



Raisonnement embarqué et distribué pour le Web des Objets

Projet CoSWoT

Alexandre Bento, Lionel Médini, Kamal Singh, Frédérique Laforest



Plan

- CoSWoT
- Raisonnement à base de règles
- Vers un raisonnement embarqué
- Vers un raisonnement distribué
- Conclusion

Contexte et objectifs : le projet CoSWoT

Développement et exécution d'applications intelligentes et décentralisées pour le WoT, malgré des objets contraints

Modèles de connaissances basés sur les graphes pour spécifier déclarativement :

- la sémantique des messages échangés entre les noeuds
- le domaine et la logique applicative

Distribuer et traiter les tâches de raisonnement au sein de noeuds hétérogènes, y compris des objets contraints en tenant compte :

- de l'infrastructure matérielle
- des caractéristiques des objets

Plan

- CoSWoT
- Raisonement à base de règles
- Vers un raisonnement embarqué
- Vers un raisonnement distribué
- Conclusion

Raisonnement à base de règles

- Format RDF
 - Les faits sont des triplets < sujet, prédicat, objet > ou des quads < sujet, prédicat, objet, graphe >
- Inférence
 - Production de faits implicites à partir de règles conjonctives et de faits explicites
 - Chaînage avant : itérations jusqu'à ce qu'aucun nouveau fait ne soit produit
 - L'ordre d'application des règles n'a pas d'importance

Conditions (corps)

→ Conclusion (tête)

(?p a fini ses devoirs) ET (Il fait beau)

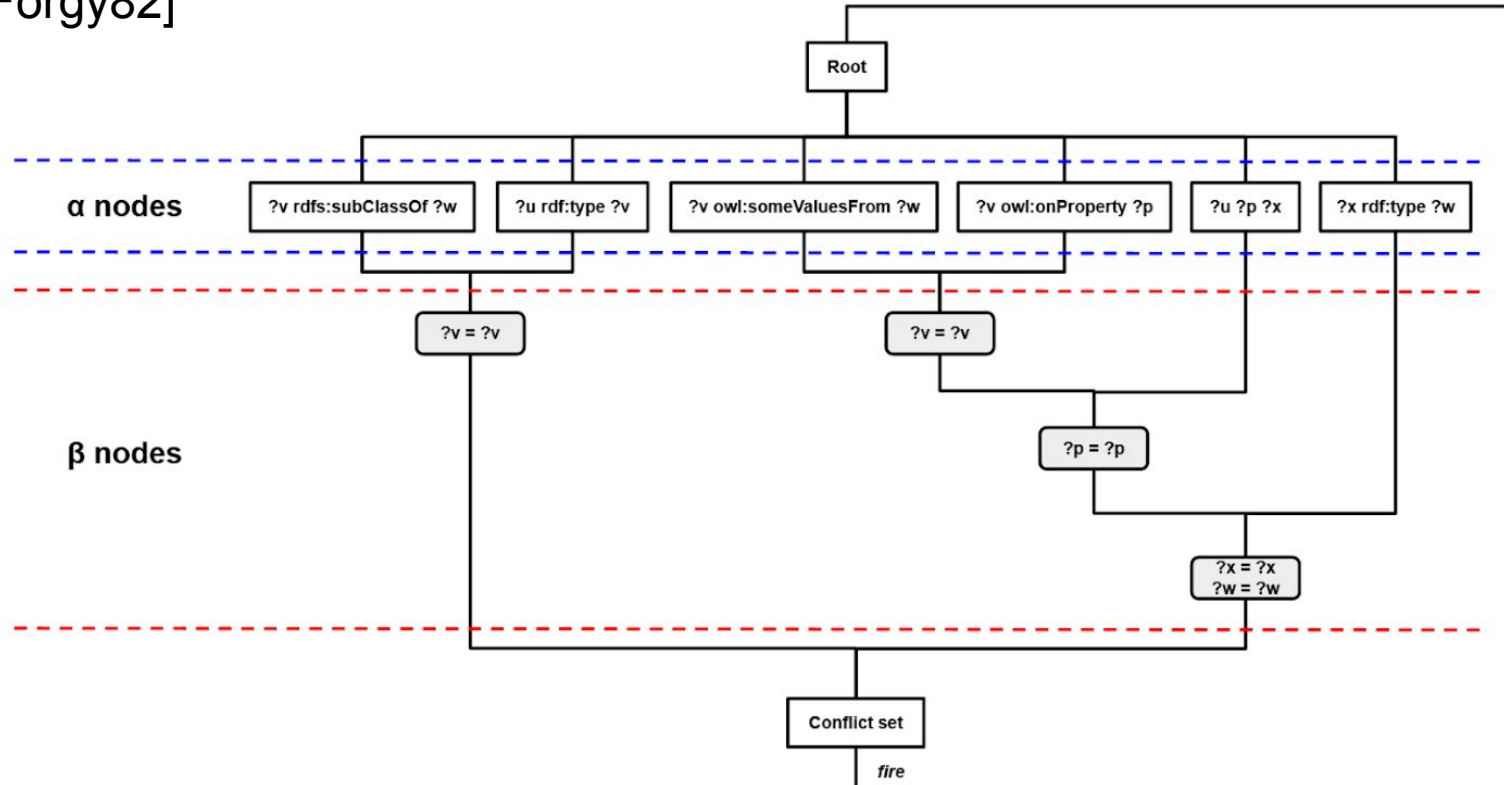
→ **?p peut aller marcher**

Plan

- CoSWoT
- Raisonnement à base de règles
- Vers un raisonnement embarqué
 - RETE
 - DRed
 - Backward-Forward
 - Tag-Based
- Vers un raisonnement distribué
- Conclusion

Vers un raisonnement embarqué

RETE [Forgy82]



Vers un raisonnement embarqué

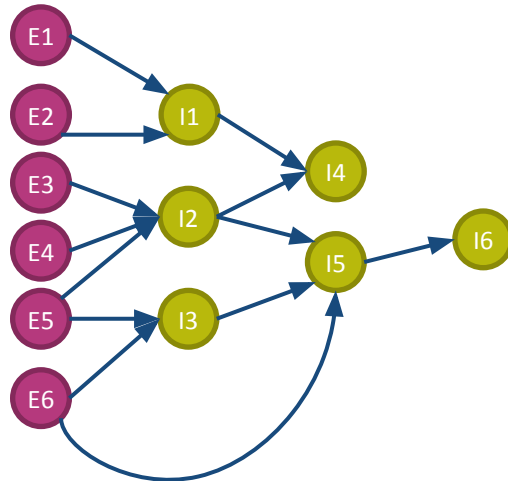
- Optimisations de RETE :
 - Classer les noeuds par sélectivité [Woensel18]
 - Construire un arbre de dépendances entre règles pour ne charger que les règles nécessaires lorsque de nouveaux faits explicites sont ajoutés [Wei15]

Vers un raisonnement embarqué

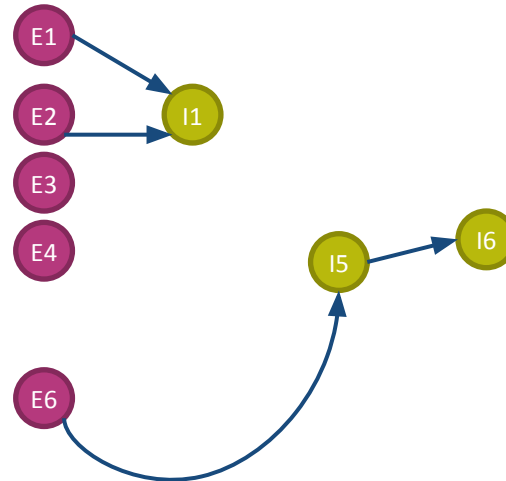
Delete/Rederive [Gupta 1993]

- Exemple: suppression de E5

Sur-suppression



Redérivation

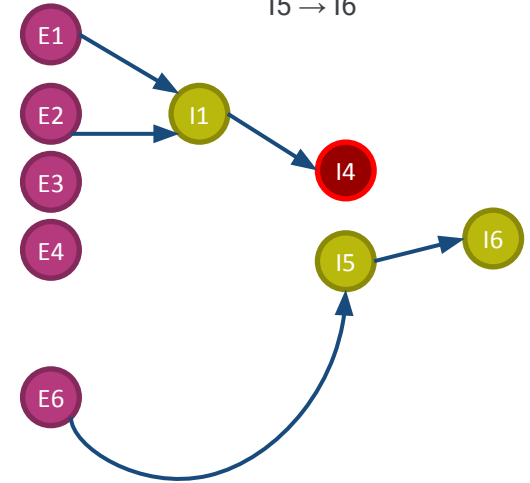
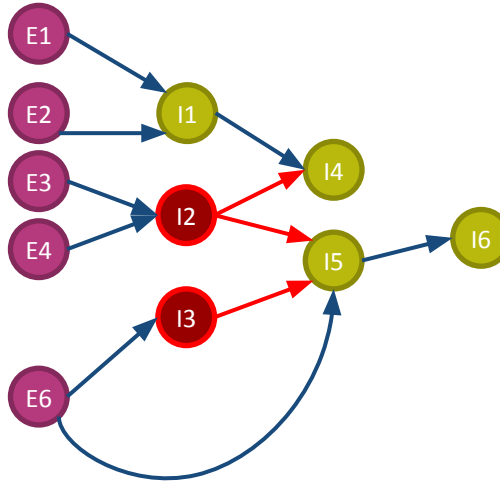
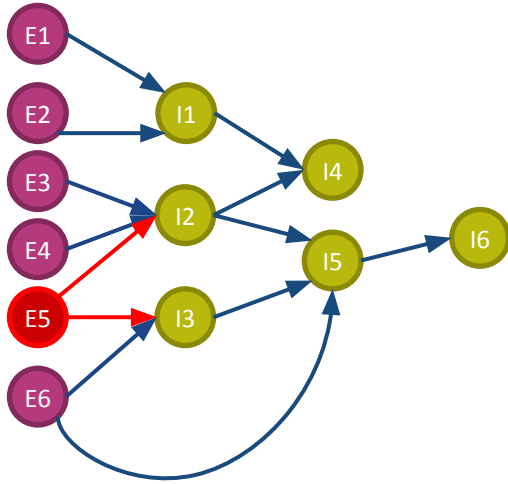


E1 → I1
E2 → I1
E3 et E4 et E5 → I2
E5 et E6 → I3
I1 et I2 → I4
I2 → I5
I3 → I5
E6 → I5
I5 → I6

Vers un raisonnement embarqué

Backward/Forward [Motik2015]

- Exemple: suppression de E5



E1 → I1
E2 → I1
E3 et E4 et E5 → I2
E5 et E6 → I3
I1 et I2 → I4
I2 → I5
I3 → I5
E6 → I5
I5 → I6

Vers un raisonnement embarqué

Solutions qui gardent une trace des raisonnements précédents

- Tag-Based Reasoning (TBR): exemple

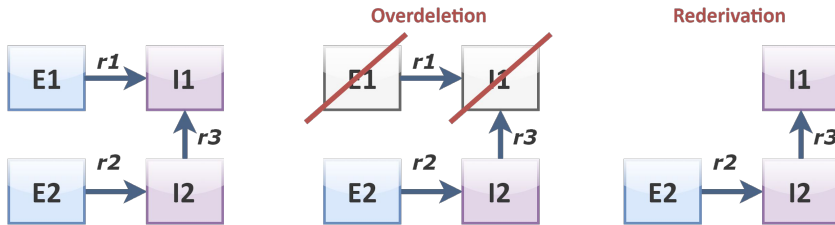
- Règles

- $r1 : E1 \rightarrow I1$
 - $r2 : E2 \rightarrow I2$
 - $r3 : I2 \rightarrow I1$

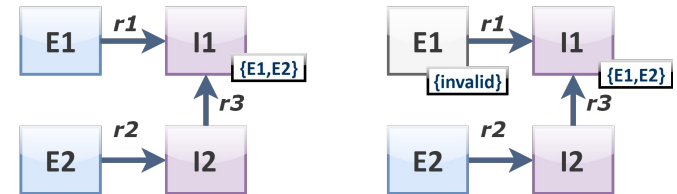
- Scénario

- E1 suppression / ré-insertion

IR



TBR



Plan

- CoSWoT
- Raisonnement à base de règles
- Vers un raisonnement embarqué
- Vers un raisonnement distribué
 - Parallélisation :
 - RDFox
 - Adaptations de RETE
 - Distribution dirigée par :
 - Les données
 - Les règles
- Conclusion

Vers un raisonnement distribué

TBox

R1	(?v rdfs:subClassOf ?w),	(?u rdf:type ?v)	→	(?u rdf:type ?w)
R2	(?v owl:someValuesFrom ?w),	(?v owl:onProperty ?p),	(?u ?p ?x),	(?x rdf:type ?w) → (?u rdf:type ?v)

ABox

T1	ex:Car	rdf:type	rdfs:Class
T2	ex:Car	rdfs:subClassOf	ex:Vehicule
T3	ex:Fiat	rdfs:subClassOf	ex:Car
T4	ex:Engine	rdf:type	rdfs:Class
T5	ex:hasEngine	rdf:type	owl:Restriction
T6	ex:hasEngine	owl:onProperty	ex:hasComp
T7	ex:hasEngine	owl:someValuesFrom	ex:Engine
T8	ex:myCar	rdf:type	ex:Car
T9	ex:azrTurbo	rdf:type	ex:Engine
T10	ex:myCar	ex:hasComp	ex:azrTurbo
T11	ex:myCar	ex:hasComp	ex:alcon
T12	ex:myCar	ex:hasComp	ex:energyMX1

Thread 1

Fait : T7

Règle : R2

Match : (?u ?p ?x)

Substitution : ?u = ex:hasEngine
?p = owl:someValuesFrom
?x = ex:Engine

Règle partielle :

(?v owl:someValuesFrom ?w),
(?v owl:onProperty
owl:someValuesFrom),
(ex:Engine rdf:type ?w)
→ (ex:Engine rdf:type ?v)



Thread 2



Vers un raisonnement distribué

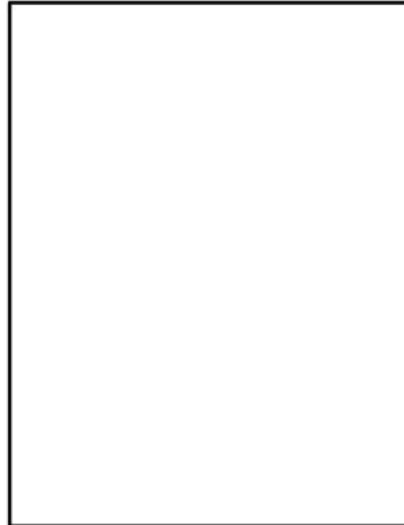
TBox

R1	(?v rdfs:subClassOf ?w),	(?u rdf:type ?v)	→	(?u rdf:type ?w)
R2	(?v owl:someValuesFrom ?w),	(?v owl:onProperty ?p),	(?u ?p ?x),	(?x rdf:type ?w) → (?u rdf:type ?v)

ABox

T1	ex:Car	rdf:type	rdfs:Class
T2	ex:Car	rdfs:subClassOf	ex:Vehicule
T3	ex:Fiat	rdfs:subClassOf	ex:Car
T4	ex:Engine	rdf:type	rdfs:Class
T5	ex:hasEngine	rdf:type	owl:Restriction
T6	ex:hasEngine	owl:onProperty	ex:hasComp
T7	ex:hasEngine	owl:someValuesFrom	ex:Engine
T8	ex:myCar	rdf:type	ex:Car
T9	ex:azrTurbo	rdf:type	ex:Engine
T10	ex:myCar	ex:hasComp	ex:azrTurbo
T11	ex:myCar	ex:hasComp	ex:alcon
T12	ex:myCar	ex:hasComp	ex:energyMX1

Thread 1



Thread 2

Fait : T8

Règle : R1

Match : (?u rdf:type ?v)

Substitution : ?u = ex:myCar
?v = ex:Car

Règle partielle :

(ex:Car rdfs:subClassOf ?w)
→ (ex:myCar rdf:type ?w)

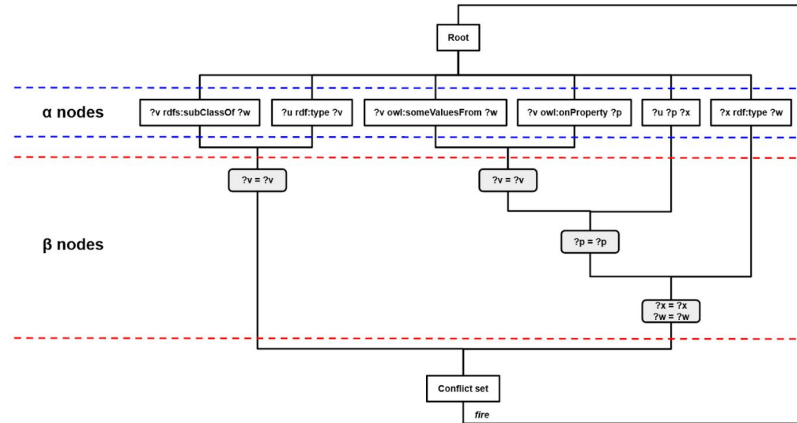
Déduction : (ex:myCar rdf:type
ex:Vehicule)



Vers un raisonnement distribué

Parallélisation

- RETE:
 - Traitement parallèle de noeuds indépendants sur plusieurs faits



Vers un raisonnement distribué

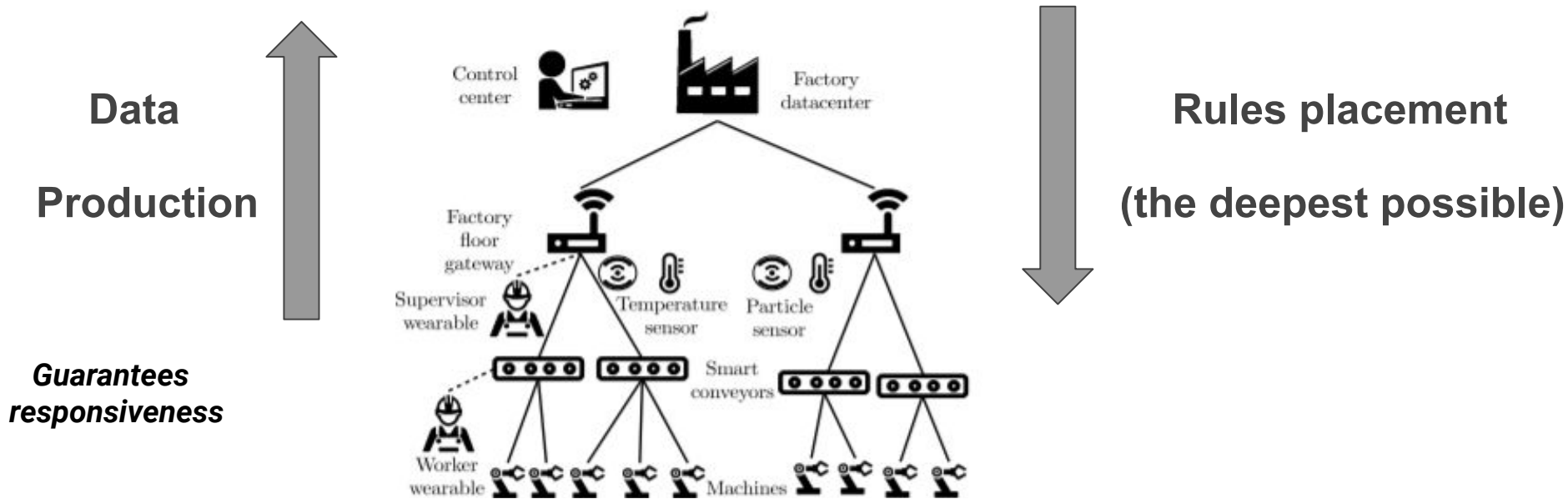
Distribution dirigée par les données :

- Les caractéristiques des données définissent le mode de répartition du raisonnement
 - Sources de données distribuées
 - Contraintes de répartition de charges
 - Découpage intelligent des données (ex : en fonction du schéma)

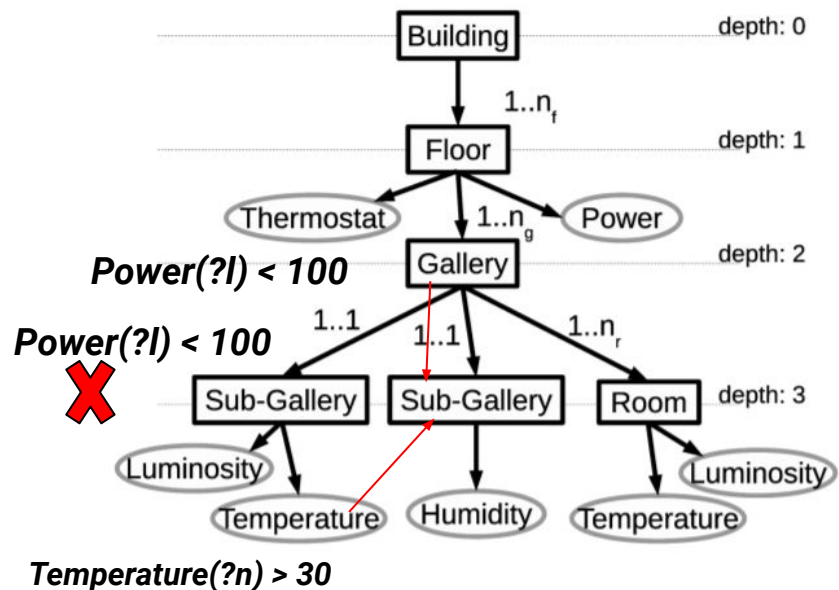
Vers un raisonnement distribué : Distribution dirigée par les règles

[Seydoux2019] : EDR: A Generic Approach for the Distribution of Rule-Based Reasoning in a Cloud-Fog continuum

Examples : detecting whether power consumption is economical or working environment is secure



Vers un raisonnement distribué : Distribution dirigée par les règles ...



Règle de déduction :

$Power(?l) > 100 \wedge Temperature(?n) > 30 \rightarrow$ **The gallery is not economical**

Préconditions

Postconditions (déduction)

Conclusion

- Constats :
 - La plupart des approches embarquées reposent sur l'algorithme Rete
 - Les approches multithread ne correspondent pas aux contraintes du raisonnement embarqué et distribué
 - Les langages bas-niveau offrent de meilleures performances sur les objets contraints
 - Il faut développer des méthodes de raisonnement basées sur des données distribuées
- Questions ouvertes :
 - Le raisonnement embarqué requiert une empreinte mémoire/énergie/bande-passante faible → quelles structures de données devons-nous utiliser ?
 - Le processus de raisonnement devrait-il être distribué selon les règles ou les données ?
 - La gestion du raisonnement devrait-elle être centralisée ou décentralisée ?
 - Quel format de règles devrait être utilisé pour réduire l'empreinte mémoire et réseau ?

Questions ?

Références

- [Forgy82] Charles L Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. In *Readings in Artificial Intelligence and Databases*, pages 547–559. Elsevier, 1989
- [Woensel18] William Van Woensel and Syed Sibte Raza Abidi. Optimizing semantic reasoning on memory-constrained platforms using the rete algorithm. In *European Semantic Web Conference*, pages 682–696. Springer, 2018
- [Wei15] Wei Tai, John Keeney, and Declan O’Sullivan. Resource-constrained reasoning using a reasoner composition approach. *Semantic Web*, 6(1):35–59, 2015

Références

- [Gupta1993] Ashish Gupta, Inderpal Singh Mumick, and Venka-tramanan Siva Subrahmanian. Maintaining views in-crementally. ACM SIGMOD Record, 22(2):157–166,1993
- [Motik2015] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. Rdflox: A highly-scalable rdf store. In International Semantic Web Conference, pages 3–20. Springer, 2015
- [Seydoux2019] Nicolas Seydoux, Khalil Drira, Nathalie Hernandez, and Thierry Monteil. Edr: A generic approach for the dynamic distribution of rule-based reasoning in a cloud-fog continuum. Semantic Web Journal, 2019