Formale Modellierung
Vorlesung 10 vom 10.06.13: Modellierung von Programmen

Serge Autexier & Christoph Lüth

Universität Bremen

Sommersemester 2013

Fahrplan

- ► Teil I: Formale Logik
- ► Teil II: Spezifikation und Verifikation
 - ► Modellierung von Programmen
 - ▶ Die Z-Notation
 - ► Formale Modellierung mit der UML und OCL
- ► Teil III: Schluß

2 [13]

Das Tagesmenü

- ► Bis jetzt:
 - 1. Grundlagen formale Logik
 - 2. Statische Datentypen
- ► Heute: Programme
 - 1. Funktional,
 - 2. imperativ,
 - 3. objektorientiert.

[13]

Funktionale Programme

- ▶ Wir haben:
- 1. Logik höherer Stufe
- 2. terminierende rekursive Funktionen
- 3. induktive (generierte) Datentypen
- ► Funktionale Programme:
 - 1. Funktionen höherer Ordnung
 - 2. Beliebige Rekursion
 - 3. Beliebige Datentypen

4 [1

Modellierung beliebiger Rekursion

Definition 1 (Partielle Ordnung)

 (X, \sqsubseteq) mit \sqsubseteq antisymmetrisch, reflexiv und transitiv.

Definition 2 (CPO)

 (X,\sqsubseteq) mit \sqsubseteq partielle Ordnung und jede Kette C

 $x_1 \sqsubseteq x_2 \sqsubseteq \dots x_n \sqsubseteq \dots$

hat kleinste obere Schranke $\bigsqcup_{i<\omega} x_i$

Definition 3 (Stetige Funktion)

Funktion $f:(X,\sqsubseteq) \to (Y,\preccurlyeq)$ ist Scott-stetig, wenn f die Ordnung \sqsubseteq und $|\;|$ bewahrt.

5 [13]

Eigenschaften von CPOs

Theorem 4 (Kleene-Tarski-Fixpunktsatz)

Sei $f:(X,\sqsubseteq) \to (X,\sqsubseteq)$ Scott-stetig und $\bot \in X$ kleinstes Element, dann hat f einen kleinsten Fixpunkt lfp_f mit $f(\mathit{lfp}_f) = \mathit{lfp}_f$.

- ► CPOs und stetige Funktionen modellieren funktionale Programme (Logic of computable functions, LCF)
- ► CPOs modellieren Domänen (Datentypen für funktionale Programme)
- ► Möglich: konservative Einbettung in HOL (HOLCF)

6 [13

Ein Beispiel

Beispiel: Flugbuchungssystem

- ► Ein Flug hat eine eindeutige Flugnummer, Start und Ziel, sowie die Anzahl Sitze (verfügbar und insgesamt).
- ▶ Das Flugbuchungssystem identifiziert Flüge anhand ihrer Flugnummer.
- ► Das Flugbuchungsystem soll folgende Operationen unterstützen:
 - ► Anfrage nach Verbindung (Start und Ziel) sowie Anzahl Plätze;
 - ► Buchung mit Flugnummer, Anzahl Plätze
- ► Modellierung funktional (Isabelle/HOL): axiomatisch vs. konservativ
- Modellierung imperativ: Zustandsübergang

Die Z Notation

- ► Basiert auf getypter Mengenlehre
- ► Entwickelt seit 1980 (Jean-Claude Abrial, Oxford PRG)
- Industriell genutzt (IBM, Altran Praxis (vorm. Praxis Critical Systems))
- ► LATEX-Notation und Werkzeugunterstützung (Community Z Tools, HOL-Z, ProofPower)

7 [1

8 [1

```
Modellierung in Z (2)

Lookup

FlightDB

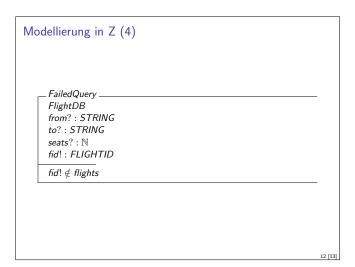
flid?: FLIGHTID

flight!: Flight

flid? \in flights

flight! = data(flid?)
```

Modellierung in Z (3) SuccessfulQuery FlightDB from?: STRING to?: STRING seats?: \mathbb{N} fid!: FLIGHTID fid! \in flights (data(fid!)).avail \geq seats? (data(fid!)).dept \in from? (data(fid!)).arr \in to?



Zusammenfassung

- Modellierung funktionaler Programme: cpos (LCF) für beliebige Rekursion
 - ► Datenmodellierung: erzeugte Datentypen
- ► Modellierung imperativer Programme: Zustandsübergang
 - ▶ Datenmodellierung: als Mengen (Z) oder als Klassen (UML/OCL)
- ▶ Nächste Woche: Z im Gesamtüberblick

13 [13]