

Alunos: Luiz Gabriel, Ailesson Nando, Larissa Favaris.

Turma: EC01

- 1) Na primeira questão, foi proposto uma linguagem de representação para instanciar os blocos considerando os fatores solicitados. Para rodar a solução, basta carregar o arquivo Prolog e consultar a regra ``consulta``. Isso irá inicializar o mundo dos blocos e tentar empilhar os blocos de acordo com as condições de vacância e estabilidade.

Exemplo de Saída:

```
?- consulta.  
Empilhado bloco b2 sobre b1  
Empilhado bloco b3 sobre b2  
Empilhado bloco b4 sobre b3  
true.
```

Para comparar com o mundo dos blocos do livro do Bratko, nesse código, o programa verifica as questões de vacância e estabilidade, diferentemente do código do livro, onde os blocos são de mesmo tamanho, fazendo apenas a verificação de se a posição está livre e se o bloco não está em baixo de outro. No código do trabalho, as questões que influenciam o problema, como estabilidade, centro de massa, devem ser consideradas no problema de empilhar os blocos.

- 2) Para modificar o código do planner para manipular corretamente variáveis sobre goals e ações, precisamos fazer as seguintes mudanças:
  - Adicionar suporte para variáveis em goals e ações:
  - Em `satisfied/2`, precisamos verificar se as variáveis em Goals são unificadas com as variáveis em State.
  - Em `achieves/2`, precisamos verificar se as variáveis em Action são unificadas com as variáveis em Goal.
  - Em `preserves/2`, precisamos verificar se as variáveis em Action não destroem as variáveis em Goals.
  - Em `regress/3`, precisamos verificar se as variáveis em NewRelations são unificadas com as variáveis em Goals.
  - Modificar a função `addnew/3` para lidar com variáveis:
  - Em `addnew/3`, precisamos verificar se as variáveis em NewGoals são compatíveis com as variáveis em OldGoals.

O código modificado é o código fornecido no arquivo  
"goal\_reg\_me\_planner.pl" fornecido no classroom.

- 3) Situação 1: `s_inicial=i1` ate o estado `s_final=i2`  
Para ir do estado inicial `i1` ao estado final `i2`, precisamos realizar as seguintes ações:
  - Mover o bloco d da posição 3 para a posição 2.
  - Mover o bloco a da posição 3 para a posição 0.
  - Mover o bloco c da mesa para a posição 0.

- Mover o bloco d da posição 2 para a posição 0 sobre o bloco c.
- Mover o bloco b da posição 5 para a posição 5 sobre o bloco d.
- Plano de ações: [move(d, 3, 2), move(a, 3, 0), move(c, mesa, 0), move(d, 2, 0), move(b, 5, 5)]

Situação 2: s\_inicial=i2 ate o estado s\_final=i2 (a)

Para ir do estado inicial i2 ao estado final i2 (a), precisamos realizar as seguintes ações:

- Mover o bloco d da posição 0 para a posição 3.
- Mover o bloco c da posição 0 para a posição 4 sobre o bloco a.
- Mover o bloco a da posição 0 para a posição 4 sobre o bloco d.
- Mover o bloco b da posição 5 para a posição 5 sobre o bloco d.
- Plano de ações: [move(d, 0, 3), move(c, 0, 4), move(a, 0, 4), move(b, 5, 5)]

Situação 3: s\_inicial=i2 ate o estado s\_final=i2 (b)

Para ir do estado inicial i2 ao estado final i2 (b), precisamos realizar as seguintes ações:

- Mover o bloco d da posição 0 para a posição 3.
- Mover o bloco c da posição 0 para a posição 4 sobre o bloco b.
- Mover o bloco a da posição 0 para a posição 4 sobre o bloco c.
- Mover o bloco b da posição 5 para a posição 5 sobre o bloco c.
- Plano de ações: [move(d, 0, 3), move(c, 0, 4), move(a, 0, 4), move(b, 5, 5)]

Situação 4: s\_inicial=i2 ate o estado s\_final=i2 (c)

Para ir do estado inicial i2 ao estado final i2 (c), precisamos realizar as seguintes ações:

- Mover o bloco c da posição 0 para a posição 0 sobre a mesa.
- Mover o bloco a da posição 0 para a posição 2.
- Mover o bloco d da posição 0 para a posição 0 sobre o bloco c.
- Mover o bloco b da posição 5 para a posição 5 sobre o bloco d.
- Plano de ações: [move(c, 0, 0), move(a, 0, 2), move(d, 0, 0), move(b, 5, 5)]

#### 4) A última questão:

A situação 2 possui uma complexidade maior para o planejador, pois é necessário um processo de "desempilhar" os blocos antes de poder movê-los para a posição final.

S0: {A em cima de B, B em cima de C, C no chão}

S1: {B em cima de C, C no chão} (ação: mover A para o chão)

S2: {A em cima de C, C no chão} (ação: mover B para o chão)

S3: {A em cima de B, B no chão} (ação: mover C para o chão)

S4: {A em cima de B, B no chão, C no chão} (ação: mover C para cima de B)

S5: {A em cima de B, B em cima de C, C no chão} (ação: mover C para cima de A)

Estado final: {A em cima de B, B em cima de C, C no chão}

A situação 3 possui uma complexidade menor, pois os blocos já estão em posições "desempilhadas".

S0: {A em cima de B, B em cima de C, C no chão}

S1: {B em cima de C, C no chão} (ação: mover A para o chão)

S2: {C no chão} (ação: mover B para o chão)

S3: {B em cima de C, C no chão} (ação: mover C para cima de B)

S4: {A em cima de B, B em cima de C, C no chão} (ação: mover A para cima de B)

Estado final: {A em cima de B, B em cima de C, C no chão}

Linguagem:

Para a linguagem do planejador, seria necessário definir ações como "mover(bloco, posição)", onde "bloco" é o bloco a ser movido e "posição" pode ser "chão" ou "outro bloco". O planejador precisaria verificar as condições prévias para realizar cada ação, por exemplo, para mover um bloco para cima de outro, o bloco destino deve estar no chão.

Padrões:

Desempilhar: O planejador deve reconhecer o padrão de desempilhar blocos, onde é necessário mover os blocos para o chão antes de movê-los para cima de outros blocos.

Empilhar: O planejador deve reconhecer o padrão de empilhar blocos, onde é necessário mover blocos para cima de outros blocos para alcançar o estado final desejado.

Análise:

Em geral, o planejador para a situação 2 precisa de mais passos para chegar ao estado final do que o planejador para a situação 3. Isso porque a situação 2 requer que os blocos sejam desempilhados antes de serem empilhados novamente na ordem correta.

Conclusão:

A linguagem do planejador deve ser capaz de representar ações como mover blocos e verificar condições prévias para essas ações. O planejador deve ser capaz de reconhecer padrões de desempilhar e empilhar blocos para gerar planos eficientes. A situação 2 requer um planejamento mais complexo do que a situação 3, devido à necessidade de desempilhar os blocos.

