



TP 2 : Unification

1 Implantation de l'unification

On s'intéresse à implanter l'algorithme d'unification vu en cours.

Exercice 1 (Termes et opérations sur les termes)

1. Définissez un type **term** qui correspond au type des termes vu en cours, avec des constructeurs **Var** pour des variables et **App1** pour l'application d'une fonction à une liste de termes.
2. Représentez quelques termes comme instances de ce type. Vous pourrez vous inspirer des termes vus en TD. Spécifiez déjà maintenant quelques cas de test pour l'algorithme d'unification.
3. Définissez une fonction **fv** (*free variables*) qui calcule l'ensemble des variables libres d'un terme. (Les ensembles seront implantés comme des listes sans doublons.)

Exercice 2 (Substitutions)

1. Choisissez une représentation appropriée pour des substitutions. Une suggestion : vous pourriez choisir une liste d'association entre variables et termes.
2. Définissez une fonction **subst** qui prend un terme et une substitution et renvoie le terme où la substitution a été effectuée.
3. Il faudra ensuite programmer quelques fonctions auxiliaires, par exemple pour effectuer une substitution dans une substitution. Implantez ces fonctions selon vos besoins dans l'algorithme d'unification.

Exercice 3 (Unification) Définissez maintenant la fonction **unif** qui implante l'algorithme d'unification. Le *fail* de l'algorithme d'unification sera implanté par une exception de Caml (**failwith**). L'implantation est une traduction assez directe de l'algorithme vu en cours, avec un filtrage sur l'ensemble E : vide ou non-vide, et dans ce cas, un filtrage sur la structure du couple de termes à unifier.