

Início	Terça, 21 Novembro 2017, 14:06
Estado	Teste enviado
Data de submissão:	Terça, 21 Novembro 2017, 16:06
Tempo gasto	1 hora 59 minutos
Nota	11,20 de um máximo de 20,00 (56%)

Informação

Considere uma base de factos para guardar informação sobre utilizadores e jogos da plataforma de jogos Vapor.

O predicado *player/3* contém informação dos jogadores registados na plataforma Vapor.

```
%player(Name, UserName, Age).  
player('Danny', 'Best Player Ever', 27).  
player('Annie', 'Worst Player Ever', 24).  
player('Harry', 'A-Star Player', 26).  
player('Manny', 'The Player', 14).  
player('Jonny', 'A Player', 16).
```

O predicado *game/3* contém informação sobre os jogos disponíveis na plataforma Vapor.

```
%game(Name, Categories, MinAge).  
game('5 ATG', [action, adventure, open-world, multiplayer], 18).  
game('Carrier Shift: Game Over', [action, fps, multiplayer, shooter], 16  
).  
game('Duas Botas', [action, free, strategy, moba], 12).
```

O predicado *played/4* contém informação sobre os jogos jogados por cada jogador (existe no máximo um facto por cada par jogador/jogo), nomeadamente número de horas jogadas e percentagem do jogo concluída.

```
%played(Player, Game, HoursPlayed, PercentUnlocked)  
played('Best Player Ever', '5 ATG', 3, 83).  
played('Worst Player Ever', '5 ATG', 52, 9).  
played('The Player', 'Carrier Shift: Game Over', 44, 22).  
played('A Player', 'Carrier Shift: Game Over', 48, 24).  
played('A-Star Player', 'Duas Botas', 37, 16).  
played('Best Player Ever', 'Duas Botas', 33, 22).
```

Responda às perguntas 1 a 6 **SEM** utilizar predicados de obtenção de soluções múltiplas (findall, setof e bagof), e **SEM** usar qualquer biblioteca do SICStus.

Pergunta 1

Respondida

Pontuou 1,000 de 1,000

Implemente o predicado ***achievedLittle(+Player)*** que sucede se o jogador *Player* completou menos de 10% de algum jogo.

Exemplos:

```
I ?- achievedLittle('Best Player Ever').  
no
```

```
I ?- achievedLittle('Worst Player Ever').  
yes
```

```
achievedLittle(Player) :-  
    played(Player, Game, _, PercentUnlocked),  
    PercentUnlocked < 10.
```

Comentário:

Pergunta 2

Respondida

Pontuou 1,000 de 1,000

Implemente o predicado ***isAgeAppropriate(+Name, +Game)*** que sucede se *Game* é um jogo adequado à idade da pessoa *Name*.

Exemplos:

```
I ?- isAgeAppropriate('Danny', '5 ATG').  
yes
```

```
I ?- isAgeAppropriate('Manny', '5 ATG').  
no
```

```
isAgeAppropriate(Name, Game) :-  
    player(Name, _, Age),  
    game(Game, _, MinAge),  
    Age >= MinAge.
```

Comentário:

Pergunta 3

Respondida

Pontuou 0,100 de 1,500

Implemente o predicado ***timePlayingGames(+Player, +Games, -ListOfTimes, -SumTimes)*** que determina o número total de horas que o jogador *Player* investiu a jogar cada um dos jogos indicados na lista *Games*, retornando em *ListOfTimes* os tempos de cada um dos jogos na ordem respetiva, e em *SumTimes* o número total de horas.

Exemplos:

```
| ?- timePlayingGames('Best Player Ever', ['5 ATG', 'Duas Botas'], LT, S
T).
LT = [3, 33],
ST = 36 ? ;
no
```

```
| ?- timePlayingGames('Worst Player Ever', ['Duas Botas'], LT, ST).
LT = [0],
ST = 0 ? ;
no
```

```
timePlayingGames(Player, Games, ListOfTimes, SumTimes) :-
    loop(Player, Games, ListOfTimes, []),
    sumlist(ListOfTimes, SumTimes).

loop(_, [], _ []).
loop(Player, [H | T], ListOfTimes, Aux) :-
    played(Player, Game, HoursPlayed, _),
    append(Aux, HoursPlayed, ListOfTimes),
    loop(Player, T, ListOfTimes, ListOfTimes).
```

Comentário:

Pergunta 4

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***listGamesOfCategory(+Cat)*** que imprime na consola os títulos de todos os jogos que se enquadram na categoria *Cat*, indicando ainda a idade mínima recomendada para cada jogo. Note que o predicado sucede sempre.

Exemplos:

```
| ?- listGamesOfCategory(multiplayer).  
5 ATG (18)  
Carrier Shift: Game Over (16)  
yes
```

```
| ?- listGamesOfCategory(simulation).  
yes
```

```
listGamesOfCategory(Cat) :-  
    game(Game, CatsList, Age),  
    member(Cat, CatsList),  
    write(Game), write(' '), write(Age), write(' '), nl,  
    fail.  
listGamesOfCategory(_).
```

Comentário:

Pergunta 5

Respondida

Pontuou 0,600 de 1,500

Implemente o predicado ***updatePlayer(+Player, +Game, +Hours, +Percentage)*** que atualiza a base de conhecimento relativamente ao número de horas que o jogador *Player* jogou o jogo *Game*.

Exemplo:

```
| ?- played('Best Player Ever', 'Duas Botas', Hours, Percent).  
Hours = 33,  
Percent = 22 ? ;  
no
```

```
| ?- updatePlayer('Best Player Ever', 'Duas Botas', 5, 12).  
yes
```

```
| ?- played('Best Player Ever', 'Duas Botas', Hours, Percent).  
Hours = 38,  
Percent = 34 ? ;  
no
```

```
updatePlayer(Player, Game, Hours, Percentage) :-  
    clause(A, played(Player, Game, H, P)),  
    retract(A),  
    assert(played(Player, Game, H+Hours, P+Percentage)).
```

Comentário:

Pergunta 6

Respondida

Pontuou 0,300 de 1,500

Implemente o predicado ***littleAchievement(+Player, -Games)***, que devolve em *Games* a lista de jogos nos quais o jogador *Player* atingiu menos de 20% de completude.

Exemplos:

```
I ?- littleAchievement('Best Player Ever', G).  
G = [] ? ;  
no
```

```
I ?- littleAchievement('Worst Player Ever', G).  
G = ['5 ATG'] ? ;  
no
```

```
littleAchievement(Player, Games) :-  
    loop(Player, Games, []).
```

```
loop(Player, Games, Aux) :-  
    played(Player, Game, _, PercentUnlocked),  
    PercentUnlocked < 20,  
    append(Aux,[Game],Games).
```

Comentário:

Falta recursividade...

Informação

Nas perguntas seguintes pode fazer uso de predicados de obtenção de múltiplas soluções (findall, setof e bagof) e das bibliotecas do SICStus.

Pergunta 7

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***ageRange(+MinAge, +MaxAge, -Players)*** que devolve em *Players* a lista dos nomes dos jogadores com idade compreendida entre *MinAge* e *MaxAge* (limites inclusivos).

Exemplos:

```
I ?- ageRange(12, 16, P).  
P = ['Manny', 'Jonny'] ? ;  
no
```

```
I ?- ageRange(24, 34, P).  
P = ['Danny', 'Annie', 'Harry'] ? ;  
no
```

```
ageRange(MinAge, MaxAge, Players) :-  
    findall(Name, (player(Name, _, Age), Age >= MinAge, Age <= MaxAge), Players).
```

Comentário:

Pergunta 8

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***averageAge(+Game, -AverageAge)*** que determina a idade média dos jogadores que jogam o jogo *Game*.

Exemplos:

```
I ?- averageAge('5 ATG', AA).  
AA = 25.5 ? ;  
no
```

```
I ?- averageAge('Duas Botas', AA).  
AA = 26.5 ? ;  
no
```

```
averageAge(Game, AverageAge) :-  
    findall(Age, (played(Player, Game, _, _), player(_, Player, Age)), AgesList),  
    length(AgesList, NAges),  
    sumlist(AgesList, Sum),  
    AverageAge is Sum / NAges.
```

Comentário:

Pergunta 9

Respondida

Pontuou 2,000 de 2,000

Implemente o predicado ***mostEffectivePlayers(+Game, -Players)*** que determina o jogador ou jogadores que jogam o jogo *Game* com maior eficiência. O jogador mais eficiente é aquele que conseguiu alcançar uma maior percentagem de conclusão do jogo num menor número de horas.

Exemplos:

```
I ?- mostEffectivePlayers('5 ATG', BP).  
BP = ['Best Player Ever'] ? ;  
no
```

```
I ?- mostEffectivePlayers('Carrier Shift: Game Over', BP).  
BP = ['The Player', 'A Player'] ? ;  
no
```

```
mostEffectivePlayers(Game, Players) :-  
    findall(Ratio, (played(Player, Game, HoursPlayed, PercentUnlocked), Ratio is  
PercentUnlocked / HoursPlayed) , List),  
    max_member(Max, List),  
    findall(Player2, (played(Player2, Game, HoursPlayed2, PercentUnlocked2), Ratio2 is  
PercentUnlocked2 / HoursPlayed2, Ratio2 == Max) , Players).
```

Comentário:

Pergunta 10

Respondida

Pontuou 1,500 de 1,500

Indique o que faz o predicado ***whatDoesItDo(?X)***, sugerindo nomes mais apropriados para o predicado e variáveis usadas.

Indique ainda qual a funcionalidade do cut presente no código, justificando a sua resposta.

```
whatDoesItDo(X):-  
    player(Y, X, Z), !,  
    \+ ( played(X, G, L, M),  
          game(G, N, W),  
          W > Z ).
```

```
whatDoesItDo(Username):-  
    player(Name, Username, Age),  
    \+ ( played(Username, Game, HoursPlayed, PercentUnlocked),  
          game(Game, CategoriesList, MinAge),  
          MinAge > Age ).
```

-X é Username, pois é o nome que o jogador usa para se identificar quando joga.

-Estas 4 variáveis seguintes são irrelevantes para o predicado em questão, pelo que poderiam ser substituídas por _

Y é Name por ser o nome do jogador, L é HoursPlayed por ser as horas que o jogador jogou o jogo Game, M é PercentUnlocked por ser a percentagem ultrapassada do jogo

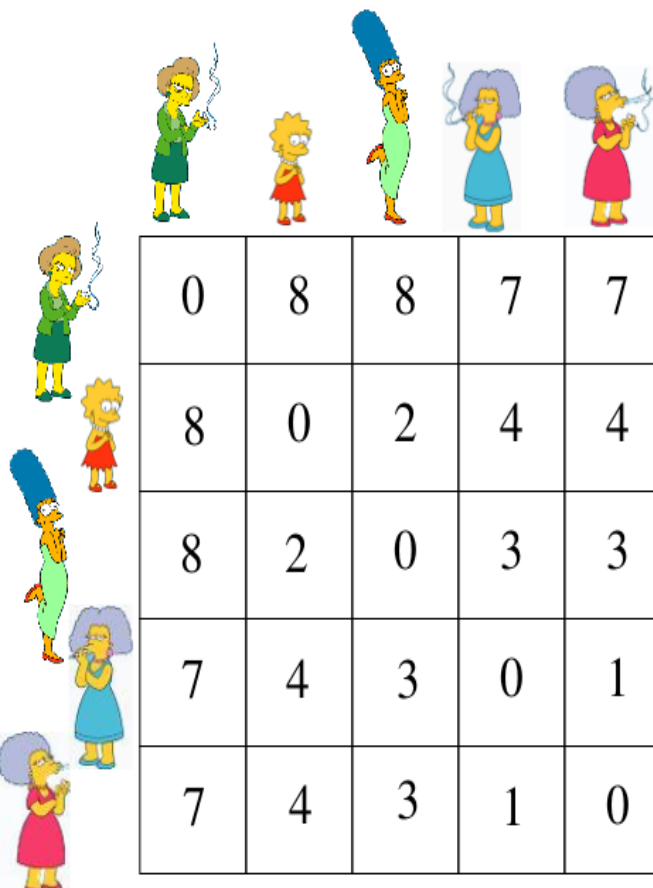
Comentário:

Informação

Uma matriz de distâncias contém em cada célula (i,j) o valor da distância entre o objeto i e objeto j.


$$D(\text{Marge}, \text{Lisa}) = 8$$


$$D(\text{Maggie}, \text{Marge}) = 1$$



	Marge	Lisa	Bart	Maggie	Marge
Marge	0	8	8	7	7
Lisa	8	0	2	4	4
Bart	8	2	0	3	3
Maggie	7	4	3	0	1
Marge	7	4	3	1	0

Figura 1 - Exemplo de uma matriz de distâncias entre 5 objetos.

Pergunta 11

Respondida

Pontuou 0,200 de 0,500

Tendo em consideração a simetria existente nestas matrizes, escolha uma representação em Prolog que considere ser eficiente em termos de utilização de espaço para armazenar uma matriz e represente a informação contida na Figura 1.

Lista de listas ([0,8,8,7,7],[8,0,2,4,4]...)

Comentário:

Não respondida Pontuação 2,000

Não respondida

Pontuação 2,000

Codifique em Prolog o predicado ***areFar/3*** que recebe como primeiro argumento uma distância mínima, como segundo argumento uma matriz de distâncias e instancia o terceiro argumento com a lista de pares de objetos que estão a uma distância igual ou superior à distância dada no primeiro argumento. A lista resultado não deve ter pares simétricos. Identifique os objetos pelo índice da sua linha (ou coluna) na Figura 1.

Exemplo (*MatDist* é a sua codificação da matriz da Figura 1):

```
| ?- areFar(5, MatDist, Pares).
Pares = [2/1, 3/1, 4/1, 5/1] ;
no
```

Informação

Considere o dendograma da Figura 2. Considere que um dendograma é uma árvore binária em que os objetos estão guardados nas folhas e cada nó tem um identificador único. Os objetos têm apenas um nome que é um átomo e os nós internos têm um número inteiro que é único (não há dois nós internos com o mesmo número).

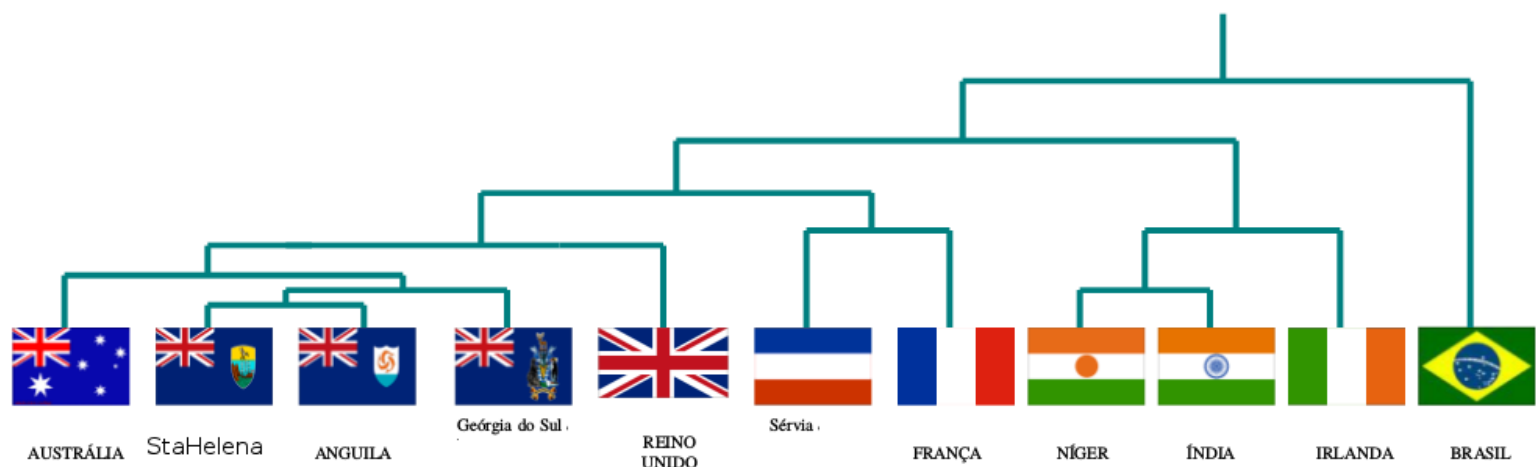


Figura 2 Dendograma de bandeiras de países.

Pergunta 13

Não respondida

Pontuação 1,000

Proponha uma estrutura para codificar dendogramas e represente nessa estrutura o dendograma da Figura 2.



Pergunta 14

Não respondida

Pontuação 2,000

A distância entre dois quaisquer objetos é definida como a altura do nó mais baixo comum aos dois objetos. A altura a que estão as folhas é zero. A altura de um nó interno é 1 mais o máximo entre a altura da sub-árvore esquerda e sub-árvore direita.

Por exemplo:

- a altura da raiz da árvore na Figura 2 é 7
- a distância (na Figura 2) entre a bandeira do Brasil e da Irlanda é 4.

Implemente em Prolog o predicado ***distance/4*** que recebe nos dois primeiros argumentos o nome de dois objetos, recebe um dendograma no terceiro argumento e devolve no quarto argumento a distância entre os dois objetos dados. Assuma que os objetos existem sempre no dendograma.

Exemplo:

```
I ?- distance(brasil, niger, DendogramaFigura2, Distancia).  
Distancia = 4 ? ;  
no
```