Solution to Problem 17

Task. Write down:

- an expression for which an optimizing compiler improves the estimates provided by straightforward interval computations, and
- an expression for which an optimizing compiler worsens improves the estimates provided by straightforward interval computations.

Your examples should be different from examples given in class (similar is OK).

First example. Let us consider the problem of estimating the expression $a \cdot b + a \cdot c$ when $a \in [-2, 2]$, $b = 1$ (i.e., $b \in [1, 1]$) and $c = -1$ (i.e., $c \in [-1, -1]$).

In this example, straightforward interval computations lead to

$$[-2, 2] \cdot [1, 1] + [-2, 2] \cdot [-1, -1] = [-2, 2] + [-2, 2] = [-4, 4].$$

For this example, optimizing compiler – aiming to minimize the number of multiplications – will transform the original expression into $a \cdot (b + c)$. For this new expression, straightforward interval computations lead to a narrower interval:

$$[-2, 2] \cdot ([1, 1] + [-1, -1]) = [-2, 2] \cdot [0, 0] = [0, 0].$$

Second example. Let us consider the problem of estimating the expression

$$\frac{1}{1 + \dfrac{a}{2b}}$$

when $\mathbf{a} = \mathbf{b} = [1, 2]$. In this case, straightforward interval computations lead to

$$\frac{1}{1 + \dfrac{[1, 2]}{2 \cdot [1, 2]}} = \frac{1}{1 + \dfrac{[1, 2]}{[2, 4]}} = \frac{1}{1 + [0.25, 1]} = \frac{1}{[1.25, 2]} = [0.5, 0.8].$$

For this example, optimizing compiler – aiming to minimize number of divisions – will transform the original expression into

$$\frac{2b}{2b + a}.$$

For this new expression, straightforward interval computations lead to a wider interval:

$$\frac{2 \cdot [1,2]}{2 \cdot [1,2] + [1,2]} = \frac{[2,4]}{[2,4] + [1,2]} = \frac{[2,4]}{[3,6]} = \left[\frac{1}{3}, \frac{4}{3}\right] = [0.33\ldots, 1.33\ldots].$$