Exercise Week 04

GianAndrea Müller mailto:muellegi@student.ethz

March 21, 2018

Time Schedule

- 3' Nachbesprechung
- 10' Dezimalzahlen im Binärsystem mit Übung
- 10' Fliesskommasystem $\mathcal{F}^*(2, 24, -126, 127)$
- 10' Tips zu Fliesskommazahlen
- 10' Funktionen
- 15' Pause
- 10' Funktionsdefinition- und deklaration
- 10' Pre- and post-conditions
- 10' Übung zu Funktionen

Learning Objectives

- Verständnis des Fliesskommasystems
- Nutzung von Funktionen

Nachbesprechung

- Kommentieren heisst nicht: Dasselbe in grün.
- Snippets: Verständnis zeigen!

Dezimalzahlen im Binärsystem

binär:	1	1	1	1	1	1	1	
dezimal:	8	4	2	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	

Dezimalzahlen im Binärsystem

X	di	$x-d_i$
1.934	1	0.934
9.34	9	0.34
3.4	3	0.4
4	4	0

X	bi	$x - b_i$
1.9	1	0.9
1.8	1	0.8
1.6	1	0.6
1.2	1	0.2
0.4	0	0.8
1.6	1	0.6
	:	

 $1.1\overline{1100}$

Exercise 4.1 \sim 2'

Berechne die binäre Darstellung folgender Dezimalzahlen:

- 0.25
- **2** 11.1

Solution 4_1

Lösung 1.

b_i	$x-b_i$
0	0.25
0	0.5
1	0
	0

Lösung 2.

X	bi	$x - b_i$
0.1	0	0.1
0.2	0	0.2
0.4	0	0.4
8.0	0	0.8
1.6	1	0.6
1.2	1	0.2

Unser kleines 10bit Fliesskommasystem

Beschreibung von Fliesskommasystemen Anzahl Stellen ≥ 1 $\mathcal{F}(\underbrace{\beta}, \underbrace{p}, \underbrace{e_{min}, e_{max}})$ Basis ≥ 2 Kleinster und Grösster Exponent

Unser kleines 10bit Fliesskommasystem

- Vorkommastelle
- Nachkommastellen
- Exponent

 $\mathcal{F}(2,6,0,15)$

- Vorkommastelle
- Nachkommastellen
- Exponent

```
Beispiele
```

```
\underbrace{1.11111 \cdot 2^{15}}_{\text{Grösste Zahl}} \underbrace{0.00001 \cdot 2^{0}}_{\text{Kleinste Zahl}}
```

$$\mathcal{F}(2,6,-8,7)$$

000000000

- Vorkommastelle
- Nachkommastellen
- Exponent

Beispiele

 $\underbrace{\frac{1.11111 \cdot 2^7}{\text{Grösste Zahl}}}$

 $0.00001 \cdot 2^{-8}$

Kleinste Zahl

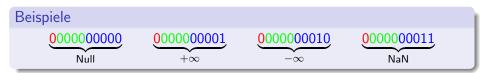
$$\mathcal{F}^*(2,6,-8,7)$$

000000000

- Vorzeichenstelle
- Nachkommastellen
- Exponent

$$\mathcal{F}^*(2,6,-7,7)$$

- Vorzeichenstelle
- Nachkommastellen
- Exponent



Exercise $4_2 \sim 5'$

Floating point systems following IEEE 754

- float (IEEE 745): $\mathcal{F}^*(2, 24, -126, 127)$
- double (IEEE 745): $\mathcal{F}^*(2,53,-1022,1023)$
- What is the largest possible normalized single and double precision floating point number?
- What is the smallest possible normalized single and double precision floatig point number?

Solution 4_2

Floating point systems following IEEE 754

- Smallest normalized number: 2^{e_{min}}
- 2 Largest normalized number, float: +1.11111111111111111111111 · 2¹²⁷

$$\left(1-\left(rac{1}{eta}
ight)^p
ight)eta^{e_{max}+1}$$

Solution 4_2

$$\mathcal{F}^*(2,6,-7,7)$$

Grösste positive Zahl:

$$1.111111 \cdot 2^7$$

② Das entspricht:

- **3** Grösste positive Zahl mit 8bit: $2^8 1$
- **4** Grösste positive Zahl mit 2bit: $2^2 1$
- Mit variablen:

$$(\beta^{e_{max}+1} - 1) - (\beta^{e_{max}+1-p} - 1) = (\beta^{e_{max}+1} - \beta^{e_{max}+1-p})$$

$$= \left(1 - \left(\frac{1}{2}\right)^{p}\right) \beta^{e_{max}+1}$$

Tipps zu Fliesskommazahlen

```
//Kein Vergleich gerundeter Zahlen
_2 double a = 1.1;
 if(100*a == 110) cout << true << endl;
4
 //Keine Add. versch. grosser Zahlen
  float a = 67108864.0f + 1.0f
  //output: 67108864
  //Keine Subtr. aehnlich grosser Zahlen
  float x_0 = 0.2;
10
  //represented as: 0.20000000298
11
  float x_1 = 6*x_0 - 1; //is not 0.2
12
```

Funktionsdefinition und -deklaration

```
void g (...); //declaration of g
  void f (...)
  g(...);
  void g (...) // definition of g
  f(...);
10
11
```

PRE- und POST-Bedingungen

```
#include <cmath>
  int main(){
    // PRE: Value representing angle
4
       expressed in radians
    // POST: Cosine of x
    // double cos(double x);
7
    double x = M_PI;
    double result = cos(M_PI);
10
```