

Roteiro de Laboratório - Planejamento

Marcelo S. Menegol

1 Introdução

Neste laboratório, iremos explorar um dos **planners** mais simples, o **STRIPS** [RN03]. Criado na década de 70, **STRIPS** se refere a um programa e também à linguagem utilizada por este programa para se definir problemas. Ele é um planejador de ordem total que faz uma busca regressiva no espaço de situações (como visto na aula sobre **planning**).

Nesta aula, vamos explorar como se descrever um problema em **STRIPS** para que o programa possa resolvê-lo e nos dar um plano factível para atingir nosso objetivo. Vamos utilizar uma implementação em **Python**, disponível em [STR], ou no moodle da disciplina.

No repositório alguns exemplos já estão disponíveis. O que cada descrição do problema precisa, basicamente, é:

- Descrição parcial do estado inicial;
- Descrição parcial do objetivo;
- Descrição das possíveis ações.

Após a descrição estar completa, é possível se executar o planner para se obter o plano. Em um terminal, vá até a pasta onde salvou o repositório e execute `python strips.py monkeys.txt` para testar um exemplo.

Para acompanhar a execução passo a passo do **planner**, encontre no arquivo `strips.py` a linha de código com `debug = False` (perto da linha 343), e substitua por `debug = True`. Execute então o exemplo novamente e veja os passos feitos pelo **planner** para se chegar no resultado.

2 Atividades

2.1 Robô aspirador

Vamos começar com um exemplo simples visto em sala de aula: o robô aspirador de pó. Crie um arquivo `vaccum.txt` (ou `aspirador.txt`). Nele, descreva a seguinte situação:

- Existem duas posições: P1 e P2. Ambas estão sujas. O robô começa na posição P1.
- Queremos que ambas as posições, ao final da execução, estejam limpas.
- O robô pode fazer duas coisas:
 - Se mover de uma posição para a outra;
 - Sugar a sujeira da posição na qual está;

2.2 Anomalia de Sussman

Nesta atividade, devemos reproduzir a Anomalia de Sussman [Ano]. Para tal, amos inicialmente representar um mundo com blocos e mesas:

- Existem 3 blocos. Bloco A, bloco B e bloco C;
- Existe uma mão ou garra, que pode estar segurando algo ou não;
- Inicialmente, C está em cima de A e B está livre;
- O objetivo final é que A esteja em cima de C e C em cima de B;

- As ações são:
 - Pegar um bloco da mesa;
 - Colocar o bloco na mesa;
 - Empilhar um bloco no outro;
 - Desempilhar um bloco do outro.

Teste, em modo debug, com os objetivos declarados em sequencia diferente. A em cima de C e C em cima de B, C em cima de B e A em cima de C. O que ocorre? Porque? Depois que a aplicação funcionar para este caso, altere as condições para testar a Anomalia de Sussman.

2.3 Galpões Autônomos

Existem 3 galpões que podem armazenar produtos, 3 pacotes distribuídos entre esses galpões e dois AGVs que podem carregar e descarregar os pacotes, além de se mover de um galpão para outro. Inicialmente, tem-se os pacotes 1 e 2 no galpão 1, e o pacote 3 no galpão 2. O AGV 1 está no galpão 1, o AGV 2 está no galpão 3. Gere um plano que transporte o pacote 1 para o galpão 3, o pacote 2 para o galpão 2, e o pacote 3 para o galpão 1.

Na sua solução, se o número de pacotes, AGV e/ou galpões mudarem, o que se altera no código STRIPS?

Referências

- [Ano] Anomalia de Sussman. https://en.wikipedia.org/wiki/Sussman_Anomaly/.
- [RN03] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.
- [STR] STRIPS em Python. <https://github.com/tansey/strips>.