

Terceiro Exercício-Programa (EP3)

IF 2019 - MAC115 - Entrega: até 26/11/2019 23:55 pelo PACA

Caça ao Ouro no Mundo do Wumpus

Problema

O Mundo do Wumpus é um jogo antigo de computador, criado por Gregory Yob em 1975, que fornece grande motivação para o raciocínio lógico num ambiente dinâmico, adverso e com percepção local. Apesar de parecer um jogo muito simples quando comparado aos jogos modernos de computador, o Mundo do Wumpus apresenta desafios interessantes de programação.

O jogo original consiste em uma caverna com salas conectadas por passagens sendo que em uma delas está o Wumpus: um monstro que devora qualquer um que entrar em sua sala. Para piorar a situação, algumas salas possuem poços profundos (abismos) que engolem qualquer um que entrar nelas. A única motivação para o agente permanecer nesse ambiente é a caça ao ouro. Ao percorrer as salas do mundo, o agente recebe dados sensoriais que o ajudará a compreender o que pode estar nas salas adjacentes, e assim poder evitar alguns destinos trágicos, como cair em um poço ou entrar na sala do Wumpus e ser devorado por ele. O agente possui uma flecha, que pode ser disparada uma única vez, preferencialmente quando ele tiver acumulado evidências suficientes de que existe um Wumpus na direção e orientação da flecha (o Wumpus é um monstro preguiçoso que nunca sai de sua sala). Após o Wumpus ser morto, o agente poderá entrar seguramente em sua sala (se conseguir aguentar seu fedor horrendo!). A tarefa do agente é planejar suas ações para encontrar o ouro, pegá-lo e sair da caverna. O interessante desse jogo é a possibilidade de simulação da percepção do agente e dos efeitos de suas várias ações.

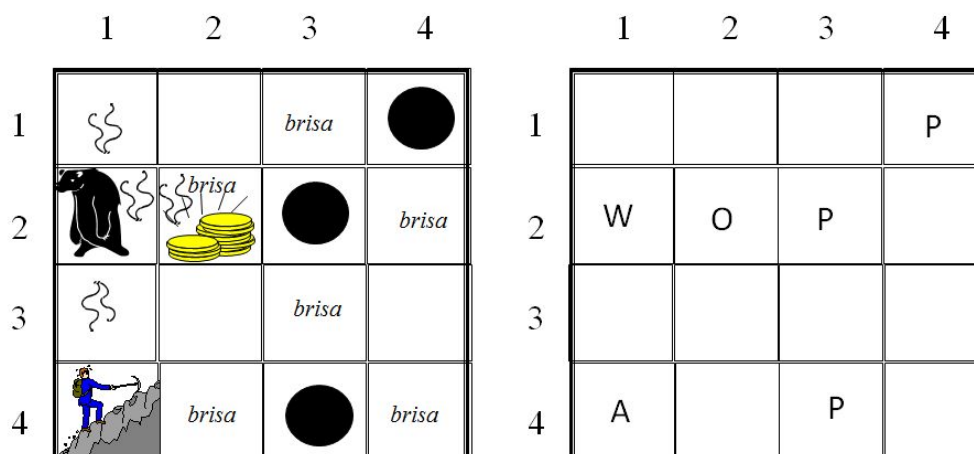


Figura 1. Ilustração de uma instância 4x4 do Mundo do Wumpus e a representação codificada da localização do Wumpus ("W"), dos poços ("P"), do ouro ("O") e do agente ("A").

Vamos considerar uma versão do Mundo do Wumpus (Fig. 1) em que o ambiente é uma grade NxN cercada por muros e que cada quadrado representa uma sala. A posição (N,1) é onde o agente entra e sai do mundo. Nessa versão, existem vários poços e apenas um Wumpus que permanece sempre em sua sala. Observe que o agente, ao pegar o ouro e colocá-lo em sua mochila, este deixa de brilhar ("R") nas salas em que o agente passar.

Ao chegar numa sala, o agente que está orientado para Norte, Sul, Leste ou Oeste (representado por "^", "v", ">", "<", respectivamente), recebe uma coleção de percepções a respeito dos arredores:

- se ele estiver em uma sala adjacente (não na diagonal) a um Wumpus ("W"), sentirá um fedor ("F");
- se ele estiver em uma sala adjacente (não na diagonal) a um poço ("P"), sentirá uma brisa ("B");
- se ele estiver na mesma sala onde o ouro ("O") está, o agente perceberá o reflexo do seu brilho ("R");
- se ele tentar ir para além das bordas do ambiente, o agente percebe um choque com o muro ("C"); e
- se ele conseguir disparar uma flecha certa (na mesma direção e sentido da localização do Wumpus), ouvirá um urro ("U") agonizante.

As ações do agente nesse mundo são restritas a um conjunto de 5 ações:

- mover ("M") para a sala imediatamente à sua frente (caso exista um muro o agente perceberá um choque e permanecerá na mesma localização);
- girar 90° para a sua direita ("D"), i.e., giro no sentido horário;
- girar 90° para a sua esquerda ("E"), i.e., giro no sentido anti-horário;
- atirar ("T") sua única flecha para frente (o Wumpus será morto e emitirá um urro caso a flecha conseguir alcançar a sala em que o Wumpus vive);
- pegar (*grab*) o ouro ("G") na sala em que ele estiver (caso o ouro não estiver na sala, nada acontecerá); e
- sair da caverna ("S"), somente se o agente estiver na sala inicial.

O agente ganha 100 pontos por escalar a caverna para fora carregando o ouro; 50 pontos se matar o Wumpus, -1 para cada ação que ele executar e -10000 pontos por ser morto pelo Wumpus, cair num poço ou sair da caverna sem carregar o ouro.

O programa na linguagem Python

Nesse EP3 você deverá implementar o jogo Mundo do Wumpus em que um usuário tomará as decisões pelo agente. A cada passo do jogo, o usuário/agente escolhe uma ação (M/T/D/E/G/S) e após sua execução o mundo é impresso na tela com as seguintes informações atualizadas:

- a localização do agente e sua orientação (">", "<", "^", "v");
- as percepções sentidas pelo agente logo após uma ação; e
- as percepções sentidas pelo agente desde o início do jogo (a menos do urro e do choque).

O seu programa deverá ler de um arquivo de entrada com a descrição de uma instância do mundo do Wumpus (obviamente desconhecida pelo usuário), representada por uma matriz como a ilustrada na Figura 1 (direita). Assumimos que a orientação inicial do agente é sempre voltada para o Norte ("^"). Com base na configuração de entrada, o programa deverá informar ao usuário as percepções do agente e os efeitos das ações (isto é, o seu programa será um simulador da dinâmica do Mundo do Wumpus, devolvendo as percepções do agente e o novo estado do mundo após cada ação executada). O programa termina se o agente for morto ou se ele sair da caverna. Nesse caso, o mundo completo é impresso na tela, bem como a pontuação total acumulada pelo agente.

Exemplo de arquivo de entrada (entrada.txt)

As instâncias do Mundo do Wumpus são matrizes quadradas $N \times N$, com $N < 9$. Assim, a 1a. linha do arquivo entrada.txt contém o valor de N ; as demais linhas do arquivo indicam as localizações dos demais itens. Adotaremos os inteiros 1, 2 e 3, para indicar os itens **poço**, **Wumpus** ou **ouro**, respectivamente:

```
N
LINHA COLUNA ITEM
LINHA COLUNA ITEM
LINHA COLUNA ITEM
...
```

Por exemplo, o Mundo do Wumpus da Figura 1, será codificado no arquivo de entrada da seguinte maneira:

```
4
2 1 2
1 4 1
2 2 3
2 3 1
4 3 1
```

Note que a localização inicial do agente não é indicada no arquivo de entrada, uma vez que sua posição inicial é padronizada como sendo a posição (N,1) com orientação Norte.

Exemplo de execução

Considere a instância do Mundo do Wumpus mostrada na Fig. 1. Inicialmente seu programa deve mostrar a seguinte tela para o usuário/agente (que desconhece o verdadeiro mundo):

```
Percepção após a última ação:
[]
Mundo do Wumpus conhecido pelo agente:

-----
|  ?  |  ?  |  ?  |  ?  |
-----
|  ?  |  ?  |  ?  |  ?  |
-----
|  ?  |  ?  |  ?  |  ?  |
-----
|  ^  |  ?  |  ?  |  ?  |
-----
Digite a ação desejada (M/T/D/E/G/S): M
```

Essa tela indica a localização inicial do agente, sala (4, 1), com orientação para o Norte ("^"). As casas contendo "?" são casas que ainda não foram visitadas pelo agente. Como não há nenhuma percepção na sala (4,1) o usuário/agente pode concluir que as duas salas adjacentes, (3,1) e (4,2), estão livres de ameaças. Certifique-se de compreender os índices usados para representar as salas (Figura 1).

Ao digitar “M” para comandar uma ação de “mover”, chegaremos à seguinte situação:

Percepção após a última ação:

[F]

Mundo conhecido pelo agente:

```
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
| ^ F  |  ?  |  ?  |  ?  |  
-----  
|      |  ?  |  ?  |  ?  |  
-----
```

Digite a ação desejada (M/T/D/E/G/S):

A tela mostra que a nova localização do agente é a sala (3, 1), com orientação Norte ("^") e que o agente sentiu uma percepção de fedor ("F"), indicada tanto na 2a linha da tela como na matriz. Com essa percepção podemos inferir que o Wumpus está em uma das salas adjacentes à sala (3,1), isto é, salas (2,1), (4,1) e (3,2). Como a sala (4,1) já foi visitada e estava segura, podemos concluir que o Wumpus está em uma das duas salas (2,1) e (3,2). Assim, a melhor escolha que um usuário prudente deve fazer é voltar para a localização inicial e se mover para a direita. Mas antes ele deve executar a seguinte sequência de ações:

D, D, M, E, M,

que resultam na seguinte tela:

Percepção após a última ação:

[B]

Mundo conhecido pelo agente:

```
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
| F    |  ?  |  ?  |  ?  |  
-----  
|      | > B  |  ?  |  ?  |  
-----
```

Digite a ação desejada (M/T/D/E/G/S): E

que indica que o agente está agora na sala (4,2), voltado para o leste (">"), e percebe uma brisa ("B"). Como um fedor não foi percebido na sala (4,2) (e uma brisa não havia sido percebida na sala (4,1)), então podemos concluir que a sala (3,2) é segura e é para lá que o agente deve se mover executando as ações E e M, que resultam na seguinte tela:

Percepção após a última ação:

[]

Mundo conhecido pelo agente:

```
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
|  ?  |  ?  |  ?  |  ?  |  
-----  
| F    | ^    |  ?  |  ?  |  
-----  
|      | B    |  ?  |  ?  |  
-----
```

Digite a ação desejada (M/T/D/E/G/S): M

Novamente, como a sala (3,2) não possui nenhuma percepção de brisa ou fedor, sabemos agora que o Wumpus está na sala (2,1) e que não há um poço na sala (2,2) que é a sala para onde o agente se move em seguida.

Percepção após a última ação:

[FBR]

Mundo conhecido pelo agente:

	?		?		?

	?		^FBR		?

	F				?

			B		?

Digite a ação desejada (M/T/D/E/G/S): G

Na sala (2,2) o agente percebe um fedor, uma brisa e o reflexo do ouro ("FBR"). O agente então decide pegar o ouro, ação G.

Percepção após a última ação:

[FBR]

Mundo conhecido pelo agente:

	?		?		?

	?		^FBR		?

	F				?

			B		?

Digite a ação desejada (M/T/D/E/G/S):

Em seguida, o agente pode decidir matar o Wumpus e sair da caverna, executando as ação E e T, o que resulta em matar o Wumpus e escutar o urro vindo de sua localização ("U").

Percepção após a última ação:

[FBRU]

Mundo conhecido pelo agente:

	?		?		?

	?		<FBR		?

	F				?

			B		?

Digite a ação desejada (M/T/D/E/G/S):

(Note que a percepção do Urro não é indicada na matriz mas somente na 2a linha da tela.) Finalmente, o agente pode decidir sair da caverna após a seguinte sequência de ações:

M, E, M, M, S,

que resulta na tela de fim de jogo, revelando a configuração completa do mundo, com as localizações de poços, do Wumpus e do ouro, bem como a pontuação acumulada pelo agente:

Mundo completo:

```
-----
|       |       |       |   P   |
-----
|   W   |   O   |   P   |       |
-----
|       |       |       |       |
-----
|       |       |   P   |       |
-----
Fim de jogo! Pontuação final: 133
```

(Note que o agente não é indicado na tela uma vez que ele terminou executando a ação de sair do mundo.)

Exemplo da percepção de Choque

Vamos ver um exemplo em que no início do jogo o agente executa as ações E e M que fazem com que o agente sinta um choque com o muro à esquerda da posição inicial.

Percepção após a última ação:

[C]

Mundo do Wumpus conhecido pelo agente:

```
-----
|   ?   |   ?   |   ?   |   ?   |
-----
|   ?   |   ?   |   ?   |   ?   |
-----
|   ?   |   ?   |   ?   |   ?   |
-----
|   <   |   ?   |   ?   |   ?   |
-----
```

Digite a ação desejada (M/T/D/E/G/S):

Note que, nesse caso, o agente sentiu um choque que é impresso na 2a linha da tela porém, o choque não é indicado na matriz (o mesmo acontece com a percepção do urro do Wumpus, se ele for morto).

Estruturas de Dados e Funções Python

Você vai precisar criar as seguintes matrizes e listas:

- **Mundo:** matriz de inteiros de dimensão NxN, que guarda as informações do mundo de entrada. Em cada casa **Mundo[i][j]**, existe um inteiro 0, 1, 2 ou 3, indicando se a casa está livre, contém um poço, contém o Wumpus, ou contém o ouro, respectivamente (assumimos que somente um desses itens pode ocorrer em uma casa);
- **Percebe:** matriz de inteiros que guarda as percepções do mundo recebidas pelo agente. Em cada casa **Percebe[i][j]**, tem um inteiro formado por "0"s e "1"s codificando as percepções: fedor, brisa e reflexo do ouro, isto é, FBR, nessa ordem. Por exemplo, o inteiro 100 significa que naquela casa o agente sentiu apenas fedor; o inteiro 110 significa que naquela casa o agente sentiu fedor e brisa; o inteiro 111 significa que naquela casa o agente sentiu fedor, brisa e o reflexo do ouro; o inteiro 000

indica que o agente não recebe nenhuma percepção naquela casa; etc.. Se a casa ainda não foi visitada, ela deve conter o valor -1.

- **Agente:** lista de tamanho 3 que guarda as informações sobre a localização e orientação do agente:
 - **Agente[0]:** indica a linha da matriz Mundo em que o agente se encontra,
 - **Agente[1]:** indica a coluna da matriz Mundo em que o agente se encontra, e
 - **Agente[2]:** indica a orientação atual do agente se encontra.

O valor inicial da lista **Agente** é [N, 1, "^"], que indica que o agente se localiza na linha N, coluna 1 e está orientado para o Norte ("^").

- **Estado:** lista de tamanho 4 que guarda as seguintes informações:
 - **Estado[0]** indica se o monstro está vivo (valor 1) ou morto (valor 0),
 - **Estado[1]** indica se existe a flecha (valor 1) ou não (valor 0),
 - **Estado[2]** indica se existe ouro para ser pego pelo agente (valor 1) ou não (valor 0) , e
 - **Estado[3]** indica os pontos que o usuário acumulou até o momento.

O valor inicial da lista **Estado** é [1, 1, 1, 0], que indica que o Wumpus está vivo, a flecha ainda não foi atirada, o ouro ainda não foi pego e a pontuação do usuário é 0.

Você deverá implementar as seguintes funções em Python:

le_mundo(Mundo): Essa função lê do arquivo de entrada ("entrada.txt") com as descrições do Mundo do Wumpus de dimensões NxN e guarda na matriz **Mundo**.

imprime_percepcao(Percebe, Agente): Essa função imprime, a cada jogada, o mundo contendo somente com as percepções recebidas pelo agente até a jogada atual, bem como a localização do agente e sua orientação através dos símbolos ">", "<", "^" e "v" ; nas posições ainda não visitadas pelo agente, a função imprime "?", como mostramos no exemplo abaixo (Note que nessa tela não mostramos os itens, W, P ou O para o usuário).

```
-----
|  ?  |  ?  |  ?  |  ?  |
-----
|  ?  | <FBR |  ?  |  ?  |
-----
|  F  |      |  ?  |  ?  |
-----
|      |  B  |  ?  |  ?  |
-----
```

imprime_mundo(Mundo): Essa função imprime o Mundo lido, mas somente no final do jogo, como no exemplo abaixo.

```
-----
|      |      |      |  P  |
-----
|  W  |  O  |  P  |      |
-----
|      |      |      |      |
-----
|      |      |  P  |      |
-----
```

atualiza_percepcaoEagente(Percebe, Mundo, Acao, Agente, Estado): Essa função recebe a matriz atual de percepções **Percebe**, a matriz **Mundo**, a **Acao** selecionada pelo usuário, a lista **Agente** e a lista **Estado** e faz todas as atualizações necessárias com relação aos efeitos da execução de **Acao** na matriz **Percebe** e nas listas **Agente** e **Estado**.

***Dica: para cada tipo de percepção testada e NÃO percebida você pode imprimir um espaço em branco " ". Isso pode ajudar para manter a formatação da tela.

Lembrete:

- Ler as Instruções para Entrega de EPs no PACA;
- Não postar códigos no Fórum Geral, nem os compartilhar com colegas;
- Não deixar o EP para a última hora;
- Divertir-se programando!