

## Übungen zu „Formale Methoden“

### Aufgabe 1 E/R-Modellierung und OO-Datenmodellierung

Zu modellieren ist ein Auskunftssystem eines Flughafens, in dem nationale und internationale Flüge repräsentiert werden können. Das System soll Auskunft über Flugnummer, Start- und Zielflughafen sowie Ankunft- und Abflugzeiten geben können. Außerdem sollen aus dem System die durch den Flug verbundenen Flughäfen ersichtlich sein.

- (a) Geben Sie ein mögliches E/R-Modell des Auskunftsystems an. Dabei sind folgende Eigenschaften umzusetzen:
- Jeder Flughafen liegt in einer Stadt. Eine Stadt kann aber mehrere Flughäfen haben.
  - Jeder Flug hat einen bestimmten Start- und Zielflughafen.
  - Auf seiner Route (Verbindung Start-/Zielflughafen) kann ein Flug mehrere Zwischenstops haben.

Erstellen Sie dafür ein E/R-Diagramm und die entsprechende Beschreibung der Bestandteile des Diagramms. Geben Sie auch die Kardinalitäten der Beziehungen an.

- (b) E/R-Diagramme können als eine kompakte Schreibweise für algebraische Spezifikationen aufgefasst werden. Mittels Formeln können in algebraischen Spezifikation jedoch weitere Eigenschaften beschrieben werden, die in E/R-Diagrammen so nicht direkt ausdrückbar sind.  
Veranschaulichen Sie sich diese Aussagen anhand kurzer Beispiele im Bezug auf das modellierte Flugauskunftssystem.
- (c) Wandeln Sie den Flug-Teil des erstellten E/R-Diagramms in ein OO-Diagramm um. Wie werden dabei Beziehungen aus dem E/R-Diagramm umgesetzt? Wie sieht das konkrete OO-Teildiagramm für den Sachverhalt aus, dass sowohl *Cargo-Flüge* als auch *Linien-Flüge* Flüge sind.

## Aufgabe 2 Fehlermodellierung

*Beblätterte Binärbäume* sind spezielle, geordnete Binärbäume, deren Blätter die einzigen Knoten sind, die mit Werten (Atomen) markiert sind. Man kann beblätterte Bäume auf folgende Weise induktiv definieren:

- Ein leerer Baum ist ein beblätterter Binärbaum.
- Ein 1-Blättriger Baum ist ein beblätterter Binärbaum.
- Ein Baum, der genau zwei Unterbäume hat, die selbst beblätterte Binärbäume sind, ist ein beblätterter Binärbaum.

Geben Sie eine algebraische Spezifikation BTREE für beblätterte Binärbäume an. Welche Fehlerfälle sind denkbar und wie kann der Umgang mit Fehlerfällen in der Spezifikation berücksichtigt werden? Erweitern Sie die Spezifikation entsprechend.

## Aufgabe 3 (Hausaufgabe) Algebraische Spezifikation

(12 Punkte; Abgabe bis spätestens 27.11.2006 zu Beginn der Übung.)

Gegeben sei eine algebraische Spezifikation DATE zur Darstellung von Terminen in einem Terminkalender (D.h. ein Termin ist eine im Kalender belegte Zeitspanne.)

$$\text{SPEC DATE} = \left\{ \begin{array}{l} \mathbf{based\_on} \text{ BOOL,} \\ \mathbf{sort} \text{ Date,} \\ \dots \\ \mathbf{==} : \text{ Date, Date} \rightarrow \text{Bool,} \\ \mathbf{conflict} : \text{ Date, Date} \rightarrow \text{Bool,} \\ \dots \end{array} \right\}$$

Die darin spezifizierte Funktion `conflict` gibt an, ob sich zwei Zeitspannen/Termine (ganz oder teilweise) überlagern, d.h. `conflict(d, e)` liefert *true*, falls sich die beiden Zeitspannen/Termine *d* und *e* zeitlich überschneiden, andernfalls *false*. Die Funktion `==` liefert *true* falls zwei Zeitspannen/Termine identisch sind (Beide dauern gleich lang und beginnen zum gleichen Zeitpunkt).

### **Aufgabe:**

Geben Sie eine algebraische Spezifikation für die Datenstruktur DIARY eines Terminkalenders an. Verwenden Sie zur Darstellung von Terminen bzw. Zeitspannen Elemente der Sorte `Date`. Für Ihre Lösung können Sie die Vorlage auf Seite 4 verwenden.

(a) Ergänzen Sie zuerst die Spezifikation um die Signaturen der folgenden Funktionen:

- Eine Funktion `emptyd`, die einen neuen (leeren) Kalender der Sorte `Diary` erzeugt.
- Eine Funktion `isfreed`, die überprüft ob eine angegebene Zeitspanne in einem Terminkalender noch frei (leer) ist, d.h. keine Termine enthält bzw. sich mit keinen Terminen überschneidet. Eine Zeitspanne überschneidet sich mit einem Termin genau dann, wenn der Start- oder der Endzeitpunkt des Termins innerhalb der Zeitspanne liegt.

- Eine Funktion `putd` zum Eintragen eines neuen Termins in einen Kalender.
- Eine Funktion `getd` zum Abfragen von Terminen eines Terminkalenders, die (komplett oder teilweise) innerhalb einer Zeitspanne liegen. `getd` liefert dabei die *Menge* aller solcher Termine.
- Eine Funktion `deld`, die einen Termin aus einem Kalender entfernt.

Geben Sie den/die Generator(en) nach dem Schlüsselwort `generated.by` an und ergänzen Sie die benötigte(n) Spezifikation(en) nach dem Schlüsselwort `based.on` entsprechend.

(b) Formulieren Sie die folgenden Aussagen als Axiome in der Spezifikation `DIARY`.

- (A1) In einem neuen (leeren) Terminkalender ist noch kein Eintrag belegt.
- (A2) Das Anlegen eines neuen Termins  $d$  in einer vormals freien Zeitspanne  $d$  ändert an der Belegung einer anderen Zeitspanne  $e$  nur dann etwas, falls sich der neue Termin  $d$  und die Zeitspanne  $e$  überschneiden.
- (A3) Falls eine Zeitspanne  $d$  eines beliebigen Terminkalenders  $y$  noch frei (leer) ist, enthält sie *keine* Termine.
- (A4) Falls man eine vormals unbelegte Zeitspanne  $d$  eines Terminkalenders komplett durch einen Termin belegt, so enthält diese Zeitspanne  $d$  danach ausschließlich diesen einen Termin.
- (A5) Falls ein Termin in eine vormals leere Zeitspanne  $d$  eingefügt wird, und falls  $d$  (teilweise oder ganz) innerhalb einer zweiten Zeitspanne  $e$  liegt, so taucht dieser Termin bei der Abfrage aller in der Zeitspanne  $e$  enthaltenen Termine zusätzlich zu den anderen Terminen in  $e$  mit auf.
- (A6) Falls sich ein neuer Termin  $d$  beim Einfügen in einen Terminkalender  $y$  *nicht* mit einem freien Eintrag  $e$  überschneidet, so taucht dieser Termin  $d$  auch bei der Abfrage aller Termine, die (ganz oder teilweise) innerhalb der Zeitspanne  $e$  liegen, nicht auf.
- (A7) Die Löschoption `deld` wirkt sich nur auf die Termine  $d$  aus, die *identisch* mit der zu löschenden Zeitspanne  $e$  sind. Alle anderen Termine bleiben beim Löschen von  $e$  unberührt.

**SPEC DIARY = {**

**based\_on**

**sort** Diary,

emptyd :

isfreed :

putd :

getd :

deld :

Diary **generated\_by**

(A1)

(A2)

(A3)

(A4)

(A5)

(A6)

(A7)

}