4º Trabalho Prático
Planeamento com o POP

Licenciatura em Eng. Informática

Inteligência Artificial

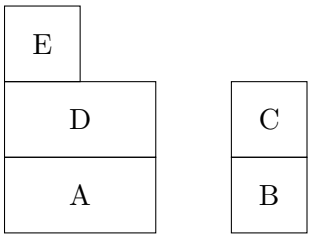
UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Helder Godinho 42741
Mariana Silva 54389

Docente: Irene Pimenta

Considere o problema: Têm um robot com dois braços (1 e 2), e quatro blocos: (A, B, C, D e E). Os blocos A e D têm o dobro da largura dos blocos B, C e E. Os blocos com o mesmo tamanho podem ser empilhados uns sobre os outros e os blocos pequenos podem ficar sobre os grandes. O robot pode agarrar num bloco pequeno com um só braço, mas para agarrar num bloco grande precisa dos dois braços.

Estado inicial



Estado final

1. Construa um vocabulário (condições e ações) para modelar este problema.

Condições:

- **sobre(X, Y):** Bloco X está sobre o bloco Y.
- **livre(X):** O bloco X está livre (nada está em cima dele).
- **no_chao(X):** Bloco X está no chão.
- **no_braco(X, B):** Bloco X está no braço B.
- **braco_livre(B):** Braço B está livre.
- **pequeno(X):** Bloco X é pequeno.
- **grande(X):** Bloco X é grande.

Ações:

- **agarrar_pequeno(X, B):** Agarra o bloco pequeno X com o braço B.
- **agarrar_grande(X):** Agarra o bloco grande X com ambos os braços.
- **peq_sobre_peq(X, Y):** Coloca o bloco pequeno X sobre o bloco pequeno Y.
- **peq_sobre_grande(X, Y):** Coloca o bloco pequeno X sobre o bloco grande Y.

- **grande_sobre_grande(X, Y):** Coloca o bloco grande X sobre o bloco grande Y.
- **por_no_chao_pequeno(X):** Coloca o bloco pequeno X no chão.
- **por_no_chao_grande(X):** Coloca o bloco grande X no chão.

2. Descreva este problema na notação STRIPS usando o vocabulário proposto.

```
accao(agarrar_pequeno(X, B),
  [pequeno(X), livre(X), braco_livre(B)],   %precond
  [no_braco(X, B)],                         %efeitos adicionados
  [livre(X), braco_livre(B)]                %efeitos removidos
).

accao(agarrar_grande(X),
  [grande(X), livre(X), braco_livre(1), braco_livre(2)],
  [no_braco(X, 1), no_braco(X, 2)],
  [livre(X), braco_livre(1), braco_livre(2)]
).

accao(peq_sobre_peq(X, Y),
  [pequeno(X), pequeno(Y), livre(Y), no_braco(X, B)],
  [sobre(X, Y), livre(X), braco_livre(B)],
  [no_braco(X, B), livre(Y)]
).

accao(peq_sobre_grande(X, Y),
  [pequeno(X), grande(Y), livre(Y), no_braco(X, B)],
  [sobre(X, Y), livre(X), braco_livre(B)],
  [no_braco(X, B), livre(Y)]
).

accao(grande_sobre_grande(X, Y),
  [grande(X), grande(Y), livre(Y), no_braco(X, 1), no_braco(X, 2)],
  [sobre(X, Y), livre(X), braco_livre(1), braco_livre(2)],
  [no_braco(X, 1), no_braco(X, 2), livre(Y)]
).

accao(por_no_chao_grande(X),
  [grande(X), no_braco(X, 1), no_braco(X, 2)],
  [no_chao(X), braco_livre(1), braco_livre(2), livre(X)],
  [no_braco(X, 1), no_braco(X, 2)]
).

accao(por_no_chao_pequeno(X),
  [pequeno(X), no_braco(X, B)],
  [no_chao(X), braco_livre(B), livre(X)],
  [no_braco(X, B)]
).
```

3. Represente o estado inicial deste problema com o seu vocabulário

```

estado_inicial([
    grande(a),
    grande(d),
    pequeno(e),
    pequeno(c),
    pequeno(b),
    no_chao(a),
    sobre(d, a),
    sobre(e, d),
```

```
    no_chao(b),
    sobre(c, b),
    livre(e),
    livre(c),
    braco_livre(1),
    braco_livre(2)
]).
```

4. Como é que o pop (planeador de ordem parcial) resolveria o problema de ir do estado inicial ao estado fnal.
a) Indique o conjunto de passos, de links e a ordem entre os passos.

1. agarrar_pequeno(c)
2. agarrar_pequeno(e)
3. por_no_chao_pequeno(c)
4. agarrar_pequeno(b)
5. peq_sobre_peq(b,c)
6. peq_sobre_peq(a,b)
7. agarrar_grande(d)
8. por_no_chao_grande(d)
9. agarrar_grande(a)
10. grande_sobre_grande(a,d)

Links Causais e Ameaças:

1. **agarrar_pequeno(c)**
   - **Links:**
     - link(s0, agarrar_pequeno(c), [pequeno(c), livre(c), braco_livre(1)], [no_braco(c, 1)])
   - **Ameaças:** Nenhuma
2. **agarrar_pequeno(e)**
   - **Links:**
     - link(s0, agarrar_pequeno(e), [pequeno(e), livre(e), braco_livre(2)], [no_braco(e, 2)])
   - **Ameaças:** Nenhuma
3. **por_no_chao_pequeno(c)**
   - **Links:**
     - link(agarrar_pequeno(c), por_no_chao_pequeno(c), [no_braco(c, 1)], [no_chao(c), livre(c), braco_livre(1)])
   - **Ameaças:** agarrar_pequeno(b) pode ameaçar livre(c) para por_no_chao_pequeno(c)
   - **Resolução:** Promoção de agarrar_pequeno(b) após por_no_chao_pequeno(c)

4. **agarrar_pequeno(b)**
   - **Links:**
     - link(por_no_chao_pequeno(c), agarrar_pequeno(b), [livre(c), braco_livre(1)], [no_braco(b, 1)])
   - **Ameaças:** peq_sobre_peq(b, c) pode ameaçar livre(c) para agarrar_pequeno(b)
   - **Resolução:** Promoção de peq_sobre_peq(b, c) após agarrar_pequeno(b)
5. **peq_sobre_peq(b, c)**
   - **Links:**
     - link(agarrar_pequeno(b), peq_sobre_peq(b, c), [no_braco(b, 1), livre(c)], [sobre(b, c), livre(b), braco_livre(1)])
   - **Ameaças:** Nenhuma
6. **peq_sobre_peq(a,b)**
   - **Links:**
     - link(agarrar_pequeno(e), peq_sobre_peq(a, b), [no_braco(e, 2), livre(b)], [sobre(a, b), livre(a), braco_livre(1)])
   - **Ameaças:** agarrar_pequeno(b) pode ameaçar livre(b) para peq_sobre_peq(a, b)
   - **Resolução:** Promoção de agarrar_pequeno(b) após peq_sobre_peq(a, b)
7. **agarrar_grande(d)**
   - **Links:**
     - link(peq_sobre_peq(a, b), agarrar_grande(d), [livre(d)], [no_braco(d, 1), no_braco(d, 2)])
   - **Ameaças:** Nenhuma
8. **por_no_chao_grande(d)**
   - **Links:**
     - link(agarrar_grande(d), por_no_chao_grande(d), [no_braco(d, 1), no_braco(d, 2)], [no_chao(d), livre(d), braco_livre(1), braco_livre(2)])
   - **Ameaças:** agarrar_grande(a) pode ameaçar livre(d) para por_no_chao_grande(d)
   - **Resolução:** Promoção de agarrar_grande(a) após por_no_chao_grande(d)
9. **agarrar_grande(a)**
   - **Links:**
     - link(por_no_chao_grande(d), agarrar_grande(a), [livre(d)], [no_braco(a, 1), no_braco(a, 2)])
   - **Ameaças:** por_no_chao_grande(d) pode ameaçar livre(d) para agarrar_grande(a)
   - **Resolução:** Promoção de por_no_chao_grande(d) antes de agarrar_grande(a)
10. **grande_sobre_grande(a, d)**

- **Links:**
  - ink(agarrar_grande(a), grande_sobre_grande(a, d), [no_braco(a, 1), no_braco(a, 2), livre(d)], [sobre(a, d), livre(a), braco_livre(1), braco_livre(2)])
- **Ameaças:** por_no_chao_grande(d) pode ameaçar livre(d) para grande_sobre_grande(a, d)
- **Resolução:** Promoção de por_no_chao_grande(d) antes de grande_sobre_grande(a, d)