Exercise 2: *Free Variables* *4+4 Points*

a) A recursive definition of $free : Expr \to 2^{Vars}$ is defined using the auxiliary function $f\colon Expr \times 2^{Vars} \to 2^{Vars}$, which is defined as follows. Then, $free(e) = f(e, \emptyset)$.

$$f(e,\Omega) = \begin{cases} \emptyset & e \equiv b \\ \{x\} \Leftrightarrow x \notin \Omega \, else \, \emptyset & e \equiv x \\ f(e_1,\Omega) & e \equiv (\Box_1 \, e_1) \\ f(e_1,\Omega) \cup f(e_2,\Omega) & e \equiv (e_1 \, \Box_2 \, e_2) \\ f(e_1,\Omega) \cup f(e_2,\Omega) \cup f(e_3,\Omega) & e \equiv (\textbf{if} \, e_1 \, \textbf{then} \, e_2 \, \textbf{else} \, e_3) \\ f(e',\Omega) \cup \left(\bigcup_{i=1}^{k} f(e_i,\Omega)\right) & e \equiv (e' \, e_1 \ldots e_k) \\ f(e',\Omega \cup \{x_1,\ldots,x_k\}) & e \equiv (\textbf{fun} \, x_1,\ldots,x_k \to e') \\ f(e,\Omega \cup \{x_1\}) & e \equiv (\textbf{let} \, x_1 = e_1 \, \textbf{in} \, e_k) \\ \bigcup_{i=1}^{k} \left[f\left(e_i, \Omega \cup \left(\bigcup_{j=1}^{k-1} \{x_j\}\right)\right)\right] & e \equiv (\textbf{let rec} \, x_1 = e_1 \, \textbf{and} \ldots \, x_{k-1} = e_{k-1} \, \textbf{in} \, e_k) \end{cases}$$

As can be seen from the above, Ω is used to accumulate any “local” variables. Hence the set of free variables can be determined from the set of those variables in an expression that are not contained in Ω.

However, attention must be paid to the clear distinction of the variables’ identifiers. For example, consider the following expression:

```
let x1 = 5
in let x1 = x1
in x1
```

Clearly, the outer `x1` does *not* determine the same variable as the inner `x1`, although at first sight both are denoted by identical *literals*. Thus, when adding variables to Ω during any recursive descent through $f(e,\Omega)$, some meaningful way has to be established so as to be able to clearly distinguish any variables despite of possible identical identifiers. This could, for instance, be achieved by replacing the identifiers symbols with unique symbols before evaluating $free(e)$.

b) The resulting sets of free variables are:

- {`y`}
- {`z`}
- {`y`, `z`}
- {`f`, `x`, `y`, `z`}