Ontology-Based Query Answering

with **Existential Rules**

Marie-Laure Mugnier

University of Montpellier

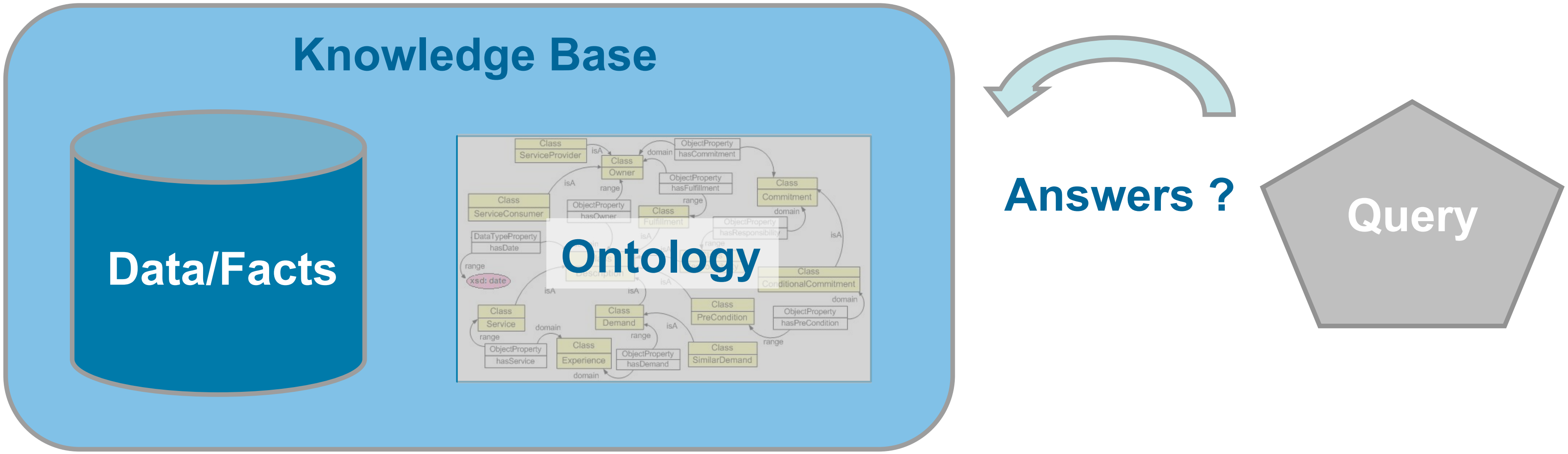
Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier

LIRMM

Inria

CrEDIBLE Workshop, 4/10/2013

Ontology-based Query Answering



Patient records

Medical ontology

« Patient P suffers from leukemia and
has been prescribed drug D against hypertension »

Query: « find all cancer patients treated for high-blood pressure »

→ Use ontological knowledge to infer all deducible answers

Data / Facts

Relational Database

parentOf

A	B
A	C
C	*x*
…	…

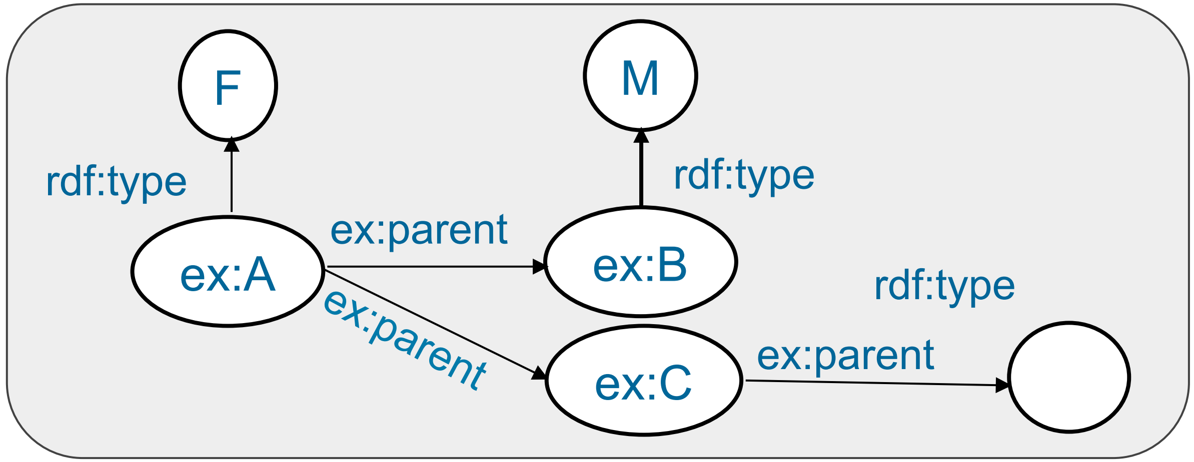
Male

B
…

Fem.

A
…

RDF (Semantic Web)



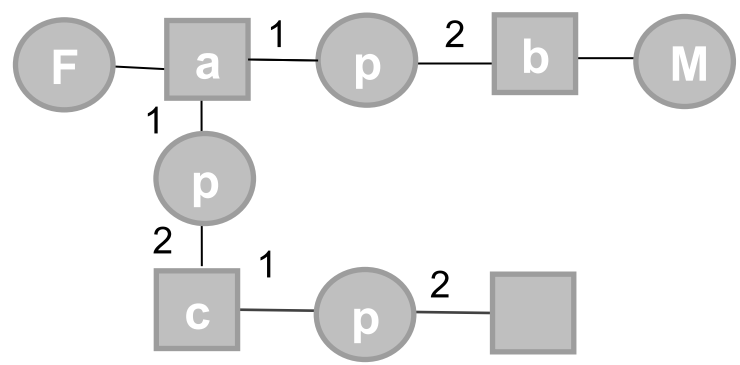
Etc.

Abstraction in First-Order Logic

$$\exists x(\ parentOf(A,B) \wedge parentOf(A,C) \wedge parentOf(C,x) \wedge F(A) \wedge M(B)\ )$$

We generalize here the classical notion of a fact by allowing existential variables

Or in graphs / hypergraphs



Conjunctive Queries (CQ)

« Find all x *such that x is a female and has a child who is a female »*

∃y (Female(x) ∧ childOf(x, y) ∧ Female(y))	First-Order Logical formula
ans(x) ← Female(x), childOf(x, y), Female(y)	Datalog notation
SELECT … FROM … WHERE …	SQL
SELECT … WHERE <Graph Pattern>	SPARQL

Easy generalization to a union (OR) of conjunctive queries

Existential Rules

$$\forall X\ \forall Y\ (\ \underbrace{B[X, Y]}_{\text{Body}} \rightarrow \exists Z\ \underbrace{H[X, Z]}_{\text{Head}}\ )$$

X, Y, Z : *tuples of variables*

Any conjunction of atoms (on variables and constants)

$\forall x\ \forall y\ (siblingOf(x,y) \rightarrow \exists z\ (parentOf(z,x) \wedge parentOf(z,y)))$

Simplified form: $siblingOf(x,y) \rightarrow parentOf(z,x) \wedge parentOf(z,y)$

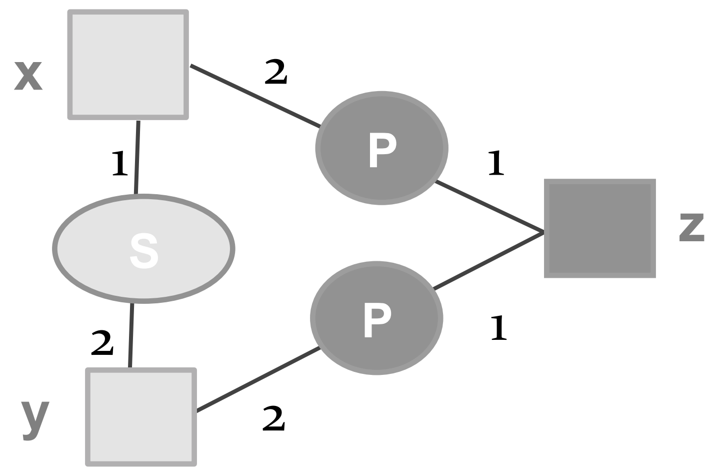
- See also Datalog +/- [Cali Gottlob Lukasiewicz PODS 2009]
- Same as the logical translation of Conceptual Graph rules
- Generalize lightweight Description Logics used for OBQA

Graphical View of Existential Rules

« Value Invention »

$$\forall X\ \forall Y\ (\underbrace{B[X, Y]}_{\text{graph}} \rightarrow \exists Z\ \underbrace{H[X, Z]}_{\text{graph}})$$

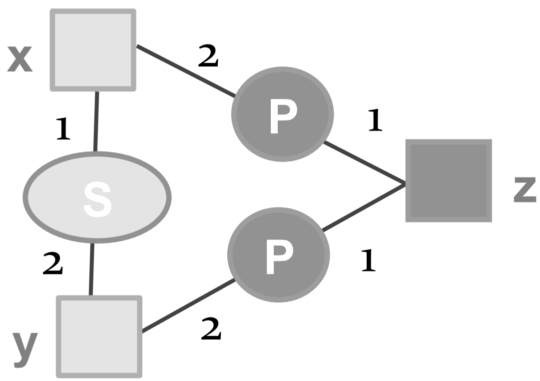
$\forall x\ \forall y\ (siblingOf(x,y) \rightarrow \exists z\ (parentOf(z,x) \wedge parentOf(z,y)))$



Value Invention (Generation of Fresh Existentials)

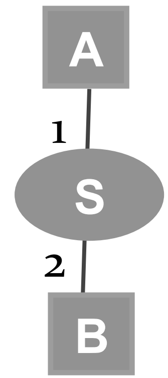
R = ∀x ∀y (siblingOf(x,y) → ∃ z (parentOf(z,x) ∧ parentOf(z,y)))

F = siblingOf(A,B)



h: body → F

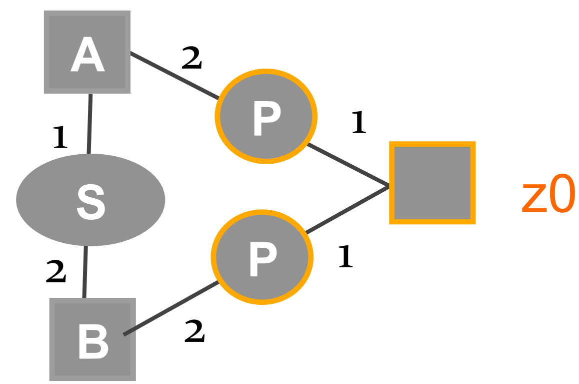
h ={(x,A), (y,B)}



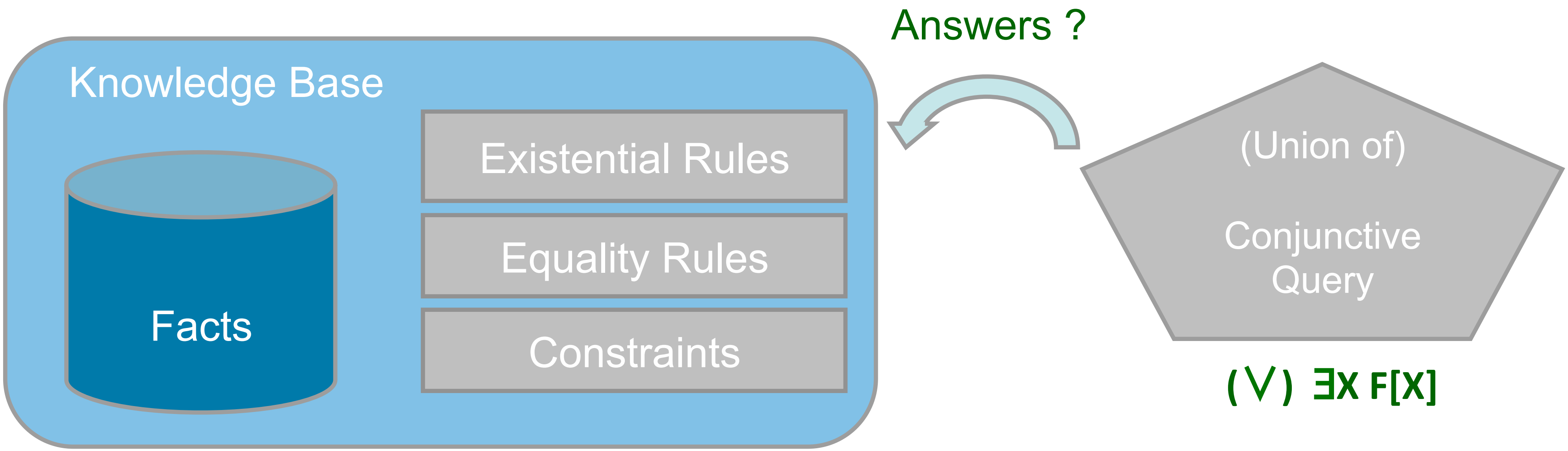
A rule *body*→ *head* is applicable to a fact *F* if
there is a homomorphism *h* from *body* to *F*

The resulting fact is *F'= F ∪ h(head)*
[with renaming existential variables of head *]*

F'= ∃ z0 (siblingOf(A,B)
∧ parentOf(z0,A) ∧ parentOf(z0,B))



Existential Rules Framework (Datalog+/-)



Existential Rule: $\forall X\, \forall Y\, (\, B[X, Y] \rightarrow \exists Z\, H[X, Z]\,)$

Equality Rule: $\forall X\ \ (B[X] \rightarrow x = e)$ with x,e variables or constants

Negative Constraint: $\neg\, \exists X\, B[X]$ or $\forall X\, (B[X] \rightarrow \bot)$

Existential Rules cover « lightweight » Description Logics

- New DLs tailored for ontology-based data access:

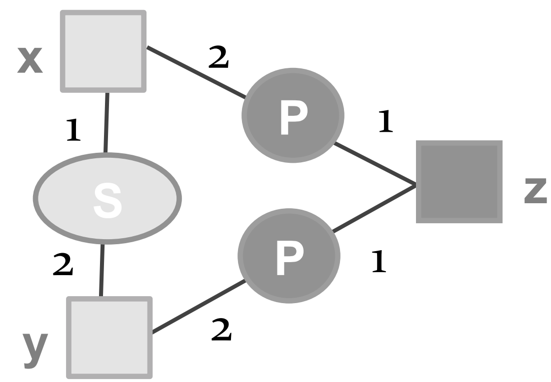
DL-Lite
$\mathcal{EL}$ } Core of the « tractable profiles » of OWL2

Human $\sqsubseteq$ $\exists$ parentOf^{-}.Human

Human(x) → parentOf(y,x) ∧ Human (y)

- Existential rules are more expressive:

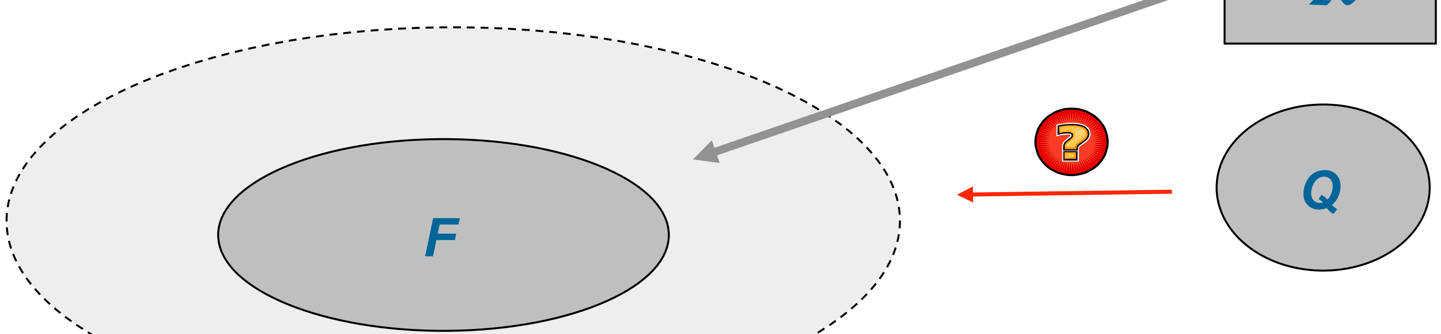
siblingOf(x,y) → parentOf(z,x) ∧ parentOf(z,y)
not expressible in DL



- Non-bounded predicate arity provides more flexibility:
  → direct correspondence with database relations
  → adding contextual information is easy

Forward vs Backward Chaining

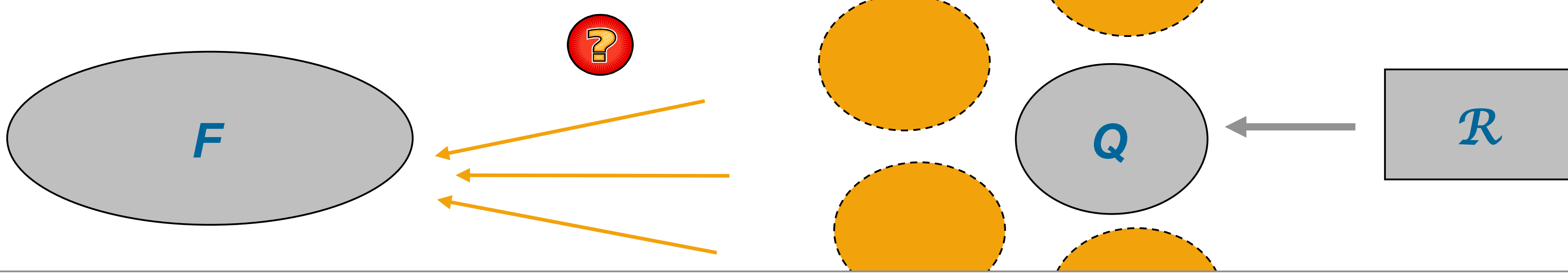
FC **Fact saturation (« bottom-up »)**



F, $\mathcal{R}$ |= *Q iff Q* can be mapped to ***F'***,
where ***F'*** is obtained by a derivation sequence from *F* with $\mathcal{R}$

BC **Query rewriting (« top-down »)**

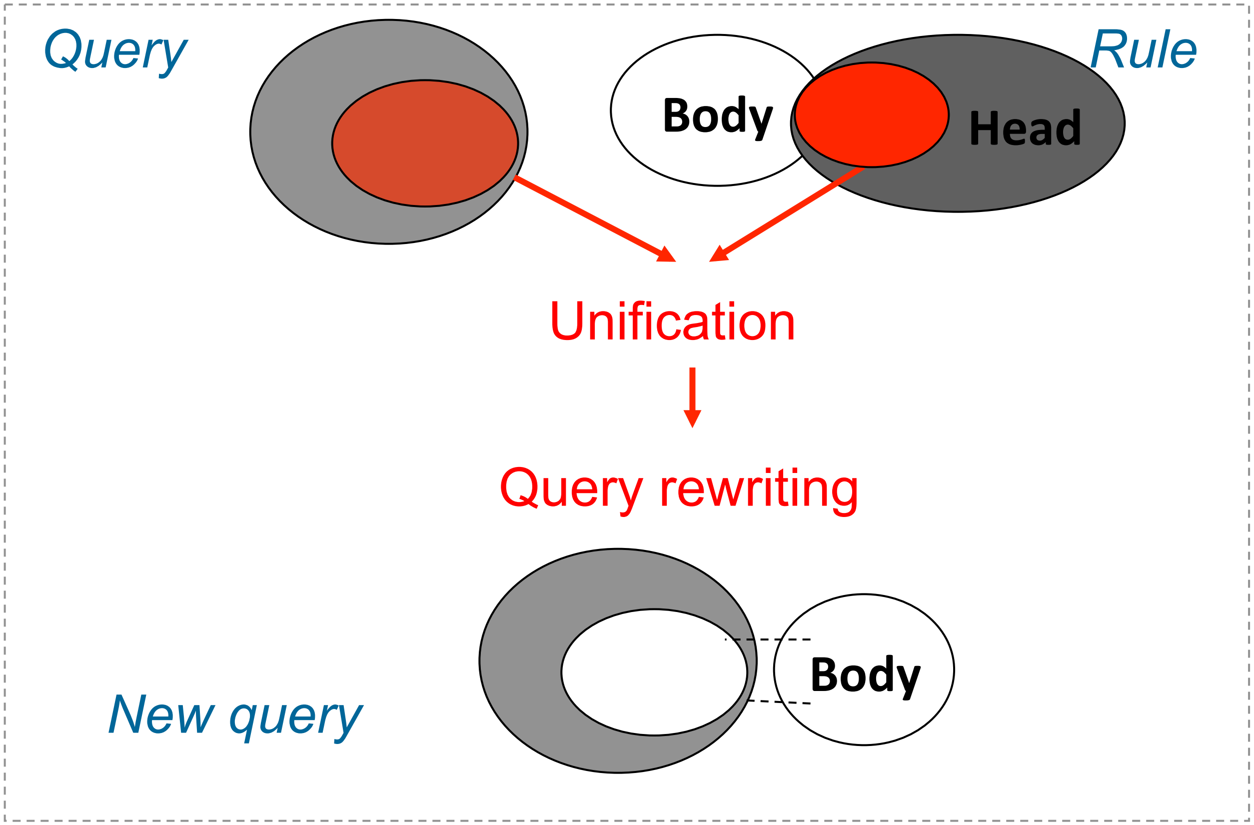
[Decomposition into 2 steps: DL-Lite]



F, $\mathcal{R}$ |= *Q iff* ***Q'*** can be mapped to *F*, where ***Q'*** is obtained by a rewriting sequence from *Q* with $\mathcal{R}$

Backward Chaining Scheme

- Basic step:



Forward chaining may not halt

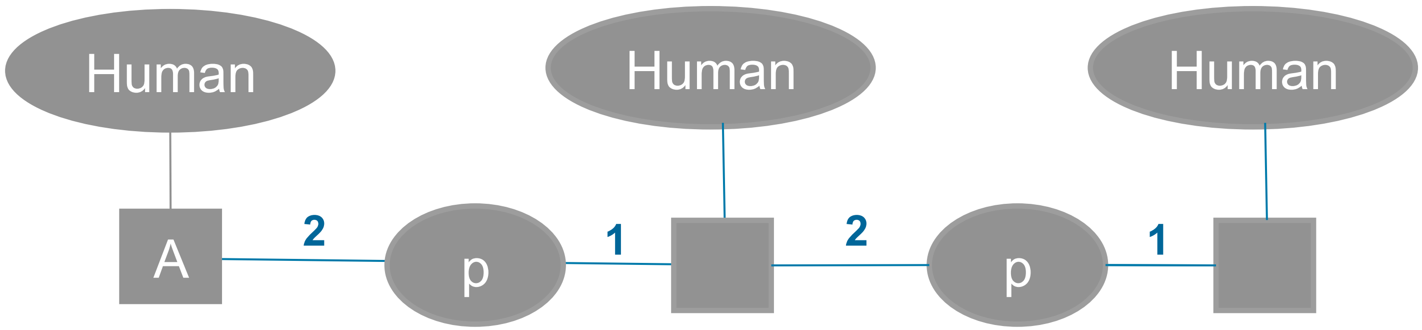
R = Human(x) → parentOf(y,x) ∧ Human(y)   *(Quantifiers omitted)*

F = Human(A)

∧ Human(y0) ∧ parentOf(y0, A)

∧ Human(y1) ∧ parentOf(y1, y0)

Etc.



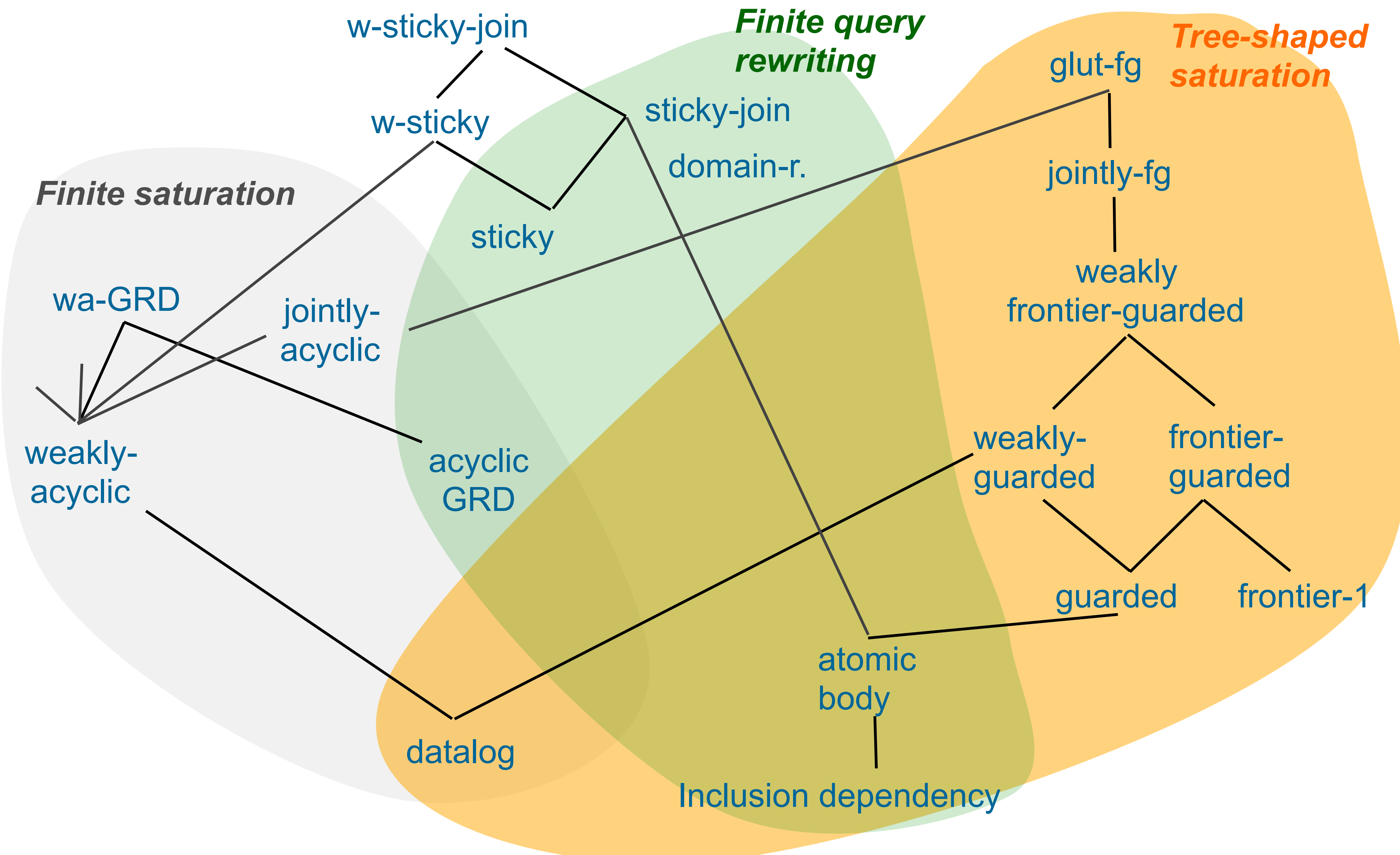
[the same trouble with backward chaining]
Finding a halting method is hopeless: entailment is undecidable

Decidability Issues

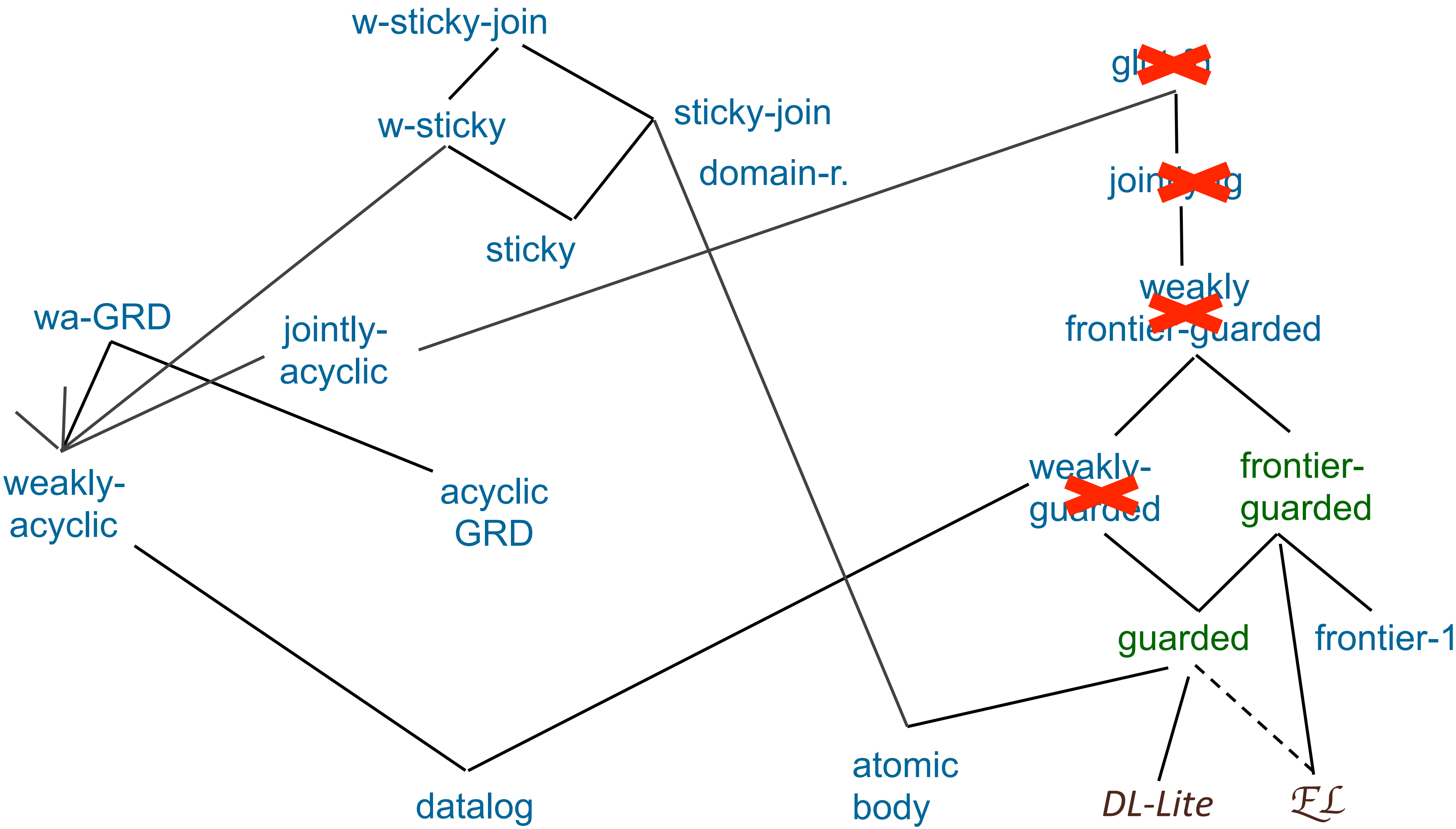
- Entailment is not decidable
- Many « decidable » classes exhibited in databases and KR
- Three generic kinds of properties ensuring decidability:
  - Saturation by Forward Chaining halts
  - Query rewriting by Backward Chaining halts
  - Saturation by Forward Chaining does not halt
    but the generated facts have a tree-like structure

(Partial) inclusion map of decidable classes

E.g. [Mugnier RR 2011] for details



Decidable Classes with Polynomial Data Complexity



Conclusion on Existential Rules for Ontology-Based QA

- An emerging rule-based framework
  - simple, expressive and flexible
  - suitable to Ontology-Based Query Answering

- Currently:
  - A quite clear picture of decidable classes of rules with complexity analysis
  - Effervescence around new algorithmic techniques (in particular other kinds of rewritings-
  - First implementations for very specific subclasses

- Challenges
  - Scalability
  - Querying knowledge bases with inconsistent data