Mestrado em Engenharia Electrotécnica e de Computadores (MEEC)
Licenciatura em Engenharia Electrotécnica e de Computadores (LEEC)
Licenciatura em Engenharia Aeroespacial (LEA)
Departamento de Engenharia Electrotécnica e de Computadores
Instituto Superior Técnico

Inteligência Artificial e Sistemas de Decisão

2003/2004

1º Exame

14/Janeiro/2004

Notas:

- O exame tem a duração de 3h00m.
- Preencha cuidadosamente o cabeçalho da folha de exame e identifique todas as restantes folhas utilizadas.
- Leia calmamente todo o enunciado e comece a responder às perguntas que lhe pareçam mais acessíveis.
- Justifique claramente todas as respostas dadas.
- Os alunos que queiram desistir só o poderão fazer após ter decorrido uma hora de exame.

Boa Sorte!

---

1. [2] Para os seguintes exemplos, escolha justificadamente a estrutura de agente inteligente que considera mais adequada à resolução do problema envolvido.

   (a) Robot empacotador. Percepções: pixeis; Acções: pegar e colocar peças; Objectivo: colocar as peças nos locais correctos; Ambiente: tapete rolante com peças.

   (b) Tutor interactivo de português. Percepções: palavras escritas; Acções: exercícios, sugestões e correcções; Objectivo: aprendizagem do estudante; Ambiente: escola.

2. [2.5] Habitualmente nos problemas de procura em jogos assume-se que as regras do jogo estabelecem uma função de utilidade que é utilizada pelos dois jogadores (MAX e MIN) e que uma utilidade de x para MAX representa uma utilidade de $-x$ para MIN. Jogos com esta propriedade designam-se por "jogos de soma nula" (*zero-sum games*). Considere em contrapartida o problema de procura em jogos de soma não nula. Para tal assuma que a função de avaliação devolve uma lista com dois valores, indicando a utilidade do jogador MAX e MIN, respectivamente. Suponha ainda que cada jogador procura maximizar a sua utilidade independentemente da utilidade do adversário.

   (a) Para a árvore de procura seguinte indique os valores de avaliação de todos os nós em falta incluindo o do nó raiz. Com base nesses valores, identifique a jogada do MAX.

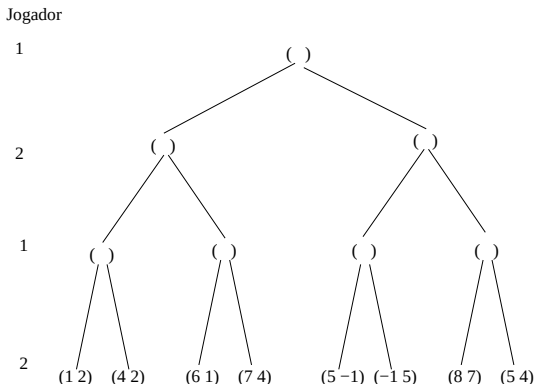

Figure 1: Árvore de um jogo de soma não nula

(b) Considere as funções seguintes escritas em pseudo-código (retiradas do livro base da disciplina) que implementam o algoritmo MINIMAX. Altere-as de modo a que o algoritmo MINIMAX funcione correctamente para um jogo de soma não nula, nas condições da alínea anterior.

```
function Minimax-Decision ( game ) returns an operator

         for each op in Operators[game] do
             Value[op] = Minimax-Value ( Apply( op,game ), game )
         end

         return the op with the highest Value[op]

function Minimax-Value ( state, game ) returns a utility value

         if Terminal-Test[game]( state ) then
            return Utility[game]( state )
         else if MAX is to move in state then
            return the highest Minimax-Value of Successors( state )
         else
            return the lowest Minimax-Value of Successors( state )
```

(c) Explique porque é que o algoritmo ALFABETA não pode ser facilmente generalizado para este tipo de jogos.

3. [2.5] Considere um problema de procura em espaço de estados e a possibilidade de utilizar o algoritmo A* para resolver o problema, recorrendo a uma função de avaliação $f = g + h$ estritamente monótona.

   Um algoritmo alternativo ao A* é o algoritmo *depth-first branch-and-bound* (BBS). Ao contrário do A* o BBS é um algoritmo de procura em profundidade, o que significa que é razoável utilizá-lo quando a árvore for finita. Essencialmente, o algoritmo mantém o valor do melhor nó objectivo encontrado, actualizando-o sempre que encontrar um melhor. Faz procura em profundidade e retorna (*backtracking* sempre que encontra um nó terminal ou nó com um valor superior ao ao melhor até ai encontrado. Nesta última situação, o nó (e consequentemente todos os seus sucessores) é eliminado (pruned). O algoritmo utiliza uma variável global *upper_bound* para guardar o melhor nó encontrado, inicializada a ∞, e é iniciado chamando *BBS(initial_state)*. Termina quando todos os nós foram explorados ou eliminados. O custo óptimo será dado pelo valor da variável *upper_bound*, no final da procura.

```
BBS1(node)
{
  if (f(node) >= upper_node)
     upper_node = f(node);
     return
  if (isGoal(node))
     upper_node =
        max(upper_node, f(node));
  else
     for each s in Successors(node)
        BBS1(s);
  return;
}
```

```
BBS2(node)
{
  if (f(node) >= upper_node)
     return
  if (isGoal(node))
     upper_node =
        min(upper_node, g(node));
  else
     for each s in Successors(node)
        BBS2(s);
  return;
}
```

```
BBS3(node)
{
  if (f(node) >= upper_node)
     return
  if (isGoal(node))
     upper_node = min(upper_node, g(node));
  else
     for each s in Successors(node)
        if (f(s) <= upper_node)
           upper_node = g(s)
        BBS3(s);
  return;
}
```

(a) Indique justificadamente qual das funções anteriores implementa correctamente o algoritmo BBS.

(b) Mostre que o algoritmo BBS é completo e óptimo.

(c) Demonstra-se que o A* nunca expande mais nós que o BBS. Todavia, o BBS é bastante utilizado como algoritmo de procura óptimo. Apresente duas razões para preferir o BBS ao A*.

4. [3]

(a) Considere o conceito de independência condicional. Suponha que se pretende determinar $P(H|E_1 \wedge E_2)$ e não há qualquer informação sobre independência condicional.

i. Indique **justificadamente** qual (ou quais) dos seguintes conjuntos de probabilidades é suficiente para determinar a probabilidade pretendida.

A. $P(E_1 \wedge E_2), P(H), P(E_1|H), P(E_2|H)$

B. $P(E_1 \wedge E_2), P(H), P(E_1 \wedge E_2|H)$

C. $P(E_1|H), P(E_2|H), P(H)$

ii. Suponha agora que tem a informação que E_1 e E_2 são independentes condicionalmente dado H. Indique **justificadamente** qual (ou quais) dos conjuntos anteriores é suficiente para determinar a probabilidade pretendida.

(b) Suponha que dois astrónomos, em locais diferentes do mundo, realizam medições M_1 e M_2 do número de estrelas N, numa pequena região do céu, utilizando os seus telescópios. Normalmente, existe uma possibilidade (baixa) de erro de contagem de 1 estrela. Cada telescópio pode ainda, com baixa probabilidade, estar ligeiramente desfocado, o que introduz um erro de contagem de 2 estrelas. Considere o seguinte conjunto de valores possíveis para o número de estrelas: $\{0, 1, 2, 3, > 3\}$.

i. Proponha uma rede bayesiana que modele a informação anterior. Escolha com critério as probabilidades envolvidas.

ii. Determine com base na rede proposta, a probabilidade de haver 2 estrelas dado que os astrónomos obtiveram as contagens $M_1 = 1$ e $M_2 = 3$ e os dois telescópios estavam correctamente focados.

5. [2]

   (a) Utilize o sistema de dedução natural de Fitch para provar que

   $$\{\phi \Rightarrow \neg\psi\} \vdash \psi \Rightarrow \neg\phi$$

   (b) Mostre, recorrendo ao teorema da dedução e ao resultado anterior, que

   $$\text{se} \quad \Delta \cup \{\phi\} \vdash \neg\psi \quad \text{então} \quad \Delta \cup \{\psi\} \vdash \neg\phi$$

6. [2.5] Considere as seguintes frases em linguagem natural:

   (a) "Flores coloridas são sempre bem-cheirosas"

   (b) "Eu não gosto de flores que não cresçam em espaço aberto"

   (c) "Nenhuma flor que cresça em espaço aberto é descolorida"

   (d) "Eu não gosto de flores que não sejam bem-cheirosas".

   Prove por resolução que a quarta frase é consequência lógica das três primeiras.

7. [2]

   (a) Explique o significado de "consequência lógica" e "inconsistência". Mostre que, dadas as fórmulas $F_1, \ldots, F_n$ e a fórmula G, G é consequência lógica de $F_1, \ldots, F_n$ se e só se a fórmula $(F_1 \wedge \ldots \wedge F_n \wedge \neg G)$ fôr inconsistente.

   (b) Complete a frase "Uma lógica é decidível se ..."

8. [1.5] Considere um problema, para resolver em LISP, em que os dados são listas cujos elementos ou são listas ou números inteiros.

   Escreva uma função LISP que devolva a sub-lista iniciada no elemento maior da lista de topo.

   Nota: A lista de topo é constituída por números inteiros e listas. Para os efeitos desta função, o valor númerico de uma lista é a soma de todos os elementos que a constituem.

9. [2] Suponha que é um médico que procura aconselhar um paciente que ocasionalmente revela uma reacção alérgica após ter tomado uma refeição num restaurante. Depois de conversar com o paciente, conseguiu produzir uma tabela com exemplos, tendo identificado os seguintes atributos: i) restaurante (Samuel, Luis ou Sara); ii) refeição (pequeno-almoço, almoço); iii) dia-semana (sexta, sábado, domingo); iv) custo da refeição (barato, caro).

Exemplos	Restaurante	Refeição	Dia	Custo	Alergia
X_1	Samuel	PA	Sex	B	S
X_2	Luis	A	Sex	C	N
X_3	Samuel	A	Sab	B	S
X_4	Sara	PA	Dom	B	N
X_5	Samuel	PA	Dom	C	N

   (a) Com base nestes exemplos, construa uma árvore de decisão através do algoritmo de aprendizagem indutiva que permita antecipar a reacção do paciente a uma refeição.

   (b) Escreva as fórmulas de lógica de primeira ordem que traduzem o conhecimento expresso pela árvore de decisão criada.

10. Questões relacionadas com o trabalho prático

    (a) Descreva resumidamente a forma como o seu grupo para o trabalho prático implementou o modelo de representação de fórmulas bem formadas na Lógica Proposicional.

    (b) Explique com o detalhe necessário a forma como o programa desenvolvido deve ser executado, enumerando e descrevendo os eventuais argumentos da função a ser chamada, e/ou as perguntas colocadas ao utilizador e/ou os parâmetros fixos do programa.