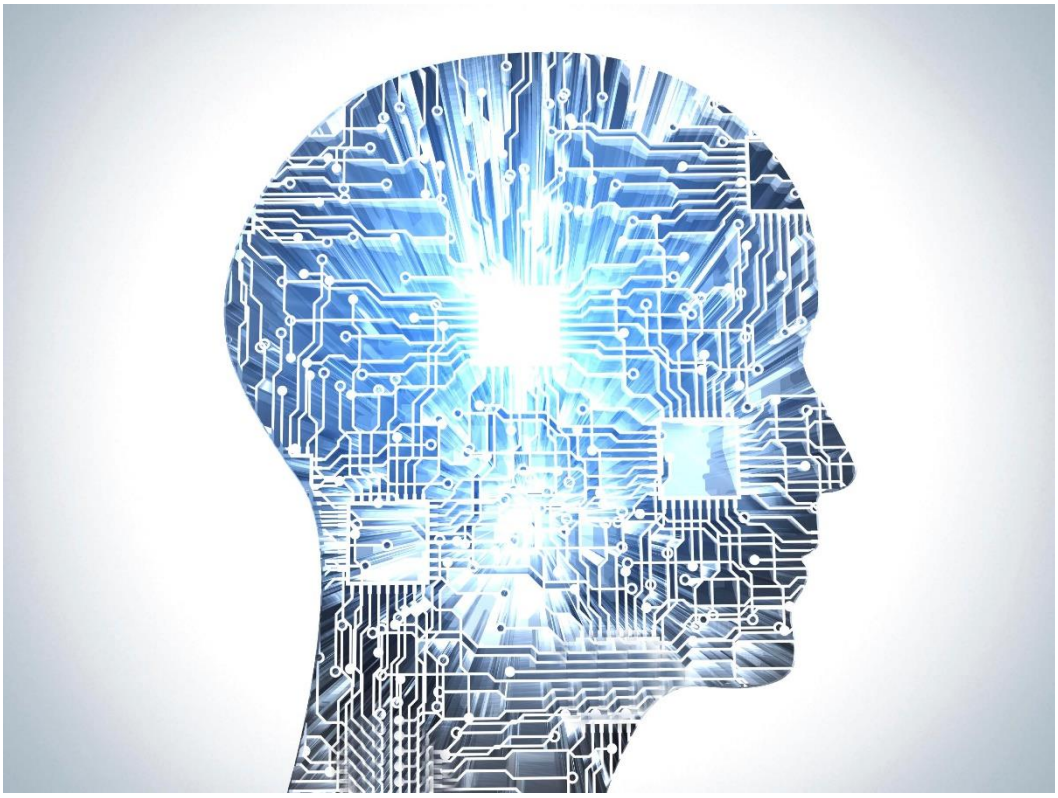
COMPTE RENDU :

TP1 : CHAINAGE AVANT



Hammouda Hajer & Chaabene Amal

RT4 / Groupe 2

Introduction :

Le but de ce TP est de préparer les fonctions Python permettant d'importer une base de connaissances , d'écrire l'algorithme du moteur d'inférence d'ordre 0 en chainage AVANT et de tester notre système expert sur différentes bases de connaissances.

Explication du travail réalisé :

Nous avons préparé des fichiers qui contiennent les bases des règles et les bases des faits. Chaque ligne du fichier contient une règle et le fichier des faits contient une ligne où les faits sont séparés par des virgules.

Dans le code, nous avons défini trois classes qui sont :

- Classe « Regle » : qui définit une règle sous forme de tableau dont les colonnes sont [premisse,conclusion,etat,idn].

- Classe « Fait » : qui définit un fait sous forme de tableau [fait,numRegle].

- Classe « definition » : c'est dans cette classe où les traitements sont divisés sur plusieurs fonctions.

  - Fonction « listeRegles » : dans cette fonction nous avons défini une liste vide qui va contenir toutes les règles que nous allons extraire du fichier. Chaque colonne de cette liste est un objet de type « Regle ». L'utilisateur va entrer le nom du fichier désiré et le traitement va être effectué sur les lignes de ce fichier pour extraire les donnée de chaque règle et les mettre dans la liste des règles.
    Cette fonction va retourner la liste des règles.

  - Fonction « listeFaits » : cette fonction va nous retourner les faits sous forme d'une liste. Chaque colonne contient un objet de type « Fait ». L'utilisateur va choisir le fichier de base des faits après le traitement cette fonction va retourner la liste des faits.

  - Fonction « regleDecl » : c'est une fonction qui va retourner la liste des règles déclenchables. Elle prend en argument la liste des règles, la liste des faits, le chemin et un entier x qui sert à décider sur le but s'il est atteint ou pas.
    On déclare une liste vide des faits « faits » et une autre des règles déclenchagles « decl » . Pour chaque élément de la liste des faits on extrait le nom du fait et on l'ajoute à la liste déclarée précédemment « faits » et pour chaque règle de la liste passée en argument on teste : si les prémisses de la règle existent tous dans la liste des faits « faits » on l'ajoute à la liste « decl ». Après, si la liste des règles

declenchable n'est pas vide la fonction retourne cette liste sinon on affiche « but non atteint » et on demande à l'utilisateur s'il veut enregistrer le chemin dans un fichier trace.

- Fonction « choixRegle » : cette classe prend en paramètre la liste des règle declenchables et le choix de l'utilisateur s'il souhaite un traitement en choisissant la 1ère règle ou bien la règle ayant le plus des prémisses.

- Fonction « change » : cette fonction prend en paramètre une règle et un indice et qui sert à changer l'état d'une règle en « false » après qu'elle soit exécutée. Et elle retourne cette règle.

- Fonction « saturation » : cette fonction prend en paramètre la liste des règle, la liste des faits et le choix de l'utilisateur concernant le type de traitement (1ere règle/plus de prémisses). Elle va extraire la liste des règles déclenchables et la 1ère règle. Elle extrait la liste des faits et elle va commencer le traitement tant qu'il existe une règle déclenchable, si la conclusion n'existe pas dans cette liste elle sera ajoutée et sera supprimée de la liste des règles et ainsi de suite.
  Si la conclusion de la règle existe dans la liste des faits ou le non du fait existe aussi dans la liste des faits elle affiche une contradiction.
  → Cette fonction n'est déclenchée que si l'utilisateur choisit un traitement avec saturation de la base des faits.

- Fonction « butAtteint » : cette fonction prend en paramètre la liste des règles, la liste des faits, le but à chercher, et le choix de l'utilisateur concernant le choix des règles. Elle procède par les mêmes étapes que la fonction « saturation » mais si elle trouve le but elle arrête le traitement.

- Fonction « choixSortie » : cette fonction interagit avec l'utilisateur en lui demandant le type de traitement qu'il souhaite appliquer.

- Maintenant nous avons l'appel du constructeur de la classe « defnition » et après la fonction « choixSortie() » de cette classe.

Ce document est accompagné par une Démo qui montre le fonctionnement de l'algorithme réalisé.