

Fliesskommazahlen für Schüler der Gymnasialstufe

amaximov

März 2020

Inhaltsverzeichnis

1 Einführung	1
2 Fliesskommazahlen	2
2.1 Kleinste und grösste positive Zahlen	2
2.2 Darstellbare Zahlen	3
3 Addition	4
4 Zusammenfassung	5
5 Beispiellösungen	6

1 Einführung

Wie können Computer so viele unterschiedliche Dinge darstellen, wie Text, Bilder und Videos, wenn sie nur Nullen und Einser speichern können? Wir haben schon gesehen, wie Computer ganze Zahlen darstellen können und letzte Woche hatten wir angefangen zu sehen, wie Computer reelle Zahlen approximieren können. Die Grundidee, wie wir schon gesehen haben, ist im Grunde genommen die Exponentialschreibweise, die wir schon aus der Chemie und aus der Physik kennen. Der Unterschied ist, dass Computer meistens in Basis 2 arbeiten und schränken die Anzahl der möglichen Exponenten und signifikanten Stellen ein.

Um diese Einschränkungen sichtbar und anfassbar zu machen, werden wir folgende Analogie nutzen. Die Mantisse ist wie ein Wagen. Der Fahrer muss eine Eins sein. Der Wagen hat eine bestimmte Anzahl Sitzplätze. Die Zahlen, die da nicht reinpassen, werden nicht gespeichert. Der Wagen ist mobil und kann vom Universalkomma (alle wissen ja, dass im Zentrum vom Universum das Universalkomma steht, an welchem alle Zahlen dranhängen) zum Speicher gefahren werden. Aber damit man immer weiss, welche Zahl ursprünglich gemeint war, muss man auch wissen, wo der Wagen bezüglich dem Universalkomma stehen soll. Dazu ist ein Nagel zwischen dem Fahrersitz und dem ersten Passagiersitz

eingehämmert und daran ist ein Seil gebunden. Dieser Seil hat zwei Enden: das negative Ende (nach links) und das positive Ende (nach rechts). Am Seil wird markiert, wie weit das Universalkomma ist.

Beispiel 1.1 *Wir werden jetzt zusammen die Zahl 6.5 darstellen.*

Am Universalkomma hängt die Zahl dran. Der Wagen mit 5 Plätze kommt. Der Fahrer ist die grösste Zahl. Das ist ja das wichtigste. Wenn wir nichts anderes von der Zahl wissen werden als das, was im Wagen steht, wollen wir die signifikanteste Stelle sicher im Wagen drin haben.

[Bild mit Wagen]

Wir verbinden das Universalkomma mit dem Nagel am Wagen und markieren die Stelle am Seil.

Jetzt können wir die Zahl speichern

[Bild im Speicher]

Im Computer wird das als Bitmuster gespeichert. Zuerst Vorzeichen, dann Exponent (Markierungen am Seil), dann Mantisse (die erste Eins ist eigentlich überflüssig).

Dank dem Seil kann man den Wagen wiederherstellen im Bezug auf das Universalkomma und wieder ausrechnen, was das für eine Zahl war.

Beispiel 1.2 *Nicht alle Zahlen lassen sich im Fliesskommazahlensystem genau darstellen. Manche müssen gerundet werden. Zum Beispiel, die Zahl 10.75 würde in Binär so aussehen (hängt am Universalkomma).*

[Bild mit Universalkomma]

Dann kommt der Wagen mit 5 Plätze. Die erste Eins und noch 4 Bits steigen ein. Das Seil wird entsprechend gebunden und markiert.

[Bild mit Wagen]

Die letzte Eins wird nicht gespeichert. Es gibt kein Platz. Sie geht verloren.

2 Fließkommazahlen

2.1 Kleinste und grösste positive Zahlen

Die Länge vom Seil (die Anzahl der möglichen Exponenten) und die Sitzplätze im Wagen (die verfügbaren Bits in der Mantisse) sind endlich. Deswegen können wir nur eine endliche Anzahl von Zahlen darstellen. Dann gibt es unter den endlich vielen positiven Zahlen eine grösste und eine kleinste.

Beispiel 2.1 *Mantissenlänge: 5 Bits, Exponentenbereich: -3 bis 3.*

Wie konstruieren wir die grösste Zahl?

Was wäre die Mantisse? So gross wie möglich. 1.1111

Was wäre der Exponent? So gross wie möglich. 3

Wagen-Darstellung:

Bitmuster: 0 111 (1)1111

Mathematische Darstellung: $1.1111 \cdot 2^3$

Wert: 15.5

Beispiel 2.2 *Mantissenlänge: 5 Bits, Exponentenbereich: -3 bis 3.*

Wie konstruieren wir die kleinste Zahl?

Was wäre die Mantisse? So klein wie möglich, dennoch mit einer führenden

Eins: 1.0000

Was wäre der Exponent? So klein wie möglich. -3

Wagen-Darstellung:

Bitmuster: 0 001 (1)0000

Mathematische Darstellung: $1.0000 \cdot 2^{-3}$

Wert: $1/8$

Aufgabe 2.1 *Mantissenlänge: 3 Bits, Exponentenbereich: -1 bis 1. Finde die grösste und kleinste positive Zahlen in diesem Fließkommasystem. Stelle sie als Bitmuster und in der Exponentialschreibweise dar (Wert im Dezimalsystem angeben). (Für den Exponentenbereich: 01 kodiert -1, 10 kodiert 0, 11 kodiert 1).*

Aufgabe 2.2 *Mantissenlänge: 4 Bits, Exponentenbereich: -1 bis 1, kodiert wie in der vorherigen Aufgabe. Finde die grösste und kleinste positive Zahlen (Werte im Dezimalsystem angeben). Stelle sie als Bitmuster und in der Exponentialschreibweise dar.*

Im Allgemeinen für einen Fließkommasystem, wo die Mantisse m Bits lang ist und der Exponentenbereich von e_{min} bis e_{max} geht, findet man die grösste positive Zahl indem man in der Mantisse nur Einser schreibt, so dass diese so gross wie möglich wird: $1.1111 \dots 111$ und den Exponenten so gross wie möglich wählt: e_{max} . Die grösste Zahl ist also:

$$1.1111111 \dots 111 \cdot 2^{e_{max}}$$

Und hat den Bitmuster: 0 1111...11 (1)1111...11.

Die kleinste positive Zahl findet man, indem man die Mantisse so klein wie möglich wählt: $1.00000 \dots 000$ und den Exponenten so klein wie möglich: e_{min} . Die kleinste Zahl ist also:

$$1.0000000 \dots 000 \cdot 2^{e_{min}}$$

Und hat den Bitmuster: 0 0000...01 (1)000000...0.

2.2 Darstellbare Zahlen

In diesem Abschnitt werden wir ein Gefühl dafür bekommen, welche Zahlen sie darstellen lassen und welche nicht. Hier und in den folgenden Kapiteln, falls nicht speziell vermerkt, werden wir mit Mantissenlänge 5 und Exponentenbereich von -3 bis 3 arbeiten.

Beispiel 2.3 Nehmen wir eine Zahl zwischen $1/8$ und 15.5 (die kleinste und grösste darstellbare Zahlen in diesem Fliesskommazahlensystem), zum Beispiel 10.25 . Lässt sich diese Zahl darstellen?

[Bild mit Wagen]

Diese Zahl lässt sich in gegebenem System nicht exakt darstellen, weil in der Mantisse gibt es Platz nur für 5 Bits, so geht die letzte 1 verloren.

Beispiel 2.4 Wir werden jetzt die nächste und die vorherige darstellbare Zahl von 1 finden.

[Bild mit Wagen]

Die nächste Zahl finden wir, indem wir in der Mantisse ganz rechts eine Eins setzen. Das ist die nächstgrösste Mantisse nach 1.0000 . Der Exponent bleibt gleich. Die nächste Zahl ist also $1 + 1/16$.

Die grösste Zahl kleiner als Eins finden wir, indem wir die Mantisse so gross wie möglich machen und den Exponenten um eins verkleinern. Die vorherige Zahl ist also $1 - 1/32$.

Aufgabe 2.3 Finde den nächstkleinsten und nächstgrössten darstellbaren Nachbarn von den folgenden Zahlen. Schreibe die Werte im Dezimalsystem auf und stelle sie als Bitmuster und in der Exponentialschreibweise dar. Sind alle Nachbarn gleich entfernt?

(a) 2

(b) 3

(c) 4

Kontrollfragen :

1. Kann man im Fliesskommazahlensystem alle reelle Zahlen darstellen?
2. Gibt es eine grösste Zahl im Fliesskommazahlensystem? Falls nein, warum? Falls ja, wie findet man sie?

3. Gibt es eine kleinste Zahl im Fließkommazahlensystem? Falls nein, warum? Falls ja, wie findet man sie?
4. Warum kann man nicht alle Zahlen zwischen f_{\min} und f_{\max} darstellen? Gib ein Beispiel von einer Zahl, die sich nicht im System mit Mantissenlänge 3 und Exponentenbereich von -1 bis 1 darstellen lässt.
5. Sind alle Zahlen im Fließkommazahlensystem gleichverteilt oder gibt es kleinere und grössere Lücken? Falls es Lücken gibt, wie sind sie verteilt? Wo sind sie grösser, wo sind sie kleiner?

3 Addition

Man kann Zahlen nicht nur speichern, sondern auch damit rechnen. In diesem Kapitel werden wir die Addition kennenlernen.

Beispiel 3.1 *In diesem Beispiel rechnen wir $1/8 + 1/4$.*

Beispiel 3.2 *In diesem Beispiel rechnen wir $2 + 3$.*

Aufgabe 3.1 *Rechne folgende Summen aus. Die Mantissenlänge beträgt 5 Bits, der Exponent geht von -3 bis 3.*

(a) $5/8 + 3/4$

(b) $10 + 4.25$

(c) $17/16 + 2$

Die Addition im Fliesskommazahlensystem ist wie gewöhnlich kommutativ, weil wir immer die kleinste Zahl so verschieben, dass der "Wagen" unter dem "Wagen" der grösseren Zahl liegt und dann die Bits in den "Wagen" zusammen addieren. In der folgenden Aufgabe werden wir prüfen, ob sie auch assoziativ ist.

Lernaufgabe : Wir werden jetzt herausfinden, ob die Addition bei den Fliesskommazahlen assoziativ ist. Dazu berechnen wir zweimal die gleiche Summe. Einmal als: $1/8 + 2/8 + 3/8 + 4/8 + 5/8 + 6/8 + 7/8 + 8/8$ und einmal als: $8/8 + 7/8 + 6/8 + 5/8 + 4/8 + 3/8 + 2/8 + 1/8$. Welchen Resultat erwartest du? Sind die zwei Summen gleich oder unterschiedlich? Ist Addition assoziativ? Nimm dir Zeit und rechne die zwei Summen tatsächlich aus bevor du weiter liest.

Nein, die Addition im Fliesskommazahlensystem ist nicht assoziativ, weil man nur Zahlen zusammen rechnen kann, die innerhalb der Mantissenlänge liegen.

Aufgabe 3.2 *Bei den reellen Zahlen, wenn man $1/8 + 1/8 + \dots + 1/8$ rechnet, kann man potenziell unendlich viele immer grössere Zahlen erreichen. Wir wissen, dass das Fliesskommazahlensystem nur endlich viele Zahlen darstellen kann. Also unsere Vermutung ist, dass wir nicht höher als 15.5 gehen können, wenn wir $1/8$ mehrmals addieren. Können wir 15.5 tatsächlich erreichen? Falls ja, wie viele $1/8$ brauchen wir in der Summe, um 15.5 zu erreichen? Falls nein, was ist die grösste Zahl, die wir erreichen können, wenn wir beliebig viele $1/8$ zusammen rechnen?*

Kontrollfragen

1. Warum muss man den "Wagen verschieben" wenn man zwei unterschiedlich grosse Zahlen zusammen addiert?

2. Gregory behauptet, dass "der Wagen bei der Summe dort stehen bleibt, wo er bei der grösseren Zahl steht". Hat er Recht? Argumentiere.
3. Hannah behauptet, dass die Addition bei den Fließkommazahlen nicht kommutativ und nicht assoziativ ist. Hat sie recht? Argumentiere.
4. Der Ameisenkönig möchte ausrechnen, wie viele Ameisen braucht er, um 10 Reiskörnchen zu transportieren. Er weiss, dass eine Ameise $1/4$ Reiskorn transportiert. Der Ameisefinanzminister hat dazu folgendes Programm geschrieben. Die Ameisencomputer arbeiten mit Fließkommazahlen mit Mantissenlänge 5 und Exponenten zwischen -3 und 3 darstellt.

```
1 def nof_ameisen():
2     sum = 0.0
3     i = 0
4     while node != 10.0:
5         i += 1
6         sum += 0.25
7     return i
```

Listing 1: Freds Programm

Kann der Ameisenkönig mit diesem Programm die gewünschte Anzahl Ameisen herausfinden? Falls ja, wie viele Ameisen braucht er, um 10 Reiskörnchen zu transportieren laut diesem Programm? Falls nein, was ist die maximale Summe, die das Programm erreichen kann?

4 Zusammenfassung

Wir haben gesehen, wie man im Computer reelle Zahlen approximiert. Da wir eine endliche Darstellung verwenden, gibt es eine endliche Anzahl Zahlen, die wir darstellen können. Es gibt eine grösste und eine kleinste positive Zahl.

Die grösste positive Zahl kriegen wir, wenn wir in der Mantisse nur Einsen schreiben und den Exponent so weit wie möglich hochdrehen. Die kleinste positive Zahl kriegen wir, wenn wir in der Mantisse eine führende Eins schreiben und sonst nur Nullen und den kleinstmöglichen Exponenten wählen.

Wir haben gesehen, dass der kleinste Abstand zwischen zwei darstellbaren Zahlen wächst, wenn die Zahlen wachsen.

Wir haben gelernt, wie man zwei Zahlen zusammen addiert. Wir haben gesehen, dass zwei darstellbare Zahlen sich nicht immer exakt addieren lassen. Wir haben auch gesehen, dass es einen Unterschied macht, in welcher Reihenfolge man Berechnungen ausführt. Die Addition ist bei Fließkommazahlen kommutativ aber nicht assoziativ.

Die Fließkommazahlen sind eine mächtige Darstellung, die mit wenig Bits sehr unterschiedliche Zahlen speichern kann. Das hat aber auch seine Grenzen. Wir müssen in Kauf nehmen, dass die Resultate der Berechnungen nicht immer 100% genau sind.

5 Beispiellösungen