Inclusion proofs for `interval`

Luna-Flow `floating`, design attachment

This note proves the facts the `interval` design page relies on. Notation: $\boldsymbol{x} = [\underline{x}, \overline{x}]$ is a closed interval of the extended reals with $\underline{x} \le \overline{x}$, $\underline{x} \ne +\infty$, $\overline{x} \ne -\infty$, read as the set of *real* numbers between the endpoints; $\varnothing$ is the empty interval. RD_p and RU_p round a real number to the nearest p-bit binary number below and above. F_p is the set of p-bit binary numbers (with unbounded exponent).

1. Directed rounding

Lemma 1 (directed rounding). For all reals $a \le b$: (i) $\mathrm{RD}_p(a) \le a \le \mathrm{RU}_p(a)$; (ii) $\mathrm{RD}_p(a) \le \mathrm{RD}_p(b)$ and $\mathrm{RU}_p(a) \le \mathrm{RU}_p(b)$; (iii) for a finite set S, $\min_{s\in S} \mathrm{RD}_p(s) = \mathrm{RD}_p(\min S)$ and $\max_{s\in S} \mathrm{RU}_p(s) = \mathrm{RU}_p(\max S)$.

Proof. $\mathrm{RD}_p(a) = \max\{f \in F_p : f \le a\}$, which gives (i). If $a \le b$, every $f \in F_p$ with $f \le a$ also satisfies $f \le b$, so the maximum over the larger set is not smaller: (ii). For (iii), let $m = \min S$. By (ii) $\mathrm{RD}_p(m) \le \mathrm{RD}_p(s)$ for every $s \in S$, and the minimum is attained at $s = m$. The RU case is symmetric. □

Corollary 2 (outward rounding preserves containment). If $L \le \inf A$ and $U \ge \sup A$ for a set A of reals, then $A \subseteq \big[\mathrm{RD}_p(L), \mathrm{RU}_p(U)\big]$.

Proof. By Lemma 1(i), $\mathrm{RD}_p(L) \le L \le \inf A$ and $\mathrm{RU}_p(U) \ge U \ge \sup A$. □

Lemma 3 (nested grids). If $q \le p$ then $\mathrm{RD}_q\big(\mathrm{RD}_p(a)\big) = \mathrm{RD}_q(a)$ and likewise for RU.

Proof. $F_q \subseteq F_p$: a q-bit significand is a p-bit significand padded with zeros. Let $r = \mathrm{RD}_p(a)$. Every $f \in F_q$ with $f \le a$ lies in F_p, hence $f \le r$ by maximality of r; conversely $f \le r$ implies $f \le a$. So $\{f \in F_q : f \le r\} = \{f \in F_q : f \le a\}$ and the maxima agree. □

2. The fundamental theorem

For a real function f with domain D_f, an *interval extension* F is a map on intervals with $f\big(\boldsymbol{x} \cap D_f\big) \subseteq F(\boldsymbol{x})$ for every $\boldsymbol{x}$. Interval extensions are *inclusion isotone* if $\boldsymbol{x} \subseteq \boldsymbol{y} \Rightarrow F(\boldsymbol{x}) \subseteq F(\boldsymbol{y})$.

Theorem 4 (fundamental theorem of interval arithmetic). Let e be an arithmetic expression in variables $\xi_1, \ldots, \xi_n$ built from operations each of which is evaluated by an interval extension. Write f for the real function of e (defined where every subexpression is defined) and E for its interval evaluation. Then for all intervals $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$,

$$\big\{f(\xi) : \xi \in \boldsymbol{x}_1 \times \ldots \times \boldsymbol{x}_n, \xi \in D_f\big\} \subseteq E(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n). \tag{1}$$

Proof. Induction on e. A variable ξ_i evaluates to $\boldsymbol{x}_i$ and $\xi_i \in \boldsymbol{x}_i$. For $e = g(e_1, \ldots, e_k)$ with g evaluated by the extension G: take $\xi \in D_f$. Then each e_j is defined at ξ, and by the induction hypothesis $y_j := f_{e_j}(\xi) \in E_j$. Since $(y_1, \ldots, y_k) \in D_g$ and G is an extension, $f(\xi) = g(y_1, \ldots, y_k) \in g\big((E_1 \times \ldots \times E_k) \cap D_g\big) \subseteq G(E_1, \ldots, E_k) = E(\boldsymbol{x})$. □

Nothing in the proof requires the two occurrences of a variable to take the same value, which is why $\boldsymbol{x} - \boldsymbol{x} \ne \{0\}$ (the *dependency problem*): E encloses the larger set where every occurrence varies independently.

3. Endpoint formulas

Lemma 5 (corners of a bilinear map). On a box $\boldsymbol{x} \times \boldsymbol{y}$ of bounded intervals, $h(\xi, \eta) = \xi\eta$ attains its minimum and maximum at corners.

Proof. For fixed η, h is affine in ξ, so its extrema over $\boldsymbol{x}$ are at $\underline{x}$ or $\overline{x}$; therefore $\max_{\boldsymbol{x}\times\boldsymbol{y}} h = \max_{\eta\in\boldsymbol{y}} \max(\underline{x}\eta, \overline{x}\eta)$, and each $\eta \mapsto \underline{x}\eta$ is affine, attaining its maximum at $\underline{y}$ or $\overline{y}$. The minimum is analogous. □

With Corollary 2 this gives $\boldsymbol{xy} \subseteq [\mathrm{RD}\min S, \mathrm{RU}\max S]$ for $S = \left\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\right\}$. The sign cases used by the code select from S: if $\underline{x}, \underline{y} \geq 0$ then $\xi\eta$ is increasing in both arguments on the box, so the minimum is $\underline{x}\underline{y}$ and the maximum $\overline{x}\overline{y}$; the other three single-sign cases follow by $\xi\eta = (-\xi)(-\eta) = -((-\xi)\eta)$. In the remaining cases at least one interval, say $\boldsymbol{x}$, has 0 in its interior. Then $\min S \leq 0 \leq \max S$, because $\underline{x}\eta$ and $\overline{x}\eta$ have opposite signs. If the minimum is attained at $\underline{x}\underline{y} < 0$, then $\underline{y} > 0$ and $\underline{x}\overline{y} \leq \underline{x}\underline{y}$; if at $\overline{x}\overline{y} < 0$, then $\overline{y} < 0$ and $\overline{x}\underline{y} \leq \overline{x}\overline{y}$. So the minimum is always attained at $\underline{x}\overline{y}$ or $\overline{x}\underline{y}$, and symmetrically the maximum at $\underline{x}\underline{y}$ or $\overline{x}\overline{y}$; the code evaluates these four products whenever an operand has 0 in its interior. For unbounded intervals the corner lemma holds in the limit, with $0 \cdot (\pm\infty)$ contributing 0: a zero endpoint multiplies an unbounded side to 0 because the real points near that endpoint give products near 0.

Sum and difference are monotone in each argument (increasing, resp. increasing in ξ and decreasing in η), so $\boldsymbol{x} + \boldsymbol{y} = \left[\underline{x} + \underline{y}, \overline{x} + \overline{y}\right]$ and $\boldsymbol{x} - \boldsymbol{y} = \left[\underline{x} - \overline{y}, \overline{x} - \underline{y}\right]$ exactly; Corollary 2 makes the rounded versions enclosures.

4. Far addends

The endpoint sum $A + s$ with $|s| \ll |A|$ is not formed exactly. Let $t(v)$ be the exponent of the leading bit of $v \neq 0$ (so $2^{t(v)} \leq |v| < 2^{t(v)+1}$), $e(A)$ the exponent of the last stored bit of A, $M = \max(65536, \mathrm{prec}(A), \mathrm{prec}(s))$ and $c = \min(e(A), t(A) - M)$. If $t(s) < c - 2$, the code replaces s by $s' = \mathrm{sign}(s)2^{c-2}$ when s pushes the sum in the requested rounding direction, and by $s' = 0$ otherwise.

Lemma 6 (soundness). For upward sums $A + s' \geq A + s$; for downward sums $A + s' \leq A + s$.

Proof. $|s| < 2^{t(s)+1} \leq 2^{c-2}$. Upward, $s > 0$: $s' = 2^{c-2} > s$. Upward, $s < 0$: $s' = 0 > s$. The downward cases are mirror images. □

Lemma 7 (no change after rounding). If $p \leq M$ and $s \neq 0$, then $\mathrm{RD}_p(A + s) = \mathrm{RD}_p(A + s')$ and $\mathrm{RU}_p(A + s) = \mathrm{RU}_p(A + s')$.

Proof. Near $|A|$ (in the binade of A and the one below it) the p-bit numbers are multiples of $2^{t(A)-p}$, and $t(A) - p \geq t(A) - M \geq c$; A itself is a multiple of $2^{e(A)}$, hence of 2^c. So the open interval $I = (A - 2^c, A + 2^c)$ contains no p-bit number other than possibly A. Consequently RU_p is constant on $(A - 2^c, A]$ (its value is $\mathrm{RU}_p(A)$) and on $(A, A + 2^c)$, and RD_p is constant on $[A, A + 2^c)$ and on $(A - 2^c, A)$. Since $|s|, |s'| \leq 2^{c-2}$, both sums lie in I. Upward with $s > 0$: both lie in $(A, A + 2^c)$. Upward with $s < 0$: $s' = 0$, so the sums are A and a point of $(A - 2^c, A)$, both in $(A - 2^c, A]$. The downward cases are symmetric. □

Lemma 6 is what the inclusion property needs and holds for every precision; Lemma 7 says the surrogate costs no tightness when the result precision is at most M. Alignment of A and s' shifts by at most $t(A) - c + 2 \leq M + 2$ bits plus the width of A, instead of the full exponent gap.

5. Certified elementary endpoints

Lemma 8 (acceptance test). Let $L \leq y \leq U$ with rational L, U. If $\mathrm{RD}_p(L) = \mathrm{RD}_p(U)$ then $\mathrm{RD}_p(L) = \mathrm{RD}_p(y)$; if $\mathrm{RU}_p(L) = \mathrm{RU}_p(U)$ then $\mathrm{RU}_p(U) = \mathrm{RU}_p(y)$.

Proof. By Lemma 1(ii), $\mathrm{RD}_p(L) \leq \mathrm{RD}_p(y) \leq \mathrm{RD}_p(U) = \mathrm{RD}_p(L)$. □

The trigonometric and arctangent kernels accept a work precision only when both tests pass, so their endpoints are the correctly directed roundings of the exact values; otherwise the work precision w grows to $w + \max(32, \frac{w}{2})$, and at most 12 work precisions are tried. The exponential and logarithm kernels skip the test and

return $\mathrm{RD}_p(L)$ and $\mathrm{RU}_p(U)$, which are enclosures by Corollary 2 and, with 64 guard bits, almost always the correct directed roundings.

Series tails. For $0 \le x \le \frac{1}{8}$ the terms $t_k = \frac{x^k}{k!}$ satisfy $\frac{t_{k+1}}{t_k} = \frac{x}{k+1} \le \frac{1}{16}$ for $k \ge 1$ (for $k = 0$ the ratio is x, up to $\frac{1}{8}$). The code stops after a term t_n with $n \ge 1$, so every ratio in the tail is at most $\frac{1}{16}$ and the tail after t_n is at most $t_{n+1} \sum_{j \ge 0} 16^{-j} = \frac{16}{15} \cdot t_{n+1} \le 2t_{n+1}$, the bound the code adds. For $\ln y = 2 \sum_{k \ge 0} \frac{z^{2k+1}}{2k+1}$ with $z = \frac{y-1}{y+1} \in \left[0, \frac{1}{3}\right]$ $(1 \le y \le 2)$, let t' be the first omitted term. Consecutive terms have ratio $z^2 \frac{2k+1}{2k+3} \le z^2 \le \frac{1}{9}$, so the omitted tail is at most $t' \sum_{j \ge 0} 9^{-j} = \frac{9}{8} t'$; after doubling it is at most $\frac{9}{4} t' \le 3t'$, the bound the code adds.