

Informatik A, WS 2016/17 — 12. Übungsblatt

Abgabe bis Freitag, 20. Januar 2017, 12:00 Uhr, in die Fächer der Tutor/inn/en

67. Natürliche Zahlen, Programmieraufgabe, 10 Punkte

Man kann sich in HASKELL seine eigenen natürlichen Zahlen als algebraischen Datentyp definieren:

```
data Nat = Zero | Succ Nat
```

`Zero` steht für 0 und `Succ` für Nachfolger (*successor*). Die Zahl 3 wird zum Beispiel durch `Succ (Succ (Succ Zero))` dargestellt.

- (a) Schreiben Sie eine Funktion `natToInt :: Nat -> Int` zum Umwandeln von einer Darstellung in die andere.
- (b) Die Addition kann man folgendermaßen definieren:

```
add :: Nat -> Nat -> Nat
add Zero m = m           -- add.1
add (Succ n) m = Succ (add n m) -- add.2
```

Beweisen Sie mit struktureller Induktion, dass diese Addition assoziativ ist: $(x \text{ 'add' } y) \text{ 'add' } z = x \text{ 'add' } (y \text{ 'add' } z)$. Überlegen Sie, welche Variable Sie als Induktionsvariable wählen. Begründen Sie jeden Schritt, indem Sie angeben, welche Definitionsgleichung (add.1 oder add.2) oder welche anderen Eigenschaften sie verwenden.

68. Entwurf eines Schaltnetzes, 10 Punkte

Konstruieren Sie ein Schaltnetz mit drei Eingängen $x, y, z \in \{0, 1\}$ und zwei Ausgängen $a, b \in \{0, 1\}$. Der Ausgang a soll zu 0 ausgewertet werden, wenn y oder z gleich 1 ist. Der Ausgabe b wird zu 1 ausgewertet, falls die Zahl $x + y \cdot 2 + z \cdot 2^2$ durch 3 teilbar ist. Zeichnen Sie die Gatter entsprechend der DIN-Norm.

69. Endliche Automaten, 0 Punkte

Konstruieren Sie einen endlichen Automaten mit Eingabealphabet $In = \{A, C, T\}$ und Ausgabealphabet $Out = \{0, 1\}$, der genau dann 1 ausgibt, wenn das bisherige Eingabewort die Folge *ACAT* als Teilwort (siehe Aufgabe 29 vom 5. Blatt) enthält. Zeichnen Sie das Zustandsdiagramm.

70. Lauflängenkodierung (run-length encoding), Programmieraufgabe, 10 Punkte

Bei der Lauflängenkodierung wird eine Zeichenkette "aaaabbbaaac" mit vielen wiederholten Zeichen komprimiert, indem man die Länge jedes *Laufes* von gleichen Zeichen nimmt: [(4, 'a'), (2, 'b'), (3, 'a'), (1, 'c')]. Schreiben Sie eine Funktion `komp`, die diese Kodierung berechnet, und die Umkehrfunktion `dekomp` für die Dekodierung.

71. Nichtassoziative Faltung von Listen, 0 Punkte

Beschreiben Sie das Ergebnis der Funktion

```
differenzen :: Integer -> Integer -> Integer -> Integer
differenzen a b c = foldr (-) a [b..c]
```

durch eine explizite Formel in den Größen a, b, c (oder mehrere Formeln, die für verschiedene Fälle passen). Beweisen Sie Ihre Formel, zum Beispiel durch vollständige Induktion nach $c - b$, oder auch direkt.