Explications d’inférences pour Datalog

Lucas Rouquette

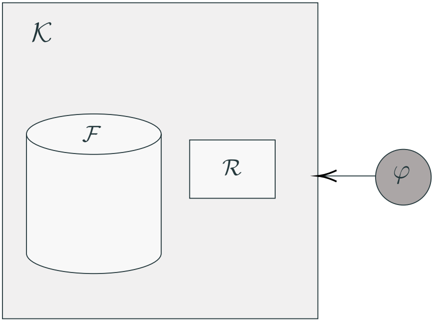
18 novembre 2022

INRIA - Équipe Boréal

Sommaire

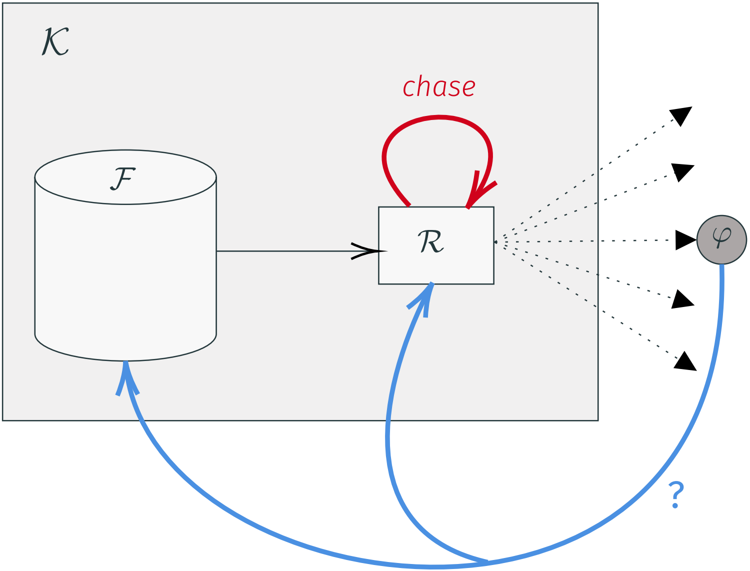
Contexte et motivations

Contexte



- Base de connaissances
- Connaissances
  $\rightarrow$ *faits*
- Règles de déduction
  $\rightarrow$ *datalog*
- Inférence de fait
  $\rightarrow$ requêtes *ground*

Motivations



Notions de bases

Datalog

- Règles

$$\forall \bar{x}_1, \ldots, \bar{x}_n. \left(\bigwedge_{i=1}^{n} \beta_i(\bar{x}_i) \rightarrow \alpha(\bar{z}) \right)$$

où $\beta_1(\bar{x}_1), \ldots, \beta_n(\bar{x}_n)$ *et* $\alpha(\bar{z})$ *sont des atomes,* $n \geq 1$*, et toute variable est universellement quantifiée.*

- Règles

$$\forall \bar{x}_1, \ldots, \bar{x}_n . \left(\bigwedge_{i=1}^{n} \beta_i(\bar{x}_i) \rightarrow \alpha(\bar{z}) \right)$$

où $\beta_1(\bar{x}_1), \ldots, \beta_n(\bar{x}_n)$ *et* $\alpha(\bar{z})$ *sont des atomes,* $n \geq 1$*, et toute variable est universellement quantifiée.*

- Approche par chaînage avant (*Chase*)

Remarque

Ch($\mathcal{K}$) = résultat du chase sur la base de connaissances $\mathcal{K}$.

- Règles

$$\forall \bar{x}_1, \ldots, \bar{x}_n. \left(\bigwedge_{i=1}^{n} \beta_i(\bar{x}_i) \rightarrow \alpha(\bar{z}) \right)$$

où $\beta_1(\bar{x}_1), \ldots, \beta_n(\bar{x}_n)$ *et* $\alpha(\bar{z})$ *sont des atomes,* $n \geq 1$*, et toute variable est universellement quantifiée.*

- Approche par chaînage avant (*Chase*)

Remarque

$\mathrm{Ch}(\mathcal{K})$ = résultat du chase sur la base de connaissances $\mathcal{K}$.

- Inférence de fait (*entailment*)

Proposition

$\mathcal{K} \models \varphi$ **si et seulement si** $\varphi \in \mathrm{Ch}(\mathcal{K})$

Grounding

Definition

Soit une KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$, le *grounding* $\boldsymbol{gr}_{\mathcal{K}}(r)$ de $r \in \mathcal{R}$ est l'ensemble de règles contenant:

$$h(body(r)) \rightarrow h(head(r))$$

pour tout homomorphismes h de $body(r)$ dans $\mathrm{Ch}(\mathcal{K})$.
De plus, $\boldsymbol{gr}_{\mathcal{K}}(\mathcal{R}) = \bigcup_{r \in \mathcal{R}} \boldsymbol{gr}_{\mathcal{K}}(r)$.

Exemple

$$r_1 = p(X) \rightarrow r(X, X)$$
$$r_2 = r(X, Y) \wedge q(Y) \rightarrow r(Y, X)$$
$$r_3 = r(X, X) \wedge r(X, Y) \rightarrow \text{goal}(X)$$
$$r_4 = s_1(X, X) \rightarrow w(X)$$
$$r_5 = s_2(X, X) \rightarrow w(X)$$
$$r_6 = v(X) \wedge w(X) \rightarrow \text{goal}(X)$$

$p(a)$, $v(a)$, $s_1(a, a)$, $s_2(a, a)$, $q(a)$
$r(b, a)$, $s_1(b, b)$, $p(c)$

Exemple

$r_1 = p(X) \rightarrow r(X,X)$

$r_2 = r(X,Y) \wedge q(Y) \rightarrow r(Y,X)$

$r_3 = r(X,X) \wedge r(X,Y) \rightarrow \text{goal}(X)$

$r_4 = s_1(X,X) \rightarrow w(X)$

$r_5 = s_2(X,X) \rightarrow w(X)$

$r_6 = v(X) \wedge w(X) \rightarrow \text{goal}(X)$

$p(a)$, $v(a)$, $s_1(a,a)$, $s_2(a,a)$, $q(a)$
$r(b,a)$, $s_1(b,b)$, $p(c)$

$$p(a) \rightarrow r(a,a) \in \boldsymbol{gr}_{\mathcal{K}}(\mathcal{R})$$

$$p(c) \rightarrow r(c,c) \in \boldsymbol{gr}_{\mathcal{K}}(\mathcal{R})$$

mais

$$p(b) \rightarrow r(b,b) \notin \boldsymbol{gr}_{\mathcal{K}}(\mathcal{R})$$

MUS *(Minimal Unsatisfiable Subset)*

$$\Phi = (L_1^1 \vee \ldots \vee L_{n_1}^1) \wedge \ldots \wedge (L_1^m \vee \ldots \vee L_{n_m}^m)$$

Definition

Soit une formule de Horn Φ, un *ensemble minimal insatisfiable* (MUS) de Φ est un ensemble $\mathcal{M} \subseteq \text{clauses}(\Phi)$ tel que:

- $\mathcal{M}$ est insatisfiable, et
- minimal, i.e., $\forall c \in \mathcal{M}, \mathcal{M} \setminus \{c\}$ est satisfiable.

MUS *(Minimal Unsatisfiable Subset)*

$$\Phi = (L_1^1 \vee \ldots \vee L_{n_1}^1) \wedge \ldots \wedge (L_1^m \vee \ldots \vee L_{n_m}^m)$$

Definition

Soit une formule de Horn Φ, un *ensemble minimal insatisfiable* (MUS) de Φ est un ensemble $\mathcal{M} \subseteq$ clauses(Φ) tel que:

- $\mathcal{M}$ est insatisfiable, et
- minimal, i.e., $\forall c \in \mathcal{M}, \mathcal{M} \setminus \{c\}$ est satisfiable.

$$a \wedge b \wedge \neg g \wedge (\neg a \vee \neg b \vee g) \wedge (\neg a \vee c)$$

MUS *(Minimal Unsatisfiable Subset)*

$$\Phi = (L_1^1 \vee \ldots \vee L_{n_1}^1) \wedge \ldots \wedge (L_1^m \vee \ldots \vee L_{n_m}^m)$$

Definition

Soit une formule de Horn Φ, un *ensemble minimal insatisfiable* (MUS) de Φ est un ensemble $\mathcal{M} \subseteq \text{clauses}(\Phi)$ tel que:

- $\mathcal{M}$ est insatisfiable, et
- minimal, i.e., $\forall c \in \mathcal{M}, \mathcal{M} \setminus \{c\}$ est satisfiable.

$$a \wedge b \wedge \neg g \wedge (\neg a \vee \neg b \vee g) \wedge (\neg a \vee c)$$

Calcul des explications

Definition

Soit une base de connaissances $\mathcal{K}$ et un fait φ, une *explication* pour l'inférence de φ est une KB $\mathcal{K}' \subseteq \mathcal{K}$ telle que:

- $\mathcal{K}' \models \varphi$, et
- $\mathcal{K}'' \not\models \varphi$ pour toute $\mathcal{K}'' \subset \mathcal{K}'$

Remarque

Explanations($\mathcal{K}, \varphi$) = union des explications de φ par rapport à $\mathcal{K}$.

Exemple

$r_1 = p(X) \to r(X, X)$

$r_2 = r(X, Y) \wedge q(Y) \to r(Y, X)$

$r_3 = r(X, X) \wedge r(X, Y) \to \text{goal}(X)$

$r_4 = s_1(X, X) \to w(X)$

$r_5 = s_2(X, X) \to w(X)$

$r_6 = v(X) \wedge w(X) \to \text{goal}(X)$

$$p(a),\ v(a),\ s_1(a, a),\ s_2(a, a),\ r(b, a),\ q(a),\ s_1(b, b),\ p(c)$$

$$\varphi = goal(a)?$$

Exemple

$r_1 = p(X) \rightarrow r(X,X)$

$r_2 = r(X,Y) \wedge q(Y) \rightarrow r(Y,X)$

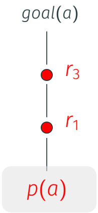
$r_3 = r(X,X) \wedge r(X,Y) \rightarrow \mathrm{goal}(X)$

$r_4 = s_1(X,X) \rightarrow w(X)$

$r_5 = s_2(X,X) \rightarrow w(X)$

$r_6 = v(X) \wedge w(X) \rightarrow \mathrm{goal}(X)$

$$p(a), v(a), s_1(a,a), s_2(a,a), r(b,a), q(a), s_1(b,b), p(c)$$



Exemple

$r_1 = p(X) \rightarrow r(X, X)$

$r_2 = r(X, Y) \wedge q(Y) \rightarrow r(Y, X)$

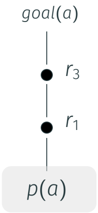
$r_3 = r(X, X) \wedge r(X, Y) \rightarrow \mathrm{goal}(X)$

$r_4 = s_1(X, X) \rightarrow w(X)$

$r_5 = s_2(X, X) \rightarrow w(X)$

$r_6 = v(X) \wedge w(X) \rightarrow \mathrm{goal}(X)$

$$p(a), v(a), s_1(a, a), s_2(a, a), r(b, a), q(a), , s_1(b, b), p(c)$$



Exemple

$r_1 = p(X) \rightarrow r(X,X)$

$r_2 = r(X,Y) \wedge q(Y) \rightarrow r(Y,X)$

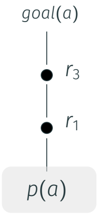
$r_3 = r(X,X) \wedge r(X,Y) \rightarrow \mathrm{goal}(X)$

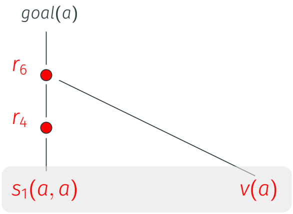
$r_4 = s_1(X,X) \rightarrow w(X)$

$r_5 = s_2(X,X) \rightarrow w(X)$

$r_6 = v(X) \wedge w(X) \rightarrow \mathrm{goal}(X)$

$$p(a), v(a), s_1(a,a), s_2(a,a), r(b,a), q(a), s_1(b,b), p(c)$$





Exemple

$r_1 = p(X) \rightarrow r(X,X)$

$r_2 = r(X,Y) \wedge q(Y) \rightarrow r(Y,X)$

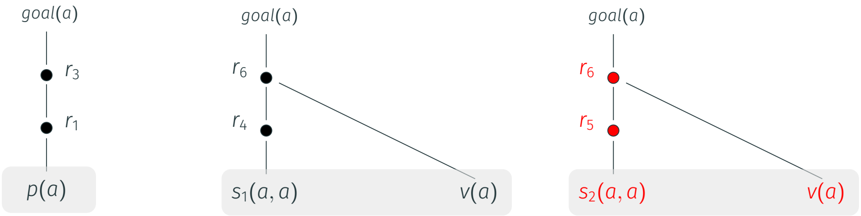
$r_3 = r(X,X) \wedge r(X,Y) \rightarrow \mathrm{goal}(X)$

$r_4 = s_1(X,X) \rightarrow w(X)$

$r_5 = s_2(X,X) \rightarrow w(X)$

$r_6 = v(X) \wedge w(X) \rightarrow \mathrm{goal}(X)$

$$p(a), v(a), s_1(a,a), s_2(a,a), r(b,a), q(a), s_1(b,b), p(c)$$

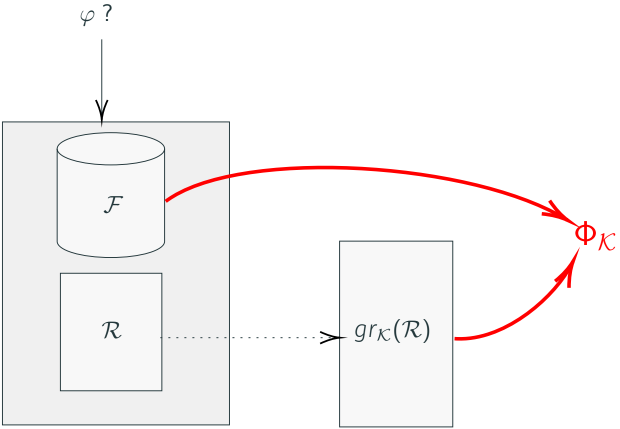


Réduction logique de Horn - *intuition*

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$

Grounding $gr_{\mathcal{K}}(\mathcal{R})$

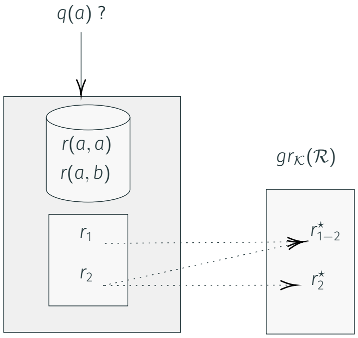
Unique variable propositionnelle $pr(\psi)$ pour tout fait ψ.



Réduction logique de Horn - *exemple*

$$r_1 = r(X,X) \rightarrow q(X) \qquad r_2 = r(X,Y) \rightarrow q(X)$$

$$r^\star_{1-2} = r(a,a) \rightarrow q(a) \qquad r^\star_2 = r(a,b) \rightarrow q(a)$$



1. Ajout clauses pour $\mathcal{F}$:
   - $\mathtt{pr}\,(r(a,a))$
   - $\mathtt{pr}\,(r(a,b))$
2. Ajout clauses pour $\mathcal{R}$:
   - $\mathtt{pr}(r(a,a)) \rightarrow \mathtt{pr}(q(a))$
   - $\mathtt{pr}(r(a,b)) \rightarrow \mathtt{pr}(q(a))$

Réduction logique de Horn - *formalisation*

$$\mathtt{hClause}(\psi) = \mathtt{pr}(\psi), \text{ pour tout } \psi \in \mathcal{F}.$$

$$\mathtt{hClause}(r^\star) = \bigwedge_{i=1}^{n} \mathtt{pr}(\beta_i(\bar{b}_i)) \rightarrow \mathtt{pr}(\alpha(\bar{a})), \text{ pour}$$

$$r^\star = \bigwedge_{i=1}^{n} \beta_i(\bar{b}_i) \rightarrow \alpha(\bar{a}) \in \mathtt{gr}_{\mathcal{K}}(\mathcal{R}).$$

Réduction logique de Horn - *formalisation*

$$\mathtt{hClause}(\psi) = \mathtt{pr}(\psi), \text{ pour tout } \psi \in \mathcal{F}.$$

$$\mathtt{hClause}(r^\star) = \bigwedge_{i=1}^{n} \mathtt{pr}(\beta_i(\bar{b}_i)) \rightarrow \mathtt{pr}(\alpha(\bar{a})), \text{ pour}$$

$$r^\star = \bigwedge_{i=1}^{n} \beta_i(\bar{b}_i) \rightarrow \alpha(\bar{a}) \in \mathtt{gr}_{\mathcal{K}}(\mathcal{R}).$$

Réduction Horn

$$\Phi_{\mathcal{K}} = \left(\bigwedge_{\psi \in \mathcal{F}} \mathtt{hClause}(\psi) \wedge \bigwedge_{r^\star \in \mathtt{gr}_{\mathcal{K}}(\mathcal{R})} \mathtt{hClause}(r^\star) \right)$$

Exemple - *problème MUS*

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$

Requête φ

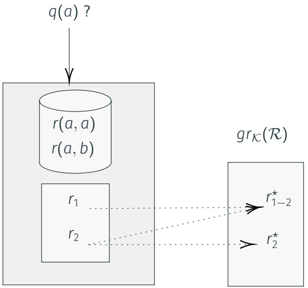
$$\text{Explanations}(\mathcal{K}, \varphi) \stackrel{?}{=} \text{MUSes}(\Phi_{\mathcal{K}} \land \neg \boldsymbol{pr}(\varphi))$$

Exemple - *problème MUS*

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ Requête φ

$$\boxed{\text{Explanations}(\mathcal{K}, \varphi) \stackrel{?}{=} \text{MUSes}(\Phi_{\mathcal{K}} \wedge \neg \boldsymbol{\mathit{pr}}(\varphi))}$$

$$r^{\star}_{1-2} = r(a, a) \rightarrow q(a) \qquad r^{\star}_{2} = r(a, b) \rightarrow q(a)$$



MUS

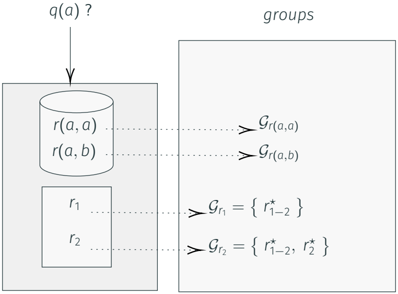
$\{ \neg \boldsymbol{\mathit{pr}}(q(a)), \mathtt{hClause}(r(a, a)), \mathtt{hClause}(r^{\star}_{1-2}) \}$

$\Downarrow$

Explications

$\{ r(a, a),\ r_1 \}$ $\{ r(a, a),\ r_2 \}$

Grouper les clauses ?



(group-)MUS

$\{\ \mathcal{G}_{r(a,a)},\ \mathcal{G}_{r_1}\ \}\qquad \{\ \mathcal{G}_{r(a,a)},\ \mathcal{G}_{r_2}\ \}$

$\Downarrow$

Explications

$\{\ r(a,a),\ r_1\ \}\qquad \{\ r(a,a),\ r_2\ \}$

Group-MUS et explications (1)

Definition

Soit une formule de Horn Φ, un ensemble de groupes $\{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$ sur Φ, et un ensemble de clauses $\mathcal{U}$, un *group-MUS* de $\langle \Phi, \mathcal{U} \rangle$ est un ensemble de groupes $\mathscr{G} \subseteq \{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$ tel que:

- $\mathcal{U} \cup \bigcup_{\mathcal{G}_i \in \mathscr{G}} \mathcal{G}_i$ est insatisfiable, et
- il n'existe pas $\mathcal{G}_j \in \mathscr{G}$ tel que $\mathcal{U} \cup (\mathscr{G} \setminus \mathcal{G}_j)$ est insatisfiable.

Group-MUS et explications (2)

On définit groups($\Phi_{\mathcal{K}}$) comme l'ensemble des groupes contenant:

- $\{\mathtt{hClause}(\psi)\}$, pour tout $\psi \in \mathcal{F}$.
- $\{\mathtt{hClause}(r^\star) \mid r^\star \in \mathtt{gr}_{\mathcal{K}}(r)\}$, pour tout $r \in \mathcal{R}$.

Group-MUS et explications (2)

On définit groups($\Phi_{\mathcal{K}}$) comme l'ensemble des groupes contenant:

- $\{\mathtt{hClause}(\psi)\}$, pour tout $\psi \in \mathcal{F}$.
- $\{\mathtt{hClause}(r^\star) \mid r^\star \in \mathbf{gr}_{\mathcal{K}}(r)\}$, pour tout $r \in \mathcal{R}$.

Théorème

Soit une KB $\mathcal{K}$ et un fait φ,

$\mathcal{K}' \subseteq \mathcal{K}$ est une explication de φ

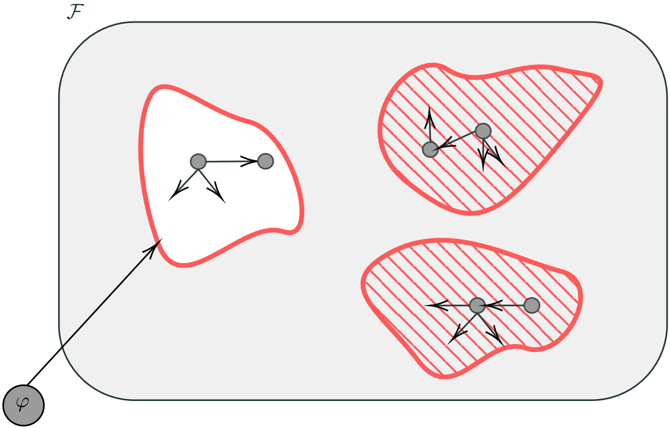
si et seulement si

groups($\Phi_{\mathcal{K}'}$) est un group-MUS de $\langle \Phi_{\mathcal{K}}, \{\neg \mathtt{hClause}(\varphi)\}\rangle$.

Approche par filtrage

Limites MUS

- Calcul des MUS coûteux.
- KB conséquente $\rightarrow$ explosion du nombre de clauses.



$\Rightarrow$ **Filtrer la base au préalable par rapport à la requête φ**

Pertinence

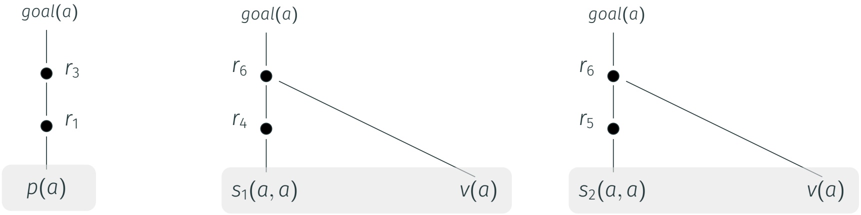
Definition

Soit une base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ et un fait φ, un fait $\psi \in \mathcal{F}$ est *pertinent* pour φ s'il existe $\langle \mathcal{R}', \mathcal{F}' \rangle \in \text{Explanations}(\mathcal{K}, \varphi)$ tel que $\psi \in \mathcal{F}'$.

Remarque

Relevant$(\mathcal{K}, \varphi)$ = union des faits pertinent pour φ par rapport à $\mathcal{K}$.

Exemple



$$\text{Relevant}(\mathcal{K}, goal(a)) = \{p(a), s_1(a,a), v(a), s_2(a,a)\}$$

FACT RELEVANCE($\mathcal{R}$)

INSTANCE: Un ensemble de faits $\mathcal{F}$, un fait φ et un fait $\psi \in \mathcal{F}$.
QUESTION: ψ est-il *pertinent* pour l'inférence de φ par rapport à $\langle \mathcal{R}, \mathcal{F} \rangle$?

Résultats théoriques - *Complexité des données*

Fact Relevance($\mathcal{R}$)

Instance: Un ensemble de faits $\mathcal{F}$, un fait φ et un fait $\psi \in \mathcal{F}$. Question: ψ est-il *pertinent* pour l'inférence de φ par rapport à $\langle \mathcal{R}, \mathcal{F} \rangle$?

Théorème

Fact Relevance($\mathcal{R}$) est NP-complet.

(réduction SAT)

Graphe des déclencheurs

Definition (Graphe des déclencheurs/Trigger graph)

Soit une KB $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$, alors TgGraph($\mathcal{K}$) = $\langle V, E \rangle$ est le graphe orienté tel que:

- $V = \mathrm{Ch}(\mathcal{K})$,
- $E \subseteq V \times V$, et $(\psi, \phi) \in E$ si il y a un déclencheur $\rho = (r, h)$ sur V tel que $\psi \in h(body(r))$ et $\phi \in h(head(r))$.

Approximation des faits pertinents

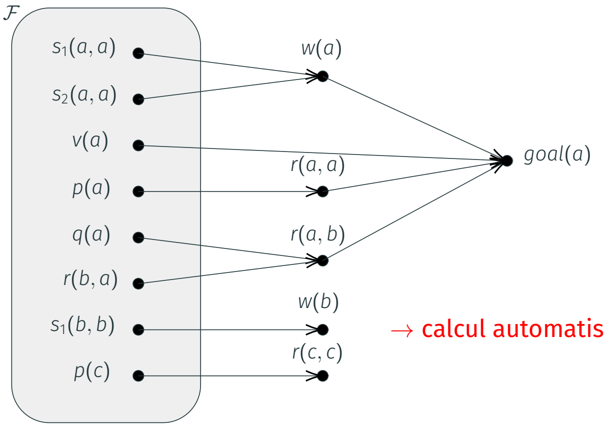
Pour un fait φ,

$$\mathrm{SupRelevant}(\mathcal{K}, \varphi) = \{\psi \in \mathcal{F} \mid \psi \text{ ancêtre de } \varphi \text{ dans } \mathrm{TgGraph}(\mathcal{K})\ \}$$

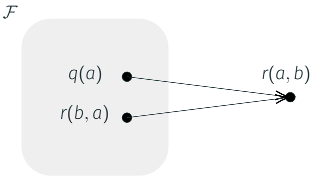
Exemple

$$p(X) \rightarrow r(X,X)$$
$$r(X,Y) \wedge q(Y) \rightarrow r(Y,X)$$
$$r(X,X) \wedge r(X,Y) \rightarrow \mathrm{goal}(X)$$

$$s_1(X,X) \rightarrow w(X)$$
$$s_2(X,X) \rightarrow w(X)$$
$$v(X) \wedge w(X) \rightarrow \mathrm{goal}(X)$$



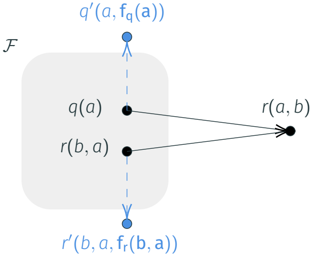
$\rightarrow$ calcul automatisé ?



Definition (Traduction)

Soit un ensemble de règles $\mathcal{R}$, $\mathtt{buildTgGraph}(\mathcal{R})$ est l'ensemble minimal contenant:

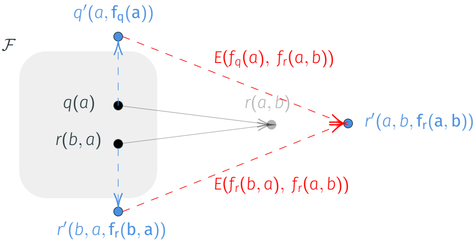
Calcul du graphe



Definition (Traduction)

Soit un ensemble de règles $\mathcal{R}$, $\mathtt{buildTgGraph}(\mathcal{R})$ est l'ensemble minimal contenant:

- $P(\bar{x}) \rightarrow P'(\bar{x}, f_P(\bar{x}))$, pour tout prédicat P de $\mathcal{R}$.



Definition (Traduction)

Soit un ensemble de règles $\mathcal{R}$, $\texttt{buildTgGraph}(\mathcal{R})$ est l'ensemble minimal contenant:

- $P(\bar{x}) \rightarrow P'(\bar{x}, f_P(\bar{x}))$, pour tout prédicat P de $\mathcal{R}$.
- $\bigwedge_{i=1}^{k} B'_i(\bar{x}_i, y_i) \rightarrow H'(\bar{x}, f_H(\bar{x})) \wedge \left(\bigwedge_{i=1}^{k} E(y_i, f_H(\bar{x}))\right)$, pour toute règle $r = \bigwedge_{i=1}^{k} B_i(\bar{x}_i) \rightarrow H(\bar{x}) \in \mathcal{R}$.

Calcul du filtrage

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$:

Calcul du filtrage

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$:

1. calcul du graphe des déclencheurs:
   $\rightarrow$ *chase* sur $\mathcal{K}' = \langle \texttt{buildTgGraph}(\mathcal{R}), \mathcal{F} \rangle$.
   $\rightarrow$ $\mathrm{TgGraph}(\mathcal{K}) = \{ \boldsymbol{E}(t, u) \mid \boldsymbol{E}(t, u) \in \mathrm{Ch}(\mathcal{K}') \}$.

Calcul du filtrage

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$:

1. calcul du graphe des déclencheurs:
   $\rightarrow$ *chase* sur $\mathcal{K}' = \langle \mathtt{buildTgGraph}(\mathcal{R}), \mathcal{F} \rangle$.
   $\rightarrow$ $\text{TgGraph}(\mathcal{K}) = \{E(t,u) \mid E(t,u) \in \text{Ch}(\mathcal{K}')\}$.

2. pour un fait φ et son terme fonctionnel t_φ, *traçage* dans le graphe:
   $\rightarrow$ *chase* sur $\mathcal{K}$"

$$\text{TgGraph}(\mathcal{K}) \cup \{\mathtt{Rel}(t_\varphi)\}$$

$$\mathtt{Rel}(x) \wedge E(y,x) \rightarrow \mathtt{Rel}(y)$$

$\rightarrow$ $\text{SupRelevant}(\mathcal{K}, \varphi) = \{\psi \mid \psi \in \mathcal{F} \text{ et } \mathcal{K}'' \models \mathtt{Rel}(t_\psi)\}$

Calcul du filtrage

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$:

1. calcul du graphe des déclencheurs:
   $\rightarrow$ *chase* sur $\mathcal{K}' = \langle \texttt{buildTgGraph}(\mathcal{R}), \mathcal{F} \rangle$.
   $\rightarrow$ $\mathrm{TgGraph}(\mathcal{K}) = \{ \texttt{E}(t, u) \mid \texttt{E}(t, u) \in \mathrm{Ch}(\mathcal{K}') \}$.

2. pour un fait φ et son terme fonctionnel t_φ, *traçage* dans le graphe:
   $\rightarrow$ *chase* sur $\mathcal{K}$"

$$\mathrm{TgGraph}(\mathcal{K}) \cup \{ \texttt{Rel}(t_\varphi) \}$$

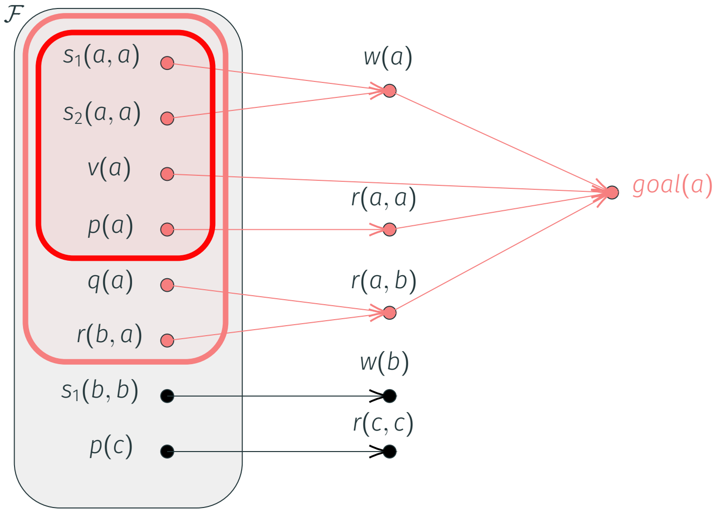
$$\texttt{Rel}(x) \wedge \texttt{E}(y, x) \rightarrow \texttt{Rel}(y)$$

   $\rightarrow$ $\mathrm{SupRelevant}(\mathcal{K}, \varphi) = \{ \psi \mid \psi \in \mathcal{F} \text{ et } \mathcal{K}'' \models \texttt{Rel}(t_\psi) \}$

Complexité

Filtrage calculable en **temps polynomial** en la taille de $\mathcal{F}$

Exemple



$$\text{SupRelevant}(\mathcal{K}, goal(a)) = \text{Relevant}(\mathcal{K}, goal(a)) \cup \{q(a), r(b,a)\}$$

Évaluation expérimentale

Benchmark et outils

Benchmark

$\Downarrow$

Sélection de requêtes à expliquer

$\Downarrow$

Évaluation du filtrage

Benchmark

Ontologies du dépôt d'Oxford (Biopax, LUBM)

- Normalisation.
- Génération de différents échantillons de données (100K,500K,1M).
- Traduction OWL $\rightarrow$ DLGP.
- Suppression des règles non-Datalog.

$\Downarrow$

Sélection de requêtes à expliquer

$\Downarrow$

Évaluation du filtrage

Benchmark

$\Downarrow$

Sélection de requêtes à expliquer

Génération de 100 **requêtes atomiques** *ground* parmi les inférences.

$\Downarrow$

Évaluation du filtrage

Benchmark

$\Downarrow$

Sélection de requêtes à expliquer

$\Downarrow$

Évaluation du filtrage

2 algorithmes (filtrage, sans filtrage) pour calculer les explications.

Outils

InteGraal grounding, graphe des déclencheurs, traçage

MARCO énumération des group-MUS

Algorithmes

entrées:

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ Ensemble de requêtes $\mathcal{Q}$

SANS FILTRAGE

FILTRAGE

entrées:

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ Ensemble de requêtes $\mathcal{Q}$

SANS FILTRAGE

1. **Grounding de $\mathcal{K}$.** *(InteGraal)*
2. **Pour tout $\varphi \in \mathcal{Q}$,** énumération des group-MUS de $\langle \Phi_{\mathcal{K}}, \neg \mathtt{hClause}(\varphi) \rangle$. *(MARCO)*

FILTRAGE

entrées:

Base de connaissances $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ Ensemble de requêtes $\mathcal{Q}$

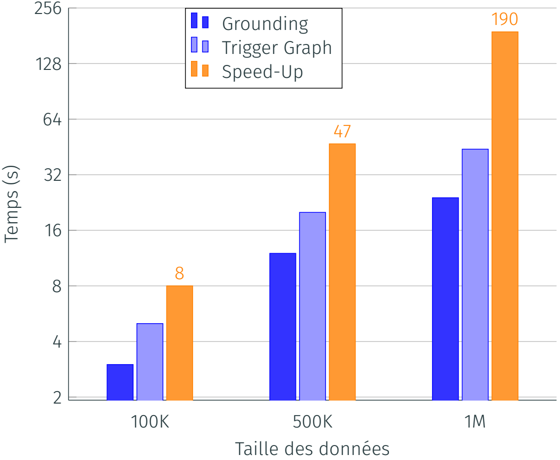
SANS FILTRAGE

1. Grounding de $\mathcal{K}$. *(InteGraal)*
2. Pour tout $\varphi \in \mathcal{Q}$, énumération des group-MUS de $\langle \Phi_{\mathcal{K}}, \neg \mathtt{hClause}(\varphi) \rangle$. *(MARCO)*

FILTRAGE

1. Calcul du graphe des déclencheurs de $\mathcal{K}$. *(InteGraal)*
2. Pour tout $\varphi \in \mathcal{Q}$,
   - traçage $\rightarrow$ SupRelevant$(\mathcal{K}, \varphi)$. *(InteGraal)*
   - grounding de $\langle \mathcal{R}, \text{SupRelevant}(\mathcal{K}, \varphi) \rangle$. *(InteGraal)*
   - énumération des group-MUS de $\langle \Phi_{\langle \mathcal{R}, \text{SupRelevant}(\mathcal{K}, \varphi) \rangle}, \neg \mathtt{hClause}(\varphi) \rangle$ *(MARCO)*

Résultats préliminaires - *Biopax*

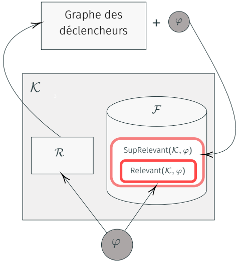


	Timeouts (300s)		
	100K	*500K*	*1M*
Filtrage	0	0	7
Sans Filtrage	0	8	89

	100K	*500K*	*1M*
#SupRelevant	17	22	41
#explications	17	21	36
|explications|	2	3	5

Conclusion et perspectives

Conclusion



1. Explications d'inférences en Datalog.
2. Réduction énumération (group-)MUS.
3. Filtrage de la base de faits:
   - $\rightarrow$ pertinence par rapport à une requête.
   - $\rightarrow$ graphe des déclencheurs.

$\rightarrow$ Résultats expérimentaux
$\rightarrow$ Datalog disjonctif
$\rightarrow$ Inférence probabiliste

Merci de votre attention !

Complexité Pertinence - *appartenance*

Machine de Turing non-déterministe avec un oracle pour Fact Entailment($\mathcal{R}$).
Accepte sur l'entrée $\langle \mathcal{F}, \varphi, \psi \rangle$ si et seulement si ψ est pertinent pour l'inférence de φ avec $\langle \mathcal{R}, \mathcal{F} \rangle$.

1. Devine non-déterministiquement un ensemble $\mathcal{M} \subseteq \mathcal{F}$ tel que $\psi \in \mathcal{M}$. $\rightarrow$ NP
2. Vérifie avec l'oracle sur Fact Entailment($\mathcal{R}$) si $\langle \mathcal{R}, \mathcal{M} \rangle \models \varphi$. Si ce n'est pas le cas, alors elle *rejette*. $\rightarrow$ oracle P
3. Vérifie la minimalité de $\mathcal{M}$. S'il existe $\mathcal{M}' \subset \mathcal{M}$ avec $|\mathcal{M}'| = |\mathcal{M}| - 1$ tel que $\langle \mathcal{R}, \mathcal{M}' \rangle \models \varphi$ (avec l'oracle) alors elle *rejette*, sinon elle *accepte*. $\rightarrow$ nb. linéaire d'appels à l'oracle P

Fact Entailment($\mathcal{R}$) est dans P (complexité des données).
Machine nécessite un oracle sur P et $\mathsf{NP}^{\mathsf{P}} = \mathsf{NP}$.
Donc, Fact Relevance($\mathcal{R}^{\wedge}$) est dans NP.

SAT $\leq_p$ Fact Relevance($\mathcal{R}$)

Formule CNF $\Phi = C_1 \wedge \ldots \wedge C_m$ définie en utilisant les variables $\{V_1, \ldots, V_n\}$.

Base de connaissances déterministe $\mathcal{K} = \langle \mathcal{R}, \mathcal{F} \rangle$ avec les prédicats d'arité 0 Source et Target.

Φ est satisfiable si et seulement si Source est pertinent pour l'inférence de Target.

$$
\begin{aligned}
\mathcal{F} &= \{\mathrm{T}(v_i), \mathrm{F}(v_i) \mid 1 \leq i \leq n\} \cup \{\mathsf{Source}\} \\
\mathcal{R} &= \{\mathrm{T}(v_k) \rightarrow \mathrm{T}(c_i) \mid 1 \leq i \leq m \text{ et pour tout littéral positif } V_k \text{ dans } C_i\} \ \cup && (1)\\
&\quad \{\mathrm{F}(v_k) \rightarrow \mathrm{T}(c_i) \mid 1 \leq i \leq m \text{ et pour tout littéral négatif } \neg V_k \text{ dans } C_i\} \ \cup && (2)\\
&\quad \{\mathrm{Source} \wedge \bigwedge_{i=1}^{m} \mathrm{T}(c_i) \rightarrow \mathsf{Target}\} \ \cup && (3)\\
&\quad \{\mathrm{T}(x) \wedge \mathrm{F}(x) \rightarrow \mathsf{Target}\} && (4)
\end{aligned}
$$

Complexité Pertinence - *difficulté* (2)

(Correction) On montre que si Source est pertinent pour l'inférence de Target avec $\mathcal{K}$ alors la formule CNF Φ est satisfiable.

ⓐ Par hypothèse il y a un ensemble minimal $\mathcal{A} \subseteq \mathcal{F}$ tel que Source $\in \mathcal{A}$ et $\langle \mathcal{R}, \mathcal{A} \rangle \models$ Target.

ⓑ On montre d'abord que pour toute constante v_i figurant dans $\mathcal{A}$, on a $\mathsf{T}(v_i) \in \mathcal{A}$ si et seulement si $\mathsf{F}(v_i) \notin \mathcal{A}$. En effet, supposons que $\mathsf{T}(v_i), \mathsf{F}(v_i) \in \mathcal{A}$ pour une constante v_i. Alors, avec la règle (4) on a $\langle \mathcal{R}, \{\mathsf{T}(v_i), \mathsf{F}(v_i)\} \rangle \models$ Target. Ce qui contredit le fait que $\mathcal{A}$ est un ensemble minimal qui, avec $\mathcal{R}$, infère Target.

ⓒ Soit $\sigma_{\mathcal{A}}$ l'assignation telle que $\sigma_{\mathcal{A}}(V_i) = \mathit{Vrai}$ si et seulement si $\mathsf{T}(v_i) \in \mathcal{A}$ pour toute variable V_i dans Φ et sa constante correspondante v_i dans $\mathcal{K}$.

ⓓ Par ⓑ, on sait que (4) ne peut pas être appliquée sur $\mathcal{A}$. Donc, la règle (3) a été appliquée pour inférer Target.
Donc, $\langle \mathcal{R}, \mathcal{A} \rangle \models \mathsf{T}(c_i)$ pour tout $1 \leq i \leq m$.

ⓔ Pour $1 \leq i \leq m$ on sait que $\mathsf{T}(c_i)$ a été inféré soit par une règle dans (1) ou dans (2).
   - Si une règle de la forme (1) a été appliquée alors il y a une constante v_k tel que $\mathsf{T}(v_k) \in \mathcal{A}$ et $\mathsf{T}(v_k) \to \mathsf{T}(c_i) \in \mathcal{R}$. Soit V_k la variable correspondant à v_k et C_i la clause correspondant à c_i dans Φ, alors par la construction des règles dans (1) on a $V_k \in C_i$ et par ⓒ on sait que $\sigma_{\mathcal{A}}(V_k) = \mathit{Vrai}$, donc $\sigma_{\mathcal{A}}(C_i) = \mathit{Vrai}$.
   - Si une règle de la forme (2) a été appliquée alors il y a une constante v_k tel que $\mathsf{F}(v_k) \in \mathcal{A}$ et $\mathsf{F}(v_k) \to \mathsf{T}(c_i) \in \mathcal{R}$. Soit V_k la variable correspondant à v_k et C_i la clause correspondant à c_i dans Φ, alors par la construction des règles dans (2) on a $\neg V_k \in C_i$ et par ⓒ on sait que $\sigma_{\mathcal{A}}(V_k) = \mathit{Faux}$ donc $\sigma_{\mathcal{A}}(C_i) = \mathit{Vrai}$.

Donc, $\sigma_{\mathcal{A}}(C_i) = \mathit{Vrai}$ pour toute clause $C_i \in \Phi$, ainsi $\sigma(\Phi) = \mathit{Vrai}$ et Φ est satisfiable.

Complexité Pertinence - *difficulté* (3)

(Complétude) On montre que si Φ est satisfiable alors Source est pertinent pour Target avec $\mathcal{K}$.

ⓐ Par hypothèse il existe une assignation σ telle que σ est un modèle de Φ. Soit l'ensemble de faits $\mathcal{A} = \{\mathsf{Source}\} \cup \{\mathsf{T}(v_i) \mid \sigma(V_i) = \mathit{Vrai}\} \cup \{\mathsf{F}(v_i) \mid \sigma(V_i) = \mathit{Faux}\} \subseteq \mathcal{F}$.

ⓑ Par hypothèse, on a $\sigma(C_i) = \mathit{Vrai}$ pour toute clause C_i dans Φ et on distingue deux cas.
- Il y a un littéral positif $V_k \in C_i$ tel que $\sigma(V_k) = \mathit{Vrai}$. Dans ce cas, soient v_k et c_i les constantes correspondant à V_k et C_i respectivement. Par ⓐ on a $\mathsf{T}(v_k) \in \mathcal{A}$, et $\mathsf{T}(v_k) \rightarrow \mathsf{T}(c_i) \in \mathcal{R}$ par la construction des règles dans (1).
- Il y a un littéral négatif $\neg V_k \in C_i$ tel que $\sigma(V_k) = \mathit{Faux}$. Soient v_k et c_i les constantes correspondant à V_k et C_i respectivement. Par ⓐ on a $\mathsf{F}(v_k) \in \mathcal{A}$, et $\mathsf{F}(v_k) \rightarrow \mathsf{T}(c_i) \in \mathcal{R}$ par la construction des règles dans (2).

Dans les deux cas, $\langle \mathcal{R}, \mathcal{A} \rangle \models \mathsf{T}(c_i)$ pour tout $1 \leq i \leq m$.

ⓒ Soit $\mathcal{A}'$ un ensemble minimal de $\mathcal{A}$ tel que $\langle \mathcal{R}, \mathcal{A}' \rangle \models$ Target. On sait que (3) et (4) sont les seules règles pouvant inférer Target. En revanche, (4) ne peut pas être appliquée sur $\mathcal{A}'$. En effet, $\mathcal{A}$ est construit à partir d'une assignation σ et ainsi soit $\mathsf{T}(v_i)$ ou $\mathsf{F}(v_i)$ appartient à $\mathcal{A}'$ pour toute constante v_i correspondant à une variable V_i dans Φ. Donc la règle (3) a été appliquée pour dériver Target.

ⓓ Pour conclure, notons que pour appliquer (3) l'atome Source doit appartenir à $\mathcal{A}'$. Donc, Source est pertinent pour Target.

(Polynomialité) On montre que la taille de $\mathcal{K}$ est polynomiale en fonction de la taille de Φ. Dans l'ensemble de faits $\mathcal{F}$ on a $2n + 1$ atomes avec n étant le nombre de variables propositionnelles dans Φ. Ensuite l'ensemble de règles contient au plus $k \times m + 2$ règles où m est le nombre de clauses dans Φ et k le plus grand nombre de littéraux dans une clause. Donc l'instance $\mathcal{K}$ est polynomiale par rapport à Φ et la traduction peut être calculée en temps polynomial.