Cut (5A)

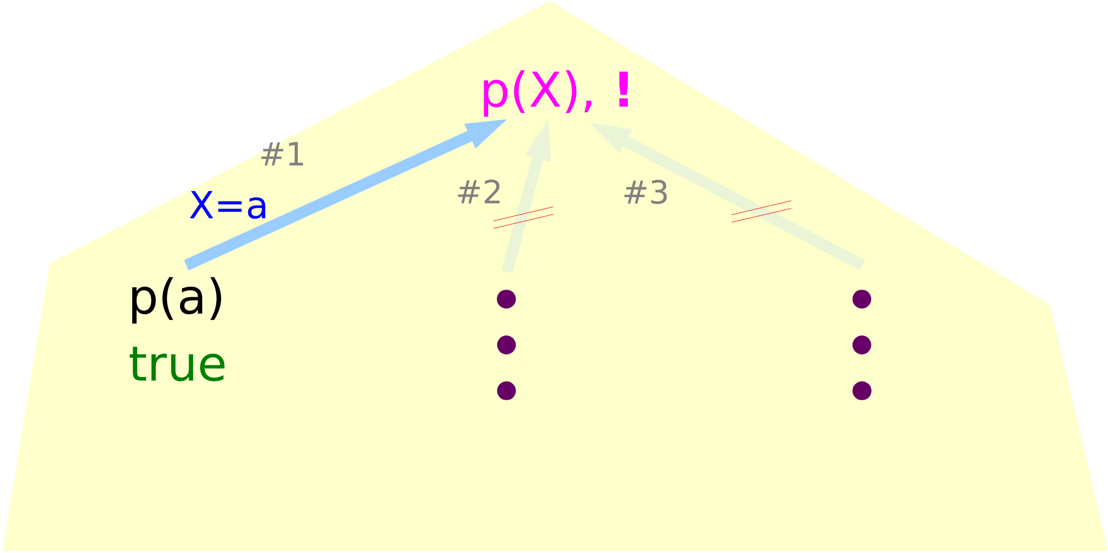
Young W. Lim
2/13/14

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice/OpenOffice.

Cut Examples (1)

```
p(a).                    /* #1 */
p(X) :- q(X), r(X).      /* #2 */
p(X) :- u(X).            /* #3 */
q(X) :- s(X).            /* #4 */
r(a).                    /* #5 */
r(b).                    /* #6 */
s(a).                    /* #7 */
s(b).                    /* #8 */
s(c).                    /* #9 */
u(d).                    /* #10 */
```



```
?- p(X), !.
X=a ;
no
```

always true

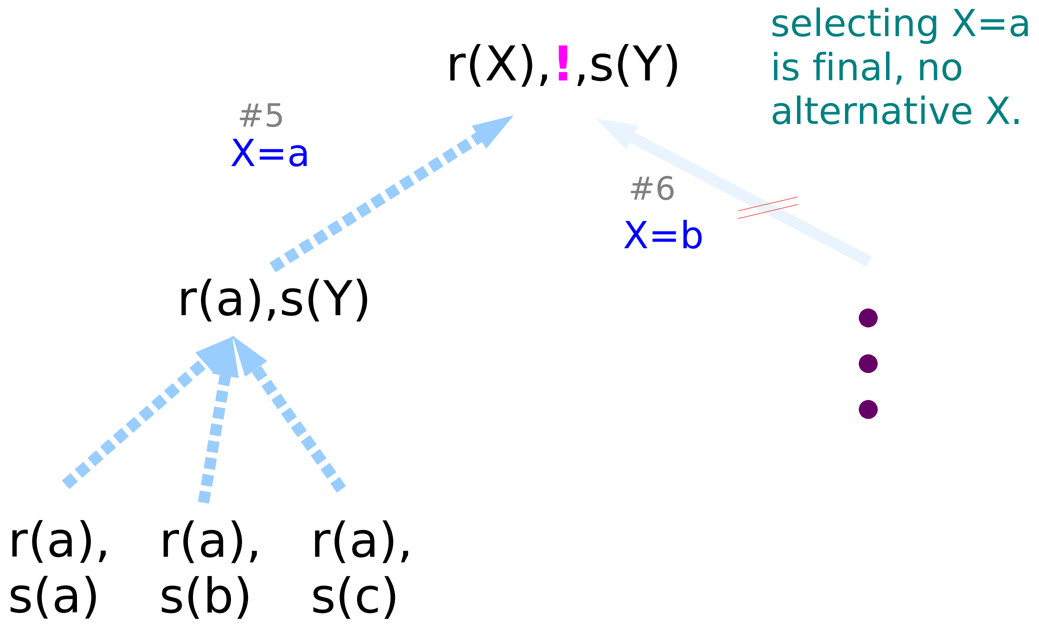
seeking
alternative fails

The derivation tree is **trimmed**
of all other choices
at the point where the cut was
introduced as one of the
sequence of subgoals

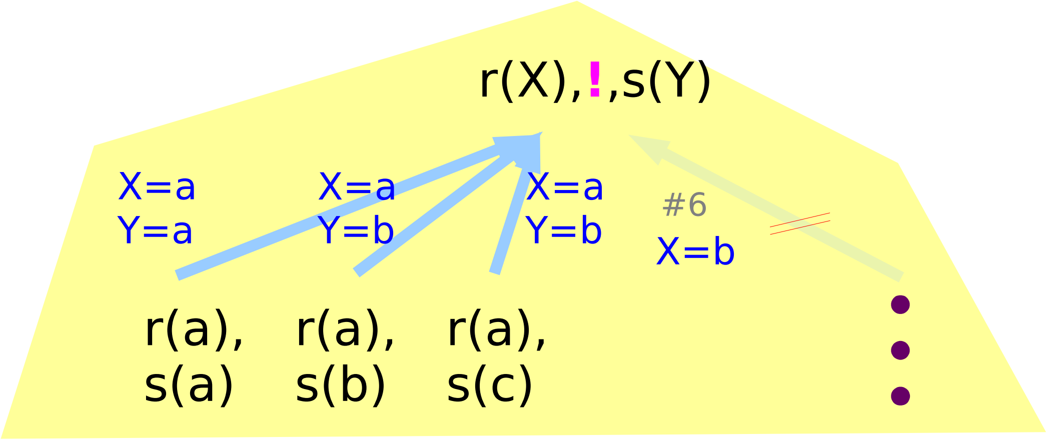
Cut Examples (2)

```
p(a).                    /* #1 */
p(X) :- q(X), r(X).      /* #2 */
p(X) :- u(X).            /* #3 */
q(X) :- s(X).            /* #4 */
r(a).                    /* #5 */
r(b).                    /* #6 */
s(a).                    /* #7 */
s(b).                    /* #8 */
s(c).                    /* #9 */
u(d).                    /* #10 */
```

```
?- r(X),!,s(Y).
X=a Y=a ;
X=a Y=b ;
X=a Y=c ;
no
```

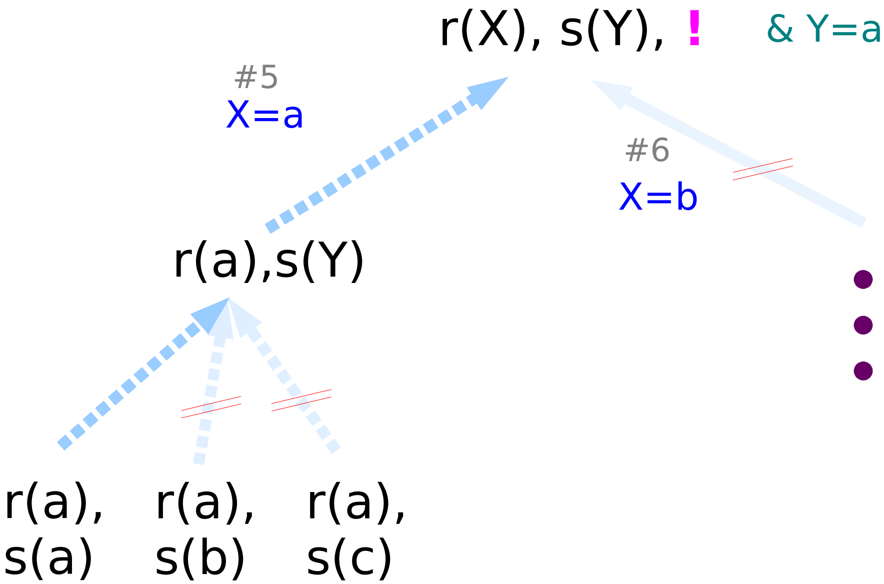


selecting X=a is final, no alternative X.



Cut Examples (3)

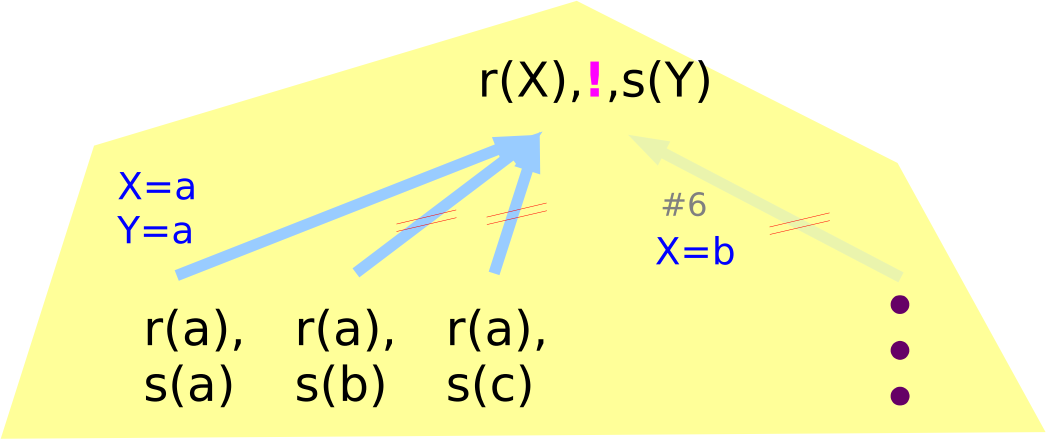
```
p(a).                    /* #1 */
p(X) :- q(X), r(X).      /* #2 */
p(X) :- u(X).            /* #3 */
q(X) :- s(X).            /* #4 */
r(a).                    /* #5 */
r(b).                    /* #6 */
s(a).                    /* #7 */
s(b).                    /* #8 */
s(c).                    /* #9 */
u(d).                    /* #10 */
```



selecting X=a
& Y=a are final

```
?- r(X), s(Y), !.
X=a Y=a ;
no
```

seeking
alternative fails



Cut as a Called Subgoal (1)

three alternatives for **color**...

```
red(a). black(b).
color(P,red) :- red(P), !.
color(P,black) :- black(P), !.
color(P,unknown).
```

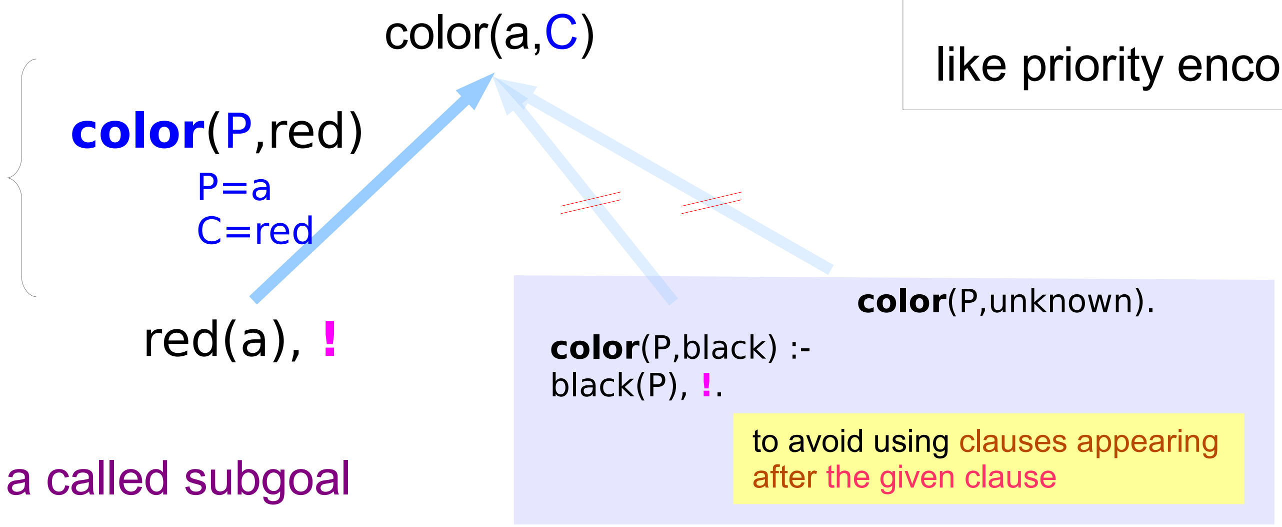
to avoid using clauses appearing after the given clause in the program.

```
?- color(a,C)
C = red
```

```
if (A) { ... }
else if (B) { ...}
else { C }
```

like priority encoder

a cut occurs in the body of the program



color(a,C)

color(P,red)

P=a
C=red

red(a), !

color(P,unknown).

color(P,black) :- black(P), !.

to avoid using clauses appearing after the given clause

the cut appears as a called subgoal

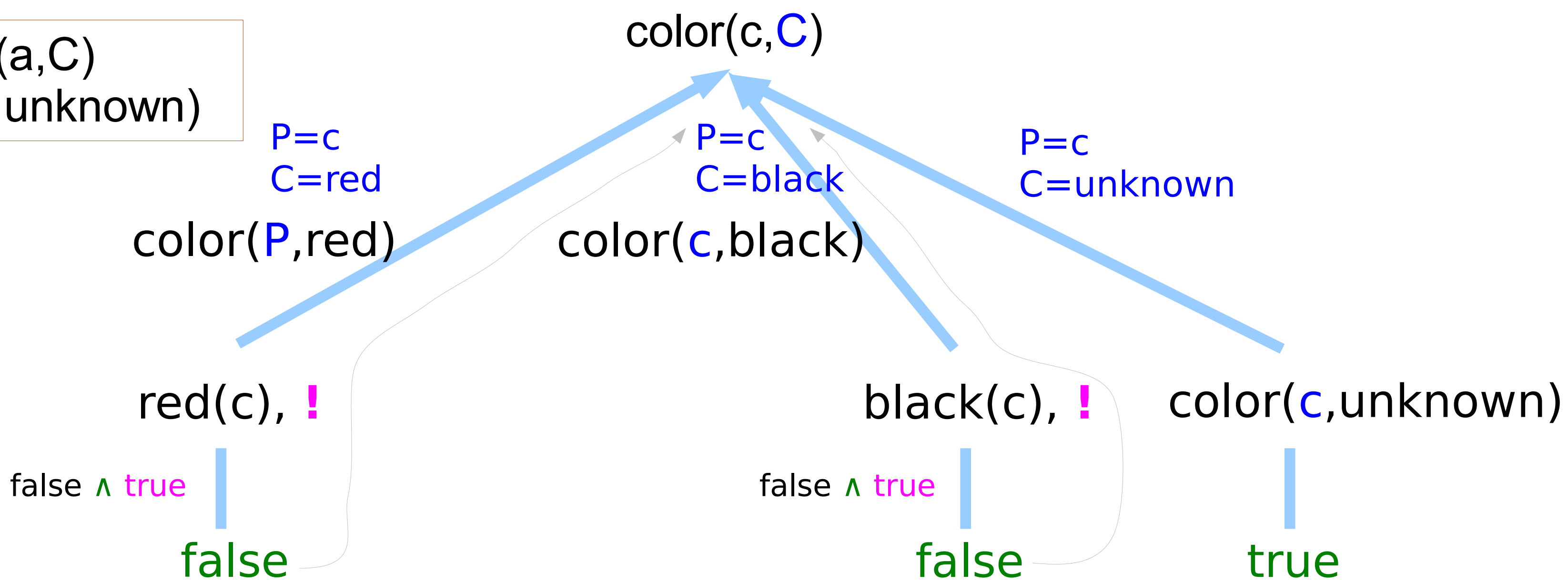
Cut as a Called Subgoal (2)

three alternatives for **color**...

```
red(a). black(b).
color(P,red) :- red(P), !.
color(P,black) :- black(P), !.
color(P,unknown).
```

to avoid using clauses appearing after the given clause in the program.

?- color(a,C)
color(c, unknown)



color(c,C)

P=c
C=red

P=c
C=black

P=c
C=unknown

color(P,red)

color(c,black)

red(c), !

black(c), !

color(c,unknown)

false ∧ true

false

false ∧ true

false

true

the 1st rule cannot be matched

neither the 2nd rule can be matched

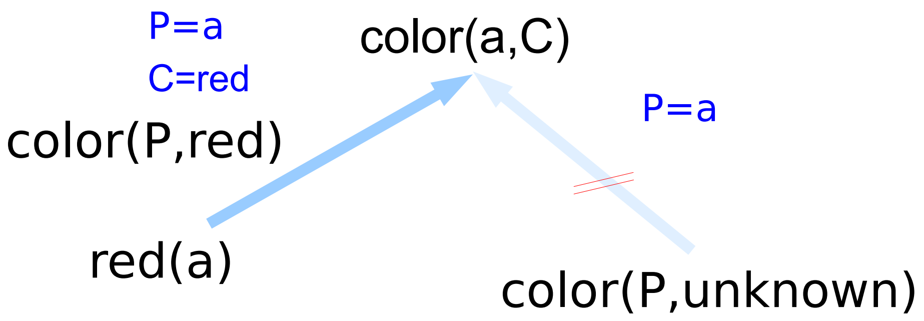
the 3rd rule can be matched

Cut as a Called Subgoal (3)

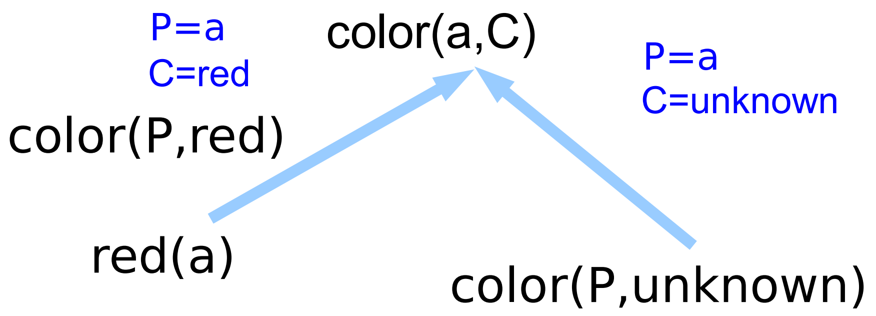
a procedural device for controlling goal satisfaction.

```
red(a). black(b).
color(P,red) :- red(P), !.
color(P,black) :- black(P), !.
color(P,unknown).
```

```
red(a). black(b).
color(P,red) :- red(P).
color(P,black) :- black(P).
color(P,unknown).
```



P=a
C=red
color(a,C)
P=a
color(P,red)
red(a)
color(P,unknown)



P=a
C=red
color(a,C)
P=a
C=unknown
color(P,red)
red(a)
color(P,unknown)

The cut is used in the body of a given clause so as to avoid using clauses appearing after the given clause in the program.

```
?- color(a,C).
C = red.
```

the following program clause tree says that 'color(a,unknown)' should be a consequence of the program:

```
?- color(a,C).
C = red;
C = unknown.
```

Cut

! : cut, the predefined predicate

can be anywhere in a rule’s body
can be a part of a sequence of subgoals in a query

The subgoal **!** is always succeed

backtracking into subgoals
is not possible anymore, which are
placed before the cut
inside the same rule body

Whenever a cut is encountered in a rule’s body,
all choices made between
the time that rule’s head has been matched
with the parent goal
and the time the cut is passed
are ***final***, i.e. any choicepoints are being discarded.

A Cut Example

```
bride(Girl) :-
      beautiful(Girl), !,
      intelligent(Girl).
```

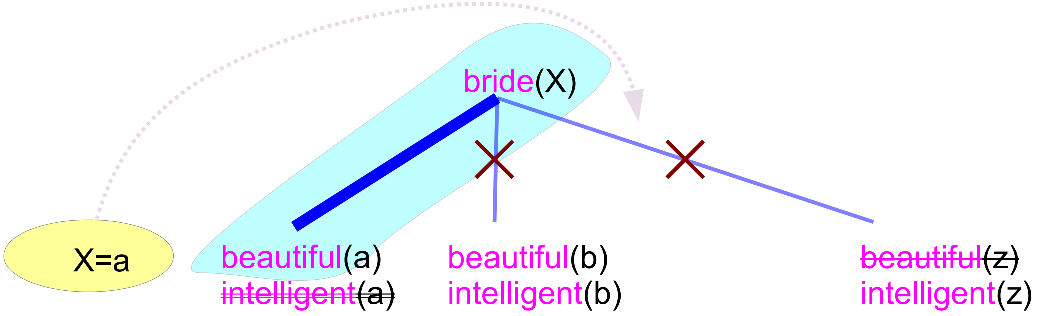
Bound variable before the cut are ***final***
Cannot change the first choice

Considered as True
regardless of the possible failure
of the subgoals **after the cut**

```
?- bride(X).
No
```

the subgoals before the cut
must match (true)

The first choice X=a is final
cannot try other alternatives



No match after
the cut
Result: **No**

Without the cut
Result: **X=b**

References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, “Lecture Notes : Introduction to Prolog Programming”
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html