

Grundkurs Informatik – Das Übungsbuch

Errata, 2. Auflage 2020

Jochen Schmidt

Stand: 31. August 2023

Die Fehler sind umgekehrt chronologisch nach Aufnahme in dieses Dokument geordnet, um den Lesern das Auffinden von Ergänzungen zu erleichtern. Die neuesten Änderungen stehen damit immer am Anfang des Dokuments.

S. 58, Aufgabe 3.15. Bei der Rechnung mit mod 97 reduzierten Gewichten sind alle Gewichte außer bei der Ziffer ganz rechts um eine Stelle verschoben, das Gewicht 10 fehlt. Korrekt ist:

$$\begin{aligned} &56 \cdot 5 + 25 \cdot 6 + 51 \cdot 7 + 73 \cdot 8 + 17 \cdot 1 + 89 \cdot 2 + 38 \cdot 3 + 62 \cdot 4 + 45 \cdot 1 + 53 \cdot 2 + 15 \cdot 3 + \\ &50 \cdot 4 + 5 \cdot 5 + 49 \cdot 6 + 34 \cdot 7 + 81 \cdot 8 + 76 \cdot 9 + 27 \cdot 0 + 90 \cdot 1 + 9 \cdot 3 + 30 \cdot 1 + 3 \cdot 4 + \\ &10 \cdot 9 + 1 \cdot 1 \pmod{97} = 1. \end{aligned}$$

S. 93, Aufgabe 6.13b: Die akzeptierte Sprache in Mengenschreibweise ist: $L = \{2^i 0^j 1^k 2^k \mid i, j, k \in \mathbb{N}, j \geq 2\}$

S. 94, Aufgabe 6.13d: Die Regel für M muss ersetzt werden durch: $M \rightarrow 0M1 \mid 0011$

S. 92, Aufgabe 6.12c, S. 98, Aufgabe 6.17c, Schritt 3 sollte lauten: „Schritt 3: Ersetzen von Produktionen mit einer einzigen Variablen auf der rechten Seite durch die rechte Seite der Zielvariable ist hier nicht erforderlich.“

S. 96, Aufgabe 6.15e, Schritt 3 sollte lauten: „Schritt 3: Ersetzen von Produktionen mit einer einzigen Variablen auf der rechten Seite durch die rechte Seite der Zielvariable:“

S. 101, Aufgabe 6.21c, Schritt 3 sollte lauten: „Schritt 3: Ersetzen von Produktionen mit einer einzigen Variablen auf der rechten Seite durch die rechte Seite der Zielvariable:“.

Die neue Regel für die Variable A muss lauten:

$$A \rightarrow V_a B \mid V_b B \mid V_c B \mid a \mid b \mid c \mid AA_1 \mid AA_2 \mid OA_3$$

S. 106, Aufgabe 7.1: Es fehlt an einigen Stellen bei der Zuweisung $x_0 := 0$; der Doppelpunkt.

S. 107, Aufgabe 7.1e: Es fehlt ein ELSE im LOOP-Programm. Hier die korrigierte Lösung:

Modulo $x_0 := x_1 \bmod x_2$:

$x_0 := 0$;

LOOP x_1 DO

 IF $x_2 > x_1$ THEN $x_0 := x_1$

 ELSE $x_1 := x_1 - x_2$ END;

END

Im Programm wurde der Ausdruck IF $x_2 > x_1$ THEN P_1 ELSE P_2 END verwendet, von dem noch nachgewiesen muss, dass er LOOP-berechenbar ist:

```

x3 := x2 - x1;
x4 := 0;
x5 := 1;
LOOP x3 DO x4 := 1; x5 := 0 END;
LOOP x4 DO P1 END;
LOOP x5 DO P2 END

```

Kombiniert:

```

x0 := 0;
LOOP x1 DO
  x3 := x2 - x1;
  x4 := 0;
  x5 := 1;
  LOOP x3 DO x4 := 1; x5 := 0 END;
  LOOP x4 DO x0 := x1 END;
  LOOP x5 DO x1 := x1 - x2 END
END

```

END

Eine weitere Möglichkeit zur Darstellung der Modulo-Operation wäre die Verwendung der Ergebnisse aus den Aufgaben (a), (c) und (d) (Subtraktion, Multiplikation bzw. Division). Damit ergibt sich das LOOP-Programm $x_0 := x_1 - (x_1/x_2) * x_2$, oder ohne Klammern:

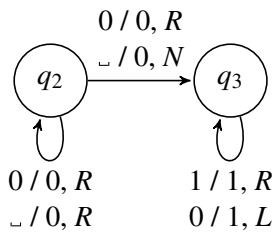
```

x0 := x1/x2;
x0 := x0 * x2;
x0 := x1 - x0

```

S. 98, Aufgabe 6.17c: Die Produktionsregel für die neu eingeführte Variable C muss überall lauten: $C \rightarrow 3$. (In Aufgabe (d) ist sie dann korrekt verwendet, hier ist keine Korrektur nötig).

S. 17, Aufgabe 6.13: Der Übergang zwischen Zustand q_2 und q_3 ist für Band 2 falsch beschriftet:



S. 39, Aufgabe 1.12: Kein Fehler im eigentlichen Sinn, aber bei $43 - 11$ im Zweierkomplement sollte der Überlauf gezeigt und gestrichen sein:

```

43 - 11:
  00101011    43
+ 11110101   -11
-----
= 100100000  32

```

Bei $43 - 11$ im Einerkomplement ist im Zwischenergebnis der ersten Addition eine 1 zu viel, es muss heißen:

```

43 - 11:
  00101011    43
+ 11110100   -11
-----
= 100011111  Überlauf
→ 00011111
+           1
-----
= 00100000    32

```

S. 3, Aufgabe 1.1: Die vierte Zahl muss Basis 5 sein (nicht 3), also 1234_5

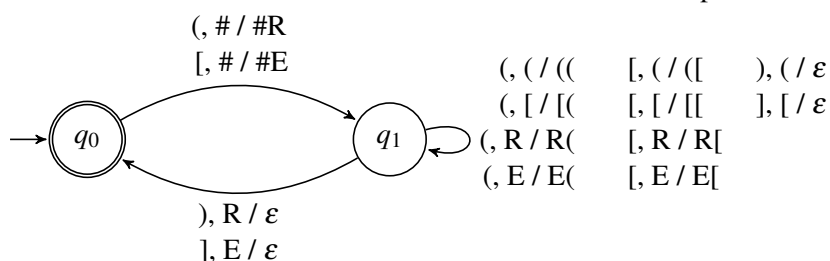
S. 35, Aufgabe 1.1: Die Rechnung für die vierte Zahl lautet dann:

$$1234_5 = 1 \cdot 5^3 + 2 \cdot 5^2 + 3 \cdot 5^1 + 4 \cdot 5^0 = 194_{10}$$

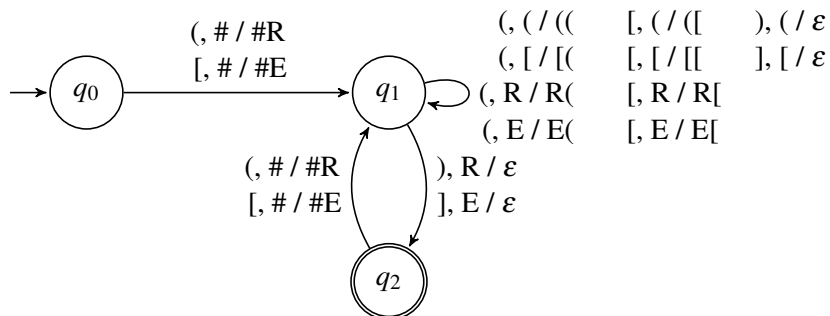
S. 24/115/116, Aufgaben 7.16 und 7.17: Der Schleifenzähler i muss jeweils bei 1 starten (statt bei 0).

S. 93, Ende Aufgabe 6.12d: Die beiden letzten Kellerautomaten sind nicht äquivalent zu den vorherigen, sie akzeptieren mehr Wörter. Eine korrekte Lösung wäre:

Da bei den Kellerautomaten, die über Endzustand akzeptieren, das Löschen des Kellersymbols # nicht erforderlich ist, kann man diese auch auf einfache Weise ohne ϵ -Übergänge und deterministisch formulieren. Dazu führen wir zwei zusätzliche Kellersymbole R und E ein, die die erste runde bzw. eckige Klammer markieren. Hier der Kellerautomat inkl. leerem Wort in der Sprache:



Und der Automat ohne leeres Wort:



S. 20 und S. 104, Aufgabe 6.22c: p ist hier überflüssig.

S. 104, Aufgabe 6.23a: Es fehlt in der zweiten Aufzählung „Beim Pumpen entstehen in den obigen Fällen“: (5) nur neue 2

S. 24, Abschnitt 7: Es fehlt eine Aufgabe (Nummer 7.18); in den Lösungen ist diese auf S. 116 enthalten, wodurch sich die Nummerierung der folgenden Aufgabenlösungen um eins verschiebt. Die Aufgabe lautet: Bestimmen Sie die Komplexität des folgenden Codeausschnitts in \mathcal{O} -Notation (n Datenmenge) für den besten und den schlechtesten Fall (bzgl. der Struktur der Daten):

```
void sort(int *a, int n){
    int s = 1;
    while(s == 1){
        s = 0;
        for(int i = 1; i < n; i++){
            if(a[i] < a[i - 1]) {
                int t = a[i];
                a[i] = a[i - 1];
                a[i - 1] = t;
                s = 1;
            }
        }
    }
}
```

}
}
}

S. 109, Aufgabe 7.3d: Es muss überall die Vorgängerfunktion v durch die Nachfolgerfunktion s ersetzt werden.