

Examen MIF06 - 25 janvier 2023 – 2h (+ 40 min Tiers-temps)

Documents Interdits

PARTIE 1 - MODELISATION DE PROBLEMES – 5 PTS

Paul et Bernard sont dans un restaurant. Paul a commandé 2 fois et Bernard 4 fois (chacun pour eux, sans rien commander pour l'autre).

A chaque commande, Bernard prend 3 petites frites et 1 hamburger, tandis que Paul prend 1 grande frite et 3 hamburgers.

Sachant qu'une grande frite vaut 2 petites frites, mais qu'à chaque fois que Bernard commande, Paul lui pique un paquet de frites et à chaque fois que Paul commande, Bernard lui prend un hamburger, lequel a mangé le plus ?

Question 1 : Pour m'aider à répondre à la question, modéliser le problème sous forme d'un CSP afin de savoir qui a mangé quoi. Il vous est demandé de modéliser et non pas de résoudre le problème.

Indices de correction

Exemple de formalisation (pas la seule) :

(1 pt) Les variables : On considère 4 variables, une pour chaque aliment et pour chaque personne :

PaulPartFrite, PaulHamburger, BernardPartFrite, BernardHamburger.

(1 pt) Le domaine : Chaque variable a pour domaine de valeur le nombre de portions, soit un entier naturel positif ou nul.

(3 pts) Les contraintes :

Soit PaulCommande = 2 et BernardCommande = 4, il faut :

- $\text{PaulPartFrite} = \text{PaulCommande} * (2*1) + 1 * \text{BernardCommande}$
- $\text{PaulHamburger} = \text{PaulCommande} * (3-1)$
- $\text{BernardPartFrite} = \text{BernardCommande} * (3-1)$
- $\text{BernardHamburger} = \text{BernardCommande} * 1 + 1 * \text{PaulCommande}$

Résultat non demandé :

- PaulPartFrite = 8, PaulHamburger = 4,
 - BernardPartFrite = 8, BernardHamburger = 6
-
-

PARTIE 2 – RAISONNEMENT LOGIQUE – 4 PTS

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

Question 2 : Montrez que la formule F est valide, en expliquant bien les étapes du raisonnement.

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

$F \Leftrightarrow (\neg F) \vdash \square + \text{étapes}$

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

Mise sous forme clausale

(1 pt) 1 - Mise sous forme normale conjonctive (FNC) : on enlève les implications et on ramène les \neg 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

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

$H_0 = \{ a \}$ où a est une constante de Skolem

(0,5 pt) Clauses : $\{ C_1 = A(a) ; C_2 = \neg C(a) ; C_3 = \neg A(x_1) \vee B(x_1) ; C_4 = \neg B(x_2) \vee C(x_2) \}$

(1 pt) Preuve par réfutation :

$$C_1 \text{ et } C_3 \text{ avec } a/x_1 \Rightarrow C_5 = B(a)$$

$$C_5 \text{ et } C_4 \text{ avec } a/x_2 \Rightarrow C_6 = C(a)$$

$$C_6 \text{ et } C_2 \Rightarrow \square$$

PARTIE 3 – SYSTEME A BASE DE CONNAISSANCES – 6 PTS

On dispose d'un système à base de connaissances permettant de valider un trajet en fonction d'un contexte. La base de connaissances est constituée d'un ensemble de 9 règles. Le moteur d'inférence de ce système à base de connaissances fonctionne en chaînage arrière et sa stratégie de résolution de conflit consiste à choisir la 1^{ère} règle disponible dans sa base.

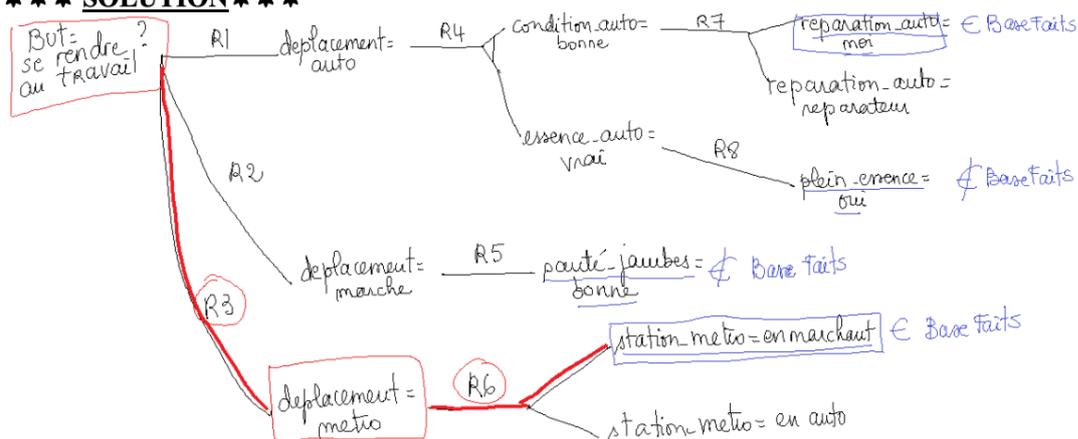
À partir des 9 règles données ci-dessous, je voudrais savoir si Paul peut se rendre au travail, et comment. On sait seulement que Paul répare lui-même son auto et qu'il a un problème de santé à ses jambes. Cependant, il est capable d'effectuer de courts trajets à pied comme se rendre à la station de métro.

- R1 : SI déplacement_auto ALORS se_rendre_au_travail
- R2 : SI déplacement_marche ALORS se_rendre_au_travail
- R3 : SI déplacement_métro ALORS se_rendre_au_travail
- R4 : SI condition_auto_bonne ET essence_auto ALORS déplacement_auto
- R5 : SI santé_jambes_bonne ALORS déplacement_marche
- R6 : SI station_métro_en_marchant OU station_métro_en_auto ALORS déplacement_métro
- R7 : SI reparation_auto_a_domicile OU reparation_auto_au_reparateur ALORS condition_auto_bonne
- R8 : SI plein_essence ALORS essence_auto
- R9 : SI condition_auto_bonne ET essence_auto ALORS station_métro_en_auto

Question 3 : Déroulez le fonctionnement du système, en expliquant à chaque étape les règles utilisées et le contenu de la base de faits.

Indices de correction

*** SOLUTION ***



Le graphe se lit de haut en bas et de gauche à droite. C'est donc le 3^{ème} chemin qui valide le trajet de Paul. Les faits sont encadrés en bleu (de la base de faits) et en rouge (faits déduits). Le trajet de solution est en gras rouge.

Cet exercice est tiré du cours : <https://www.ulaval.ca/etudes/cours/ift-2003-intelligence-artificielle-i>

Sachant maintenant que je n'ai pas de problème de santé et donc que j'ai de bonnes jambes, que je ne sais pas réparer mon auto mais que j'ai un garagiste en qui j'ai toute confiance et qui m'a fait le plein d'essence, je voudrais savoir si je peux me rendre au travail et par quel moyen de transport ?

Question 4 : En changeant le moteur d'inférence du système pour fonctionner en chaînage avant, quelles informations peux-tu déduire à partir de ma situation ? Déroulez le fonctionnement du système, en expliquant à chaque étape les règles utilisées et le contenu de la base de faits.

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

BF initiale = { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur }

R5 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche }

R7 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne }

R8 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne, essence_auto }

R9 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne, essence_auto, station_métro_en_auto }

On recommence :

R2 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne, essence_auto, station_métro_en_auto, se_rendre_au_travail }

R4 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne, essence_auto, station_métro_en_auto, se_rendre_au_travail, deplacement_auto }

R6 => { santé_jambes_bonne, non(reparation_auto_a_domicile), reparation_auto_au_reparateur, deplacement_marche, condition_auto_bonne, essence_auto, station_métro_en_auto, se_rendre_au_travail, deplacement_auto, deplacement_métro }

On recommence : R1 et R3 activée pour même BF

Fin

=====

PARTIE 4 – PROLOG – 5 PTS

Recherche dans un graphe d'états : problème de la simplification d'un nombre

On souhaite simplifier un nombre considéré comme une liste de chiffres. On prendra l'exemple du nombre 3413242341. L'objectif est d'arriver au mot vide. Les règles de simplification sont les suivantes :

- R1 : si deux chiffres adjacents sont égaux, on peut les supprimer tous les deux.
- R2 : on peut permuter deux chiffres adjacents si la valeur absolue de leur différence est strictement supérieure à 1.
- R3 : la séquence $(n, n-1, n)$ peut être remplacée par la séquence $(n-1, n, n-1)$.

Soient les prédicats `recherche` et `resoudre` vus en TP :

```
recherche(Ef, Ef, _, []) :- !.
recherche(Ec, Ef, Letats, [Op|Lop]) :-
    operateur(Ec, Op, Es),
    not(member(Es, Letats)),
    not(interdit(Es)),
    write(Ec), write(' '),
    write(Op), write(' '), write(Es), nl,
    recherche(Es, Ef, [Es|Letats], Lop).
```

```
resoudre(Sol) :- initial(Ei), final(Ef), recherche(Ei, Ef, [Ei], Sol).
```

Question 5 : Définissez les prédicats `initial`, `final` et `operateur` pour ce problème.

 Indices de correction

```
(2pts) remplace(L1, L2, L, R) :- append(Deb, L3, L), append(L1, Fin, L3),
    append(Deb, L2, L4), append(L4, Fin, R).
```

```
(0.5pts) initial([3, 4, 1, 3, 2, 4, 2, 3, 4, 1]).
    final([]).
```

```
(0.5pts) operateur(E1, r1, E2) :- remplace([N, N], [], E1, E2).
```

```
(1pt) operateur(E1, r2, E2) :- remplace([M, N], [N, M], E1, E2), MmN is abs(M-N),
    MmN > 1.
```

```
(1pt) operateur(E1, r3, E2) :- remplace([M, N, M], [N, M, N], E1, E2), N =:= M-1.
```
