

## Prova 2 - IA (08/11/2013)

### Questão 1 (10 pontos)

Diga em que situação de representação de domínios de especialidade e/ou de instâncias de problemas nesses domínios você aplicaria um algoritmo de busca heurística local ao invés de global.

### Questão 2 (20 pontos)

Descreva o domínio de problemas do Trabalho 2 da disciplina. Caracterize instâncias de problemas nesse domínio. Crie um exemplo de domínio e um exemplo de uma instância de problema. Escolha um algoritmo de busca heurística para resolver instâncias de problemas. Justifique a escolha por meio da descrição dos aspectos principais do algoritmo, enfatizando a sua face heurística.

### Questão 3 (40 pontos)

Considere o problema de transformação de estados constituído de um único robô carregador de bagagens em um hotel. É dada uma quantidade de malas a serem transportadas das portas dos quartos para o chão do saguão do hotel (lado mais à esquerda do hotel). O hóspede deve deixar sua mala na porta para o robô não ter que entrar no quarto. A disposição dos quartos e do saguão é aproximadamente a que aparece nas figuras dos estados Inicial e Final.

Estado inicial			Estado final		
		m	mm		
rm			r		

As restrições que se aplicam ao domínio são:

- Cada quarto só tem direito ao máximo de 1 (uma) mala a ser carregada pelo robô;
- O robô pode segurar no máximo 2 (duas) malas em um dado instante;
- Cada mala tem peso 1 (um);
- O robô tem peso 1 (um);
- O Estado Final tem sempre o robô parado no saguão com todas as malas no chão.

Note que, no Estado Inicial da figura acima, o robô está na frente de um quarto, segurando uma mala.

As operações atômicas possíveis para o robô são (em ordem de produção atômica):

- Andar para a esquerda, de uma porta até a porta adjacente ou até o saguão (*custo* = peso total, incluindo o peso do robô);
- Equivalente ao item anterior, para a direita;
- Pegar uma mala do chão se houver uma na porta e se o robô aguentar (*custo* = 1);
- Colocar 1 (uma) mala do chão, se o robô estiver carregando alguma e se o chão da porta estiver livre (*custo* = 0) - só o saguão pode receber mais de uma mala.

Com base no exposto acima, responda os três itens abaixo.

- Seja a função heurística  $h_1(n)$  = número de malas que não estão no chão do saguão. Essa heurística é admissível? (Justifique).
- Construa uma função heurística  $h_2(n)$  que domina  $h_1(n)$ . Tente fazer com que  $h_2(n)$  fique "bem melhor" que  $h_1(n)$  para o algoritmo  $A^*$ . Explique o origem numérica da função  $h_2(n)$ .
- Resolva a instância de problema da figura acima por meio do algoritmo  $A^*$  usando ambas as funções  $h_1(E)$  e  $h_2(E)$ , em duas árvores diferentes. Desenhe as árvores de busca geradas por ambas, de maneira que fique claro que a função  $h_2(E)$  resulta em uma quantidade menor de expansões de nodos. As árvores devem conter só os nodos criados na execução do algoritmo. Para cada nodo, indique os valores retornados pelas funções  $g(E)$ ,  $h_1(E)$  e  $h_2(E)$ .

### Questão 4 (30 pontos)

Imagine um predicado binário em Prolog, chamado *completada*, o qual realiza o trabalho de um algoritmo de busca heurística por *Satisfação de Restrições*. O predicado é capaz de preencher molduras de palavras cruzadas. Para ilustrar a aplicação do predicado *completada*, veja a execução da seguinte consulta:

```
?- completada([[ x1, x2, x3, *, x4 ],
               [ x5, *, x6, x7, x8 ],
               [ x9, *, x10, x11, x12 ],
               [ *, *, *, x13, * ]],
              [pal([x1,x2,x3]), pal([x6,x7,x8]), pal([x10,x11,x12]),
              pal([x1,x5,x9]), pal([x3,x6,x10]), pal([x7,x11,x13]),
              pal([x4,x8,x12]), dif([[x1,x2,x3], [x6,x7,x8], [x10,x11,x12],
              [x1,x5,x9], [x3,x6,x10], [x7,x11,x13], [x4,x8,x12]])]).
```

```
d o m * v
a * a t a
o * r i o
* * * a *
```

O primeiro termo de *completada* representa uma moldura retangular de  $N$  linhas por  $M$  colunas ( $4 \times 5$  para o exemplo) da palavra cruzada. As constantes do Prolog  $x_1, x_2, \dots, x_k$  ( $k = 13$  para o exemplo) deverão ser consideradas como variáveis da instância de problema a serem preenchidas pelo algoritmo de busca para formar palavras válidas do dicionário (ver abaixo). Onde há definição de variável no cruzamento entre uma linha e uma coluna, a letra a ser preenchida precisa coincidir para ambas as palavras (vertical e horizontal). Adicionalmente, as ocorrências dos asteriscos (\*) indicam que não haverá preenchimento de letra na posição.

O segundo termo de *completada* representa as restrições sobre os elementos da moldura. Elas podem ser de dois tipos:

- $pal(x_a, x_b, \dots, x_p)$  em que necessariamente a sequência de letras  $x_a, x_b, \dots, x_p$  deve formar uma palavra do dicionário;
- $dif([[x_a, x_b, \dots, x_p], \dots, [x_i, x_j, \dots, x_q]])$  em que necessariamente cada sequência de letras é diferente de qualquer outra sequência da lista.

Tais restrições *completam* as restrições básicas que derivam dos cruzamentos de palavras da moldura. Note ainda que o domínio de cada variável (valores possíveis) compreende apenas as letras minúsculas do alfabeto, ou seja, de a até z. Assuma que o dicionário de palavras é pequeno (com 200 palavras) e só contém itens de 3 (três) letras para facilitar o trabalho. Ele é composto apenas de fatos baseados no predicado *pal*. Um fragmento dele é:

```
pal([c,a,i]).
pal([c,a,l]).
pal([c,a,o]).
pal([c,e,m]).
pal([c,e,u]).
pal([c,h,a]).
```

Com base no exposto acima, responda os três itens abaixo.

- Aplicando-se logo de início a heurística da variável mais restritiva, por qual das 13 variáveis você começaria o preenchimento no exemplo acima? Justifique a escolha.
- Imagine a situação em que, ao invés do item anterior, a busca já avançou e a decisão é de aplicar a heurística da variável mais restrita. Tente descrever uma situação, mesmo que hipotética, para exemplificar isso.
- Como você começaria o preenchimento das variáveis da moldura: você usaria cada letra do alfabeto (a até z) para guiar o processo, ou usaria direto o dicionário de palavras para preencher de 3 em 3 variáveis? Justifique sua escolha com base em algum fator heurístico, tal como o do valor menos restritivo.

**BOM TRABALHO!**