

---

**Compilerbau**

---

## Blatt 2

Abgabe: 31.10.2001

1. [5 Punkte] Implementieren Sie gewöhnliche Mengen als Datentyp, der mit dem Typ seiner Elemente parametrisiert ist. Eine Menge soll dabei durch ihre charakteristische Funktion repräsentiert werden. Realisieren Sie die Operationen Konstruktion einer einelementigen Menge, Element-Test, Vereinigung, Durchschnitt, Differenz und Komplement.
2. [5 Punkte] Geben Sie Definitionen von `map` und `length` unter Verwendung von `List.fold_right` an, welche selbst nicht rekursiv sind.
3. [5 Punkte] Schreiben Sie eine Funktion

```
partition_list : ('a -> bool) -> 'a list -> ('a list * 'a list)
```

so daß `partition_list f, l` ein Paar  $(l_{\text{true}}, l_{\text{false}})$  liefert, in dem  $l_{\text{true}}$  genau die Elemente  $x$  von  $l$  enthält mit  $fx = \text{true}$  und  $l_{\text{false}}$  die Elemente  $x$  mit  $fx = \text{false}$ .

4. [5 Punkte] Definieren Sie den Datentyp `Rational` für rationale Zahlen mit den arithmetischen Operationen Addition, Subtraktion, Multiplikation und Division. Benutzen Sie eine eindeutige Darstellung.
5. [10 Punkte] Schreiben Sie eine Caml-Funktion

```
lcs : string -> string -> string
```

so daß `lcs x y` die *längste gemeinsame Teilfolge* zweier Zeichenketten  $x$  und  $y$  liefert. (So ist `tie` die längste gemeinsame Teilfolge von `striped` und `tiger`.)

Schreiben Sie außerdem eine Funktion

```
diff : string -> string -> int
```

so daß `diff x y` die *Distanz* zweier Zeichenketten berechnet, also die minimale Anzahl von Einfügungen und Löschungen, die erforderlich sind, um  $x$  in  $y$  zu überführen.