

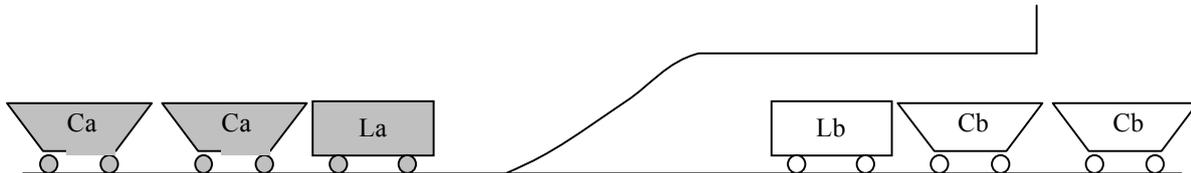
## Introdução à Inteligência Artificial

### Ano 2003/04 – Época Recurso

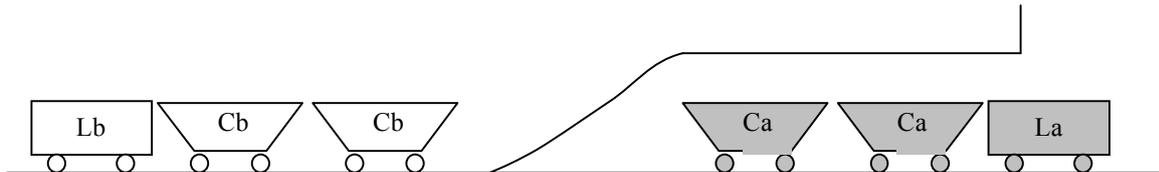
3 horas / com consulta

#### Grupo 1 (Pesquisa)

Dois comboios de mercadorias encontram-se frente a frente numa linha, ambos com duas carruagens e uma locomotiva. Existe um ramal de manobra que se encontra ligado ao lado esquerdo da linha principal (ver Figura), mas com espaço suficiente apenas para a locomotiva e uma carruagem. Salientamos novamente que o ramal só é acessível a partir do lado esquerdo da linha principal. A linha principal pode conter qualquer número de carruagens e locomotivas, de ambos os lados do ramal. As locomotivas podem movimentar-se em ambos os sentidos. As locomotivas puxam sempre todas as carruagens (e locomotiva) que lhe estão opcionalmente atreladas, mas empurrando todas as carruagens (e outra locomotiva) que se encontram à sua frente no sentido do movimento. A situação de partida está descrita na figura abaixo:

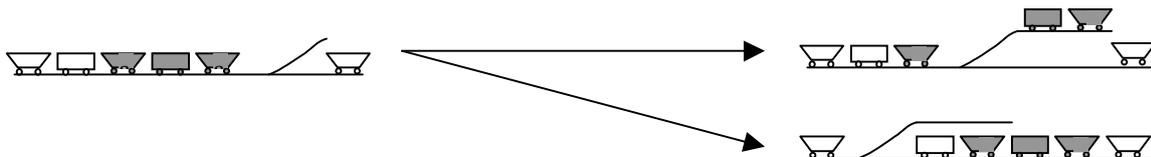


O objectivo é descobrir o conjunto mínimo de manobras (movimentos das locomotivas) de maneira a ambas as composições poderem prosseguir viagem, como indicado na figura abaixo:



**NOTA:** Não é necessário a escrita de código na resolução das alíneas deste grupo.

**1a)** Para compreender melhor o problema comece por descrever todas as possibilidades de manobras a partir da configuração representada abaixo. Para o ajudar já apresentamos dois dos movimentos que podem ser efectuados; complete com os restantes.



**1b)** Formule claramente o problema para ser resolvido recorrendo a algoritmos de pesquisa em espaço de estados, indicando o estado inicial, teste de estado objectivo e função que devolve os sucessores de um estado.

**1c)** Proponha uma heurística não constante que lhe permita obter a solução óptima do problema com o algoritmo A\*. Indique o valor da heurística para cada um dos três estados do enunciado da alínea **1a**. Justifique porque pode utilizar essa heurística.

**1d)** Comente a seguinte afirmação “Somente um algoritmo de pesquisa informado permite resolver o problema das locomotivas.”

## Grupo 2 (Planeamento)

A agência de matrimónios Casa&Descasa decidiu investir num sistema de Inteligência Artificial para melhorar o atendimento aos seus clientes. O sistema mantém informação sobre os homens e mulheres registados no sistema, os casais (heterossexuais) e indivíduos solteiros. O objectivo do sistema é aconselhar casamentos e divórcios sugerindo ainda “falecimentos” de pessoas. A poligamia não é permitida.

As acções a considerar são: casar, divorciar, matar\_homem\_casado e matar\_mulher\_casada.

Por exemplo, o sistema deverá ser capaz de indicar os passos que levam o indivíduo Aníbal, homem e solteiro, a ficar casado com Belarmina que inicialmente está casada com César

**2a)** Modele o sistema anterior na linguagem STRIPS recorrendo **apenas** às proposições **homem(Ind)**, **mulher(Ind)**, **solteiro(Ind)** e **casados(IndHomem,IndMulher)**, em que **Ind**, **IndHomem** e **IndMulher** são constantes com o significado óbvio.

**Sugestão:** comece por implementar as acções casar e divorciar. Tenha particular atenção no tratamento das mortes de indivíduos casados nas acções matar\_homem\_casado e matar\_mulher\_casada: depois de um indivíduo falecer não pode voltar a casar, divorciar-se ou ser morto...

**2b)** Apresente um Plano com Ordem Parciais, e sequência(s) de acções, que resolvem a situação do Aníbal num número mínimo de passos. Sugere-se a utilização do algoritmo POP para construir este plano.

**2c)** Verifique se a sua modelação permite a geração de um número infinito de planos para o caso do Aníbal, Belarmina e César, justificando adequadamente.

**2d)** Explique sucintamente como é que a sua representação juntamente com o algoritmo POP impedem a geração de planos em que um indivíduo (e.g., homem) está casado simultaneamente com outros dois indivíduos (e.g., mulheres).

## Grupo 3 (Redes de Bayes)

Um agricultor decidiu contrair um empréstimo para comprar pesticidas para proteger as suas culturas de infestações por organismos nocivos. A probabilidade de existir uma infestação é de 0,7. O banco poderá recusar o empréstimo, ou dar parte ou a totalidade do montante pedido. À priori, a probabilidade do empréstimo ser recusado é de 0,2 enquanto que as restantes hipóteses são equiprováveis.

O agricultor pretende utilizar o dinheiro do empréstimo para comprar pesticidas. Se o dinheiro for emprestado na totalidade então ele compra os pesticidas; se o dinheiro lhe for emprestado em parte então ele adquire os produtos com 80% de probabilidade; caso o empréstimo seja recusado, o agricultor não poderá comprar os pesticidas. Caso não haja uma infestação, a probabilidade de se ter uma boa colheita é sempre de 0,9. Contudo, se surgir uma infestação e não se tenha comprado pesticidas, a probabilidade de não se ter uma boa colheita é 0,9. A probabilidade de se ter uma boa colheita é de 60% quando se dá uma infestação e se compraram pesticidas.

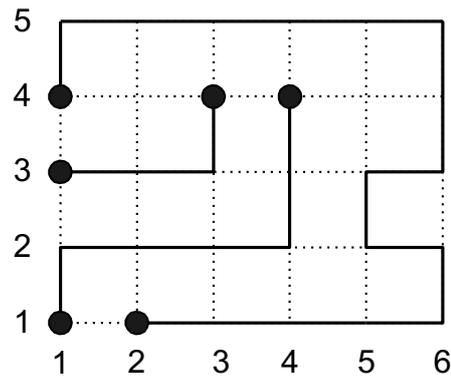
**3a)** Modele a situação anterior com uma rede de Bayes, indicando as variáveis aleatórias, seus domínios, topologia da rede e tabelas de probabilidade condicionada.

**3b)** Calcule a probabilidade de haver uma boa colheita sabendo que o agricultor obteve um empréstimo parcial e ocorreu uma infestação.

**3c)** Obtenha o valor de probabilidade do banco ter emprestado o dinheiro todo sabendo que houve uma infestação, comprou pesticidas e a colheita foi boa.

## Grupo 4 (Representação do Conhecimento)

A construção de circuitos integrados requer o desenho de diversas conexões eléctricas entre dois pontos distintos do circuito. Habitualmente utiliza-se uma grelha rectangular discreta  $N \times M$  para modelar o problema, ligando-se segmentos horizontais ou verticais entre dois pontos imediatamente adjacentes. Obviamente, as conexões não podem partilhar pontos. Na figura encontra-se uma solução para o caso de uma grelha  $6 \times 5$  e três conexões: entre o ponto  $(1,1)$  e o ponto  $(4,4)$ ; entre o ponto  $(2,1)$  e o ponto  $(1,4)$ ; entre o ponto  $(1,3)$  e  $(3,4)$ .



Iremos utilizar programação por conjuntos de resposta (answer set programming) para solucionar o problema, modelando-o tal que a cada modelo estável corresponda uma solução. Para começar, já se adicionaram ao programa alguns factos representando os dados do problema, nomeadamente:

- Factos do tipo **abcissa** ( $X$ ) para cada coordenada  $X$  possível da grelha, de 1 a  $N$ ;
- Factos do tipo **ordenada** ( $Y$ ) para cada coordenada  $Y$  possível da grelha, de 1 a  $M$ ;
- Factos do tipo **conexao** ( $C$ ) para cada conexão  $C$ , numeradas sequencialmente a partir de 1;
- Factos do tipo **inicio** ( $C, X, Y$ ) indicando que a conexão  $C$  se inicia no ponto com coordenadas  $(X, Y)$ ;
- Factos do tipo **fim** ( $C, X, Y$ ) indicando que a conexão  $C$  termina no ponto com coordenadas  $(X, Y)$ ;

Para o caso concreto apresentado na figura incluem-se os seguintes factos:

```
% FACTOS
abcissa(1). abcissa(2). abcissa(3). abcissa(4). abcissa(5). abcissa(6).
ordenada(1). ordenada(2). ordenada(3). ordenada(4). ordenada(5).

conexao(1).          conexao(2).          conexao(3).
inicio(1,1,1).       inicio(2,2,1).       inicio(3,1,3).
fim(1,4,4).          fim(2,1,4).          fim(3,3,4).
```

Também já existem no programa as seguintes regras:

```
% REGRAS

ponto(X,Y) :- abcissa(X), ordenada(Y).

segmento(X1,Y1,X2,Y2) :- ponto(X1,Y1), ponto(X2,Y2), adj(X1,Y1,X2,Y2).

adj(X,Y1,X,Y2) :- ponto(X,Y1), ponto(X,Y2), Y1 = Y2 + 1.
adj(X,Y1,X,Y2) :- ponto(X,Y1), ponto(X,Y2), Y2 = Y1 + 1.
adj(X1,Y,X2,Y) :- ponto(X1,Y), ponto(X2,Y), X1 = X2 + 1.
adj(X1,Y,X2,Y) :- ponto(X1,Y), ponto(X2,Y), X2 = X1 + 1.

pertence(P,X1,Y1,X2,Y2) :- conexao(P), segmento(X1,Y1,X2,Y2),
                           not nao_pertence(P,X1,Y1,X2,Y2).
nao_pertence(P,X1,Y1,X2,Y2) :- conexao(P), segmento(X1,Y1,X2,Y2),
                               not pertence(P,X1,Y1,X2,Y2).

atinge(P,X1,Y1) :- inicio(P,X1,Y1).
atinge(P,X2,Y2) :- conexao(P), atinge(P,X1,Y1),
                   segmento(X1,Y1,X2,Y2), pertence(P,X1,Y1,X2,Y2).
```

4a) Explique o papel das seguintes regras no programa,

```

pertence (P,X1,Y1,X2,Y2) :- conexao (P) , segmento (X1,Y1,X2,Y2) ,
                               not nao_pertence (P,X1,Y1,X2,Y2) .
nao_pertence (P,X1,Y1,X2,Y2) :- conexao (P) , segmento (X1,Y1,X2,Y2) ,
                               not pertence (P,X1,Y1,X2,Y2) .

```

O número de modelos estáveis obtidos para o programa contendo apenas os factos e as regras apresentados no enunciado é:

- a) 0      b) 1      c) 2      d) 80      e) mais de 1000

Indique qual das opções anteriores é a correcta, justificando sucintamente.

4b) Para cada regra ou conjunto de regras correspondentes a restrições de integridade, indique se deve(m), ou não, ser adicionada(s) ao programa acima. Justifique.

- a.  
 :- **conexao** (P1) , **conexao** (P2) , **ponto** (X,Y) ,  
    **atinge** (P1,X,Y) , **atinge** (P2,X,Y) , P1 = P2 .
- b.  
 :- **inicio** (P,X,Y) , **not aux** (P,X,Y) .  
**aux** (P,X,Y) :- **conexao** (P) , **segmento** (X,Y,X1,Y1) , **pertence** (P,X,Y,X1,Y1) .
- c.  
 :- **conexao** (P) , **segmento** (X1,Y1,X,Y) , **inicio** (P,X,Y) , **pertence** (P,X1,Y1,X,Y) .  
 :- **conexao** (P) , **segmento** (X,Y,X1,Y1) , **fim** (P,X,Y) , **pertence** (P,X,Y,X1,Y1) .

4c) Apresente uma restrição obrigando a que o ponto (X1,Y1) seja atingível a partir do ponto inicial da conexão P quando **pertence**(P,X1,Y1,X2,Y2) se verifica. Explique porque não é necessário introduzir uma restrição semelhante para o ponto (X2,Y2).

4d) Claramente falta uma restrição garantindo que um determinado ponto numa determinada conexão só pode estar ligado no máximo a um único ponto adjacente da mesma conexão. Modele esta restrição podendo para o efeito introduzir predicados auxiliares, caso entenda necessário.