

Introdução à Inteligência Artificial

Ano 2002/03 – 1ª Chamada

3 horas / com consulta

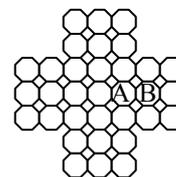
Grupo 1 (Pesquisa)

Um agricultor está na margem dum rio com uma raposa, uma galinha e um saco de milho, e quer levar tudo para a outra margem. Para tal dispõe de um barco pequeno onde só pode levar um (ou nenhum) dos itens de cada vez. Além disso, para garantir que não perde nada durante o processo, não pode dar-se ao luxo de deixar a raposa com a galinha numa das margens enquanto ele está na outra margem (se o fizer, arrisca-se a que a raposa coma a galinha). O mesmo se passa relativamente à galinha e ao milho. O que se pretende é saber que viagens é que o agricultor terá que fazer no barco, e transportando o quê em cada uma delas, por forma a no final ter a raposa, a galinha e o milho sãos e salvos na outra margem. Além disso pretende-se que tudo seja feito com o menor número de viagens possível.

- 1 a) Diga como formalizaria o problema por forma a que este pudesse ser resolvido por um algoritmo normal de pesquisa em espaço de estados. Ou seja, diga como definiria estados, o estado inicial, a função de teste para estados objectivo e a função que devolve os sucessores dum estado.
- 1 b) Uma pesquisa em profundidade seria completa para a resolução **deste problema**? Justifique, gizando um exemplo concreto de evolução deste algoritmo neste problema que mostre da completude ou não.
- 1 c) Defina uma heurística admissível (não trivial) para que este problema possa ser melhor resolvido usando o algoritmo A*. Justifique que de facto a heurística que apresenta é admissível.
- 1 d) Suponha agora uma variante deste problema em que, em vez de se pretender um número mínimo de viagens, se pretende saber a solução com um número de viagens fixo à partida (por exemplo 7 viagens). Se quisesse resolver este novo problema com um algoritmo iterativo de melhoramento (tipo “hill climbing”), como definiria *solução*, *vizinhança* e a *função de avaliação* do custo de cada solução.

Grupo 2 (Planeamento)

Certamente que conhece o jogo do solitário. Neste jogo existe um tabuleiro, semelhante ao que se mostra na figura, e onde inicialmente todas as casas do tabuleiro, com excepção da do meio, têm uma pedra. O objectivo é chegar a uma configuração onde só exista uma pedra no centro do tabuleiro. Para retirar uma peça do tabuleiro há que fazer um movimento em que se desloca uma peça adjacente àquela que se quer retirar para um posição livre do tabuleiro, também adjacente à peça a retirar, e em linha com a peça que se está a mover. No fundo, isto é semelhante à forma como se comem peças num jogo de damas, só que aqui também se podem comer peças com movimentos na horizontal, vertical e quer para a frente quer para trás (ao invés das damas em que só se podem comer peças na diagonal, para a frente). Na figura, para retirar uma peça que esteja na posição A, pode deslocar-se uma peça em B para o centro do tabuleiro. Pretende resolver-se o jogo do solitário usando um algoritmo de planeamento de ordens parciais.



- 2 a) Comece por descrever as acções de *start* e *finish* com que o algoritmo do POP começaria para resolver este problema. Para tal vai ter que se usar um conjunto de predicados que descrevam um estado do problema. Sugere-se o uso dos predicados *livre(Posição)* e *com_Peça(Posição)* para a descrição de estados.
- 2 b) Apresente um operador STRIPS que descreva um movimento no jogo do solitário. Assuma que está definido um predicado *em_linha(P1,P2,P3)* que é verdade sempre que as posições P1, P2 e P3 estão em linha e que P1 é adjacente de P2 e P2 é adjacente de P3.
- 2 c) Descreva como é que pode ocorrer uma situação em que, na aplicação do algoritmo do POP a este problema, um passo ameaça um link já existente no plano parcial que se está a construir. De um exemplo concreto numa tal ameaça.
- 2 d) Descreva brevemente, por palavras suas, como é que o algoritmo do POP evoluiria para a resolução deste problema, por oposição à forma como evoluiria um algoritmo de pesquisa em profundidade.
- 2 e) Parece-lhe uma boa ideia resolver este problema usando um planeador de ordens parciais? Será que se ganha alguma coisa na complexidade de resolução deste problema pelo facto do resultado pretendido ser uma ordem parcial e não uma ordem total, como se obteria por exemplo usando um algoritmo de pesquisa? Justifique.

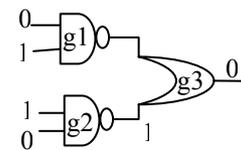
Grupo 3 (Redes de Bayes)

Um jardim perto de minha casa é regado de noite com alguma frequência. Diria que é regado em cerca de 50% das noites. Mesmo quando sei que o jardim foi regado à noite, verifico que nem sempre a relva do jardim está molhada de manhã. Além disso, por vezes chove durante a noite (considere-se que 10% das noites chove), o que também influencia o facto da relva estar ou não molhada pela manhã. Fazendo uma análise à posteriori, verifica-se que se chover então se o jardim for regado durante a noite 90% das vezes a relva está molhada pela manhã, e se não for regado só 80% das vezes é que está molhado. Se não chover então, mesmo havendo rega à noite, só 60% das vezes é que a relva está molhada (talvez porque quando não chove está em regra mais calor, fazendo com que mais vezes a relva entretanto seque). Se não chover e não houver rega, então a relva está garantidamente seca. Além disso, quando chove não é só a relva que pode ficar molhada. De facto, em 50% das manhãs seguintes a uma noite em que choveu, o chão da rua também está molhado. Sempre que não chove durante a noite o chão está seco na manhã seguinte.

- 3 a) Diga que variáveis aleatórias escolheria para representar o conhecimento acima e, para cada uma delas, qual o seu domínio.
- 3 b) Apresente uma rede Bayesiana (topologia e tabelas nos nós) que represente esse conhecimento
- 3 c) Num dado dia, ia a sair de casa e, antes mesmo de olhar para o chão da rua, verifiquei que a relva estava molhada. Calcule:
- 3 c1) a probabilidade de ter chovido durante essa noite.
- 3 c2) a probabilidade de, ao depois olhar para o chão da rua, vir a constatar que este também está molhado.

Grupo 4 (Representação do Conhecimento)

Pretende-se implementar um sistema de diagnóstico de circuitos digitais usando *Answer-Set Programming*. Para simplificar a tarefa, vamos apenas considerar circuitos com dois tipos de gates (nands e ors de 2 entradas) e assumir que nunca há avarias nas ligações entre as gates. Assume-se ainda que sempre que uma gate está avariada tem o valor 0 (zero) na sua saída.



A ideia é fazer uma modelação genérica de problemas de diagnóstico destes

circuitos de tal forma que, se a ela adicionarmos uma representação da topologia do circuito e factos sobre observações feitas nalguns dos pontos do mesmo, haja um conjunto de resposta por cada diagnóstico possível do circuito em causa contendo átomos da forma `avariado(gate)` por cada gate avariada de acordo com esse diagnóstico. No exemplo da figura, terá que haver um conjunto de resposta com `{avariado(g3)}` e outro com `{avariado(g1), avariado(g3)}`.

Para isso há que escolher uma forma adequada de representar circuitos, e observações medidas em circuitos. Vamos adoptar a seguinte: para cada gate do tipo `Tipo`, identificada pelo nome `Id`, há um facto `gate(Id,Tipo)`; os pontos do circuito correspondem unicamente a entradas ou saídas de gates, representadas, respectivamente, por termos `in(NumPorta,IdGate)` e `out(IdGate)`; para cada ligação há um facto da forma `ligado(Ponto1,Ponto2)`; para cada valor observado há um facto da forma `val(Ponto,Valor)`. Por exemplo, o circuito e observações da figura são representados pelos factos:

```
ligado(out(g1),in(1,g3)).      gate(g1,nand).  val(in(1,g1),0).  val(in(2,g1),1).
ligado(out(g2),in(2,g3)).      gate(g2,nand).  val(in(1,g2),1).  val(in(2,g2),0).
gate(g3,or).                   val(out(g2),1). val(out(g3),0).
```

Repare que com esta representação até é possível prever o valor em todos os pontos do circuito para os quais é conhecido o input. Por exemplo, para o caso duma gate nand a funcionar correctamente:

```
val(out(G), V) :- gate(G,nand), not avariada(G),
                  val(in(1,G),V1), val(in(2,G),V2), nand(V1,V2,V).
```

onde `nand/3` é um predicado com 4 factos descrevendo o comportamento normal de uma gate nand.

Há problema sempre que o valor previsto num qualquer ponto do circuito for diferente do valor observado nesse ponto.

- 4 a) Faça a modelação completa do sistema de diagnóstico descrito acima.
- 4 b) Em geral o número de diagnósticos possíveis é muito grande, mas só muito raramente é que o que se está de facto a passar envolve mais do que uma gate. Altere a modelação que fez por forma a que passem a ser devolvidos apenas os diagnóstico que envolvem uma ou zero gates avariadas.
- 4 c) Na prática, para além do tipo de avaria aqui considerado (*stuck_at_zero*) há outro: *stuck_at_one*, *shunted*, *negated_result*, etc. Diga brevemente como alteraria a sua modelação para tratar vários tipos de avarias.