6. Aufgabenblatt zur Vorlesung

Informatik A

WiSe 2015/16

Wolfgang Mulzer

Abgabe am 1. Dezember 2015 bis 12 Uhr in die Tutorenfächer

Aufgabe 1 Einfaches Haskell *10 Punkte*

(a) Implementieren Sie die folgenden Funktionen direkt. Geben Sie auch jeweils eine geeignete Signatur an. Testen Sie Ihre Funktionen an geeigneten Eingaben.

(i) Eine Funktion `threeDiff`, die als Eingabe drei ganze Zahlen erhält, und genau dann `True` liefert, wenn die Argumente paarweise verschieden sind.

(ii) Eine Funktion `threeEqual`, die als Eingabe drei ganze Zahlen erhält, und genau dann `True` liefert, wenn alle Argumente gleich sind.

(iii) Eine Funktion `howManyDiff`, die als Eingabe drei ganze Zahlen erhält, und welche ermittelt, wie viele verschiedene Argumente auftreten.

(b) Implementieren Sie `threeDiff` und `threeEqual` mit Hilfe von `howManyDiff`.

Aufgabe 2 Einfache Rekursion *10 Punkte*

(a) Schreiben Sie eine Funktion `klammerTest`, die testet, ob die in einer Zeichenkette vorkommenden öffnenden und schließenden runden Klammern `'('` und `')'` einen korrekten Klammerausdruck bilden. Das bedeutet: in jedem Präfix (Anfangsstück) treten mindestens so viele öffnende wie schließende Klammern auf, und insgesamt gibt es ebenso viele öffnende wie schließende Klammern. Beispielsweise sind `()()(())` und `((())()(()))` gültig, nicht aber `(()()` oder `)()()(`. Ihre Funktion soll einen entsprechenden Wahrheitswert zurückliefern.

Definieren Sie ggf. geeignete Hilfsfunktionen. Geben Sie alle Signaturen an, und testen Sie Ihre Funktionen an geeigneten Beispielen.

(b) Schreiben Sie eine rekursive Funktion `powerOfTwo`, die rekursiv für gegebenes n die Potenz 2^n ausrechnet. Um den Vorgang zu beschleunigen, soll Ihre Funktion durch wiederholtes Quadrieren vorgehen, anstatt immer wieder mit 2 zu multiplizieren.

Geben Sie alle Signaturen an, und testen Sie Ihre Funktionen an geeigneten Beispielen.

Hinweis: Unterscheiden Sie, ob n gerade oder ungerade ist.

Aufgabe 3 Zusammengesetzte Datentypen *10 Punkte*

Ein Kreis in der Ebene kann durch drei Gleitkommazahlen dargestellt werden: Die ersten zwei Zahlen sind die x- und die y-Koordinate des Mittelpunktes, die dritte ist der Radius. Schreiben Sie ein Haskell-Skript `Circle.hs` oder `Circle.lhs`, das einen geeigneten Datentypen `Circle` definiert und mindestens folgende Funktionen enthält.

(a) Die Funktion `isCirc` überprüft, ob die drei Zahlen tatsächlich einen Kreis mit nichtleerer Fläche darstellen.

(b) Die Funktion `areaBBox` berechnet die Fläche der sogenannten *bounding box* zweier Kreise, also des kleinsten achsenparallelen Rechtecks, das beide enthält.

(c) Die Funktion `isContained` überprüft von zwei Kreisen, ob einer davon im anderen völlig enthalten ist.

Hinweis: Eine Möglichkeit besteht darin, eine Funktion zu benutzen, die den Abstand der Mittelpunkte bestimmt. Wie?

Geben Sie alle Signaturen an, und testen Sie Ihre Funktionen an geeigneten Beispielen.