

Examen MIF06 - 22 janvier 2026 – 1h30 (+ 30 min Tiers-temps)

Documents Interdits

PARTIE 1 - MODELISATION DE PROBLEMES – 7 PTS

Le musée de l'Histoire de l'Informatique prépare une nouvelle exposition dédiée à l'IA. Elle doit présenter six moments importants de l'IA qui doivent être présentés dans des salles différentes, numérotées de 1 à 6, que les visiteurs parcourront dans l'ordre de 1 à 6.

Les trois moments sont :

- 1950 – le test de Turing ;
- 1956 – la conférence de Dartmouth ;
- 1957 – le perceptron de Rosenblatt ;
- 1966 – ELIZA, le premier chatbot ;
- 1997 – Deep Blue bat le champion d'échecs Garry Kasparov ;
- 2016 – AlphaGo bat le champion du Go.

L'équipe du musée doit décider quel moment placer dans quelle salle, mais plusieurs informations doivent être prises en compte.

Pour montrer clairement l'évolution des idées sur le dialogue homme-machine, le musée souhaite que le test de Turing soit présenté avant ELIZA. Cependant, le perceptron ne peut pas être placé juste après la conférence de Dartmouth, car les organisateurs estiment que cela créerait trop de répétitions historiques.

Pour illustrer ensuite la manière dont les machines ont surpassé les humains dans certains jeux, Deep Blue doit apparaître dans une salle située après ELIZA. En revanche, la salle n°1 ne peut pas accueillir AlphaGo, car cette invention est jugée trop moderne pour ouvrir la visite. Le dernier stand doit être réservé à une invention liée aux jeux.

De plus, ELIZA ne peut pas être placée juste avant AlphaGo, car les deux thèmes sont considérés comme trop éloignés. Par ailleurs, le perceptron doit être présenté après la conférence de Dartmouth, afin de respecter la chronologie historique réelle.

Pour des raisons esthétiques, la conférence de Dartmouth doit occuper une salle portant un numéro impair.

Pour finir, chaque salle ne peut présenter qu'une seule invention, et toutes les inventions doivent apparaître exactement une fois dans le parcours.

Question 1 : Modélisez ce problème sous forme d'un CSP afin de pouvoir le résoudre avec les algorithmes vus en cours. Il vous est demandé de modéliser non pas de résoudre le problème.

PARTIE 2 – SYSTEME A BASE DE CONNAISSANCES – 6 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 2 : 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 3 : En 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 4 : En 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.

PARTIE 3 – PROLOG – 7 PTS

On souhaite utiliser la **recherche dans un graphe d'états** pour résoudre le petit casse-tête suivant.

On dispose de trois bidons A, B, C de capacités respectives 3, 5 et 8 litres. Au départ, on a 8 litres d'eau dans A. Les bidons B et C sont vides. On souhaite obtenir exactement 4 litres dans l'un des trois bidons, en transvasant le liquide d'un bidon à l'autre. On peut transvaser d'un bidon vers un autre que jusqu'à ce que le bidon cible soit plein ou que le bidon source soit vide. On ne peut pas jeter de liquide : le volume total reste 8 litres.

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 : Expliquez à quoi servent les prédicats `initial` et `final`. Définissez-les pour ce problème.

Question 6 : Expliquez à quoi sert le prédicat `operateur` et chacun de ses paramètres en précisant les entrées et les sorties. Définissez un des prédicats `operateur`.

Question 7 : En TP, nous avons implémenté le prédicat `saturer`. L'algorithme utilisé exploite une double boucle. Pour quoi faire ? Et comment la coder en Prolog ?