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

Thibault Gauthier

December 2, 2013

Lieu du stage: Canberra

Deux façon différentes de démontrer des théorèmes

	Prouveur interactif	Prouveur automatique
Prouveurs	HOL4, Coq, ...	Beagle, SPASS, ...
Expressivité	Ordre supérieur.	Premier ordre.
Efficacité	Guidé.	Automatique.
Sûreté	Petit noyau.	Code assez long.

Enoncé du problème

Problème Voilà deux prouveurs internes à HOL4.

- Metis: ordre supérieur
- Cooper: arithmétique

Enoncé du problème

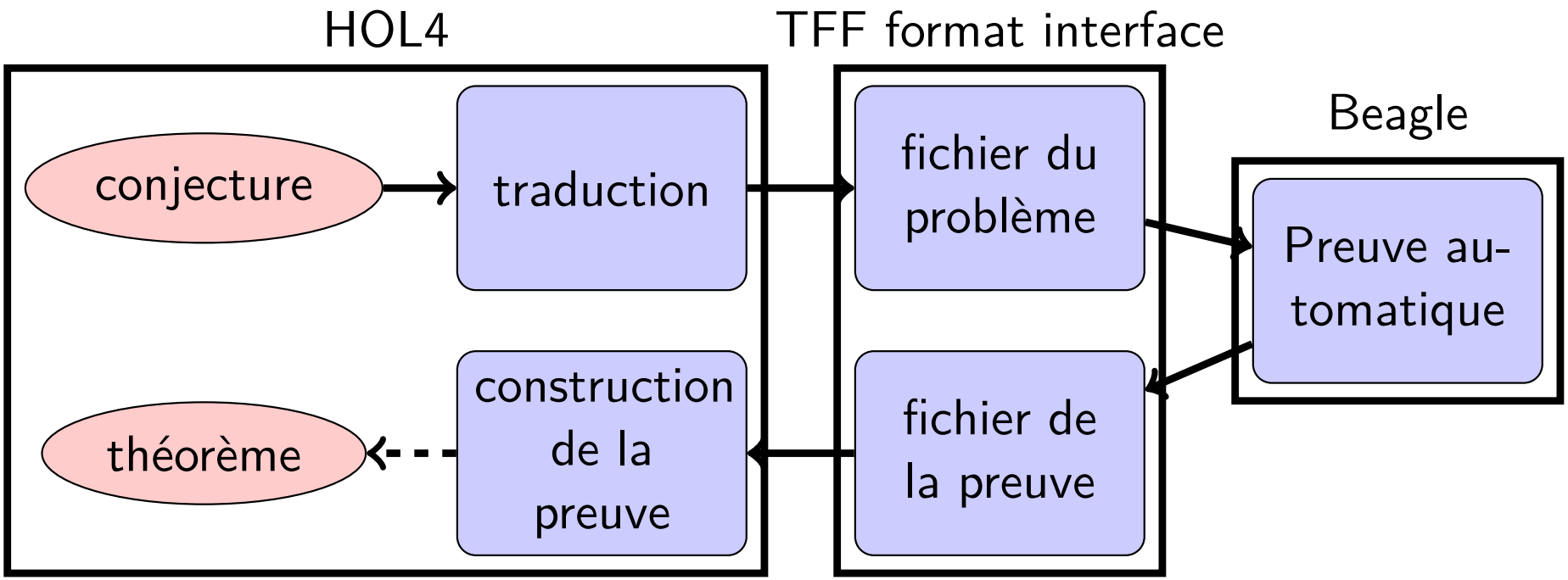
Problème Voilà deux prouveurs internes à HOL4.

- Metis: ordre supérieur
- Cooper: arithmétique

Solution Un prouveur externe.

- Beagle: premier ordre et arithmétique

BEAGLE_TAC


HOL4
TFF format interface
Beagle
conjecture
traduction
fichier du problème
Preuve automatique
théorème
construction de la preuve
fichier de la preuve

Les avantages du format TFF(TPTP)

Voici une courte description du format TFF:

- Utilisation: répandu
- Formules: arithmétiques typées du premier ordre.

Ses avantages sont:

- Lisible par un humain
- Lisible par d'autres prouveurs automatiques

HOL4

Voici les caractéristiques de HOL4:

- Prouveur interactif
- Logique: ordre supérieur et types polymorphes
- Language: écrit en SML
- Sûreté: type [thm] abstrait

Une preuve facile

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_i}{A \vdash B \Rightarrow (A \wedge B)}\Rightarrow_i}{\vdash A \Rightarrow (B \Rightarrow (A \wedge B))}\Rightarrow_i$$

```
(* forward proof *)
val th1 = ASSUME ``A:bool``;
val th2 = ASSUME ``B:bool``;
val th3 = CONJ th1 th2;
val th4 = DISCH ``B:bool`` th3;
val th5 = DISCH ``A:bool`` th4;
```

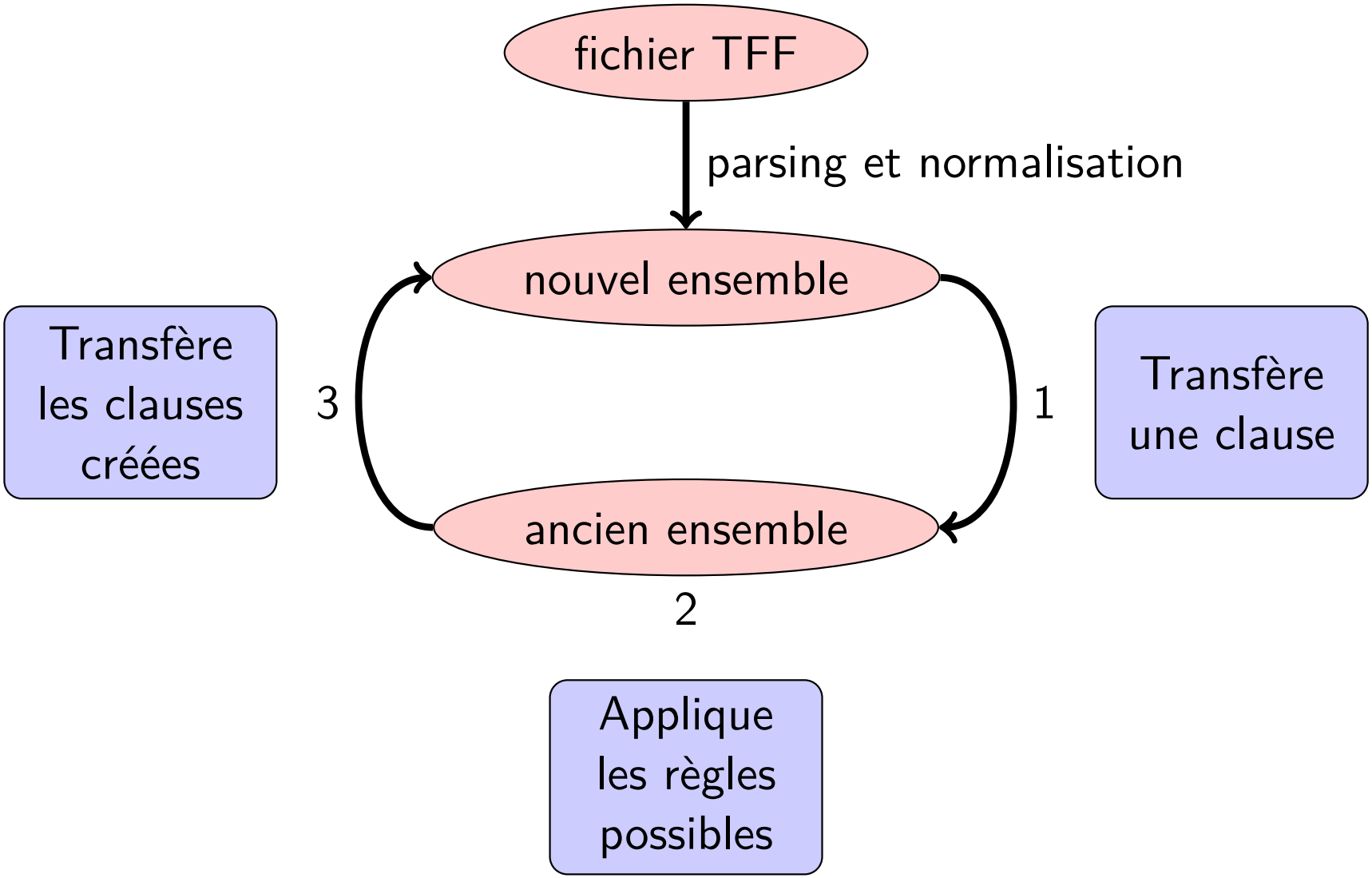
```
(* backward proof *)
g(`A ==> B ==> A /\ B `);
e(DISCH_TAC);
e(DISCH_TAC);
e(CONJ_TAC);
e(ACCEPT_TAC th1);
e(ACCEPT_TAC th2);
```

Beagle

Voici les caractéristiques de Beagle:

- Prouveur automatique
- Logique: premier ordre et types monomorphes
- Raisonnement: premier ordre et arithmétique
- Réponse: insatisfaisable, satisfaisable ou inconnu

Une preuve par saturation



Ordre de la traduction vers le premier ordre

Voici les étapes de la traduction:

1. Monomorphisation
2. Négation de la conclusion (Preuve par l'absurde)
3. Mise en forme normale conjonctive (Arrive plusieurs fois)
4. λ-lifting
5. Elimination des booléens
6. Mise sous forme d'un ensemble de clauses
7. Défonctionnalisation
8. Injection dans les entiers
9. Instantiation des variables booléennes quantifiées

Motivation

Instanciation des types polymorphes de HOL4 par des types monomorphes.

Problème

Théorème: $\forall x : a.\ C\ x\ x$
Conjecture: $C\ 42\ 42$

Unification du type des constantes $C : a \to a \to bool$ et $C : num \to num \to bool$

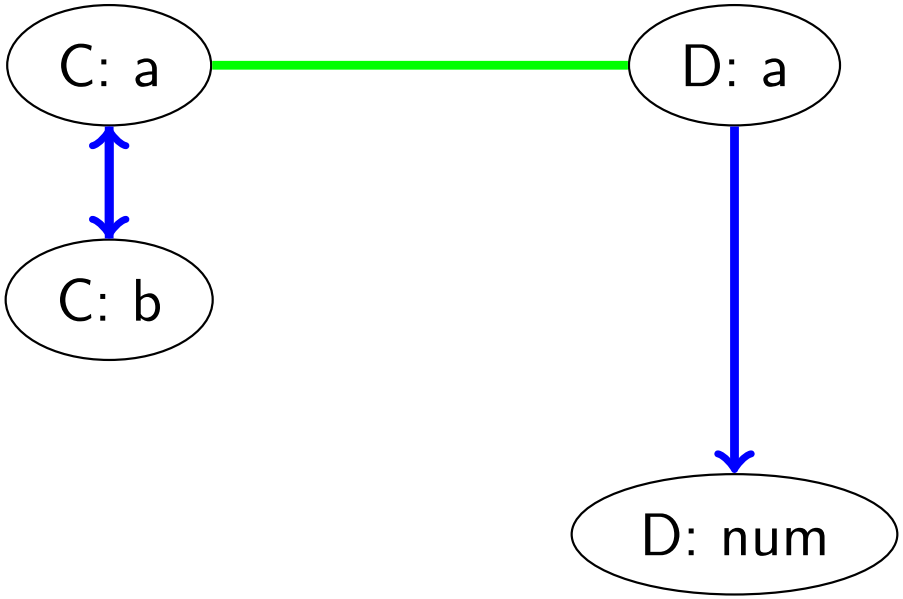
Nouveau problème

Théorème: $\forall x : num.\ C\ x\ x$
Conjecture: $C\ 42\ 42$

Graphe de dépendance

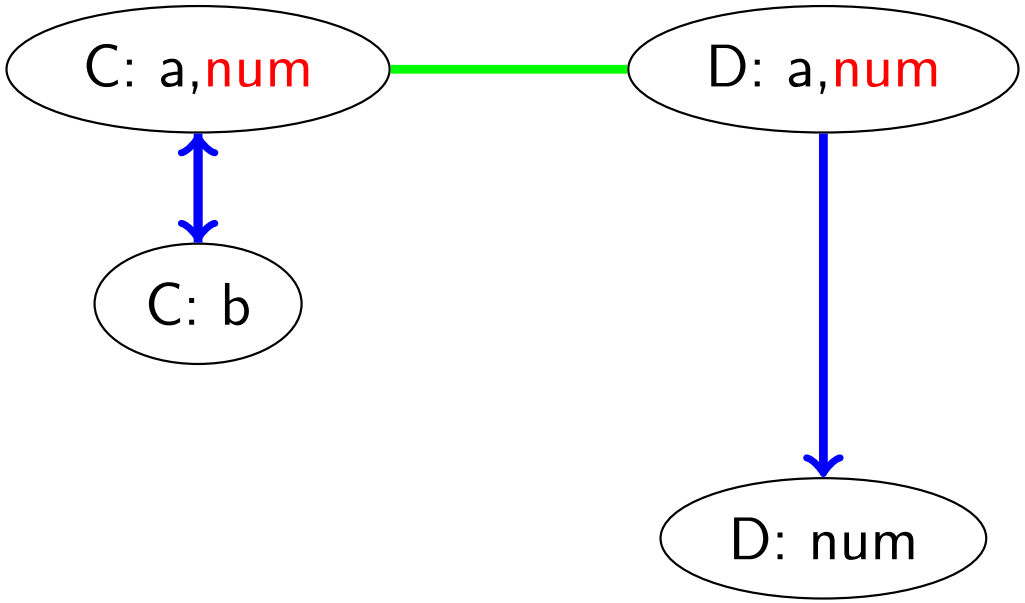
Problème

Théorème 1: $\forall x : a.\ C\ x \Rightarrow D\ x$
Théorème 2: $\forall x : b.\ C\ x$
Conjecture : $D\ 42$



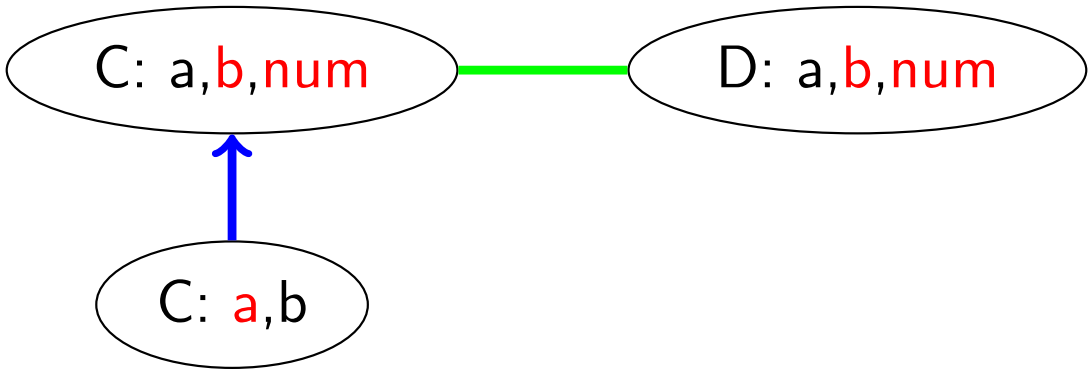
Exemple 1: Une co-instanciation

La flèche de substitution à droite induit une co-instanciation des constantes du premier théorème:



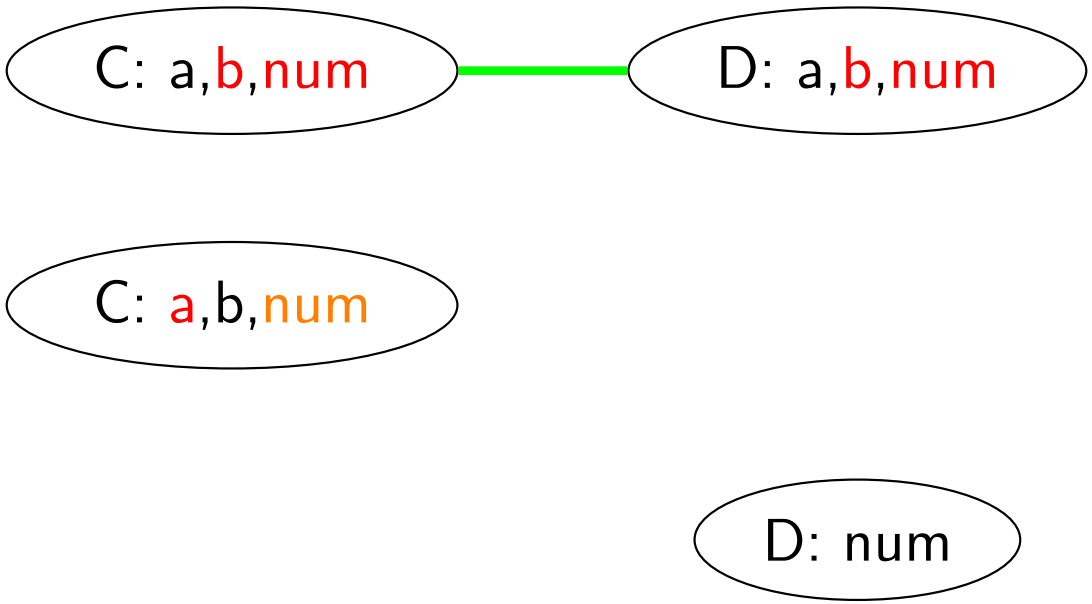
Exemple 1: Toutes les co-instanciations en parallèles

Nous nommerons $\mathcal{T}$ cette transformation du graphe.



Exemple 1: Répétition des co-instanciations

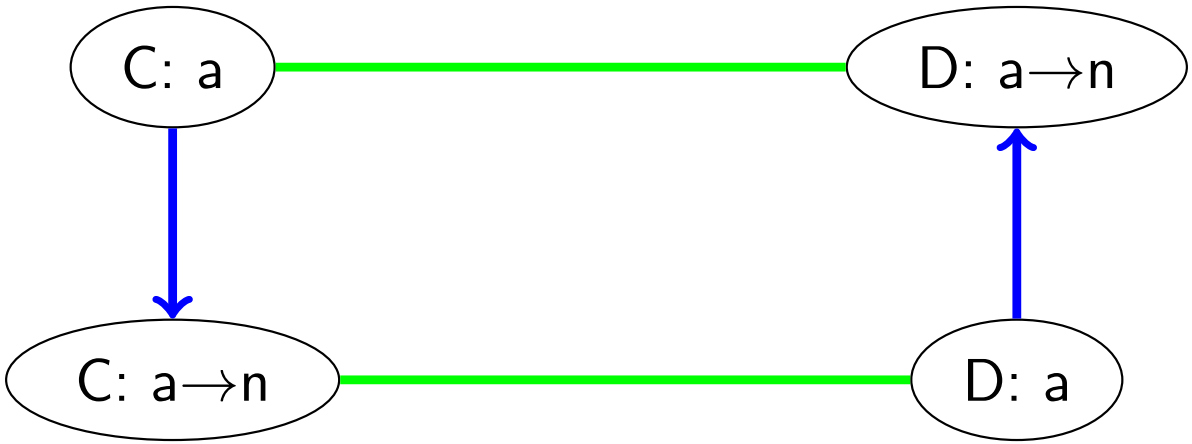
La transformation $\mathcal{T}$ précédente peut être répétée sur le nouveau graphe de dépendance que nous avons obtenu.



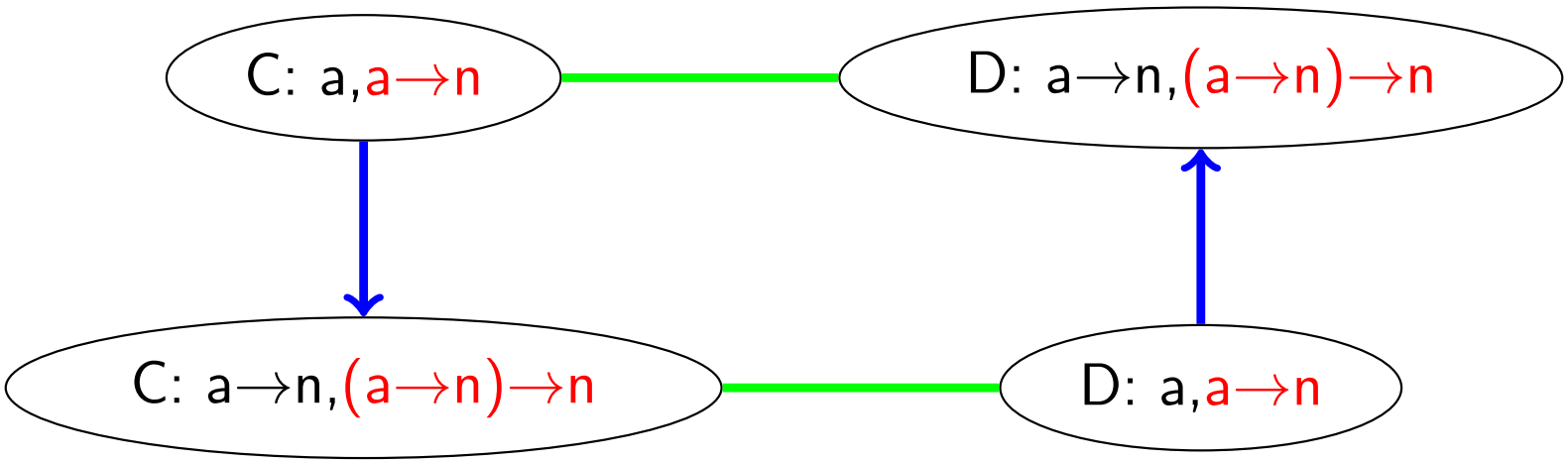
Ce graphe est un point fixe pour $\mathcal{T}$.

Exemple 2: Un exemple sans point fixe

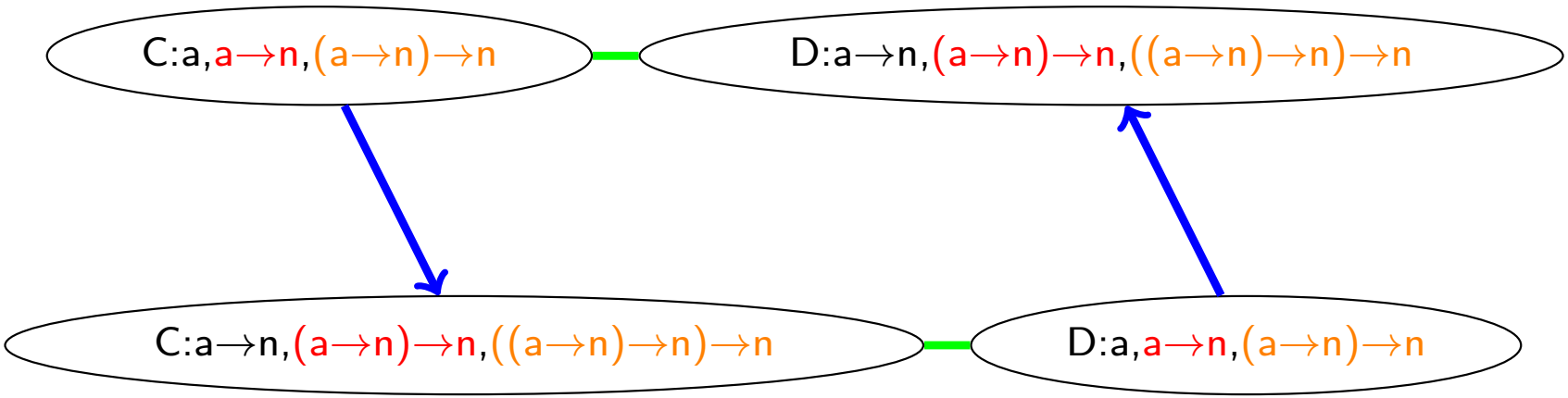
Recommençons avec un autre graphe.



Exemple 2: Un exemple sans point fixe



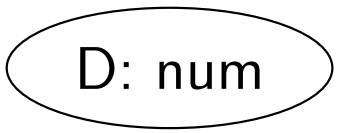
Exemple 2: Un exemple sans point fixe



La transformation $\mathcal{T}$ ne trouve pas de point fixe.

Instanciation des théorèmes données par l'utilisateur

1. Terminaison: un point fixe ou une borne arbitraire
2. Extraction d'un ensemble de substitutions pour chaque théorème



3. Instanciation

Ordre de la traduction vers le premier ordre

Voici les étapes de la traduction:

1. Monomorphisation
2. Négation de la conclusion (Preuve par l'absurde)
3. Mise en forme normale conjonctive (Arrive plusieurs fois)
4. λ-lifting
5. Elimination des booléens
6. Mise sous forme d'un ensemble de clauses
7. Défonctionnalisation
8. Injection dans les entiers
9. Instantiation des variables booléennes quantifiées

λ-lifting

Cette étape élimine les λ-abstractions restantes.

$$P\ (\lambda x.x + 1)$$

$$\exists g.\ (\forall x.\ g\ x = x + 1) \wedge P\ g$$

Elimination des booléens

Supposons que la formule booléenne $!x.\ x = 0$ soit l'argument d'une fonction f.

$$P(!x.\ x = 0)$$

$$((!x.\ x = 0) \Rightarrow P\ \mathit{true}) \wedge (\neg(!x.\ x = 0) \Rightarrow P\ \mathit{false})$$

Défonctionnalisation

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

- est quantifiée

$$!h.\ h\ x\ y = 0$$

$$!h.\ App\ (App\ h\ x)\ y = 0$$

- a le même type qu'une fonction quantifiée
- a un nombre d'arguments qui varie

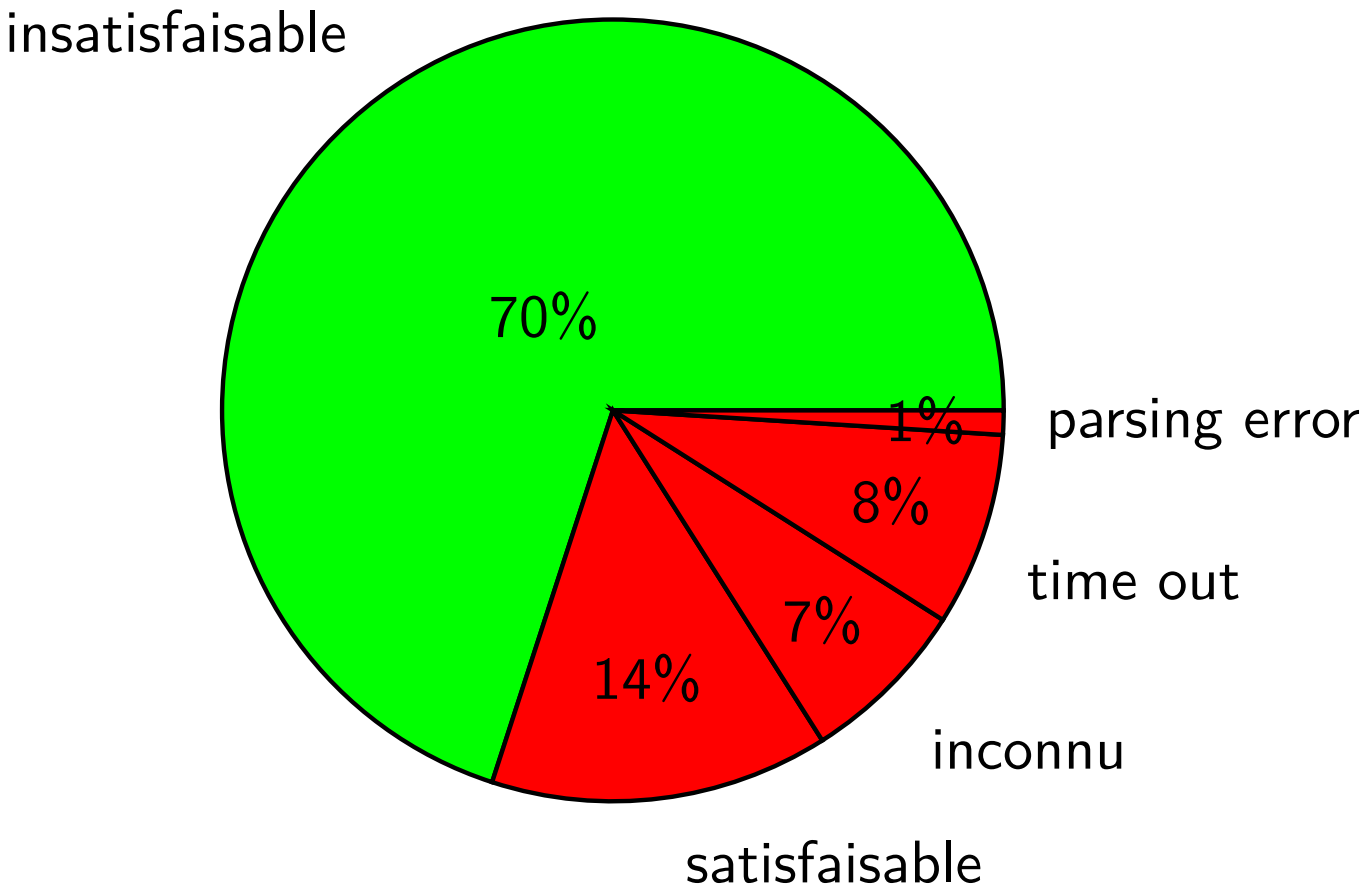
$$h\ x\ y\ z \wedge h\ x = g$$

$$App\ (App\ (h\ x)\ y)\ z \wedge h\ x = g$$

.

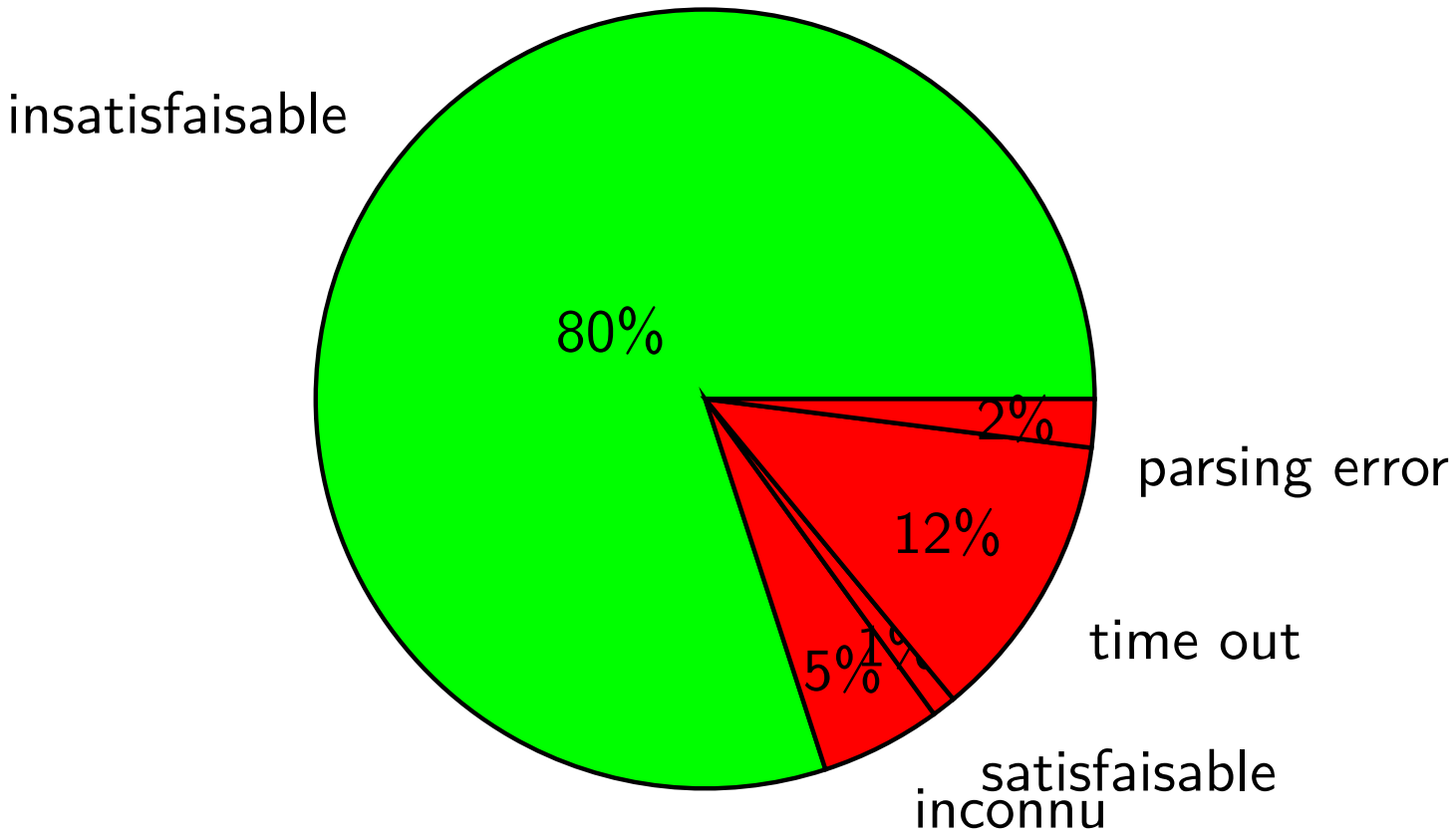
Résultats sans monomorphisation

Nombre total de problèmes: 271

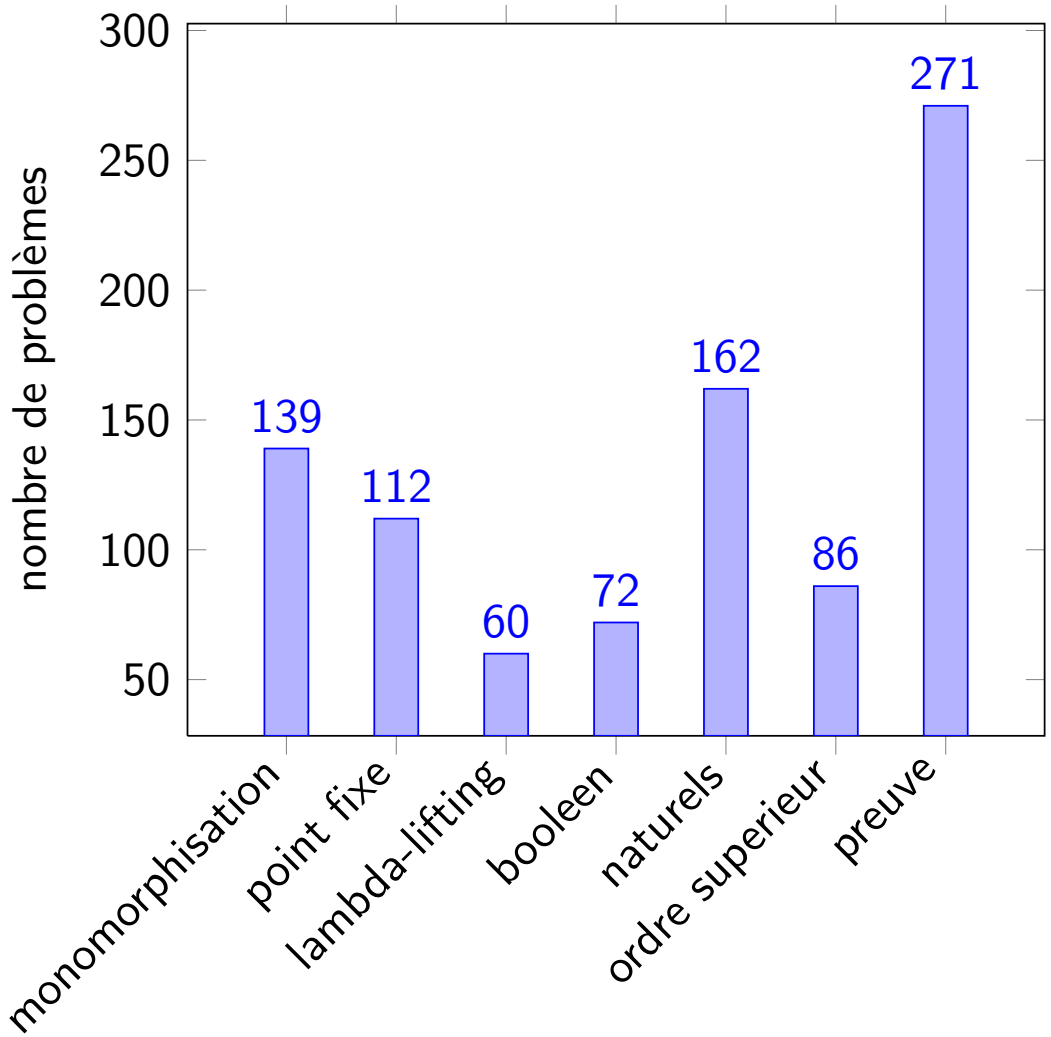


Résultats avec monomorphisation

Nombre total de problèmes: 271

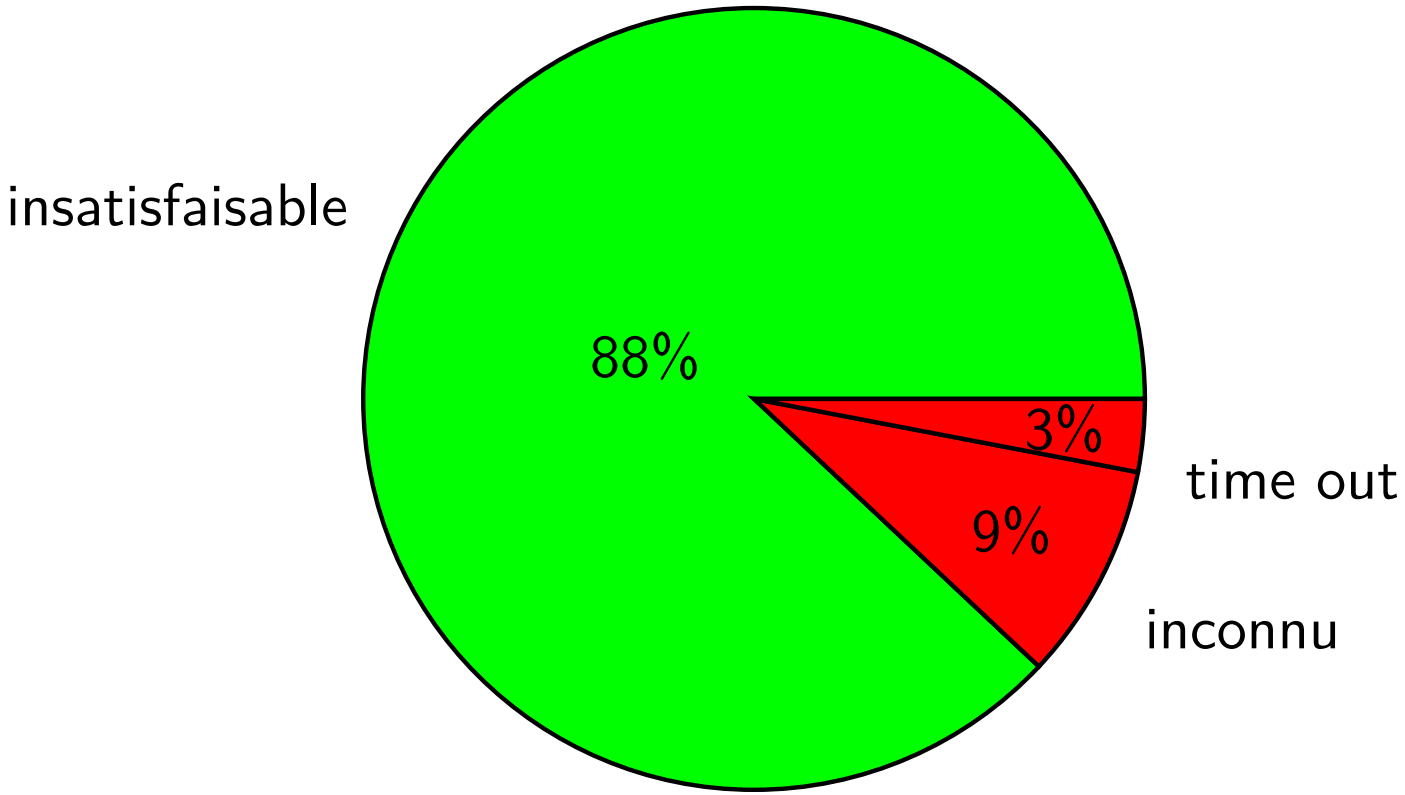


Utilisation des différentes parties de la traduction

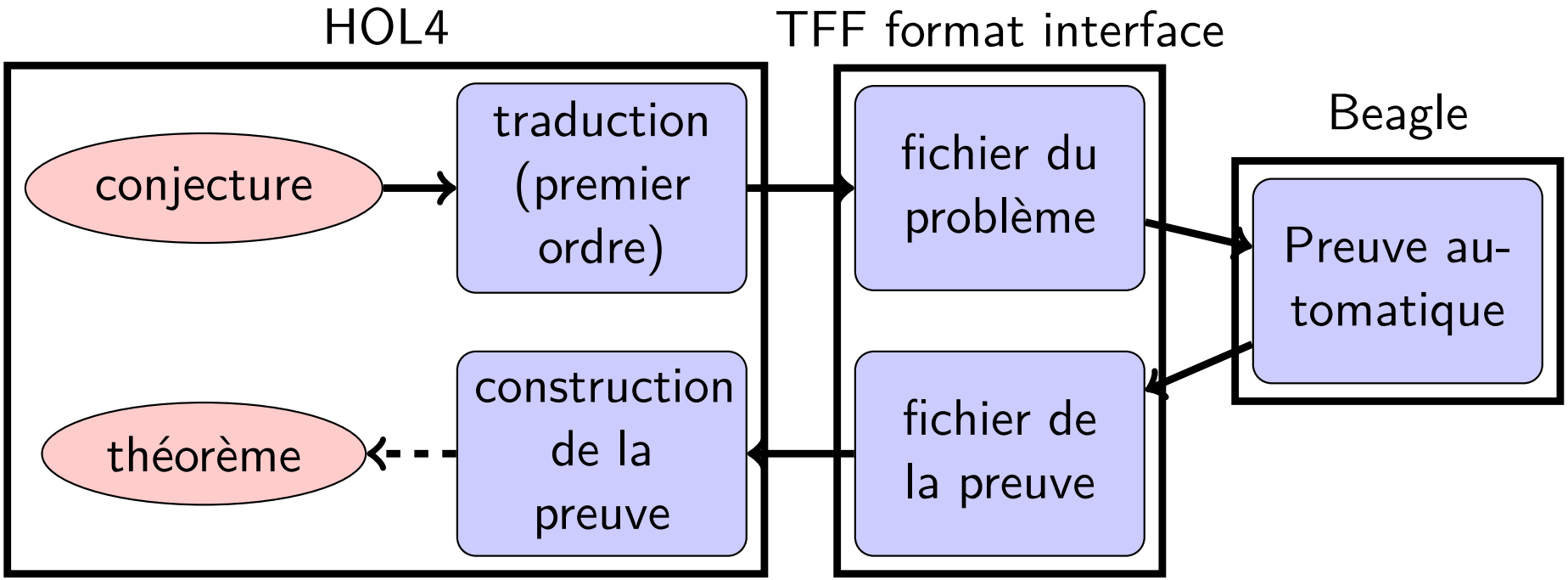


Résultats sur des problèmes arithmétiques

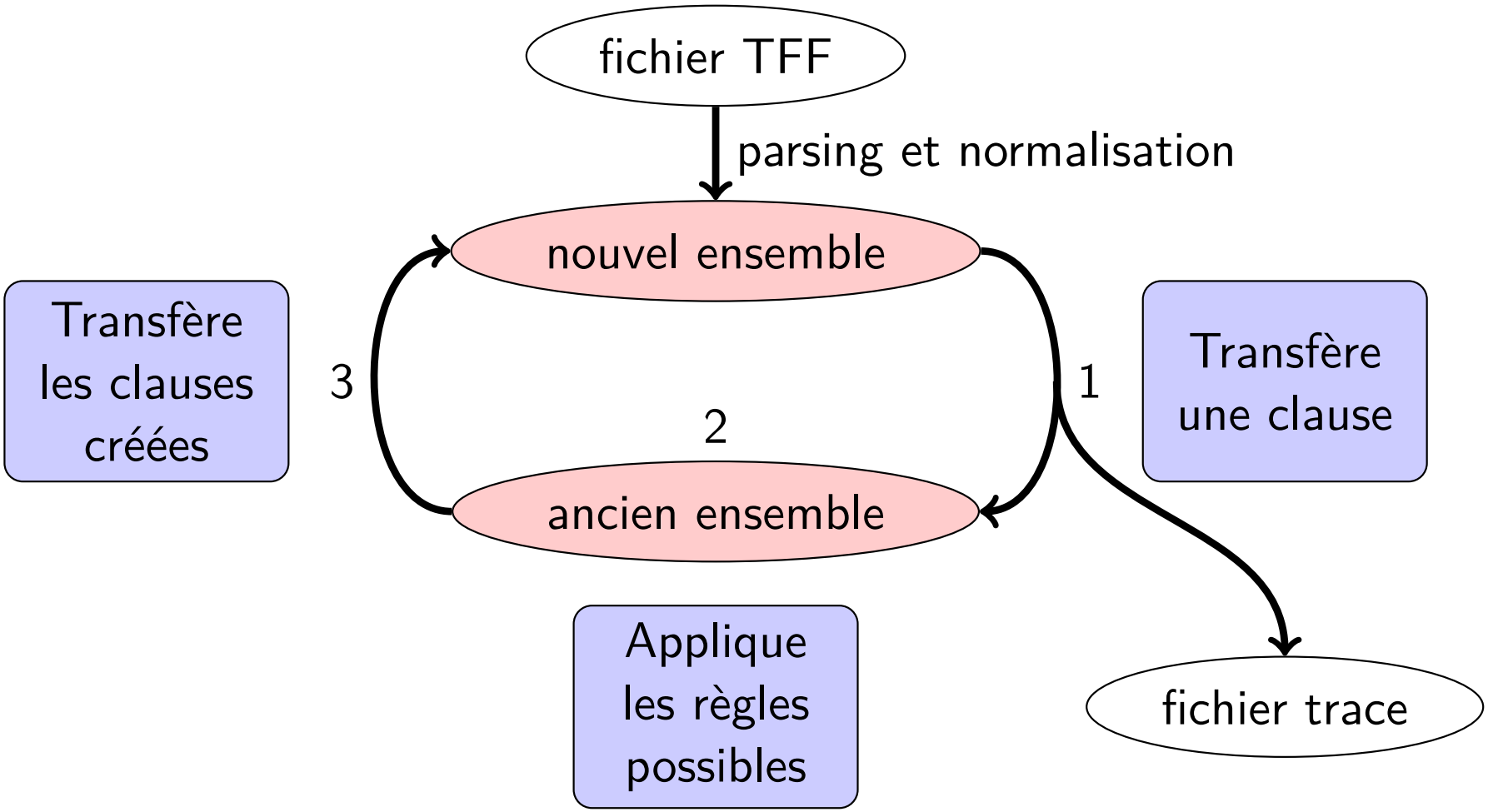
Nombre de problèmes: 65.
Dans ces problèmes, tous les théorèmes (79) ne concernant que l'arithmétique ont été effacés.



Principe général



Extraction d'une preuve



Construction de la preuve

- Lecture de la preuve: utilise des dictionnaires de variables et de types
- Rejouage de la preuve: chaque étape peut être résolue par metis, cooper ou une combinaison des deux. (en théorie!!)

Qualités et limites de l'interaction HOL4-Beagle

Qualités:

- Résout des problèmes arithmétiques sans guidage
- Un format de communication répandu
- Une traduction correcte préservant l'insatisfaisabilité

Limites:

- 20% des conjectures non prouvées par Beagle
- Echec du rejouage de la preuve dans beaucoup de cas
- Une traduction incomplète
- Une traduction ne préservant pas la satisfaisabilité

Améliorations possibles de la traduction

- Traduire les rationnels et les réels
- Générer automatiquement des théorèmes aidant à prouver la conjecture
- Normalisation des formules arithmétiques dans les preuves.

$$x < y + 1 + 2 \longrightarrow x - y < 3$$