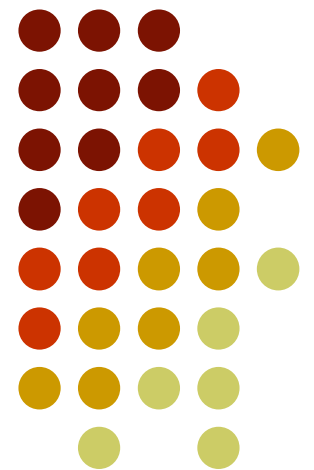


# Introduction to Prolog (More) Examples

CS181: Programming Languages





# Topics:

- Append()
- Build a list using [H|T]
- Reverse elements of a list



# Append

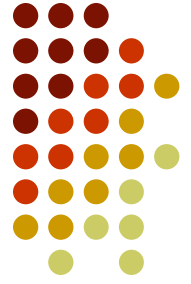
- You can use `append()` to concatenate two lists:

?- `append([a, b, c], [d, e, f], Result)`.

`Result = [a, b, c, d, e, f]`

- Or to divide a list in all possible ways:

?- `append(List1, List2, [a, b, c])`.



# Append

List1 = []  
List2 = [a, b, c] ;

List1 = [a]  
List2 = [b, c] ;

List1 = [a, b]  
List2 = [c] ;

List1 = [a, b, c]  
List2 = [] ;

No



# Building a list

- Build a list of all natural numbers less than the argument X:
  - **Base Case:** If the number entered is just 0, then the answer will be just []:

```
build_list(0, L) :- L=[].
```

- **Recursive Case:** If we're dealing with a number, say N, then we can assume that we know how to collect all the numbers up to N-1 (thanks to recursion) so we just need to know how to add the current element:

```
build_list(N, L) :- N>0, N1 is N-1, build_list(N1,T), L=[N|T].
```



# Building a list

- Or, in slightly more compact way:

```
build_list(0,[ ]).
```

```
build_list(N, [N|T]) :- N>0, N1 is N-1, build_list(N1, T).
```

- Note that here `[H|T]` was used to append a new head to the old list (tail of the new list).



# Reversing a list

- Reverse the elements of the given list:
  - **Base Case:** If the list entered is [], then the answer will be just []:

`reverse_l([], []).`

- **Recursive Case:** If the list is not empty, then take the head from the given list and append it to the end of the solution list. Of course, do this recursively until you hit rock bottom (the empty list), and then move up, adding heads along the way:

`reverse_l([H|T], L2) :- reverse_l(T, NT), append(NT, [H], L2).`



## References:

- Clocksin, W.F., and Mellish C.S. *Programming in Prolog*. 4th edition. New York: Springer-Verlag. 1994.
- Aaby, A. *Prolog Tutorial*. Walla Walla College. 1997. On line.  
[http://cs.wwc.edu/~cs\\_dept/KU/PR/Prolog.html](http://cs.wwc.edu/~cs_dept/KU/PR/Prolog.html)
- Power, J. *Prolog Tutorial*. On line. National University of Ireland, Maynooth.