5. The decoration system at Level 1

5.1 Overview

An implementation makes the decoration system available by providing:

– a decorated version of each interval extension of an arithmetic operation, of each interval constructor, and of some other operations; and

– various auxiliary functions, e.g., to extract the interval and decoration parts, and to apply a standard initial decoration to an interval.

The decoration system is specified here at a mathematical level, with the finite-precision aspects presented in Clause 6. Subclauses 5.2, 5.3, and 5.4 give the basic concepts; 5.5 and 5.6 define how intervals are given an initial decoration, and how decorations are bound to library interval arithmetic operations to give correct propagation through expressions; 5.7 is about non-arithmetic operations.

5.2 Definitions and properties

A decoration d is a property (that is, a boolean-valued function) $p_d(f, \boldsymbol{x})$ of pairs $(f, \boldsymbol{x})$, where f is a real-valued function with domain $\mathrm{Dom}(f) \subseteq \mathbb{R}^n$ for some $n \geq 0$ and $\boldsymbol{x} \in \overline{\mathbb{IR}}^n$ is an n-dimensional box, regarded as a subset of $\mathbb{R}^n$.

The notation $(f, \boldsymbol{x})$ unless said otherwise denotes such a pair, for arbitrary n, f and $\boldsymbol{x}$. Equivalently, d is identified with the set of pairs for which the property holds:

$$d = \{\, (f, \boldsymbol{x}) \mid p_d(f, \boldsymbol{x}) \text{ is true} \,\}. \tag{4}$$

The set $\mathbb{D}$ of decorations has five members:

Value	Short description	Property	Definition
`com`	common	$p_{\texttt{com}}(f, \boldsymbol{x})$	$\boldsymbol{x}$ is a bounded, nonempty subset of $\mathrm{Dom}(f)$; f is continuous at each point of $\boldsymbol{x}$; and the computed interval $f(\boldsymbol{x})$ is bounded.
`dac`	defined & continuous	$p_{\texttt{dac}}(f, x)$	$\boldsymbol{x}$ is a nonempty subset of $\mathrm{Dom}(f)$, and the restriction of f to $\boldsymbol{x}$ is continuous.
`def`	defined	$p_{\texttt{def}}(f, \boldsymbol{x})$	$\boldsymbol{x}$ is a nonempty subset of $\mathrm{Dom}(f)$.
`trv`	trivial	$p_{\texttt{trv}}(f, \boldsymbol{x})$	always true (so gives no information).
`ill`	ill-formed	$p_{\texttt{ill}}(f, \boldsymbol{x})$	Not an Interval; formally $\mathrm{Dom}(f) = \emptyset$, see 5.3.

(5)

These are listed according to the propagation order (10), which may also be thought of as a quality-order of $(f, \boldsymbol{x})$ pairs—decorations above `trv` are "good" and `ill` is "bad".

A **decorated interval** is a pair, written interchangeably as $(\boldsymbol{u}, d)$ or $\boldsymbol{u}_d$, where $\boldsymbol{u} \in \overline{\mathbb{IR}}$ is a real interval and $d \in \mathbb{D}$ is a decoration. $(\boldsymbol{u}, d)$ may also denote a decorated box $\big((\boldsymbol{u}_1, d_1), \ldots, (\boldsymbol{u}_n, d_n)\big)$, where $\boldsymbol{u}$ and d are the vectors of interval parts $\boldsymbol{u}_i$ and decoration parts d_i, respectively.

The set of decorated intervals is denoted by $\overline{\mathbb{DIR}}$, and the set of decorated boxes with n components is denoted by $\overline{\mathbb{DIR}}^n$.

When several named intervals are involved, the decorations attached to $\boldsymbol{u}, \boldsymbol{v}, \ldots$ are often named $du, dv, \ldots$ for readability, for instance $(\boldsymbol{u}, du)$ or $\boldsymbol{u}_{du}$, etc.

An interval may be called a **bare** interval to emphasize that it is not a decorated interval.

Treating the decorations as sets as in (4), `trv` is the set of all $(f, \boldsymbol{x})$ pairs, and the others are nonempty subsets of `trv`. By design, they satisfy the **exclusivity rule**

For any two decorations, either one contains the other or they are disjoint. (6)

Namely, the definitions (5) give:

$$\texttt{com} \subset \texttt{dac} \subset \texttt{def} \subset \texttt{trv} \supset \texttt{ill}, \qquad \text{note the change from } \subset \text{ to } \supset \tag{7}$$

$$\texttt{com}, \texttt{dac} \text{ and } \texttt{def} \text{ are disjoint from } \texttt{ill}. \tag{8}$$

Property (6) implies that for any $(f, \boldsymbol{x})$ there is a unique tightest (in the containment order (7)) decoration, such that $p_d(f, \boldsymbol{x})$ is true, called the **strongest decoration of** $(f, \boldsymbol{x})$, or of f over $\boldsymbol{x}$, and written $\text{Dec}(f \,|\, \boldsymbol{x})$. That is,

$$\text{Dec}(f \,|\, \boldsymbol{x}) = d \iff p_d(f, \boldsymbol{x}) \text{ holds, but } p_e(f, \boldsymbol{x}) \text{ fails for all } e \subset d.$$

NOTE—Like the exact range $\text{Rge}(f \,|\, \boldsymbol{x})$, the strongest decoration is theoretically well-defined, but its value for a particular f and $\boldsymbol{x}$ may be impractically expensive to compute, or even undecidable.

5.3 The ill-formed interval

An ill-formed decorated interval is also called NaI, **Not an Interval**. Conceptually, there shall be only one NaI. Its interval part has no value at Level 1.

The `ill` decoration results from invalid constructions and propagates unconditionally through arithmetic expressions. Namely, the `ill` decoration arises as a return value of

– a constructor when it cannot construct a valid decorated interval, or of

– a library arithmetic operation if and only if one of its inputs is ill-formed.

Formally, `ill` may be identified with the property $\text{Dom}(f) = \emptyset$ of $(f, \boldsymbol{x})$ pairs.

[Example. The constructor call `numsToInterval`$(2, 1)$ is invalid, so its decorated version returns NaI.]

Information may be stored in a NaI in an implementation-defined way (like the payload of an IEEE 754 floating-point NaN), and functions may be provided for a user to set and read this for diagnostic purposes. An implementation may provide means for an exception to be signaled when a NaI is produced.

5.4 Permitted combinations

A decorated interval $\boldsymbol{y}_{dy}$ shall always be such that

$$\boldsymbol{y} \supseteq \text{Rge}(f \,|\, \boldsymbol{x}) \text{ and } p_{dy}(f, \boldsymbol{x}) \text{ holds, for some } (f, \boldsymbol{x}).$$

If $dy =$ `dac`, `def` or `com`, then by definition $\boldsymbol{x}$ is nonempty, and f is everywhere defined on it, so that $\text{Rge}(f \,|\, \boldsymbol{x})$ is nonempty, implying $\boldsymbol{y}$ is nonempty. Hence the decorated intervals

$$\emptyset_{\texttt{dac}}, \ \emptyset_{\texttt{def}}, \ \text{ and } \boldsymbol{x}_{\texttt{com}} \text{ if } \boldsymbol{x} \text{ is empty or unbounded}$$

are contradictory—implementations shall not produce them.

No other combinations are forbidden.

5.5 Operations on/with decorations

5.5.1 Initializing

A bare interval is initialized with a decoration by the operation

$$\texttt{newDec}(\boldsymbol{x}) = \boldsymbol{x}_d \quad \text{where} \quad d = \begin{cases} \texttt{com} & \text{if } \boldsymbol{x} \text{ is nonempty and bounded,} \\ \texttt{dac} & \text{if } \boldsymbol{x} \text{ is unbounded, and} \\ \texttt{trv} & \text{if } \boldsymbol{x} \text{ is empty.} \end{cases}$$

5.5.2 Disassembling and assembling

For a decorated interval $\boldsymbol{x}_{dx}$, the operations $\texttt{intervalPart}(\boldsymbol{x}_{dx})$ and $\texttt{decorationPart}(\boldsymbol{x}_{dx})$ shall be provided, with value $\boldsymbol{x}$ and dx, respectively. For the case of NaI, $\texttt{decorationPart}(\text{NaI})$ has the value $\texttt{ill}$, but $\texttt{intervalPart}(\text{NaI})$ has no value at Level 1.

Given an interval $\boldsymbol{x}$ and a decoration dx, the operation $\texttt{setDec}(\boldsymbol{x}, dx)$ returns the decorated interval $\boldsymbol{x}_{dx}$ if this is an allowed combination. The cases of forbidden combinations are as follows:

– $\texttt{setDec}(\emptyset, dx)$, where dx is one of $\texttt{def}$, $\texttt{dac}$ or $\texttt{com}$, returns $\emptyset_{\texttt{trv}}$;

– $\texttt{setDec}(\boldsymbol{x}, \texttt{com})$, for any unbounded $\boldsymbol{x}$, returns $\boldsymbol{x}_{\texttt{dac}}$; and

– $\texttt{setDec}(\boldsymbol{x}, \texttt{ill})$ for any $\boldsymbol{x}$, whether empty or not, returns NaI.

5.5.3 Comparisons

For decorations, comparison operations for equality $=$ and its negation $\neq$ shall be provided, as well as comparisons $>, <, \geq, \leq$ with respect to the propagation order (10).

5.6 Decorations and arithmetic operations

Given a scalar point function φ of k variables, a **decorated interval extension** of φ—denoted here by the same name φ—adds a decoration component to a bare interval extension of φ. It has the form $\boldsymbol{w}_{dw} = \varphi(\boldsymbol{v}_{dv})$, where $\boldsymbol{v}_{dv} = (\boldsymbol{v}, dv)$ is a k-component decorated box $((\boldsymbol{v}_1, dv_1), \ldots, (\boldsymbol{v}_k, dv_k))$. By the definition of a bare interval extension, the interval part $\boldsymbol{w}$ depends only on the input intervals $\boldsymbol{v}$; the decoration part dw generally depends on both $\boldsymbol{v}$ and dv. In this context, NaI is regarded as being $\emptyset_{\texttt{ill}}$.

The definition of a bare interval extension implies

$$\boldsymbol{w} \supseteq \text{Rge}(\varphi \,|\, \boldsymbol{v}) \qquad \text{(enclosure)}.$$

The decorated interval extension of φ determines a dv_0 such that

$$p_{dv_0}(\varphi, \boldsymbol{v}) \text{ holds} \qquad \text{(a “local decoration”)}. \tag{9}$$

It then evaluates the output decoration dw by

$$dw = \min\{dv_0, dv_1, \ldots, dv_k\}, \qquad \text{(the “min-rule”)},$$

where the minimum is taken with respect to the **propagation order**:

$$\texttt{com} > \texttt{dac} > \texttt{def} > \texttt{trv} > \texttt{ill}. \tag{10}$$

5.7 Decoration of non-arithmetic operations

5.7.1 Interval-valued operations

This give interval results but are not interval extensions of point functions.

– the cancellative operations $\texttt{cancelPlus}(\boldsymbol{x}, \boldsymbol{y})$ and $\texttt{cancelMinus}(\boldsymbol{x}, \boldsymbol{y})$ of 4.5.3;

– the set-oriented operations $\texttt{intersection}(\boldsymbol{x}, \boldsymbol{y})$ and $\texttt{convexHull}(\boldsymbol{x}, \boldsymbol{y})$ of 4.5.4

No one way of decorating these operations gives useful information in all contexts. Therefore, a *trivial* decorated interval version is provided as follows. If any input is NaI, the result is NaI; otherwise the corresponding operation is applied to the interval parts of the inputs, and its result decorated with $\texttt{trv}$. The user may replace this by an appropriate nontrivial decoration via $\texttt{setDec}()$, see 5.5, where this can be deduced in a given application.

5.7.2 Non-interval-valued operations

These give non-interval results:

– the numeric functions of 4.5.6 and

– the boolean-valued functions of 4.5.7.

For each such operation, if any input is NaI, the result has no value at Level 1. Otherwise, the operation acts on decorated intervals by discarding the decoration and applying the corresponding bare interval operation.