



UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE
UV IA01

Compte-rendu de TP n°3

Noé CONNAN
Julien PONTOIRE

Semestre : A23

Table des matières

1	Introduction	2
2	Explication de la base de règles	3
2.1	Base de règle	3
2.2	Arbre de déduction	6
2.3	Sources	7
3	Implémentation en LISP	8
3.1	Explication du moteur d'inférence choisi	8
3.2	Chaînage avant	8
3.3	Chaînage arrière	9
3.4	Main	9
4	Test des moteurs	11
4.1	Chaînage avant	11
4.2	Chaînage arrière	11
5	Problèmes et Améliorations	13
6	Conclusion	15

1 Introduction

L'UTC regorge d'associations variées qui peinent à toucher leur public cible - parfois par leur manque de communication, parfois à cause de leur petite taille. Par conséquent, les UTCéen.nes se désengagent de plus en plus du monde des assos malgré leur intérêt certain, à la fois au sein de la formation et pour la création de lien social sur le campus. La crise au sein du monde associatif nous a particulièrement marquée lorsque nous avons constaté la quantité d'assos mortes faute de gens pour reprendre le bureau - Japon UTC et Work 'n chill, pour ne citer qu'elles - et de la disparition de services et d'événements faute de trésorerie - le regretté Polar, les événements d'Etuville, l'Imaginarium Festival.

Nous pensons que cette crise du monde associatif, qui n'est pas exclusive à l'UTC mais qui se ressent aussi sur le campus, est en partie imputable à la difficulté de choisir une ou plusieurs associations qui correspondent à nos caractéristiques personnelles (centres d'intérêt, valeurs et le niveau d'investissement souhaité par exemple). En réalisant un système expert d'aide au choix d'une association pour les étudiants, nous espérons à la fois permettre aux utilisateurs de découvrir les associations méconnues de leur campus et apporter de nouvelles recrues potentielles dans le monde associatif UTCéen.

2 Explication de la base de règles

Pour construire notre base de règles, nous nous sommes tout d'abord aidés du site des associations de l'UTC du BDE. Actuellement, les associations sont divisées en 4 pôles : le Pôle Vie Du Campus (PVDC), le Pôle Artistique et Événementiel (PAE), le Pôle Solidarité Et Citoyenneté (PSEC) et enfin le Pôle Technologie et Entrepreneuriat (PTE). Chacun de ces pôles est géré par une association du même nom. Nous avons également le Bureau Des Étudiants (BDE), le PIC et le Service Informatique de la Maison Des Étudiants (SIMDE). Si on ne compte pas les assos dans le "cimetière des assos", le BDE en dénombre actuellement 147 actives, et dans les faits environ 135 sont réellement peuplées ce semestre.

Le site du BDE comporte des informations très lacunaires concernant la description des assos, dont le nom n'indique pas toujours la fonction - si pour Japon UTC c'est plutôt clair, pour PTP, BDBDEC ou Tactose ça l'est moins; et compte tenu de la quantité d'informations à intégrer pour réaliser correctement notre système expert, nous avons tout d'abord décidé de réunir l'ensemble des connaissances que nous avons concernant ces associations, puis demandé des informations supplémentaires directement aux personnes concernées lorsque cela était nécessaire.

Suite à ce gros recensement, nous avons établi une première arborescence avec l'intégralité des assos (elle fait 5 pages et c'est très moche, donc on a préféré épargner votre rétine et ne pas l'intégrer au document). Nous avions à l'origine pensé à établir un système de "score" avec une modélisation différente qui s'est finalement avéré trop complexe à implémenter. Nous sommes donc partis sur un arbre dont les associations sont les feuilles en nous inspirant d'un système expert répertoriant des sports trouvé en ligne.

Cependant, l'implémentation des 147 assos aurait nécessité un nombre de règles considérable, nous avons donc choisi de nous limiter aux branches "artistique", "culturel" et "gestion" (avec les pôles, le BDE et le SIMDE) des associations.

2.1 Base de règle

Voici la Base de Règles que nous avons implémentée en LISP.

```
(setq *BDR*  
  '(  
    (R1 ((type0 eq ART)) (type1 eq ASSO_ART))  
    (R2 ((type0 eq CULTURE)) (type1 eq ASSO_CULTURE))  
    (R3 ((type0 eq GESTION)) (type1 eq ASSO_GESTION))  
    (R4 ((type1 eq ASSO_ART) (activite eq THEATRE))  
      (type2 eq ASSO_THEATRE))  
    (R5 ((type1 eq ASSO_ART) (activite eq MUSIQUE))
```

```

        (type2 eq ASSO_MUSIC))
(R6 ((type1 eq ASSO_ART) (activite eq MEDIA))
    (type2 eq ASSO_MEDIA))
(R7 ((type2 eq ASSO_THEATRE)) (asso eq PROFITROLES))
(R8 ((type2 eq ASSO_MUSIC) (nb <= 2)) (asso eq FSC))
(R9 ((type2 eq ASSO_MUSIC) (nb >= 3) (nb <= 6))
    (asso eq LARSEN))
(R10 ((type2 eq ASSO_MUSIC) (nb >= 7))
    (type3 eq ASSO_MUSIC_G))
(R11 ((type3 eq ASSO_MUSIC_G) (mode eq ORCHESTRE))
    (type4 eq ASSO_ORCH))
(R12 ((type3 eq ASSO_MUSIC_G) (mode eq FANFARE))
    (type4 eq ASSO_FANF))
(R13 ((type4 eq ASSO_ORCH) (genre eq JAZZ))
    (asso eq OCATA))
(R14 ((type4 eq ASSO_ORCH) (genre eq AUTRE_GENRE))
    (asso eq STRAVAGANZA))
(R15 ((type4 eq ASSO_FANF)) (asso eq CAPHARNAUM))
(R16 ((type2 eq ASSO_MEDIA) (activite eq ECRITURE))
    (type3 eq ASSO_ECRITURE))
(R17 ((type2 eq ASSO_MEDIA) (activite eq DESSIN))
    (type3 eq ASSO_DESSIN))
(R18 ((type2 eq ASSO_MEDIA) (activite eq RADIO))
    (ASSO eq GRAPHIT))
(R19 ((type2 eq ASSO_MEDIA) (activite eq PHOTO-VIDEO))
    (type3 eq ASSO_PHOTO))
(R20 ((type2 eq ASSO_MEDIA) (activite eq VIDEOLUDIQUE))
    (ASSO eq ARCADIA))
(R21 ((type2 eq ASSO_MEDIA) (activite eq AUTRE))
    (ASSO eq PIKECOUD))
(R22 ((type3 eq ASSO_ECRITURE) (genre eq INFORMATIF))
    (ASSO eq LE_FIL))
(R23 ((type3 eq ASSO_ECRITURE) (genre eq ARTISTIQUE))
    (ASSO eq LE_CID))
(R24 ((type3 eq ASSO_DESSIN) (genre eq DESIGN))
    (ASSO eq AZERO))
(R25 ((type3 eq ASSO_DESSIN) (genre eq TOUT))
    (ASSO eq DADA))
(R26 ((type3 eq ASSO_PHOTO) (genre eq OUI))

```

```

        (ASSO eq ETUSEXY))
(R27 ((type3 eq ASSO_PHOTO) (genre eq NON))
      (ASSO eq CINEMUT))
(R28 ((type1 eq ASSO_CULTURE) (culture eq ETRANGERE))
      (type2 eq ASSO_CULTURE_ETRANGERE))
(R29 ((type1 eq ASSO_CULTURE) (culture eq FRANCAISE))
      (type2 eq ASSO_CULTURE_FRANCAISE))
(R30 ((type1 eq ASSO_CULTURE) (culture eq POP))
      (type2 eq ASSO_CULTURE_POP))
(R31 ((type2 eq ASSO_CULTURE_ETRANGERE))
      (asso eq JAPONUTC))
(R32 ((type2 eq ASSO_CULTURE_FRANCAISE))
      (asso eq DEVOIR_DE_MEMOIRE))
(R33 ((type2 eq ASSO_CULTURE_POP) (activite eq PHYSIQUE))
      (type3 eq ASSO_POP_PHY))
(R34 ((type2 eq ASSO_CULTURE_POP)
      (activite eq VIDEOLUDIQUE)) (asso eq LANUTC))
(R35 ((type2 eq ASSO_CULTURE_POP) (activite eq AUTRE))
      (asso eq UTESCAPE))
(R36 ((type3 eq ASSO_POP_PHY) (materiel eq JEUX))
      (asso eq LE_COIN_DU_JOUEUR))
(R38 ((type3 eq ASSO_POP_PHY) (materiel eq LIVRES))
      (asso eq BDBDEC))
(R39 ((type1 eq ASSO_GESTION) (pole eq VIE_CAMPUS))
      (asso eq PVDC))
(R40 ((type1 eq ASSO_GESTION) (pole eq ART_EVENT))
      (asso eq PAE))
(R41 ((type1 eq ASSO_GESTION) (pole eq SOLIDARITE))
      (asso eq PSEC))
(R42 ((type1 eq ASSO_GESTION)
      (pole eq TECH_ENTREPRENEURIAT)) (asso eq PTE))
(R43 ((type1 eq ASSO_GESTION) (pole eq INFORMATIQUE))
      (asso eq SIMDE))
(R44 ((type1 eq ASSO_GESTION) (pole eq GENERAL))
      (asso eq BDE))
    )
  )

```

Ces règles sont définies sous forme de listes. Pour chacune d'entre elles, nous retrouvons en premier élément le numéro de la liste, suivi d'une liste contenant toutes les prémisses

2.2 Arbre de déduction

```

graph TD
    10["PROJECT 10: COUNTRY BRIEFING  
(ID 10)"] --> 10.1["10.1: INTRODUCTION  
(ID 10.1)"]
    10 --> 10.2["10.2: ANALYSIS  
(ID 10.2)"]
    10 --> 10.3["10.3: CONCLUSIONS  
(ID 10.3)"]

    10.1 --> 10.1.1["10.1.1: Overview  
(ID 10.1.1)"]
    10.1 --> 10.1.2["10.1.2: Objectives  
(ID 10.1.2)"]
    10.1 --> 10.1.3["10.1.3: Scope  
(ID 10.1.3)"]

    10.1.1 --> 10.1.1.1["10.1.1.1: Introduction  
(ID 10.1.1.1)"]
    10.1.1 --> 10.1.1.2["10.1.1.2: Overview  
(ID 10.1.1.2)"]
    10.1.1 --> 10.1.1.3["10.1.1.3: Objectives  
(ID 10.1.1.3)"]
    10.1.1 --> 10.1.1.4["10.1.1.4: Scope  
(ID 10.1.1.4)"]
    10.1.1 --> 10.1.1.5["10.1.1.5: Summary  
(ID 10.1.1.5)"]

    10.1.2 --> 10.1.2.1["10.1.2.1: Introduction  
(ID 10.1.2.1)"]
    10.1.2 --> 10.1.2.2["10.1.2.2: Overview  
(ID 10.1.2.2)"]
    10.1.2 --> 10.1.2.3["10.1.2.3: Objectives  
(ID 10.1.2.3)"]
    10.1.2 --> 10.1.2.4["10.1.2.4: Scope  
(ID 10.1.2.4)"]
    10.1.2 --> 10.1.2.5["10.1.2.5: Summary  
(ID 10.1.2.5)"]

    10.1.3 --> 10.1.3.1["10.1.3.1: Introduction  
(ID 10.1.3.1)"]
    10.1.3 --> 10.1.3.2["10.1.3.2: Overview  
(ID 10.1.3.2)"]
    10.1.3 --> 10.1.3.3["10.1.3.3: Objectives  
(ID 10.1.3.3)"]
    10.1.3 --> 10.1.3.4["10.1.3.4: Scope  
(ID 10.1.3.4)"]
    10.1.3 --> 10.1.3.5["10.1.3.5: Summary  
(ID 10.1.3.5)"]

    10.2 --> 10.2.1["10.2.1: Introduction  
(ID 10.2.1)"]
    10.2 --> 10.2.2["10.2.2: Overview  
(ID 10.2.2)"]
    10.2 --> 10.2.3["10.2.3: Objectives  
(ID 10.2.3)"]
    10.2 --> 10.2.4["10.2.4: Scope  
(ID 10.2.4)"]
    10.2 --> 10.2.5["10.2.5: Summary  
(ID 10.2.5)"]

    10.2.1 --> 10.2.1.1["10.2.1.1: Introduction  
(ID 10.2.1.1)"]
    10.2.1 --> 10.2.1.2["10.2.1.2: Overview  
(ID 10.2.1.2)"]
    10.2.1 --> 10.2.1.3["10.2.1.3: Objectives  
(ID 10.2.1.3)"]
    10.2.1 --> 10.2.1.4["10.2.1.4: Scope  
(ID 10.2.1.4)"]
    10.2.1 --> 10.2.1.5["10.2.1.5: Summary  
(ID 10.2.1.5)"]

    10.2.2 --> 10.2.2.1["10.2.2.1: Introduction  
(ID 10.2.2.1)"]
    10.2.2 --> 10.2.2.2["10.2.2.2: Overview  
(ID 10.2.2.2)"]
    10.2.2 --> 10.2.2.3["10.2.2.3: Objectives  
(ID 10.2.2.3)"]
    10.2.2 --> 10.2.2.4["10.2.2.4: Scope  
(ID 10.2.2.4)"]
    10.2.2 --> 10.2.2.5["10.2.2.5: Summary  
(ID 10.2.2.5)"]

    10.2.3 --> 10.2.3.1["10.2.3.1: Introduction  
(ID 10.2.3.1)"]
    10.2.3 --> 10.2.3.2["10.2.3.2: Overview  
(ID 10.2.3.2)"]
    10.2.3 --> 10.2.3.3["10.2.3.3: Objectives  
(ID 10.2.3.3)"]
    10.2.3 --> 10.2.3.4["10.2.3.4: Scope  
(ID 10.2.3.4)"]
    10.2.3 --> 10.2.3.5["10.2.3.5: Summary  
(ID 10.2.3.5)"]

    10.2.4 --> 10.2.4.1["10.2.4.1: Introduction  
(ID 10.2.4.1)"]
    10.2.4 --> 10.2.4.2["10.2.4.2: Overview  
(ID 10.2.4.2)"]
    10.2.4 --> 10.2.4.3["10.2.4.3: Objectives  
(ID 10.2.4.3)"]
    10.2.4 --> 10.2.4.4["10.2.4.4: Scope  
(ID 10.2.4.4)"]
    10.2.4 --> 10.2.4.5["10.2.4.5: Summary  
(ID 10.2.4.5)"]

    10.2.5 --> 10.2.5.1["10.2.5.1: Introduction  
(ID 10.2.5.1)"]
    10.2.5 --> 10.2.5.2["10.2.5.2: Overview  
(ID 10.2.5.2)"]
    10.2.5 --> 10.2.5.3["10.2.5.3: Objectives  
(ID 10.2.5.3)"]
    10.2.5 --> 10.2.5.4["10.2.5.4: Scope  
(ID 10.2.5.4)"]
    10.2.5 --> 10.2.5.5["10.2.5.5: Summary  
(ID 10.2.5.5)"]

    10.3 --> 10.3.1["10.3.1: Introduction  
(ID 10.3.1)"]
    10.3 --> 10.3.2["10.3.2: Overview  
(ID 10.3.2)"]
    10.3 --> 10.3.3["10.3.3: Objectives  
(ID 10.3.3)"]
    10.3 --> 10.3.4["10.3.4: Scope  
(ID 10.3.4)"]
    10.3 --> 10.3.5["10.3.5: Summary  
(ID 10.3.5)"]

    10.3.1 --> 10.3.1.1["10.3.1.1: Introduction  
(ID 10.3.1.1)"]
    10.3.1 --> 10.3.1.2["10.3.1.2: Overview  
(ID 10.3.1.2)"]
    10.3.1 --> 10.3.1.3["10.3.1.3: Objectives  
(ID 10.3.1.3)"]
    10.3.1 --> 10.3.1.4["10.3.1.4: Scope  
(ID 10.3.1.4)"]
    10.3.1 --> 10.3.1.5["10.3.1.5: Summary  
(ID 10.3.1.5)"]

    10.3.2 --> 10.3.2.1["10.3.2.1: Introduction  
(ID 10.3.2.1)"]
    10.3.2 --> 10.3.2.2["10.3.2.2: Overview  
(ID 10.3.2.2)"]
    10.3.2 --> 10.3.2.3["10.3.2.3: Objectives  
(ID 10.3.2.3)"]
    10.3.2 --> 10.3.2.4["10.3.2.4: Scope  
(ID 10.3.2.4)"]
    10.3.2 --> 10.3.2.5["10.3.2.5: Summary  
(ID 10.3.2.5)"]

    10.3.3 --> 10.3.3.1["10.3.3.1: Introduction  
(ID 10.3.3.1)"]
    10.3.3 --> 10.3.3.2["10.3.3.2: Overview  
(ID 10.3.3.2)"]
    10.3.3 --> 10.3.3.3["10.3.3.3: Objectives  
(ID 10.3.3.3)"]
    10.3.3 --> 10.3.3.4["10.3.3.4: Scope  
(ID 10.3.3.4)"]
    10.3.3 --> 10.3.3.5["10.3.3.5: Summary  
(ID 10.3.3.5)"]

    10.3.4 --> 10.3.4.1["10.3.4.1: Introduction  
(ID 10.3.4.1)"]
    10.3.4 --> 10.3.4.2["10.3.4.2: Overview  
(ID 10.3.4.2)"]
    10.3.4 --> 10.3.4.3["10.3.4.3: Objectives  
(ID 10.3.4.3)"]
    10.3.4 --> 10.3.4.4["10.3.4.4: Scope  
(ID 10.3.4.4)"]
    10.3.4 --> 10.3.4.5["10.3.4.5: Summary  
(ID 10.3.4.5)"]

    10.3.5 --> 10.3.5.1["10.3.5.1: Introduction  
(ID 10.3.5.1)"]
    10.3.5 --> 10.3.5.2["10.3.5.2: Overview  
(ID 10.3.5.2)"]
    10.3.5 --> 10.3.5.3["10.3.5.3: Objectives  
(ID 10.3.5.3)"]
    10.3.5 --> 10.3.5.4["10.3.5.4: Scope  
(ID 10.3.5.4)"]
    10.3.5 --> 10.3.5.5["10.3.5.5: Summary  
(ID 10.3.5.5)"]
  
```

Voici des exemples de jeux de données :

- (ART | THEATRE) donne pour résultat l'association PROFIT'RÔLES
- (ART | MEDIA | RADIO) donne pour résultat GRAPH'IT
- (CULTURE | POP | MATÉRIEL PHYSIQUE | JEUX) donne LE COIN DU JOUEUR

2.3 Sources

Lien du site des associations de l'UTC : <https://assos.utc.fr/assos>

3 Implémentation en LISP

3.1 Explication du moteur d'inférence choisi

Pour notre sujet, il pouvait être intéressant de réaliser un moteur d'inférence par chaînage avant mais également par chaînage arrière. En effet le chaînage avant permettrait de trouver à partir des caractéristiques données par l'utilisateur l'association qui lui correspond le mieux. Le chaînage arrière quant à lui permettrait de partir d'une association donnée par l'utilisateur et de vérifier si les informations qu'il a donné correspondent à celles que nécessite cette association. Nous avons donc décidé d'implémenter les deux types de chaînage dans notre code LISP.

Pour le chaînage avant, nous voulions originellement réaliser un algorithme de recherche en profondeur mais nous avons décidé de finalement faire une recherche en largeur qui nous paraissait plus adéquate compte tenu du fait que notre arbre n'a pas de "branche morte", chaque branche comportera toujours une solution.

Cependant pour le chaînage arrière, notre code se rapproche plus d'une recherche en profondeur mais n'en est pas vraiment une. Étant donné que pour chaque feuille il n'y a qu'un seul chemin pour y accéder, l'algorithme va juste parcourir cette branche et s'arrêter une fois qu'il a trouvé le résultat.

3.2 Chaînage avant

Pour le chaînage avant, nous avons eu recours à plusieurs fonctions de services. Tout d'abord, nous avons implémenté les fonctions *cclRegle*, *premiseRegle* et *numRegle* qui donnent respectivement la conclusion d'une règle, la liste des prémisses (conditions) d'une règle et enfin le numéro d'une règle.

La fonction *possibilites* va permettre de récupérer toutes les possibilités de couple dans la base de règles (hors couples qui ne servent pas de prémisses) en itérant sur la liste des prémisses des règles de la base de règle. Cette fonction nous permettra de récupérer le type de la donnée entrée par l'utilisateur lors du chaînage avant grâce à la fonction *find-by-value*. Ce choix d'implémentation apporte quelques problèmes qui seront discutés dans la partie *Problèmes et Améliorations*.

Nous avons également mis en place une fonction *verifPremise* qui va prendre en paramètre une règle et va aller vérifier si toutes ses conditions sont remplies (c'est-à-dire si toutes ses prémisses se trouvent dans la base de faits). Si c'est le cas, elle va retourner *True*.

La fonction *chainage_avant* va permettre d'ajouter les conclusions des règles validées par la fonction *verifPremise* dans la base de fait. Il aurait été possible de fusionner les

fonctions *verifPremisse* et *chainage_avant* mais nous avons préféré faire ainsi pour une meilleure visibilité lors du codage.

Enfin, la fonction *main_avant* va permettre de poser les questions à l'utilisateur et appeler la fonction *chainage_avant* pour appliquer les règles à chaque ajout d'une donnée par l'utilisateur.

Pour poser les questions, nous avons défini une variable globale **BDQ** qui contient un ensemble de variables questions définies à l'avance. Ces variables sont de la forme (valeur question possibilités). Pour poser les questions nous récupérons le dernier couple ajouté dans la base de faits et nous recherchons sa valeur dans les variables de questions. Ce choix n'est pas optimal et sera également discuté dans la partie 5 de ce rapport.

3.3 Chaînage arrière

Nous avons également eu recours à quelques fonction de services pour la réalisation du chaînage arrière.

Tout d'abord nous retrouvons une fonction *regles_candidates* qui pour un but donné en paramètre (la conclusion d'une règle) va renvoyer toutes les règles qui ont pour conclusion ce but.

La fonction *chainage_arriere* prend en paramètre un but et va retourner tous les premisses de ce but.

La fonction *main_arriere* prend en paramètre le but désiré. Pour cette fonction, nous avons été obligé d'utiliser une variable globale **liste** pour stocker les conditions non remplies par l'utilisateur. Si le but en paramètres a des conditions (c'est à dire s'il est la conclusion d'une règle), on va alors tester si chacune de ces conditions est présente dans la base de règle et si elle ne l'est pas, on rappelle récursivement la fonction en remplaçant le but par cette condition. Si le but n'a pas de conditions (donc qu'il n'est pas conclusion d'une règle), on va tester s'il est présent dans la base de fait. S'il ne l'est pas, on le rajoute dans la variable globale **liste** ce qui arrête la fonction puisqu'on a trouvé un élément non présent dans la base de fait.

3.4 Main

Nous avons également dû faire une fonction de service pour cette partie. Cette dernière appelée *liste_associations* va tout simplement renvoyer une liste contenant toutes les associations.

Pour la fonction `main`, nous commençons par demander à l'utilisateur s'il souhaite faire un chaînage avant ou arrière. S'il choisit "avant", la fonction va simplement exécuter la fonction `main_avant`. Cependant s'il choisit "arriere", la fonction va tout d'abord lui afficher la liste des associations présentent dans le SE et va lui demander d'en choisir une. Suite à cela elle va exécuter la fonction `main_arriere` et ensuite faire un test sur la valeur de `*liste*` pour savoir si cette association correspond à l'utilisateur.

Pour cette partie, nous n'avons pas réussi à trouver comment faire pour demander les caractéristiques à l'utilisateur à cause de la structure de notre arbre de déduction. De ce fait cette partie n'est pas entièrement opérationnelle, il faut remplacer les valeurs de la base de fait à la main.

4 Test des moteurs

4.1 Chaînage avant

Voici différents exemples d'exécutions du chaînage avant :

- Accès à l'association BDBDEC :

```
CG-USER(169): (main)
Bienvenue sur Choisis ton Asso ! Veuillez sélectionner le mode que vous souhaitez utiliser :
chaînage AVANT ou ARRIERE
avant
"Veuillez sélectionner le thème qui vous intéresse le plus parmi ces choix : ART, CULTURE et GESTION." culture
"Quel genre de culture vous intéresse le plus ? Culture ETRANGERE, FRANCAISE ou POP ?" pop
"Préférez-vous que cette association soit basée sur un support PHYSIQUE, VIDEOLUDIQUE ou bien un AUTRE type ?" physique
"Quel type de support physique ? Des JEUX de société ou bien des LIVRES ?" livres
D'après les informations que vous nous avez fournit, l'association qui vous convient le mieux est :
BDBDEC
```

FIGURE 2 – Exemple 1 de chaînage avant

- Accès à l'association Stravaganza :

```
CG-USER(172): (main)
Bienvenue sur Choisis ton Asso ! Veuillez sélectionner le mode que vous souhaitez utiliser :
chaînage AVANT ou ARRIERE
avant
"Veuillez sélectionner le thème qui vous intéresse le plus parmi ces choix : ART, CULTURE et GESTION." art
"Quel genre d'art préférez vous ? THEATRE, MUSIQUE ou MEDIA ?" musique
"A combien souhaitez-vous que se fassent les représentations ? (entre 1 et 10)" 8
"Que préférez-vous entre un ORCHESTRE et une FANFARE ?" orchestre
"Souhaitez-vous jouer du JAZZ ou un AUTRE_GENRE de musique ?" autre_genre
D'après les informations que vous nous avez fournit, l'association qui vous convient le mieux est :
STRAVAGANZA
```

FIGURE 3 – Exemple 2 de chaînage avant

4.2 Chaînage arrière

Pour cette partie, il est plus difficile d'effectuer les tests en raison de la non prise en compte de la mise à jour de la base de faits. Vous retrouverez donc en plus des exemple d'exécution les valeurs de la base de faits pour chaque exemple.

- Test avec GRAPHIT avec une base de faits qui correspond

```
(setq *bdf* '((type0 eq ART) (activite eq MEDIA)
(activite eq RADIO)))
```

```

CG-USER(177): (setq *bdf* '((type0 eq ART) (activite eq MEDIA) (activite eq RADIO)))
((TYPE0 EQ ART) (ACTIVITE EQ MEDIA) (ACTIVITE EQ RADIO))
CG-USER(178): (main)
Bienvenue sur Choisis ton Asso ! Veuillez sélectionner le mode que vous souhaitez utiliser :
chainage AVANT ou ARRIERE
arriere
Voici la liste des associations disponibles :
BDE
SIMDE
PTE
PSEC
PAE
PVDC
BDBDEC
LE_COIN_DU_JOUEUR
UTESCAPE
LANUTC
DEVOIR_DE_MEMOIRE
JAPONUTC
CINEMUT
ETUSEXY
DADA
AZERO
LE_CID
LE_FIL
PIKECOUD
ARCADIA
GRAPHIT
CAPHARNAUM
STRAVAGANZA
OCATA
LARSEN
FSC
PROFITROLES
Veuillez entrer le nom de l'asso désirée : graphit
"Cette association convient bien à vos critères !"

```

FIGURE 4 – Exemple 1 chaînage arrière

- Test avec la même base de faits mais avec une autre association

```

CG-USER(179): (main)
Bienvenue sur Choisis ton Asso ! Veuillez sélectionner le mode que vous souhaitez utiliser :
chainage AVANT ou ARRIERE
arriere
Voici la liste des associations disponibles :
BDE
SIMDE
PTE
PSEC
PAE
PVDC
BDBDEC
LE_COIN_DU_JOUEUR
UTESCAPE
LANUTC
DEVOIR_DE_MEMOIRE
JAPONUTC
CINEMUT
ETUSEXY
DADA
AZERO
LE_CID
LE_FIL
PIKECOUD
ARCADIA
GRAPHIT
CAPHARNAUM
STRAVAGANZA
OCATA
LARSEN
FSC
PROFITROLES
Veuillez entrer le nom de l'asso désirée : JAPONUTC
"Malheureusement, cette association ne convient pas à vos critères"

```

FIGURE 5 – Exemple 2 chaînage arrière

On peut afficher la valeur de **liste** pour voir quels sont les éléments manquants à la base de faits :

```
CG-USER(180): *liste*  
((CULTURE EQ ETRANGERE) (TYPE0 EQ CULTURE))
```

FIGURE 6 – Affichage de *liste*

5 Problèmes et Améliorations

Notre implémentation de ce système expert en LISP comporte beaucoup de défauts et de problèmes que nous n'avons pas pu résoudre par manque de temps, mais également de nombreux qui sont sujets à des améliorations.

Tout d'abord, nous n'avons pu implémenter qu'une faible partie des associations de l'UTC dans notre base. Il n'est donc pas vraiment représentatif de la diversité des associations de l'UTC.

Il y a également certains problèmes dans la conception du chaînage avant. Le fait de trouver le type de la valeur entrée par l'utilisateur en faisant une comparaison en est un exemple. Cela empêche d'avoir plusieurs fois la même valeur avec des types différents. La méthode que nous avons utilisé pour poser les questions n'est également pas optimale. Nous utilisons la dernière valeur ajoutée à la base de fait pour trouver la question à poser, ce qui pourrait potentiellement poser des problèmes.

Pour la partie chaînage arrière, le majeur problème est le fait que nous n'ayons pas trouvé comment faire pour demander à l'utilisateur de donner les caractéristiques à ajouter dans la base de fait. En effet avec notre implémentation certains types tels que *materiel* n'étaient remplis que dans certains cas et de ce fait nous ne voyions pas comment faire pour passer ce problème. Une solution envisageable serait de refaire entièrement la base de règles en faisant attention à cette contrainte que nous n'avons pas envisagé au début de ce projet. Un autre problème de ce moteur d'inférence est le fait qu'il ne réalise pas réellement de recherche en profondeur puisque nous n'avons pas implémenté de retour en arrière possible. Ici cela ne pose pas de problème mais si notre base de règle était plus développée nous aurions sûrement des problèmes.

Enfin, outre le fait de l'élargir, nous aurions souhaités pouvoir améliorer notre base de règle. En effet ici chaque branche a pour feuille une association. C'est un choix que nous avons fait au début car il nous semblait que cela nous faciliterait le travail, ce qui est vrai, mais cela nous a surtout freiner dans la créativité de ce système expert. Le fait de ne pas avoir de branche qui se rejoigne est également un choix que nous avons pris au début et qui serait à changer, c'est ces embranchements qui rendent vraiment intéressants le chaînage arrière.

Il aurait également été pertinent de rajouter une autre option dans la fonction *main* qui

permettrait à un utilisateur de sélectionner une association et l'algorithme lui afficherait toutes les caractéristiques nécessaire pour cette association.

Dans le but de perfectionner encore ce système expert, nous aurions aimé créer une autre base qui aurait contenu des listes de la forme (nom_association site_association) telle que (BDBDEC <https://assos.utc.fr/assos/bdbdec>). Cette liste nous aurait ensuite permis lors du résultat du chaînage avant d'afficher en plus du nom de l'association le lien vers sa page web pour que les utilisateurs puissent aller s'informer plus en détail sur l'association.

6 Conclusion

Bien que le rendu final soit encore imparfait et très améliorable, nous avons tous les deux apprécié travailler sur ce projet de système expert. Nous avons trouvé un sujet qui, en tant que membres de plusieurs associations de l'UTC, nous intéressait beaucoup. Nous espérons avoir réussi à convertir nos connaissances - et celles de nos camarades du monde associatif UTCéen - en un système expert fonctionnel. Nous nous sommes aussi rendus compte de la quantité de travail à fournir en dehors de l'implémentation en LISP des bases de faits et règles et du moteur d'inférences, et que la modélisation des connaissances est une expertise en soi. Nous regrettons de ne pas avoir pu passer plus de temps sur ce système expert, et, comme vous pouvez le constater dans la partie précédente il y a énormément de points que nous aurions souhaité changer si nous en avions eu le temps.