Technische Universität München
Fakultät für Informatik
Prof. Dr. A. Knoll

SS 2006
Übungsblatt 5
30. Juni 2006

Übungen zu Einführung in die Informatik II

Aufgabe 15 Kleidung

Ein beliebtes Beispiel für den Ablauf eines Prozesses in der Realität ist das morgendliche Anlegen von Kleidung. Dabei zieht sich eine Person nacheinander unterschiedliche Kleidungsstücke an, wobei evtl. auf die richtige Reihenfolge geachtet werden muss, um ein sinnvolles Ergebnis zu erhalten (so muss das Unterhemd sicher vor dem Mantel angezogen werden).

a) Modellieren Sie einen entsprechenden Prozess für die folgenden Kleidungsstcke: Socken, Unterhemd, Mantel, Schuhe, Hemd, Unterhose, Hut, Krawatte, Sacko, Hosenträger, Jacket, Gürtel. Dabei soll es noch zulässig sein sich das (Unter-) Hemd nachträglich in die (Unter-) Hose zu stopfen.

b) Geben Sie ein Aktionsdiagramm für den oben definierten Prozess an, bei dem alle zur Verfügung stehenden Kleidungstücke angelegt werden.

c) Geben Sie eine vollständige Sequenzialisierung zu dem Aktionsdiagramm an.

d) Zeigen Sie durch Bildung der transitiven Hülle, dass das Jacket nach der Unterhose angezogen werden muss.

Aufgabe 16 Kleinstes Präfix

Ausgehend von der Ereignismenge $E = \{e_i : i \in IN_0\}$und der Aktionenmenge $A = \{a_0, a_1, a_2, a_3\}$ sei der Prozess $P = (E, <, \alpha)$ gegeben durch

$\alpha : E \to A$ def. durch $\alpha(e_i) = a_{imod4}$ für $i \in I\!N_0$,

$< \subseteq E \times E$ def. als die reflexiv transitive Hülle der Menge $\{(e_i, e_{i+2}), (e_{4i}, e_{4i+3}), (e_{4i+3}, e_{4i+4}) : i \in I\!N_0\}$.

a) Beschreiben Sie graphisch das kleinste Präfix $P_{4,5}$ von P, das die Ereignisse e_4 und e_5 enthält.

b) Geben Sie einige unvollständige Sequentialisierungen von $P_{4,5}$ an.

c) Geben Sie die Spuren von $P_{4,5}$ an.

Aufgabe 17 Aktionsstruktur (Prozess)

a) Zur Definition der Aktionsstruktur $(E_0, <_0, \alpha)$ über einer Menge E von Ereignissen und einer Menge A von Aktionen gehört die Forderung, dass die Kausalitätsrelation $<_0$ endlich fundiert ist.

(i) Welche intuitive Vorstellung modelliert diese Forderung?

(ii) Jede endlich fundierte Kausalordnung ist Noethersch. Geben Sie ein Beispiel dafür, dass die Umkehrung nicht gilt.

b) Ist der Prozess $(\{e_1, e_2\}, \{(e_1, e_1), (e_2, e_2)\}, \{e_1 \mapsto a_1, e_2 \mapsto a_2\})$

(i) ein Teilprozess von,

(ii) eine Sequentialisierung von,

(iii) isomorph zu

dem Prozess $(\{e_1, e_2\}, \{(e_1, e_1), (e_2, e_2)\}, \{e_1 \mapsto a_2, e_2 \mapsto a_1\})$?
(Begründung?)

Aufgabe 18 Bank-Konto in Java

Wird bei mehreren Threads auf gemeinsame Daten zugegriffen, muss dies synchronisiert werden. In diesem Beispiel sollen mehrere nebeläufige Prozesse Geld auf ein Bank-Konto einzahlen bzw. abheben können. Für die Synchronisation von Threads stellt Java u.a. das `synchronized`-Konzept zur Verfügung. Code-Segmente, die auf das selbe Ojekt von konkurrierenden Threads aus zugreifen, werden als kritische Sektionen bezeichnet. Kritische Sektionen können sowohl Funktionsblöcke als auch vollständige Methoden sein und werden durch `synchronized` gekennzeichnet. Bei Eintritt in einen `synchronized`-Bereich wird das betreffende Objekt von Java gesperrt und erst bei Verlassen dieses Bereiches wieder freigegeben. Die Methoden `wait`, `notify` und `notifyAll` ermöglichen Threads auf Bedingungen zu warten und andere Threads zu benachrichtigen, sobald diese eintreten.

```
public class MyData {
  private int data;
  ...
  public synchronized int get(...) {
    // locked
    ...
    // unlocked
  }
  ...
  public synchronized void set(...) {
    // locked
    ...
    // unlocked
  }
}
```

a) Implementieren Sie eine Verwaltung für ein Bank-Konto geeignet, dass das Einzahlen und Abheben eines `int`-Wertes erlaubt. Das Bank-Konto darf einen Maximalbetrag nicht überschreiten. Das Abheben ist nur zulässig wenn der gewünschte Betrag auf dem Konto vorhanden ist.

b) Implementieren Sie einen Thread, der in zufälligen Abständen einen festgelegten `int`-Wert auf das Konto einzahlt.

c) Implementieren Sie einen Thread, der in zufälligen Abständen einen festgelegten `int`-Wert vom Konto abhebt.

d) Erzeugen Sie mehrere Threads, die sowohl auf das Konto einzahlen, als auch von ihm abheben.