```
function subsets (xs) {
    return accumulate ((ele, ss) => append(ss, map (s => pair(ele, s),
                                                    ss)),
                      list (null),
                      xs);
}
```

How to find all possible subsets?
I look at every element. I either want that element in my subset, or I don't want it.
If I want that element, then I add that element in the subset, and it forms a new subset.

The idea is to build your subsets from the ground up, by looking at each and every element and choosing if you want that element or not. If you want it, then it forms a new subset, if not, then what you have is still a valid subset. So you want both cases:

- the subset with the new element, and
- the subset without the new element

Example:

subsets (list (1, 2, 3))

		current subsets
initially :	list (1, 2, 3)	[] → no elements
look at 3 :	list (1, 2, 3) ↑	[], [3] → both are valid subsets!
look at 2 :	list (1, 2, 3) ↑	[], [3], without new element [2], [2,3] with new element
look at 1 :	list (1, 2, 3) ↑	[], [3], [2], [2,3], without new element [1], [1,3], [1,2], [1,2,3] with new element