Universität des Saarlandes
FR 6.2 - Informatik
Prof. Dr. W.J. Paul
Dipl.-Ing. Christoph Baumann

Schul-Subtraktionsalgorithmus Spezifikation und Beweis

Informatik II 2008

Aufgabe aus der Klausur: (Subtraktionsalgorithmus) **(10*+10* Punkte)**

Erinnere dich an den Schul-Algorithmus zur schriftlichen Subtraktion! Dieser soll im Folgenden untersucht werden. Wir betrachten die n-stelligen Dezimaldarstellungen $a, b \in \{0, \ldots, 9\}^n$ zweier natürlicher Zahlen $\langle a \rangle_Z, \langle b \rangle_Z < Z^n$. Der Algorithmus verwendet zusätzlich Überträge $c_i \in \{0, 1\}$ für $i \in \{0, \ldots, n\}$ und berechnet die Ziffern der Differenz $d[n-1:0] \in \{0, \ldots, 9\}^n$.

a) Spezifiziere den Schul-Subtraktionsalgorithmus, d.h. definiere die Differenzziffern und Überträge entsprechend dem bekannten Schema für schriftliche Subtraktion!
Definiere das Ergebnis des Subtraktionsalgorithmus!

b) Beweise, dass der Algorithmus das gewünschte Resultat liefert!
Wenn nötig, darf $\langle a \rangle_Z \geq \langle b \rangle_Z$ und $c_0 = 0$ angenommen werden.

Lösungsvorschlag

Im Folgenden sollen verschiedene Herangehensweisen an obige Aufgabenstellung erläutert werden. Dabei wird nicht ausgeschlossen, dass es weitere Möglichkeiten gibt, die Aufgabe effizient zu lösen. Die Kenntnis des Algorithmus zur schriftlichen Subtraktion aus der Schule wird vorausgesetzt.

Wir untersuchen zunächst wie die einzelnen Stellen d_i der Differenz berechnet werden. Im Falle $a_i > b_i$ werden die Zahlen einfach voneinander subtrahiert und man erhält ein positives Ergebnis. Allerdings muss auch ein eventueller Überträge c_i aus der Subtraktion der vorhergehenden Stelle beachtet werden. Man erhält daher für $a_i \geq b_i + c_i$

$$d_i = a_i - b_i - c_i$$

Diese Überträge können als negative carries interpretiert werden. Für den Fall $a_i < b_i + c_i$ würde das Ergebnis der Subtraktion negativ, weshalb wir uns eine Zehn *"borgen"* müssen. Das heißt, wir müssen zusätzlich noch Z addieren, damit wir auf einen positiven Wert für die Ziffer für die Differenz kommen.

$$d_i = a_i + Z - b_i - c_i$$

Der Übertrag c_{i+1} für die nächsthöhere Stelle ist stets dann 1, wenn man sich eine Zehn borgen musste.

$$c_{i+1} = \begin{cases} 1 & : \quad a_i < b_i + c_i \\ 0 & : \quad a_i \geq b_i + c_i \end{cases}$$

Zusammenfassend gilt nun für die Ziffern der Differenz:

$$d_i = \begin{cases} a_i + Z - b_i - c_i & : \quad a_i < b_i + c_i \\ a_i - b_i - c_i & : \quad a_i \geq b_i + c_i \end{cases}$$

Dies lässt sich mit Hilfe von c_{i+1} in eine Formel ohne Fallunterscheidung umwandeln.

$$d_i = a_i + c_{i+1} \cdot Z - b_i - c_i$$

Dann soll im Fall $\langle a[n-1:0]\rangle \geq \langle b[n-1:0]\rangle - c_0$ gelten:

$$\langle d[n-1:0]\rangle = \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle$$

Dies löst Teil a) der Aufgabenstellung, wobei $c_0 = 0$ angenommen werden kann. Zudem gilt für die Notation "$\langle\cdot\rangle \hat{=} \langle\cdot\rangle_Z$". Ob die obige Spezifikation zweckmäßig ist, wird der Beweis zeigen. Im Grunde genommen müsste dieser genauso funktionieren wie der des Schul-Additionsalgorithmus. Allerdings stößt man im Induktionsschritt von $n-1 \to n$ auf folgendes Problem.

Trennt man die obersten Bits der Operanden a_{n-1} und b_{n-1} mit Hilfe des Zerlegungslemmas ab und versucht dann die Induktionsvoraussetzung auf die unteren $n-1$ Bits anzuwenden, so würde man die Bedingung $\langle a[n-2:0]\rangle \geq \langle b[n-2:0]\rangle$ benötigen. Dies kann man aber nicht ohne Weiteres annehmen. Zumindest würde man dann die Korrektheit des Schulalgorithmus nur für Zahlenpaare $\langle a\rangle, \langle b\rangle$ zeigen, bei denen $\forall i : a_i \geq b_i$ gilt.

Man kann dieses Dilemma umgehen, indem man das Ergebnis auch für den Fall $\langle a\rangle < \langle b\rangle$ definiert. Dieses lautet nämlich, wie man leicht nachprüfen kann,

$$\langle d[n-1:0]\rangle = \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle - c_0 + Z^n.$$

Dann müsste man allerdings im Beweis eine Fallunterscheidung über den Vergleich von $\langle a[n-2:0]\rangle$ mit $\langle b[n-2:0]\rangle$ einführen. Zudem müsste man auch den Übertrag c_n betrachten und diverse Lemmata beweisen, die Schlüsse von $\langle a[n-2:0]\rangle$ und $\langle b[n-2:0]\rangle$ auf c_n zulassen. Dies alles lässt sich durch einen simplen Trick umgehen

Beim Induktionsschritt spaltet man einfach nicht - wie beim Beweis des Additionsalgorithmus und auch sonst üblich - das vorderste Bit der Operanden ab, sondern das letzte, niederwertigste Bit a_0 bzw. b_0. Dann kann auch für die Induktionsvoraussetzung $\langle a[n-1:1]\rangle \geq \langle b[n-1:1]\rangle$ angenommen werden, denn es gilt:

$$\forall i \in \{0,\ldots,n-1\}: \quad \langle a[n-1:0]\rangle \geq \langle b[n-1:0]\rangle + c_0 \Rightarrow \langle a[n-1:j]\rangle \geq \langle b[n-1:j]\rangle + c_j$$

Dies kann man leicht per Induktion über n zeigen. Der Beweis des Subtraktionsalgorithmus lautet dann wie folgt.

Beweis: per vollständige Induktion über n, wenn $\langle a[n-1:0]\rangle \geq \langle b[n-1:0]\rangle + c_0$, dann gilt:

$$\langle d[n-1:0]\rangle = \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle + c_0$$

Induktionsanfang: $n = 1$, wegen $\langle a\rangle \geq \langle b\rangle + c_0$ gilt $a_0 \geq b_0 + c_0$.

$$\begin{aligned}
\langle d[n-1:0]\rangle &= \langle d[0:0]\rangle \\
&= d_0 && \text{(Zahlendarstellung)} \\
&= a_0 - b_0 - c_0 && \text{(Definition } d_0,\ c_1 = 0) \\
&= \langle a[0:0]\rangle - \langle b[0:0]\rangle - c_0 \\
&= \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle - c_0 && \checkmark
\end{aligned}$$

Induktionsvoraussetzung: wenn $\langle a[n-1:1]\rangle \geq \langle b[n-1:1]\rangle + c_1$, dann gilt:

$$\langle d[n-1:1]\rangle = \langle a[n-1:1]\rangle - \langle b[n-1:1]\rangle + c_1$$

Induktionsschritt: $n-1 \to n$, für $\langle a[n-1:0]\rangle \geq \langle b[n-1:0]\rangle + c_0$ gilt:

$$\begin{aligned}
&\langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle - c_0 \\
&= \langle a[n-1:1]\rangle \cdot Z + a_0 - \langle b[n-1:1]\rangle \cdot Z - b_0 - c_0 && \text{(Zerlegungslemma)} \\
&= (\langle a[n-1:1]\rangle - \langle b[n-1:1]\rangle) \cdot Z + a_0 - b_0 - c_0 && \text{(Distributivgesetz)} \\
&= (\langle a[n-1:1]\rangle - \langle b[n-1:1]\rangle) \cdot Z + a_0 - b_0 + c_1 \cdot Z - c_1 \cdot Z - c_0 && \text{(Addition mit Null)} \\
&= (\langle a[n-1:1]\rangle - \langle b[n-1:1]\rangle - c_1) \cdot Z + a_0 - b_0 + c_1 \cdot Z - c_0 && \text{(Distributivgesetz)} \\
&= (\langle d[n-1:1]\rangle) \cdot Z + a_0 + c_1 \cdot Z - b_0 - c_0 && \text{(Induktionsvoraussetzung)} \\
&= (\langle d[n-1:1]\rangle) \cdot Z + d_0 && \text{(Definition } d_0) \\
&= \langle d[n-1:0]\rangle && \text{(Zerlegungslemma)} \quad \square
\end{aligned}$$

Nun ist dies ein recht spezieller Beweis. Es wäre wünschenswert, eine allgemeinere Aussage über $\langle d[n-1:0]\rangle$ ohne Anforderungen an die Operanden und Überträge treffen zu können. Dazu betrachten wir erneut die Definition von Stelle d_i der Differenz. Subtrahiert man auf beiden Seiten $c_{i+1} \cdot Z$ so erhält man:

$$d_i - c_{i+1} \cdot Z = a_i - b_i - c_i$$

Das "Borgen" wird somit auf die linke Seite verschoben und man erhält das genaue, eventuell negative Ergebnis der Subtraktion auf der rechten Seite. Bei genauerer betrachtung fällt auch auf, dass die obige Formel ausreicht, den Subtraktionsalgorithmus komplett zu spezifizieren. Die linke Seite der Gleichung kann nur negativ werden für $c_{i+1} = 1$. c_{i+1} muss also 1 sein, falls $a_i - b_i - c_i < 0$, dies ist aber genau dann der Fall, wenn $a_i < b_i + c_i$. Dies entspricht widerum unserer separaten Definition von c_{i+1}. Ist $a_i \geq b_i + c_i$, so ist die rechte Seite positiv und $c_{i+1} = 0$. Dann ist $d_i = a_i - b_i - c_i$, ansonsten $d_i = a_i + Z - b_i - c_i$. Für das Ergebnis des Subtraktionsalgorithmus $d[n-1:0]$ gilt mit $c_0 = 0$ allgemein

$$\langle d\rangle = \langle a\rangle - \langle b\rangle + c_n \cdot Z^n$$

Auch diese Gleichung lässt sich entsprechend umstellen und wir erhalten unsere Induktionsbehauptung.

Beweis: durch vollständige Induktion über n, Behauptung:

$$\langle d[n-1:0]\rangle - c_n \cdot Z^n = \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle$$

Induktionsanfang: $n = 1$

$$\begin{aligned}
\langle d[n-1:0] - c_n \cdot Z^n\rangle &= d_0 - c_1 \cdot Z \\
&= a_0 - b_0 - c_0 && \text{(Algorithmus, } i = 0\text{)} \\
&= \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle - c_0 && (c_0 = 0) \quad \checkmark
\end{aligned}$$

Induktionsvoraussetzung: $\langle d[n-2:0]\rangle - c_{n-1} \cdot Z = \langle a[n-2:0]\rangle - \langle b[n-2:0]\rangle$
Induktionsschritt: $n-1 \to n$

$$\begin{aligned}
&\langle d[n-1:0]\rangle - c_n \cdot Z^n \\
&= d_{n-1} \cdot Z^{n-1} - c_n \cdot Z^n + \langle d[n-2:0]\rangle && \text{(Zerlegungslemma)} \\
&= (d_{n-1} - c_n \cdot Z) \cdot Z^{n-1} + \langle d[n-2:0]\rangle && \text{(Distributivgesetz)} \\
&= (d_{n-1} - c_n \cdot Z) \cdot Z^{n-1} + \langle d[n-2:0]\rangle - c_{n-1} \cdot Z^{n-1} + c_{n-1} \cdot Z^{n-1} && \text{(Addition mit Null)} \\
&= (d_{n-1} - c_n \cdot Z) \cdot Z^{n-1} + \langle a[n-2:0]\rangle - \langle b[n-2:0]\rangle + c_{n-1} \cdot Z^{n-1} && \text{(Induktionsvoraussetzung)} \\
&= (a_{n-1} - b_{n-1} - c_{n-1}) \cdot Z^{n-1} + \langle a[n-2:0]\rangle - \langle b[n-2:0]\rangle + c_{n-1} \cdot Z^{n-1} && \text{(Algorithmus, } i = n-1\text{)} \\
&= a_{n-1} \cdot Z^{n-1} + \langle a[n-2:0]\rangle - (b_{n-1} \cdot Z^{n-1} + \langle b[n-2:0]\rangle) + (c_{n-1} - c_{n-1}) \cdot Z^{n-1} && \text{(Distributivgesetz)} \\
&= \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle && \text{(Zerlegungslemma)} \quad \square
\end{aligned}$$

Somit haben wir den Subtraktionsalgorithmus ohne Einschränkungen verifiziert.

Die Spezifikation erinnert dabei an die Form des Additionsalgorithmus. Zum Vergleich:

$$\begin{aligned}
c_{i+1} \cdot Z + s_i &= a_i + b_i + c_i && \text{Additionsalgorithmus} \\
\langle c_n s\rangle &= \langle a\rangle + \langle b\rangle + c_0 \\
-c_{i+1} \cdot Z + d_i &= a_i - b_i - c_i && \text{Subtraktionsalgorithmus} \\
-c_n \cdot Z^n + d &= \langle a\rangle - \langle b\rangle - c_0
\end{aligned}$$

Die linken Seiten der Gleichungen zum Subtraktionsalgorithmus erinnern zudem an die Struktur von Two's-Complement-Zahlen. In der Tat kann man eine Art "*Ten's-Complement*"-Darstellung definieren, welche die Spezifikation und den Beweis noch etwas eleganter macht. Wir definieren:

$$[c_n\ d[n-1:0]]_Z = -c_n \cdot Z^n + \langle d[n-1:0]\rangle$$

Damit verwandelt sich die Spezifikation in:

$$\begin{aligned}
[c_{i+1}\ d_i]_Z &= a_i - b_i - c_i \\
[c_n\ d[n-1:0]]_Z &= \langle a[n-1:0]\rangle - \langle b[n-1:0]\rangle - c_0
\end{aligned}$$