Übungen zur "Theorie der Programmierung" - SoSe2015

Freitag 12:15-13:45, Martenstr. 3, Raum 02.133-113

Daniel Hausmann

daniel.hausmann@fau.de

Friedrich-Alexander-Universität Erlangen
Department Informatik

Lehrstuhl 8

June 24, 2015

G-Koalgebra

Für einen **Mengenoperator** (Funktor) G ist eine G-**Koalgebra** $\mathfrak{M}$ eine Struktur der Form (M Trägermenge von $\mathfrak{M}$)

$$\alpha : M \to GM$$

G-**Koalgebramorphismus** $f : \mathfrak{M} \to \mathfrak{N}$:

$$\begin{tikzcd} M \arrow[r, "f"] \arrow[d, "\alpha"'] & N \arrow[d, "\beta"] \\ GM \arrow[r, "Gf"'] & GN \end{tikzcd}$$

G-Koalgebra $\mathfrak{M}$ heißt **final**, gdw. es für jede G-Koalgebra $\mathfrak{N}$ genau einen G-Koalgebramorphismus $t : \mathfrak{N} \to \mathfrak{M}$ gibt.

Kodatentypen (mit Alternativen)

Für $GX = \sum_{i=1}^{n} A_i \times X^{k_i}$ wird G-Koalgebra definiert durch Signatur Σ:

- Alternativen $d_1, \ldots d_n$
- für $i \in \{1, \ldots n\}$ je $k_i + 1$ **Observer** (Destruktoren):

$$t_{ij} : d_i \to d \text{ für } j \in \{1, \ldots k_i\} \text{ und } h_i : d_i \to a_i \text{ mit } [\![a_i]\!] = A_i$$

Koalgebra $\mathfrak{N}$ besteht aus Träger $N = \biguplus_{i=1}^{n} \mathfrak{N}[\![d_i]\!]$ und Funktionen

$$\mathfrak{N}[\![t_{ij}]\!] : \mathfrak{N}[\![d_i]\!] \to N \qquad \mathfrak{N}[\![h_i]\!] : \mathfrak{N}[\![d_i]\!] \to A_i$$

Korekursion

Ein G-Koalgebramorphismus $f : \mathfrak{M} \to \mathfrak{N}$ in die finale G-Koalgebra $\mathfrak{N}$ kann durch Korekursion eindeutig definiert werden:

- $f[\mathfrak{M}[\![d_i]\!]] \subseteq \mathfrak{N}[\![d_i]\!]$
- $\mathfrak{N}[\![h_i]\!](f(x)) = \mathfrak{M}[\![h_i]\!](x)$
- $\mathfrak{N}[\![t_{ij}]\!](f(x)) = f(\mathfrak{M}[\![t_{ij}]\!](x))$

Übung 8 - Ein koinduktives Animationsstudio

codata Animation where

$$\begin{aligned} &\mathit{sprite} &&: \textbf{Animation} \to \textbf{Sprite} \\ &\mathit{advance} &&: \textbf{Animation} \to \textbf{Animation} \end{aligned}$$

$$\begin{aligned} &\mathit{sprite}\ (\mathit{loop}\ (\mathit{Cons}\ s\ ss)) &&= s \\ &\mathit{advance}\ (\mathit{loop}\ (\mathit{Cons}\ s\ ss)) &&= \mathit{loop}\ (\mathit{snoc}\ ss\ s) \\ &\mathit{sprite}\ (\mathit{loop}\ \mathit{Nil}) &&= \mathit{blankSprite} \\ &\mathit{advance}\ (\mathit{loop}\ \mathit{Nil}) &&= \mathit{loop}\ \mathit{Nil} \end{aligned}$$

1. Es sei $\mathit{walk\_right} = [s1, s2, s3, s4, s5, s6]$:

Was ist das Ergebnis der Auswertung des folgenden Terms?

$$\mathit{sprite}\ (\mathit{advance}\ (\mathit{advance}\ (\mathit{advance}\ (\mathit{loop}\ \mathit{walk\_right}))))$$

Übung 8 - Ein koinduktives Animationsstudio

2. Definieren Sie eine korekursive Funktion

$$\mathit{delay} \;:\; \textbf{Animation} \to \textbf{Animation},$$

die die folgende Eigenschaft für alle $n \in \mathbb{N}$ erfüllt:

$$\mathit{sprite}\ (\mathit{advance}^n\ (\mathit{delay}\ a)) = \begin{cases} \mathit{sprite}\ a & \text{falls } n = 0 \\ \mathit{sprite}\ (\mathit{advance}^{n-1}\ a) & \text{falls } n > 0 \end{cases}$$

3. Definieren Sie eine korekursive Funktion $\mathit{halfspeed} \;:\; \textbf{Animation} \to \textbf{Animation}$, die jeden Frame verzögert.

4. Definieren Sie zwei Funktionen $\mathit{doublespeed\_e},\ \mathit{doublespeed\_o} \;:\; \textbf{Animation} \to \textbf{Animation}$ korekursiv, so dass $\mathit{doublespeed\_e}$ alle ungeraden und $\mathit{doublespeed\_o}$ alle geraden Frames überspringt.
Hinweis: Möglicherweise werden Sie eine Hilfsfunktion definieren müssen.

Übung 8 - Ein koinduktives Animationsstudio

5. Definieren Sie mittels Korekursion eine Funktion *prepend* : **List Sprite** → **Animation** → **Animation** welche eine einmalig abzuspielende Startsequenz vor einer Animation einfügt.
6. Wir nehmen an, dass eine Funktion *compatible* : **Sprite** → **Sprite** → **Bool** gegeben ist. Definieren Sie eine Funktion *transition* : **Animation** → **Animation** → **Animation**, so dass *transition a1 a2* eine Animation ist, die so lange *a1* abspielt, bis ein mit dem ersten Sprite von *a2* kompatibles Sprite erreicht wird, und sodann zu *a2* übergeht.

Bisimulation

Eine binäre Relation $R \subseteq M \times M$ heißt **Bisimulation** auf einer G-Koalgebra $\mathfrak{M}$ mit n Alternativen $d_1, \ldots, d_n$ genau dann wenn für alle $(s, t) \in R$ und alle $1 \leq i \leq n$ gilt:

Wenn $s @ \mathfrak{M}[\![d_i]\!]$*, dann*

- $t @ \mathfrak{M}[\![d_i]\!]$
- $\mathfrak{M}[\![h_i]\!](s) = \mathfrak{M}[\![h_i]\!](t)$
- $\mathfrak{M}[\![t_{ij}]\!](s)\ R\ \mathfrak{M}[\![t_{ij}]\!](t)$ *für alle* $1 \leq j \leq k_i$

Satz: Koinduktion

Wenn R Bisimulation ist und $s\,R\,t$, dann $s = t$.

Beobachte: id_M ist Bisimulation; sind R und S Bisimulationen, dann ist auch $R \cup S$ Bisimulation.

Übung 9 - Koinduktion

1. Es seien a und b zwei Terme des Typs **Animation**. Um mittels Koinduktion zu zeigen, dass $a = b$ gilt, müssen wir eine **Bisimulation** R (für den Typ **Animation**) finden, so dass $a\ R\ b$. Definieren Sie den Begriff einer Bisimulation für den Typ **Animation**.
2. Beweisen Sie die folgenden Eigenschaften mittels Koinduktion.
    1. $\forall a,\ doublespeed\_e\ (halfspeed\ \ a) = a$
    2. $\forall a,\ doublespeed\_e\ a = doublespeed\_o\ (delay\ \ a)$
    3. $\forall s\ \ t\ \ ts,\ compatible\ s\ \ t = False \Rightarrow$ $transition\ \ (loop\ [s])\ (loop\ (Cons\ t\ \ ts)) = loop\ [s]$
    4. $\forall s\ \ ss\ \ t\ \ ts,\ compatible\ s\ \ t = True \Rightarrow$ $transition\ \ (loop\ (Cons\ s\ \ ss))\ (loop\ (Cons\ t\ \ ts)) =$ $prepend\ [s]\ \ (loop\ (snoc\ \ ts\ \ t))$

Hinweis: Möglicherweise müssen Sie den Kandidaten für Ihre Bisimulation **vergrössern**.

Übung 10 - Ein koinduktiver zustandsloser Server

$$
\begin{array}{l}
\textbf{codata}\ \mathit{Requests}\ a\ \textbf{where} \\
\quad \begin{array}{lcll}
\mathit{eof} & : & \mathit{Requests}\ a@\mathit{closed} & \to () \\
\mathit{retry} & : & \mathit{Requests}\ a@\mathit{waiting} & \to \mathit{Requests}\ a \\
\mathit{request} & : & \mathit{Requests}\ a@\mathit{ready} & \to a \\
\mathit{consume} & : & \mathit{Requests}\ a@\mathit{ready} & \to \mathit{Requests}\ a
\end{array}
\end{array}
$$

Drei Verbindungen:

$$\mathit{eof}\ \ \mathit{oops} = ()$$

$$\mathit{retry}\ \ \ \mathit{indecisive}\ \ =\ \ \mathit{indecisive}$$

$$
\begin{array}{l}
\mathit{request}\ \ (\mathit{forever}\ \ a)\ = a \\
\mathit{consume}\,(\mathit{forever}\ \ a)\ = \mathit{forever}\ \ a
\end{array}
$$

Übung 10 - Ein koinduktiver zustandsloser Server

Serielle Komposition von Verbindungen:

$$
\begin{aligned}
&\mathit{eof}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{closed}\ \ \mathit{ys}@\mathit{closed}) &&= ()\\
&\mathit{retry}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{waiting}\ \ \mathit{ys}) &&= \mathit{serial}\ (\mathit{retry}\ \ \mathit{xs})\ \ \mathit{ys}\\
&\mathit{retry}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{closed}\ \ \mathit{ys}@\mathit{waiting}) &&= \mathit{retry}\ \ \mathit{ys}\\
&\mathit{request}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{ready}\ \ \mathit{ys}) &&= \mathit{request}\ \mathit{xs}\\
&\mathit{request}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{closed}\ \ \mathit{ys}@\mathit{ready}) &&= \mathit{request}\ \mathit{ys}\\
&\mathit{consume}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{ready}\ \ \mathit{ys}) &&= \mathit{serial}\ (\mathit{consume}\ \mathit{xs})\ \mathit{ys}\\
&\mathit{consume}\ (\mathit{serial}\ \ \mathit{xs}@\mathit{closed}\ \ \mathit{ys}@\mathit{ready}) &&= \mathit{consume}\ \mathit{ys}
\end{aligned}
$$

1. Definieren Sie den Begriff einer Bisimulation für den Typ *Requests a*.
2. Beweisen Sie die folgenden Eigenschaften mittels Koinduktion:
   1. $\forall s,\ \ \mathit{serial}\ \ \mathit{oops}\ s = s$
   2. $\forall r,\ \ \mathit{serial}\ \ r\ \mathit{oops} = r$
   3. $\forall s,\ \ \mathit{serial}\ \ \mathit{indecisive}\ \ s = \mathit{indecisive}$
   4. $\forall a\ s,\ \ \mathit{serial}\ \ (\mathit{forever}\ \ a)\ \ s = \mathit{forever}\ \ a$

Übung 10 - Ein koinduktiver zustandsloser Server

3. Definieren Sie eine Funktion *par*, die zwei Verbindungen **parallelisiert** und beweisen Sie:
   1. $\forall s,\ par\ oops\ s = s$
   2. $\forall r,\ par\ r\ oops = r$
   3. $\forall s,\ par\ \ indecisive\ \ s = serial\ \ s\ \ indecisive$
   4. $\forall r,\ par\ r\ \ indecisive\ = serial\ \ r\ \ indecisive$
   5. $\forall a\ b,\ par\ (\ intermittent\ \ a)\ (\ intermittent\ \ b) = alternate\ a\ b,$ wobei

$$
\begin{aligned}
&request\ \ (\,alternate\ \ a\ \ b) = a\\
&consume\,(\,alternate\ \ a\ \ b) = alternate\ \ b\ a
\end{aligned}
$$

$$
\begin{aligned}
&request\ \ (\ intermittent\ \ a) = a\\
&consume\,(\,intermittent\ \ a) = waiting\ a\\
&\quad \textbf{where}\ retry\ (waiting\ \ a) = intermittent\ \ a
\end{aligned}
$$