

Übungen
Deskriptive Programmierung
SS 15

Blatt 4

Hinweis: Wenden Sie bitte weiterhin `hlint` auf Ihre Einreichungen an (auch um Punktabzüge zu vermeiden).

Hinweis: Für die folgende Autotool-Aufgabe gilt eine Ausnahme, dass beliebig viele Fehlversuche (statt lediglich 5 oder 10) erlaubt sind.

Aufgabe 18 (zu lösen/einzureichen über Autotool, ∞ Fehlversuche erlaubt, [6P]).

Hinweis: Für die folgende Autotool-Aufgabe ist lediglich 1 Fehlversuch erlaubt. Nutzen Sie `QuickCheck` lokal!

Aufgabe 19 (zu lösen/einzureichen über Autotool, 1 Fehlversuch erlaubt, [6P]).

Aufgabe 20 (einzureichen über eCampus, als Quelldatei, [6P]).

Im Graphikpaket `Gloss` werden Animationen als Abbildungen von Zeitpunkten in Bilder gefasst.

type Animation = Float \rightarrow Picture

Schreiben Sie zwei Haskell-Funktionen, die Listen solcher Animationen nacheinander anzeigen.

animateSequenceRel :: [(Animation, Float)] \rightarrow Animation \rightarrow Animation
animateSequenceAbs :: [(Animation, Float)] \rightarrow Animation \rightarrow Animation

Der erste Parameter ist eine Folge von Animationen und ihrer jeweiligen positiven Anzeigedauer. Die Animationen sollen jeweils für diese Dauer nacheinander angezeigt werden. Nach Ablauf aller Animationen in der Liste soll die Animation aus dem zweiten Parameter gezeigt werden. In der ersten Funktion sollen die Animationen jeweils zum Zeitpunkt 0 starten,

¹Bei Fragen wenden Sie sich bitte via E-Mail an Janis Voigtländer (jv@informatik.uni-bonn.de).

während bei der zweiten Funktion die Animationen den tatsächlichen Zeitparameter erhalten sollen. Beispielsweise soll *animateSequenceAbs* (*cycle* [(*clock*₁, 3), (*clock*₂, 3)]) *undefined* alle 3 Sekunden zwischen zwei Uhren wechseln, wobei die angezeigte Zeit weiter voranschreitet. Versuchen Sie möglichst viel der Implementation der Funktionen *animateSequenceRel* und *animateSequenceAbs* zu teilen, statt getrennte Implementationen anzugeben. Idealerweise finden Sie eine Funktion, die beide Funktionalitäten abdeckt und mittels derer Sie obige Funktionen als parametrisierte Aufrufe schreiben können. Geben Sie eine *main*-Definition an, die eine interessante Verwendung einer der beiden Funktionen *animateSequenceRel* und *animateSequenceAbs* demonstriert.

Hinweis: Wenn Sie Ihre Lösungen zu der folgenden Autotool-Aufgabe zunächst mit einem lokal installierten **GHC** entwickeln und testen, müssen Sie (je nach Betriebssystem und Einstellung) zum sinnvollen Ausprobieren der Interaktion eventuell jeweils noch eine zusätzliche Bibliothek per „**import System.IO**“ einbinden und Ihre Programme (in *main*) mit „*hSetBuffering stdout NoBuffering*“ beginnen.
(In Autotool hat dies keinen Effekt und kann weggelassen werden – die eventuell hinzugefügte Zeile „**import System.IO**“ *muss* da sogar wieder weggelassen werden.)

Aufgabe 21 (zu lösen/einzureichen über Autotool, 5 Fehlversuche erlaubt, [6P]).