Assistants de preuve

TD 8- Sémantique du Calcul des Constructions et du Calcul des Constructions Inductives

1 Expressivité : les entiers de Church

Un entier de Church est un entier représenté par une fonctionnelle de la forme

$$\lambda x.\lambda f.f(\ldots(f(x))\ldots)\,.$$

On se place dans le calcul des constructions (les sortes sont appelées `Prop` et `Type` et le produit est noté avec $\forall$). Si A est un type, on définit l'égalité sur A par

$$(x =_A y) \triangleq \forall P : (A \to Prop)\ P(x) \to P(y)$$

et la négation $\neg A$ de A par

$$\neg A \triangleq A \to \forall C.\, C\,.$$

A- Définition des entiers dans la couche `Prop` du Calcul des Constructions

Dans la suite s représente la sorte `Prop`.

1. Donner une expression close N de type s exprimant le type des entiers de Church. Comment s'expriment alors 0 et l'opération « successeur » (notée S) ?

2. Peut-on définir le prédécesseur sur N ?

3. On pose $IND(n) \triangleq \forall P : N \to \texttt{Prop}.(P(0) \to (\forall m : N.\, P(m) \to P(S(m))) \to P(n)$. Montrer que le principe de récurrence $\forall n : N.\, IND(n)$ (5e axiome de Peano) n'est pas dérivable en l'interprétant dans le modèle booléen dont les propositions habitées sont habitées par tous les lambda-termes.

4. En l'interprétant dans le modèle booléen avec indiscernabilité des preuves, montrer que l'énoncé $\forall n : N, S(n) \neq 0$ (3e axiome de Peano) n'est pas prouvable.

5. Montrer qu'on peut prouver $\forall n, m : N, IND(n) \to IND(m) \to S(n) = S(m) \to n = m$ (4e axiome de Peano) ?

B- On se repose les même questions en prenant pour s la sorte `Type` du Calcul des Constructions (le type N n'est alors pas polymorphe). Notez qu'en ce cas, on n'a pas besoin de toute la richesse du Calcul des Constructions : celle du système F_ω suffit. Le codage diffère-t-il alors vraiment de celui dans le λ-calcul simplement typé ? Observer en particulier que, bien que l'addition et la multiplication soient définissables, le prédécesseur, lui, ne l'est pas.

C- Mêmes questions en se plaçant dans $F_{\omega.2}$, c'est-à-dire dans le système avec $\texttt{Type}_2$ et quantification de $\texttt{Type}_2$ sur `Type` donnant in produit dans $\texttt{Type}_2$. Montrer alors que tous les axiomes de Peano sont prouvables.

2 Restrictions d'élimination liées aux sortes

A- On pose

```
Inductive True : Prop := I : True.
```

Exhiber une fonction de `unit` vers `True` dont on peut montrer qu'elle est une bijection.

B- On pose

```
Inductive BOOL : Prop := TRUE : BOOL | FALSE : BOOL.
```

Peut-on montrer l'équivalence de `bool` et `BOOL` ? Montrer que si c'était le cas, on pourrait nier le principe d'indiscernabilité des preuves (proof-irrelevance) dans `Prop`.

3 Type coinductifs

On considère le langage de processus concurrent définit par

$$p \quad ::= \quad (p||p) \mid ?\overline{\alpha}(x).p \mid (!\alpha(t).p + ?\overline{\beta}(x).p)$$

où α et β sont des canaux (en réception si surlignés, en émission sinon). Représenter ce langage (appelé CBS) par un type coinductif (on supposera avoir un type des canaux `channel` et des valeurs `dom`).

4 Paradoxes liés à la positivité non stricte

On considère le type non strictement positif suivant :

```
Inductive T : Type := I : ((T->Prop)->Prop)->T.
```

Donner une injection de $T \to$ `Prop` dans $(T \to$ `Prop`$) \to$ `Prop`.

En déduire une fonction d'injection ϕ de $T \to$ `Prop` vers T. En montrer l'injectivité.

En déduire la dérivabilité du paradoxe de Russell (l'ensemble de tous les ensembles qui n'appartiennent pas à eux-mêmes est contradictoire).