

Ficha Prática 4

4.1 Objectivos

1. Perceber o operador de unificação.
2. Praticar a escrita de predicados envolvendo expressões aritméticas.

4.2 Conceitos

4.2.1 Unificação

A unificação é a operação mais importante que o interpretador de Prolog realiza. A unificação representa-se pelo símbolo “=” ($A=B$ lê-se: A unifica com B) e é essencialmente um processo de comparação. Este processo é imediato para átomos e números. Dois átomos unificam-se e só se são o mesmo átomo. Do mesmo modo, dois números unificam-se e só se são o mesmo número. Por exemplo, a unificação `ola=ola` sucede, a unificação `3=2` não sucede.

Para termos mais complexos a unificação é efectuada tentando unificar os sub-termos que os compõem. Se algum dos sub-termos for, por sua vez, um termo complexo, o algoritmo é aplicado recursivamente até o processo de unificação ficar reduzido a comparação de termos elementares. Por exemplo, a unificação `hora(12,45)=hora(12,45)` sucede pois as unificações `hora=hora`, `12=12` e `45=45` todas sucedem.

Se os termos a unificar possuírem variáveis não instanciadas, estas são instanciadas durante o processo de unificação. Na verdade, a unificação é o único modo pelo qual as variáveis são instanciadas. Se um dos termos a unificar é uma variável não instanciada e o outro um átomo, a unificação sucede e a variável é instanciada com o átomo. Se os dois termos a unificar forem variáveis não instanciadas, elas unificam e os nomes das duas variáveis tornam-se sinónimos para a mesma variável (não instanciada). Por exemplo, a unificação `hora(12,45)=hora(12,M)` sucede e a variável M fica instanciada com 45. A unificação `hora(12,M)=hora(12,N)` sucede e as variáveis M e N passam a ser a mesma variável. Já a expressão `hora(12,M)=hora(12,N)`, $M=12$, $N=5$ não sucede. Porquê?

4.2.2 Aritmética em Prolog

Tal como descrito na secção anterior, o operador de unificação procede à comparação de termos e efectua as instanciações necessárias para que essa comparação suceda. O operador é simbólico pelo que não efectua qualquer interpretação dos termos em causa. Assim, a unificação $A=2+1$ sucede com A a ficar instanciado com a expressão $2+1$ (e não com o valor 3) e a unificação $3=2+1$ falha pois os dois termos (3 e $2+1$) são diferentes.

O cálculo aritmético em Prolog é efectuado utilizando um conjunto pré-definido de predicados aritméticos que manipulam expressões aritméticas. A tabela 4.2 apresenta uma lista de predicados aritméticos do SWI-Prolog⁷.

⁷A indicação “+”, “-” e “?” nos argumentos tem o seguinte significado: “+” — o argumento deve estar instanciado; “-” — o argumento deve ser uma variável não instanciada; “?” — o argumento pode, ou não, estar instanciado.

Predicado	Significado
<code>between(+Min, +Max, ?Valor)</code>	sucede se $\text{Min} < \text{Valor} < \text{Max}$
<code>succ(?Int1, ?Int2)</code>	sucede se $\text{Int2} = \text{Int1} + 1$
<code>plus(?Int1, ?Int2, ?Int3)</code>	sucede se $\text{Int3} = \text{Int1} + \text{Int2}$ (pelo menos dois argumentos devem estar instanciados)
<code>+Expr1 > +Expr2</code>	sucede se o valor da expressão Expr1 for maior que o valor da expressão Expr2
<code>+Expr1 < +Expr2</code>	sucede se o valor da expressão Expr1 for menor que o valor da expressão Expr2
<code>+Expr1 >= +Expr2</code>	sucede se o valor da expressão Expr1 for maior ou igual ao o valor da expressão Expr2
<code>+Expr1 =< +Expr2</code>	sucede se o valor da expressão Expr1 for menor ou igual ao o valor da expressão Expr2
<code>+Expr1 =\= +Expr2</code>	sucede se o valor da expressão Expr1 for diferente do valor da expressão Expr2
<code>+Expr1 := +Expr2</code>	sucede se o valor da expressão Expr1 for igual ao valor da expressão Expr2
<code>-Número is +Expr</code>	sucede se Número unificar com o o valor da expressão Expr

Tabela 4.2: Predicados aritméticos do SWI-Prolog

As expressões aritméticas podem ser construídas a partir de funções aritméticas, números e variáveis. O SWI-Prolog disponibiliza as funções aritméticas usuais (soma, subtração, etc.). Para uma lista completa das funções existentes, e respectiva notação, consultar a secção 4.27 do manual⁸.

4.3 Exercícios

4.3.1 Unificação

Para cada uma das expressões seguintes diga qual é o resultado da unificação (verifique as suas resposta no Prolog):

- `Y=5, X is 10+Y.`
- `Y=X+5, Z is Y+1, X=5.`
- `data(D,M,A)=data(5,D+1,M+1990).`
- `data(2000,12,21,H)=data(A,M,D,hora(14+D,33,0)).`
- `data(2000,12,21,hora(H,M,S))=data(A,M,D,hora(14,X,0)).`
- `data(2000,12,21,hora(H,Min,S))=data(A,M,D,hora(14,X,0)).`

4.3.2 Expressões Aritméticas

Para cada uma das seguintes funções matemáticas, defina um predicado Prolog equivalente:

- factorial:

$$fact(n) = \begin{cases} 1 & \Leftarrow n = 0 \\ fact(n-1) * n & \Leftarrow n > 0 \end{cases}$$

⁸Utilize a *querie*:
`?- help(4-27).`

2. multiplicação de números naturais por somas sucessivas:

$$\text{mult}(x, y) = \begin{cases} 0 & \Leftarrow y = 0 \\ x + \text{mult}(x, y - 1) & \Leftarrow y > 0 \end{cases}$$

3. divisão inteira de números naturais por subtrações sucessivas:

$$\text{div}(x, y) = \begin{cases} 0 & \Leftarrow x < y \\ 1 + \text{div}(x - y, y) & \Leftarrow x \geq y \end{cases}$$

4. resto da divisão inteira de números naturais por subtrações sucessivas:

$$\text{mod}(x, y) = \begin{cases} x & \Leftarrow x < y \\ \text{mod}(x - y, y) & \Leftarrow x \geq y \end{cases}$$

5. função de Fibonacci:

$$F(N) = \begin{cases} F(N - 1) + F(N - 2) & \Leftarrow N > 2 \\ 1 & \Leftarrow N = 2 \\ 1 & \Leftarrow N = 1 \end{cases}$$

6. função de Ackerman:

$$A(M, N) = \begin{cases} N + 1 & \Leftarrow M = 0 \\ A(M - 1, 1) & \Leftarrow N = 0 \\ A(M - 1, A(M, N - 1)) & \text{noutro caso} \end{cases}$$

4.3.3 Planeamento arquitectónico

Escreva um programa Prolog que permita realizar o planeamento arquitectónico de um edifício com base nas seguintes restrições:

- o edifício deverá consistir em duas divisões rectangulares;
- as divisões deverão estar alinhadas com os pontos cardeais;
- cada divisão deverá possuir uma janela e uma porta interior;
- as divisões estão ligadas pela porta interior;
- uma das divisões também deverá possuir uma porta exterior;
- uma parede pode apenas conter uma janela ou uma porta;
- nenhuma janela deverá estar virada a norte;
- as janelas não podem estar situadas em faces opostas do edifício.

O programa deverá definir o predicado `plano/2` de tal forma que:

```
plano(sala(PFrente, PInt1, Jan1), sala(PInt2, Jan2)) :- a primeira
sala tem a porta da frente na parede PFrente9, a porta interior na parede PInt1 e
a janela na parede Jan1, e a segunda sala tem a porta interior na parede PInt2 e
a janela na parede Jan2.
```

⁹Norte, Sul, Este ou Oeste.