

## Exercises Week 9

Study w9-1x2.pdf or w9-2x2.pdf and answer the following questions.

- [1+1 POINTS] Implement (a) `ordered` and (b) `slowsort`.
- [1+1+1+1 POINTS] Implement (a) `union` ( $\cup$ ), (b) `inter` ( $\cap$ ), (c) `equal` ( $=$ ), and (d) `powerset` ( $\mathcal{P}$ ).

```
# union [1;2;3] [2;3;4];;
- : int list = [1; 2; 3; 4]
# inter [1;2;3] [2;3;4];;
- : int list = [2; 3]
# equal [1;2;3] [2;3;1];;
- : bool = true
# powerset [1; 2; 3];;
- : int list list =
  [[]; [3]; [2]; [2; 3]; [1]; [1; 3]; [1; 2]; [1; 2; 3]]
```

Hint: The powerset function  $\mathcal{P}(X)$  on sets has the following properties:  
 $\mathcal{P}(\emptyset) = \{\emptyset\}$  and  $\mathcal{P}(\{x\} \cup X) = \mathcal{P}(X) \cup \{\{x\} \cup Y \mid Y \in \mathcal{P}(X)\}$ .

- [1+1+1+1 POINTS] Implement (a) `succ`, (b) `reachable_from` (c) `pred`, and (d) `reachable_to`.
- [1+2 POINTS] Consider the following code:

```
type expr =
  | Const of int
  | Add of expr * expr
  | Mul of expr * expr
type instr = Push | Addint | Mulint | Value of int
exception Bad_code
```

- Implement a bytecode compiler `compile`, which translates `expr` to `instr list`.
- Implement `interpret`, a bytecode interpreter for `instr list`.

```
# let code = compile (Mul (Const 10, Add (Const 20, Const 30)));;
val code : instr list =
  [Push; Value 10; Push; Value 20; Push; Value 30; Addint; Mulint]
# interpret code;;
- : int = 500
```

Hint: `List.nth [x0;x1;...;xm] n = xn`

Submit your `MatrNr.ml` before 23:59 on **December 14**.