

Exercise sheet 2a

(Submit as a plain text email message with subject "Homework 2a" to dm@ling.osu.edu on Wednesday, Feb. 1)

Provide Prolog definitions for the following relations. Define them using the minimal means necessary—in particular, there is no need to make use of other relations defined in class or predefined predicates.

Thoroughly test your predicates before handing them in!

1. `last/2`: a two place relation which takes a list as first argument and returns the last element of that list (if there is one) as second argument; i.e., `last(+List, -Last-List-element)`

Example queries:

- `?- last([a,b,c,d],X). => X=d`
- `?- last([a,b,c],X). => X=c`
- `?- last([],X). => no`

2. `interleave/3`: a three place relation interleaving the elements from the first with the elements of the second list (of same length).

Example queries:

- `?- interleave([a,b,c],[1,2,3],X). => X=[a,1,b,2,c,3]`
- `?- interleave([],[],X). => X=[]`

3. `firstLastSwap/2`: a two place relation which takes a list and returns the same list with one difference: the first list element and the last list element are exchanged; i.e., `firstLastSwap(+List, -List-With-First-Last-Swapped)`

Example queries:

- `?- firstLastSwap([a,b,c,d],X). => X=[d,b,c,a]`
- `?- firstLastSwap([a,b,c],X). => X=[c,b,a]`
- `?- firstLastSwap([a,c],X). => X=[c,a]`
- `?- firstLastSwap([],X). => no`

4. `next_element/3`: a three place relation which takes a shorter list and a longer list as arguments, checks whether the first list is contained in the second, and returns the element immediately following the match (backtracking if there are multiple possible matches).

Example queries:

- `?- next_element([b],[k,a,b,c,d,b,a,e,f],X). ⇒ X=a; X=d`
- `?- next_element([a,b,c],[k,a,b,c,d,e,f],X). ⇒ X=d`
- `?- next_element([a,b,c],[k,a,b,c,d],X). ⇒ X=d`
- `?- next_element([a,b,c],[k,a,b,c,d,e,a,b,c,f,g],X). ⇒ X=d; X=f`
- `?- next_element([a,b,c],[b,a,c,b,d],X). ⇒ no`
- `?- next_element([a,b,c],[a,b,c],X). ⇒ no`
- `?- next_element([a,b,c],[],X). ⇒ no`
- `?- next_element([],Y,X). ⇒ no`

5. `preceding_element/3`: a three place relation which takes a shorter list and a longer list as arguments, checks whether the first list is contained in the second, and returns the element immediately preceding the match (backtracking if there are multiple possible matches)..

Example queries:

- `?- preceding_element([a,b,c,d],[k,a,b,c,d,e,f],X). ⇒ X=k`
- `?- preceding_element([a,b,c,d],[d,a,q,a,b,c,d,e,f],X). ⇒ X=q`
- `?- preceding_element([a,b,c],[b,a,c,b,d],X). ⇒ no`
- `?- preceding_element([a,b,c],[a,b,c],X). ⇒ no`
- `?- preceding_element([a,b,c],[],X). ⇒ no`
- `?- preceding_element([],Y,X). ⇒ no`