

- Auswertungsreihenfolge kann Ergebnisgenauigkeit beeinflussen

- **Beispiel:** Assoziativgesetz möglicherweise nicht erfüllt *(bei untersch. Genauigkeiten)*

=>  $(a + b) + c == a + (b + c)$  ?

```
float flt1 = (2.0f + FLT_EPSILON) + FLT_EPSILON;  
float flt2 = 2.0f + (FLT_EPSILON + FLT_EPSILON);  
int equals = (flt1 == flt2);          /* equals: 0 (false) */
```

- **FLT\_EPSILON** wird bei zweimaliger Addition zu `2.0f` vollständig absorbiert
  - Wird hingegen zunächst **FLT\_EPSILON + FLT\_EPSILON** berechnet, kann das (nun größere) Ergebnis zu `2.0f` addiert werden, ohne absorbiert zu werden
- Möglicherweise unterschiedliche Ergebnisse durch unterschiedlich genaue Typen (Operanden, Berechnungstyp, Zuweisungstyp)
  - **Beispiel:** `0,25` ist sowohl in `float` als auch in `double` präzise repräsentierbar, `0,1` hingegen nicht präzise und mit unterschiedlicher Genauigkeit repräsentierbar

```
int equals = (0.25F == 0.25); /* 1 (true) */  
int equals = (0.1F == 0.1);   /* 0 (false) */
```