

Programming Embedded Systems

Lecture 6 (cont)
**Real-valued data in
embedded software**

Wednesday Feb 8, 2012

Philipp Rümmer
Uppsala University
`Philipp.Ruemmer@it.uu.se`

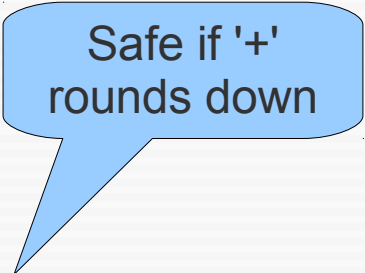
Interval methods

Motivation

- Is it possible to draw reliable conclusions from computations done with limited precision?

Motivation (2)

- **Yes** ... if we can control the rounding!
- E.g.



Safe if '+'
rounds down

```
if (a + b >= 0) {  
    // do something dangerous that  
    // that only works if a+b >= 0  
}
```

Interval methods

- Common in scientific computing, numerical mathematics; rich body of research
- Often an extremely efficient method to obtain estimates on the precise result of computations
- Are sound in the sense that no wrong answers are given, but can fail since error bounds might be too large to be useful

Idealised interval arithmetic

- Domain of real intervals:

$$\mathbb{IR} = \{ \mathbf{x} = [\underline{x}, \bar{x}] \mid \underline{x} \in \mathbb{R} \cup \{-\infty\}, \bar{x} \in \mathbb{R} \cup \{+\infty\}, \underline{x} \leq \bar{x} \}$$

- Mathematical operations are lifted to intervals, e.g.:

$$\mathbf{x} + \mathbf{y} = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$$

$$\mathbf{x} - \mathbf{y} = [\underline{x} - \bar{y}, \bar{x} - \underline{y}]$$

- Generally defined such that:

$$\mathbf{x} \text{ op } \mathbf{y} \supseteq \{ x \text{ op } y \mid x \in \mathbf{x}, y \in \mathbf{y} \}$$

Machine interval arithmetic

- Intervals represented using two machine numbers (floating-point or fixed-point):

$$\mathbb{I}D = \{ \mathbf{x} = [\underline{x}, \bar{x}] \mid \underline{x} \in D \cup \{-\infty\}, \bar{x} \in D \cup \{+\infty\}, \underline{x} \leq \bar{x} \}$$

- Operations are used *rounding outward*:

$$\mathbf{x} \tilde{+} \mathbf{y} = [\text{round}_{\downarrow}(\underline{x} + \underline{y}), \text{round}_{\uparrow}(\bar{x} + \bar{y})]$$

- This ensures that the precise result is always within the computed machine interval

Example

- Fixed-point arithmetic with rounding:

$$\begin{aligned} 0.5 \tilde{\cdot} 1.5 &= 0.5 \\ (1 \cdot 2^{-1}) \tilde{\cdot} (3 \cdot 2^{-1}) &= (1 \cdot 3 \gg 1) \cdot 2^{-1} \end{aligned}$$

- In interval arithmetic:

$$\begin{aligned} [0.5, 0.5] \tilde{\cdot} [1.5, 1.5] &= [\mathit{round}_{\downarrow}(0.75), \mathit{round}_{\uparrow}(0.75)] \\ &= [0.5, 1.0] \end{aligned}$$

Inputs are
singleton
intervals

Output bounds
the precise result

Further reading

- **Library on interval methods:**
<http://www.cs.utep.edu/interval-comp/>