



UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CURSO: BACHARELADO EM SISTEMAS DE
INFORMAÇÃO



DISCIPLINA: Programação para Web I
PROFESSOR (A): Juliana Oliveira de Carvalho
ALUNO: Luís Eduardo Silva Brito
C.H.: 60 h CRÉDITOS: 1.1.0 PERÍODO: 2022.2

**Análise de predicados, para resolver problemas do
cotidiano, por meio da programação lógica**

Resumo

O projeto trata da discussão dos problemas apresentados pela professora, referente a disciplina de programação lógica, e de como resolver esses problemas por meio da programação lógica, apresentando predicados que são eficientes para a solução do problema então abordado. Os testes dos predicados são feitos através de uma ferramenta, o SWI-Prolog, o objetivo principal do projeto é verificar se uma solução realmente resolve o problema então discutido.

I. Introdução

Os problemas do cotidiano são muito comuns para a área de programação, os programadores buscam soluções eficientes para que um determinado problema seja resolvido, por meio do raciocínio lógico e codificação. Com o uso da ferramenta SWI Prolog foi possível testar as soluções então propostas, com intuito de satisfazer a todos os possíveis casos de entrada de variáveis. A parte de resultados foi feita através de tabelas, onde foram inseridas algumas entradas, cada problema abordava um assunto diferente, e dessa forma, foi necessário inserir tanto valores positivos quanto valores negativos, afim de verificar a eficiência do predicado, e a partir dessas verificações, comprovar que de fato o predicado satisfaz o que é realmente pedido no problema.

II. Sessões Específicas

QUESTÃO 1:

Na questão 1, foram usadas os predicados de `imprimirimpostoDrestituicao` e `aliquota`. No predicado de imprimir o imposto e a restituição, foram passados por parâmetros de entrada o salário mensal, o imposto pago mensalmente, e como parâmetros de saída o imposto devido e a restituição.

Cláusula 1:

imprimirimpostoDrestituicao(Salario_Bruto_Mensal,Imposto_Pago_Mensal,ImpostoDevido,Restituicao):-

A cláusula 1 têm como objetivo calcular o valor do imposto devido e da restituição de uma pessoa, permitindo somente valores de entrada maiores ou iguais a 0, e a alíquota sendo diferente de 0.

Cláusula 2:

imprimirimpostoDrestituicao(Salario_Bruto_Mensal,_,ImpostoDevido,Restituicao):-

A cláusula 2 têm como objetivo calcular o imposto devido e a restituição de uma pessoa, caso o valor da alíquota seja igual a 0.

Cláusula 3:

imprimirimpostoDrestituicao(_,_,0,0):-

A cláusula 3 imprime uma mensagem de erro caso os valores de entrada sejam menores que 0.

O predicado de alíquota tem como objetivo retornar o valor da alíquota de acordo com a tabela presente na questão, para cada cláusula o valor da alíquota é alterado, foram passados por parâmetro a base do calculo que está relacionado ao salario anual, e um parâmetro de saída, no caso a aliquota.

5 cláusulas alíquota:

aliquota(BaseDecalculo, Aliquota)

QUESTÃO 2:

Na questão 2, foram usados os predicados imprimiringressosessoes e custototal. No predicado de imprimir os ingressos, foram passados como parâmetros de entrada o custo do filme, o ingresso, a capacidade, o custo adicional, e como parâmetros de saída o Result1 que representa a quantidade de ingressos que precisam ser vendidos, e Result2 a quantidade de sessões necessárias.

Cláusula 1:

imprimiringressosessoes(Custo_filme,Ingresso,Capacidade,Custoadicional,Result1,Result2):-

A cláusula 1 têm como objetivo calcular e imprimir a quantidade de ingressos e a quantidade de sessões necessárias, permitindo somente valores de entrada maiores que 0, e o custo total do filme sendo diferente de 0.

Cláusula 2:

imprimiringressosessoes(_,_,_,_,0,0):-

A cláusula 2 imprime uma mensagem de erro caso os valores das entradas sejam menores ou iguais a 0.

O predicado custototal tem como objetivo calcular o preço que foi investido pelo dono, retornando o custototal de investimentos.

Cláusula 1:

custototal(Custo_filme,Ingresso,Capacidade,Custoadicional,CustoTot):-

Na cláusula 1, foram passados por parâmetro de entrada o custo do filme, o preço do ingresso, a capacidade de cada sessão, e o custo adicional que é um valor adicionado por quantidade de sessões, tem como objetivo retornar o custo total do filme.

QUESTÃO 3:

Na questão 3 foram usados os predicados `quadruplo(Numero,Cont,Result)`, `mult5(Numero,Rest)`, `imprimir(Numero,Cont,Quad,Mult)`, No predicado de `quadruplo` foram passados como parâmetros de entrada um numero e um contador, e como parâmetro de saída o `Result`, que retorna o quadruplo do numero.

Cláusula 1:

`quadruplo(Numero,Cont,Result)`

A cláusula 1 tem como objetivo calcular o quadruplo de um numero caso ele seja maior do que 0.

Cláusula 2:

`quadruplo(Numero,_,Result)`

A cláusula 2 retorna o quadruplo do numero.

No predicado `mult5(Numero,Rest)`, foram passados por parâmetro de entrada um número e como parâmetro de saída o `Rest`, que é o resto.

Cláusula 1:

`mult5(Numero,Rest)`

A cláusula 1 subtrai do numero o valor de 5, até que o numero seja menor que 5.

Cláusula 2:

`mult5(Numero,Rest)`

A cláusula 2 verifica se o resto é maior que 0, se ele for, imprime uma mensagem de que o numero não é múltiplo de 5.

Cláusula 3:

`mult5(Numero,Rest)`

A cláusula 2 verifica se o resto é 0, se ele for, imprime uma mensagem de que o numero é múltiplo de 5.

No predicado de `imprimir(Numero,Cont,Quad,Mult)`, foram passados por parâmetro de entrada o numero e um contador, e como parâmetros de saída o `Quad` e o `Mult`.

Cláusula 1:

`imprimir(Numero,Cont,Quad,Mult)`

A cláusula 1 retorna o quadruplo de um numero e se ele é múltiplo de 5.

QUESTÃO 4:

Na questão 4 foram usados os predicados `convert_Binario_para_decimal`, `convert_decimal_para_Binario`, `produtoNM`. No predicado de converter binário para decimal, foram passados como parâmetros de entrada um número binário, um

contador e um acumulador, e como parâmetro de saída o Result, que retorna um decimal.

Cláusula 1:

convert_Binario_para_decimal(Binario,Cont,Acumula,Result)

A cláusula 1 confere se o decimal é maior que 1, e acumula os restos das divisões, e multiplica eles por 2 elevado ao contador.

Cláusula 2:

convert_Binario_para_decimal(_,_,Acumula,Result)

A cláusula 2 retorna a conversão do número binário para um número decimal.

O predicado converte decimal em binário, foi passado por parâmetro de entrada um decimal e como parâmetro de saída um número binário.

Cláusula 1:

convert_decimal_para_Binario(Decimal,Bin)

A cláusula 1 confere se o decimal é maior ou igual a 2, e converte ele em um número binário.

Cláusula 2:

convert_decimal_para_Binario(Decimal,Bin)

A cláusula 2 retorna a conversão de um número decimal para um número binário.

No predicado produtoNM foram passados como parâmetros de entrada um número decimal e um número binário, e como parâmetros de saída o Result, que tem como objetivo retornar o produto entre eles.

Cláusula 1:

produtoNM(Decimal,Binario,Resultado1, Resultado2)

A cláusula 1 chama os predicados de conversão, e realiza o produto entre naturais e retorna esse produto em forma de número binário.

QUESTÃO 5:

Na questão 5, foram usados os predicados adicFinal, pertence, len, naoExisteLista2, naoExisteLista1, elementoIndicePar, elementoIndiceImpar, programaprincipal.

Predicado 1:

adicFinal(X,[Y|L], [Y|W]),

O predicado 1, tem como objetivo inserir no final da lista um elemento

Predicado 2:

pertence(X,[_|T])

O predicado 2, tem como objetivo descobrir se um elemento pertence a uma lista.

Predicado 3:

len([_|T],N)

O predicado 3, tem como objetivo calcular o tamanho da lista.

Predicado 4:

naoExisteLista2 ([Elem1|C1], [Elem2|C2], Acumula, ListaFinal)

O predicado 4, tem como objetivo descobrir se um elemento não pertence a lista 2.

Predicado 5:

naoExisteLista1([Elem1|C1], [Elem2|C2], Acumula, ListaFinal)

O predicado 5, tem como objetivo descobrir se um elemento não pertence a lista 1.

Predicado 6:

elementoIndicePar([Elem|R], Indice, ListaAcumula, ListaFinal)

O predicado 6, tem como objetivo descobrir se o elemento é da posição par.

Predicado 7:

elementoIndiceImpar([Elem|R], Indice, ListaAcumula, ListaFinal)

O predicado 7, tem como objetivo descobrir se o elemento é da posição ímpar.

Predicado 8:

programaprincipal(Lista1, Lista2, ResultA, ResultB)

O predicado 8 tem como objetivo retornar a união das listas.

III. Resultados da execução do programa

O problema 1 trata do imposto devido e a restituição de uma pessoa, onde envolve o salário bruto anual que essa pessoa recebe, o salario bruto mensal, o imposto pago anual, o imposto pago mensal. O predicado criado em Prolog tem como objetivo realizar o cálculo do imposto devido por uma pessoa, levando em consideração a alíquota, que dependendo da situação da pessoa, pode possuir um valor Isento ou um valor que vai alterando de acordo com o salário anual. E a partir do imposto devido, calcular a restituição.

A tabela 1 está relacionada aos valores de entrada e saída que foram usados nos testes, como podemos perceber os valores de entrada são o salário mensal da pessoa, e o imposto mensal que é pago, e os valores de saída são o imposto devido e a restituição.

Entrada	Saída
Salário bruto mensal (1300) Imposto pago mensal (100)	Isento: Imposto Devido = 0, Restituição = 1200.
Salário bruto mensal (4500) Imposto pago mensal (200)	Imposto Devido = 4968, Restituição = 2568.
Salário bruto mensal (-3500)	Valores inválidos

Imposto pago mensal (500)	Imposto Devido= Restituição, Restituição = 0.
---------------------------	--

Tabela 1: Testes de entradas para a solução do problema 1

No primeiro teste a saída foi isenta e o imposto devido 0, ou seja, o cálculo realizado em relação do salário anual foi inferior a 35000, ocasionando em Isento, por esse motivo o imposto devido foi igual a 0. O programa atende até mesmo a entradas negativas, como é o exemplo do terceiro teste, onde os valores do salário são negativos, resultando em valores inválidos, ou seja, o programa não calcula o imposto devido e a restituição, atribuindo o valor 0 a eles.

O problema 2 trata do dono de um cinema onde, ele precisa saber quantos ingressos precisarão ser vendidos para que ele obtenha lucro em cima do valor que foi investido por ele, e também a quantidade de sessões que irão ser utilizadas.

A tabela 2, está relacionada aos valores de entrada e saída que foram usados nos testes, os valores de entrada são o custo do filme, preço do ingresso, capacidade de pessoa em uma sessão, custo adicional por sessão, e os valores de saída são o total de ingressos para que o dono obtenha lucro, e a quantidade de salas que irão ser ocupadas pelas pessoas.

Entrada	Saída
Custo do filme (150000) Ingresso (30) Capacidade (400) Custo adicional (3000)	Tem que vender: 6251 ingressos, Quantidade de salas:15,6275
Custo do filme (-120000) Ingresso (40) Capacidade (400) Custo adicional (2000)	Valores incorretos Result1 = Result2, Result2 = 0.
Custo do filme (100000) Ingresso (20) Capacidade (200) Custo adicional (2500)	Tem que vender: 8126 ingressos, Quantidade de salas: 40,63

Tabela 2: Testes de entradas para a solução do problema 2

Como podemos observar um dos testes entram com valores negativos, porém o programa retorna que os valores são incorretos, ou seja, ele não permite valores negativos. E nos outros testes ele retorna o total de ingressos, e a quantidade de salas.

O problema 3, trata de um número, os predicados devem retornar o quadruplo desse número e se ele é múltiplo de 5.

A tabela 3 está relacionada aos valores de entrada e saída que foram usados nos testes, os valores de entrada são um número e um contador, e os valores de saída são o quadruplo desse número e uma mensagem se ele é ou não múltiplo de 5.

Entrada	Saída
Número(4) Cont(0)	não é múltiplo de 5, Quad = 16,
Número(5) Cont(0)	E múltiplo de 5, Quad = 20
Número(-7) Cont(0)	False

Tabela 3: Testes de entradas para a solução do problema 3

Como podemos perceber o último teste entra com um numero negativo, porém o programa retorna False, ou seja, ele não permite números negativos. Os demais testes retornam o quadruplo e a mensagem se ele for múltiplo de 5 ou não for múltiplo de 5.

O problema 4, trata de conversão de bases numéricas para efetuar um produto entre um número decimal e um número binário.

A tabela 4 está relacionada aos valores de entrada e saída que foram usados nos testes, os valores de entrada são um número decimal e um número binário, e o valor de saída é o produto entre esses números convertido em número binário.

Entrada	Saída
Decimal(10), Binario(101011), Cont(0), Recebe(0)	ResultBinario = 11001001100100
Decimal(3), Binario(111), Cont(0), Recebe(0)	ResultBinario = 10111101
Decimal(-7), Binario(1111), Cont(0), Recebe(0)	Valor invalido ResultBinario = 0.

Tabela 4: Testes de entradas para a solução do problema 4

É possível observar que o predicado atende ao problema em questão, e trata até mesmo das entradas negativas de decimais, como por exemplo no teste 3, que a entrada é negativa o predicado retorna um valor inválido.

O problema 5, trata da união entre duas listas de inteiros, onde deve retornar valores da segunda lista que não existem na primeira, e valores da primeira que não

existem na segunda. E também deve retornar a união ordenada dos elementos das posições pares da primeira, com elementos das posições ímpares da segunda.

A tabela 5 está relacionada aos valores de entrada e saída que foram usados nos testes, os valores de entrada são duas listas de inteiros, e os valores de saída são duas listas com os retornos descritos na abordagem da questão.

Entrada	Saída
[1,2,3,4,5], [1,5,6,7]	ResultA = [2, 3, 4, 6, 7], ResultB = [1, 3, 5, 7]
[1,2,3,4,5,6,7,8,9,10], [1,3,5,7,9]	ResultA = [2, 4, 6, 8, 10], ResultB = [1, 3, 5, 7, 9]
[2,4,6,8,10], [1,3,5,7,9]	ResultA = [1, 2, 3, 4, 5, 6, 7, 8, 9 ...], ResultB = [2, 3, 6, 7, 10]

Tabela 5: Testes de entradas para a solução do problema 5

Como podemos observar as entradas são listas com números inteiros, e a saída está de acordo com o que a questão propôs. O resultado A retorna a união dos elementos que não pertencem, e o resultado B retorna a união dos elementos pares da primeira com os elementos ímpares da segunda.

IV. Conclusão

Portanto, podemos concluir que as soluções que foram propostas para resolver os problemas abordados são eficientes, pois resolvem o problema de forma correta, e não deixam passar valores negativos nos parâmetros de entrada, que por consequência poderia ocorrer em um erro. Dessa forma, atendem de maneira eficaz, evitando o usuário de digitar valores errados nos valores de entrada, então atingindo os objetivos propostos em sala e avaliando a eficácia dos predicados para a resolução dos problemas.