TP 2 - Unification
Préparation et implémentation de l'algorithme d'unification de Robinson

Binôme: Binous Mohammed Chedly & Bouker Rania
RT4-Groupe1

Objectifs :

- Implémenter et tester l'algorithme d'unification de robinson vue en cours en python

Task1 : Création des classes qui représente le type des éléments à unifier

- ✓ Déclarer les classes adéquates pour représenter les informations en entrées.
- ✓ Les expressions à unifier peuvent être des atomes (Variable, Constante, Liste Vide) ou bien des fonctions. Mais la fonction peut être représenter de la manière suivante :
- ✓ F (a, b, c) = [F, a, b, c] : F est le premier élément de la liste et représente le nom de la fonction qui est constant (on ne peut pas modifier le nom de la fonction !) et les paramètres de la fonction (a, b,c) sont les autres éléments de la liste .
- ✓ Cette représentation nous permet de simplifier le travail à effectuer et n'a aucun impact sur le résultat de l'unification.
- ✓ Et donc il suffit de créer deux classes : « Constante » pour les constantes et « variable » pour les variables ayant chacune l'attribut nom.

Task2 : Implémentation de l'algorithme d'unification de robinson :

- ✓ Dans cette partie, nous allons procéder à l'implémentation de l'algorithme ayant en entrées deux expressions E1 et E2 avec des variables et donc on aura besoin d'implémenter trois fonctions essentielles :
  - La fonction **UnifierAtomes :** Prend en entrée un atome et un élément (atome ou liste) effectuer les tests nécessaires :
    - Interchanger E1 et E2 de sorte que E1 soit l'atome.
    - Si E1 et E2 identique => retourne null (pas de changement)
    - Si E1 est une variable et E2 est une liste deux cas se présente : E2 contient E1 et donc la fonction retourne Echec ou bien E2 ne contient pas E1 et donc on fait appel à la méthode Unifier.
    - Si E1 (variable ou constante) et E2 variable alors on remplace E2 par E1
    - Si E1 et E2 sont deux constantes la fonction retourne : ECHEC
  - La fonction **Substituer** : Permet de mettre à jour les expressions E1 et E2 lorsqu'il y a du changement.
  - La fonction **Unifier** : Prend en entrée les deux expressions E1 et E2 : prélève les premiers éléments de chacune des expressions. Teste s'il s'agit d'atomes ou de listes, si l'un des

éléments est un atome alors elle fait appel à la fonction **UnifierAtomes** qui lui retourne le résultat de l'unification (les modification). Ces modifications sont utilisées par la méthode Substituer pour mettre à jour les expressions.

La fonction unifier retourne le résultat de l'unification !

Task3 : Fichier de sauvegarde :

✓ Dans cette dernière étape on va ajouter un fichier de trace dans lequel on enregistrera tous les résultats du jeu de tests de cet algorithme.

- Pour ouvrir le fichier de trace en mode écriture on utilise la syntaxe suivante :

```
fichier = open("Path de fichier ","w")
```

- A chaque fois on écrit le résultat dans le fichier avec la commande suivante :

```
fichier.write("Resultat")
```

- Il ne faut pas oublier de fermer le fichier de trace a la fin pour sauvegarder les modifications !

```
fichier.close()
```

✓ Résultat :