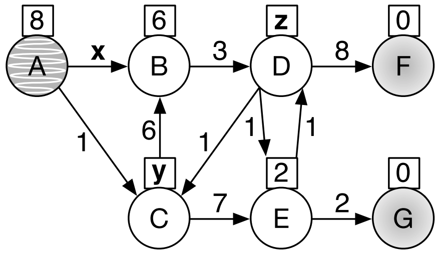
Inteligência Artificial – 2018/19

27/Outubro/18 — 9:00 — Duração: 2h

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

1º Teste
– Com consulta limitada –

I) [6val] Considere o seguinte grafo de estados de um problema de procura. Os valores apresentados nos arcos correspondem ao custo do operador (ação) respectivo, enquanto os valores nos rectângulos correspondem ao valor de uma heurística (estimativa do custo de chegar desse estado ao objectivo). Não se representam os nomes dos operadores, correspondendo cada arco a um operador distinto. Assuma que os sucessores de um nó são sempre gerados por ordem alfabética do nome do nó (ou seja, por exemplo, dos sucessores de D, o primeiro a ser gerado é o C, o segundo é o E e o terceiro é o F). A ordem alfabética deverá também ser utilizada sempre que for necessário desempatar entre nós. Pretende-se encontrar um caminho desde o estado A até ao estado F ou G.

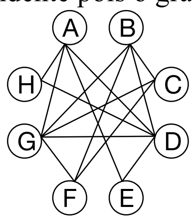


a) Indique os intervalos de valores para **x**, **y** e **z** de forma a que a heurística seja admissível. Justifique.

b) Considere os algoritmos de procura em largura primeiro (optimizada) , procura de custo uniforme (em grafos) e procura sôfrega (em grafos). Para cada um dos algoritmos, assumindo que x=4, y=1 e z=0, indicar:
   - Qual o caminho encontrado pelo algoritmo
   - Quais os nós que são expandidos

c) Ainda assumindo que x=4, y=1 e z=0, ilustre como se comporta o algoritmo de procura A* em grafos na resolução deste problema. Use a versão do algoritmo que garante a obtenção da solução óptima dadas as características da heurística. Deve explicitar os conteúdos das estruturas de dados auxiliares ao longo das iterações do algoritmo, colocando entre parêntesis o valor da função de avaliação para cada nó na lista. Indique o caminho encontrado pelo algoritmo, assim como o seu custo.

II) [3val] Seja G um grafo não dirigido com N vértices. Um **conjunto independente** em G é um conjunto de vértices Z tal que nenhum par de vértices I,J pertencentes a Z estão ligados por um arco. Por exemplo, no grafo da figura abaixo, o conjunto {E,F,H} é um conjunto independente pois o grafo não contém nenhum dos arcos EF, EH e FH.



O problema do conjunto independente é: dado um grafo G e um inteiro K≤N, encontrar um conjunto independente em G de tamanho K. Pretendemos resolver o problema do conjunto independente usando um algoritmo de procura cego (e.g. procura em largura primeiro, procura em profundidade primeiro ou procura por aprofundamento progressivo).

a) Descreva o espaço de estados para este problema, de forma a que possa ser resolvido de forma eficiente. Deverá descrever:
   i. Os estados ii. Os operadores, ou os estados sucessores iii. O estado inicial iv. O objectivo.

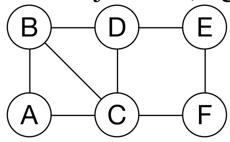
b) Qual é a profundidade e o factor de ramificação do espaço de estados? A profundidade do objectivo menos profundo é conhecida? Em caso afirmativo, qual é?

c) Qual seria o melhor algoritmo para este problema: procura em largura primeiro, procura em profundidade primeiro, ou procura por aprofundamento progressivo? Justifique sucintamente.

III) [3val] Verifique, usando o algoritmo DPLL, se as seguintes cláusulas são satisfazíveis:

$$(\neg P \lor Q \lor R) \land (P \lor \neg Q) \land (P \lor \neg R) \land (\neg P \lor \neg Q \lor S) \land (\neg Q \lor R \lor \neg S) \land (\neg R \lor \neg S) \land (\neg A \lor P) \land (\neg A \lor R)$$

IV) [4val] Considere o seguinte grafo de restrições, representando um mapa que pretendemos colorir com três cores (R)osa, (V)ermelha e (P)úrpura, de modo a que países adjacentes, ligados por um arco, não tenham a mesma cor.



a) Assumindo que à variável C já foi atribuída a cor Vermelha, aplique o algoritmo de Verificação para a Frente (Forward Checking) e elimine (na folha de resposta) as cores que seriam descartadas para as restantes variáveis.

b) Assumindo que à variável A já foi atribuída a cor Púrpura e á variável C a cor Rosa, aplique o algoritmo de Consistência de Arcos (AC-3) e elimine (na folha de resposta) as cores que seriam descartadas para as restantes variáveis.

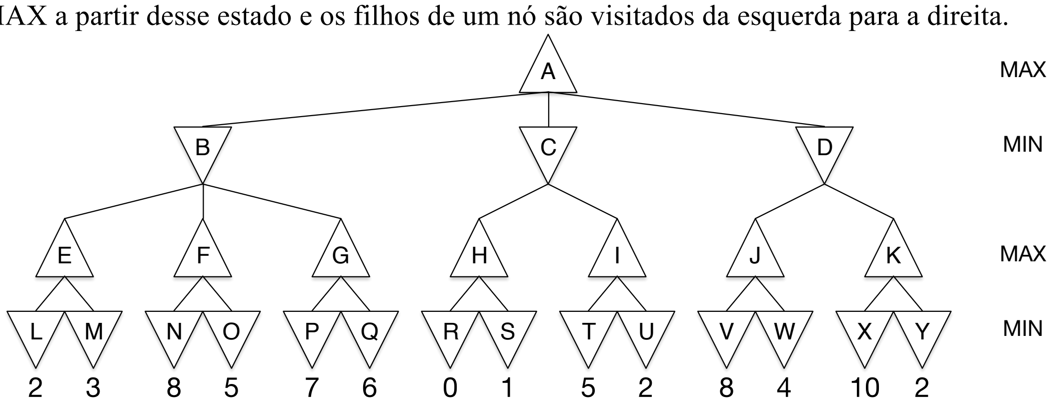
c) Considere os seguintes domínios possíveis, obtidos após seleção da variável B e atribuição da cor Vermelha, seguida da propagação de restrições. Das restantes 5 variáveis, quais seriam selecionadas pela heurística da variável mais constrangedora para serem consideradas a seguir?

A	**B**	**C**	**D**	**E**	**F**
R P	V	R P	R P	R V P	R V P

d) Considere o seguinte conjunto inconsistente de atribuições. A variável C acabou de ser escolhida para lhe ser atribuída uma cor durante a execução de um algoritmo de procura local para encontrar uma atribuição completa e consistente. Que cor iria ser atribuída a C de acordo com a heurística min-conflitos?

A	**B**	**C**	**D**	**E**	**F**
P	V		V	V	P

V) [4val] Considere a árvore de jogo de dois jogadores (MAX e MIN), onde os valores nas folhas são estimativas do ganho para MAX a partir desse estado e os filhos de um nó são visitados da esquerda para a direita.



a) Calcule o valor MINIMAX de cada nó não terminal e indique qual o movimento que MAX deverá escolher.

b) Suponha que pode alterar os valores nas folhas de modo a maximizar a quantidade de nós que nunca chegariam a ser visitados pelo algoritmo de busca $\alpha-\beta$. Nesse caso, que nós não chegariam a ser visitados?

c) Considere agora que os nós B, C e D da árvore da figura são nós estocásticos (CHANCE) onde cada sucessor é equiprovável. (Por exemplo, no nó B, existe uma probabilidade de 1/3 de ir para cada um dos sucessores.) Calcule o valor EXPECTEDMINIMAX de cada nó não terminal.

VI) [Bónus: até 2val] Para cada alínea, indique se ela é verdadeira ou falsa. Cada resposta correta vale 0,4val, cada resposta errada vale -0,4val. A pergunta tem uma cotação mínima de 0 valores.

a) Para um problema de procura, o caminho devolvido pelo algoritmo de procura de custo uniforme pode mudar se adicionarmos um valor constante positivo *c* ao custo de cada uma das acções.

b) Para um problema de procura, o caminho devolvido pelo algoritmo de procura A* usando uma função heurística $h(s)=c$ (onde *c* é um valor constante positivo) é garantidamente óptimo.

c) Se $h_1(s)$ e $h_2(s)$ são duas heurísticas diferentes, ambas admissíveis, então é melhor usar a heurística $h_3(s)=max(h_1(s),h_2(s))$ do que a heurística $h_4(s)=min(h_1(s),h_2(s))$.

d) Para um problema de procura, o algoritmo de pesquisa A* expande sempre menos nós do que o algoritmo de procura de custo uniforme.

e) Num espaço de estados finito que não contém qualquer estado objectivo, todos os estados são atingíveis a partir do estado inicial, e o custo de todos os operadores é positivo, o algoritmo de procura A* irá sempre explorar todos os estados.

Inteligência Artificial – Folha de Respostas

27/Out/2018 – 9:00 **DI/FCT/UNL, 1º Teste, 2018/19, Duração: 2h** **Teste:** adbdadca

Nome: **Número:**

I.a)

x: **[2, +∞]** y: **[0, 9]** z: **[0, 3]**

Justificação: Uma heurística h é admissível quando, para cada estado n, o valor de h(n) é menor ou igual ao menor custo c*(n) para se chegar ao objectivo a partir de n.

Para o cálculo de **x**, dado que h(A) é menor do que o custo do melhor caminho através de C i.e. h(A) < c(A→C→D→E→G), a única restrição que resta satisfazer é que h(A) também seja menor do que o custo do melhor caminho através de B i.e. h(A) ≤ c(A→B→D→E→G). Ou seja, temos que garantir que 8 ≤ x + 3 + 1 + 2. Daqui resulta que **x** ≥ 2.

Para o caso de **y**, a única restrição a satisfazer é que h(C) ≤ c*(C) onde h(C)=**y** e c*(C)=c(C→E→G) = 9. Daqui resulta que **y** ≤ 9.

Para o caso de **z**, a única restrição que resta satisfazer é que h(D) ≤ c*(D) onde h(D)=**y** e c*(D)=c(D→E→G) = 3. Daqui resulta que **y** ≤ 3.

I.b) Algoritmo	**Solução**	**Expandidos**
largura primeiro	**A→B→D→F**	**A B C D**
custo uniforme	**A→C→E→G**	**A B C D E**
sôfrega	**A→C→E→D→F**	**A C E D**

I.c)

Solução: **A→C→E→G** (há outra execução que resulta em **A→B→D→E→G**) Custo da Solução: **10**

* dado que a heurística não é consistente, é necessário usar a versão do A* adequada.

Fronteira:	A(8)	C(2) B(10)	B(10)E(10)B(13)	D(7) E(10) B(13)	C(9) E(10) E(10)' B(13) F(15)
Explorados:		A(8)	A(8) C(2)	A(8) C(2) B(10)	A(8) C(2) B(10) D(7)

Fronteira (cont.):	E(10) E(10)' B(13) F(15)	D(9) E(10)' G(10) B(13) F(15)	E(10)' G(10) B(13) F(15)
Explorados (cont.):	A(8) C(2) B(10) D(7)	A(8) C(2) B(10) D(7) E(10)	A(8) C(2) B(10) D(7) E(10)

Fronteira (cont.):	G(10) B(13) F(15)	B(13) F(15)
Explorados (cont.):	A(8) C(2) B(10) D(7) E(10)	A(8) C(2) B(10) D(7) E(10)

II.a)

i. **Estado**: Lista ordenada alfabeticamente com no máximo K vértices, tal que nenhum par de vértices esteja ligado por um arco.

ii. Operador: dado um estado S, acrescentar um vértice ao fim da lista (i.e. alfabeticamente maior do que os vértices em S) que não esteja ligado por um arco a qualquer dos vértices de S.

iii. Estado Inicial: Lista vazia.

iv. Objectivo: Um estado com K vértices.

II.b)
Profundidade do espaço de estados: **K** Factor de Ramificação: **N (na raiz)**

Profundidade do objectivo menos profundo (desconhecida / indique a profundidade): **K**

II.c)
Resposta: **procura em profundidade primeiro**
Justificação: Como as soluções se encontram todas à profundidade K, não há razão para estar a pesquisar por aprofundamento progressivo, pois não se iria encontrar qualquer solução com profundidade menor do que K. Por outro lado, como a profundidade do espaço de pesquisa está limitada a K – a mesma profundidade onde estão as soluções – ao usar o algoritmo de procura em profundidade primeiro, não corremos o risco de estar à procura num ramo mais profundo do que o necessário para encontrar uma solução, que seria a razão para usar o algoritmo de pesquisa em largura primeiro.

III)
Satisfazível (sim/não): **SIM** Se sim, indique um modelo: A→F; P→V; Q→F; R→V; S→F.
Verificação: Já está na forma clausal.
Aplicação do DPLL:

$$\begin{array}{c} (\neg P \lor Q \lor R) \\ (P \lor \neg Q) \\ (P \lor \neg R) \\ (\neg P \lor \neg Q \lor S) \\ (\neg Q \lor R \lor \neg S) \\ (\neg R \lor \neg S) \\ (\neg A \lor P) \\ (\neg A \lor R) \end{array} \xRightarrow[(\neg A)]{Pure} \begin{array}{c} (\neg P \lor Q \lor R) \\ (P \lor \neg Q) \\ (P \lor \neg R) \\ (\neg P \lor \neg Q \lor S) \\ (\neg Q \lor R \lor \neg S) \\ (\neg R \lor \neg S) \end{array} \xRightarrow[(P)]{Split} \begin{array}{c} (Q \lor R) \\ (\neg Q \lor S) \\ (\neg Q \lor R \lor \neg S) \\ (\neg R \lor \neg S) \end{array} \xRightarrow[(Q)]{Split} \begin{array}{c} (S) \\ (R \lor \neg S) \\ (\neg R \lor \neg S) \end{array} \xRightarrow[(S)]{Unit} \begin{array}{c} (R) \\ (\neg R) \end{array} \xRightarrow[(R)]{Unit} \bot$$

$$\xRightarrow[(\neg Q)]{BackTrack} \begin{array}{c} (R) \\ (\neg R \lor \neg S) \end{array} \xRightarrow[(R)]{Unit} (\neg S) \xRightarrow[(\neg S)]{Unit} \top$$

Há outras alternativas válidas...

IV.

a)

A	B	C	D	E	F
R ▮ P	R ▮ P	V	R ▮ P	R V P	R ▮ P

b)

A	B	C	D	E	F
P	▮ V ▮	R	▮ ▮ P	R V ▮	▮ V P

c)
Variáveis: **C**

d)
Cor: **Rosa**

V.a)
A:**8** B:**3** C:**1** D:**8** E:**3** F:**8** G:**7** H:**1** I:**5** J:**8** K:**10** Movim. de MAX: **D**

b)
Nós não visitados: **O Q I T U K X Y**

c)
A:**9** B:**6** C:**3** D:**9** E:**3** F:**8** G:**7** H:**1** I:**5** J:**8** K:**10**

VI.a)	b)	c)	d)	e)
VERDADERIA	**VERDADERIA**	**VERDADERIA**	**FALSA**	**VERDADERIA**