Prof. Dr. Leif Kobbelt
Stefan Dollase, Ira Fesefeldt, Alexandra Heuschling, Gregor Kobsik

Lösung - Übung 1

Tutoraufgabe 1 (Vollständige und Strukturelle Induktion):

a) In dieser Aufgabe wiederholen wir vollständige Induktion. Beweisen Sie hierfür mittels vollständiger Induktion über n, dass

$$\sum_{i=1}^{n} i \quad = \quad \frac{n \cdot (n+1)}{2} \ .$$

b) Wenn wir Beweise für Datenstrukturen aufschreiben wollen, ist es häufiger eleganter die *Struktur* einer Datenstruktur ausnutzen. Dafür beweisen wir einen entsprechenden Satz mithilfe der *Strukturellen Induktion*. Nehmen wir beispielsweise die Struktur einer Liste, so haben wir die Prädikate

- CREATE(),
- INSERT(a, L), und
- INSERT*(a, L).

Für eine Strukturelle Induktion müssen wir dafür zunächst als Induktions Anfang den Satz für alle atomaren Strukturelemente beweisen, hier also für CREATE(). Danach etablieren wir die Induktions Hypothese und nehmen an, dass der Satz für eine beliebige aber feste Liste L gilt. Anschließend beweisen wir den Satz auch für nicht-atomare Strukturelemente, hier also für INSERT(a, L) und INSERT*(a, L), mithilfe der Induktions Hypothese.

Wir definieren die Funktion LENGTH wie folgt:

$$\begin{array}{lcl} \text{LENGTH(CREATE())} & = & 0 \\ \text{LENGTH(INSERT(}a, L\text{))} & = & 1 + \text{LENGTH(}L\text{)} \\ \text{LENGTH(INSERT*(}a, L\text{))} & = & 1 + \text{LENGTH(}L\text{)} \end{array}$$

Zusätzlich nutzen wir die aus der Vorlesung bekannte Funktion JOIN, welche wie folgt definiert ist:

$$\begin{array}{lcl} \text{JOIN(CREATE(), }z\text{)} & = & z \\ \text{JOIN(INSERT(}x, y\text{), }z\text{)} & = & \text{INSERT*(x, JOIN(}y, z\text{))} \\ \text{JOIN(INSERT*(}x, y\text{), }z\text{)} & = & \text{INSERT*(x, JOIN(}y, z\text{))} \end{array}$$

Beweisen Sie nun, dass für alle Listen L_1 und L_2 folgender Satz gilt:

$$\text{LENGTH(JOIN(}L_1, L_2\text{))} \quad = \quad \text{LENGTH(}L_1\text{)} + \text{LENGTH(}L_2\text{)}$$

Lösung

a) Wir zeigen die Aussage mittels vollständiger Induktion über n wie gefordert.

Für den Induktions Anfang haben wir mit $n = 1$, dass

$$\sum_{i=1}^{1} i = 1 = \frac{1 \cdot (1+1)}{2}.$$

Als Induktions Hypothese nehmen wir an, dass die Aussage für ein festes aber beliebiges n gelte.

Für den Induktions Schritt haben wir, dass

$$\begin{aligned}
&\sum_{i=1}^{n+1} i \\
&= n+1+\sum_{i=1}^{n} i \\
&= n+1+\frac{n\cdot(n+1)}{2} && \text{(Induktions Hypothese)} \\
&= \frac{2\cdot(n+1)}{2}+\frac{n\cdot(n+1)}{2} \\
&= \frac{2\cdot(n+1)+n\cdot(n+1)}{2} \\
&= \frac{(n+1)\cdot(n+2)}{2}
\end{aligned}$$

b) Wir zeigen die Aussage mittels struktureller Induktion über L_1 wie gefordert.

Für den Induktions Anfang haben wir mit $L_1 = \text{CREATE}()$ folgendes:

$$\begin{aligned}
&\text{LENGTH}(\text{JOIN}(L_1, L_2)) \\
&= \text{LENGTH}(\text{JOIN}(\text{CREATE}(), L_2)) \\
&= \text{LENGTH}(L_2) \\
&= 0 + \text{LENGTH}(L_2) \\
&= \text{LENGTH}(\text{CREATE}()) + \text{LENGTH}(L_2) \\
&= \text{LENGTH}(L_1) + \text{LENGTH}(L_2)
\end{aligned}$$

Als Induktions Hypothese nehmen wir an, dass die Aussage für ein festes aber beliebiges L_1 gelte.

Für den Induktions Schritt haben wir folgendes:

1. Falls $L_1' = \text{INSERT}(a, L_1)$:

$$\begin{aligned}
&\text{LENGTH}(\text{JOIN}(L_1', L_2)) \\
&= \text{LENGTH}(\text{JOIN}(\text{INSERT}(a, L_1), L_2)) \\
&= \text{LENGTH}(\text{INSERT*}(a, \text{JOIN}(L_1, L_2))) \\
&= 1 + \text{LENGTH}(\text{JOIN}(L_1, L_2)) \\
&= 1 + \text{LENGTH}(L_1) + \text{LENGTH}(L_2) && \text{(Induktions Hypothese)} \\
&= \text{LENGTH}(\text{INSERT}(a, L_1)) + \text{LENGTH}(L_2) \\
&= \text{LENGTH}(L_1') + \text{LENGTH}(L_2)
\end{aligned}$$

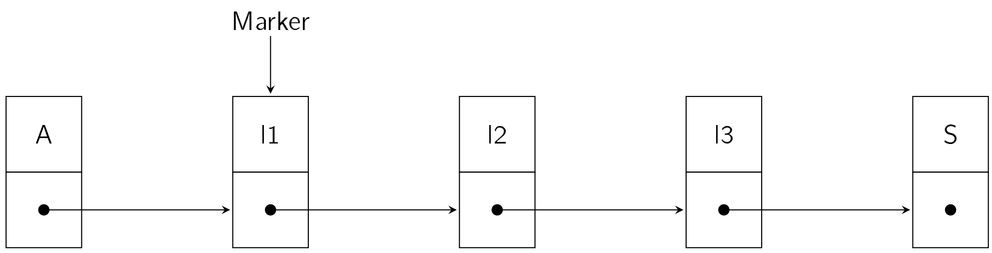
2. Falls $L_1' = \text{INSERT*}(a, L_1)$:

$$\begin{aligned}
& \text{LENGTH}(\text{JOIN}(L_1', L_2)) \\
= {} & \text{LENGTH}(\text{JOIN}(\text{INSERT*}(a, L_1), L_2)) \\
= {} & \text{LENGTH}(\text{INSERT*}(a, \text{JOIN}(L_1, L_2))) \\
= {} & 1 + \text{LENGTH}(\text{JOIN}(L_1, L_2)) \\
= {} & 1 + \text{LENGTH}(L_1) + \text{LENGTH}(L_2) && \text{(Induktions Hypothese)} \\
= {} & \text{LENGTH}(\text{INSERT*}(a, L_1)) + \text{LENGTH}(L_2) \\
= {} & \text{LENGTH}(L_1') + \text{LENGTH}(L_2)
\end{aligned}$$

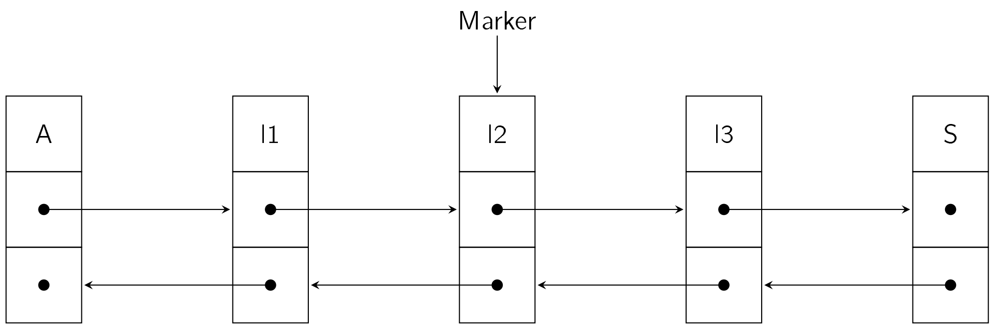
Tutoraufgabe 2 (Listen Operationen):

Wir stellen in dieser Aufgabe Listen, welche mittels Manipulation von Zeigern implementiert werden, wie unten dargestellt graphisch dar. Dabei signalisieren die Punkte · ein Zeiger, der Pfeil zu welchem Objekt der Zeiger zeigt und die Symbole über den Punkten welchen Inhalt das Listenelement hat. Das Anchor Element hat das Symbol A, das Sentinel Element hat das Symbol S. Der Marker Zeiger wird als eigenständiger, markierter Pfeil oben drüber illustriert.

a) Führen Sie erst eine Insert-Operation mit dem Element I4, dann eine Next-Operation und schließlich eine Delete-Operation durch. Geben Sie die Liste nach jeder Operation graphisch an.

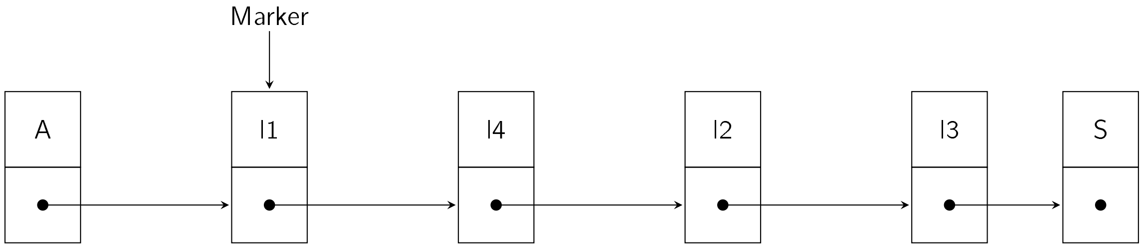


b) Führen Sie erst eine Next-Operation, dann eine Delete-Operation, dann eine Previous-Operation und schließlich eine Insert-Operation mit dem Element I4. Geben Sie die Liste nach jeder Operation graphisch an.

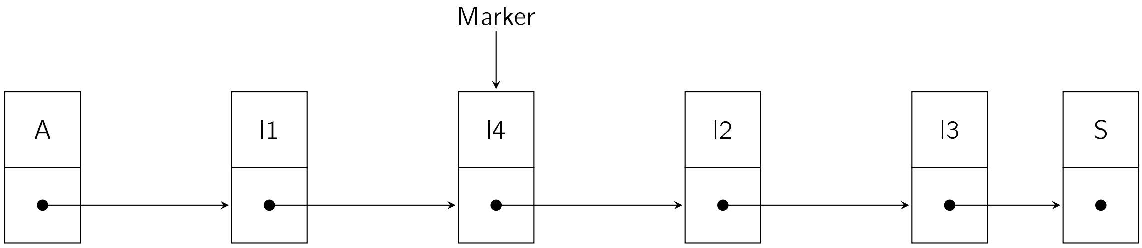


Lösung

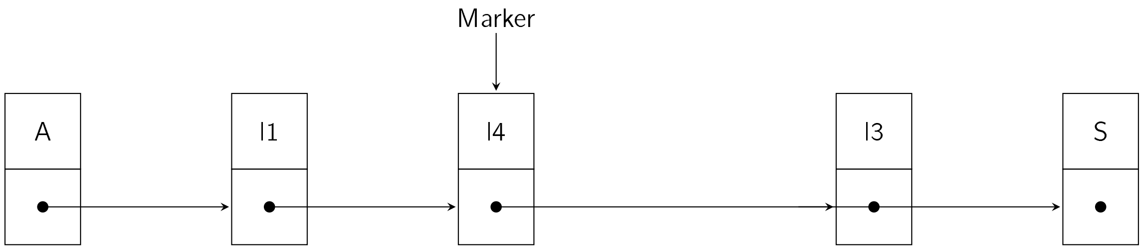
a) Nach der Insert-Operation mit dem Element I4:



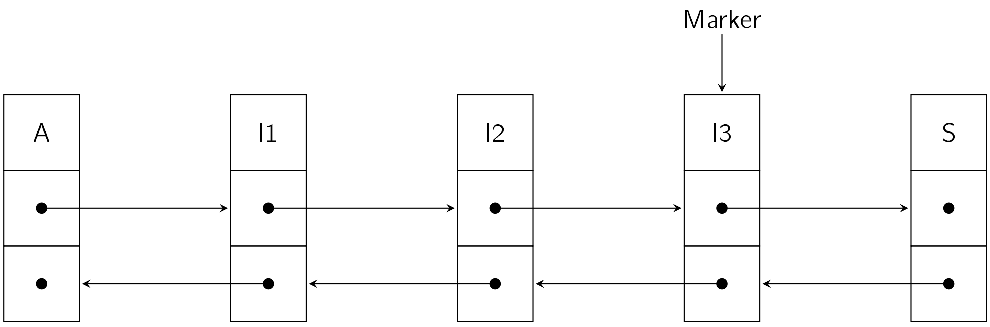
Nach der Next-Operation:



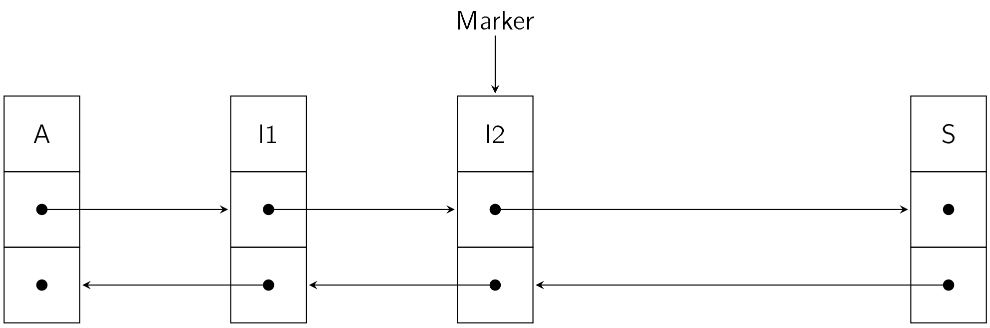
Nach der Delete-Operation:



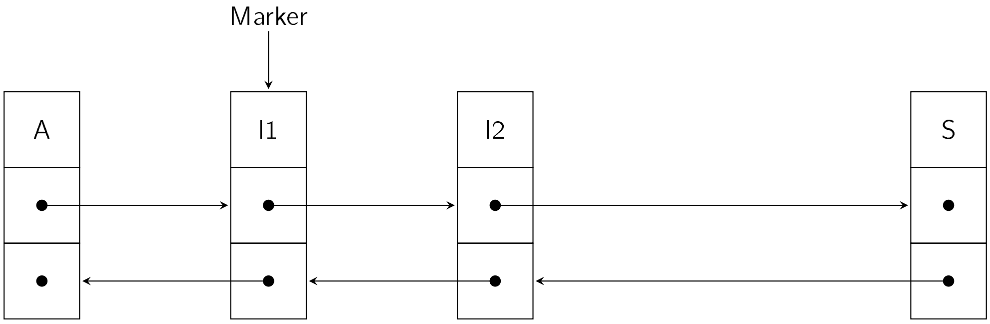
b) Nach der Next-Operation:



Nach der Delete-Operation:



Nach der Previous-Operation:



Nach der Insert-Operation mit dem Element I4: