

Guia de Referência de Prolog

Interpretação de predicados

nome/2 "nome" é o nome do predicado e 2 é o número de parâmetros do predicado

Execução de Consultas

. termina uma consulta no swipl
; procura o próximo resultado da consulta
[programa]. carrega as cláusulas do arquivo `programa.pl`.

Termos aritméticos

+/2 Adição: $3 + 5$.
+/1 Prefixo positivo (geralmente desnecessário): **+3**
-/2 Subtração: $5 - 3$
-/1 Prefixo negativo: **-1**
/ /2 Divisão: $3/5 = 0.6$
// /2 Divisão inteira: $3//5 = 0$
mod/2 Módulo: $5 \bmod 3 = 2$
log/1 Logaritmo: $\log(1) = 0.0$

Operadores

is/2 Avaliação: Z is T avalia o termo aritmético T e o resultado é unificado com a variável Z. X is $3 + 5$. X = **8**
== Igualdade aritmética
=\= Desigualdade aritmética
< Relações de ordem
>
=<
>=

Predicados Meta-lógicos

var(X) X é atualmente uma variável livre
nonvar(X) X não é uma variável livre
ground(X) X é um termo sem variáveis

Predicados de Tipos

integer(X) Checa quando X é um inteiro
float(X) Checa quando X é um número em ponto flutuante
number(X) Checa quando X é um número
atom(X) Checa quando X é uma constante diferente de um número
atomic(X) Checa quando X é uma constante

Negação

\+ “não” ou “não pode provar”. Corresponde à **negação por falha**.

Comparação de Termos

== igualdade entre termos (desigualdade pode ser a negação de igualdade), quando os argumentos já tem o mesmo valor. $X==Y$ testa se dois parâmetros X e Y possuem o mesmo valor.

\== negação de **==** (equivalente a **\+ ==**)

= igualdade entre termos, tentando unifica os dois parâmetros. $X=Y$ tenta unificar X e Y

Operadores de Pruning (Interrupção)

!/0 Operador de corte ("cut")
fail/0 Força uma falha na avaliação.

Meta-Chamadas (Alta ordem)

call(X) Aceita o termo e o invoca como se ele aparecesse no programa. $\text{call}(\text{p}(\text{X}))$ é equivalente a ter $\text{p}(\text{X})$ no texto do programa. O argumento X de $\text{call}(\text{X})$ (e onde o poder de uma meta-chamada está) não precisa estar explicitamente no código fonte, mas apenas corretamente instanciado em tempo de execução.

findall(Termo, Objetivo, Lista) Deixa em Lista uma instância de Termo para cada sucesso do Objetivo.

bagof(Termo, Objetivo, Lista) Permite o backtracking (retrocesso) seletivo de variáveis livres no objetivo

setof(Termo, Objetivo, Lista) Se comporta como **bagof/3**, mas a lista retornada é ordenada e não possui repetições.

Entrada/Saída

write(X) Escreve o termo X na stream de saída

nl Inicia uma nova linha ($\backslash n$) na stream de saída

read(X) Lê um termo (terminado por parada completa) da stream de entrada atual e unifica com X

put(N) Escreve o código ASCII de N, que deve ser uma string de tamanho um

get(X) Lê o código do próximo caracter e unifica seu código ASCII com N

see(Arquivo) Arquivo passa a ser a stream de entrada atual

seeing(Arquivo) A stream de entrada atual é Arquivo
seen Fecha a stream de entrada

tell(Arquivo) Arquivo se torna a stream de saída atual

telling(Arquivo) A stream de saída atual é Arquivo

told Fecha a stream de saída atual