Unum-Arithmetik

Laslo Hunhold

Oberseminar AG Kunoth im SS 2016, Universität zu Köln
2016-07-27

Ziel dieses Vortrags: Einführung in die Unum-Arithmetik

- Probleme mit IEEE 754 Fließkommazahlen
- Theoretische Herleitung der Unums
- Unum-C-Toolbox
- Ausblick zu Anwendungsmöglichkeiten

Verwendete Literatur:

[Gus16] John L. Gustafson, *A Radical Approach to Computation with Real Numbers* http://www.johngustafson.net/presentations/Unums2.0.pdf, Februar 2016. Abgerufen: 2016-07-26.

[Kah06] William Kahan. *How futile are mindless assessments of roundoff in floating-point computation?* https://people.eecs.berkeley.edu/~wkahan/Mindless.pdf, Januar 2006. Abgerufen: 2016-07-26.

[MBdD+10] Jean-Michel Muller, Nicolas Brisebarre, Florent de Dinechin, Claude-Pierre Jeannerod, Vincent Lefèvre, Guillaume Melquiond, Nathalie Revol, Damien Stehlé, und Serge Torres. *Handbook of Floating-Point Arithmetic*. Birkhäuser Boston, Boston, MA, USA, 1. Ausgabe, Dezember 2010. ISBN 9780817647049.

Probleme mit IEEE 754 Fließkommazahlen

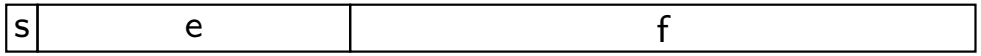


Figure: IEEE 754 Fließkommazahl ($s = 1, (e, f) \in \{(5, 10), (8, 23), (11, 52)\}$)

- Verschwendung von Bitmustern mit NaN, Infinity, ± 0 ($2^f + 1$)
  $\approx 0.02\%, \approx 0.04\%, \approx 1.6\%$ für double, single, half precision
- Vorzeichenbehaftete Null
  $x = y \not\Rightarrow \frac{1}{x} = \frac{1}{y}$, da $0 = -0$, aber $\frac{1}{0} \neq \frac{1}{-0}$
- Schwierige Implementierung wegen vieler Spezialfälle
- Rundungsfehler

Probleme mit IEEE 754 Fließkommazahlen

1. Beispiel: Teufelsfolge [MBdB+10]

Betrachte $\{u_n\}_{n\in\mathbb{N}_0}$ definiert als

$$u_n := \begin{cases} 2 & n = 0 \\ -4 & n = 1 \\ 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}\cdot u_{n-2}} & n \geq 2 \end{cases}$$

Charakteristisches Polynom: Nullstellen bei 5, 6 and 100.

Mit den Startwerten $(2, -4)$

$$\lim_{n\to\infty} u_n = 6$$

Wie verhält sich der IEEE 754 Löser?

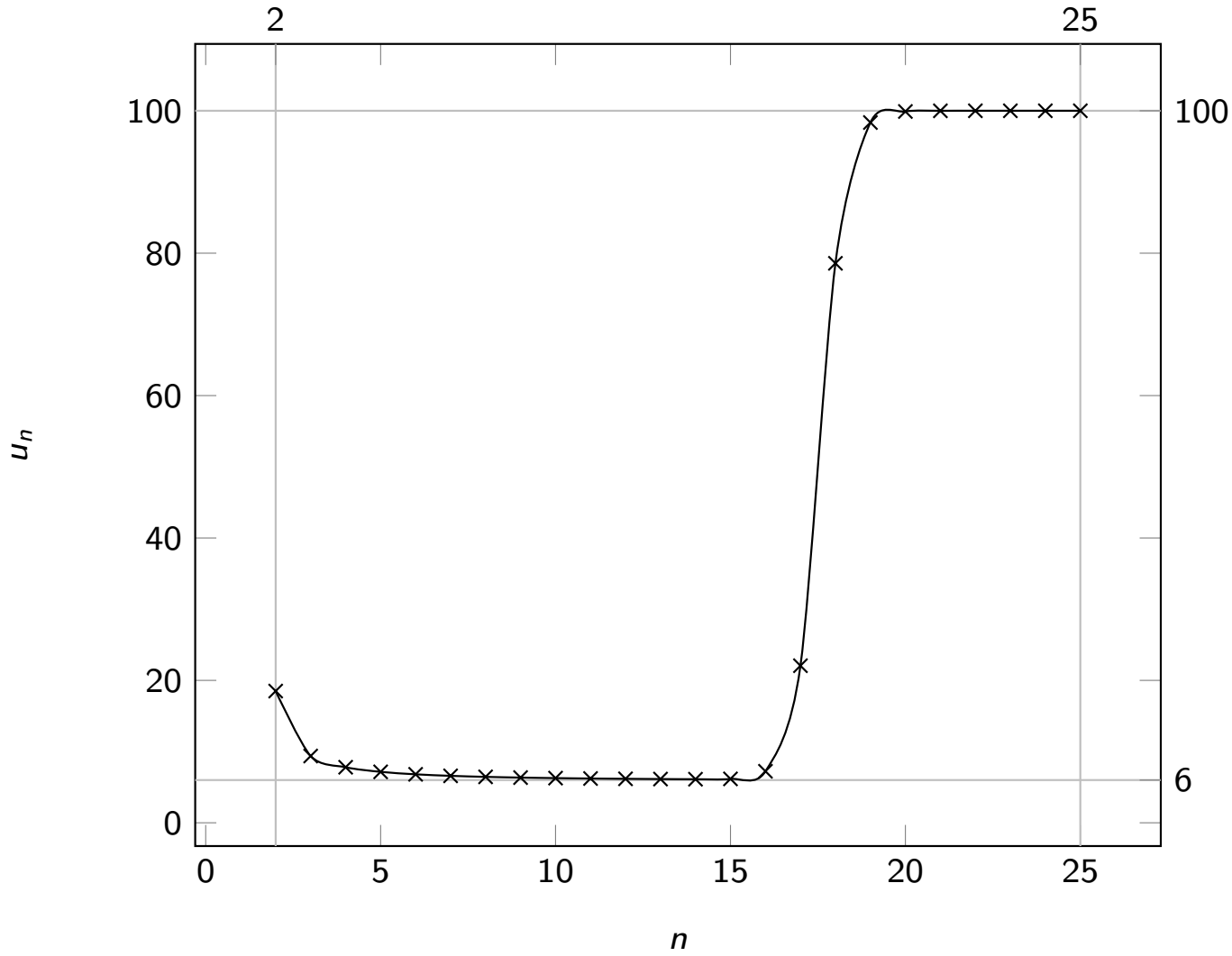


Figure: IEEE 754 (double) Verhalten der Teufelsfolge.

Probleme mit IEEE 754 Fließkommazahlen

2. Beispiel: Stumme Polstelle [Kah06]

Betrachte $f : \mathbb{R} \to \mathbb{R}$ definiert als

$$f(x) := \frac{\ln(|3 \cdot (1 - x) + 1|)}{80} + x^2 + 1.$$

Es gilt

$$\lim_{x \downarrow \frac{4}{3}} \{f(x)\} = \lim_{x \uparrow \frac{4}{3}} \{f(x)\} = -\infty.$$

Wie verhält sich der IEEE 754 Löser?

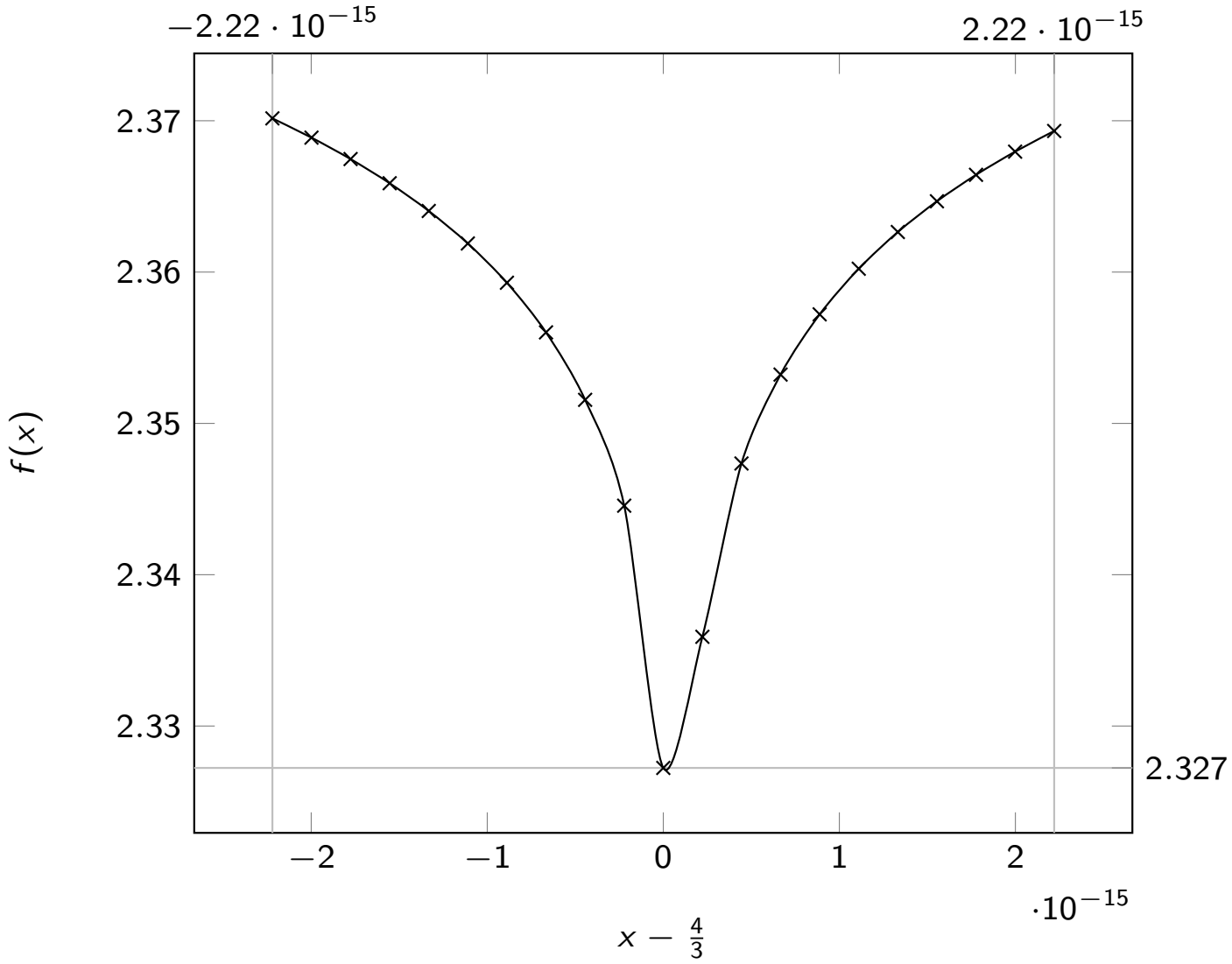


Figure: IEEE 754 (double) Verhalten von f um $\frac{4}{3}$ für alle doubles in $\left[\frac{4}{3} - 2.22 \cdot 10^{-15}, \frac{4}{3} + 2.22 \cdot 10^{-15}\right]$.

Probleme mit IEEE 754 Fließkommazahlen

Zwischenfazit

Völlig falsche Ergebnisse durch Rundungsfehler, selbst mit dem double-precision-Quasistandard.

- Verschwendung von Arbeitsspeicher und Pipeline-Bandbreite.
- Falsches Gefühl der Sicherheit.
- Fehlentscheidungen (Mensch und Maschine, Finanzen, Militär, ...).

Intervallarithmetik?

- Abhängigkeitsproblem erfordert große Vorsicht.
- Liefert oft zu pessimistische Lösungsschranken.
- Baut meistens auf IEEE 754 Fließkommazahlen auf.

Ziele von Unums 2.0 [Gus16]

- Brauchbare Lösungen bei geringer Präzision.
- Garantierte Lösungsschranken.
- Schnelle und einfache Maschinenarithmetik.

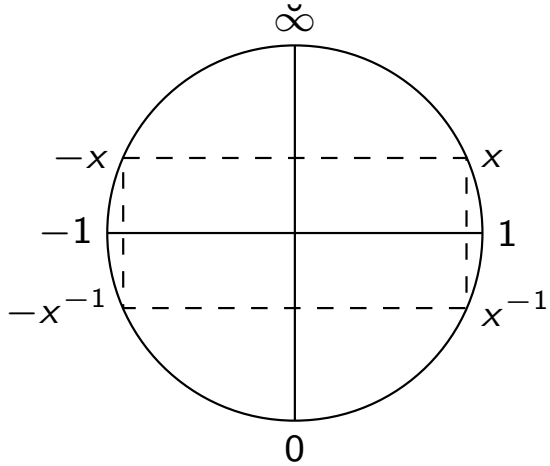
Projektiv erweiterte reelle Zahlen

Definiere

$$\mathbb{R}^* := \mathbb{R} \cup \{\breve{\infty}\}$$

mit den Axiomen für $a, b \in \mathbb{R}$ und $b \neq 0$

1. $-(\breve{\infty}) := \breve{\infty}$
2. $a + \breve{\infty} = \breve{\infty} + a := \breve{\infty}$
3. $b \cdot \breve{\infty} = \breve{\infty} \cdot b := \breve{\infty}$
4. $a/\breve{\infty} := 0$
5. $b/0 := \breve{\infty}$



Nicht definiert sind $\breve{\infty} + \breve{\infty}$, $\breve{\infty} \cdot \breve{\infty}$ und $0 \cdot \breve{\infty}$.

$\mathbb{R}^*$-Intervallarithmetik

Offenes $\mathbb{R}^*$-Intervall

Seien $\underline{a}, \overline{a} \in \mathbb{R}^*$ mit $\underline{a} \leq \overline{a}$.

$$\mathbb{R}^* \supset (\underline{a}, \overline{a}) := \begin{cases} \mathbb{R} & \underline{a} = \overline{a} = \breve{\infty} \\ \left\{ x \in \mathbb{R} \,\middle|\, x < \overline{a} \right\} & \underline{a} = \breve{\infty} \wedge \overline{a} \in \mathbb{R} \\ \left\{ x \in \mathbb{R} \,\middle|\, \underline{a} < x \right\} & \underline{a} \in \mathbb{R} \ \wedge \overline{a} = \breve{\infty} \\ \left\{ x \in \mathbb{R} \,\middle|\, \underline{a} < x < \overline{a} \right\} & \underline{a}, \overline{a} \in \mathbb{R} \end{cases}$$

Menge der offenen $\mathbb{R}^*$-Intervalle $\mathbb{I}$

$$\mathbb{I} := \left\{ (\underline{a}, \overline{a}) \,\middle|\, \underline{a}, \overline{a} \in \mathbb{R}^* \right\} \text{ mit } \oplus, \otimes$$

$\mathbb{R}^*$-Einermengen $\mathbb{R}^*_S$

$$\mathbb{R}^*_S := \left\{ \{x\} : x \in \mathbb{R}^* \right\}.$$

$\mathbb{R}^*$-Flocken $\mathbb{F}$

$$\mathbb{F} := \mathbb{I} \sqcup \mathbb{R}^*_S \text{ mit } \boxplus, \boxtimes, \text{Inversion } / \text{ und Negation } -$$

$\mathcal{P}(\mathbb{F})$ enthält u.a. alle offenen, geschlossenen und halboffenen Intervalle.

Menge der Unums $\mathbb{U}$

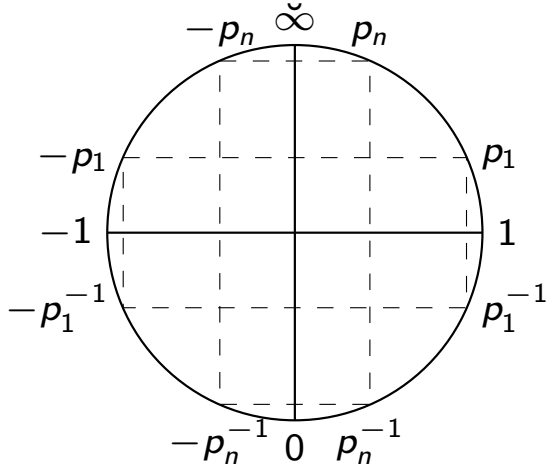
Definition

Wähle Gitter

$$P = \{p_1, \ldots, p_n \mid \forall i < j : p_i < p_j\} \subset (1, \breve{\infty}),$$

und definiere

$$\begin{aligned}\mathbb{F} \supset \mathbb{U}(P) := & \bigsqcup_{i=1}^{n} \left[\{p_i\} \sqcup /\{p_i\} \sqcup -\{p_i\} \sqcup -/\{p_i\}\right] \sqcup \\ & \bigsqcup_{i=0}^{n} \left[\{(p_i, p_{i+1})\} \sqcup \{/(p_i, p_{i+1})\} \sqcup \{-(p_i, p_{i+1})\} \sqcup \{-/(p_i, p_{i+1})\}\right] \sqcup \\ & \{1\} \sqcup \{-1\} \sqcup \{0\} \sqcup \{\breve{\infty}\}\end{aligned}$$



SORNS (Sets of real numbers)

Betrachte nun Teilmengen von $\mathcal{P}(\mathbb{U}(P))$ als Mittelpunkt der Unum 2.0-Arithmetik.

Blur-Operator

$$\begin{aligned} \text{blur} : \mathbb{F} &\to \mathcal{P}(\mathbb{U}(P)) \\ f &:\mapsto \{u \in \mathbb{U} : f \subseteq u\}. \end{aligned}$$

Duale Operationen

Sei $\star : \mathbb{F} \times \mathbb{F} \to \mathbb{F}$ eine Operation auf $\mathbb{F}$ und $\sim$ eine Relation.

$$\langle\star\rangle : \mathcal{P}(\mathbb{U}(P)) \times \mathcal{P}(\mathbb{U}(P)) \to \mathcal{P}(\mathbb{U}(P))$$

$$(U, V) :\mapsto \bigcup_{u \in U} \bigcup_{v \in V} \begin{cases} \emptyset & U \sim V \wedge u \neq v \\ \mathbb{R}^* & u \star v = \emptyset \\ \text{blur}(u \star v) & \text{sonst} \end{cases}$$

Paarweise Auswertung abhängiger SORNs, totale Auswertung unabhängiger SORNs.

Addition, Multiplikation, Subtraktion und Division sind nun wohldefiniert auf SORNs.

Unum C-Toolbox

```
unum u;
struct sorn a, b;
```

Arithmetik

```
uadd(&a, &b);
usub(&a, &b);
umul(&a, &b);
udiv(&a, &b);
upot(&a, &b);

uneg(&a);
uinv(&a);
```

Mengenoperationen

```
uemp(&a);
uset(&a, &b);
ucut(&a, &b);
uuni(&a, &b);
uint(&a, 1.0, INFINITY);
```

Ausgabe

```
uout(&a);
```

Umgebung (Datentypen, Tabellen, ...) wird automatisch generiert aus beliebigem Eingabegitter

Modellierung von SORNs in 8 Bits

```
#include "unum.h"

int
main(void)
{
    struct sorn a;

    uemp(&a);
    uint(&a, -300, -20);
    uint(&a, -2, 4);
    uint(&a, 5, 5);
    uint(&a, 92.5, 160);
    uout(&a);
    putchar('\n');

    return 0;
}
```

```
$ ./model
(∝̌,-20] [-2,4] [5,5] (90,∝̌)
```

Abhängigkeitsproblem in 8 Bits

```
#include "unum.h"

int
main(void)
{
    struct sorn a;
    int i;

    uemp(&a);
    uint(&a, -4, 4);
    uout(&a);
    putchar('\n');

    for (i = 0; i < 6; i++) {
        usub(&a, &a);
        uout(&a);
        putchar('\n');
    }

    return 0;
}
```

```
$ ./dep
[-4,4]
(-0.7,0.7)
(-/10,/10)
(-/90,/90)
(-0.009,0.009)
(-0.009,0.009)
(-0.009,0.009)
```

Ergebnisse und weiterer Ausblick

- Einsparungen bei Arbeitsspeicher und Energie (brauchbare Ergebnisse mit weniger als 64 Bits)
- Neuer Ansatz für parallele Löser (Iterative Verfahren von groben auf feine Gitter)
- Solidität bei Problemen für Fließkommazahlen und traditionelle Intervallarithmetik
- Anwendungsspezifische Zahlensysteme

Vielen Dank für Ihre Aufmerksamkeit!