Cuts (Prolog book 4.3–4.4)

When hit the cut, removes ability to backtrack to other solutions predicates in clause prior to backtrack.

Yet another Example

```
main :- a.
main.
a :- b(B,C), !, c(C,D), d(D,B).
a :- e(B,C), f(C,D), g(D,B).
a :- h(B,B).
```

What is the effect of the above cut?

Depth First Search Example

```
path(Y,[Y|Path],[Y|Path]).
path(Y,[Top|Rest],NewPath) :-
    connectedto(Top,Z), + member(Z,Rest), !,
    path(Z,Y,[Z,Top|Path],NewPath), !.
```

?- path(16,[1],Path).

Another Example

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

What is the effect of the above cut?

- Locks you into current solution.
- frowned alternative search for current goal.
- Backtracking point where Prolog can pursue different proofs.

CSE560 Class Cut: 2

© P. Heeman, 2010

CSE560 Class Cut: 1

© P. Heeman, 2010

CSE560 Class Cut: 3

© P. Heeman, 2010
Conditions

Red cut:
- How can we change it to work with '
- How can we change it to work with '+'
- How can we change it to work with '-'

Will it work right if we backtrace through code for 'print([])?

Problem: print a message whether argument is an empty list or
- print([]) :- write('argument is an empty list').
- print([|]) :- write('argument is a non-empty list').
- print(A) :- atom(A), write('argument is an atom').
- print([]) :- write('argument is other').

Green cut:
- if you know you are not throwing away possible solutions
  - adding cut doesn't change behavior of program
  - just, just makes it more efficient
- doesn't waste time to do fruitless check of alternatives
- doesn't store choice points

print([]) :- !,
    write('argument is an empty list').
print([|]) :- !,
    write('argument is a non-empty list').
print(A) :- atom(A), !,
    write('argument is an atom').

And another example:
Problem: print a message about the type of argument

main :-
    print([]),
    fail.

How can we make code more efficient?

And another example:
• Strings
- List of ASCII characters

• Printing strings
- Use `put` to print a character
- How do you print a string?