Property Tracking with Decorations

Nathan T. Hayes
Sunfish Studio, LLC

May 28, 2011

Abstract

Property tracking with decorations can be seen as a complimentary extension to the Fundamental Theorem of Interval Arithmetic, where pessimism in the interval enclosure may cause the tracking decoration to be similarly conservative. As the interval enclosure becomes less pessimistic, the tracking decoration likewise becomes less conservative and more certain.

1 Introduction

This motion presents the author's view of decorations that is a culmination of many offline discussions occurring over the past five months. For a history of references leading to (or influencing) the development of this motion, see [1], [2], [4] as well as [7] and [6].

2 Motion Text

2.1 Properties

If f is a real function with domain D_f evaluated over interval box X, then the following properties are defined.

Definition 1 (Defined and Continuous)

$$C(f,X) \iff \textit{the restriction of } f \textit{ to } X \textit{ is defined and continuous.}$$

Note that by definition, $C(f,X)$ is true whenever X is empty.

Definition 2 (Domain Tetrit)

$$\begin{aligned} D(f,X)^{+} &\iff (\exists x \in X) : (x \in D_f)\,, \\ D(f,X)^{-} &\iff (\exists x \in X) : \neg\,(x \in D_f)\,. \end{aligned}$$

2.2 Decorations

A decoration is an element of the set

$$\{\mathbb{D}_0, \mathbb{D}_1, \mathbb{D}_2, \mathbb{D}_3, \mathbb{D}_4\} \tag{1}$$

such that each decoration represents one of the possible (valid) states of truth for all properties defined in Section 2.1. Below is a tabulated presentation of all possible (valid) combinations:

Decoration	$D(f, X)^+$	$D(f, X)^-$	$C(f, X)$
$\mathbb{D}_4$	F	F	T
$\mathbb{D}_3$	T	F	T
$\mathbb{D}_2$	T	F	F
$\mathbb{D}_1$	T	T	F
$\mathbb{D}_0$	F	T	F

Decorations have the linear quality order

$$\mathbb{D}_0 < \mathbb{D}_1 < \mathbb{D}_2 < \mathbb{D}_3 < \mathbb{D}_4. \tag{2}$$

The notation $S(f, X)$ is shorthand for the unique decoration associated with the function f when evaluated over the interval box X. For example

$$\begin{aligned} S(sqrt, Empty) &= \mathbb{D}_4, \\ S(sqrt, [0, 4]) &= \mathbb{D}_3, \\ S(sqrt, [-1, 2]) &= \mathbb{D}_1, \\ S(sqrt, [-3, -1]) &= \mathbb{D}_0. \end{aligned}$$

The more explicit notation $S(f, X_1, X_2, ..., X_n)$ may also be used if f has more than one operand.

2.3 Property Tracking

Definition 3 (Property Tracking with Decorations) *If f is any basic operation (addition, subtraction, multiplication, division, sqrt) or a recommended library function (sin, tan, log, etc.), and if (X_1, D_1), (X_2, D_2), ..., (X_n, D_n) are the decorated interval operands of f, then the historical (tracking) decoration of f is the greatest element that is less-than or equal to all the elements of the set*

$$\{S(f, X_1, X_2, ..., X_n), D_1, D_2, ..., D_n\}.$$

If $(X, D) = ((X_1, D_1), (X_2, D_2), ..., (X_n, Dn))$ is a decorated interval vector, the notation $T(f, (X, D))$ is shorthand for the tracking decoration

$$T(f, (X, D)) = inf(S(f, X_1, X_2, ..., X_3), D_1, D_2, ..., D_n). \tag{3}$$

Note that unlike the decoration $S(f, X)$ of the current operation, the tracking decoration $T(f, (X, D))$ depends on both $S(f, X)$ as well as the decorations of the input operands.

2.4 The Min and Max Operations

For purposes of decorations and property tracking, the interval operations *min* and *max* shall be treated as normal binary arithmetic operations whose natural domain is everywhere defined and continuous. In other words, they shall be treated the same as, say, the addition operation.

3 Rationale

The Fundamental Theorem of Interval Arithmetic (FTIA) says the naive interval extension of any real expression will be a valid though possibly pessimistic enclosure of the optimal range of the expression. Property tracking with decorations is a proof by structural induction that likewise gives a conservative decoration for such pessimistic enclosures; and in the case the enclosure is optimal, the decoration is also exact.

The source of pessimism in FTIA is unrecognized interval dependence. This phenomenon is well-known in the interval community, and occurs when more than one instance of a variable appears in the syntax tree of an algebraic expression. The unrecognized interval dependence between the multiple instances of the variable cause excess widening in the computed interval results. Though pessimistic, the widened results still contain the optimal range of the expression.

Even in the face of pessimistic results caused by unrecognized interval dependence during evaluation of an expression, the following table is a summary of facts about the tracking decoration for such an expression:

Tracked Decoration	Result Interval	Meaning
D4	Empty	All independent variables were empty
D4	Nonempty	By definition, this case can never occur
D3	Empty	At least one independent variable was nonempty and at least one independent variable was empty; the expression is certainly defined and continuous
D3	Nonempty	All independent variables were nonempty and the expression is certainly defined and continuous
D2	Empty	At least one independent variable was nonempty and at least one independent variable was empty; the expression is certainly defined
D2	Nonempty	All independent variables were nonempty and the expression is certainly defined

```
D1          Empty     At least one independent variable was
nonempty and at least one independent variable was empty; the
expression is possibly defined or possibly undefined

D1          Nonempty  All independent variables were nonempty
and the expression is possibly defined

D0          Empty     At least one independent variable was
nonempty and at least one independent variable was empty; the
expression is certainly undefined

D0          Nonempty  All independent variables were nonempty
and the expression is certainly undefined
```

As is the case with FTIA, Computer Algebra Systems (CAS) can often be employed to perform expression rearrangement in order to find less pessimistic enclosures of an expression. For property tracking, finding less pessimistic enclosures may likewise cause a conservative tracking decoration ($\mathbb{D}_1$) of an expression to change to an exact decoration ($\mathbb{D}_0$, $\mathbb{D}_2$, or $\mathbb{D}_3$).

Sometimes finding narrow or optimal enclosures is intractable, even with the aid of a very powerful CAS. In this case, branch-and-bound is another application of FTIA that can be used to compute less pessimistic enclosures: as the domain of the independent variables is bisected, the widening due to unrecognized interval dependence becomes less pronounced. As the bisection is repeated iteratively the amount of pessimism that might remain eventually becomes smaller than some acceptable tolerance specified by the user. In this scenario, the conservative tracking decoration ($\mathbb{D}_1$) of an expression may likewise change to an exact decoration ($\mathbb{D}_0$, $\mathbb{D}_2$, or $\mathbb{D}_3$) when evaluated over the various subsets of the bisected domain of the independent variables.

3.1 Lattice Operations

An original motivation in [1] of introducing decorations was to provide a semantic difference between the result of intersecting two disjoint and nonempty intervals. Such a result can in some circumstances be considered “good,” but in other circumstances may be considered an error.

Property tracking with decorations can provide both semantics, where the semantic of the result depends on the decorations of the input operands. If one assumes that lattice operations min, max and intersection behave the same as arithmetic operations, then one may have by Definition 3, e.g.,

$$([1, 2], \mathbb{D}_0) \cap ([3, 4], \mathbb{D}_0) = (\varnothing, \mathbb{D}_0).$$

In this case, the empty result is “bad” because both operands were everywhere undefined. This in contrast to

$$([1,2], \mathbb{D}_3) \cap ([3,4], \mathbb{D}_3) = (\varnothing, \mathbb{D}_3)$$

where the empty result is "good" because both operands were everywhere defined and continuous.

These semantics of the intersection operation described above are required to avoid the unpredictable failure of certain kinds of interval algorithms, as demonstrated in [3]. The same semantics are also required by the min and max operations for tracking the decorations of intersection and unions of implicit functions in branch-and-bound algorithms.

3.2 Boundedness

This motion intentionally leaves the property of "boundedness" open for a future motion. One feasible possibility would be that no further properties or decorations are added to track unbounded intervals. Rather, boundedness could be an intrinsic property of the bare interval portion of a decorated interval by allowing the interval endpoints to distinguish between infinity and overflow. Such a proposal has been discussed in the P1788 several times, and a description of the necessary arithmetic has been presented by Ian McIntosh. It has also been noted this may have the additional benefit of possibly making IEEE 754 a better standard.

3.3 Towards a Foundation of Decorations

Decorations have an intrinsic linear quality order, such as (2). It was proposed in [5] that decorations also possess a (partial) containment order, leading to a *Fundamental Theorem of Decorated Interval Arithmetic,* or FTDIA for short. The definitions for decorations presented in this current motion are different than those in [5]; a few reasons why have been outlined in this paper. A new containment order is not yet presented in this motion. In our laboratory, we have developed a working prototype based on the new definitions presented in this motion. Empirical verification of these definitions lead us to believe that a new FTDIA does exist for these new definitions, since clearly the prototype should fail if this was not true. We solicit the help of our resident mathematicians to help put this on paper and give a formal proof.

3.4 Conclusion

Property tracking with decorations can be seen as a complimentary extension to FTIA where pessimism in the interval enclosure may cause the tracking decoration to be similarly conservative. Various well-known methods may be employed to find less pessimistic interval enclosures of an interval expression. These methods include expression rearrangement as well as branch-and-bound. As the interval enclosures become less pessimistic, the tracking decoration may likewise become less conservative and more certain.

References

[1] Hayes, N. and A. Neumaier, “Exception Handling for Interval Arithmetic,” P1788 Motion 8.02, Oct. 13, 2009

[2] Hayes, N. “Trits to Tetrits,” P1788 Motion 18.02, May 28, 2010

[3] Hayes, N., Mail to P1788, Jan. 20, 2011
http://grouper.ieee.org/groups/1788/email/msg03570.html

[4] Hayes, N. “Exception Handling for Interval Arithmetic,” DRAFT position paper, Nov. 14, 2010
http://grouper.ieee.org/groups/1788/email/msg03506.html

[5] Pryce, J., DRAFT P1788 Standard, v3.01, Jan. 14, 2011

[6] Tupper, J. “Graphing Equations with Generalized Interval Arithmetic,” Ph. D. Thesis, 1996
http://www.dgp.toronto.edu/~mooncake/msc.html

[7] Walster, G. W., “Empty Intervals,” Technical Report, Sun Microsystems, April 1998