Combining Existential Rules and Description Logics

Antoine Amarilli[1,2], Michael Benedikt[2]

1: Télécom ParisTech; Institut Mines-Télécom; CNRS LTCI; Paris, France
2: University of Oxford; Oxford, United Kingdom

Open-World Query Answering (QA)
or query containment, or query entailment...

Instance I: set of **ground facts** *(or A-box)*
Example: *parent(Joe)*

Constraints Σ: logical **rules** *(or T-box)*
Example: *∀p parent(p) → ∃c child(p, c)*

Boolean **conjunctive query *q***
Example: *∃c child(Joe, c)*

QA problem: given I, Σ, q:
- for all **completions** $J \supseteq I$
- such that J **satisfies** Σ
- does J **always satisfy** q?

i.e.: • *is there a **counterexample** $J \supseteq I$ satisfying Σ but not q?* • *is q **certain** given I and Σ?* • *does $I \wedge \Sigma$ **entail** q?*

Two families of decidable constraint languages for Σ

Rich description logics	Frontier-guarded existential rules
Emp ⊑ CEO ⊔ (∃Mgr⁻.Emp)	∀*pwv* Accept(*p*, *w*, *v*) → ∃*f* Trip(*p*, *v*, *f*)
Rich constraints *Can express disjunction, disjointness, etc.*	**Poor** constraints *Conjunction and implication only*
Arity-two only	**Arbitrary** arity
Functionality asserts *Example: Funct(Mgr)*	**n/a** *Not part of the language*

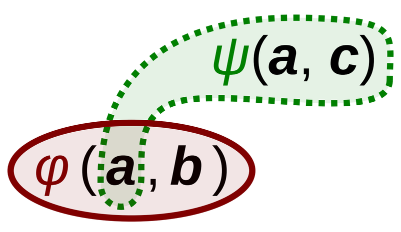
→ QA is **decidable** for **either** of these languages...

Problem statement: *Can QA be **decidable** when allowing **both** rich description logics and existential rules?*

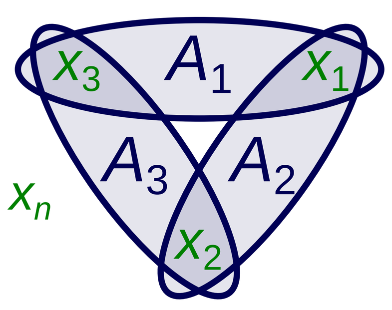
Terminology: A rule class *C* is **destructive** / **non-destructive** when QA for rules in class *C* and for rich DLs is **decidable** / **undecidable**

Rule languages

- **Rich description logics** (rich DLs): anything expressible in **GC²**
  (two-variable guarded first-order logic with counting quantifiers)
- **Existential rules** (TGDs):
  $\forall \mathbf{xy}\ \varphi(\mathbf{x}, \mathbf{y}) \rightarrow \exists \mathbf{z}\ \psi(\mathbf{x}, \mathbf{z})$
  *where **x**, **y**, **z** are disjoint sets of variables and **φ** (body) and **ψ** (head) are conjunctions of atoms*



- **Frontier-one** (FR[1]): ***x*** is a **singleton**
  i.e., only one variable shared between body and head
- **Non-looping**: no **bad cycle**
  - **Berge cycle**: distinct atoms and variables $A_1, x_1, \ldots, A_n, x_n$ such that x_i occurs in A_i and A_{i+1} for all i



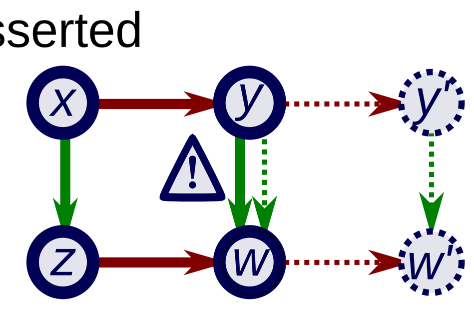
  → **Bad cycle**: Berge cycle where ***n*>2** or some A_i has **arity >2**
  Examples: *R(x, y) S(y, z) T(z, y) or A(x, x, y) R(x, y)*
- **Head-non-looping:** no bad cycle in head atoms

Negative results

- **Frontier-two** FR[2] is **destructive**
  In fact frontier-two **inclusion dependencies** (ID[2]) are sufficient (only one atom in head and body, no repeated variables)
  Problem: entailment of Funct() and ID[2] is undecidable

→ Must restrict to **frontier-one** FR[1]

- **Frontier-one** FR[1] is **destructive**
  Problem: the existence of **cycles** can be asserted
  ∀x φ(x) → ∃yzw *R*(x y) *D*(x z) *R*(z w) *D*(y w)
  and Funct(*R*) Funct(*D*) yields a **grid**

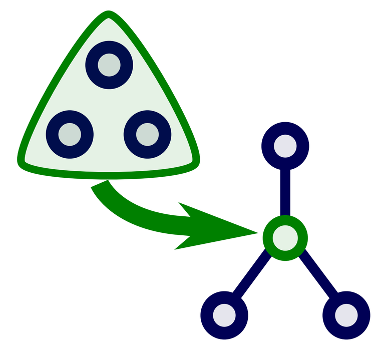


→ Must impose **non-looping**

Positive results: head-non-looping FR[1]

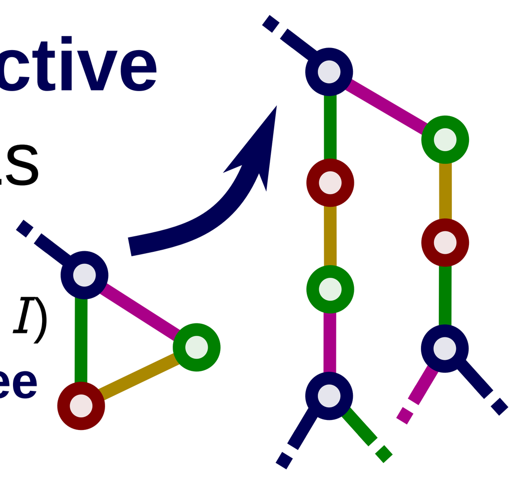
- **Non-looping FR[1]** is **non-destructive:**
  → QA for this class + rich DLs **reduces** to QA for rich DLs

Idea: ***shred*** *R*(*a*, *b*, *c*) to $R_1(f, a)\ R_2(f, b)\ R_3(f, c)$ in I and q
Use rich DLs to impose well-formedness constraints on the signature
Lemma: can inductively **rewrite** non-looping FR[1] to DL constraints
Example: ∀*ux U*(*u*), *T*(*x*, *u*), *S*(*x*) → ∃*yz T*(*x*, *y*), *U*(*y*), *R*(*x*, *x*, *z*, *z*)
shreds to (∃T⁻.U) ⊓ S ⊑ (∃T.U) ⊓ (∃(R_1^- ⊓ R_2^-).(∃(R_3 ⊓ R_4).⊤))



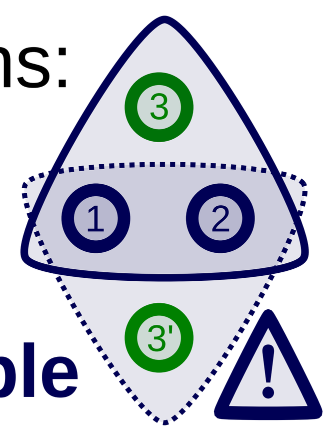
- **Head-non-looping FR[1]** is **non-destructive**
  → reduces to QA for non-looping + rich DLs

Idea: head-non-looping FR[1] can be **treeified** to non-looping
(consider all possible variable identifications and matches to I)
Unravelling: any counterexample $J \supseteq I$ can be made **cycle-free**
→ **Lemma:** replacing rules by their treeification is **sound**



Positive results: functional dependencies (FDs)

FDs generalize Funct(•) to arbitrary arity relations:
$\forall \mathbf{xy}\ R(x_1\ x_2\ x_3),\ R(y_1\ y_2\ y_3),\ x_1{=}y_1,\ x_2{=}y_2 \rightarrow x_3{=}y_3$
Example: *Talk[speaker,session]* ***determines*** *Talk[title]*



- QA with just FR[1] rules and FDs is **undecidable** but decidable with **non-conflicting condition:**
  - all FR[1] rules are **single-head** — and hence **head-non-looping**
  - for each ∀x φ(x) → ∃**y** R(<u>x</u>, y_1, <u>x</u>, y_2, ...), head positions with **frontier variable** are
    - not a **strict superset** of an FD determiner (= left-hand-side of an FD)
    - if **equal** to a determiner, all variables in ***y*** occur only once

→ What about QA for rules, **FDs**, and rich DLs?

- **Single-head FR[1]** and **non-conflicting FDs** are **non-destructive**
  Idea: modify unravelling to ensure FDs are **respected**
  (when unravelling high-arity facts, distinguish variables based on FD determiners)
  → The non-conflicting condition ensures that such changes cannot **violate** the rules