Égalité avec symboles non interprétés

David Delahaye

Faculté des Sciences David. Delahaye@lirmm.fr

Master Informatique M2 2021-2022

Définition

- Appelée également théorie libre de l'égalité, cette théorie donne un sens au prédicat d'égalité « = » en présence de symboles de fonctions d'arité quelconque dont le sens n'est pas défini.
- Les termes qui interviennent dans cette théorie sont les termes du premier ordre (appartenant à \mathcal{T}). On rappelle la définition de \mathcal{T} , à savoir le plus petit ensemble t.q. :

```
Si x ∈ V alors x ∈ T;
Si f ∈ S<sub>F</sub> d'arité n et t<sub>1</sub>,..., t<sub>n</sub> ∈ T, alors f(t<sub>1</sub>,..., t<sub>n</sub>) ∈ T.
Dù V ≡ ensemble de variables d'individu x, y, etc., et S<sub>F</sub> ≡ ensemble de symboles de fonctions f, g, etc.
```

Définition

- Appelée également théorie libre de l'égalité, cette théorie donne un sens au prédicat d'égalité « = » en présence de symboles de fonctions d'arité quelconque dont le sens n'est pas défini.
- Les termes qui interviennent dans cette théorie sont les termes du premier ordre (appartenant à \mathcal{T}). On rappelle la définition de \mathcal{T} , à savoir le plus petit ensemble t.q. :
 - Si $x \in \mathcal{V}$ alors $x \in \mathcal{T}$;
 - ▶ Si $f \in \mathcal{S}_{\mathcal{F}}$ d'arité n et $t_1, \ldots, t_n \in \mathcal{T}$, alors $f(t_1, \ldots, t_n) \in \mathcal{T}$.
 - où $V \equiv$ ensemble de variables d'individu x, y, etc., et $S_F \equiv$ ensemble de symboles de fonctions f, g, etc.

Contraintes élémentaires

- Les contraintes élémentaires de cette théorie sont soit des égalités, soit des différences entre des termes.
- Par exemple :

$$x = f(y, z)$$
 $g(x) \neq h(y)$ $x = y$ $f(x) = f(b)$

On va rechercher un modèle de ces contraintes élémentaires.
 Elles sont donc reliées par un « et » logique.
 Comme pour le problème SMT, les variables sont implicitement existentiellement quantifiées.

Quels axiomes pour cette théorie?

 La théorie est définie à partir de trois axiomes et d'un schéma d'axiomes (congruence) :

```
(réflexivité) \forall x.x = x

(symétrie) \forall x, y.x = y \Rightarrow y = x

(transitivité) \forall x, y, z.x = y \land y = z \Rightarrow x = z

(congruence) Pour tout f \in \mathcal{S}_{\mathcal{F}} d'arité n : \forall x_1, \dots, x_n, y_1, \dots, y_n. x_1 = y_1 \land \dots \land x_n = y_n \Rightarrow f(x_1, \dots, x_n) = f(y_1, \dots, y_n)
```

Le problème

ullet Le problème consiste à décider de la satisfiabilité d'une conjonction ${\cal C}$ de contraintes élémentaires de la forme :

$$\bigwedge_{i} t_{i} \bowtie u_{i}, \text{ où } \bowtie \in \{=, \neq\}$$

• Comme dit précédemment, la formule est implicitement existentiellement quantifiée et on décide de la formule suivante

$$\exists \vec{x}. \bigwedge_{i} t_{i} \bowtie u_{i}$$
, où $\bowtie \in \{=, \neq\}$ et $\vec{x} = \bigcup_{i} Var(t_{i}) \cup Var(u_{i})$

Avec $Var(t) \equiv$ ensemble des variables du terme t.

Le problème

ullet Le problème consiste à décider de la satisfiabilité d'une conjonction ${\cal C}$ de contraintes élémentaires de la forme :

$$\bigwedge_{i} t_{i} \bowtie u_{i}, \text{ où } \bowtie \in \{=, \neq\}$$

 Comme dit précédemment, la formule est implicitement existentiellement quantifiée et on décide de la formule suivante :

$$\exists \vec{x}. \bigwedge_{i} t_{i} \bowtie u_{i}$$
, où $\bowtie \in \{=, \neq\}$ et $\vec{x} = \bigcup_{i} Var(t_{i}) \cup Var(u_{i})$

Avec $Var(t) \equiv$ ensemble des variables du terme t.

Algorithme de congruence closure

- Algorithme initiallement décrit par Shostak en 1978.
- Algorithme pour déterminer la satisfiabilité d'une conjonction (∧) d'égalités (=) et d'inégalités (≠) avec des symboles de fonctions non interprétés.

Principe de l'algorithme

- On réalise la clôture congruente des égalités puis on regarde si la nouvelle relation obtenue est compatible avec l'ensemble des inégalités.
- On ne va pas générer une clôture congruente complète (qui est potentiellement infinie) mais uniquement sur la partie des termes intervenant dans la formule initiale.

Algorithme de congruence closure

- Algorithme initiallement décrit par Shostak en 1978.
- Algorithme pour déterminer la satisfiabilité d'une conjonction (∧) d'égalités (=) et d'inégalités (≠) avec des symboles de fonctions non interprétés.

Principe de l'algorithme

- On réalise la clôture congruente des égalités puis on regarde si la nouvelle relation obtenue est compatible avec l'ensemble des inégalités.
- On ne va pas générer une clôture congruente complète (qui est potentiellement infinie) mais uniquement sur la partie des termes intervenant dans la formule initiale.

Algorithme de congruence closure

Soit F une conjonction d'égalités et d'inégalités avec des symboles de fonctions non interprétés :

$$F = (\bigwedge_{i=1}^m s_i = t_i) \wedge (\bigwedge_{j=m+1}^n s_j \neq t_j)$$

Soit S l'ensemble des égalités et inégalités dans F.

Soit T l'ensemble des termes et sous-termes dans F.

Algorithme de congruence closure

On construit une partition de T de la façon suivante :

• Mettre initialement tous les termes et sous-termes dans leur propre classe de congruence :

$$\{\{t\}\} \mid t \in T\}$$

- 2 Pour tout $1 \le i \le m$:
 - Avec $s_i = t_i$, fusionner les classes de s_i et t_i .
 - Propager la nouvelle congruence avec les règles de symétrie, transitivité et congruence.

Algorithme de congruence closure

- ullet La partition construite sur ${\cal T}$ induit une relation congruente \sim sur ${\cal T}$.
- Cette relation est une congruence car elle satisfait les axiomes d'une relation d'équivalence (refléxivité, symétrie et transitivité) et respecte aussi la propriété de congruence.
- Une clôture congruente d'une relation R est la plus petite relation congruente qui contient R.
- La relation \sim est la clôture congruente qui contient toutes les égalités dans la formule initiale F.
- D'où le nom d'algorithme de congruence closure.

Algorithme de congruence closure

Shostak en 1978 a démontré le théorème suivant :

F est satisfiable ssi $\not\exists s_i, t_i \in T$ t.q. $s_i \sim t_i$ et $(s_i \neq t_i) \in S$.

On rappelle que :

$$F = (\bigwedge_{i=1}^m s_i = t_i) \wedge (\bigwedge_{j=m+1}^n s_j \neq t_j)$$

Soit S l'ensemble des égalités et inégalités dans F.

Soit T l'ensemble des termes et sous-termes dans F.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

Partition initiale

$$\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$$

- Imposer f(a,b) = a:
 - $\{\{a, f(a,b)\}, \{b\}, \{f(f(a,b),b)\}\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

Partition initiale :

$$\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$$

- Imposer f(a, b) = a: { $\{a, f(a, b)\}, \{b\}, \{f(f(a, b), b)\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

- Partition initiale :
 - $\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$
- Imposer f(a,b) = a:
 - $\{\{a, f(a,b)\}, \{b\}, \{f(f(a,b),b)\}\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

- Partition initiale :
 - $\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$
- Imposer f(a, b) = a: { $\{a, f(a, b)\}, \{b\}, \{f(f(a, b), b)\}\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

- Partition initiale :
 - $\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$
- Imposer f(a, b) = a: { $\{a, f(a, b)\}, \{b\}, \{f(f(a, b), b)\}\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule insatisfiable

$$f(a,b) = a \wedge f(f(a,b),b) \neq a$$

- Partition initiale :
 - $\{\{a\}, \{b\}, \{f(a,b)\}, \{f(f(a,b),b)\}\}$
- Imposer f(a, b) = a: $\{\{a, f(a, b)\}, \{b\}, \{f(f(a, b), b)\}\}$
- $a \sim f(a, b)$, donc $f(a, b) \sim f(f(a, b), b)$ (congruence) : $\{\{a, f(a, b), f(f(a, b), b)\}, \{b\}\}$

La partition donne $f(f(a,b),b) \sim a$ mais la formule initiale contient l'inégalité $f(f(a,b),b) \neq a$.

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), b)\}, \{$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): { $\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$

Il n'y a aucune inégalité qui contredit la relation \sim .

Formule satisfiable

$$a = b \wedge b = c \wedge g(f(a), b) = g(f(c), a) \wedge f(a) \neq b$$

- Partition initiale : $\{\{a\}, \{b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer a = b: $\{\{a, b\}, \{c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- Imposer b = c: $\{\{a, b, c\}, \{f(a)\}, \{f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $b \sim c$, donc $f(a) \sim f(c)$ (congruence): $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b)\}, \{g(f(c), a\}\}\}$
- $f(a) \sim f(c)$ et $b \sim a$, donc $g(f(a), b) \sim g(f(c), a)$ (congruence) : $\{\{a, b, c\}, \{f(a), f(c)\}, \{g(f(a), b), g(f(c), a\}\}$