Trabalho - Mundo dos Blocos

1. Formulação completa do problema do mundo dos blocos

A. Tabela PEAS (Performance, Environment, Actuators, Sensors)

Componente	Descrição
Performance	- Minimizar o número de movimentos para atingir o estado objetivo
	- Garantir a estabilidade das estruturas de blocos
	- Maximizar a eficiência do planejamento
Environment	- Grade 2D com posições discretas
	- Blocos de diferentes tamanhos
	- Restrições de estabilidade (blocos maiores não podem ficar sobre menores)
	- Gravidade (blocos devem estar apoiados)
Actuators	- Garra para mover blocos (pegar e soltar)
	- Movimento da garra (horizontal e vertical)
Sensors	- Detecção de posição e tamanho dos blocos
	- Sensor de estabilidade
	- Sensor de colisão

B. Descrição detalhada

1. Estados:
    - Representados por uma lista de relações on(Bloco, Suporte), onde Suporte pode ser outro bloco ou 'mesa'.
    - Exemplo: [on(a, mesa), on(b, a), on(c, mesa)]
2. Ações:
    - move (Bloco, De, Para): Move um bloco de uma posição para outra.
3. Estado Inicial:
    - Configuração inicial dos blocos na grade, definida para cada cenário.
4. Estado Final:
    - Configuração desejada dos blocos, definida para cada cenário.
5. Restrições:
    - Um bloco só pode ser movido se estiver livre (nenhum bloco sobre ele).
    - Um bloco só pode ser colocado sobre a mesa ou sobre outro bloco livre.

Blocos maiores não podem ser colocados sobre blocos menores.

6. Conceito de local possível: Um local possível é definido como uma posição na grade onde um bloco pode ser colocado de forma estável, considerando suas dimensões. Isso inclui:
    - A superfície da mesa, que pode acomodar qualquer bloco.

- O topo de outro bloco, desde que o bloco a ser colocado não seja maior que o bloco de suporte.

7. Conceito de espaço livre: Um espaço livre é definido como uma área na grade que não está ocupada por um bloco e está acima de uma superfície estável (mesa ou topo de outro bloco). O espaço livre deve ser grande o suficiente para acomodar o bloco que será movido.

Esta formulação permite desenvolver uma solução a mais geral possível, considerando blocos de diferentes tamanhos e restrições de estabilidade.

2. Adaptação do código do planner

O código a seguir é uma adaptação do planner da figura 17.6 do livro do Bratko, modificado para manipular corretamente variáveis sobre goals e ações conforme discussão na seção 17.5:

Segue em anexo no github.

Explicação das mudanças:

O predicado plano/3 foi modificado para usar solve/3, que agora retorna uma lista de ações necessárias para atingir uma meta específica.

solve/3 foi expandido para lidar com casos onde um bloco precisa ser movido primeiro antes de mover o bloco desejado.

Variáveis não instanciadas são tratadas naturalmente em Prolog. Por exemplo, em solve(Estado, on(Bloco, Destino), Acoes), Bloco e Destino podem ser variáveis não instanciadas que serão ligadas durante a execução.

Adicionamos verificações de estabilidade (estavel/2) para garantir que blocos maiores não sejam colocados sobre menores.

O predicado aplicar_acoes/3 foi adicionado para aplicar uma sequência de ações ao estado.

Estas mudanças permitem que o planner lide com metas e ações mais complexas, incluindo movimentos intermediários necessários para atingir um objetivo.

3. Geração manual dos planos de ações para a Situação 1
Para gerar os planos de ações para cada cenário da Situação 1, vamos usar o código Prolog fornecido:
1. s_inicial=i1 até o estado s_final=i2

```
?- resolver_cenario(i1, i2, Plano).
Plano = [move(d,mesa,c), move(a,mesa,b)].
```

2. s_inicial=i2 até o estado s_final=i2 (a)

```
?- resolver_cenario(i2, i2, Plano).
Plano = [].
```

(Nenhuma ação necessária, já está no estado final)

3. s_inicial=i2 até o estado s_final=i2 (b)

```
?- estado_inicial(i2, EstadoInicial),
   plano(EstadoInicial, [on(d,c), on(c,a), on(a,mesa), on(b,mesa)], Plano).
Plano = [move(a,b,mesa), move(c,d,a), move(d,mesa,c)].
```

4. s_inicial=i2 até o estado s_final=i2 (b) (repetido) (Mesmo resultado do item 3)

5. (i1) para o estado (i2)

```
?- resolver_cenario(i1, i2, Plano).
Plano = [move(d,mesa,c), move(a,mesa,b)].
```

Estes planos de ações demonstram como o agente move os blocos para atingir os estados finais desejados em cada cenário da Situação 1, considerando as restrições de estabilidade e as dimensões dos blocos.