

Examen Session 2 – Mars 2021

Durée : 2 heures

Aucun document autorisé

Partie 1 : Résolution de CSP

8 points

Soit le réseau de contraintes $R=(X,D,C)$ où $X=\{v,w,x,y,z\}$, $D(v)=D(w)=D(x)=D(y)=D(z)=\{a,b\}$ et C contient les contraintes en extension suivante :

- $C_1\langle v,z \rangle = \{ (a,a), (a,b), (b,a) \}$
- $C_2\langle w,x \rangle = \{ (a,b), (b,a) \}$
- $C_3\langle w,z \rangle = \{ (a,a), (b,a), (b,b) \}$
- $C_4\langle y,z \rangle = \{ (a,a) \}$
- $C_5\langle x,y \rangle = \{ (a,a), (a,b), (b,b) \}$
- $C_6\langle x,z \rangle = \{ (a,a), (b,a), (b,b) \}$

Question 1. Représentez l'exécution de l'algorithme de **Backtrack** pour la recherche d'une solution en utilisant v,w,x,y,z comme **ordre** d'affectation des variables et a,b comme **ordre** de test des valeurs. Vous représenterez l'arbre de recherche développé en indiquant sur **chaque nœud la variable et la valeur assignée** et sur **chaque feuille la contrainte violée ou succès**.

Question 2. Quelle est la solution trouvée ?

Question 3. Dessinez le graphe des contraintes.

Question 4. Calculez l'ordre d'assignation des variables engendré par l'heuristique statique combinée suivante : max-card (variable connectée au plus grand nombre de variables déjà choisies), puis max-degree (plus grand degré dans le graphe de contraintes) en cas d'égalité, puis enfin l'ordre lexicographique si deux variables sont encore de même priorité. Étant donné cet ordre, **vous préciserez quelles contraintes** devront « réellement » être vérifiées par l'algorithme de backtrack après chaque assignation de variable.

Question 5. Réappliquez l'algorithme de backtrack à la recherche d'une solution en utilisant l'ordre calculé à la question 4 et l'ordre croissant sur les valeurs.

Question 6. L'ordre sur les valeurs est-il important lorsqu'on utilise l'algorithme de backtrack à la recherche d'une solution (par exemple si l'on prenait l'ordre b,a au lieu de a,b). Discutez de l'impact de l'ordre sur les valeurs sur le temps de calcul de cette première solution et sur l'arbre de recherche développé pour trouver cette première solution. Cela peut-il changer la première solution trouvée ?

Question 7. Représentez l'exécution de l'algorithme de **Forward Checking** pour la recherche de toutes les solutions en utilisant l'heuristique *dom+deg+alpha* (plus petit domaine courant, puis plus grand degré dans le graphe de contraintes puis ordre alpha-numérique si égalité) pour l'**ordre d'assignation des variables** et a,b comme **ordre** de test des valeurs. On indiquera à **chaque nœud de l'arbre la variable assignée, la valeur choisie et les modifications aux domaines** des variables restantes par propagation.

Question 8. Quelles sont toutes les solutions de ce réseau de contraintes ?

Soit la base de faits $BF = \{A, B\}$ et la base de règles BR suivante :

- $R1 : B \wedge C \rightarrow D$
- $R2 : D \wedge A \rightarrow F$
- $R3 : F \wedge G \rightarrow H$
- $R4 : B \wedge I \rightarrow H$
- $R5 : A \wedge H \rightarrow I$
- $R6 : F \wedge A \rightarrow H$
- $R7 : G \rightarrow E$
- $R8 : A \rightarrow C$

Question 1. On effectue l'algorithme de **chaînage avant** avec la technique des compteurs associés aux règles. Indiquez quel est l'état de chaque compteur à la fin de la saturation de BF par BR . On ne vous demande pas d'indiquer l'état des compteurs avant ou pendant la saturation.

Question 2. Avec la même base de connaissances, on cherche à prouver **H** en **chaînage arrière**. Dessinez l'arbre de recherche correspondant à la remontée du graphe ET-OU, en supposant que l'algorithme considère les règles par numéro croissant et les symboles par ordre d'apparition dans l'hypothèse de la règle considérée. Vous indiquerez sur chaque feuille traitée : échec, déjà échec, (appartient à) BF , déjà prouvé, boucle.

Le but H est-il finalement prouvé ?

Question 3. Le chaînage avant sur les règles positives en logique des propositions est **complet**. Choisir dans les phrases ci-dessous celle(s) qui traduisent cette propriété (où BF^* désigne la base de faits obtenue à partir de BF et BR en chaînage avant).

1. Pour tout symbole A , si $A \in BF^*$ alors $BF, BR \models A$
2. Pour tout symbole A , si $BF, BR \models A$ alors $A \in BF^*$
3. Pour tout symbole A , si $A \notin BF^*$ alors $BF, BR \not\models A$
4. Pour tout symbole A , si $BF, BR \not\models A$ alors $A \notin BF^*$

Question 4. Considérons des bases de connaissances où les règles comportent des littéraux (A ou $\neg A$) et pas seulement des symboles (littéraux positifs). La négation est ici la négation logique classique. Le chaînage avant reste-t-il complet ? Justifiez votre réponse.

Question 5. Soit la base de règles suivante où *not* représente la négation du monde clos. Cet ensemble est-il stratifiable ? Si oui, donner l'une de ses stratifications. Sinon, montrez qu'il n'est pas stratifiable.

- $R1 : A \wedge \text{not } B \rightarrow D$
- $R2 : C \wedge \text{not } A \rightarrow B$
- $R3 : E \wedge \text{not } A \rightarrow C$
- $R4 : E \wedge F \rightarrow A$
- $R5 : D \rightarrow F$

Question 6. On se place maintenant en logique du premier ordre et on considère la base de connaissances $\mathcal{K} = (\mathcal{F}, \mathcal{R})$ suivante :

- $\mathcal{R} = \{R1, R2\}$ avec :

R1 : $r(x1, y1, z1) \wedge p(x1, y1) \rightarrow p(y1, z1)$

R2 : $p(x2, y2) \wedge p(y2, z2) \rightarrow p(x2, z2)$

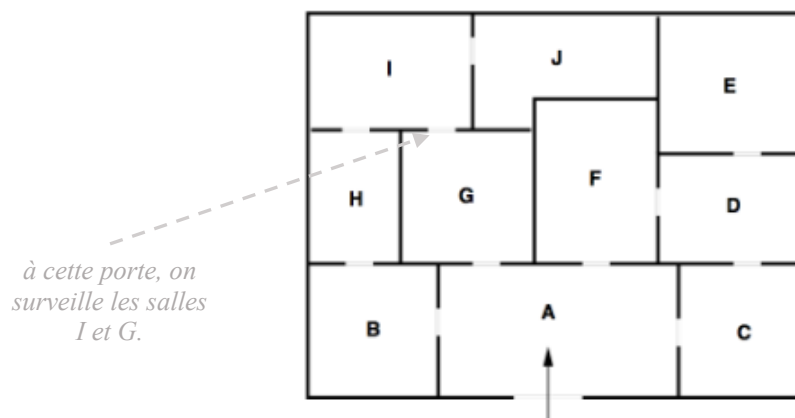
Les quantificateurs universels sont implicites

- $\mathcal{F} = \{ r(a,b,c), r(b,b,c), r(b,c,d), r(a,d,d), p(a,b), p(b,b), p(c,d) \}$ où a,b,c,d sont des constantes
- A. Soit la requête Q qui demande de trouver tous les x,y et z tels que $r(x,y,d) \wedge p(y,z)$. Quelles sont les réponses à Q sur \mathcal{F} (sans considérer la base de règles) ?
- B. Quelle est la base de faits obtenue en saturant \mathcal{F} avec les règles de \mathcal{R} ?
- C. Quelles sont les réponses à Q sur $\mathcal{K} = (\mathcal{F}, \mathcal{R})$?

Partie 3. Modélisation en CSP

4 points

Un musée dispose d'un effectif de 6 gardiens. On cherche à affecter ces gardiens de telle sorte que chaque salle soit surveillée (cf. plan ci-dessous). Les gardiens peuvent être placés uniquement aux portes de telle sorte qu'ils surveillent 2 salles (donc pas à la porte d'entrée).



Modéliser ce problème comme un CSP.