Algorithmen und Programmieren III, WS 1999/2000 — 7. Übungsblatt

Abgabe: Aufgabe 37 bis Dienstag, 4. Jänner, Aufgaben 39–40 bis Dienstag, 11. Jänner 2000.

---

37. (9 Punkte) Ändern Sie die Warteschlangensimulation aus Aufgabe 18(a) wie folgt.

    Die mittlere Zeit zwischen zwei Kunden sinkt auf $1/\lambda = 10$ Sekunden, dafür gibt es nun 12 Kassen. Bestimmen Sie zusätzlich die durchschnittliche Länge ℓ der Warteschlange im Zeitraum zwischen 12:00 und 13:00 Uhr, sowie den Zeitpunkt t, wann der letzte Kunde fertig ist. Führen Sie 100 unabhängige Simulationsläufe durch, und bestimmen Sie dem maximalen, den minimalen, und den Mittelwert von ℓ und t.

    Sie können Ihre eigene Lösung von Aufgabe 18(a) oder die Musterlösung `Simulation.java`[1] modifizieren. Markieren Sie in Ihrem Programmausdruck alle Teile, die Sie unverändert übernehmen konnten, mit einem grünen Strich am Rand.

38. (0 Punkte) Modifizieren Sie das Programm `Simulation.java`[1] wie folgt: Die Klassen `Kassen` und `Kasse` sollen aus der Klasse `Simulation` herausgenommen werden. Die Methoden und Felder aus `Kassen` sollen durch entsprechende statische Methoden und Felder der Klasse `Kasse` ersetzt werden, sodass die Klasse `Kassen` überflüssig wird.

39. (10 Punkte) Der Computer Sumo-0 verfügt über 10 Eingaberegister, die von `E0` bis `E9` nummeriert sind, sowie 26 Akkumulatoren, die von `Va` bis `Vz` bezeichnet sind. Ein Programm für diesen Rechner ist in der Assemblersprache Sumatra geschrieben und sieht zum Beispiel wie das nebenstehende Programm aus.

    ```
    Vf=E0+E1
    Vc=Vf-E2
    Vu=Vf+Vf
    Vv=Vu*Vc
    Vk=Vf/E0
    Vl=Vk+E9
    ...
    ```

    In jeder Zeile wird das Ergebnis einer zweistelligen arithmetischen Operation der Variablen auf der linken Seite zugewiesen. Jede Variable darf höchstens *einmal* auf der linken Seite einer Zuweisung auftreten. Als Operand einer arithmetischen Operation kann entweder ein Eingaberegister Ei oder Inhalt einer Variablen Vx, deren Wert in einer früheren Zeile definiert wurde, auftreten.

    Die Programmzeilen werden der Reihe nach ausgeführt. Schleifen und Verzweigungen sind unbekannt.

    (a) (10 Punkte) Schreiben Sie ein Programm, das ein gültiges Sumatra-Programm einliest und die Abhängigkeit der Ergebnisse von den Operanden als Graph abspeichert.

    Das Programm soll folgende Fragen beantworten können: Welche Variablen hängen (möglicherweise[2]) vom Wert des Eingaberegisters Ei ab? Von welchen Eingaberegistern hängt der Wert der Variablen Vx ab?

    (b) (0 Punkte) Bei gewissen Eingaben stellt sich heraus, dass das Programm mit einem Laufzeitfehler abbricht, weil durch 0 dividiert wurde. Der Sumatra-Programmierer möchte den Laufzeitfehler dadurch vermeiden, dass er den Wert eines Eingaberegisters verändert (und alle anderen Eingaberegister gleich lässt). Ihr Programm soll feststellen, welche Eingaberegister überhaupt prinzipiell einen Einfluss auf das Auftreten einer Division durch Null haben können.

40. (5 Punkte) Dem primitiven Charakter der Sprache entsprechend, werden die Zeilen eines Sumatra-Programmes auf *Lochkarten* gestanzt. Nehmen wir an, dass der Stapel von Lochkarten eines Programmes zu Boden gefallen ist und die Zeilen durcheinandergemischt worden sind. Wie kann man die Programmzeilen wieder in eine gute Reihenfolge bringen? (Sie brauchen kein lauffähiges Programm für diese Aufgabe zu schreiben.)

---

[1] siehe `http://www.inf.fu-berlin.de/~rote/Lere/alp3/Simulation.java`

[2] Dass zum Beispiel nach der Anweisung `Vy=Vx-Vx` die Variable `Vy` in Wirklichkeit nicht von `Vx` abhängt, brauchen Sie nicht zu berücksichtigen.