

KOROGLU MATHIS

FAURE

GUILLAUME

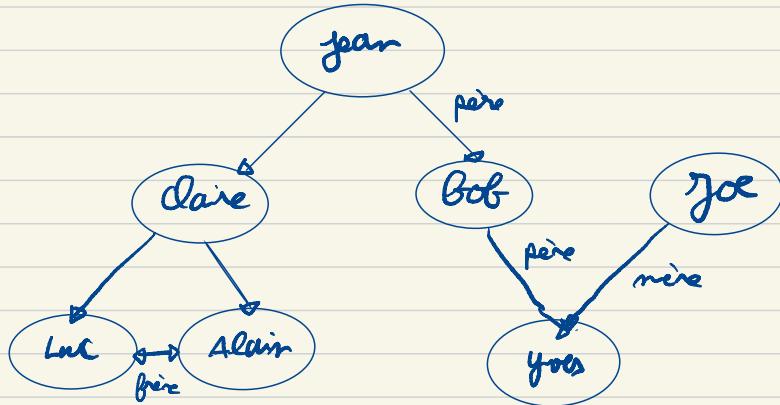
TME 3

contenu
du dossier

{ pdf TME3
+ resultat.txt
+ roug.clp
+ famille-suite.clp
+ famille unique.clp

Exercice 1

question 1



question 2 :

(reset) : retire les données de la base de fait (réinitialisation)

(rules) : permet d'afficher quelle règle pourra être appliquée

myInit : initialise la base de fait avec des faits présents dans famille.clp

grand_pere1 : père du père d'un enfant est son grand-père.

grand_mere1 : mère // mère // sa grand-mère

grand_pere2 : père d'une mère d'un enfant est son grand-père

grand_mere2 : mère d'un père // grand-mère

pere2_1 : un père est aussi un parent

pere2_2 : une mère est aussi un parent

frere_en_nen : des enfants d'un même parent sont frère et sœur ($enf1 \neq enf2$)

(facts) : affiche les faits de la base de fait connue (on peut ajouter un indice)

initialise_vide (à droite de factInit), se rapproche après (run)

(agenda) : indique si il y a des règles à appliquer

(run) : lance l'application des règles à partir de l'initialisation et explore la base de fait

question 3 :

(reset) : $\Leftarrow f=0$ (initial-fact)
 $\Rightarrow f=0$ (initial-fact)

(watch-fact) et (watch-rules) vides

(run) : déclenche les règles

(facts) : $f=0$ initial-facts

question 4 : fait un conste

question 5 :

f-0 n'est pas supprimé par le (dear), de plus après le (rest) f-0 est supprimé puis renis. On en déduit que f-0 est nécessaire pour toute la programme et si il était supprimé, la base de fait ne serait pas créée et aucune règle ne pourrait ici être appliquée

question 6 : cf fichier "famille-suite.dep"

question 7 : cf fichier

Question 8 :

La règle enfant unique est traitée avant frère et sœur ce qui fait qu'il peut exister un frère/sœur dans les que celle ci n'est pas active comme dans le but de fact

question 9 : cette fois cela fonctionne car la priorité a été modifiée

question 10 : Ajouter dans défrule dans la première
// dans la suite

(défrule ...) ?x & (predict)
⇒ (retract ?x)

Exercice 2

question 1 : après le (run) le programme continue de tourner sans que rien ne se passe

gestion fait gérée

question 2 :
→ F-8
→ F-52)

dry pred/owner? dry

→ F-53)
→ F-7)
→ F-54)

red / white? red

→ F-55)
→ F-6)
→ F-56)

light / red / full / unknown light

→ F-57)
→ F-55)
→ F-58)

delicate / very / strong strong

→ F-59
→ F-3
→ F-60

source? yes

F9
F62
F63

F64
 F1) meat/fish/poultry
 F65 meat

F10, F11

Bon de F67 F127 \Rightarrow Degrade: 92%
 : :

Question 3: cf résultat.txt

Exercice 3

question 1: cf fichier "rouge.clp"

question 2:

l'itération
 (num 1) ne permet que de décliner l'assertion de la base de fait F1 à F8 Règle 1
 (num 2) permet de décliner Règle 1 + Règle 2 \rightarrow F6, F7, F8 = F9
 (num 10) // Règle 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10

\Downarrow (num) Bon de jusqu'à saturation (plus de règle déclinable)

question 3: La stratégie de CLIPS n'en a pas modifié la séquence
 dans lequel traité 1er priorité la 1^{re} puis la 2nd ... jusqu'à saturation dans l'ordre d'écriture du programme

question 4: CLIPS procède par drainage aussi, il joint des faits pour déduire des prédictats grâce aux règles.

≠ le drainage ancien partait en dehors de la conclusion pour renfermer les prémisses un peu comme en prolog

question 5: L'équi est prouvé (en dehors de la base de fait initial):

- signe suspect
- rougeole
- douleur
- état_fébrile
- éruption cutanée
- exanthème

question 6:

- fait initial
- déduction

V = vrai
X = faux

V
Rongeole

ou

pas de nécessité d'explorer
cette branche car on a déjà
une branche à vrai

et

et

signe suspect

V

fièvre fièvre

V

et

V

peau_pâle

amydale rouge

V

taches_rouges

V

V

V