Trabalho 1 - Mundo dos Blocos

Josival S.Monteiro júnior[1]

[1]Instituto de Computação – Universidade Federal do Amazonas (UFAM)
Caixa Postal 69080-900 – Manaus – AM – Brazil

josival.salvador@icomp.ufam.edu.br

Sumário

1. Introdução

Neste trabalho, será desenvolvido um programa para planejar uma variação do mundo dos blocos, utilizando blocos de diferentes tamanhos e levando em consideração a estabilidade das estruturas. O problema inclui planejamento de trajetórias e uso de sensores para percepção do ambiente.

2. Descrição do Problema

O mundo dos blocos descrito possui as seguintes características:

- Blocos de tamanhos variados.
- Posições de blocos restritas a coordenadas numéricas inteiras.
- Estabilidade das estruturas deve ser garantida.
- Não há rotação de blocos durante os movimentos.

Os blocos devem ser suportados adequadamente para garantir a estabilidade, e o robô deve movimentar os blocos conforme as regras:

- Movimento vertical (para cima).
- Movimento horizontal.
- Movimento vertical (para baixo).

2.1. Heurísticas Especializadas

Para melhorar o planejamento neste mundo, foram introduzidas heurísticas especializadas, como o uso de sensores para percepção (câmera ou sensor tátil). Exemplos de ações incluem:

- `look(Position, Object)`: ação que permite ao robô reconhecer o objeto na posição `Position`.
- Ações para aquisição de informações em ambientes parcialmente conhecidos.

3. Planejamento

3.1. Modificação do Planejador

O planejador de regressão de metas (Figura 17.6 do material) foi modificado para lidar corretamente com variáveis em metas e ações. As modificações incluem a manipulação adequada de variáveis não instanciadas, conforme a Seção 17.5 do material. Isso melhora a eficiência e evita a criação de alternativas irrelevantes durante o planejamento.

3.2. Plano de Ação

Abaixo está o plano de ação gerado manualmente para os cenários descritos:

1. Do estado inicial $s_{\text{inicial}} = i1$ para o estado final $s_{\text{final}} = i2$:
   - Ação 1: `mover(A, X, Y).`
   - Ação 2: `look(Position, Object).`
2. Do estado inicial $s_{\text{inicial}} = i2$ para o estado final $s_{\text{final}} = i2(a)$:
   - Ação 1: `mover(B, X, Z).`
3. Do estado inicial $s_{\text{inicial}} = i2$ para o estado final $s_{\text{final}} = i2(b)$:
   - Ação 1: `look(Position, Object).`
   - Ação 2: `mover(C, Z, W).`

4. Adaptação do Código

O código do planner da Figura 17.6 do material foi adaptado para manipular corretamente variáveis sobre metas e ações. Abaixo está um trecho exemplificado da adaptação do código em Prolog:

Listing 1. Adaptação do Código do Planner

```
% Adi  o de vari veis  n o instanciadas no planejamento
planner(State, Goal) :-
    can(Act, State, [Goal | RestGoals]),
    execute(Act, State, NewState),
    planner(NewState, RestGoals).

% Verifica  o de pr -condi  es com vari veis
can(mover(Bloco, De, Para), State, [limpar(Bloco), limpar(
   Para)]) :-
    holds(limpar(Bloco), State),
    holds(limpar(Para), State).
```

A adaptação do código inclui o uso de variáveis não instanciadas, conforme discutido na Seção 17.5, para melhorar a eficiência do planejador.

5. Tabelas

5.1. Desempenho, Ambiente, Atuadores e Sensores

Seguindo o exemplo da Tabela 2.4 do livro de Russell, a Tabela 1 descreve o desempenho, ambiente, atuadores e sensores no contexto do mundo dos blocos.

5.2. Estados e Ações do Agente

A Tabela 5.2 descreve detalhadamente os estados percebidos pelo agente, as ações que ele pode realizar, o estado inicial e final de cada cenário.

—c—c—c—c—

Estado Inicial	Estado Final	Ações	Percepções
Bloco A na posição (1,1)	Bloco A movido para (2,1)	`mover(A, 1, 2)`	`look(1, A)`
Bloco B na posição (3,2)	Bloco B movido para (3,3)	`mover(B, 3, 3)`	`look(3, B)`

6. Exemplos

6.1. Exemplo do Mundo de Blocos

Considerando o exemplo da seção 3.2 do material, o mundo dos blocos pode ser visto como um problema de grade onde os blocos se movem em células adjacentes. Os blocos devem ser organizados conforme as regras de estabilidade e apoiados em blocos abaixo.

6.2. Exemplo de Código

A seguir, um exemplo de código Prolog que implementa o movimento de blocos baseado nas regras de estabilidade:

Listing 2. Exemplo de Movimento de Blocos

```
% Definindo o movimento de blocos
mover(Bloco, De, Para) :-
    limpar(Bloco),
    limpar(Para),
    assert(em(Bloco, Para)),
    retract(em(Bloco, De)).

% Definindo pr -condi es para estabilidade
limpar(Bloco) :-
    not(em(OutroBloco, Bloco)).
```

7. Conclusão

Neste trabalho, foram abordados conceitos de planejamento e execução de ações em um mundo de blocos, com a adaptação de planejadores para lidar com variáveis não instanciadas e sensores. A eficiência do planejador foi aprimorada, permitindo uma abordagem mais flexível e adaptável ao mundo descrito.

8. Referências

- Russell, S., Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson.
- Bratko, I. (2001). *Prolog Programming for Artificial Intelligence*. 3rd ed. Addison-Wesley.

Desempenho	Ambiente	Atuadores	Sensores
Estabilidade das estruturas	Mundo de blocos	Braço robótico	Câmera, Sensor tátil
Precisão no posicionamento	Tamanhos variados	Movimentos de translação	Percepção de posições

Tabela 1. Desempenho, Ambiente, Atuadores e Sensores