# Knowledge representation

Tutorial 1

20 September 2013

## Practical information


- Teaching assistant: Stéphane Lens
- `lens @ montefiore.ulg.ac.be`, 04/3662619, office I82a.
- `http://www.montefiore.ulg.ac.be/~lens`


## Operations on Lists

**1.** Define a predicate `sublist(-Ss, +Ls)` that succeeds if the list `Ss` is a sublist of the list `Ls`. Next draw a search tree for the query `sublist(Xs, [a, b])`.

**2.** Define a predicate `my_reverse(+As, -Bs)` that succeeds when the elements of the list `Bs` are in reverse order compared to the list `As`. Next draw a search tree for the queries `my_reverse([a, b], Xs)` and `my_reverse(Xs, [b, a])`.

## Proposed exercise

**3.** Define a predicate `my_flatten(+Ls, -Zs)` that succeeds if the list `Zs` is a non-nested version of the list `Ls`.

```
?- my_flatten([a, [b, [c, d], e]], X).

    X = [a, b, c, d, e] ;
    false.
```