
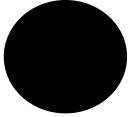


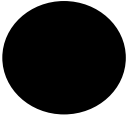


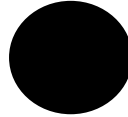


MAC425: EP2

Data de entrega: 4 de dezembro de 2006

4			<i>brisa</i>	
3		 <i>brisa</i>		<i>brisa</i>
2			<i>brisa</i>	
1		<i>brisa</i>		<i>brisa</i>
	1	2	3	4

Simulador do Mundo do Wumpus
em Prolog

Ambiente

- As alterações com relação ao livro AIMA, estão descritas em **vermelho** (para personalizar o EP :-):
 - quadrados adjacentes ao wumpus cheiram mal
 - quadrados adjacentes ao abismo possuem uma brisa
 - um quadrado brilha sss contém o ouro
 - o agente possui **três flechas**
 - atirar a flecha mata o wumpus se o agente estiver de frente para ele, isto é, na mesma linha ou coluna
 - pegar o ouro (**pode existir de 0 a 3 localizações com ouro**) só é possível se o agente estiver no mesmo quadrado
 - soltar deixa o ouro no mesmo quadrado em que o agente estiver

Ambiente

- grade 4x4 cercada de paredes
- o agente sempre inicia no quadrado (1,1) virado para a direita
- as localizações do ouro e do wumpus são aleatórias com uma distribuição uniforme
- a menos do quadrado de entrada, todos os outros podem conter um abismo, com probabilidade 0,2
- Para esse EP, você não precisa fazer a geração aleatória de ambientes mas saber que seu agente será testado em ambientes com essa configuração.

Percepção, ações e objetivo do agente

- **Percepção:** brisa, cheiro, brilho, escuta, choque (Note que o agente não percebe sua localização)
- **Ações:** escala, vira para direita (passando como parâmetro o ângulo) (**não é para implementar ação vira para esquerda**), vai para frente, segura, solta, atira
- **Objetivo:** encontrar o ouro e trazê-lo para o início (isto é, para a localização que é a entrada ou saída da caverna), o mais rápido possível (ou seja, com o menor número de passos) sem entrar num quadrado com abismo ou com wumpus
 - o agente morre se entra em um quadrado que possui um wumpus vivo ou um abismo

Pontuação do agente

- -1 para cada ação executada
- + 100 cada vez que o agente coletar o ouro
- - 100 para cada flecha lançada
- + 1000 por matar o Wumpus
- - 1000 se o agente morre

EP2: fase 1

- Implementar, em **PROLOG**, um simulador do Mundo do Wumpus que aceitará comandos do usuário (ou seja, um usuário humano irá assumir o papel do agente para o Mundo do Wumpus que discutimos em sala de aula).
- A forma de interação com o usuário pode ser **simples**, como por exemplo:
 - o usuário escreve o nome da ação (por exemplo: escala, vira para direita de 90°, vira para direita de 270°, vai para frente, segura, solta, atira) e o simulador atualiza o estado do mundo devolvendo as percepções do novo estado.

Simulador: Você está na posição [1,1] de frente para o leste e não percebe nada.

Jogador: va_para_frente.

Simulador: Você está na posição [2,1] de frente para o leste e percebe uma brisa.

Jogador: vire_180 ...

-
- (veja outros exemplos daqui a dois slides)

EP2: fase 1

- A forma de interação com o usuário pode ser mais **sofisticada**, como por exemplo (vide exemplos no próximo slide) através de uma interface gráfica em que:
 - o usuário pode visualizar as partes do mundo já exploradas
 - o usuário pode comandar o agente diretamente na caverna (nesse caso, é preciso restringir o tamanho da caverna)
 - etc..

Alguns exemplos de Simuladores do Mundo do Wumpus

- <http://www.cit.gu.edu.au/~anthony/dungeon/wumpus/> → versão muito interessante, baseada nas características originais do Mundo do Wumpus e representada no formato de um "livro jogo" de RPG.
- <http://scv.bu.edu/cgi-bin/wcl> → versão muito simples; interessante por se assemelhar à proposta original. Desenvolvida pela Universidade de Boston.
- <http://people.sunyit.edu/~schirot/Game/game.html> → um simulador cuja interação com o usuário é feita através de uma interface gráfica simplificada: as ações são selecionadas através de botões.
- <http://www.ime.usp.br/~leliane/LabVIA/laboratorio/laboratorio.html> → aqui você encontra um simulador no modo gráfico implementado em Java por um aluno de iniciação científica da Poli, com bolsa PIBIC e um aluno do BCC, com bolsa FAPESP, sob minha orientação. O usuário usa as teclas para movimentar o agente. Além disso, o usuário pode visualizar as percepções passadas e marcar com interrogação as posições de perigo (no estilo do mapa das minas).
- <http://www.cogsci.rpi.edu/Otter/Wumpus/> → aqui também foi feita uma interface gráfica de visualização do mundo e além disso, é possível ver os fatos inferidos pelo agente ... ??
- <http://cl3512.inf.tu-dresden.de:8180/TomcatFlux/wumpus/> → interface mais sofisticada que serve como motivação para uma implementação baseada em lógica (no caso, o cálculo de fluentes) mas que não está sendo pedida para esse EP. Parecida com a solução discutida em sala de aula usando o cálculo de situações e a linguagem Golog.

Ep2-fase1: algumas dicas

- Vocês vão precisar adotar uma especificação do Mundo do Wumpus em Prolog através de *fatós* Prolog como:

```
abismo([1,3]).  
wumpus([3,1]).  
agente([1,1],leste).  
ouro([2,3]).  
saida([1,1]).  
vivo_wumpus(sim).  
vivo_agente(sim).
```

...

- e através de cláusulas Prolog que especifiquem as propriedades do ambiente (ou as regras do jogo), como por exemplo:

```
tem_cheiro([X,Y], sim) :- adjacente([X,Y], [R,T]), wumpus([R,T]).  
tem_cheiro([X,Y], nao) :- adjacente([X,Y], [R,T]), not wumpus([R,T]).  
tem_brisa([X,Y], sim) :- adjacente([X,Y], [R,T]), abismo([R,T]).  
tem_brisa([X,Y], nao) :- adjacente([X,Y], [R,T]), not abismo([R,T]).
```

...

- e que finalmente permitam inferir a percepção do agente, p. ex., com a cláusula Prolog:

```
Percebe([X,Y], [CHEIRO, BRISA, BRILHO, CHOQUE, GRITO]) :-  
tem_cheiro([X,Y], CHEIRO), tem_brisa([X,Y], BRISA), tem_brilho([X,Y], BRILHO),  
tem_choque([X,Y], CHOQUE), tem_grito([X,Y], GRITO).
```

?

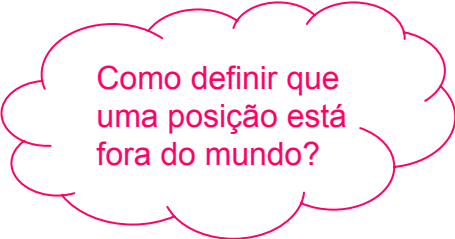
?

Ep2-fase1: algumas dicas

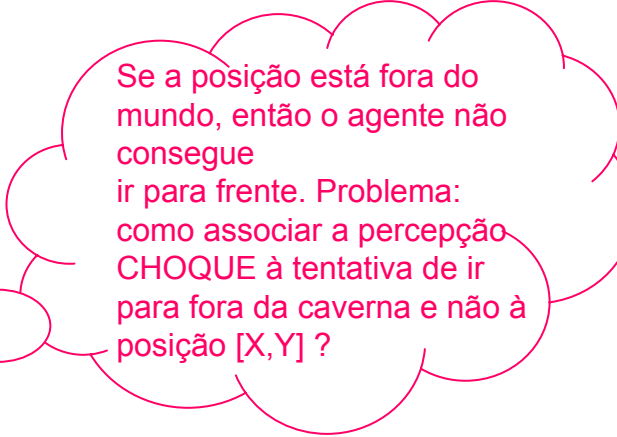
Além disso, vocês precisarão definir cláusulas Prolog que definem como as ações do agente serão executadas. Um exemplo bem simples (e incompleto) poderia ser:

```
execute(va-para-frente) :-  
    agente([X,Y], leste),  
    X1 is X+1, not fora_do_mundo(X1),  
    assert(agente([X1,Y], leste),  
    retract(agente([X,Y], leste).
```

```
execute(va-para-frente) :-  
    agente([X,Y], leste),  
    X1 is X+1, fora_do_mundo(X1),  
    tem_choque([X,Y], sim).
```



Como definir que uma posição está fora do mundo?



Se a posição está fora do mundo, então o agente não consegue ir para frente. Problema: como associar a percepção CHOQUE à tentativa de ir para fora da caverna e não à posição [X,Y] ?

Ep2-fase1: algumas dicas

- Lembrem-se que a posição do agente, por exemplo, no simulador deve ser atualizada em Prolog fazendo-se um *assert* ou *retract* do predicado *agente*.
- Em Prolog predicados desse tipo são chamados de *dynamic* e devem ser declarados da seguinte forma no início do programa:

```
:- dynamic([  
    p1/n1,  
    p2/n2,  
    ...  
]).
```

sendo p_i o nome do predicado (por exemplo, *agente*) e n_i a aridade do predicado (que no caso do predicado *agente* é 2).

Ep2-fase1: algumas dicas

- Além desse exemplos, é preciso implementar cláusulas que atualizem
 - a pontuação do agente
 - se o agente está vivo ou morto
 - se o wumpus está vivo ou morto
 - o número de flechas que o agente possui
 - etc.

EP2: fase 2

Implementar um agente reativo simples para o Mundo do Wumpus, em Prolog. Veja um exemplo de um agente reativo em:

<http://www.ime.usp.br/~leliane/LabVIA/laboratorio/Demonstracao/agentereativo/agentereativo.htm>

É importante notar que nesse exemplo, o Mundo do Wumpus é *diferente e mais complexo* do que especificamos para esse EP (com paredes somente ao redor). Nesse exemplo, não existem necessariamente paredes cercand todo o ambiente e existem paredes internas. Note que, se existir um caminho, o agente poderá ir da primeira coluna para a última coluna, e vice-e-versa. O mesmo para a primeira e última linha.

Um outro site interessante sobre a construção de agentes simples para o Mundo do Wumpus:

http://www.kr.tuwien.ac.at/students/prak_wumpusjava/simulator/Welcome.htm

EP2: fase 2 (bônus)

- Implementar em Prolog os axiomas do Cálculo de Situações para o Mundo do Wumpus.
- Objetivo: verificar hipóteses sobre o agente no Mundo do Wumpus. Veja um exemplo em:

<http://www.inf.furb.br/~jomi/logica/exercicios/wumpus/index.html>

EP2: fase 2 (bônus++)

- Implementar um agente inteligente para o Mundo do Wumpus que raciocine sobre situações passadas (no Cálculo de Situações) e que sempre selecione a melhor ação em cada situação.
- Você pode usar o Golog, se achar necessário.