

Prof. Dr. Armin Biere
Dr. Mathias Fleury

Freiburg, 9. Mai 2024

Technische Informatik Übungsblatt 4 (v3)

Achtung 1: Diese Abgabe *muss* mit dem Computer geschrieben worden sein. Handgeschriebene Antworten werden *nicht* benotet.

Achtung 2: In der zweiten Übung wird ReTi gefragt. Es *muss* so abgegeben werden, dass der Tutor mit copy-paste es in einem Emulator einfügen kann (es gibt das von letzter Woche mit einem Bug, diesen <https://reti.gim.one/> im Browser, oder den von Armin – siehe letzte Aufgabe).

Noch ein Anmerkung zur <https://reti.gim.one/>: Die nutzt eigentlich ReTi aus Betriebssysteme darum sind die STOREIN/LOADIN Befehle etwas anders definiert: Bei uns hier ist es STOREINj i bzw. LOADINj D i. In Betriebssysteme/reti.gim.one wird die etwas generelle Version LOADIN S D i bzw. STOREIN D S i benutzt. STORE ist auch anders definiert mit STORE D i statt STORE i.

Aufgabe 1 (2 + 3 Punkte)

- a) Gegeben sei $a = a_n a_{n-1} \dots a_0 a_{-1} \dots a_{-k}$. Geben Sie den Wert von a als Einerkomplementzahl mit k Nachkommastellen und als Zweierkomplementzahl mit k Nachkommastellen an.

Hinweis: Für $a_n = 1$ ergibt sich der Wert einer Einerkomplementzahl analog zu dem in der Vorlesung vorgestellten Fall $k = 0$ aus dem entsprechenden Wert für $a_n = 0$ durch Subtraktion der größten darstellbaren Zahl mit $n + 1$ Vorkommastellen und k Nachkommastellen. Um den Wert einer Zweierkomplementzahl zu erhalten, wird bei $a_n = 1$ im Vergleich zum Einerkomplement eine entsprechend größere Zahl subtrahiert, um zu verhindern, dass es zwei verschiedene Darstellungen für die 0 gibt.

- b) Stellen Sie die folgenden Zahlen als Zweierkomplementzahlen mit insgesamt 8 Bit und *zwei Nachkommastellen* dar.
- 1) Die Dezimalzahl $-1,0_{10}$
 - 2) Die Dezimalzahl $2,25_{10}$
 - 3) Die Hexadezimalzahl $-2,4_{16}$
 - 4) Die größte darstellbare Zahl.
 - 5) Die größte darstellbare Zahl, die kleiner als 1 ist.
 - 6) Die größte darstellbare Zahl, die kleiner als 0 ist.

Aufgabe 2 (8 Punkte)

Schreiben Sie ein Programm für den ReTI Rechner, das den größten gemeinsamen Teiler von $x = M(40)$ und $y = M(41)$ berechnet. Das Ergebnis soll in $M(42)$ gespeichert werden.

Erläutern Sie Ihr Programm mit Kommentaren.

Führen Sie zusätzlich Ihren Programmablauf am Beispiel $x = 4, y = 2$ vor.

Hinweis: $\text{ggT}(x, y) = \text{ggT}(\max(x, y) - \min(x, y), \min(x, y))$ und $\text{ggT}(x, 0) = x$

Aufgabe 3 (2 + 5 Punkte)

In dieser Übung benutzen wir wieder die Fibonacci sequence $f[n]$.

Betrachten wir mal ein leicht anderen Version von dem Algorithmus von der Übung 2

$$\begin{aligned}\text{ggT}(x, 0) &= x \\ \text{ggT}(a, b) &= \text{ggT}(a \bmod b, b) \text{ sonst wenn } a \geq b \\ \text{ggT}(a, b) &= \text{ggT}(b \bmod a, a) \text{ sonst wenn } a < b\end{aligned}$$

- a) Erklären in 3-4 Sätzen warum dieser Algorithmus und der von Übung 2 das selbe rechnen.
- b) Zeigen Sie per Induktion, dass $\text{ggT}(f[n+1], f[n])$ exakt n Schritte braucht um berechnet zu werden.

Es ist übrigens möglich zu zeigen, dass $a \geq f[n+1]$ und $b \geq f[n]$ reicht damit der Algorithmus mindestens n Schritte braucht, aber diese Induktion bleibt euch erspart.

Aufgabe 4 (2 + 5 Punkte)

In dieser Übung benutzen wir wieder die Fibonacci sequence $f[n]$.

Betrachten wir mal ein leicht anderen Version von dem Algorithmus von der Übung 2

$$\begin{aligned}\text{ggT}(x, 0) &= x \\ \text{ggT}(a, b) &= \text{ggT}(a \bmod b, b) \text{ sonst wenn } a \geq b \\ \text{ggT}(a, b) &= \text{ggT}(b \bmod a, a) \text{ sonst wenn } a < b\end{aligned}$$

- a) Erklären in 3-4 Sätzen warum dieser Algorithmus und der von Übung 2 das selbe rechnen.
- b) Zeigen Sie per Induktion, dass $\text{ggT}(f[n+1], f[n])$ exakt n Schritte braucht um berechnet zu werden.

Es ist übrigens möglich zu zeigen, dass $a \geq f[n + 1]$ und $b \geq f[n]$ reicht damit der Algorithmus mindestens n Schritte braucht, aber diese Induktion bleibt euch erspart.

Aufgabe 5 (2 Punkte)

Nächste Woche werden wir im Übungsblatt Reti Aufgaben haben, mit dem Tool was Armin vorgestellt hat in der Vorlesung. Um die Vorzubereiten, müssen wir manche Tools vorbereiten. Das ist nicht besonders schwer, aber es braucht etwas Zeit, darum kommt die Installation in den Übung diese Woche schon vor.

Diese Installation ist leicht System-abhängig

Linux/BSD das sollte es nichts zu tun geben, das wir nur ein Compiler brauchen. Unter Ubuntu müssen Sie `sudo apt install build-essential` ausführen. Auf anderen Distributionen: gcc und git installieren.

macOS Da müssen Sie im Terminal `xcode-select -install` ausführen um den Compiler zu installieren (dauert mehrere Minuten und holt sich mehrere Gigabyte von den Apple Servern runter).

Windows Da werden wir einfach Ubuntu in WSL installieren. Folgen Sie einfach <https://learn.microsoft.com/de-de/windows/wsl/install> mit Ubuntu als <Distribution Name>. Danach können Sie einfach WSL öffnen und dann `sudo apt install build-essential` ausführen.

Ab jetzt ist es für alle System gleich: ein Terminal öffnen (WSL terminal für Windows), dort dann das Repository clonen (also runterladen mit dem git Tool):

```
git clone https://github.com/arminbiere/reticode.git
```

Wenn das gemacht ist, einfach in dem Verzeichnis rein gehen:

```
cd reticode
```

und das Program kompilieren:

```
./configure && make test
```

Wenn das ohne Fehlermeldung läuft, schreiben einfach: "Es kompiliert!" als Abgabe.

Abgabe: 16. Mai 2024, 17⁰⁰