Examen MIF06 - 18 janvier 2018 – 2h (Tiers Temps + 40 min)
Documents Interdits

PARTIE 1 - MODELISATION DE PROBLEMES – 7 PTS

Takuzu (Binero) est un jeu de réflexion, cousin éloigné du Sudoku, dans lequel il s'agit de remplir une grille avec des 0 ou des 1 en respectant les 3 règles suivantes :

1. Pas plus de deux 0 ou de deux 1 de suite sur la ligne ou sur la colonne.
2. Autant de 0 que de 1 sur chaque ligne et sur chaque colonne.
3. Pas 2 lignes ou 2 colonnes identiques.

Ci-dessous, un exemple de problème initial puis ce même exemple résolu.

	1		0
		0	
	0		
1	1		0

0	1	1	0
1	0	0	1
0	0	1	1
1	1	0	0

Question 1 : Proposez une façon de modéliser ce problème pour une grille 4*4 selon le canevas présenté en cours : Etats (variables d'états) / Etat initial / Opérateurs (action, condition d'application et fonction de successeur) / Test de but / Fonction de coût.

Question 2 : Illustrez les deux premiers niveaux de l'arbre de résolution conforme à votre modélisation. Le niveau comprenant l'état initial ne compte pas comme le premier niveau !

Question 3 : Rappelez ce qu'est une heuristique. Quelle heuristique pourrait être utilisée par exemple pour ce problème ? Justifiez votre proposition.

================================= Indices de correction ======================================

Exemple de formalisation (pas la seule) :

(1pt) Etats : 16 variables Xi,j où i, j ont pour domaine de valeurs [1,4]. Chaque variable a pour domaine de valeur {1, 0, vide}. Certaines sont verrouillées car fixées au départ

(0,5 pts) Etat initial :

X11 = vide, X12 = 1, X13 = vide, X14 = 0
X21 = vide, X22 = vide, X23 = 0, X24 = vide
X31 = vide, X32 = 0, X33 = vide, X34 = vide
X41 = 1, X42 = 1, X43 = vide, X44 = 0

(1 pts) Opérateurs (action et fonction de successeur)

- Choisir une case puis un signe à mettre dessus. Fonction de successeur : la case Xij prend la valeur 1 ou 0

-Vider une case non verrouillée. Fonction de successeur : la case Xij prend la valeur vide

(1,5 pts) Test de but : les 16 variables sont remplies en respectant les contraintes suivantes :

Pas plus de 2 valeurs identiques de suite sur une ligne ou une colonne :

- Pour tout i, et pour j compris entre 1 et 2, si Xij = X i(j+1) = 1 alors X i(j+2) = 0 et vice versa
- Si Xij = X(i+1)j = 1 alors X (i+2)j = 0 et vice versa

Autant de 1 que de 0 sur chaque ligne et chaque colonnne :

- Pour 1<=i<=4 , Xi1 + Xi2 + Xi3 + Xi4 = 2
- Pour 1<=j<=4, X1j + X2j + X3j + X4j = 2

Lignes et colonnes différentes :

- Pour tout a,b, tel que a ≠ b, {Xa1,Xa2,Xa3,Xa4} ≠{Xb1,Xb2,Xb3,Xb4}
- Pour tout a,b, tel que a ≠ b, {X1a,X2a,X3a,X4a} ≠{X1b,X2b,X3b,X4b}

Fonction de coût (simple sans heuristique) : simple 1

Heuristique :

(1pt) Une heuristique =

- Un moyen d'ordonner dynamiquement la liste des successeurs selon leur « promesse de se rapprocher d'un but »
- L'expression d'une connaissance spécifique au problème à résoudre et donc indépendante de l'algorithme de recherche

(1 pt) h1 =

- Prendre en 1er les cases vides « seules » sur une ligne, diag ou colonne
- Placer les valeurs sur une ligne ou colonne quand 2 cases successives ont la même valeur

(1 pts) Arbre de résolution

===

PARTIE 2 – RAISONNEMENT LOGIQUE – 5 PTS

Question 4 : Soit la formule : F = $(\forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x))) \rightarrow \forall x(A(x)\rightarrow C(x))$

Montrez que cette formule est valide.

================================== Indices de correction ==

(1 pt) F ⇔ (¬F) ⊢ □ + étapes

$\neg F \equiv \neg\ ((\forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x))) \rightarrow \forall x(A(x)\rightarrow C(x))\ )$

Mise sous forme clausale

(1 pt) 1 - Mise sous forme normale conjonctive (FNC) : on enlève les implications et on ramène les ¬ au plus près

$\neg F \equiv \neg\ (\neg\ (\forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x))) \vee \forall x(A(x)\rightarrow C(x))\ )$

$= \forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x)) \wedge \exists x\ \neg(A(x)\rightarrow C(x))$

$= \forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x)) \wedge \exists x\ \neg(\neg A(x)\vee C(x))$

$= \forall x(A(x)\rightarrow B(x)) \wedge \forall x(B(x)\rightarrow C(x)) \wedge \exists x(A(x)\wedge\neg C(x))$

$= \forall x(\neg A(x)\vee B(x)) \wedge \forall x(\neg B(x)\vee C(x)) \wedge \exists x(A(x)\wedge\neg C(x))$

(1 pt) 2 - Mise sous forme prénexe

$\neg F \equiv \forall x\ ((\neg A(x)\vee B(x)) \wedge (\neg B(x)\vee C(x))) \wedge \exists x(A(x)\wedge\neg C(x))$

$\neg F \equiv \exists x\ (A(x)\wedge\neg C(x)) \wedge \forall x((\neg A(x)\vee B(x)) \wedge (\neg B(x)\vee C(x)))$

$\equiv \exists y\ \forall x\ (A(y)\wedge\neg C(y)) \wedge ((\neg A(x)\vee B(x)) \wedge (\neg B(x)\vee C(x)))$

(0,5 pt) 3 – Skolémisation

$\neg F \equiv \forall x\ (A(a)\wedge\neg C(a)) \wedge \forall x((\neg A(x)\vee B(x)) \wedge (\neg B(x)\vee C(x)))$

H0 = { a } où a est une constante de Skolem

(0,5 pt) Clauses : { $C1 = A(a)$; $C2 = \neg C(a)$; $C3 = \neg A(x1)\vee B(x1)$; $C4 = \neg B(x2)\vee C(x2)$ }

(1 pt) Preuve par réfutation :

C1 et C3 avec a/x1 => C5 = B(a)

C5 et C4 avec a/x2 => C6 = C(a)

C6 et C2 => □

==

PARTIE 3 – SYSTEME A BASE DE CONNAISSANCES – 4 PTS

On considère un système expert dont les règles et les faits sont les suivants :

R 1 : Si le candidat a actuellement un poste à responsabilité
Et le candidat a des facilités pour apprendre les langues
Et le candidat parle Français
Alors le candidat est dynamique

R 2 : Si le candidat a des facilités pour apprendre les langues
Et le candidat parle anglais
Alors le candidat a une bonne adaptabilité

R 3 : Si le candidat est slave
Et le candidat est dynamique
Alors le candidat a une bonne adaptabilité

R 4 : Si le candidat a actuellement un poste à responsabilité
Alors le candidat a une capacité de leadership

R5: Si le candidat a des facilités pour apprendre les langues
Alors le candidat parle néerlandais

R6: Si le candidat a une bonne adaptabilité
Et le candidat a une capacité de leadership
Alors le candidat est accepté

R7: Si le candidat est slave
Alors le candidat a des facilités pour apprendre les langues

R8: Si le candidat a une capacité de leadership
Et le candidat est slave
Alors le candidat a une bonne adaptabilité

Faits :

- Le candidat a actuellement un poste à responsabilité
- Le candidat est slave

Question 5 : Peut-on obtenir le but « le candidat est accepté » ? Démontrez votre réponse en utilisant un moteur d'inférence en chaînage arrière et en montrant à chaque fois la règle à utiliser et les cas de succès et d'échecs.

================================ Indices de correction =====================================

But : le candidat est accepté

(0,5 pts) Pas dans la base de fait et on utilise la règle R6
Nouveaux buts :
But 6.1 : Le candidat a une bonne adaptabilité
But 6.2 : Le candidat a une capacité de leadership

(1 pt) Pour But 6.1, pas dans la base de fait et on utilise R2
Nouveaux buts :
But 6.1.1 : Le candidat a des facilités pour apprendre les langues
But 6.1.2 : Le candidat parle anglais

Pour But 6.1.1, pas dans la base de fait et on utilise R7
Nouveaux buts :
But 6.1.1.1 : Le candidat est slave
Ok car dans la base de fait

But 6.1.2 : Le candidat parle anglais
Pas dans la base de fait et aucune règle
Donc R7 n'aboutit pas et pas d'autre règle donc but 6.1.1 ne peut être prouvé

(1 pt) Donc Pour But 6.1 R2 n'aboutit pas, on tente R3
Nouveaux buts :
But 6.1.1bis : Le candidat est slave
But 6.1.2bis : Le candidat est dynamique

Le But 6.1.1bis est dans la base de faits.

Pour But 6.1.2bis, pas dans la base de fait et on utilise R1
Nouveaux buts :
But 6.1.2bis.1 : Le candidat a actuellement un poste à responsabilité
But 6.1.2bis.2 : Le candidat a des facilités pour apprendre les langues
But 6.1.2bis.1 : Le candidat parle Français

Le But 6.1.1bis.1 est dans la base de faits.
Le but 6.1.2bis.2 est le même que le But 6.1.1 qui ne peut pas être prouvé.
Donc R1 n'aboutit pas et pas d'autre règle donc but 6.1.2bis ne peut être prouvé

(1 pt) Donc Pour But 6.1 R3 n'aboutit pas, on tente R8
Nouveaux buts :
But 6.1.1ter : Si le candidat a une capacité de leadership
But 6.1.2ter : Le candidat est slave

Pour But 6.1.1ter, pas dans la base de fait et on utilise R4
Nouveaux buts :
But 6.1.1ter.1 : Le candidat a actuellement un poste à responsabilité
Dans la base de faits donc R4 aboutit
Donc le but 6.1.1ter est prouvé avec R4

On passe au But 6.1.2ter
Ok car dans la base de faits

Donc le But 6.1 est prouvé avec R8

(0,5 pt) On passe au But 6.2 : Le candidat a une capacité de leadership
Ce but a déjà été prouvé avec R4

Donc le but « le candidat est accepté » est prouvé.

==

PARTIE 4 – PROLOG – 4 PTS

Question 6 : Définissez le prédicat `inverse(L,I)` qui inverse la liste L pour créer la liste I.

Question 7 : Dessinez les arbres de résolution et indiquez la réponse que votre programme donnera aux requêtes suivantes :

- `inverse([a,b,c],L).`
- `inverse([a,b,c,d],[d,c,b,a]).`

Question 8 : Définissez le prédicat `palindrome(L)` qui indique qu'une liste est un palindrome. Pour rappel, une phrase est un palindrome si l'ordre des éléments reste le même qu'on le lise de gauche à droite ou de droite à gauche, comme dans la phrase « Ésope reste ici et se repose » ou encore « La mariée ira mal ». Vous pouvez utiliser le prédicat `inverse` défini précédemment…

================================ Indices de correction =====================================

```
(1,5 pt)
inverse([],[]).
inverse([First|Suite], Cible) :-
      inverse(Suite,Toto),append(Toto,[First],Cible).

(1,5 pt) pour les arbres de résolutions

(1 pts)
palindrome(L) :- inverse(L, L).
```

==