ulm university universität
uulm

Institut für Numerische Mathematik

Dr. Andreas F. Borchert
Fabian Berstecher

10. November 2017
Blatt 4

Systemnahe Software I (WS 2017/2018)

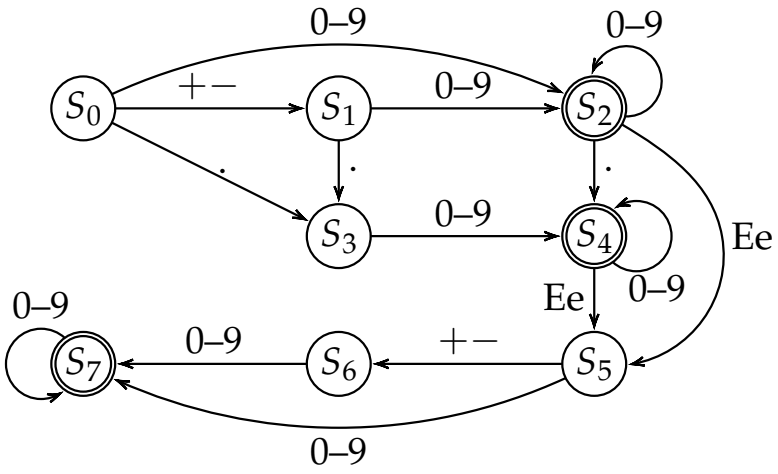
Abgabe bis zum 17. November 2017, 14:00 Uhr

Lernziele:

- Programmierung eines endlichen Automaten
- Summationsalgorithmus von Kahan

Aufgabe 6: Gleitkommazahlen aus der Eingabe aufsummieren

Die Syntax von Gleitkommazahlen lässt sich in C durch den folgenden endlichen Automaten spezifizieren. Der Anfangszustand ist S_0, zulässige Endzustände sind S_2, S_4 und S_7:



Hier sind einige Beispiele, die von dem Automaten erkannt werden:

1. Ganze Zahlen: 42 oder -128 oder 0 (endet in S_2)
2. Gleitkommazahlen: 3.141592 oder -0.10 oder $+.66$ (endet in S_4)
3. Wissenschaftliche Notation: $1E4$ oder $3141592e-6$ oder $-9.051E10$ (endet in S_7)

Im Rahmen dieser Aufgabe ist ein entsprechender Automat zu implementieren, der die Eingabe zeichenweise verarbeitet. Wenn immer eine gültige Gleitkommazahl erkannt ist, ist diese in eine **double** zu konvertieren und aufzusummieren. Dabei sind beliebige andere Zeichenfolgen innerhalb der Eingabe zulässig. Diese werden dann einfach überlesen.

Zum Einlesen der Daten ist nur die Funktion *getchar()* erlaubt. Die Konvertierung einer Zeichensequenz in einen Wert des Typs **double** muss selbst implementiert werden, d.h. Funktionen aus der Standardbibliothek wie beispielsweise *strtod* sind hierfür ebenso nicht zulässig.

Um den numerischen Fehler beim Aufsummieren zu minimieren, soll der Summationsalgorithmus von Kahan verwendet werden. Dessen Resultat ist mit dem einer trivialen Summation zu vergleichen. Hier ist ein beispielhafter Aufruf:
Hinweise: Zu beachten ist, dass der Rückgabewert von *getchar()* vom Typ **int** ist, und nicht vom Typ **char**. Die Funktion *ungetc()* ist eine nützliche Ergänzung, die Sie benötigen, um z.B. aus der Eingabe „1e2.7x“ die beiden Gleitkommazahlen „1e2“ und „.7“ zu extrahieren.

Zum Potenzieren können Sie die Funktion *pow()* verwenden. Diese wird von der Mathe-Bibliothek bereitgestellt. Um die Funktion verwenden zu können, müssen Sie auf der Thales die Mathe-Bibliothek mit „-lm“ dazubinden. Beispielsweise so:

```
gcc -std=gnu11 -Wall -o sum sum.c -lm
```

Damit die Ausgabe am Ende reproduzierbar ist, sollte sie folgendermaßen erfolgen, wobei *sum* die triviale Summe ist und *kahan_sum* die Summe entsprechend dem Kahanschen Algorithmus:

printf("Regular␣sum:␣%.17lg\n", *sum*);
printf("Kahan␣sum:␣␣␣%.17lg\n", *kahan_sum*);
printf("Difference:␣␣%lg\n", *fabs*(*sum* − *kahan_sum*));

Reichen Sie bitte Ihre Lösung mit folgendem Kommando ein:

```
thales$ submit ss1 6 sum.c
```

Viel Erfolg!