

IKI30820 – Logic Programming Cut Slide 05

Ari Saptawijaya (slides) Adila A. Krisnadhi (\LaTeX adaptation)

Fakultas Ilmu Komputer
Universitas Indonesia

2009/2010 • Semester Gasal

1 What is Cut?

2 Examples

Motivation

- Consider the following program

```
grade(X, 'A') :- X >= 90.
```

```
grade(X, 'B') :- 80 <= X, X < 90.
```

```
grade(X, 'C') :- 65 <= X, X < 80.
```

```
grade(X, 'Fail') :- X < 65.
```

- Query:

```
?- grade(92, Grade), Grade == 'C'.
```

- Is the query satisfiable?

- Prolog will automatically backtrack if this is necessary to satisfying a goal.
- Automatic backtracking is a useful programming concept
 - ▶ It relieves the programmer of the burden of backtracking explicitly
- On the other hand, uncontrolled backtracking may cause inefficiency in a program
- We need a way to control backtrack.

Controlling backtracking with cut

- 'Cut' is written as the character '!'
- Second version of previous program (now with cut to control backtrack)

```
grade(X, 'A') :- X >= 90, !.  
grade(X, 'B') :- 80 =< X, X < 90, !.  
grade(X, 'C') :- 65 =< X, X < 80, !.  
grade(X, 'Fail') :- X < 65.
```

- Does this really work in controlling backtracking?

The meaning of cut

- The “parent goal” is the goal that matched the head of the clause containing the cut.
- When the cut is encountered as a goal
 - ▶ it succeeds immediately
 - ▶ it commits the system to all choices made between the time the parent goal was invoked and the time the cut was encountered
- All the remaining alternatives between the parent goal and the cut are discarded

- Third version of previous program:

```
grade(X, 'A') :- X>=90, !.  
grade(X, 'B') :- X>=80, !.  
grade(X, 'C') :- X>=65, !.  
grade(X, 'Fail').
```

- Query:

```
grade(95, X).
```

- Compare it with:

```
grade(X, 'A') :- X>=90.  
grade(X, 'B') :- X>=80.  
grade(X, 'C') :- X>=65.  
grade(X, 'Fail').
```

- 1 What is Cut?
- 2 Examples

Example

```
p(X) :- a(X).           % rule1
p(X) :- b(X),c(X),d(X),e(X). % rule2
p(X) :- f(X).          % rule3
```

a(1).

b(1).

c(1).

b(2).

c(2).

d(2).

e(2).

f(3).

Query:

?- p(X).

.. more example

```
s(X,Y) :- q(X,Y).
```

```
s(0,0).
```

```
q(X,Y) :- i(X),j(Y). % clause3
```

```
i(1).
```

```
i(2).
```

```
j(1).
```

```
j(2).
```

```
j(3).
```

```
Query: ?- s(X,Y).
```

.. another example

- Variant 1:

```
wizard(harry) :- !.  
wizard(ron) .  
wizard(hermione) .
```

- Variant 2:

```
wizard(harry) .  
wizard(ron) :- !.  
wizard(hermione) .
```

- Variant 3:

```
wizard(harry) .  
wizard(ron) .  
wizard(hermione) :- !.
```

Advantages of Cut ...

- Often improve the efficiency of the program
 - ▶ Explicitly tell Prolog: do not try other alternatives as they are bound to fail.
- Specify mutually exclusive rules (e.g. version 3 of “grade”).

Using cut for membership

```
member2 (X, [X|_]) :- !.  
member2 (X, [_|Xs]) :- member2 (X, Xs) .
```

Using cut to define set

```
set([], []).  
set([X|Xs], Ys) :- member(X, Xs), !, set(Xs, Ys).  
set([X|Xs], [X|Ys]) :- set(Xs, Ys).
```