

**Exercise 1** Suppose we are given a predicate  $\text{flight}(\text{From}, \text{To}, \text{Time}, \text{Price})$  containing information about direct flights including the starting airport, the destination, the flight time, and the price of a ticket. Write a Prolog program computing a predicate  $\text{travel}(\text{From}, \text{To}, \text{Stops}, \text{Time}, \text{Price})$  indicating all possibilities to travel from one city to another using one or several flights.

**Exercise 2** Write a Prolog predicate  $\text{fib}(N, X)$  computing the Fibonacci sequence. Evaluate  $\text{fib}(3, X)$  and  $\text{fib}(N, 5)$ .

**Exercise 3** Write Prolog definitions of the following predicates.

|                                 |  |
|---------------------------------|--|
| $\text{length}(\text{List}, N)$ | $N$ is the length of $\text{List}$ .   |
| $\text{append}(X, Y, Z)$        | $Z$ is the concatenation of the lists $X$ and $Y$ .                            |
| $\text{reverse}(X, Y)$          | $Y$ is the reverse of the list $X$ .   |
| $\text{map}(X, Y)$              | maps a list $X = [X_1, \dots, X_n]$ to $Y = [f(X_1), \dots, f(X_n)]$ .         |
| $\text{fold\_left}(X, Y, Z)$    | maps $Y = [Y_1, \dots, Y_n]$ to $Z = f(\dots f(f(X, Y_1), Y_2) \dots, Y_n)$ .  |
| $\text{fold\_right}(X, Y, Z)$   | maps $Y = [Y_1, \dots, Y_n]$ to $Z = f(Y_1, f(Y_2, \dots, f(Y_n, X) \dots))$ . |

The Prolog notation for lists is as follows:

$[]$   $[X, Y, Z]$   $[X|Y]$   $[X, Y|Z]$ .

**Exercise 4** Write a naive sort function

$\text{naive\_sort}(X, Y) :- \text{permute}(X, Y), \text{sorted}(Y)$ .

by implementing the relations

|                          |   |
|--------------------------|---|
| $\text{sorted}(X)$       | checks that the list $X$ is sorted.   |
| $\text{insert}(X, Y, Z)$ | if the list $Z$ is obtained from $Y$ by inserting $X$ at an arbitrary position. |
| $\text{permute}(X, Y)$   | if the list $Y$ is a permutation of $X$ .                                       |

Implement merge sort using a relation

|                         |  |
|-------------------------|--|
| $\text{merge}(X, Y, Z)$ | merges two sorted lists $X$ and $Y$ into $Z$ .   |
| $\text{split}(X, Y, Z)$ | splits the list $X$ into two lists $Y$ and $Z$ . |

**Exercise 5** We consider directed graphs of the form  $\langle V, E \rangle$ . Express the following relation in relational algebra.

- $x$  and  $y$  are not connected by an edge.
- The edge  $\langle x, y \rangle$  is part of a triangle.
- $x$  has at least two neighbours.
- Every neighbour of  $x$  is also a neighbour of  $y$ .

**Exercise 6** Evaluate the following Datalog program on the tree  $\langle V, E, P \rangle$  to the right.

$U \leftarrow S(x, y) \wedge W(x) \wedge W(y)$   
 $W(x) \leftarrow P(x)$   
 $W(x) \leftarrow E(x, y) \wedge W(y)$   
 $S(x, y) \leftarrow E(z, x) \wedge E(z, y) \wedge x \neq y$   
 $R(x, y) \leftarrow P(x) \wedge x = y$   
 $R(x, y) \leftarrow E(x, z) \wedge R(z, y)$   
 $R(x, y) \leftarrow R(x, z) \wedge E(z, y)$

