

Contents

1	Types dépendants	1
2	Le problème de la description des types	2

L'objectif de ce projet de recherche est d'implémenter un système de type qui intègre la descriptions des types au sein de la théorie des types.

1 Types dépendants

Les types dépendants constituent le socle de ce projet, il est donc de bon ton de commencer par un rapide rappel sur la notion.

Le lecteur familier avec un langage de programmation typé moderne aura été exposé à la notion de type générique. Ce sont les types qui sont fonctions d'autre types. Par exemple, les listes en OCaml sont décrites par `'a list` où `'a` représente nécessairement un type (`int`, `char`, ...). Ce mécanisme rend plus ergonomique l'expression des structures de données.

Les types dépendants eux, sont fonction de termes. Un exemple habituel est celui des listes dont la longueur est représentée dans le type. On pourrait imaginer la notation `(int * 'a) list` où `(5, int) list` est le type des listes d'entier de longueur 5.

Ce qui rend cruciale l'introduction de cette expressivité, est qu'elle permet d'encoder, en plus de sa structure, la logique d'un type. Par nature, les types dépendants décrivent donc un comportement dynamique.

Pour illustrer ce propos, prenons en exemple les fonctions de choix de l'axiome éponyme.

- Mathématiquement, soit X un ensemble, alors une fonction f est de choix si elle a pour structure

$$f : \mathcal{P}(X) \rightarrow X$$

et qu'elle vérifie

$$\forall S \in \mathcal{P}(X), f(S) \in S$$

- Dans un langage simplement typé, en supposant que X et $P(X)$ désignent des types, on peut décrire la structure d'une fonction de choix:

```
val f : P(X) -> X
```

- Avec des types dépendants, il est aussi possible d'en décrire la logique:

```
val f : (S : P(X)) -> S
```

Ici, la nature de S n'est pas évidente : il est à la fois terme (à gauche) et type (à droite). Nous passons cette question sous silence, le principal est de constater la dépendance du type flèche sur S en tant que terme.

Le choix de décrire un objet mathématique dans ce précédent exemple n'est pas anodin. Il reflète un aspect important de l'intérêt pratique du domaine de ce projet de recherche.

En introduisant les types dépendants, on constate aussi un phénomène notable. Types et valeurs sont aplatis sur le même plan syntaxique. En effet, on voit

2 Le problème de la description des types