Execução sequencial, da esquerda para a direita, dos objetivos da resolvente.

Pesquisa sequencial de uma cláusula unificável e retrocesso (backtracking)

- Escolhe a primeira cláusula cuja cabeça unifica com o objetivo.
- Se não houver, a computação é desfeita até à última escolha (ponto de escolha), e é escolhida a cláusula unificável seguinte.

A maior parte das implementações de Prolog:

- Pesquisa a árvore até encontrar a primeira solução
- Permite ao utilizador indicar que quer mais soluções através do símbolo ;

Heurísticas de Ordenação

- Colocar testes primeiro
- Colocar primeiro os objetivos com menos soluções (depende da base de dados)
- Colocar primeiro os objetivos mais instanciados (depende do uso)
- Objetivo: falhar o mais rápido possível!
  - falhar significa podar a árvore de pesquisa, levando mais depressa à solução

Avaliador aritmético: is(Value, Expression)

- Value is Expression
- A expressão Expression é avaliada e o resultado é unificado com Value
  - a expressão não pode conter variáveis não instanciadas.
- Tem sucesso se a unificação tiver sucesso. Exemplos:
  - X is 3 + 5? X = 8
  - 8 is 3 + 5? yes
  - 3 + 5 is 3 + 5? false

No Prolog, a recursividade é usada para especificar algoritmos recursivos e iterativos

- Cláusula iterativa: chamada recursividade é o último objetivo do corpo.
- Procedimento iterativo: se contiver apenas factos e cláusulas iterativas.

O cut (!) permite afetar o comportamento procedimental dos programas

- Principal função: reduzir o espaço de procura podando dinamicamente a árvore de pesquisa
  - Reduz tempo de computação
  - Reduz espaço, pois alguns pontos de escolha deixam de ser necessários

O cut sucede e compromete o Prolog com todas as escolhas feitas desde que o objetivo pai foi unificado com a cabeça da cláusula onde o cut ocorre.

Em caso de retrocesso no cut, a pesquisa continuará a partir da última escolha feita antes da escolha desta cláusula.

fail: predicado que provoca falha.

Cut verde:

- não altera o significado do programa: o mesmo conjunto de soluções é encontrado com ou sem o cut.
- corta apenas caminhos de computação que não levam a novas soluções.

Cut vermelo:

- se retirado, altera o significado do programa: o conjunto de soluções será diferente.
- a ordem das cláusulas passa a ser fixa.

A omissão de condições transforma cuts verdes em vermelhos:

- Cut verde:

```
minimum(X, Y, X) :- X <= Y, !.
minimum(X, Y, Y) :- X > Y, !.
```

- Cut vermelho:

```
minimum(X, Y, X) :- X <= Y, !.
minimum(X, Y, Y).
```

Inspeção de estruturas:

- functor(Term, F, Arity) / arg(N, Term, Arg)
  - Decomposição de termos
  - Criação de termos
- Term =.. List
  - Construir um termo a partir de uma lista
  - Construir uma lista a partir de um termo

Predicados meta-lógicos:

- var(Term) / nonVar(Term)
  - Verificam se Term está ou não instanciado
- == / \==
  - Verificam se dois termos são ou não idênticos

Meta-variável: variável usada como um objetivo no corpo de uma cláusula. Durante a computação, aquando da sua invocação a variável deverá estar instanciada com um termo (se não, erro).

Strings e Códigos ASCII

- String (entre aspas) corresponde a uma lista de inteiros que são os códigos ASCII de cada carácter na string
- name(X, Ys): converte átomo X na lista Ys com códigos ASCII dos caracteres de X
- put(N): escreve carácter cujo código ASCII é N
- get0(N): lê carácter e unifica o seu código ASCII com N
- get(N): lê carácter não branco

Ficheiros:

- see(F): abre canal de leitura para o ficheiro F. Leituras passam a ser feitas a partir de F.
- tell(F): abre canal de escrita para o ficheiro F. Escritas passam a ser feitas para F.
- seeing(F)/telling(F): F é unificado com o nome do ficheiro do canal corrente.
- seen/told: fecha o canal corrente.

Acesso e Manipulação do Programa

- assertz(Clause)/asserta(Clause): adiciona Clause como última/primeira cláusula do procedimento.
- retract(C): remove a primeira cláusula que unifica com C.
- consult(File): lê e adiciona (assert) as cláusulas do ficheiro File.
- reconsult(File): faz retract das cláusulas antes de consult.
- clause(Head, Body): procura cláusula cuja cabeça unifica com Head.