



UNIVERSIDADE
DE ÉVORA

Inteligência Artificial

Trabalho 4

Miguel Grilo 58387

Jorge Couto 58656

Colégio Luís António Verney

Tabela de Conteúdos

| | |
|--------------------------|---|
| Tabela de Conteúdos..... | i |
| Exercício 1 | 1 |
| Condições | 1 |
| Fluentes..... | 1 |
| Ações | 1 |
| Exercício 2 | 1 |
| Exercício 3 | 2 |
| Exercício 4 | 3 |
| Exercício 5 | 3 |
| Alínea a)..... | 3 |
| Alínea b)..... | 4 |

Exercício 1

Condições

- $sobre(a, b)$: A posição 'a' está sobre a posição 'b';
- $esquerda(a, b)$: A posição 'a' está à esquerda da posição 'b'.

Fluentes

- $livre(a)$: A posição 'a' está livre;
- $maolivre(x)$: A mão do robot está livre;
- $na_mao(n)$: A peça 'n' está na mão do robot;
- $esta_em(a)$: A mão do robot está na posição 'a';
- $ocupa(n, a)$: A peça 'n' está na posição 'a'.

Ações

- $agarra(p, a)$: A mão do robot agarra na peça 'p' que está na posição 'a';
- $larga(p, a)$: A mão do robot larga a peça 'p' na posição 'a';
- $move_cima(e1, e2)$: A mão do robot move para cima, da posição 'e1' para a posição 'e2';
- $move_baixo(e1, e2)$: A mão do robot move para baixo, da posição 'e1' para a posição 'e2';
- $move_esquerda(e1, e2)$: A mão do robot move para a esquerda, da posição 'e1' para a posição 'e2';
- $move_direita(e1, e2)$: A mão do robot move para a direita, da posição 'e1' para a posição 'e2'.

Exercício 2

- **Ação:** $agarra(p, a)$

Pré-condições: $maolivre(x), ocupa(p, a), esta_em(a)$;

Efeitos: $na_mao(p), livre(a)$;

Perde: $maolivre(x), ocupa(p, a)$;

- **Ação:** $larga(p, a)$

Pré-condições: $livre(a), na_mao(p), esta_em(a)$;

Efeitos: $maolivre(x), ocupa(p, a)$;

Perde: $livre(a), na_mao(p)$;

- **Ação:** *move_cima(e1, e2)*
Pré-condições: *esta_em(e1), sobre(e2, e1);*
Efeitos: *esta_em(e2);*
Perde: *esta_em(e1);*
- **Ação:** *move_baixo(e1, e2)*
Pré-condições: *esta_em(e1), sobre(e1, e2);*
Efeitos: *esta_em(e2);*
Perde: *esta_em(e1);*
- **Ação:** *move_esquerda(e1, e2)*
Pré-condições: *esta_em(e1), esquerda(e2, e1);*
Efeitos: *esta_em(e2);*
Perde: *esta_em(e1);*
- **Ação:** *move_direita(e1, e2)*
Pré-condições: *esta_em(e1), esquerda(e1, e2);*
Efeitos: *esta_em(e2);*
Perde: *esta_em(e1).*

Exercício 3

- *estado_inicial([sobre(a, d), sobre(d, g), sobre(b, e), sobre(e, h), sobre(c, f), sobre(f, i),
esquerda(a, b), esquerda(b, c), esquerda(d, e), esquerda(e, f),
esquerda(g, h), esquerda(h, i), livre(e), maolivre(x), esta_em(a),
ocupa(1, a), ocupa(2, b), ocupa(3, c), ocupa(4, d), ocupa(5, f),
ocupa(6, g), ocupa(7, h), ocupa(8, i)])*.

Exercício 4

- *estado_final*([*sobre(a, d)*, *sobre(d, g)*, *sobre(b, e)*, *sobre(e, h)*, *sobre(c, f)*, *sobre(f, i)*,
esquerda(a, b), *esquerda(b, c)*, *esquerda(d, e)*, *esquerda(e, f)*,
esquerda(g, h), *esquerda(h, i)*, *livre(i)*, *maolivre(x)*, *esta_em(i)*,
ocupa(1, a), *ocupa(2, b)*, *ocupa(3, c)*, *ocupa(4, d)*, *ocupa(5, e)*,
ocupa(6, g), *ocupa(7, h)*, *ocupa(8, f)*]).

Exercício 5

Alínea a)

- Uma vez que o pop não conseguiu resolver o problema proposto devido a Stack Overflow, mudamos o nosso estado final para um com $M < 6$:
estado_final([*livre(f)*, *maolivre(x)*, *esta_em(f)*, *ocupa(1, a)*,
ocupa(2, b), *ocupa(3, c)*, *ocupa(4, d)*, *ocupa(5, e)*, *ocupa(6, g)*,
ocupa(7, h), *ocupa(8, i)*]).
- Com o nosso estado inicial igual, salvo pelo robô que começa na posição f do tabuleiro. Para isso, obtivemos a solução, através do pop:
 $P = [s1 - inicial, s3 - agarra(5, f),$
 $s245709 - move_{esquerda}(f, e),$
 $s222304 - larga(5, e),$
 $s246449 - move_{direita}(e, f), s2 - final]$
- **Número de Passos:** 6;
- **Conjunto de Passos:** Estado Inicial; Agarra peça '5' no espaço 'f'; Move para a esquerda de 'f' para 'e'; Larga a peça '5' no espaço 'e'; Move para a direita de 'e' para 'f'; Estado Final;
- **Número de Links:** 11;
- **Conjunto de Links:**
 - *link(s1, s3, livre(e));*
 - *link(s1, s3, maolivre(x));*
 - *link(s1, s3, esta_em(f));*
 - *link(s1, s3, ocupa(5, f));*
 - *link(s3, s245709, esta_em(f));*
 - *link(s245709, s222304, esta_em(e));*
 - *link(s222304, s246449, esta_em(e));*
 - *link(s222304, s2, livre(f));*
 - *link(s222304, s2, maolivre(x));*
 - *link(s246449, s2, esta_em(f));*
 - *link(s222304, s2, ocupa(5, e));*

- **Ordem entre os Passos:** Como resolvido pelo POP (começando pelo estado final e indo até ao estado inicial):
 $s2 \rightarrow s246449 \rightarrow s222304 \rightarrow s245709 \rightarrow s3 \rightarrow s1;$

Alínea b)

- As ameaças que encontramos são as seguintes:
s245709: Ameaça $link(s1, s3, esta_em(f))$, pois o efeito $esta_em(e)$ do passo implica $\neg esta_em(f)$, o que ameaça o link.
Resolução: Promoção de $s3$: $s3 < s245709$. Assim, o robot está em 'f' quando agarra a peça 5 e só depois se move para 'e'.

s246449: Ameaça $link(s245709, s222304, esta_em(e))$, pois o efeito $esta_em(f)$ do passo implica $\neg esta_em(e)$, o que ameaça o link.
Resolução: Despromoção de $s222304$. Assim, o robot só anda para 'f' depois de largar a peça em 'e'.