Inteligência Artificial (1001336)

Prof. Dr. Murilo Naldi

Agente Bombeiro – Trabalho 1

Guilherme Locca Salomão (758569)
João Victor Mendes Freire (758943)
Luís Felipe Corrêa Ortolan (759375)

Modelagem de Estado e Ambiente

$$\text{Estado} = [\text{Bombeiro, Extintores, Incêndios}]$$

$$\text{Estado} = [\ [X, Y, \text{Cargas}], [[X_1, Y_1]...[X_n, Y_n]], [[X_1, Y_1]...[X_m, Y_m]]\ ]$$

O estado é composto por três listas:

- Bombeiro:
  - X, Y representam a posição atual do bombeiro.
  - Cargas representa a quantidade de cargas do extintor ele possui.
- Extintores:
  - Lista das coordenadas dos extintores, no formato $[X_i, Y_i]$, sendo n a quantidade de extintores.
- Incêndios:
  - Lista das coordenadas dos incêndios, no formato $[X_j, Y_j]$, sendo m a quantidade de incêndios.

Modelagem de Estado e Ambiente

Regra que verifica se a posição a ser acessada pertence ao prédio:

```
% Definindo tamanho do prédio 10x5
dentro_predio([X, Y|_]) :- X > 0, Y > 0, X < 11, Y < 6.
```

Regras que ocupam o Ambiente com objetos:

```
% Mapeando parede no ambiente
ocupado_com([3, 3], parede).

% Mapeando entulho no ambiente
ocupado_com([4, 2], entulho).

% Mapeando escada no ambiente
ocupado_com([5, 1], escada).
```

Se não existe uma regra `ocupado_com()` para dada coordenada, e ela não está presente na lista de incêndios nem na de extintores, então a posição é considerada vazia.

Transições de Estado

Movimento Horizontal

- Verifica se a posição a ser acessada pertence ao prédio.
- Verifica se existe entulho, incêndio ou parede na posição a ser acessada.
- Pode acessar posições que contém extintores ou escadas

```
% Movimento horizontal à esquerda
s([[X, Y|Carga], Extintores, Incendios],
 [[X1, Y|Carga], Extintores, Incendios]) :-
    X1 is X - 1,
    dentro_predio([X1, Y]),
    not(ocupado_com([X1, Y], parede)),
    not(ocupado_com([X1, Y], entulho)),
    not(pertence([X1, Y], Incendios)).
```

```
% Movimento horizontal à direita
s([[X, Y|Carga], Extintores, Incendios],
 [[X1, Y|Carga], Extintores, Incendios]) :-
    X1 is X + 1,
    dentro_predio([X1, Y]),
    not(ocupado_com([X1, Y], parede)),
    not(ocupado_com([X1, Y], entulho)),
    not(pertence([X1, Y], Incendios)).
```

Transições de Estado

Saltos

- Acontecem se o Bombeiro continua dentro do prédio, existe entulho na posição adjacente (X2), não existe nenhum objeto na posição destino (X1), adjacente ao entulho.

```
% Salto à esquerda
s([[X, Y|Carga], Extintores, Incendios],
[[X1, Y|Carga], Extintores, Incendios]) :-
    X1 is X - 2, X2 is X - 1, Y1 is Y - 1,
    dentro_predio([X1, Y]),
    ocupado_com([X2, Y], entulho),
    not(ocupado_com([X1, Y], _)),
    not(pertence([X1, Y], Incendios)),
    not(pertence([X1, Y], Extintores)),
    not(ocupado_com([X1, Y1], escada)).
```

```
% Salto à direita
s([[X, Y|Carga], Extintores, Incendios],
[[X1, Y|Carga], Extintores, Incendios]) :-
    X1 is X + 2, X2 is X + 1, Y1 is Y - 1,
    dentro_predio([X1, Y]),
    ocupado_com([X2, Y], entulho),
    not(ocupado_com([X1, Y], _)),
    not(pertence([X1, Y], Incendios)),
    not(pertence([X1, Y], Extintores)),
    not(ocupado_com([X1, Y1], escada)).
```

Transições de Estado

Movimento Vertical

- Acontece se o Bombeiro continua dentro do prédio, existe uma escada na posição (em [X, Y] caso vá subir e em [X, Y-1] caso vá descer). As regras foram divididas em movimento vertical para cima e para baixo. Apenas o Y do bombeiro é atualizado nessa transição (para Y1).

```
% Movimento vertical para cima
s([[X, Y|Carga], Extintores, Incendios],
 [[X, Y1|Carga], Extintores, Incendios]) :-
    Y1 is Y + 1,
    dentro_predio([X, Y1]),
    ocupado_com([X, Y], escada).
```

```
% Movimento vertical para baixo
s([[X, Y|Carga], Extintores, Incendios],
 [[X, Y1|Carga], Extintores, Incendios]) :-
    Y1 is Y - 1,
    dentro_predio([X, Y1]),
    ocupado_com([X, Y1], escada).
```

Transições de Estado

Pega Extintor

- Se a carga está vazia (= 0) e existe um extintor na posição atual do Bombeiro. Então é preciso remover aquele extintor pego da lista de extintores e alterar a carga para 2.

```
% Pega extintor
s([[X, Y, Carga|Cauda], Extintores,  Incendios],
  [[X, Y,Carga1|Cauda], Extintores1, Incendios]) :-
    Carga == 0,
    pertence([X, Y], Extintores),
    retirar_elemento([X, Y], Extintores, Extintores1),
    Carga1 is Carga + 2.
```

Transições de Estado

Apaga incêndio

- Acontece se o Bombeiro tem carga (> 0), e existe fogo na posição adjacente à do Bombeiro (X1). Se houver, é necessário remover àquela posição da lista de incêndios e decrementar a Carga atual.

```
% Apaga Incêndio à esquerda
s([[X, Y, Carga|Cauda], Extintores, Incendios],
 [[X, Y, Carga1|Cauda], Extintores, Incendios1]) :-
    Carga > 0, X1 is X - 1,
    pertence([X1, Y], Incendios),
    retirar_elemento([X1, Y], Incendios, Incendios1),
    Carga1 is Carga - 1.
```

```
% Apaga Incêndio à direita
s([[X, Y, Carga|Cauda], Extintores, Incendios],
 [[X, Y, Carga1|Cauda], Extintores, Incendios1]) :-
    Carga > 0, X1 is X + 1,
    pertence([X1, Y], Incendios),
    retirar_elemento([X1, Y], Incendios, Incendios1),
    Carga1 is Carga - 1.
```

Transições de Estado

Meta

- Por fim, definimos o estado `meta()`, que no problema é qualquer estado no qual a lista de incêndios seja vazia.

```
meta([_, _, []]).
```

Funções de Busca

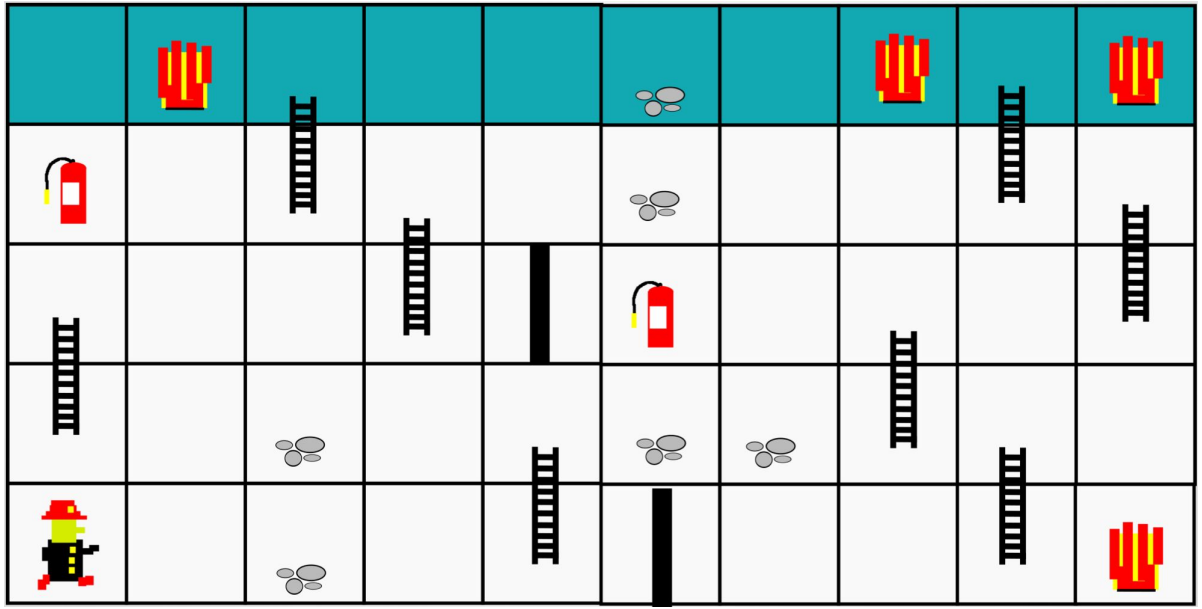
- Tanto a solução por Busca em Largura quanto a por Busca em Profundidade estão disponíveis.
- Adicionou-se uma função para deixar somente as informações do Bombeiro na resposta final, e em ordem cronológica.

```
solucao_bl(Inicial, SolucaoInv) :-
    bl([[Inicial]], Solucao),
    limpa_sol(Solucao, Solucao1),
    inverte(Solucao1, SolucaoInv).
```

```
solucao_bp(Inicial, SolucaoInv) :-
    bp([], Inicial, Solucao),
    limpa_sol(Solucao, Solucao1),
    inverte(Solucao1, SolucaoInv).
```

Exemplo de Solução

- Ambiente 3 (`amb3.pl`): caso dado pelo prof.



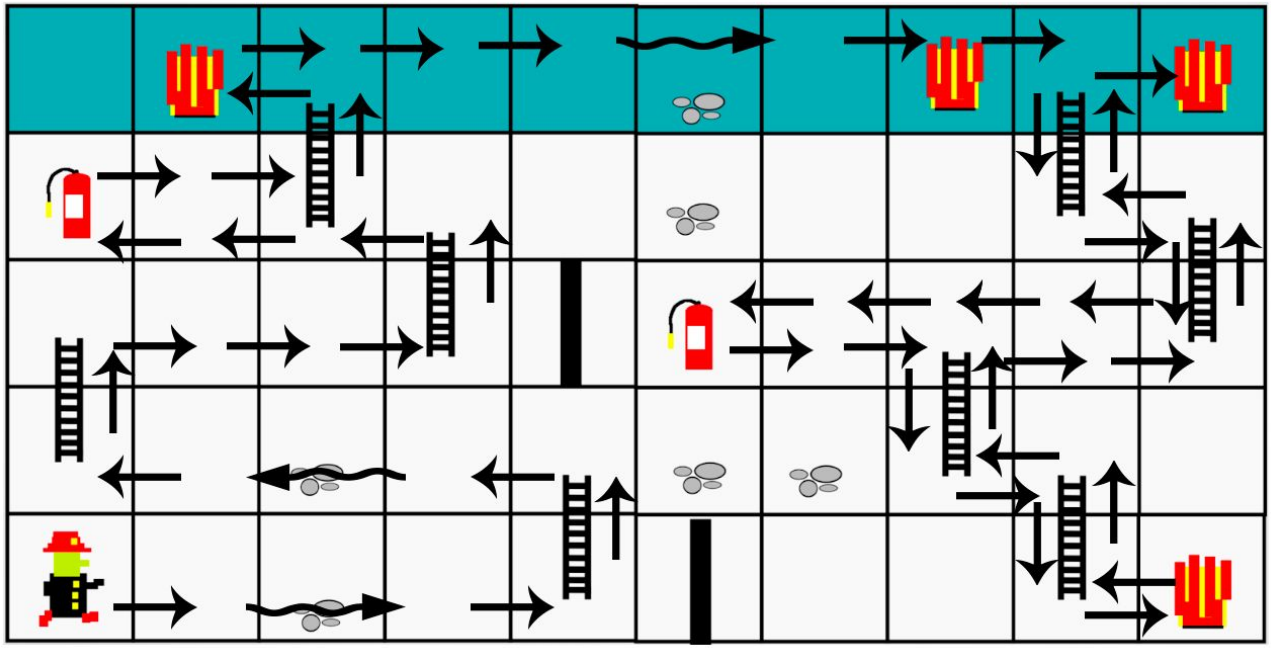
Exemplo de Solução

- Ambiente 3 (`amb3.pl`): caso dado pelo prof.

```
Entrada: solucao_bl([ [1, 1, 0], [ [6, 3], [1, 4] ],[ [10, 1], [2, 5], [8, 5], [10, 5] ] ], S).

Saída: S = [
[1, 1, 0], [2, 1, 0], [4, 1, 0], [5, 1, 0], [5, 2, 0], [4, 2, 0], [2, 2, 0], [1, 2, 0], [1, 3, 0],
[2, 3, 0], [3, 3, 0], [4, 3, 0], [4, 4, 0], [3, 4, 0], [2, 4, 0], [1, 4, 0], [1, 4, 2], [2, 4, 2],
[3, 4, 2], [3, 5, 2], [3, 5, 1], [4, 5, 1], [5, 5, 1], [7, 5, 1], [7, 5, 0], [8, 5, 0], [9, 5, 0],
[9, 4, 0], [10,4, 0], [10,3, 0], [9, 3, 0], [8, 3, 0], [7, 3, 0], [6, 3, 0], [6, 3, 2], [7, 3, 2],
[8, 3, 2], [8, 2, 2], [9, 2, 2], [9, 1, 2], [9, 1, 1], [9, 2, 1], [8, 2, 1], [8, 3, 1], [9, 3, 1],
[10,3, 1], [10,4, 1], [9, 4, 1], [9, 5, 1], [9, 5, 0]].
```

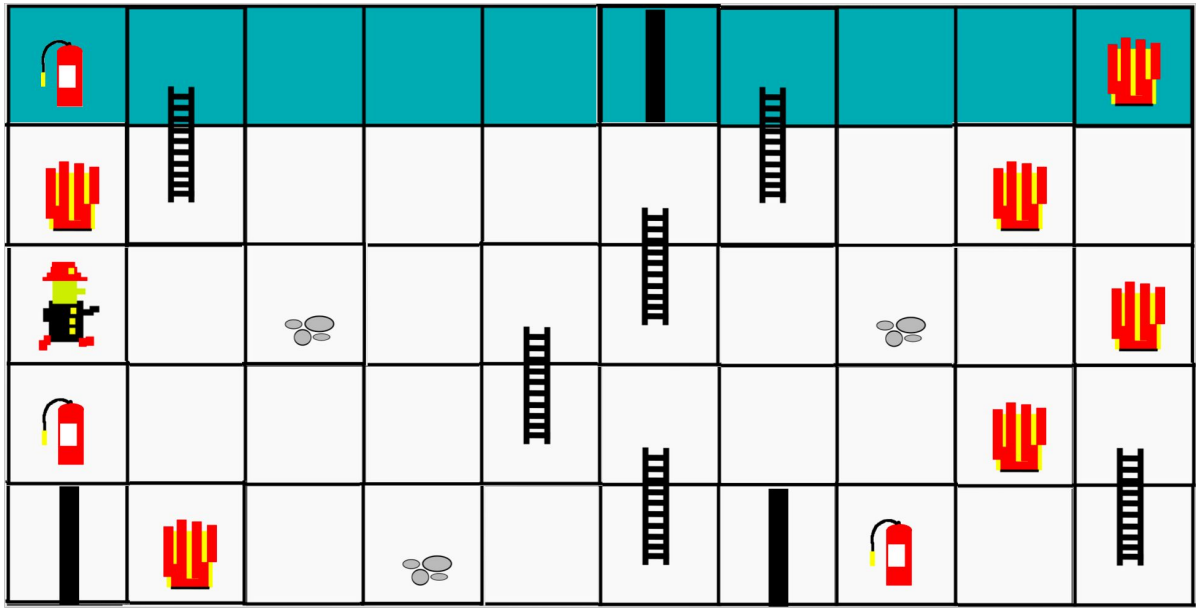
Exemplo de Solução

- Ambiente 3 (`amb3.pl`): caso dado pelo prof.



Exemplo de Solução

- Ambiente 6 (`amb6.pl`):



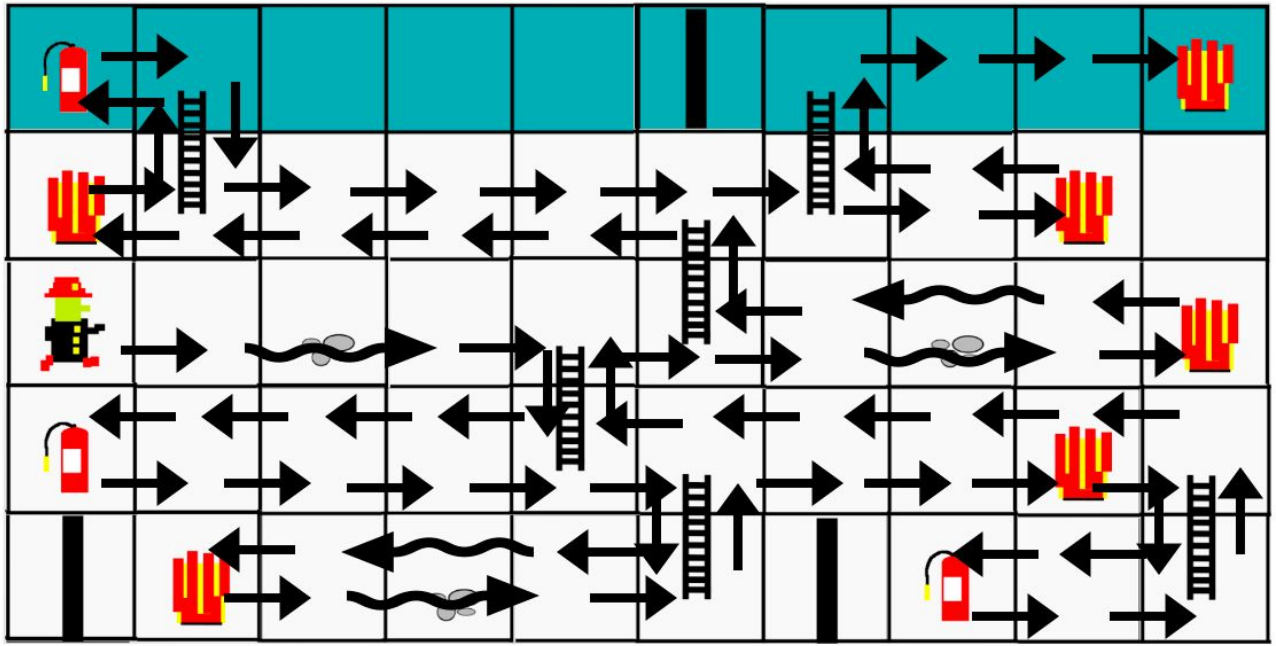
Exemplo de Solução

- Ambiente 6 (`amb6.pl`):

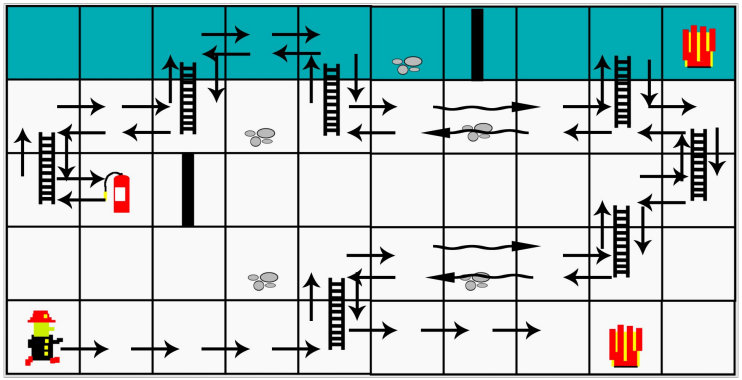
```
Entrada: solucao_bl([ [1, 3, 0], [ [8, 1], [1, 2], [1, 5] ], [ [2, 1], [9, 2], [10, 3], [1, 4],
[9, 4], [10, 5] ] ], S).

Saída: S = [
[1, 3, 0], [2, 3, 0], [4, 3, 0], [5, 3, 0], [5, 2, 0], [4, 2, 0], [3, 2, 0], [2, 2, 0], [1, 2, 0],
[1, 2, 2], [2, 2, 2], [3, 2, 2], [4, 2, 2], [5, 2, 2], [6, 2, 2], [6, 1, 2], [5, 1, 2], [3, 1, 2],
[3, 1, 1], [5, 1, 1], [6, 1, 1], [6, 2, 1], [7, 2, 1], [8, 2, 1], [8, 2, 0], [9, 2, 0], [10,2, 0],
[10,1, 0], [9, 1, 0], [8, 1, 0], [8, 1, 2], [9, 1, 2], [10,1, 2], [10,2, 2], [9, 2, 2], [8, 2, 2],
[7, 2, 2], [6, 2, 2], [5, 2, 2], [5, 3, 2], [6, 3, 2], [7, 3, 2], [9, 3, 2], [9, 3, 1], [7, 3, 1],
[6, 3, 1], [6, 4, 1], [5, 4, 1], [4, 4, 1], [3, 4, 1], [2, 4, 1], [2, 4, 0], [2, 5, 0], [1, 5, 0],
[1, 5, 2], [2, 5, 2], [2, 4, 2], [3, 4, 2], [4, 4, 2], [5, 4, 2], [6, 4, 2], [7, 4, 2], [8, 4, 2],
[8, 4, 1], [7, 4, 1], [7, 5, 1], [8, 5, 1], [9, 5, 1], [9, 5, 0]].
```

Exemplo de Solução

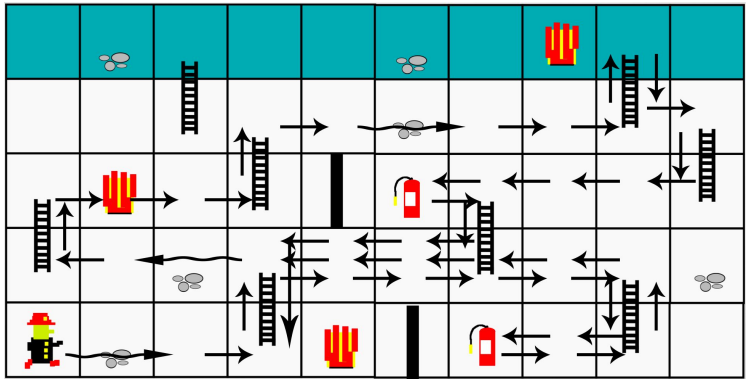
- Ambiente 6 (`amb6.pl`):



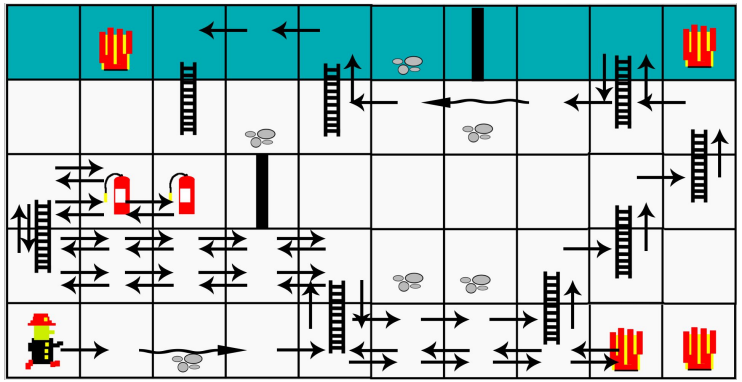
Exemplo de Solução



amb1.pl



amb4.pl



amb5.pl

Créditos

- As funções auxiliares de manipulação de listas em prolog foram retiradas dos slides de aula do Prof. Murilo Naldi, bem como as funções de Busca em Largura e em Profundidade (com pequenas alterações no retorno da solução final).

- As Imagens dos casos de teste 1 à 4 foram retiradas dos slides de aula do Prof. Murilo Naldi, partes dessas imagens foram usadas para montar os casos 5 à 8.