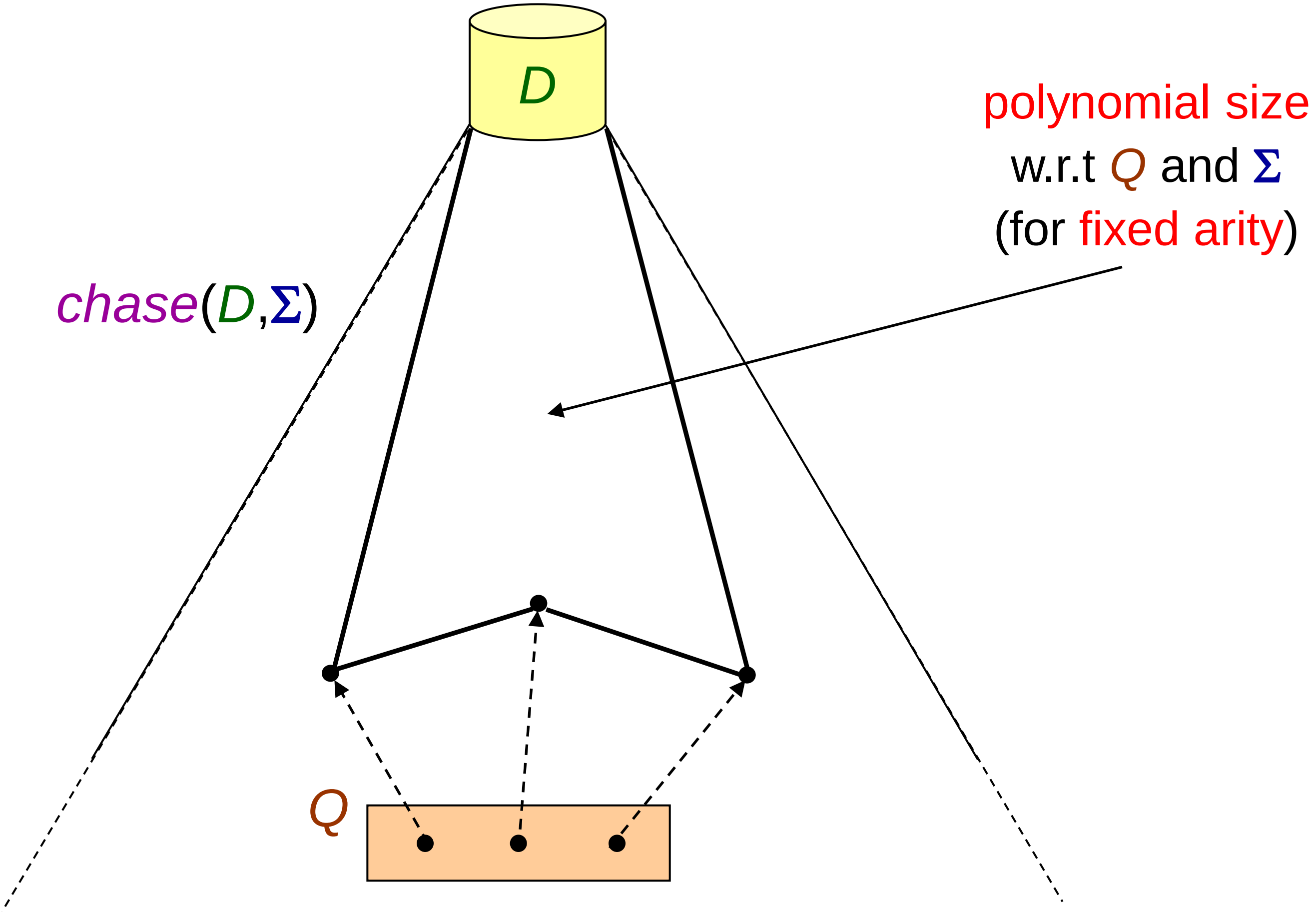
Polynomial Witness Property (PWP)



polynomial size
w.r.t *Q* and Σ
(for fixed arity)

[G. & Schwentick, KR 12]

Polynomial Witness Property (PWP)

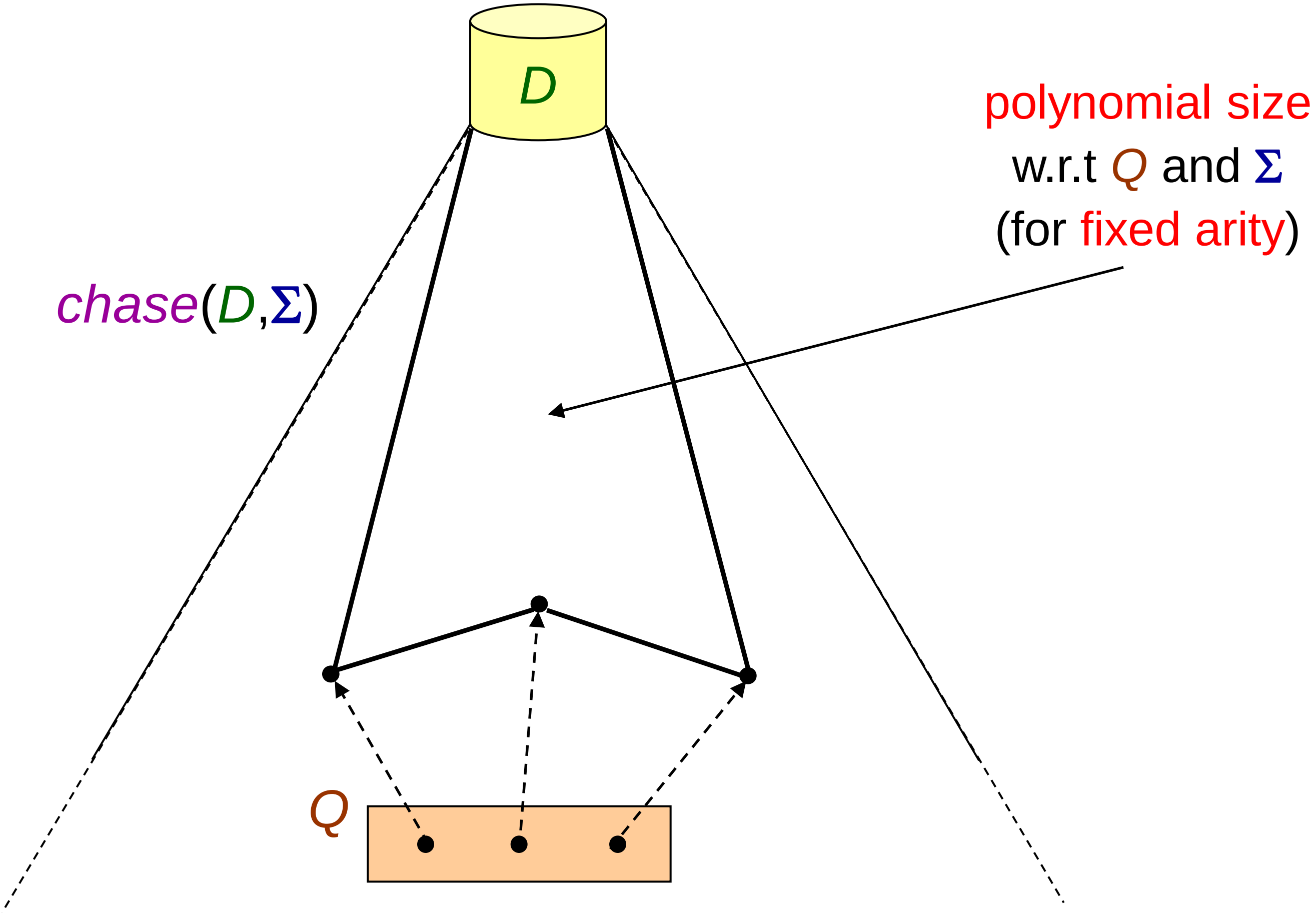


polynomial size
w.r.t *Q* and Σ
(for fixed arity)

PWP ⇒ non-recursive Datalog rewriting of polynomial size

[G. & Schwentick, KR 12]

Polynomial Witness Property (PWP)



Holds for the first-order rewritable Datalog$^{\pm}$ fragments (linear and sticky)

[G. & Schwentick, KR 12]

Finite Controllability

$$D \cup \Sigma \models Q \quad \overset{?}{\Leftrightarrow} \quad D \cup \Sigma \models_{\text{fin}} Q$$

- Holds for inclusion dependencies

  [Rosati, PODS 06]

- Holds for guarded TGDs (in fact, for the guarded fragment)

  [Bárány, Gottlob & Otto, LICS 10]

- Holds for sticky TGDs

  [Gogacz & Marcinkowski, → next talk]

Additional Features

- EGDs, e.g., $\forall X \forall Y \forall Z\ \mathit{reports}(X,Y),\mathit{reports}(X,Z) \rightarrow Y = Z$

  Non-Conflicting EGDs: do not interact with TGDs

  Preliminary check without adding complexity

- Negative constraints, e.g., $\forall X\ \mathit{emp}(X),\mathit{customer}(X) \rightarrow \bot$

  Check without adding complexity

Additional Features

- EGDs, e.g., $\forall X \forall Y \forall Z\ reports(X,Y), reports(X,Z) \rightarrow Y = Z$

  Non-Conflicting EGDs: do not interact with TGDs

  Preliminary check without adding complexity

- Negative constraints, e.g., $\forall X\ emp(X), customer(X) \rightarrow \bot$

  Check without adding complexity

Finite controllability does not hold

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

$$\Sigma = \left\{ \begin{array}{l} \forall X \forall Y\ R(X,Y) \rightarrow \exists Z\ R(Y,Z) \\ \forall X \forall Y \forall Z\ R(Y,X), R(Z,X) \rightarrow Y = Z \end{array} \right\}$$

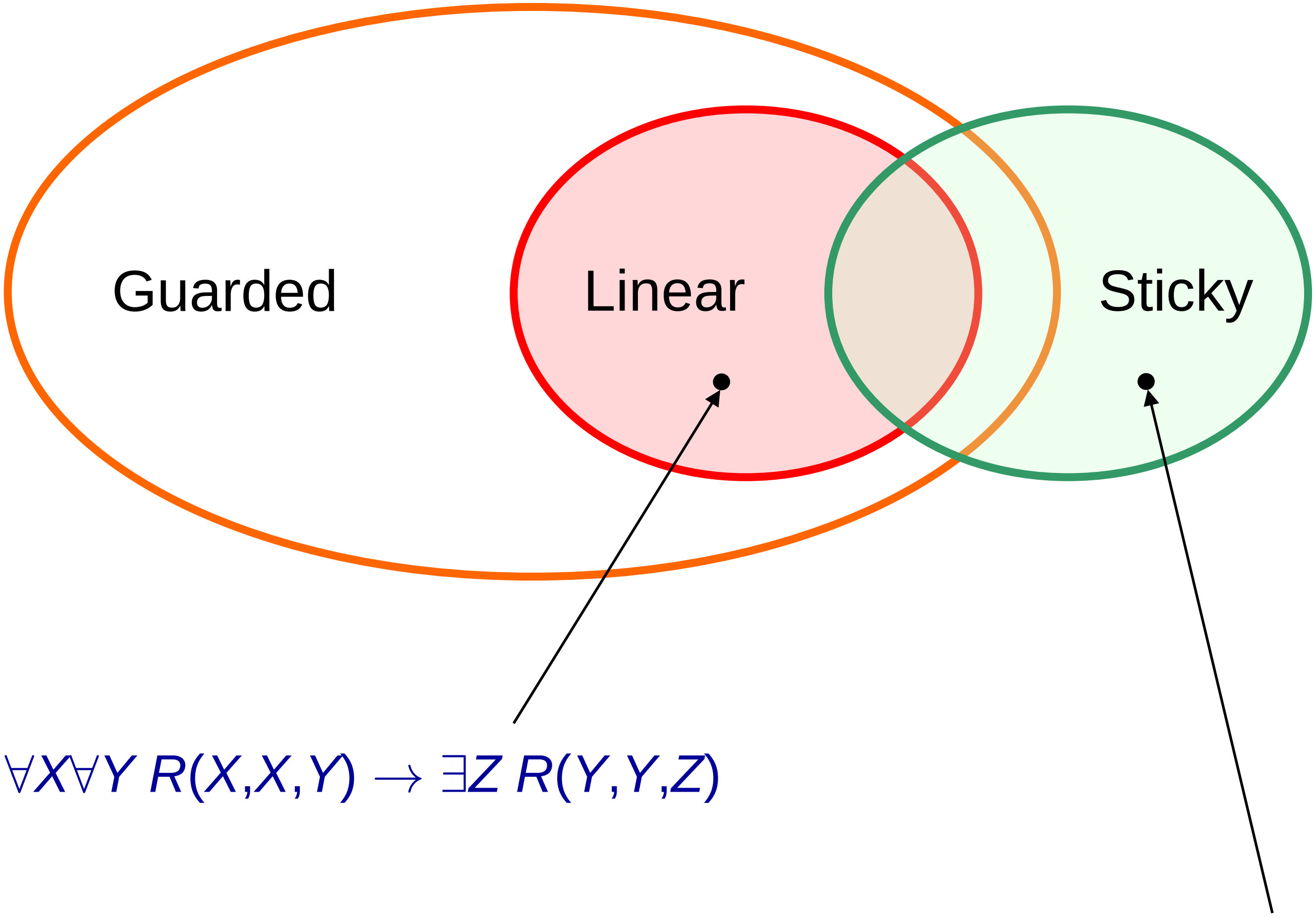
$Q \leftarrow R(A,a)$

$D \cup \Sigma \not\models Q$

but

$D \cup \Sigma \models_{\text{fin}} Q$

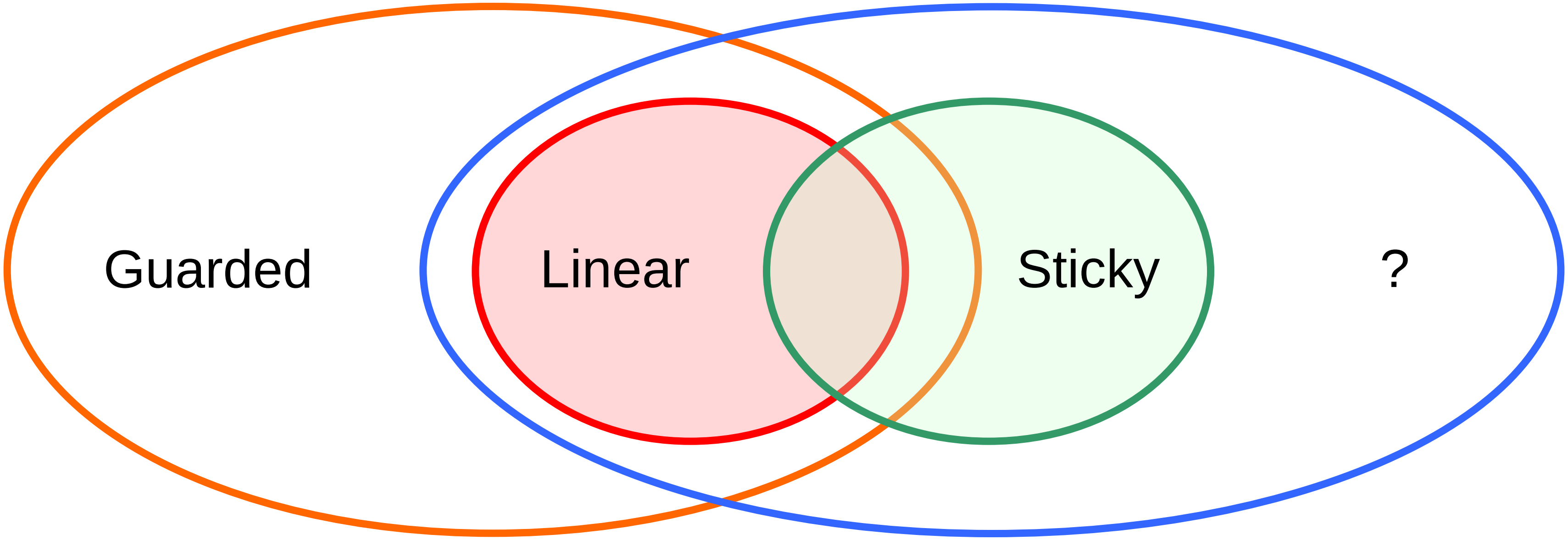
Datalog$^{\pm}$: Overview



$\forall X \forall Y\, R(X,X,Y) \rightarrow \exists Z\, R(Y,Y,Z)$

$\forall X \forall Y \forall Z \forall W\, S(X,Y),R(Z,W) \rightarrow \exists V\, S(Y,V),R(W,V)$

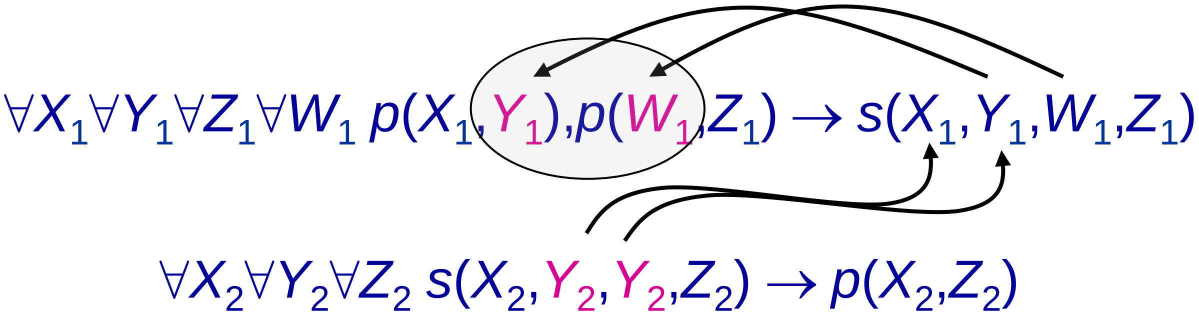
Datalog$^{\pm}$: Overview



without losing first-order rewritability and PWP

Sticky-Join TGDs

- Marked variables can appear more than once in a single atom
- Providing that the following does not happen:



$$\forall X_1 \forall Y_1 \forall Z_1 \forall W_1\; p(X_1,Y_1),p(W_1,Z_1) \rightarrow s(X_1,Y_1,W_1,Z_1)$$

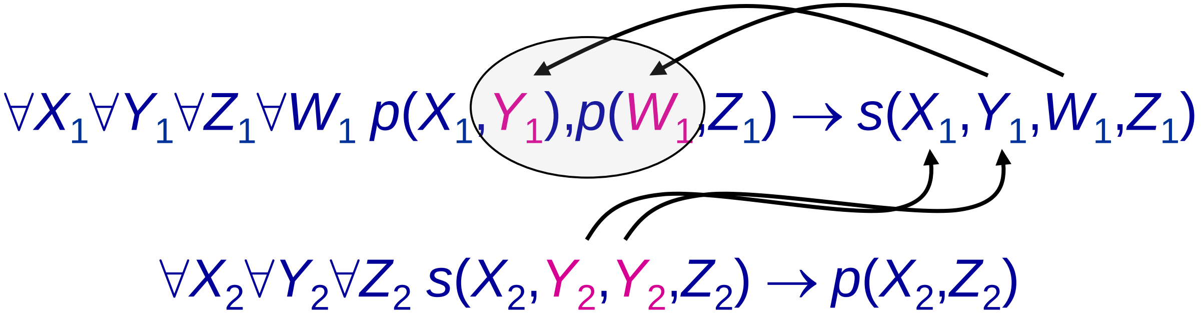
$$\forall X_2 \forall Y_2 \forall Z_2\; s(X_2,Y_2,Y_2,Z_2) \rightarrow p(X_2,Z_2)$$

Y_1 and W_1 are in join

Sticky-Join TGDs

- Marked variables can appear more than once in a single atom

- Providing that the following does not happen:



$\forall X_1 \forall Y_1 \forall Z_1 \forall W_1\; p(X_1,Y_1),p(W_1,Z_1) \rightarrow s(X_1,Y_1,W_1,Z_1)$

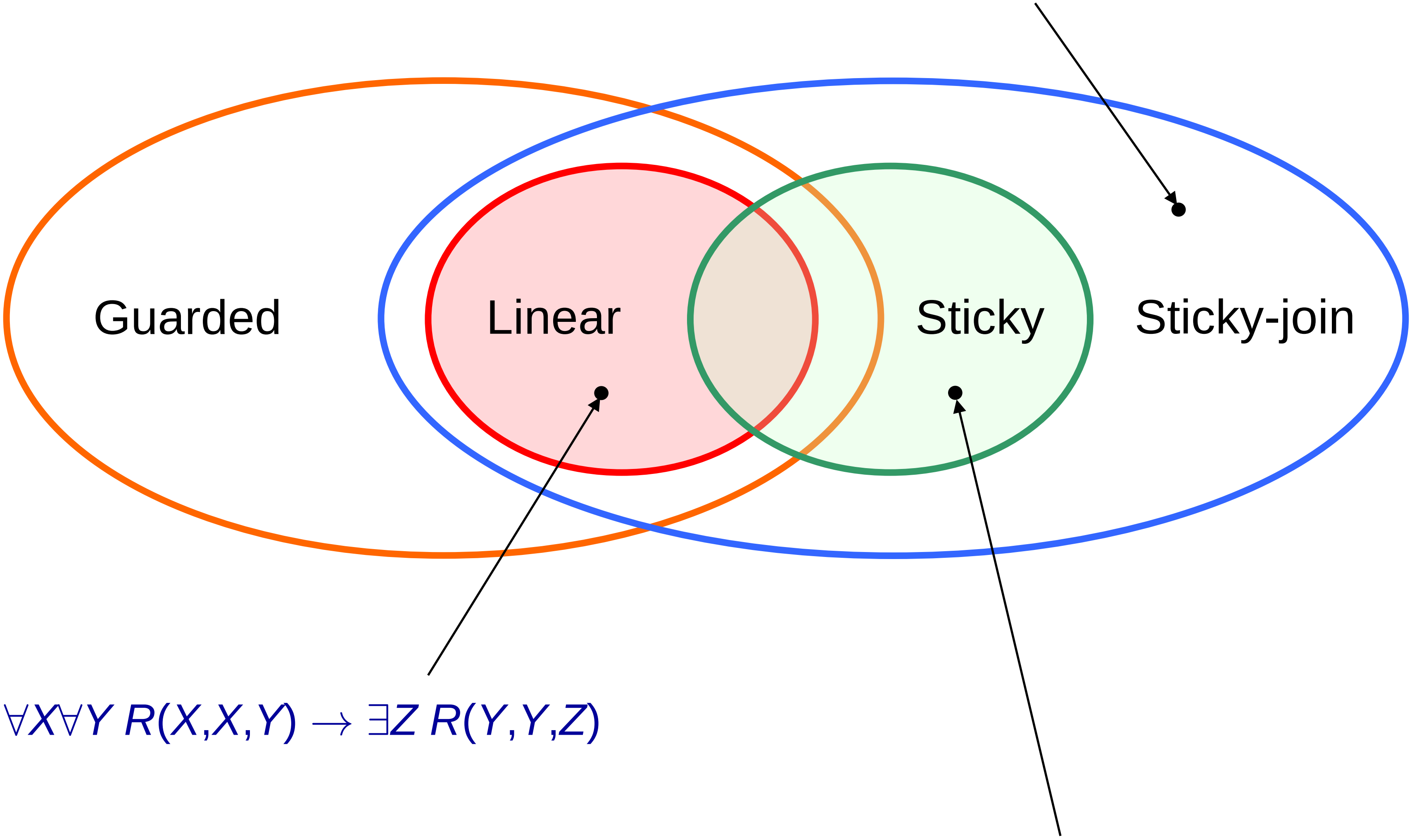
Y_1 and W_1 are in join

$\forall X_2 \forall Y_2 \forall Z_2\; s(X_2,Y_2,Y_2,Z_2) \rightarrow p(X_2,Z_2)$

- Same complexity as sticky TGDs and enjoy PWP

- But harder to identify them - PSPACE-complete

Datalog$^{\pm}$: Overview

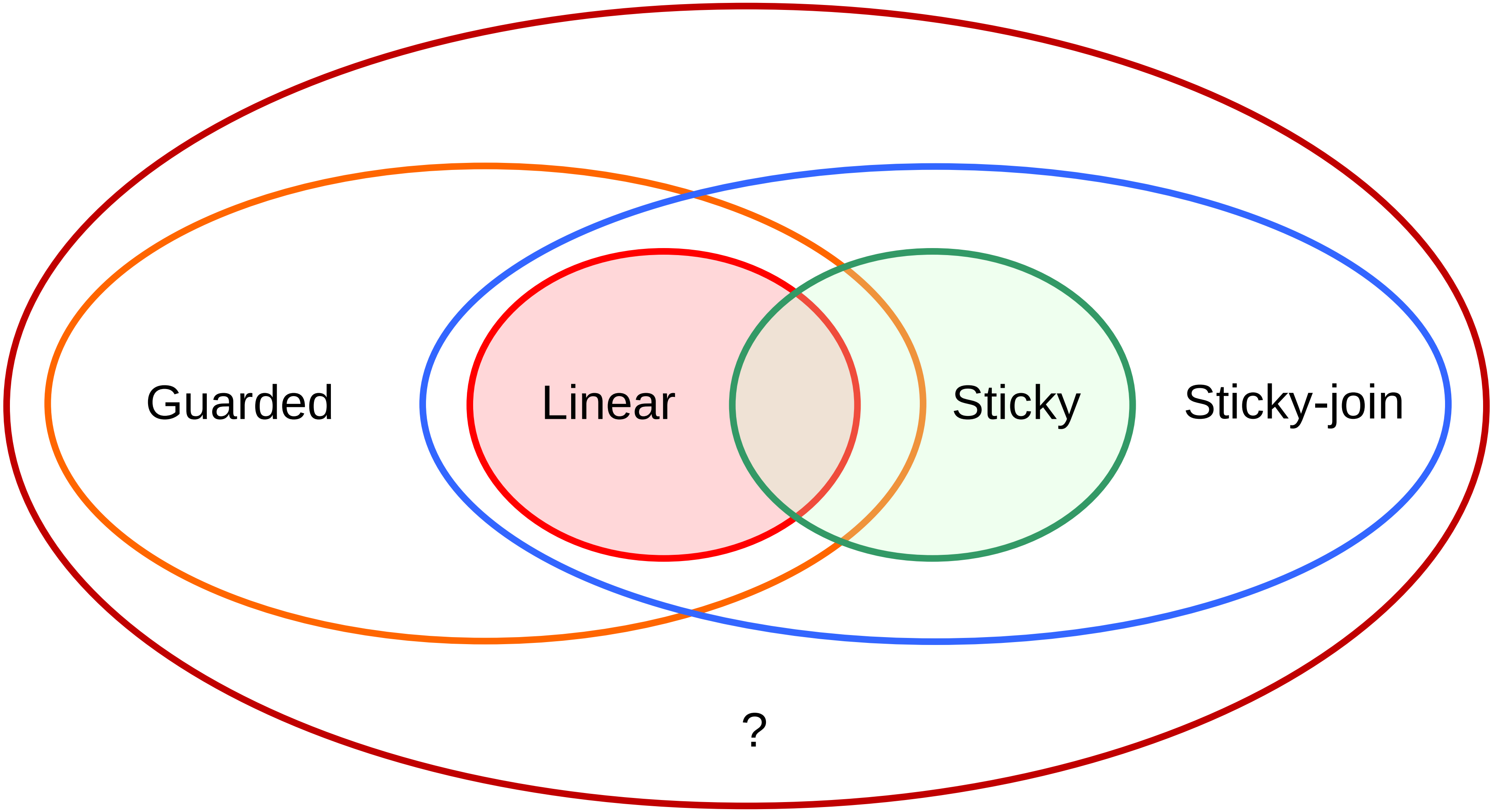


$\forall X \forall Y \forall Z\ R(X,Y), S(Y,Z,Z) \rightarrow \exists W\ P(Y,W)$

$\forall X \forall Y\ R(X,X,Y) \rightarrow \exists Z\ R(Y,Y,Z)$

$\forall X \forall Y \forall Z\ S(X,Y), R(Y,Z) \rightarrow \exists W\ P(Y,W)$

Datalog$^{\pm}$: Overview



without losing PTIME data complexity

Tame TGDs (TTGDs)

- Check that each rule is guarded* or sticky-join (and thus sticky)

$\forall X \forall Y \forall Z\ R(X,Y) \rightarrow \exists Z\ R(Y,Z)$	sticky-join and guarded*
$\forall X \forall Y\ T(X), R(X,Y) \rightarrow T(Y)$	Guarded* not sticky-join
$\forall X \forall Y\ T(X), T(Y) \rightarrow \exists Z\ P(X,Y,Z)$	sticky-join not guarded

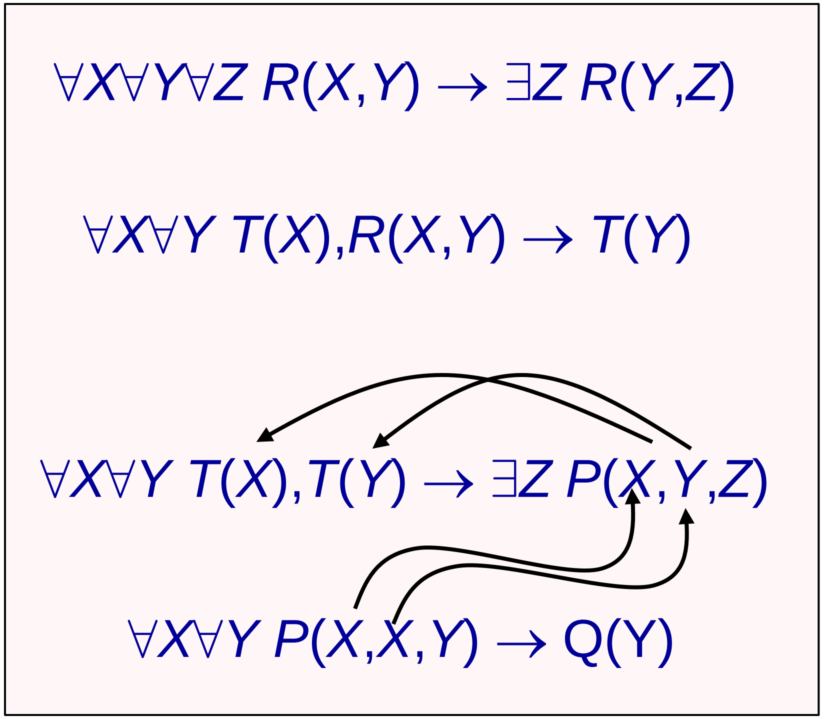
- Not first-order rewritable due to the second rule

- Chase has infinite treewidth since P forms an infinite clique

*) *Predicates in heads of non-guarded rules are unguarded. Atoms whose predicate is unguarded are not allowed to serve as guards!*

Tame TGDs

– The following set of TGDs is not tame



$\forall X \forall Y \forall Z\; R(X,Y) \rightarrow \exists Z\; R(Y,Z)$ sticky-join and guarded*

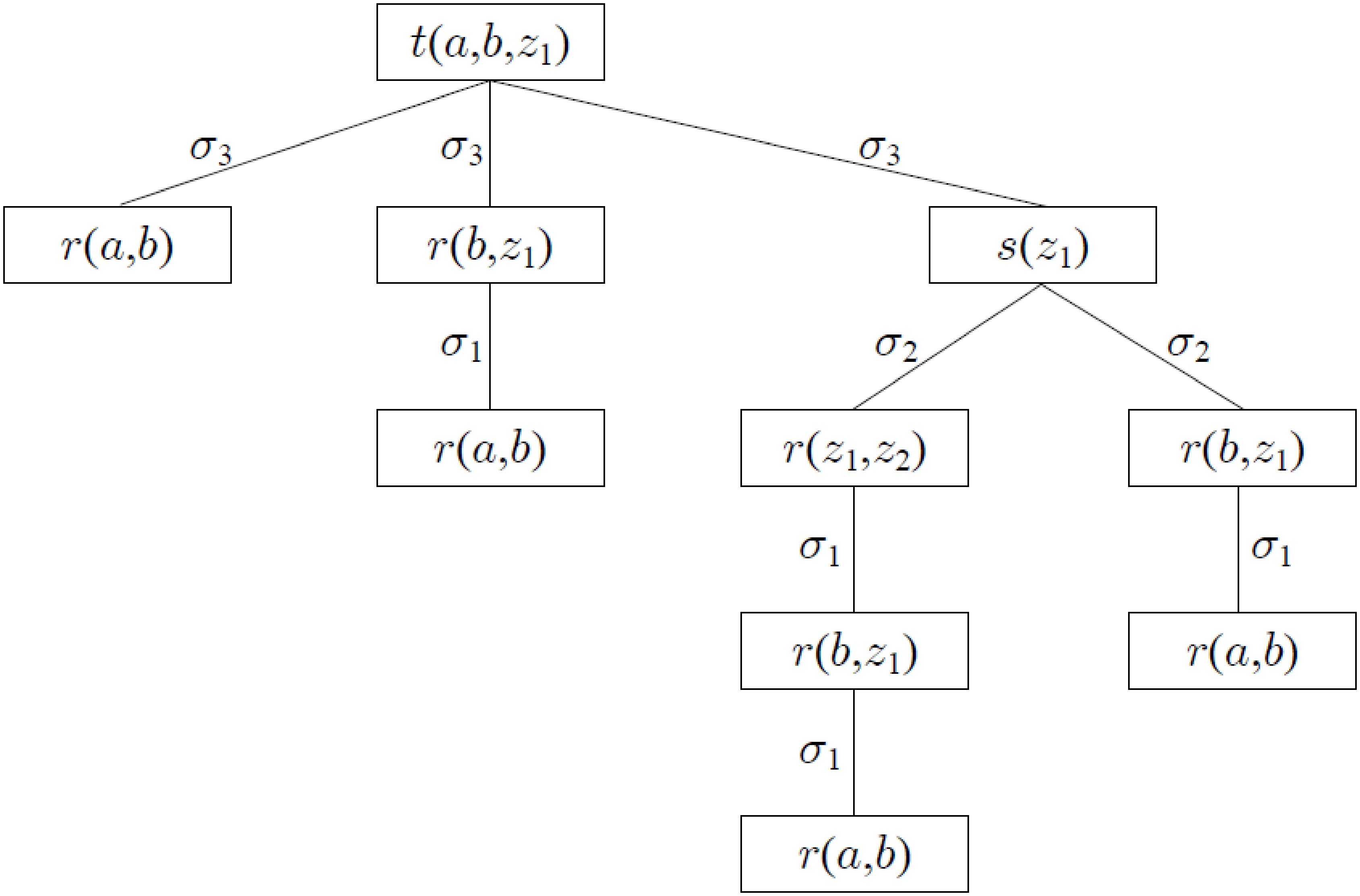
$\forall X \forall Y\; T(X), R(X,Y) \rightarrow T(Y)$ Guarded* not sticky-join

$\forall X \forall Y\; T(X), T(Y) \rightarrow \exists Z\; P(X,Y,Z)$ neither sticky-join nor guarded

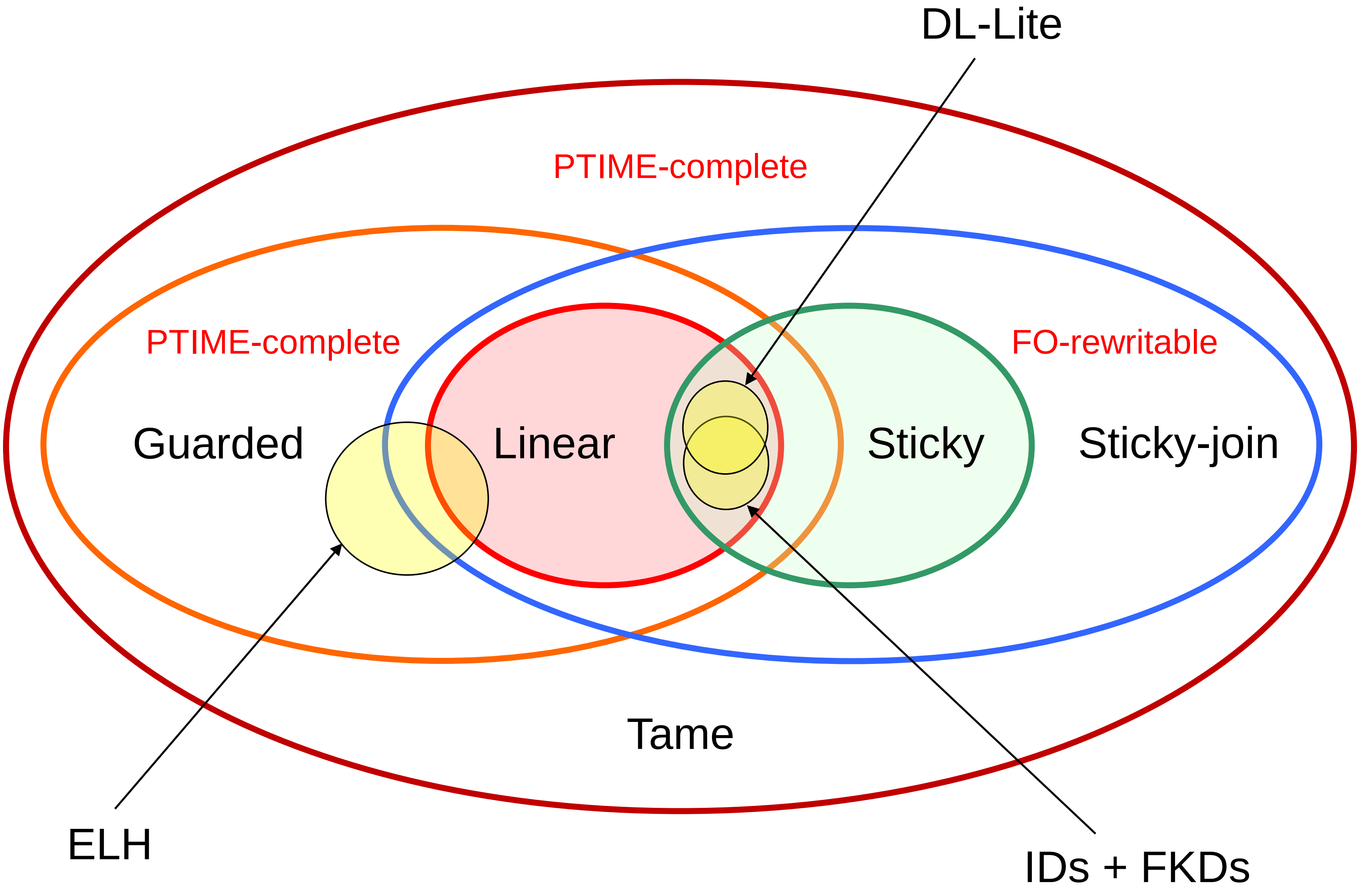
$\forall X \forall Y\; P(X,X,Y) \rightarrow Q(Y)$ sticky-join and guarded

*) *Predicates in heads of non-guarded rules are unguarded. Atoms whose predicate is unguarded are not allowed to serve as guards!*

No bounded tw model, but bounded tw resolution proof scheme!



Datalog$^{\pm}$: Overview



Datalog$^{\pm}$: Summary of Complexity Results

	Data	Fixed Σ	Combined
Guarded	PTIME	NP	2EXPTIME
Linear	in AC_0	NP	PSPACE
Sticky	in AC_0	NP	EXPTIME
Sticky-Join	in AC_0	NP	EXPTIME
Tame	PTIME	NP	2EXPTIME

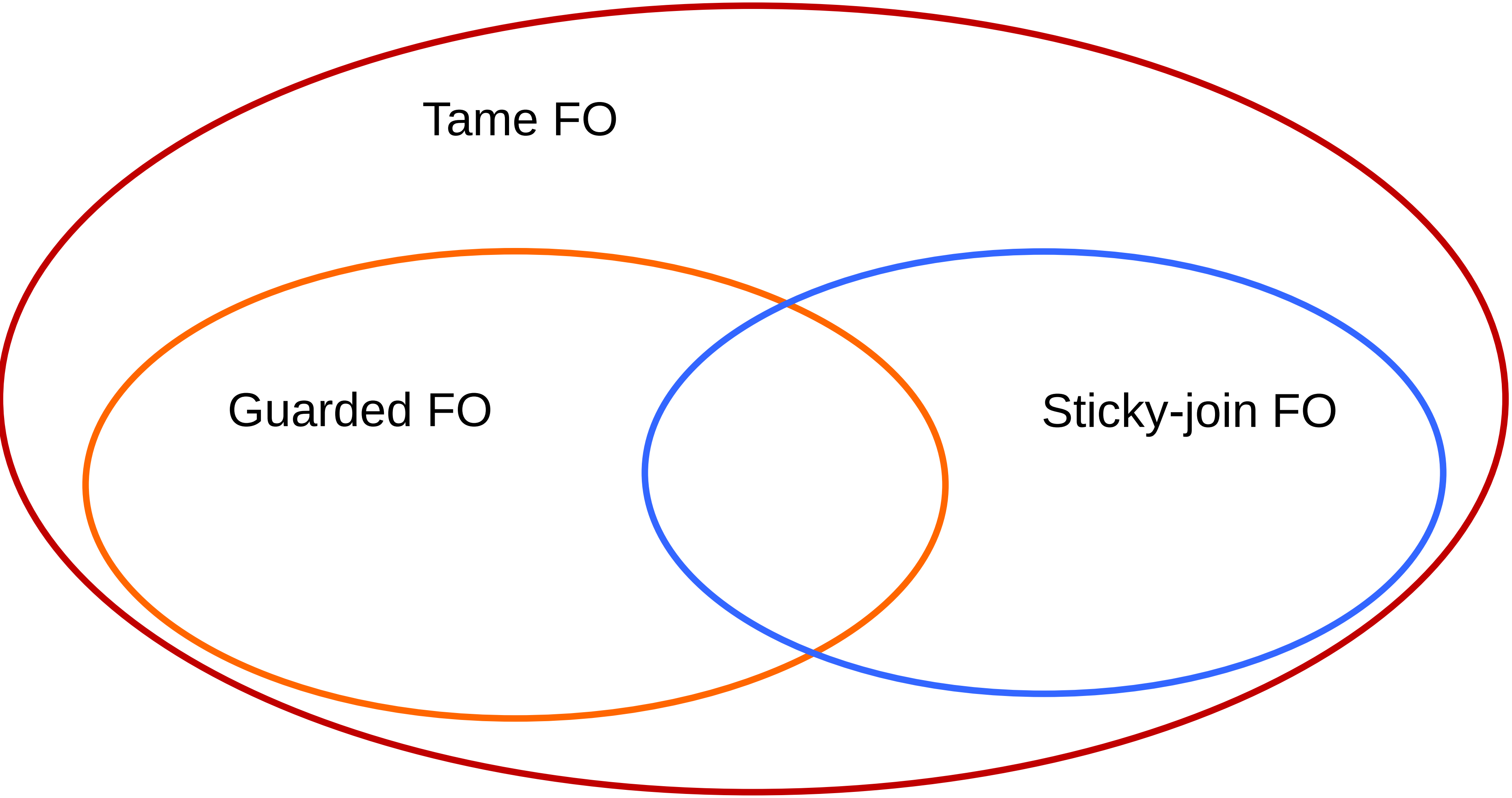
Same complexity with negative constraints and non-conflicting EGDs

current work

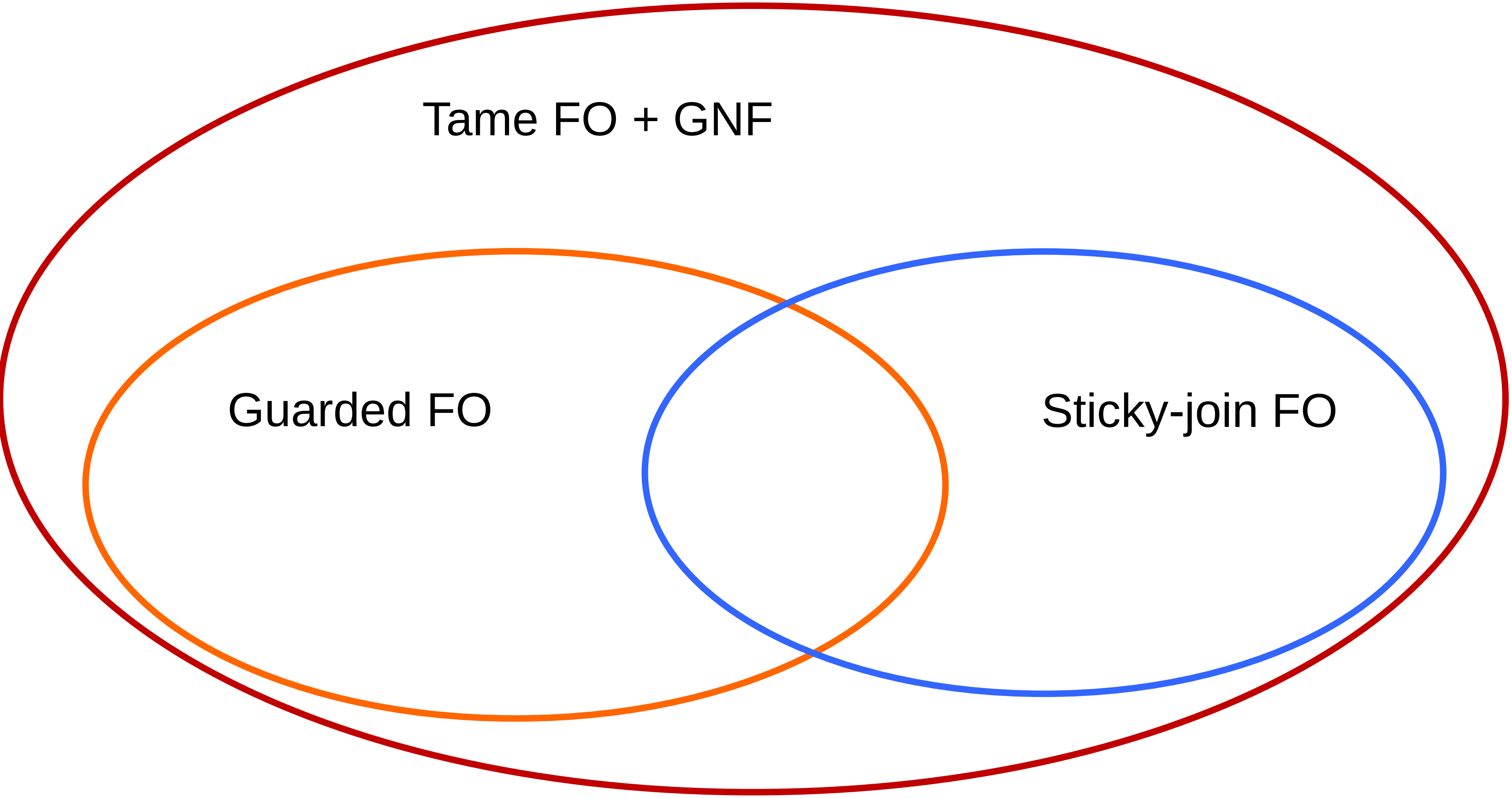
Finite controllability of Tamed TGDs:

Reduction to Sticky TGDs

current work



current work



Thank you!