dRDF: Entailment for Domain-Restricted RDF

Reinhard Pichler [1] Axel Polleres [2] Fang Wei [1] Stefan Woltran [1]
[1] Institute for Information Systems, TU Vienna, Austria
[2] DERI, National University of Ireland, Galway

Alternative subtitles:

Blank nodes are fun (at least for theoreticians)

or

Blank nodes ain't THAT evil! ☺

RDF Entailment: $G_1 \models G_2$

- Does graph G_1 entail G_2 ?
- Boils down to:
  "Is there a blank node renaming μ for blank nodes in G_2 such that $\mu(G_2) \subseteq G_1$"
- "Folklore": Well-known to be NP-complete (cf. RDF Semantics [Hayes, 2004])
- Observation: *Blank nodes* are causing the "trouble" of making the problem intractable… ground entailment well known to be in P.

Starting point for our work:

Besides completely forbidding blank nodes…

… What else can we do to make this problem tractable?

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Restrictions on RDF graphs considered in this paper:

1. ***Domain-Restricted Graphs:*** Restrict the domain blank nodes can range over to a finite set of objects.

2. ***Graphs with Bounded Treewidth:*** Restrict the graph structure of RDF graphs: bounded-treewidth (a generalization of acyclicity)

Effects:

1. …OOPS! With *finite domains*, complexity actually jumps from NP to $coNP^{NP} = \Pi^p_2$ ☹
2. Not all is lost: *bounded treewidth* guarantees tractability for general entailment and coNP bound for domain-restricted graphs.

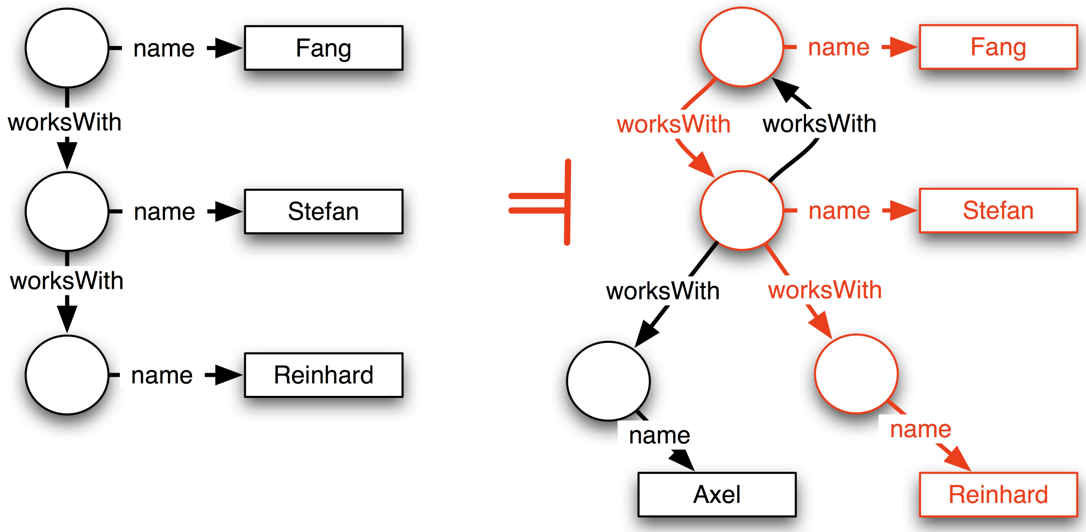
Summary:

	domain-restricted graphs	Unrestricted graphs
bounded treewitdth	**coNP-complete**	**in P** ☺
unbounded treewidth	Π^p_2 **-complete**	NP-complete

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

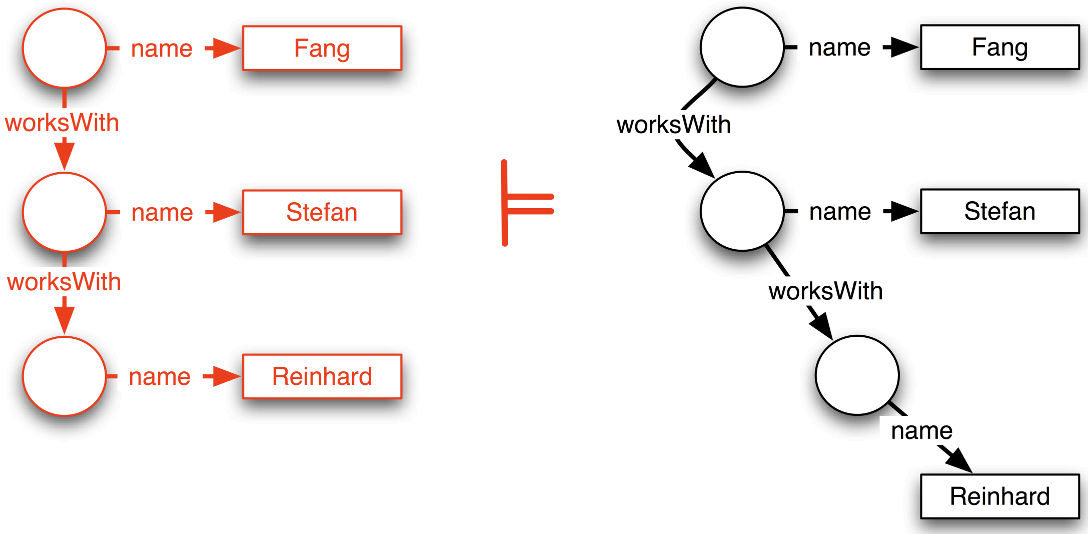
G_1	G_2	G_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1), (_:b_1,:worksWith,_:b_3), (_:b_4,foaf:name,"Axel"), (_:b_1,:worksWith, _:b_4)	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)



National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

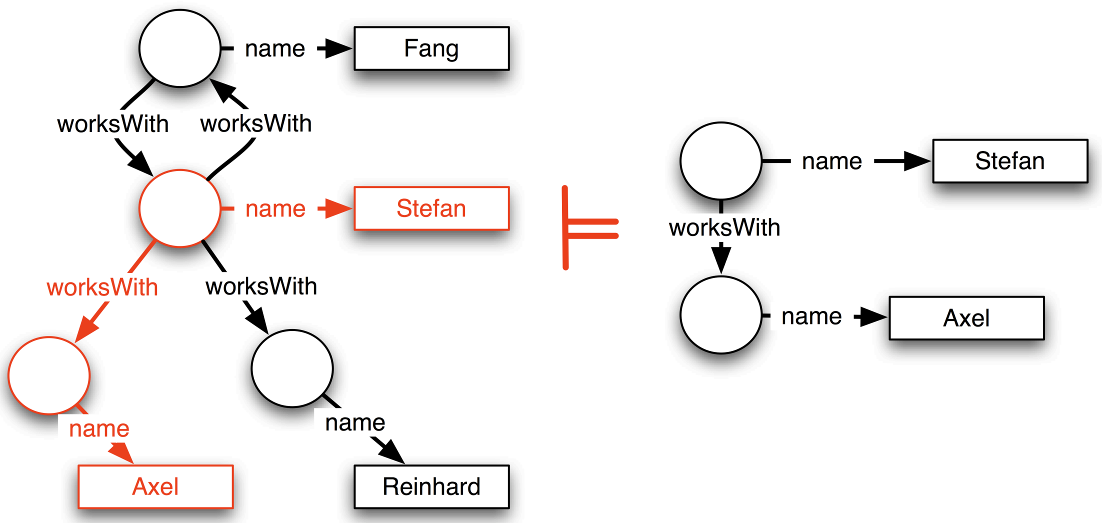
G_1	G_2	G_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1),	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)



National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

G_1	G_2	G_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1), (_:b_1,:worksWith,_:b_3), (_:b_4,foaf:name,"Axel"), (_:b_1,:worksWith, _:b_4)	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)



National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

Fang Wei	Stefan Woltran	Stefan Decker
G_1 D_1	G_2 D_2	G_3 D_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1), (_:b_1,:worksWith,_:b_3), (_:b_4,foaf:name,"Axel"), (_:b_1,:worksWith, _:b_4)	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)

D_1 = **TUV** =
{:fangwei, :stefanwoltran, :reinhardpichler, :thomaseiter, … }

D_2 = **TUV** ∪ **Alumni** =
{:fangwei, :stefanwoltran, :reinhardpichler, :thomaseiter, :axelpolleres, :manfredhauswirth, … }

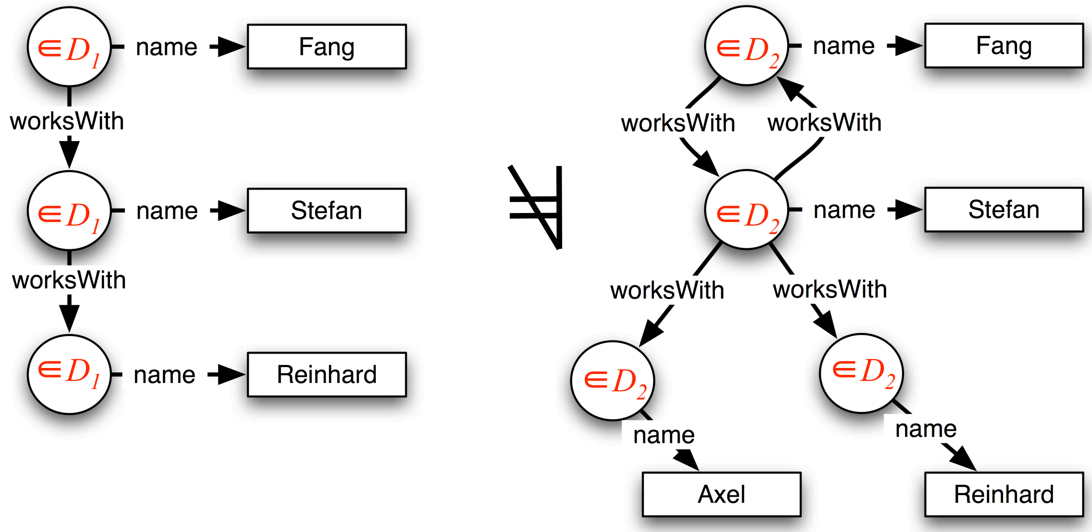
D_3 = **DERI** =
{:stefandecker, :axelpolleres, :manfredhauswirth … }

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

Fang Wei	Stefan Woltran	Stefan Decker
G_1 D_1	G_2 D_2	G_3 D_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1), (_:b_1,:worksWith,_:b_3), (_:b_4,foaf:name,"Axel"), (_:b_1,:worksWith, _:b_4)	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)

$D_1 \subseteq D_2$

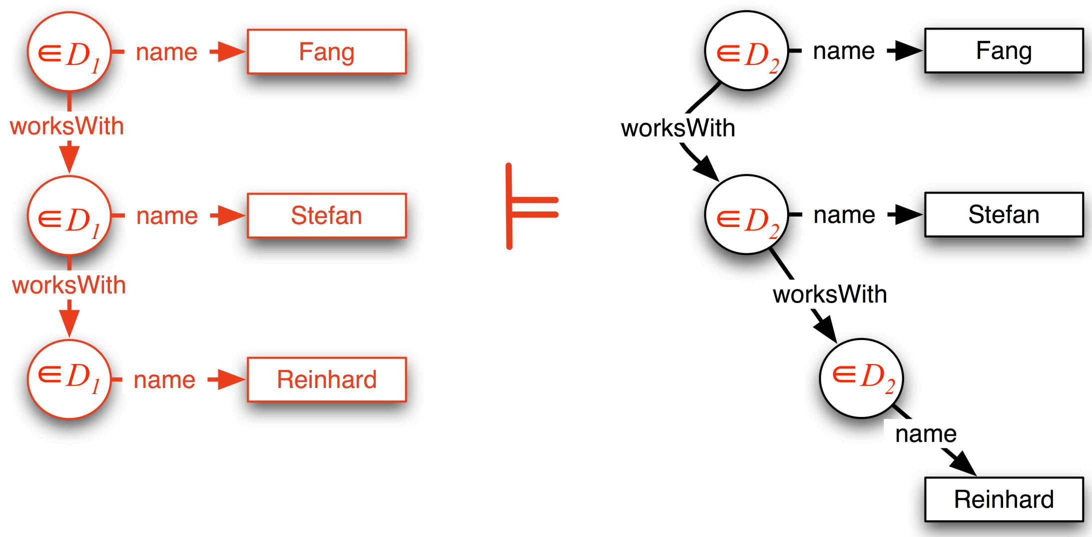


National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Example

G_1	G_2	G_3
(_:b_1,foaf:name,"Fang"), (_:b_2,foaf:name,"Stefan"), (_:b_3,foaf:name,"Reini"), (_:b_1,:worksWith,_:b_2), (_:b_2,:worksWith,_:b_3)	(_:b_1,foaf:name,"Stefan"), (_:b_2,foaf:name,"Reini"), (_:b_3,foaf:name,"Fang"), (_:b_1,:worksWith,_:b_2), (_:b_3,:worksWith,_:b_1),	(_:b_2,foaf:name,"Stefan"), (_:b_1,foaf:name,"Axel"), (_:b_2,:worksWith,_:b_1)

$D_1 \subseteq D_2$



National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

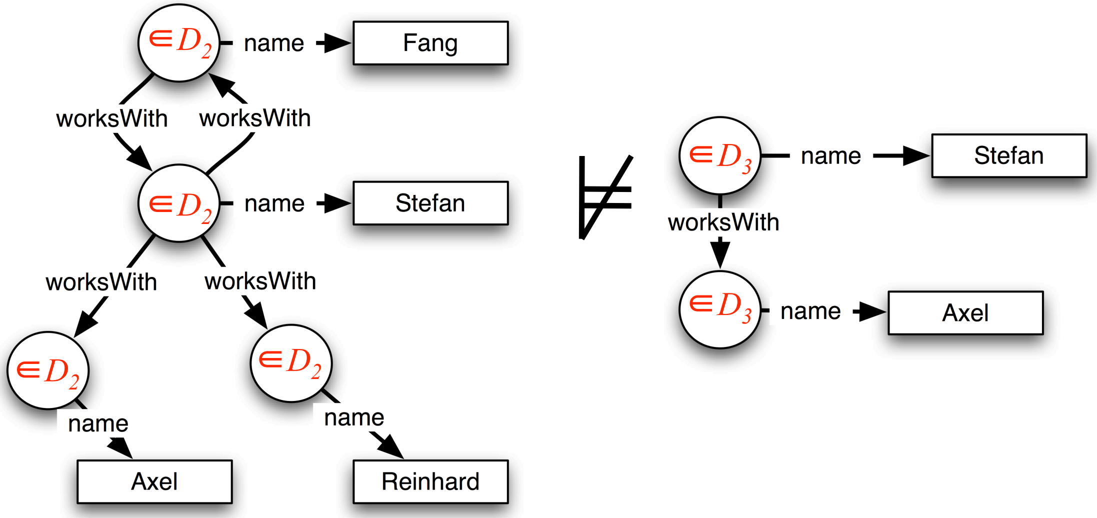
Domain-Restricted Graphs: Example

Fang Wei — Stefan Woltran — Stefan Decker

G_1 D_1	G_2 D_2	G_3 D_3

`(_:`b_1`,foaf:name,"Fang"),`
`(_:`b_2`,foaf:name,"Stefan"),`
`(_:`b_3`,foaf:name,"Reini"),`
`(_:`b_1`,:worksWith,_:`b_2`),`
`(_:`b_2`,:worksWith,_:`b_3`)`
	`(_:`b_1`,foaf:name,"Stefan"),`
	`(_:`b_2`,foaf:name,"Reini"),`
	`(_:`b_3`,foaf:name,"Fang"),`
	`(_:`b_1`,:worksWith,_:`b_2`),`
	`(_:`b_3`,:worksWith,_:`b_1`),`
	`(_:`b_1`,:worksWith,_:`b_3`),`
	`(_:`b_4`,foaf:name,"Axel"),`
	`(_:`b_1`,:worksWith, _:`b_4`)`
		`(_:`b_2`,foaf:name,"Stefan"),`
		`(_:`b_1`,foaf:name,"Axel"),`
		`(_:`b_2`,:worksWith,_:`b_1`)`

$$D_2 \not\subseteq D_3,\ D_3 \not\subseteq D_2$$



Domain-Restricted Graphs $\langle G , D \rangle$: Definition

- Base notion in RDF semantics: RDF interpretation for a Graph G

$$I = (Res, Prop, Lit, \varepsilon, IS, IL)$$

- We define the D-restriction of RDF interpretations:

$$I_D = (Res \cap D, Prop, Lit \cap D, \varepsilon, IS_{Res \cap D}, IL_{Res \cap D})$$

- Entailment for domain-restricted graphs, defined as wrt. D-restriction of RDF interpretations:

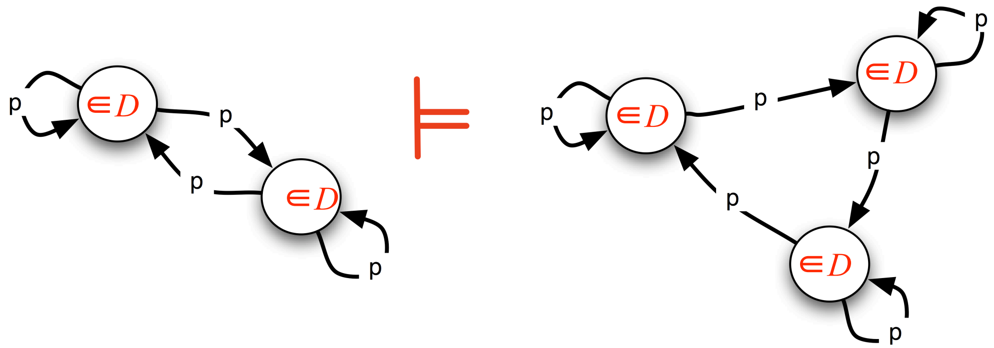
$$\langle G_1 , D_1 \rangle \models \langle G_2 , D_2 \rangle$$

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Domain-Restricted Graphs: Properties

- $D_1 \nsubseteq D_2$ implies $\langle G_1, D_1 \rangle \not\models \langle G_2, D_2 \rangle$
- $G_1 \models G_2$ implies $\langle G_1, D \rangle \models \langle G_2, D \rangle$
- *But: Complexity of D-entailment is* Π_2^p … *Uh?*

Example:
$D = \{a,b\}$



- *Intuitively:*
  More entailments by implicit equalities if $|D|$ *is small enough!*

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Complexity proof (Ideas)

- *Membership: non-entailment in* Π^p_2 *:*
  - *We can assume w.l.o.g. that* G_1 *is ground*
  - "$\langle G_1, D\rangle$ does not d-entail $\langle G_2, D\rangle$" can be decided in Σ^p_2 by
    1. *Guessing a D-interpretation such that* G_1 *is true*
    2. *Check that* G_2 *is false for all possible assignments of bnodes to elements of D*

- *Hardness proof by a reduction from a special variant of H-subsumption**, *for* $|D| \geq 4$ … long version.

* *"total binary H-subsumption" i.e., no constants are allowed in clauses and only binary predicates, fixed finite Herbrand universe*

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Now how to remedy the mess we did…

- … we saw the first “restriction” made things more complex.

- But: ***bounded treewidth*** helps!

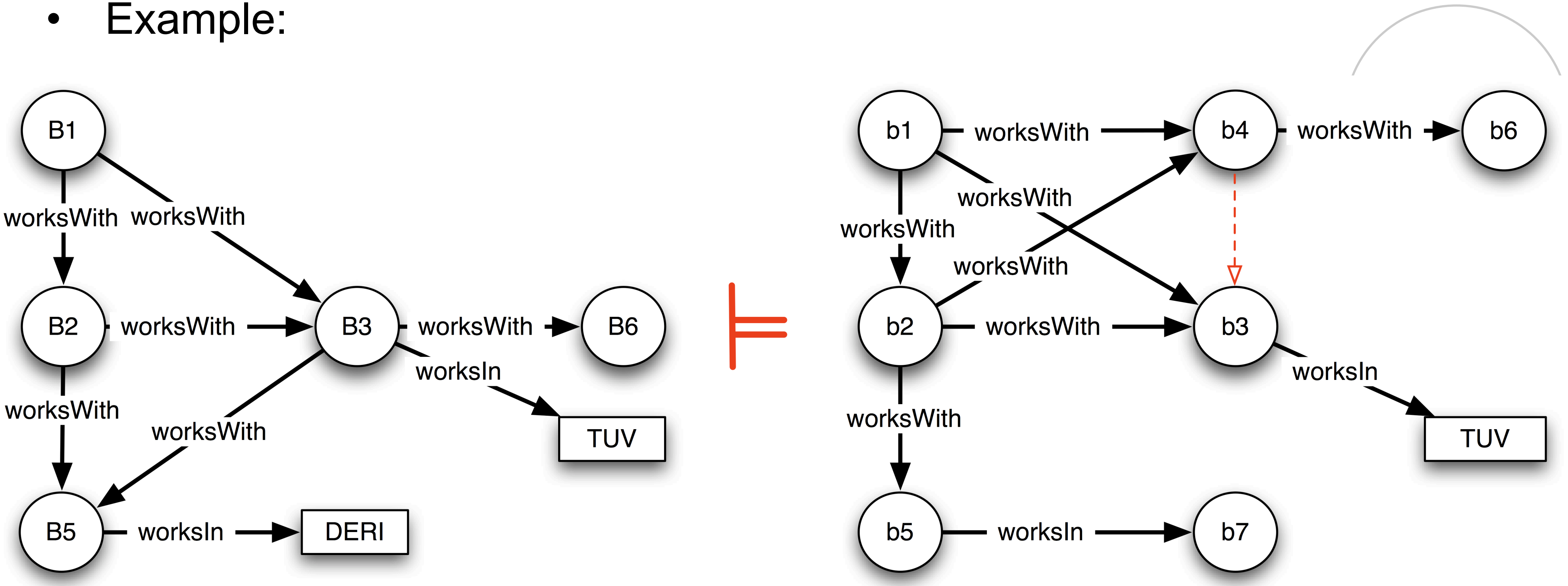
National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Bounded Treewidth for RDF graphs:

- Measure of "acyclicity"
- Roughly:

  "If I can decompose the graph to a tree of hyper-edges with at most k -1 nodes per edge, then the graph has treewidth k"

- Example:



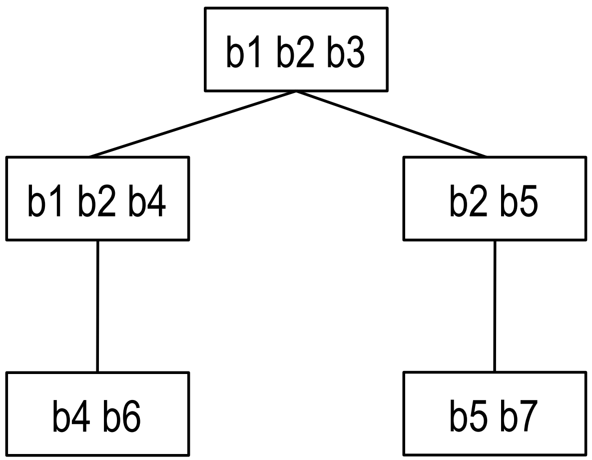
National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Bounded Treewidth for RDF graphs:

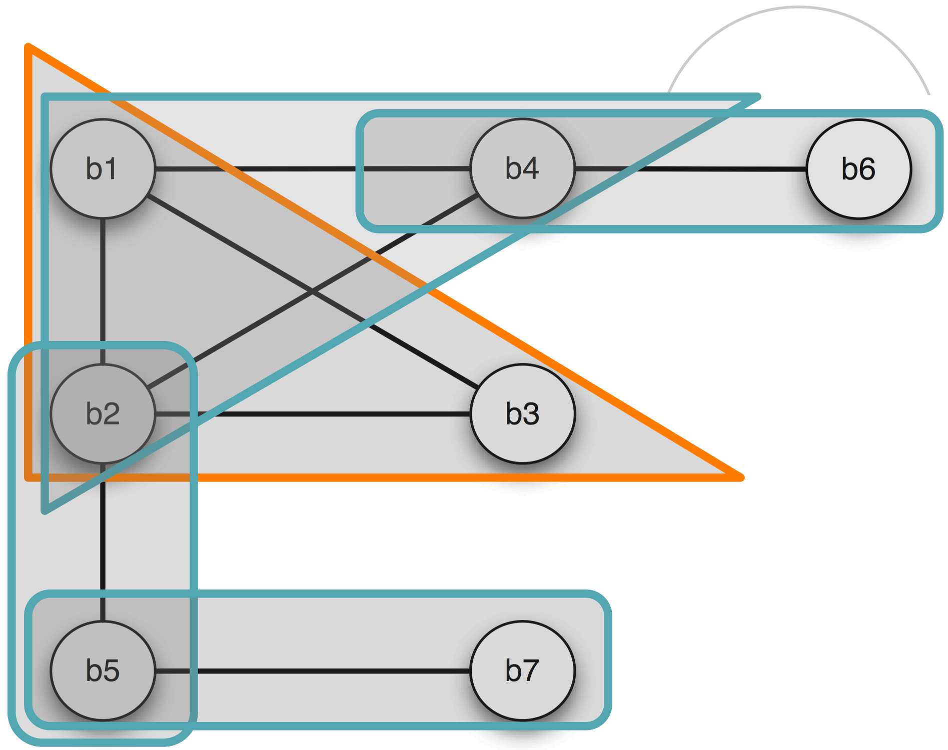
- Measure of “acyclicity”
- Roughly:

  “If I can decompose the graph to a tree of hyper-edges with at most k -1 nodes per edge, then the graph has treewidth k”
- Example:

“Skeleton” relevant for tree-decomposition:

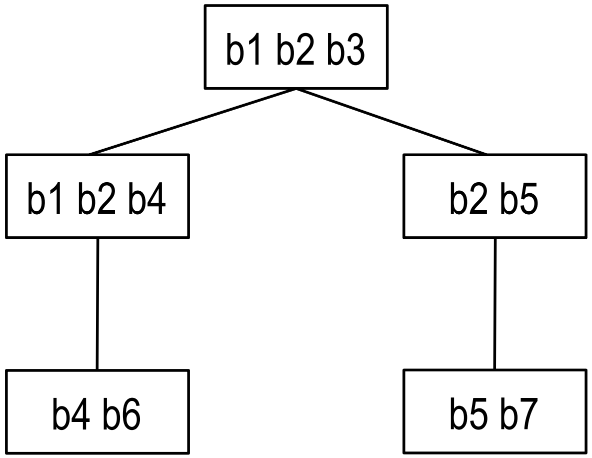


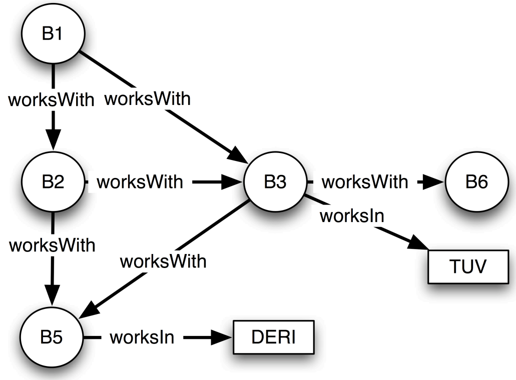
$$tw(G_2) = 2$$

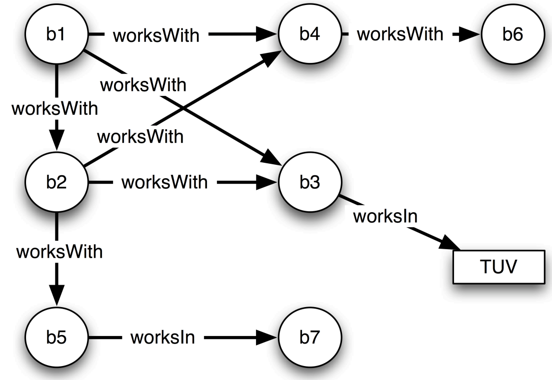


National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Polynomial time Algorithm for Entailment with Bounded Treewidth for G_2 (Idea):



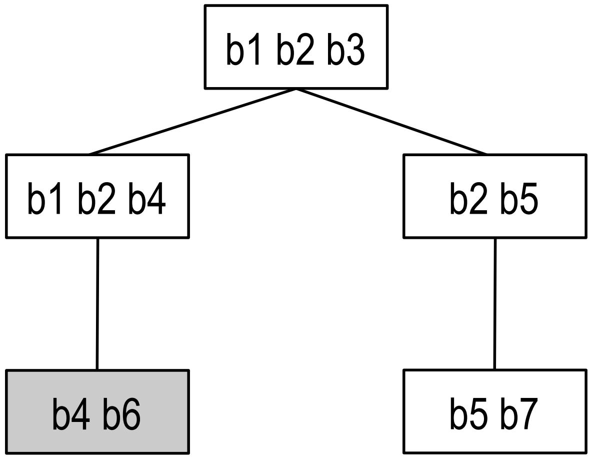


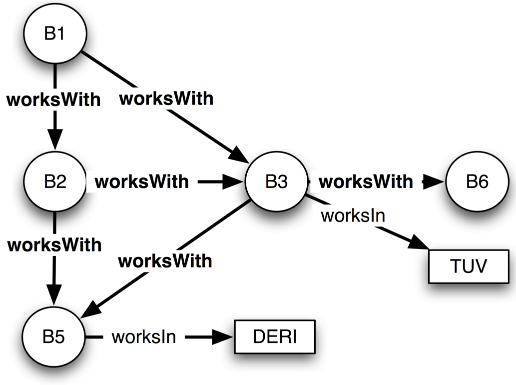


- From the decomposition, process the ***induced subgraphs*** "bottom-up" in a modular fashion, computing partial bnode assignments.
- When going upwards, filter allowed assignments by *semi-joins* with the assignments for the child nodes.
- If an assignment "survives" at the root, entailment holds.
- $O(n^k)$ for entailment checks per node
- $O(n^{2k})$ per semi-join
- Thus, for $|G_2| = m$ we get as upper bound: $O(m^2+mn^{2k})$

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Polynomial time Algorithm for Entailment with Bounded Treewidth for G_2 (Idea):



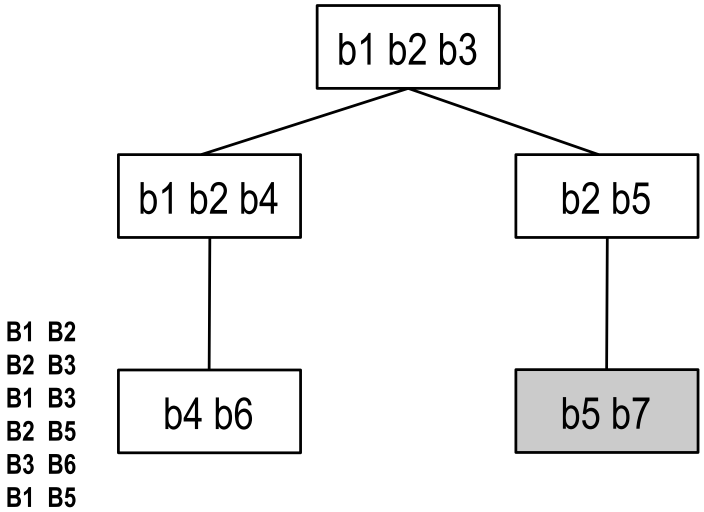


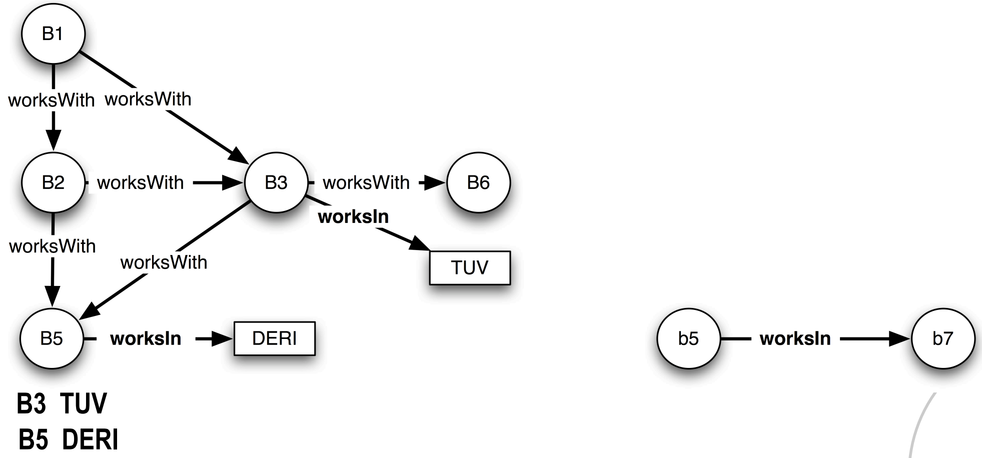
B1 B2
B2 B3
B1 B3
B2 B5
B3 B6
B1 B5

- From the decomposition, process the *induced subgraphs* “bottom-up” in a modular fashion, computing partial bnode assignments.
- When going upwards, filter allowed assignments by *semi-joins* with the assignments for the child nodes.
- If an assignment “survives” at the root, entailment holds.
- $O(n^k)$ for entailment checks per node
- $O(n^{2k})$ per semi-join
- Thus, for $|G_2| = m$ we get as upper bound: $O(m^2+mn^{2k})$

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Polynomial time Algorithm for Entailment with Bounded Treewidth for G_2 (Idea):

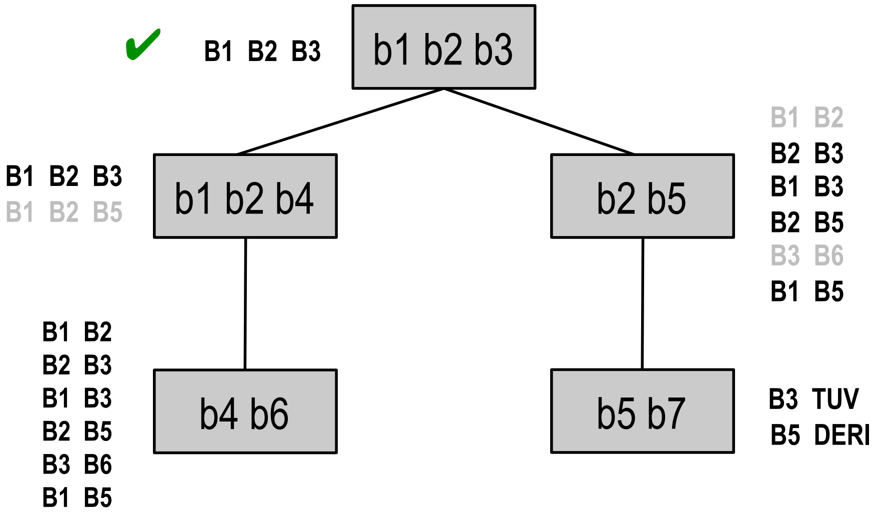




- From the decomposition, process the *induced subgraphs* “bottom-up” in a modular fashion, computing partial bnode assignments.
- When going upwards, filter allowed assignments by *semi-joins* with the assignments for the child nodes.
- If an assignment “survives” at the root, entailment holds.
- $O(n^k)$ for entailment checks per node
- $O(n^{2k})$ per semi-join
- Thus, for $|G_2| = m$ we get as upper bound: $O(m^2+mn^{2k})$

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Polynomial time Algorithm for Entailment with Bounded Treewidth for G_2 (Idea):



- From the decomposition, process the ***induced subgraphs*** “bottom-up” in a modular fashion, computing partial bnode assignments.
- When going upwards, filter allowed assignments by *semi-joins* with the assignments for the child nodes.
- If an assignment “survives” at the root, entailment holds.
- $O(n^k)$ for entailment checks per node
- $O(n^{2k})$ per semi-join
- Thus, for $|G_2| = m$ we get as upper bound: $O(m^2+mn^{2k})$

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Now what about D-entailment with bounded treewidth?

- Overall complexity drops from Π^p_2 to coNP:
  Recall from above:

  - "$\langle G_1, D\rangle$ does not d-entail $\langle G_2, D\rangle$" can be decided in Σ^p_2 by
    1. *Guessing a D-interpretation such that G_1 is true*
    2. *Check that G_2 is false for all possible assignments of bnodes to elements of D*

- Step *2.* can be done in polynomial time for bounded tree-width.

- coNP-hardness still holds (proof by 3-colorability, see paper.)

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
sfi science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.

Summary:

- Some form of domain-restriction may be useful for graphs on the Web…
  - … but comes at some cost!
  - Things are not that bad unless we expect small domains (less elements than bnodes)

- Similar results for
  - enumerated classes in (fragments of) OWL?
  - entailment with finite datatypes?, etc.

  ➔ Future work!

- Bounded treewidth is more general than acyclicity. **Good news!** (if we don't expect graphs with large cycles among bnodes)

National University of Ireland, Galway
Ollscoil na hÉireann, Gaillimh
science foundation ireland
fondúireacht eolaíochta éireann
Enabling **networked** knowledge.