Another example

```
foo(X) :- a(X).
foo(X) :- bar(X).

a(X) :- b(X), c(X), !, d(X), e(X).
a(X) :- p(X).

b(1).
b(2).
c(3).
b(4).
c(4).
d(4).
b(5).
c(5).
d(5).
e(5).

p(22).

bar(44).
```

Trace with cut

```
| ?- foo(X).
   (0) Call: foo(_47) ?
   (1) Call: a(_64) ?
   (2) Call: b(_69) ?
   (2) Exit: b(1) ?
   (3) Call: c(1) ?
   (3) Fail: c(1) ?
   (2) Redo: b(1) ?
   (2) Exit: b(2) ?
   (4) Call: c(2) ?
   (4) Fail: c(2) ?
   (2) Redo: b(2) ?
   (2) Exit: b(4) ?
   (5) Call: c(4) ?
   (5) Exit: c(4) ?
   (6) Call: d(4) ?
   (6) Exit: d(4) ?
   (7) Call: e(4) ?
   (7) Fail: e(4) ?
   (6) Redo: d(4) ?
   (6) Fail: d(4) ?
   (1) Fail: a(_64) ?
   (8) Call: bar(_64) ?
   (8) Exit: bar(44) ?
   (0) Exit: foo(44) ?

X = 44;
```

continued ...

```
   (0) Redo: foo(44) ?
   (8) Redo: bar(44) ?
   (8) Fail: bar(_64) ?
   (0) Fail: foo(_47) ?

no
```

Without cut

```
| ?- foo(X).

X = 5;

X = 22;

X = 44;

no
```