

Programmieren 1 (PRG1)

Übung 4

4.1

a)

0	10000111	10000010100101010000010
S	E	M

$$x = s * m * b^e$$

$$s = (-1)^S = (-1)^0 = 1$$

$$m = 1 + M/2^p = 1 + 0.51009 = 1.51009 \quad p = \text{Gesamtzahl der Bits in M}$$

$$b = 2$$

$$e = E - B = 135 - 127 = 8$$

$$\Rightarrow x = 1 * 1.51009 * 2^8 = 386.58$$

b)

Die Mantisse einer 64Bit Zahl kann maximal die Zahl 2^{53} darstellen, da sie 52Bits für M besitzt. Damit sollte die größtmögliche, darstellbare Zahl eigentlich $2^{53} - 1$ sein, allerdings erhöht sich diese Zahl auf 2^{53} durch das sog. *hidden bit*. Die kleinstmögliche, nicht darstellbare Zahl ist somit $2^{53} + 1$ ($9.007199254740993 \cdot 10^{15}$).

c)

Der primäre Vorteil dieser Technik ist, dass alle 8Bit (bzw. 11Bit, etc.) des Exponenten (E) genutzt werden können, um den Exponenten der Basis (e) darzustellen. Ein 8Bit Exponent (e) im *single* kann also Zahlen zwischen -127 und +128 darstellen, während ein gleichwertiger Exponent im Zweierkomplement nur Zahlen zwischen -64 und +63 darstellen könnte.

d)

Negative Exponenten sind notwendig, um Zahlen darstellen zu können die kleiner als 0 sind.

e)

Typische Probleme beim Rechnen mit Gleitkommazahlen umfassen Wandlungsfehler (z.B. Dezimal zu binär und umgekehrt), Vergleichsfehler (Vergleiche zwischen mathematisch gleichen Zahlen häufig falsch, z.B. $0.1 + 0.1 + 0.1$ in Python $\neq 0.3$), *cancellation* (niedrige Nachkommastellen bei Subtraktion

von sehr ähnlichen Werten sehr ungenau) und *absorption* (Addition, bzw. Subtraktion von sehr kleiner Zahl, ändert sehr große Zahl nicht).

Um solche Fehler zu umgehen können Funktionen wie `math.isclose()`, aus der *math library* verwendet werden (weitere Alternativen sind die `decimal/fractions` Module). Außerdem sollte beim Programmieren darauf geachtet werden, oben genannte Situationen zu vermeiden und Alternativlösungen zu suchen.

Quelle: Vorlesungsfolien „V03-04-Elementare Datentypen-korrigiert“; Seiten 16 - 19

4.2

Generell war mir die Verwendung der Programme a) und b) sehr unklar. Austesten von Grenzen der Ausführbarkeit solcher Programme in Python?

d)

Fragestellung war mir leider extrem unklar.

Habe aus „False + 2“ eine variable „false“ gemacht und diese +2 genommen, da Definition von „False“ Variable nicht möglich.