

Recherche d'une fonction par sa signature

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 18 juillet 2006

<https://www.bortzmeyer.org/recherche-par-signature.html>

Un des avantages des langages de programmation typés, c'est qu'ils permettent de rechercher une fonction par son type, donc par l'API souhaitée. C'est ce que permet, par exemple, le moteur de recherche Hoogle <<http://haskell.org/hoogle/>>.

Haskell est un langage de programmation typé. Toute fonction a une **signature**, qui comprend le type des paramètres d'entrée et le type du résultat. Cela permet de détecter beaucoup de bogues dès la compilation. Et cela sert de **contrat** : la signature indique ce que la fonction est capable de faire. À noter qu'il n'est en général pas besoin de déclarer ces types, ils sont trouvés automatiquement par inférence de types.

Un environnement de programmation comme ghc <<http://www.haskell.org/ghc/>> permet d'afficher ladite signature :

```
*Main> :t mkRegex
mkRegex :: String -> Regex
```

Ici, la fonction `mkRegex` prend une chaîne de caractères (`String`) en paramètre et rend une `Regex`

De même, le mode Haskell de l'éditeur Emacs affiche la signature d'une fonction lorsque le curseur passe devant.

On peut pousser cela plus loin et permettre de rechercher une fonction par sa signature. Supposons qu'on aie une liste de chaînes de caractères et qu'on veuille les concaténer en une seule. On cherche donc une fonction capable de faire `[String] -> String` (`[Foo]` signifie une liste de `Foo`). Il suffit de taper cette signature dans Hoogle <<http://haskell.org/hoogle/>> et on obtient :

```
Prelude. unlines :: [String] -> String
Prelude. unwords :: [String] -> String
...
```

(Prélude est la bibliothèque standard de Haskell.) Voici deux fonctions qui conviennent tout à fait et, notamment, `unlines` est exactement ce que l'on cherche :

```
let result = unlines mylist
```

Il existe une fonction plus générale, `fold`, que Hoogle trouve aussi, et qui permettrait d'écrire : `let result = foldl (++) "" mylist.`

De même, si j'ai oublié le nom de la fonction qui permet de produire un tableau à partir d'un élément et d'un nombre de répétitions (une telle fonction est utile par exemple pour du formatage de données), je cherche `a -> Int -> [a]` et je trouve tout de suite `replicate` (qui me permet ensuite de faire des choses comme `replicate 3 '0'` pour obtenir `"000"`).

À noter que le bot IRC qui écoute sur le canal `#haskell` de `irc.freenode.net` a une interface à Hoogle :

```
(16:55:00) bortzmeyer: @hoogle CalendarTime -> String
(16:55:01) lambdabot: Time.calendarTimeToString :: CalendarTime -> String
(16:55:01) lambdabot: System.Time.ctTZName :: CalendarTime -> String
(16:55:01) lambdabot: Time.formatCalendarTime :: TimeLocale -> String -> CalendarTime -> String
```