

Linguagem Prolog

Conceitos Básicos

TURMA A
PROF. MARCELO LADEIRA CIC/UNB

ADAPTADOS A PARTIR DE SLIDES OBTIDOS NA INTERNET – PARTE 4 DE 4

1

Cut - fail, cut e espaço de busca

O cut é usado em geral para manusear exceções e restringir o espaço de busca.

Cut - fail

elegivel(X) :- analfabeto(X), !, fail.

elegivel(X) :- cidadao(X), idade(X,Y), Y >= 18.

cidadao(joao).

cidadao(susana).

cidadao(roberta).

cidadao(ricardo).

cidadao(marcelo).

idade(joao, 36).

idade(susana, 15).

idade(roberta, 31).

idade(ricardo, 22).

idade(marcelo, 17).

analfabeto(joao).

?- elegivel(joao).

false. % busca mínima, cut ok

?- elegivel(ricardo).

true. % busca mínima, cut ok

?- elegivel(X).

false. % como? Nesse caso devo tirar o “!” ?

?- elegivel(X). % tirando o cut

X = joao ; % como? Ele é analfabeto!

X = roberta ;

X = ricardo ;

false.

Cut - fail

```
elegivel(X) :- cidadao(X),  
              analfabeto(X), fail.  
elegivel(X) :- cidadao(X), not( analfabeto(X)),  
              idade(X,Y), Y >= 18.  
cidadao(joao).  
cidadao(susana).  
cidadao(roberta).  
cidadao(ricardo).  
cidadao(marcelo).  
idade(joao, 36).  
idade(susana, 15).  
idade(roberta, 31).  
idade(ricardo, 22).  
idade(marcelo, 17).  
analfabeto(joao).
```

?- elegivel(X).

X = roberta ;

X = ricardo ;

false.

?- elegivel(joao).

false.

?- elegivel(roberta).

true.

?- elegivel(marcelo).

false.

?- elegivel(susana).

false.

nem sempre o cut é a melhor solução.

Cut - fail

```
a(1) :- write('um'), nl.  
a(X) :- d(X), write('dois'), nl.  
a(3) :- write('tres'), nl.  
d('2a').  
d('2b').  
?- a(N).  
um  
N = 1 ;  
dois  
N = '2a' ;  
dois  
N = '2b' ;  
tres  
N = 3  
?-
```

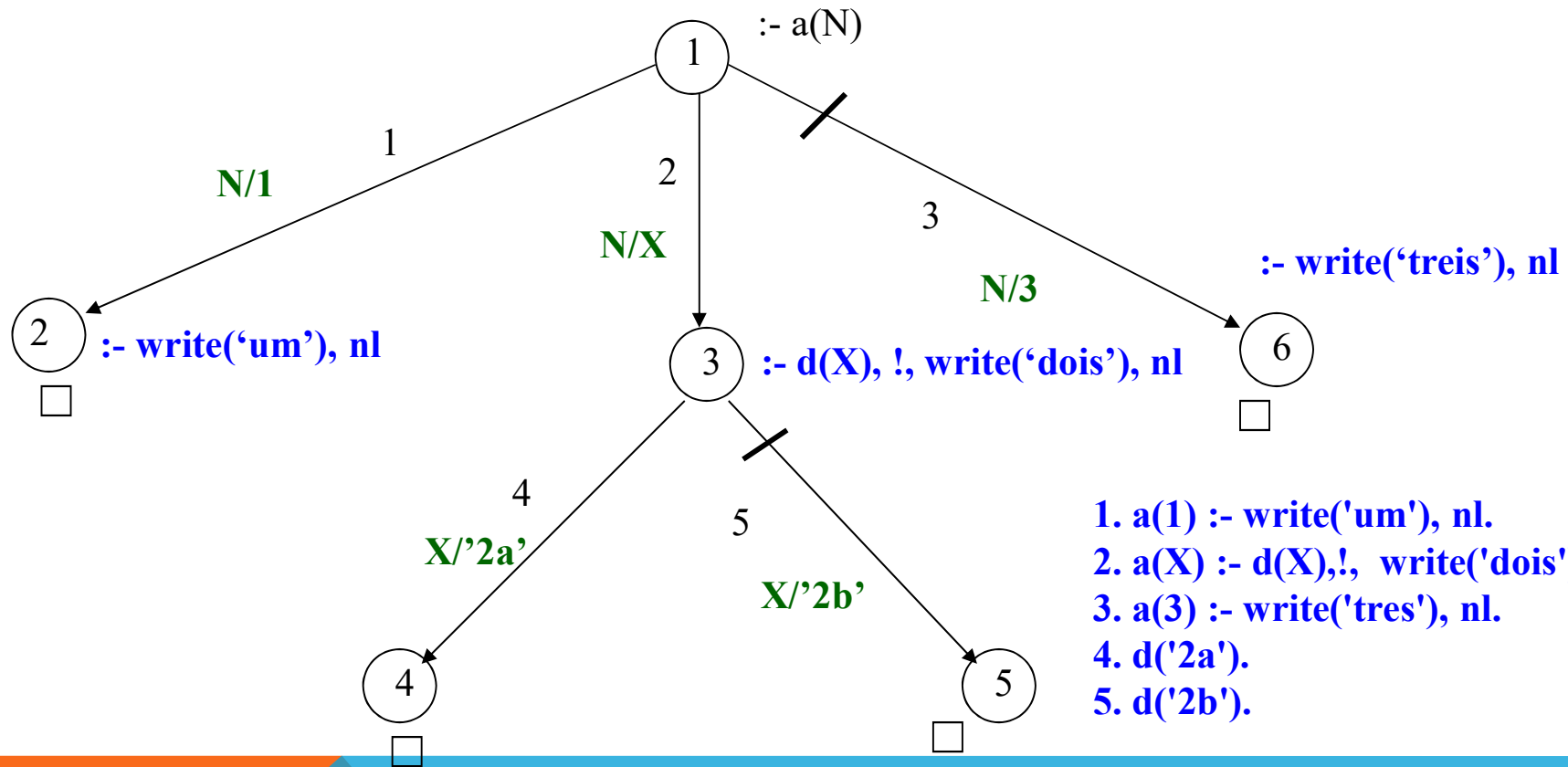
```
a(1) :- write('um'), nl.  
a(X) :- d(X),!, write('dois'), nl.  
a(3) :- write('tres'), nl.  
d('2a').  
d('2b').
```

```
?- a(N).  
um  
N = 1 ;  
dois  
N = '2a'
```

?-

Obs.: faça a árvore de refutação para ter uma melhor compreensão do cut

cut e espaço de busca



Cut e espaço de busca

O espaço de busca de uma consulta é o conjunto de todas as respostas possíveis.

Cut

O corte é usado para reduzir o espaço de busca, instruindo o interpretador a não retroceder sobre os predicados que precedem o cut.

- Desvantagens
 - sacrifica a clareza do programa
 - causa disruptura brusca na execução do programa.
- Efeito sobre consultas compostas
| ?- a(X), b(Y), !, c(X,Y,Z)
obtenha um único valor para X e Y e avalie c(X,Y,Z).
- Efeito sobre cláusulas de um mesmo predicado
 - se uma das cláusulas tem um corte no seu corpo, e o Prolog o alcança, então ele não faz retrocesso sobre outras cláusulas do conjunto.
(Ex.: ++, e a(N) com árvore de refutação). Estude o caso de ++ com corte, sobre a árvore de refutação.

Parando depois da 1a resposta

cidadao(joao).
cidadao(susana).
cidadao(roberta).
cidadao(ricardo).
cidadao(marcelo).
idade(joao, 36).
idade(susana, 15).
idade(roberta, 31).
idade(ricardo, 22).
idade(marcelo, 17).
analfabeto(joao).

?- idade(joao, Y), !.

Y = 36

Obs.: dado que cada pessoa só tem uma idade, o corte é desejável

?- idade(X,Y),!, Y > 18.

X = joao ,

Y = 36

obs.: Neste último caso, quer-se mais de uma resposta, para diferentes valores de X, contudo, o cut impediu.

Corte na submeta

temRG(X) :- homem(X), idade(X,Y),!,Y>18.

mulher(susana).

mulher(roberta).

homem(joao).

homem(ricardo).

homem(marcelo).

idade(joao, 36).

idade(susana, 15).

idade(roberta, 31).

idade(ricardo, 22).

idade(marcelo, 17).

?- temRG(X).

X = joao

?-

**Obs.: reduz-se o espaço de busca, mas
não atende o que se pretende.**

Isolando o efeito do corte

temRG(X) :- homem(X), idade(X,Y), Y>18.

mulher(susana).

mulher(roberta).

homem(joao).

homem(ricardo).

homem(marcelo).

idade(joao, 36) :- !.

idade(susana, 15) :- !.

idade(roberta, 31) :- !.

idade(ricardo, 22) :- !.

idade(marcelo, 17) :- !.

?- temRG(X).

X = joao ;

X = ricardo ;

false.

obs.: agora OK, mas..

?- idade(X,Y).

X = joao ,

Y = 36

?-

**Obs.: o corte no predicado idade/2
não permite múltiplas respostas.**

Isolando o efeito do corte

temRG(X) :- homem(X), souma(idade(X,Y)), Y>18.

mulher(susana).

mulher(roberta).

homem(joao).

homem(ricardo).

homem(marcelo).

idade(joao, 36).

idade(susana, 15).

idade(roberta, 31).

idade(ricardo, 22).

idade(marcelo, 17).

souma(P) :- P, !.

?- temRG(X).

X = joao ;

X = ricardo ;

false.

?- idade(X,Y).

X = joao ,

Y = 36 ;

X = susana ,

Y = 15 ;

X = roberta ,

Y = 31 ;

X = ricardo ,

Y = 22 ;

X = marcelo ,

Y = 17

Isolando o efeito do corte

O predicado `souma(P)` enseja um uso adequado do `cut`, quando a pretensão é eliminar buscas desnecessárias, ou restringir a uma só resposta.

- Não precisa usar `cut` diretamente na consulta
- Não precisa usar `cut` sobre cláusulas na base.
- `souma(P)` falha sob retrocesso, mas não impende que variáveis em predicados anteriores, no mesmo corpo, sejam revaloradas.
- É assim uma solução mais limpa para o `cut`.