Decorations for Affine Arithmetic

Stefan Kiel

26 Mai 2011

Contents

1 Introduction

`YalAA` [6] is a library for performing affine calculation. The library allows to plug in various policies for handling errors occurring during the calculation through a user defined `ErrorPolicy`. In this paper we consider the use of decorations, a concept that came recently up during the standardization process for interval arithmetic (IEEE P1788) [5], [3], [2], [4], [7], for error handling in `YalAA` . In the remaining sections we assume that the reader is familiar with the decoration concept for intervals.

2 Decorations

2.1 Current Status in P1788

The P1788 standard is still under heavy discussion. However, on some aspects of exception handling has already been voted on.

- Use of decorations (Motion 8) [5]
- Domain tetrit (Motion 18) [3]

While both motions have some ideas for concrete decorations, their only definite decision is the use of decoration and the domain tetrit. Some more details are given in the position paper [2]. The author proposes two further decorations:

- Defined and Continuous $C(f, X)$
- Defined and Bounded $B(f, X)$

Both are combined into a single decoration $CB(f, X) = C(f, X) \wedge B(f, X)$. However, in a motion currently under discussion by the same author the bounded decoration is left out and only $C(f, X)$ is considered [4].

2.2 Affine Decorations

The `ErrorPolDec` policy class provides two types of decoration. Decorations directly associated with function graph are called *F-decoration.* They make claims about the function f independently of the used implementation. If a decoration is also dependent on the implementation it's called a *C-decoration.* Supported *F-decorations*:

- D(f, X): `tetrit`, function f is defined over X
- CD(f, X) `bool`, function f is continuous and defined over X
- BD(f, X): `bool`, function f is bounded and defined over X

Supported *C-decorations*:

- VR$(f, X,$ `aff_t`$)$: `bool`, f has a valid range over X in the current implementation (B$(f, X) \wedge \neg$`Overflow`$\wedge \neg$NE)
- NE$(f, X,$ `aff_t`$)$: `bool`, no internal error occurred

B(f, X) denotes that the function is *bounded* over the intersection of X with its natural domain.

2.3 Discussion

The F-decorations are consistent with the decoration definition considered by the P1788 working group. The *domain tetrit* $\mathrm{D}(f, X)$ is defined as for IA in [3] and the *continuous and defined* flag $\mathrm{CD}(f, X)$ as described in [4]. Furthermore, we provide the $\mathrm{BD}(f, X)$ decorations meaning *bounded and defined* [2]. Note that the considerations in [4] regarding the bounded flag[1] are not relevant in the affine context.

The C-decorations not only depend on the function f and the affine form X but also on the concrete arithmetic implementation[2]. `YalAA` allows the user to detect overflows during computation through the $\mathrm{VR}(f, X, \texttt{aff\_t})$ decoration. Furthermore, an *internal error* flag $\mathrm{NE}(f, X, \texttt{aff\_t})$ is used for signaling an unexpected failure during the computation. We would like to point out that both flags make *no* claims on the function f but on its implementation.

3 Possible Combinations

Per definition we have the relationships

- $\mathrm{CD} \rightarrow \mathrm{D}^+ \wedge \neg\mathrm{D}^- \vee \neg(\mathrm{D}^+ \vee \mathrm{D}^-)$
- $\mathrm{BD} \rightarrow \mathrm{D}^+ \wedge \neg\mathrm{D}^- \vee \neg(\mathrm{D}^+ \vee \mathrm{D}^-)$

between F-decorations[3]. Furthermore, we can deduce:

- $\mathrm{CD} \rightarrow \mathrm{BD}$ (”every continuous function over an closed set is bounded”)

This leads to the possible combinations:

D^+	D^-	CD	BD	Dec.
F	F	T	T	$\mathbb{D}_5$
T	F	T	T	$\mathbb{D}_4$
T	F	F	T	$\mathbb{D}_3$
T	F	F	F	$\mathbb{D}_2$
T	T	F	F	$\mathbb{D}_1$
F	T	F	F	$\mathbb{D}_0$

[1]Currently it is discussed if unbounded intervals can be represented by using ∞ in interval bounds. However, as there is currently no distinction between *real infinities* and overflows in IEEE754, either one of these cases has to be a decoration or an additional infinity value is to be added for distinction of these cases.

[2]in `YalAA` especially on the used types for partial deviations and policies

[3]Following [4] functions are per definition continuous and bounded over the empty set.

The valid range flag is linked with the bounded property B, the domain tetrit D and the general error flag NE:

- $\neg B \to \neg VR$
- $\neg D^+ \to \neg VR$
- $\neg NE \to \neg VR$

This results in the following possible combinations:

D^+	(B)	VR	NE	Dec.
T	T	T	T	$\mathfrak{D}_2$
T	T	F	T	$\mathfrak{D}_1$
T	F	F	T	$\mathfrak{D}_1$
F	?	F	T	$\mathbb{D}_0$
?	?	F	F	$\mathfrak{D}_0$

As the bounded flag B is invisible to the user, two cases result in the same decoration $\mathfrak{D}_1$. The second last case is identically to $\mathbb{D}_0$, so no new decoration is introduced here. The *internal error* flag NE is a special case. If NE is false, all other flags are in an undefined state with the exception of VR which is also false in this case. This convention allows checking only the VR flag in order to detect whether the computation was error-free and the affine part has a meaningful value.

4 Meanings

The F-decorations make claims about the function f over X:

Dec	Description
$\mathbb{D}_5$	X is the empty set
$\mathbb{D}_4$	f is certainly defined, continuous and bounded over X
$\mathbb{D}_3$	f is certainly defined bounded over X
$\mathbb{D}_2$	f is certainly defined and unbounded over X
$\mathbb{D}_1$	f is possibly defined over X
$\mathbb{D}_0$	f is certainly undefined over X

Just like their archetypes in [4], [2] they are ordered:

$$\mathbb{D}_0 < \mathbb{D}_1 < \mathbb{D}_2 < \mathbb{D}_3 < \mathbb{D}_4 < \mathbb{D}_5$$

by their quality. If a computation yields $\mathbb{D}_i$ over X it is guaranteed that the same computation over some subbox $X' \subseteq X$ yields a decorations $\mathbb{D}_j$ with $j \geq i$. `ErrorPolDec` supports property tracking as defined in [4] Def. 3. It is also possible to interpret the C-decorations:

Dec.	Description
$\mathfrak{D}_2$	No errors during calculation
$\mathfrak{D}_1$	Unbounded and/or Overflow
$\mathfrak{D}_0$	Internal error

5 Mapping to Standard Affine Arithmetic

AA as described by de Figueiredo and Stolfi [1] provides two special values for error handling:

- **R** denotes the whole real line
- **[]** denotes the empty affine form (set)

We define the following mapping from the decorations to these special values:

Dec.	Special Value
$\mathbb{D}_5, \mathbb{D}_0$	**[]**
$\mathbb{D}_2, \mathfrak{D}_1$	**R**
$\mathfrak{D}_0$	**[]**

This maps all error conditions to their respective counterparts in the affine model. Mapping $\mathfrak{D}_0$ to **[]** may look a bit controverse at the first glance. However, as the special values refer to both the mathematical function and the implementation[4] we can interpret $\mathfrak{D}_0$ as if the implementation is not defined over X.

6 Implementation in YalAA

Following the original approach of [1] every affine form in `YalAA` has a field to store whether the form has a special value. Its type is determined by the used `ErrorPolicy` [6]. `ErrorPolDec` defines this type to `unsigned short`. The F- and C-decorations are stored separately. While the first three bits store the F-decoration in form of the $\mathbb{D}_i, 0 \leq i \leq 5$ defined above, the following two store the VR and the NE flags separately.

[4]R is used for indicating overflows in [1].

References

[1] L.H. de Figueiredo and J. Stolfi. *Self-Validated Numerical Methods and Applications*. IMPA, Rio de Janeiro, 1997.

[2] N. T. Hayes. Exception handling for interval arithmetic, 2010. P1788, Position Paper.

[3] N. T. Hayes. Trits to tretris, 2010. P1788, Motion 18.

[4] N. T. Hayes. Property tracking with decorations, May 2011. P1788, Proposed Motion.

[5] N. T. Hayes and A. Neumaier. Exception handling for interval arithmetic. P1788, Motion 8.

[6] Stefan Kiel. yalaa - yet another library for affine arithmetic implementation manual, 2011.

[7] Vladik Kreinovich. How to check, while performing straightforward interval computations, whether the resulting function is continuous, everywhere defined, etc.: Towards foundations of decorations, 2011.