

TD 12

Exercice 1. Appliquez l'algorithme d'unification "naïf" (exponentiel) vu en cours aux systèmes d'équations suivants. Pouvez vous donner un unificateur autre que le mgu ?

1. $\{y = f(x, z), y = f(3, 5)\}$
2. $\{f(g(x)) = f(z), g(z) = g(g(3))\}$
3. $\{a(x, x) = a(\text{int}, a(\text{int}, \text{int}))\}$
4. $\{f(x) = f(f(f(x)))\}$
5. $\{\alpha = \beta \rightarrow \beta, \beta = \gamma \rightarrow \gamma, \gamma = \delta \rightarrow \delta\}$

Exercice 2. Associez à chaque expression pureML ci-dessous un système d'équations en appliquant l'algorithme de Hindley. Donnez "de tête" son MGU lorsqu'il existe.

1. `x+3`
2. `3 4`
3. `letrec f(x) = x+3 in f(y)`
4. `letrec f(x) = if x=0 then 0 else f(x-1) in f(y)`

Exercice 3. 1. Donnez un exemple de terme clos de pureML qui ne type pas en monomorphie pureML.

2. Donnez un exemple de terme clos qui ne type pas en pureML mais qui ne se réduit pas Wrong.

Exercice 4. En imaginant la généralisation naturelle des règles de typage de pure ML, typez le programme suivant:

```
let r = ref (fun x -> x) in
  r := (fun n -> n+1);
  !r = "abc" ;;
```

Exercice 5. Ecrire en OCaml une fonction `length` pour le type suivant:

```
type 'a mycroft =
  | Nil
  | Cont of 'a * ('a list) mycroft
```

Discutez.

Exercice 6 (Unification en temps polynomial). 1. Montrez que l'algorithme classique vu en cours s'exécute en temps polynomial dans le pire des cas. Montrez qu'en réalité, le problème calculatoire est exponentiel.

2. En vous inspirant de la famille d'instance de la question précédente, proposez une structure de donnée pour les MGU qui contourne le problème évoqué à la question précédente.
3. Proposez une modification des règles de l'algorithme de Robinson adapté à cette nouvelle structure.