

HOL4-Beagle, de l'ordre supérieur vers le premier ordre

Thibault Gauthier

7 janvier 2014

Deux types de prouveurs

	HOL4	Beagle
Type	Interactif	Automatique
Expressivité	Ordre supérieur	Premier ordre
Sûreté	Petit noyau	Code assez long

Énoncé du problème

Problème Voilà deux prouveurs internes à HOL4.

- Metis : premier ordre + traduction de l'ordre supérieur vers le premier ordre
- Cooper : arithmétique

Énoncé du problème

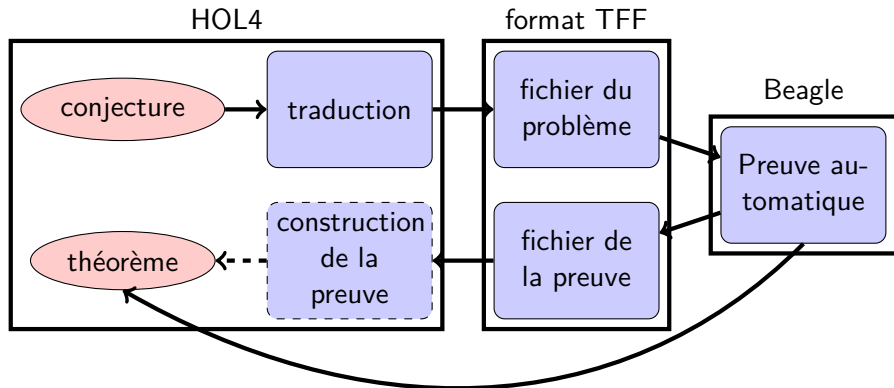
Problème Voilà deux prouveurs internes à HOL4.

- Metis : premier ordre + traduction de l'ordre supérieur vers le premier ordre
- Cooper : arithmétique

Solution Un prouveur externe.

- Beagle : premier ordre et arithmétique

Schéma d'interaction



1 Introduction

- Deux types de prouveurs
- Énoncé du problème
- Schéma d'interaction

2 Traduction vers le premier ordre

- Monomorphisation
- λ -lifting
- Défonctionnalisation

3 Conclusion

- Qualités et limites

Ordre de la traduction vers le premier ordre

- 1 Monomorphisation
- 2 Négation de la conclusion
- 3 Mise en forme normale conjonctive
- 4 λ -lifting
- 5 Élimination des booléens
- 6 Mise sous forme d'un ensemble de clauses
- 7 Défonctionnalisation

Monomorphisation

Instanciation des types polymorphes (a, \dots) par des types monomorphes $(int, bool, \dots)$.

Problème

Thm 1 : $\forall x : a. D \times 0$ Thm 2 : $C = \lambda x : a. D \times 0$

Conjecture : $C \ 2$

Monomorphisation

Instanciation des types polymorphes (a, \dots) par des types monomorphes ($int, bool, \dots$).

Problème

Thm 1 : $\forall x : a. D \times 0$ Thm 2 : $C = \lambda x : a. D \times 0$

Conjecture : $C \ 2$

Unification de $C : a \rightarrow int \rightarrow bool$ et de $C : int \rightarrow int \rightarrow bool$

Thm 1 : $\forall x : a. D \times 0$ Thm 2 : $C = \lambda x : int. D \times 0$

Conjecture : $C \ 2$

Monomorphisation

Instanciation des types polymorphes (a, \dots) par des types monomorphes $(int, bool, \dots)$.

Problème

Thm 1 : $\forall x : a. D \times 0$ Thm 2 : $C = \lambda x : a. D \times 0$

Conjecture : $C \ 2$

Unification de $C : a \rightarrow int \rightarrow bool$ et de $C : int \rightarrow int \rightarrow bool$

Thm 1 : $\forall x : a. D \times 0$ Thm 2 : $C = \lambda x : int. D \times 0$

Conjecture : $C \ 2$

Unification de $D : a \rightarrow int \rightarrow bool$ et de $D : int \rightarrow int \rightarrow bool$

Thm 1 : $\forall x : int. D \times 0$ Thm 2 : $C = \lambda x : int. D \times 0$

Conjecture : $C \ 2$

λ -lifting

Problème

Thm 1 : $\forall x. D \times 0$ Thm 2 : $C = \lambda x. D \times 0$

Conjecture : $C \ 2$

Négation de la conclusion

$$\{\forall x. D \times 0, C = \lambda x. D \times 0, \neg(C \ 2)\}$$

λ -lifting

Problème

Thm 1 : $\forall x. D \times 0$ Thm 2 : $C = \lambda x. D \times 0$

Conjecture : $C \ 2$

Négation de la conclusion

$$\{\forall x. D \times 0, C = \lambda x. D \times 0, \neg(C \ 2)\}$$

λ -lifting :

$$C = \lambda x. D \times 0 \rightsquigarrow \exists f. (\forall x. f \ x = D \times 0) \wedge C = f$$

Mise sous forme d'un ensemble de clauses

$$\{\forall x. D \times 0, \forall x. f \ x = D \times 0, C = f, \neg(C \ 2)\}$$

Défonctionnalisation

Soit App vérifiant $App\ f\ x = f\ x$. On effectue une défonctionnalisation lorsqu'une fonction non-arithmétique :

- est quantifiée universellement
$$!h. h\ x\ y \rightsquigarrow !h. App\ (App\ h\ x)\ y$$
- a le même type qu'une fonction quantifiée universellement
- a un nombre d'arguments auxquelles la fonction est appliquée qui varie
$$\{h\ x\ y\ z,\ h\ x = j\} \rightsquigarrow \{App\ (App\ (h\ x)\ y)\ z,\ h\ x = j\}$$

Défonctionnalisation

Soit App vérifiant $App\ f\ x = f\ x$. On effectue une défonctionnalisation lorsqu'une fonction non-arithmétique :

- est quantifiée universellement
 $!h. h\ x\ y \rightsquigarrow !h. App\ (App\ h\ x)\ y$
- a le même type qu'une fonction quantifiée universellement
- a un nombre d'arguments auxquelles la fonction est appliquée qui varie
 $\{h\ x\ y\ z, h\ x = j\} \rightsquigarrow \{App\ (App\ (h\ x)\ y)\ z, h\ x = j\}$

Défonctionnalisation

$$\{\forall x. D\ x\ 0, \forall x. f\ x = D\ x\ 0, C = f, \neg(C\ 2)\}$$

$$\{\forall x. D\ x\ 0, \forall x. App\ f\ x = D\ x\ 0, C = f, \neg(C\ 2)\}$$

Qualités et limites de l'interaction HOL4-Beagle

Qualités :

- est correcte (préserve la satisfaisabilité)
- prouve 80% des conjectures prouvées par Metis auxquelles on a enlevé les lemmes arithmétiques
- utilise un format de communication répandu

Limites :

- est incomplète et (ne préserve pas l'insatisfaisabilité)
- ne cherche pas automatiquement des théorèmes aidant à prouver la conjecture
- ne rejoue pas (encore) la preuve