

Le 09 avril 2018

Professeure : Mr Christian Codognet

Projet de programmation fonctionnelle

Ismaël Lachheb et Quentin Bresson

Table des matières

[Objectif du projet 2](#_Toc510986678)

[L’algorithme d’unification 3](#_Toc510986679)

[1. Les fonctions annexes 3](#_Toc510986680)

[2. L’algorithme 3](#_Toc510986681)

[3. Difficulté(s) rencontrée(s) 3](#_Toc510986682)

[L’algorithme d’anti-unification 4](#_Toc510986683)

[1. Fonctions annexes 4](#_Toc510986684)

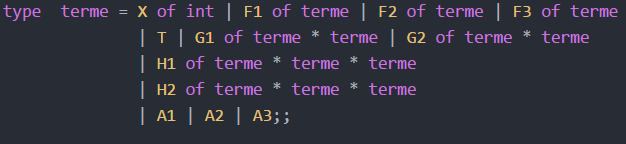
[2. L’algorithme 4](#_Toc510986685)

[3. Difficulté(s) rencontrée(s) 4](#_Toc510986686)

Objectif du projet

Ce projet a pour objectif de mieux comprendre les mécanismes permettant l’unification et l’anti-unification dans le langage OcamL.

Pour cela on utilisera le type suivant :

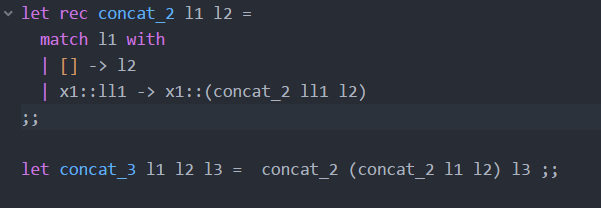


L’utilisateur devra être capable, à partir de la console, de renseigner des termes, de les unifier et de les anti-unifier.

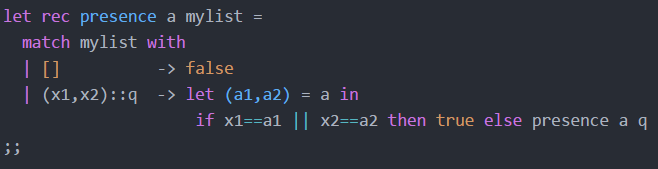
# L’algorithme d’unification

## Les fonctions annexes

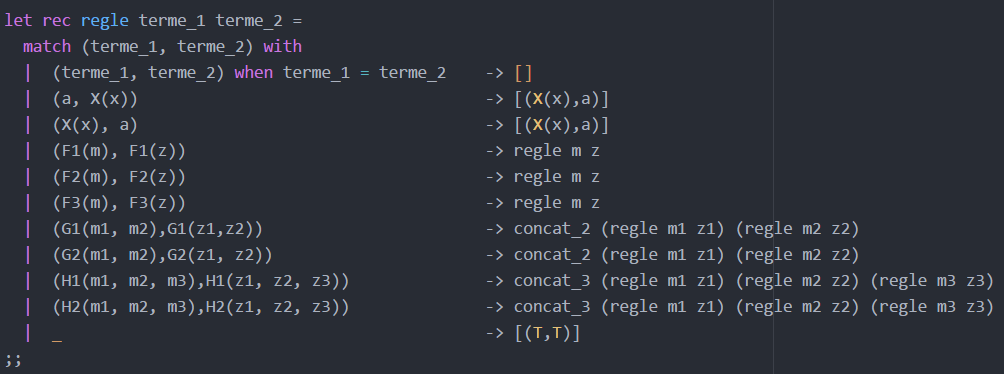
Les fonctions de concaténation (ne pour concaténer deux termes, l’autre pour en concaténer 3) :



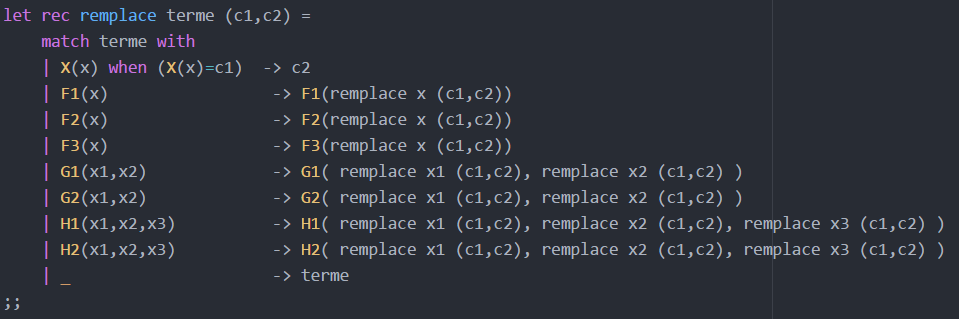
Une fonction permettant de vérifier la présence d’un élément dans une liste :



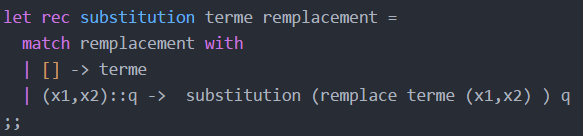
Une fonction règle représentant l’implémentation des 4 règles appliquées par l’algorithme d’unification :



Une fonction permettant de remplacer les occurrences d’un terme contenu dans un terme par un autre terme :

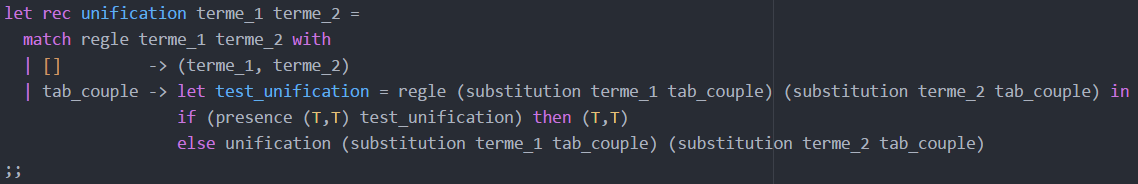


Une fonction représentant l’implémentation de la quatrième règle de l’algorithme d’unification, utilisant la fonction précédente :



## L’algorithme

En utilisant les fonctions précédentes, on arrive à ce code :

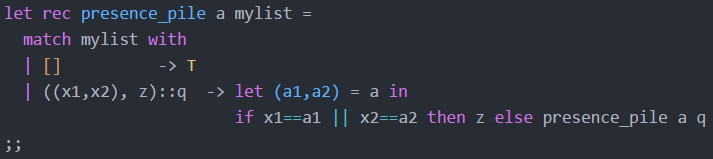


## Difficulté(s) rencontrée(s)

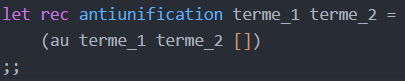
# L’algorithme d’anti-unification

## Fonctions annexes

Une fonction permettant de vérifier la présence dans la pile d’un couple de terme :

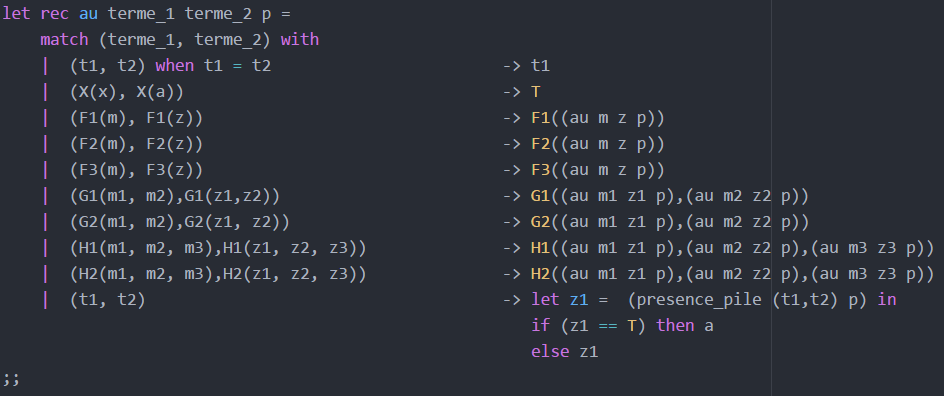


Une fonction permettant d’initialiser la pile avant le lancement de l’algorithme. C’est cette fonction que l’utilisateur appelle pour anti-unifier deux termes :



## L’algorithme

L’algorithme est principalement implémenté dans cette fonction :



## Difficulté(s) rencontrée(s)

L’anti-unification de deux termes consiste à trouver une généralisation de ces 2 termes. L’algorithme demande de gérer une pile au travers de son exécution et c’est un problème que je ne suis pas parvenu à surmonter. En effet comment prendre en compte la pile ? Comment y ajouter **dynamiquement** de nouvelles variables ?