

θ -subsomption distribuée

Hippolyte Léger, **Dominique Bouthinon**, Mustapha Lebbah,
Hanene Azzag

Journée scientifique du cluster MAGI - Juillet 2018

Introduction

Programmation Logique Inductive

La θ -subsumption

Résolution

Approche naïve

État de l'art

Subsumer

Passage à l'échelle

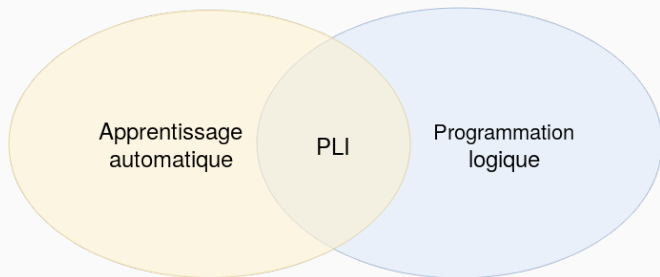
Modèle acteurs

Réseau d'acteurs pour la θ -subsumption

Expérimentations

Conclusion et perspectives

Introduction



Étant donnés

- Un ensemble d'exemples positifs
- Un ensemble d'exemples négatifs

Trouver une hypothèse qui **couvre** tous les exemples positifs et aucun exemple négatif.

Illustration

E ⁺	E ⁻
E ⁺ ₁ : parent(jean,claire), parent(claire,paul), masculin(jean)	E ⁻ ₁ : parent(olga,pierre), parent(pierre,marie), feminin(olga)
E ⁺ ₂ : parent(amine,marc), masculin(amine), parent(marc,julie)	E ⁻ ₂ : parent(olga,pierre), parent(paul,pierre)

Solution possible

grand_pere(X, Y) : parent(X, A), parent(A, Y), masculin(X)

Définition de la θ -subsumption¹

Une clause C θ -subsume une clause D , noté $C \preceq_{\theta} D$ si et seulement si il existe une substitution θ telle que $C\theta \subseteq D$.

1. [Robinson, 1965]

Définition de la θ -subsumption¹

Une clause C θ -subsume une clause D , noté $C \preceq_{\theta} D$ si et seulement si il existe une substitution θ telle que $C\theta \subseteq D$.

Exemple

- $C : \text{parent}(X, A), \text{parent}(A, Y)$
- $D : \text{parent}(\text{olga}, \text{pierre}), \text{parent}(\text{pierre}, \text{marie}), \text{feminin}(\text{olga})$

On a $C \preceq_{\theta} D$ avec $\theta = \{X/\text{olga}, A/\text{pierre}, Y/\text{marie}\}$.

1. [Robinson, 1965]

Retour sur l'exemple de PLI(1)

E^+	E^-
E^+_1 : parent(jean, claire), parent(claire, paul), masculin(jean)	E^-_1 : parent(olga, pierre), parent(pierre, marie), feminin(olga)
E^+_2 : parent(amine, marc), masculin(amine), parent(marc, julie)	E^-_2 : parent(olga, pierre), parent(paul, pierre)

$grand_parent(X, Y) : parent(X, A), parent(A, Y)$

Retour sur l'exemple de PLI(2)

E^+	E^-
E^+_1 : parent(jean, claire), parent(claire, paul), masculin(jean)	E^-_1 : parent(olga, pierre), parent(pierre, marie), feminin(olga)
E^+_2 : parent(amine, marc), masculin(amine), parent(marc, julie)	E^-_2 : parent(olga, pierre), parent(paul, pierre)

$grand_pere(X, Y) : parent(X, A), parent(A, Y), masculin(X)$

Prorogammation Logique Inductive (PLI)

- Apprentissage supervisé + programmation logique
- Applications :
 - bioinformatique
 - traitement du langage
 - théorie des jeux

θ -subsumption

- Test de couverture le plus utilisé en PLI
- Problème NP-complet dans le cas général²
- Les méthodes actuelles ne peuvent pas traiter de très grosses clauses

2. [Kapur and Narendran, 1986]

Résolution

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D?$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$\theta = \{X/a, Y/c\}$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$$\theta = \{X/a, Y/c, Z/d, U/e\}$$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$\theta = \{X/a, Y/c, Z/d, U/e\} \dots \textit{backtracking!}$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$\theta = \{X/a, Y/c\} \dots \textit{backtracking!}$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$\theta = \{X/b, Y/c\}$

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$\theta = \{X/b, Y/c, Z/a, U/e\}$

Approche naïve

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$$\theta = \{X/b, Y/c, Z/a, U/e, T/d, V/e\}$$

Approche naïve

Soient les clauses suivantes

- $C : p(X, Y), q(X, Z, U), q(Z, T, V)$
- $D : p(a, c), p(b, c), q(a, d, e), q(b, a, e)$

$C \preceq_{\theta} D$?

Construire une substitution θ en unifiant un à un les atomes de C avec les atomes de D .

$$\theta = \{X/b, Y/c, Z/a, U/e, T/d, V/e\}$$

Pire cas : $O(|D|^{|C|})$ opérations.

État de l'art

Bilan des méthodes de résolution les plus efficaces :

Nom	Eff. en temps	Eff. en mémoire	Parallélisme
Clique maximale ³	-	-	non
<i>Django</i> ⁴	+	-	non
Approche complète ⁵	-	-	non
<i>ReSumEr2</i> ⁶	++	+	non
<i>Subsumer</i> ⁷	++	++	non

3. [Scheffer et al., 1997]

4. [Maloberti and Sebag, 2001]

5. [Ferilli et al., 2003]

6. [Kuzelka and Zelezný, 2008]

7. [Santos and Muggleton, 2010]

Principe fondamental de *Subsumer*⁸ : la séparation de clause

Soit $C : p(X, Y), q(X, Z, U), q(Z, T, V)$

8. [Santos and Muggleton, 2010]

Principe fondamental de *Subsumer*⁸ : la séparation de clause

Soit $C : p(X, Y), q(X, Z, U), q(Z, T, V)$

Si X est fixé, alors :

$C_1 : p(X, Y)$ et $C_2 : q(X, Z, U), q(Z, T, V)$ indépendantes.

Donc, $C \preceq_{\theta} D \iff C_1 \preceq_{\theta} D \wedge C_2 \preceq_{\theta} D$

8. [Santos and Muggleton, 2010]

Principe fondamental de *Subsumer*⁸ : la séparation de clause

Soit $C : p(X, Y), q(X, Z, U), q(Z, T, V)$

Si X est fixé, alors :

$C_1 : p(X, Y)$ et $C_2 : q(X, Z, U), q(Z, T, V)$ indépendantes.

Donc, $C \preceq_{\theta} D \iff C_1 \preceq_{\theta} D \wedge C_2 \preceq_{\theta} D$

Si Z est fixé, alors :

$C_{21} : q(X, Z, U)$ et $C_{22} : q(Z, T, V)$ indépendantes.

Donc, $C_2 \preceq_{\theta} D \iff C_{21} \preceq_{\theta} D \wedge C_{22} \preceq_{\theta} D$

8. [Santos and Muggleton, 2010]

Passage à l'échelle

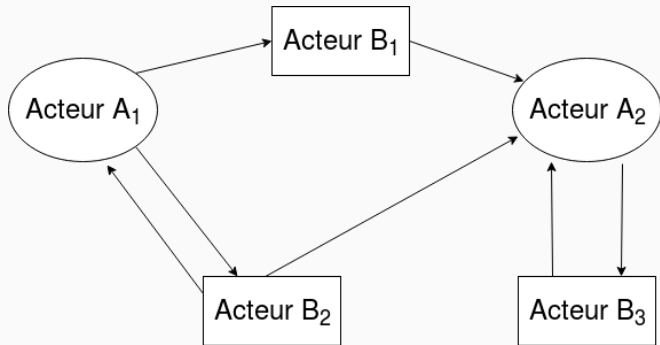
Modèle acteurs :

Un acteur est une entité pouvant :

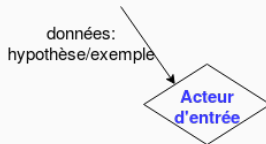
- effectuer des opérations
- maintenir un état interne
- communiquer avec d'autres acteurs

de manière concurrente.

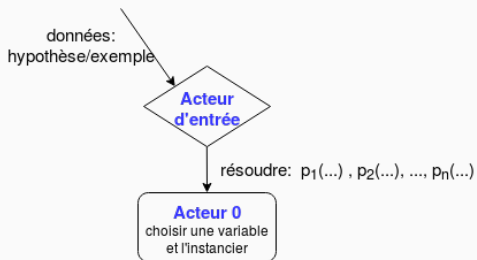
Exemple de réseau d'acteurs :



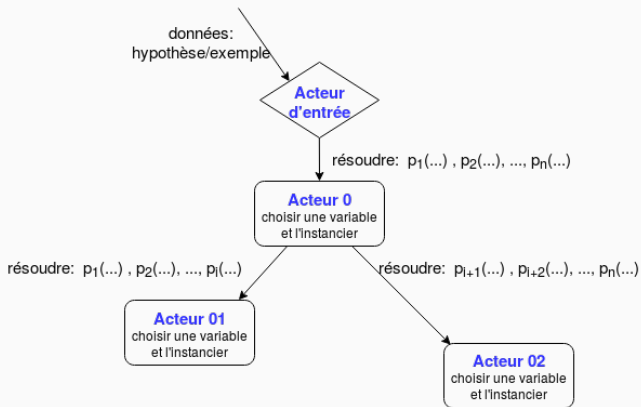
Réseau d'acteurs pour la θ -subsomption



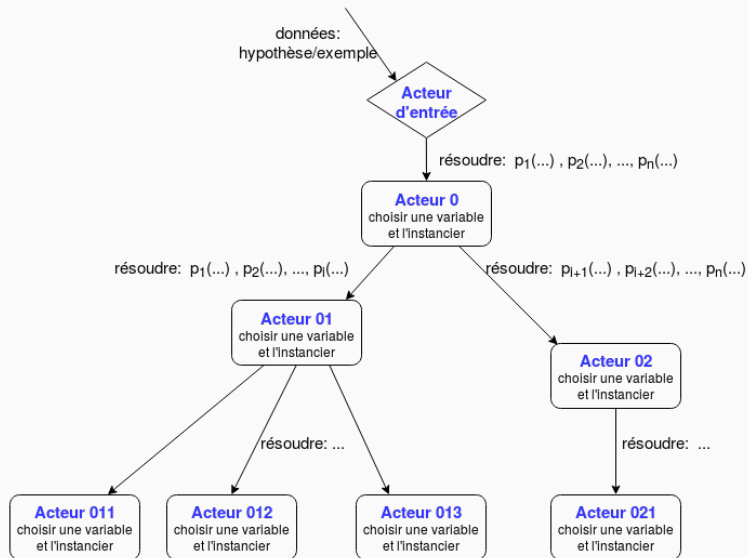
Réseau d'acteurs pour la θ -subsomption



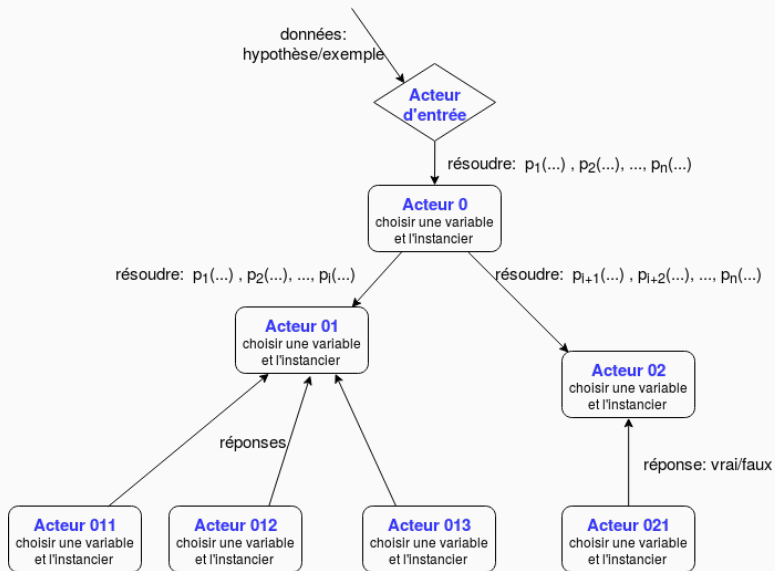
Réseau d'acteurs pour la θ -subsomption



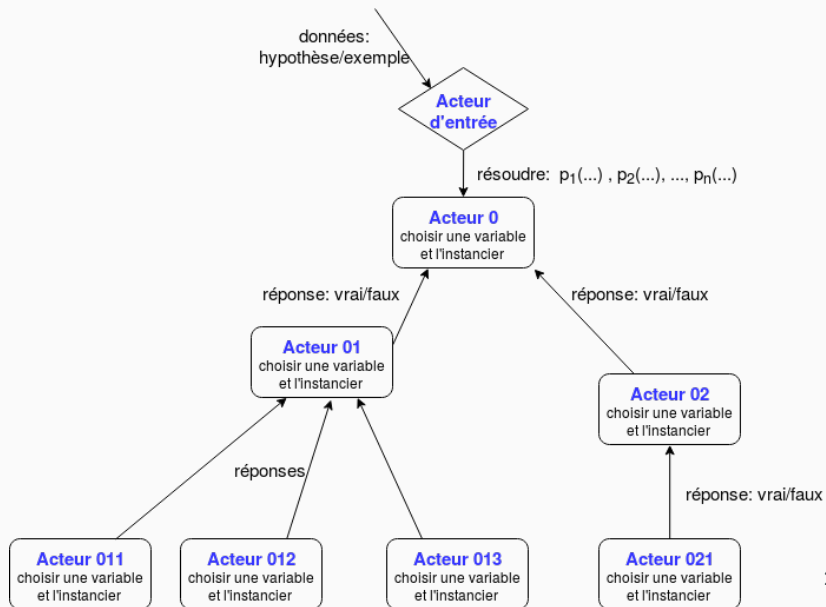
Réseau d'acteurs pour la θ -subsomption



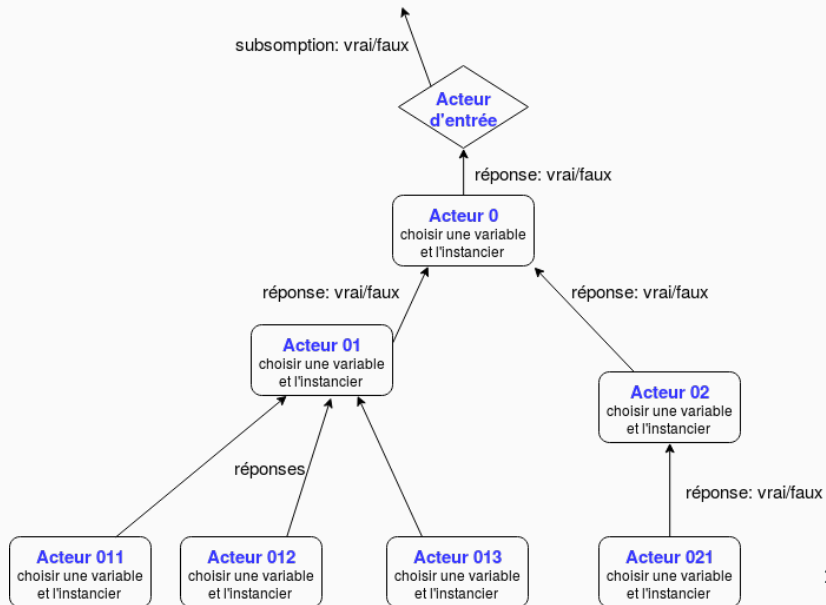
Réseau d'acteurs pour la θ -subsomption



Réseau d'acteurs pour la θ -subsomption



Réseau d'acteurs pour la θ -subsomption



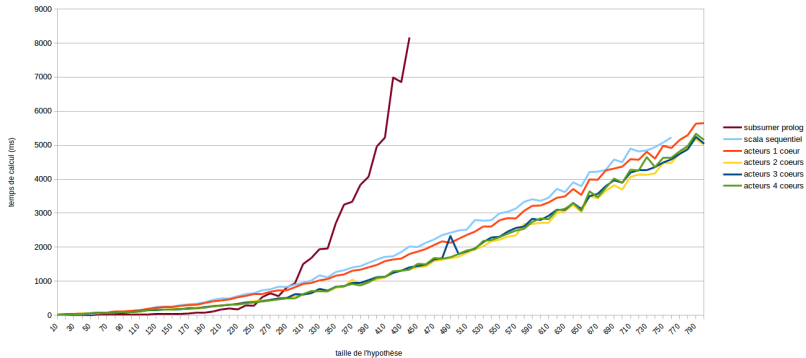
Génération de données

- plusieurs couples hypothèse/exemple
- hypothèses à taille variable
- nombre total de variables possible variable

Pour l'instant, les tests ont été effectués sur une seule machine

Expérimentations

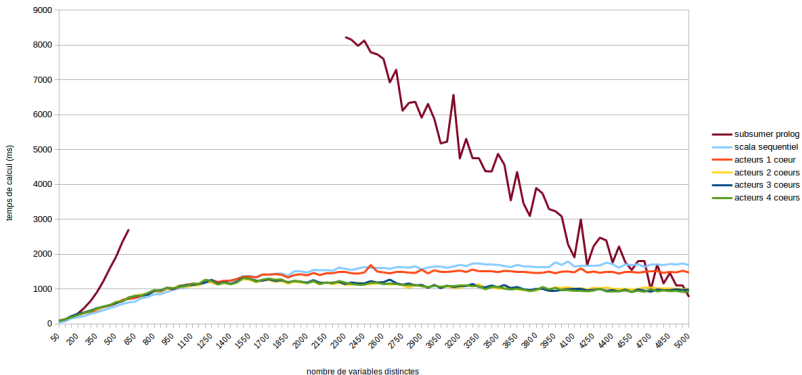
temps de subsumption de différentes méthodes
en faisant varier la taille de l'hypothèse



Expérimentations

temps de subsomption de différentes méthodes

en faisant varier le nombre total de variables distinctes



Conclusion et perspectives

Conclusion

- Expérimentations de θ -subsumption parallélisée
- Amélioration des temps de calculs et de la taille des données traitées

Perspectives

- Plus de parallélisation
- Passage en mode distribué : élargir la répartition des calculs sur plusieurs machines
- Comparaison avec d'avantage de méthodes de θ -subsumption



Ferilli, S., Mauro, N., Basile, T. M. A., and Esposito, F. (2003).

AI*IA 2003 : Advances in Artificial Intelligence : 8th Congress of the Italian Association for Artificial Intelligence, Pisa, Italy, September 2003. Proceedings, chapter A Complete Subsumption Algorithm, pages 1–13.





Springer Berlin Heidelberg, Berlin, Heidelberg.



Kapur, D. and Narendran, P. (1986).

8th International Conference on Automated Deduction : Oxford, England, July 27–August 1, 1986 Proceedings, chapter NP-completeness of the set unification and matching problems, pages 489–495.

Springer Berlin Heidelberg, Berlin, Heidelberg.

-  Kuzelka, O. and Zelezný, F. (2008).
A restarted strategy for efficient subsumption testing.
Fundam. Inform., 89(1) :95–109.
-  Maloberti, J. and Sebag, M. (2001).
Inductive Logic Programming : 11th International Conference, ILP 2001 Strasbourg, France, September 9–11, 2001 Proceedings, chapter θ -Subsumption in a Constraint Satisfaction Perspective, pages 164–178.
Springer Berlin Heidelberg, Berlin, Heidelberg.
-  Robinson, J. A. (1965).
A machine-oriented logic based on the resolution principle.
J. ACM, 12(1) :23–41.
-  Santos, J. and Muggleton, S. (2010).

Subsumer : A Prolog theta-subsumption engine.

In Hermenegildo, M. and Schaub, T., editors, *Technical Communications of the 26th International Conference on Logic Programming*, volume 7 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 172–181, Dagstuhl, Germany. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.



Scheffer, T., Herbrich, R., and Wysotzki, F. (1997). ***Inductive Logic Programming : 6th International Workshop, ILP-96 Stockholm, Sweden, August 26–28, 1996 Selected Papers***, chapter **Efficient θ -subsumption based on graph algorithms**, pages 212–228. Springer Berlin Heidelberg, Berlin, Heidelberg.