

Examen MIF06 - 6 janvier 2020 – 2h (+ Tiers Temps 40 min)

Documents Interdits

PARTIE 1 - MODELISATION DE PROBLEMES – 5 PTS

Étant données n couleurs, un carré latin d'ordre n est un carré $n \times n$ colorié tel que :

- Toute cellule est coloriée ;
- Chaque couleur apparaît exactement une fois sur chaque ligne ;
- Chaque couleur apparaît exactement une fois sur chaque colonne.



La figure ci-contre montre un exemple de carré latin d'ordre 4.

Question 1 : Proposez une modélisation de ce problème pour $N=4$ sous forme d'un CSP.

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

Exemple de formalisation (pas la seule) :

(1 pts) Variables :

$S = [[X_{11}, X_{12}, X_{13}, X_{14}], [X_{21}, X_{22}, X_{23}, X_{24}], [X_{31}, X_{32}, X_{33}, X_{34}], [X_{41}, X_{42}, X_{43}, X_{44}]]$
avec X_{ij} représentant une case (i, j) avec $1 \leq i \leq 4, 1 \leq j \leq 4$, soit 16 variables

(1 pts) Domaines :

$D_{xij} = \{1,2,3,4\}, \forall (i, j)$ ou Domaine = [rouge, vert, bleu, jaune].

(3 pts) Contraintes :

Pour tout $i \in [1 ;4]$, pour tout $j \in [1 ;4]$, pour tout $k \in [1 ;4]$, si $j \neq k$ alors $X_{ij} \neq X_{ik}$

Pour tout $i \in [1 ;4]$, pour tout $j \in [1 ;4]$, pour tout $k \in [1 ;4]$, si $i \neq k$ alors $X_{ij} \neq X_{kj}$

PARTIE 2 – RAISONNEMENT LOGIQUE – 5 PTS

Question 2 : Comment démontrer qu'un raisonnement est valide à l'aide de la logique des prédicats ? Illustrez vos propos en montrant que la formule suivante est universellement valide :

$$F = \forall x \exists y \forall z (R(x, z) \rightarrow R(x, y))$$

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

1 : formaliser le pb (toutes les clauses et la négation de la conclusion)

2 : mise sous forme clausale i.e. FNC = conjonction de disjonctions, puis préfixe, puis skolemisation (fixe toutes les \exists)

3 : preuve par réfutation avec unification dans le monde de Herbrand

4 : conclusion : si on a un ensemble d'instances de base insatisfiable alors le raisonnement est valide

La mise sous forme standard de Skolem de la négation de la conclusion (et donc de la formule) donne : **(1pt negation, 1pt prenex + skolem)**

$$\begin{aligned} \neg F &= \exists x \forall y \exists z (R(x, z) \wedge \neg R(x, y)) \\ &= \exists f \forall y (R(a, f(y)) \wedge \neg R(a, y)) \end{aligned}$$

Ce qui nous donne la forme clausale de la conjonction des expressions précédentes : **(1pt forme clausale)**

$$C\neg F = \{R(a, f(y_1)), \neg R(a, y_2)\}$$

Dans l'univers de Herbrand $H_0 = \{a, f(a)\}$ **(1 pt Herbrand)**

En unifiant y_1/a et $y_2/f(a)$, $C\neg F = \{R(a, f(a)), \neg R(a, f(a))\}$ est insatisfiable. **(1pt unification & clause vide)**

La négation de F ne peut donc être vraie, la formule est donc universellement valide.

PARTIE 3 – SYSTEME A BASE DE CONNAISSANCES – 5 PTS

Nous avons observé l'organisation des réunions au sein d'une petite entreprise et nous en avons tiré les informations suivantes :

- si la réunion commence en retard, Alphonse n'y sera pas ;
- si Jules et Eva assistent à la réunion, elle ne sera pas calme ;
- si Sophie vient, la réunion commencera en retard ;
- si Marc vient à la réunion, Eva vient aussi ;
- si Jules, Sophie et Eva assistent à la réunion, et si Alphonse ne vient pas, la réunion sera fructueuse.

Question 3 : À partir des informations recueillies, définissez une base de règles qui pourra être exploitée par un moteur d'inférence pour prédire le déroulement des réunions en fonction des participants.

Question 4 : Sachant que pour la prochaine réunion, Jules et Marc seront présents. Que peut-on déduire sur le déroulement de la réunion ? Quel mécanisme d'inférence avez-vous utilisé ? Déroulez-le sur papier.

Question 5 : Sachant que pour la réunion d'après, Jules, Marc et Sophie seront présents. Est-ce que la réunion sera fructueuse ? Quel mécanisme d'inférence avez-vous utilisé ? Déroulez-le sur papier.

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

Base de règles : (1 pts)

R1 : Si retard Alors non(reunion(alphonse))

R2 : Si reunion(jules) & reunion(eva) Alors non(calme)

R3 : Si reunion(sophie) Alors retard

R4 : Si reunion(marc) Alors reunion(eva)

R5 : Si reunion(jules) & reunion(sophie) & reunion(eva) & non(reunion(alphonse)) Alors fructueuse

(1 pour nom avant-arrière)

(1,5 pts pour solution) Avec un chainage avant

BF = {reunion(jules), reunion(marc)}

1^{ère} boucle sur les règles :

R4 => BF = {reunion(jules), reunion(marc), reunion(eva)}

On reboucle sur les règles.

R2 => BF = {reunion(jules), reunion(marc), reunion(eva), non(calme)}

On reboucle sur les règles mais sans changement, donc on s'arrête.

On peut donc dire que « la réunion ne sera pas calme » si Jules et Marc sont présents.

(1,5 pts pour solution) Avec un chainage arrière

BF = {reunion(jules), reunion(marc), reunion(sophie)}

But : prouver fructueuse

R5 prouvera fructueuse si on a : reunion(jules) & reunion(sophie) & reunion(eva) & non(reunion(alphonse))

Or reunion(jules) et reunion(sophie) sont dans la BF

Reste à prouver reunion(eva) et non(reunion(alphonse))

R4 prouvera reunion(eva) si on a reunion(marc)

Or reunion(marc) est dans la BF donc on peut prouver reunion(eva)

R1 prouvera non(reunion(alphonse)) si on a retard

Reste à prouver retard

R3 prouvera retard si on a reunion(sophie)

Or reunion(sophie) est dans la BF, donc on a prouvé retard donc on a prouvé non(reunion(alphonse))

On a donc prouvé chacun des prémisses de la règle R5, donc on a prouvé que la réunion sera fructueuse.

PARTIE 4 – PROLOG – 5 PTS

Qui a le temps de faire de la confiture...

Pour le savoir, observons la situation :

- Marie possède du sucre, des figes, des pommes, de la vanille et dispose de 45 min de libre ;
- Nathalie possède du sucre, des abricots, des noisettes et dispose de 35 min de libre ;
- Laetitia possède du sucre, des abricots, des figes, de la vanille et dispose de 45 min.

En prenant un livre de recettes de confitures, on peut savoir que pour faire de la confiture de figes, il faut des figes mais aussi du sucre et de la vanille alors que pour faire de la confiture d'abricots, il faut remplacer les figes par des abricots et garder les autres ingrédients à l'identique. La confiture de figes cuit 30 min alors que celle d'abricot cuit 25 min.

De plus, le temps de préparation de chacun des ingrédients est le suivant :

- Sucre : 1 min
- Pommes (pelées et coupées en morceaux) : 15 min
- Figes (lavées et coupées en morceaux) : 10 min
- Abricots (lavés et coupés en morceaux) : 5 min
- Noisettes (pelées et coupées en morceaux) : 15 min
- Vanille (coupée en deux) : 1 min

Nous allons maintenant coder ce problème en Prolog.

Nous utiliserons les prédicats suivant pour décrire la situation :

- `possede (P, I)` qui sera vrai si une personne possède l'ingrédient I ;
- `temps (P, T)` qui sera vrai si une personne dispose du temps T de libre ;

Nous obtenons donc la base de faits suivante :

```

possede(marie, sucre).
possede(marie, pomme).
possede(marie, figue).
possede(marie, vanille).
possede(nathalie, sucre).
possede(nathalie, noisette).
possede(nathalie, abricot).
possede(laetitia, sucre).
possede(laetitia, abricot).
possede(laetitia, figue).
possede(laetitia, vanille).

temps(marie,45).
temps(nathalie,35).
temps(laetitia,45).

```

Question 6 : Définissez les prédicats permettant de décrire les recettes des confitures de figues et d'abricots.

Question 7 : Définissez le prédicat `quialetemps(X, Conf)` permettant de savoir qui a le temps de faire quelle confiture.

Question 8 : Définissez le prédicat `quipeut(X, Conf)` permettant de savoir qui peut faire quelle confiture.

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

Question 6 : 1,5 pts

```

preparation(sucre, 1).
preparation(pomme, 15).
preparation(figue, 10).
preparation(abricot, 5).
preparation(noisette, 15).
preparation(vanille, 1).

cuisson(conf_figue, 30).
cuisson(conf_abricot, 25).

ingredients(conf_figue, [sucre, figue, vanille]).
ingredients(conf_abricot, [sucre, abricot, vanille]).

```

Question 7 : 2,5 pts

```

prepaingredient([], Td, Td).
prepaingredient([X|L], Td, Tr) :-
    preparation(X, T),
    T1 is Td-T,
    T1>=0,
    prepaingredient(L, T1, Tr).

quialetemps(X, Conf) :-
    temps(X, T),
    ingredients(Conf, Liste),
    cuisson(Conf, Tc),
    T1 is T-Tc,

```

```
T1 >= 0,  
prepaingredient(Liste, T1, TR),  
TR>=0.
```

Question 8 : 1 pts

```
prepaingredient2(_, [], Td, Td).  
prepaingredient2(P, [X|L], Td, Tr) :-  
    possede(P, X),  
    preparation(X, T),  
    T1 is Td-T,  
    T1>=0,  
    prepaingredient2(P, L, T1, Tr).
```

```
quipeut(X, Conf) :-  
    temps(X, T),  
    ingredients(Conf, Liste),  
    cuisson(Conf, Tc),  
    T1 is T-Tc,  
    T1 >= 0,  
    prepaingredient2(X, Liste, T1, TR),  
    TR>=0.
```
