

Test und Verlässlichkeit, Übungsblatt 9 (10P)

Prof. G. Kemnitz, TU Clausthal, Institut für Informatik

19. Juni 2015

Aufgabe 9.1

Eine Festkommazahl soll aus einer bis drei Dezimalziffern¹ vor und genau zwei Ziffern nach dem Komma bestehen. Vor jeder Zahl dürfen beliebig viele Trennzeichen stehen (0x20 (Leerzeichen), 0xa (Zeilenumbruch) oder 0xd (Wagenrücklauf)) und nach der Zahl muss genau ein Trennzeichen folgen.

- Beschreiben Sie dieses Eingabeformat als formale Sprache mit den Ersetzungsregeln »...|...«, »[...]« und »{...}«. 1P
- Beschreiben Sie den spracherkennenden Automaten als Graph. 2P
- Entwerfen Sie in C eine Parser-Funktion 4P

```
void parsMsg(unsigned char c);
```

zur Nachbildung des Automaten. Private globale Zustandsvariablen:

```
unsigned char state; // Automatenzustand
float wert;         // Wert der Festkommazahl
```

Testrahmen:

```
int main(){
    unsigned char i, dat[]="fg 3.45 56.16864.2 222.22 ";
    state = 0;
    printf("c: state val\n");
    for (i=0;i<16;i++){
        parsMsg(dat[i]);
        printf("%c: %i %c %8.2i\n", dat[i], state, wert);
    }
}
```

So soll etwa die Testausgabe aussehen (Zustandskodierung darf anders sein und die Kommentare in Klammern am Zeilenende sind keine notwendigen Programmausgaben:

```
c: state wert
f: F 0.00 (F -- falsches Zeichen empfangen)
g: F 0.00
: T 0.00 (T -- Trennzeichen empfangen)
3: VK1 3.00 (VK1 -- Vorkommast. 1 empf.)
.: Pkt 3.00 (Pkt -- Punkt empfangen)
4: NK1 3.40 (NK1 -- Nachkommast. 1 empf.)
5: NK2 3.45 (NK2 -- Nachkommast. 2 empf.)
: T 0.45 (Werteausgabe vor Zustandsübergang)
```

- Geben Sie ein Listing der Programmausgabe mit ab. 2P

¹Darstellung als ASCII-Zeichen '0'→0x30, ..., '9'→0x39, ','→0x2C

Aufgabe 9.2

- a) Bilden Sie für den auf Foliensatz F3 entwickelten (8,12)-Hamming-Code mit den Kontrollstellen

$$q_0 = x_0 \oplus x_1 \oplus x_3 \oplus x_4 \oplus x_6$$

$$q_1 = x_0 \oplus x_2 \oplus x_3 \oplus x_5 \oplus x_6$$

$$q_2 = x_1 \oplus x_2 \oplus x_3 \oplus x_7$$

$$q_3 = x_4 \oplus x_5 \oplus x_6 \oplus x_7$$

und der Bitzuordnung

b_{12}	b_{11}	b_{10}	b_9	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
x_7	x_6	x_5	x_4	q_3	x_3	x_2	x_1	q_2	x_0	q_1	q_0

die Codeworte für die darzustellenden Werte: 0xA4 und 0x3C.

2P

- a) Handelt es sich bei den Codeworten 0xD12, 0x91A und 0x041 um zulässige Codeworte, Codeworte mit korrigierbaren oder Codeworte mit erkenn- aber nicht korrigierbaren Verfälschungen? Wenn sie korrigierbar sind, wie lauten die zugeordneten korrekten Werte? 3P