Arbeitsblatt: Higher Order Functions III

In den letzten beiden Arbeitsblättern haben Sie map und filter neu entdeckt. In diesem Arbeitsblatt suchen wir nach einem neuen, viel mächtigeren[1] Rekursionsmuster.

Wir untersuchen dazu vier Funktionen:

```
sum []         = 0
sum (x:xs)     = x + sum xs

product []     = 1
product (x:xs) = x * product xs

or []          = False
or (x:xs)      = x || or xs

and []         = True
and (x:xs)     = x && and xs
```

a) Vergleichen Sie die vier Funktionen. Markieren Sie im Code die Gemeinsamkeiten und in welchen Teilen sie sich unterscheiden.
Hinweis: Die Namen der Funktionen sind nicht relevant, es geht um die Struktur der Funktionen.

b) Definieren Sie die Funktion aggregate. Sie soll wiederum die Gemeinsamkeiten implementieren und über die Unterschiede abstrahieren.
Hinweis: Schreiben Sie zuerst die Typsignatur der Funktion.

c) Implementieren Sie die Funktionen basierend auf Ihrer neuen Funktion aggregate:

[1] So mächtig, dass man damit die Funktionen filter und map implementieren kann.