Basic building blocks for a triple-double intermediate format

Lauter, Christoph Quirin

Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n° 5668

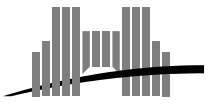
Basic building blocks for a triple-double intermediate format

Christoph Quirin Lauter

Septembre 2005

Research Report N° RR2005-38

École Normale Supérieure de Lyon
46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37
Télécopieur : +33(0)4.72.72.80.80
Adresse électronique : lip@ens-lyon.fr



Basic building blocks for a triple-double intermediate format

Christoph Quirin Lauter

Septembre 2005

Abstract

The implementation of correctly rounded elementary functions needs high intermediate accuracy before final rounding. This accuracy can be provided by (pseudo-) expansions of size three, i.e. a triple-double format.
The report presents all basic operators for such a format. Triple-double numbers can be redundant. A renormalization procedure is presented and proven. Elementary functions' implementations need addition and multiplication sequences. These operators must take operands in double, double-double and triple-double format. The results must be accordingly in one of the formats. Several procedures are presented. Proofs are given for their accuracy bounds. Intermediate triple-double results must finally be correctly rounded to double precision. Two effective rounding sequences are presented, one for round-to-nearest mode, one for the directed rounding modes. Their complete proofs constitute half of the report.

Keywords: elementary functions, multiple precision, expansions, correct rounding

Résumé

La mise en œuvre de fonctions élémentaires correctement arrondies nécessite l'utilisation d'un format intermédiaire de haute précision avant l'arrondi final. Cette précision peut être pourvue par des (pseudo-)expansions de taille trois, c'est-à-dire par un format triple-double.
Ce rapport présente tous les opérateurs de base d'un tel format. Le format des nombres triple double est redondant, aussi une procédure de renormalisation est-elle présentée et prouvée. La mise en œuvre de fonctions élémentaires a besoin de séquences d'addition et de multiplication. Ces opérateurs doivent être capables de prendre en argument des opérandes de format double, double-double ou triple-double. Leurs résultats doivent être dans un des formats correspondants. Un certain nombre de procédures sont présentées pour ces opérations avec des bornes prouvées pour leur précision.
Les résultats intermédiaires en triple-double doivent finalement être arrondis correctement vers la double précision. Deux séquences d'arrondi final efficaces sont présentées, une pour l'arrondi au plus près, une autre pour les modes d'arrondi dirigés. Leur preuve complète constitue la moitié du rapport.

Mots-clés: fonctions élémentaires, précision multiple, expansions, arrondi correct

1 Introduction

The implementation of correctly rounded double precision elementary functions needs high accuracy intermediate formats [7, 4, 1, 3]. To give an order of magnitude, in most cases 120 bits of intermediate accuracy are needed for assuring the correct rounding property [3]. In contrast, the native IEEE 754 double precision format offers 53 bits of accuracy. Well-know techniques allow to double this accuracy [5]. Nevertheless the resulting accuracy, 106 bits, is not sufficient. Tripling it would be enough.

Using expansions of floating point numbers allows to expand still more the accuracy of a native floating point type. An expansion is a non-evaluated sum of some floating point numbers in a floating point format [6, 9, 10]. However the techniques for manipulating general expansions presented in literature are too costly for the implementation of elementary functions.

In this report, we are going to consider expansions of three double precision floating point numbers, i.e. we are going to investigate on a triple double format. In our case, we will hence manipulate floating point numbers $(x_h + x_m + x_l)$ with $x_h, x_m, x_l \in \mathbb{F}$.

After making some definitions and an analysis of a procedure that we call renormalization, we will consider addition and multiplication procedures for triple-double numbers. These will also comprise operations linking the triple double format with the native double and a double-double format.

In the end, we will present and prove in particular two final rounding sequences for correctly rounding a triple-double number to the base double format.

2 Definitions

It will be necessary to define a normal form of a triple-double number because it is clear that the triple-double floating point format - or that of (pseudo-)expansions in general - is redundant: there exist numbers $\hat{x}$ such that there are different triplets $(x_h, x_m, x_l) \in \mathbb{F}^3$ such that $\hat{x} = x_h + x_m + x_l$. It is easy to see that a representation in such a format is unique if the numbers forming the expansion are ordered by decreasing 0 and if the latter are such that there is no (binary) digit represented by two bits of the mantissas of two different numbers of the expansion. The sign bit has in this case the same role as an additional bit of the significant [6].

As we will see, the need for a normal form for each triple-double number is not directly motivated by needs of the addition and multiplication operators we will have to define. As long as the latter operate correctly, i.e. between known error bounds, on numbers whose representation in a triple-double format is not unique, we are not obliged to recompute normal forms. On the other hand, when we want to round down such a higher precision number to a native double (in one of the different rounding modes), a normal form will be needed. If we could not provide one, we would assist to a explosion of different cases to be handled by the rounding sequence.
We shall define therefore:

Definition 2.1 (Overlap)
Let $x_h, x_l \in \mathbb{F}$ be two non-subnormal double precision numbers.
We will say that x_h et x_l overlap iff

$$|x_l| \geq 2^{-52} \cdot |x_h|$$

Let be $x_h, x_m, x_l \in \mathbb{F}$ the components of a triple-double number. We will suppose that they are not subnormals.
So, we will say that there is overlap iff x_h and x_m or x_m and x_l or x_h and x_l overlap.

Definition 2.2 (Normal form)
Let be $x_h, x_m, x_l \in \mathbb{F}$ three non-subnormal floating point numbers forming the triple-double number $x_h + x_m + x_l$.

We will say that $x_h + x_m + x_l$ are in normal form iff there is no overlap between its components. Further, we will say that $x_h + x_m + x_l$ is normalised.

Having made this definition, let us remark that it is deeply inspired by our direct needs and not by abstract analysises on how to represent real numbers. Anyway, the fact that an expansion is not overlapping is not a sufficient condition for its representing an unique floating point number.

The implementation of addition and multiplication operators will be finally be based on computations on the components of the operands (or partial products in the case of a multiplication) followed by a final summing up for regaining the triple-double base format. As we do not statically know the magnitudes of the triple-double operands and its components, we will not be able to guarantee that in any case there will not be any overlap in the result. On the other hand, we strive to develop a renormalization sequence for recomputing normal forms. These will be handy, as we already said, for the final rounding and, if needed, i.e. if sufficiently good static bounds can not be given, before handing over a result as an operand to a following operation. Such a renormalization sequence must guarantee that for each triple-double in argument (if needed verifying some preconditions on an initial overlap), the triple-double number returned will be normalised and that the sum of the components for the first number is exactly the same as the one of the components of the latter.

Definition 2.3 (The Add12 algorithm)
Let **A** *be an algorithm taking as arguments two double precision numbers $x, y \in \mathbb{F}$ and returning two double precision floating point numbers $r_h, r_l \in \mathbb{F}$.*
We will call **A** *the* **Add12** *algorithm iff it verifies that*

- $\forall x, y \in \mathbb{F}. r_h + r_l = x + y$
- $\forall x, y \in \mathbb{F}. |r_l| \leq 2^{-53} \cdot |r_h|$
- $r_h = \circ(x + y)$
  $r_l = x + y - r_h$

i.e. iff it makes an exact addition of two floating point numbers such that the components of the double-double expansion in result are non-overlapping and if the most significant one is the floating point number nearest to the sum of the numbers in argument.

Definition 2.4 (The Mul12 algorithm)
Let **A** *be an algorithm taking as arguments two double precision numbers $x, y \in \mathbb{F}$ and returning two double precision floating point numbers $r_h, r_l \in \mathbb{F}$.*
We will call **A** *the* **Mul12** *algorithm iff it verifies that*

- $\forall x, y \in \mathbb{F}. r_h + r_l = x \cdot y$
- $\forall x, y \in \mathbb{F}. |r_l| \leq 2^{-53} \cdot |r_h|$
- $r_h = \circ(x \cdot y)$
  $r_l = x \cdot y - r_h$

i.e. iff it makes an exact multiplication of two floating point numbers such that the components of the double-double expansion in result are non-overlapping and if the most significant one is the floating point number nearest to the sum of the numbers in argument.

We will pass over the existence proof of such algorithms. Consult [5] on this subject.

Let us still define some notations that are needed for the analysis of numerical algorithms like the ones that we are going to consider.

Definition 2.5 (Predecessor and successor of a floating point number)
Let be $x \in \mathbb{F}$ a floating point number. Let be $<$ the total ordering on $\mathbb{F}$.
If x is positive or zero we will design by x^+ the direct successor of x in $\mathbb{F}$ with regard to $<$ and we will notate x^- its predecessor.
If x is negative we will design by x^- the successor and by x^+ the predecessor of x.
In any case, we will design by $\mathrm{succ}(x)$ the successor of x in $\mathbb{F}$ with regard to $<$ and $\mathrm{pred}(x)$ its predecessor.

This definition 2.5 is inspired by [4].

Definition 2.6 (Unit on the last place – the ulp function)
Let be $x \in \mathbb{F}$ a double precision number and let be x^+ its successor (resp. predecessor if x is negative).
So

$$\mathrm{ulp}(x) = \begin{cases} x^+ - x & \text{if} \quad x \geq 0 \text{ and } x^+ \neq +\infty \\ 2 \cdot \mathrm{ulp}\left(\frac{x}{2}\right) & \text{if} \quad x \neq +\infty \text{ but } x^+ = +\infty \\ \mathrm{ulp}(-x) & \text{if} \quad x < 0 \end{cases}$$

This definition is inspired by [4], too. Compare [8] for further research on the subject of the ulp function.

Let us give already some main lemmas that can be deduced from the definition 2.6 of the ulp function.

Lemma 2.7 (The ulp functions with regard to upper bounds)
Let be x_h and x_l two non-subnormal floating point numbers.
So

$$|x_l| < \mathrm{ulp}(x_h) \Leftrightarrow |x_l| \leq 2^{-52} \cdot |x_h|$$

Proof
"$\Rightarrow$":
Let us suppose that $|x_l| < \mathrm{ulp}(x_h)$ and that $|x_l| > 2^{-52} \cdot |x_h|$.
So we get the following inequality

$$2^{-52} \cdot |x_h| < \mathrm{ulp}(x_h)$$

Without loss of generality, let us suppose now that $x_h > 0$ and that $x_h^+ \neq +\infty$ where x_h^+ is the successor of x_h in the ordered set of floating point numbers.
So we know by the definition of non-subnormal floating point number in double precision that there exists $m \in \mathbb{N}$ et $e \in \mathbb{Z}$ such that $x_h = 2^e \cdot m$ with $2^{52} \leq m < 2^{53}$. Anyway, one can check that

$$x_h^+ = \begin{cases} 2^e \cdot (m+1) & \text{if} \quad m+1 < 2^{53} \\ 2^{e+1} \cdot 2^{52} & \text{otherwise} \end{cases}$$

So 2 cases must be treated separately:

1st case: $x_h^+ = 2^e \cdot (m+1)$
So we get

$$\begin{aligned} 2^e \cdot (m+1) - 2^e \cdot m &> 2^{-52} \cdot 2^e \cdot m \\ 1 &> 2^{-52} \cdot m \end{aligned}$$

In contrast, $m \geq 2^{52}$, so we obtain the strict inequality $1 > 1$ which contradicts the hypotheses.

2nd case: $x_h^+ = 2^{e+1} \cdot 2^{52}$

In this case, we know that $m = 2^{53} - 1$.
We can deduce that

$$\begin{aligned} 2^{e+1} \cdot 2^{52} - 2^e \cdot \left(2^{53} - 1\right) &> 2^{-52} \cdot 2^e \cdot \left(2^{53} - 1\right) \\ 1 &> 2 - 2^{-52} \end{aligned}$$

This last inequality is a direct contradiction with the hypotheses.

"$\Leftarrow$":
Let us suppose now that $|x_l| \leq 2^{-52} \cdot |x_h|$ and that $|x_l| > \text{ulp}(x_h)$.
So we get

$$\text{ulp}(x_h) < 2^{-52} \cdot |x_h|$$

Without loss of generality, let us suppose that $x_h \geq 0$ and that $x_h^+ \neq +\infty$ where x_h^+ is the successor of x_h in the ordered set of double precision floating point numbers.
Thus

$$\begin{aligned} x_h^+ - x_h &< 2^{-52} \cdot x_h \\ x_h^+ &< x_h \cdot \left(1 + 2^{-52}\right) \end{aligned}$$

Thus, we get or a floating point number equal to 0 whose successor is equal to 0, too, or a double precision floating point number whose mantissa contains more than 53 bits because the inequality is strict. It is clear that we have got a contradiction in both cases. ∎

Lemma 2.8 (Commutativity of the x^+ and x^- operators with unary $-$)
Let be $x \in \mathbb{F}$ a positive floating point number.
So,

$$(-x)^+ = -\left(x^+\right)$$

and

$$(-x)^- = -\left(x^-\right)$$

Proof
Since the set of the floating point numbers is symmetrical around 0, we get

$$(-x)^+ = \text{pred}(-x) = -\text{succ}(x) = -\left(x^+\right)$$

and

$$(-x)^- = \text{succ}(-x) = -\text{pred}(x) = -\left(x^-\right)$$

∎

Lemma 2.9 (x^+ and x^- for an integer power of 2)
Let be $x \in \mathbb{F}$ a non-subnormal floating point number such that it exists $e \in \mathbb{Z}$ such that

$$x = \pm 2^e \cdot 2^p$$

where $p \geq 2$ is the format's precision.
So,

$$x - x^- = \frac{1}{2} \cdot \left(x^+ - x\right)$$

Proof
If $x > 0$, we get

$$x - x^- = 2^e \cdot 2^p - 2^{e-1} \cdot \left(2^{p+1} - 1\right) = 2^{e-1}$$

and

$$x^+ - x = 2^e \cdot (2^p + 1) - 2^e \cdot 2^p = 2^e$$

If x is negative it suffices to apply lemma 2.8. ∎

Lemma 2.10 (x^+ and x^- for a float different from an integer power of 2)
Let be $x \in \mathbb{F}$ a non-subnormal floating point number such that it does not exist any $e \in \mathbb{Z}$ such that

$$x = \pm 2^e \cdot 2^p$$

where $p \geq 2$ is the format's precision.
So,

$$x - x^- = x^+ - x$$

Proof
If $x > 0$ we know that there exist $e \in \mathbb{Z}$ and $m \in \mathbb{N}$ such that

$$x = 2^e \cdot m$$

with

$$2^p < m < 2^{p-1}$$

because x is not exactly an integer power of 2.
Further, one checks that

$$x^+ = 2^e \cdot (m+1)$$

even if $m = 2^{p-1} - 1$ and that

$$x^- = 2^e \cdot (m-1)$$

because the lower bound given for m is strict.
So one gets

$$\begin{aligned} x - x^- &= 2^e \cdot m - 2^e \cdot (m-1) \\ &= 2^e \\ &= 2^e \cdot (m+1) - 2^e \cdot m \\ &= x^+ - x \end{aligned}$$

If x is negative it suffices to apply lemma 2.8. ∎

Lemma 2.11 (Factorized integer powers of 2 and the operators x^+ and x^-)
Let be $x \in \mathbb{F}$ a non-subnormal floating point number such that $\frac{1}{2} \cdot x$ is still not subnormal.
So,

$$\left(\frac{1}{2} \cdot x\right)^+ = \frac{1}{2} \cdot x^+$$

and

$$\left(\frac{1}{2} \cdot x\right)^- = \frac{1}{2} \cdot x^-$$

Proof
Without loss of generality let us suppose that x is positive. Otherwise we easily apply lemma 2.8.
So, if x can be written $x = 2^e \cdot m$ with $m + 1 \leq 2^{p+1}$ where p is the precision then

$$\left(\frac{1}{2} \cdot x\right)^+ = \left(2^{e-1} \cdot m\right)^+ = 2^{e-1} \cdot (m+1) = \frac{1}{2} \cdot x^+$$

Otherwise,

$$\left(\frac{1}{2} \cdot x\right)^+ = \left(2^{e-1} \cdot \left(2^{p+1} - 1\right)\right)^+ = 2^{e-1+1} \cdot 2^p = \frac{1}{2} \cdot x^+$$

One can check that one obtains a completely analogous result for x^-. ∎

Lemma 2.12 (Factor 3 of an integer power of 2 in argument of the x^+ operator)
Let be $x \in \mathbb{F}$ a positive floating point number such that x is not subnormal, x^+ and $(3 \cdot x)^+$ are different from $+\infty$, and that $\exists e \in \mathbb{Z}\ .\ x = 2^e \cdot 2^p$ where $p \geq 3$ is the precision of the format $\mathbb{F}$. So the following equation holds

$$(3 \cdot x)^+ + \mathrm{ulp}\,(x) = 3 \cdot x^+$$

Proof
We can easily check the following

$$\begin{aligned}
(3 \cdot x)^+ + \mathrm{ulp}\,(x) &= (3 \cdot x) + \left(x^+ - x\right) \\
&= (3 \cdot 2^e \cdot 2^p)^+ + (2^e \cdot 2^p)^+ - 2^e \cdot 2^p \\
&= ((2+1) \cdot 2^e \cdot 2^p)^+ + 2^e \cdot (2^p + 1) - 2^e \cdot 2^p \\
&= \left(2 \cdot 2^e \cdot 2^p + 2 \cdot \frac{1}{2} \cdot 2^e \cdot 2^p\right)^+ + 2^e \\
&= \left(2^{e+1} \cdot \left(2^p + 2^{p-1}\right)\right)^+ + 2^e \\
&= 2^{e+1} \cdot \left(2^p + 2^{p-1} + 1\right) + 2^e \\
&= 2^{e+1} \cdot \left(2^p + 2^{p-1}\right) + 2^{e+1} + 2^e \\
&= 2^{e+1} \cdot \left(2^p + 2^{p-1}\right) + 3 \cdot 2^e \\
&= 3 \cdot 2^e \cdot 2^p + 3 \cdot 2^e \\
&= 3 \cdot 2^e \cdot (2^p + 1) \\
&= 3 \cdot (2^e \cdot 2^p)^+ \\
&= 3 \cdot x^+
\end{aligned}$$

∎

Lemma 2.13 (Monotony of the ulp function)
The ulp *function is monotonic for non-subnormal positive floating point numbers and it is monotonic for non-subnormal negative floating point numbers, i.e.*

$$\forall x, y \in \mathbb{F}\ .\ \mathit{denorm} < x \leq y \Rightarrow \mathrm{ulp}\,(x) \leq \mathrm{ulp}\,(y)$$

$$\vee$$

$$\forall x, y \in \mathbb{F}\ .\ x \leq y < -\mathit{denorm} \Rightarrow \mathrm{ulp}\,(x) \geq \mathrm{ulp}\,(y)$$

where denorm is the greatest positive subnormal.

Proof
As a matter of fact it suffices to show that the ulp function is monotonic for non-subnormal floating point numbers and to apply its definition 2.6 for the negative case.
Let us suppose so that we have two floating point numbers $x, y \in \mathbb{F}$ such that $\mathit{denorm} < x < y$. Without loss of generality we suppose that $x^+ \neq +\infty$ and that $y^+ \neq +\infty$. Otherwise we apply definition 2.6 of the ulp function and lemma 2.11.
So we get

$$\mathrm{ulp}\,(y) - \mathrm{ulp}\,(x) = y^+ - y - x^+ + x$$

It suffices now to show that

$$y^+ - x^+ - y + x \geq 0$$

We can suppose that we would have

$$\begin{aligned}
x &= 2^{e_x} \cdot m_x \\
y &= 2^{e_y} \cdot m_y
\end{aligned}$$

with $e_x, e_y \in \mathbb{Z}$, $2^p \leq m_x, m_y < 2^{p+1} - 1$.
Since $y > x$, we clearly see that

$$(e_y, m_y) \geq_{\text{lex}} (e_x, m_x)$$

So four different cases are possible:

1. $$\begin{aligned} x^+ &= 2^{e_x} \cdot (m_x + 1) \\ y^+ &= 2^{e_y} \cdot (m_y + 1) \end{aligned}$$

   Hence

   $$\begin{aligned} y^+ - x^+ - y + x &= 2^{e_y} \cdot (m_y + 1) - 2^{e_x} \cdot (m_x + 1) - 2^{e_y} \cdot m_y + 2^{e_x} \cdot m_x \\ &= 2^{e_y} - 2^{e_x} \\ &\geq 0 \end{aligned}$$

2. $$\begin{aligned} x^+ &= 2^{e_x} \cdot (m_x + 1) \\ y^+ &= 2^{e_y+1} \cdot 2^p \end{aligned}$$

   which yields to

   $$\begin{aligned} y^+ - x^+ - y + x &= 2^{e_y+1} \cdot 2^p - 2^{e_x} \cdot (m_x + 1) - 2^{e_y} \cdot \left(2^{p+1} - 1\right) + 2^{e_x} \cdot m_x \\ &= 2^{e_y} - 2^{e_x} \\ &\geq 0 \end{aligned}$$

3. $$\begin{aligned} x^+ &= 2^{e_x+1} \cdot 2^p \\ y^+ &= 2^{e_y} \cdot (m_y + 1) \end{aligned}$$

   One checks that

   $$\begin{aligned} y^+ - x^+ - y + x &= 2^{e_y} \cdot (m_y + 1) - 2^{e_x+1} \cdot 2^p - 2^{e_y} \cdot m_y + 2^{e_x} \cdot \left(2^{p+1} - 1\right) \\ &= 2^{e_y} - 2^{e_x} \\ &\geq 0 \end{aligned}$$

4. $$\begin{aligned} x^+ &= 2^{e_x+1} \cdot 2^p \\ y^+ &= 2^{e_y+1} \cdot 2^p \end{aligned}$$

   Thus

   $$\begin{aligned} y^+ - x^+ - y + x &= 2^{e_y+1} \cdot 2^p - 2^{e_x+1} \cdot 2^p - 2^{e_y} \cdot \left(2^{p+1} - 1\right) + 2^{e_x} \cdot \left(2^{p-1} - 1\right) \\ &= 2^{e_y} - 2^{e_x} \\ &\geq 0 \end{aligned}$$

This finishes the proof. ∎

3 A normal form and renormalization procedures

3.1 A proposition for a renormalization sequence

Let us now consider the following algorithm that, at first sight, seems to implement a renormalization of a triple-double number $a_h + a_m + a_l$:

Algorithm 3.1 (A possible renormalization)
In: $a_h, a_m, a_l \in \mathbb{F}$
Out: $r_h, r_m, r_l \in \mathbb{F}$

$$\begin{aligned}(t_{1h}, t_{1l}) &\leftarrow \mathbf{Add12}\,(a_h, a_m)\\ (t_2, r_l) &\leftarrow \mathbf{Add12}\,(t_{1l}, a_l)\\ (r_h, r_m) &\leftarrow \mathbf{Add12}\,(t_{1h}, t_2)\end{aligned}$$

Let us now show that the first idea that algorithm 3.1 returns always a non-overlapping triple-double number is not correct. We will do this by the following theorem.

Theorem 3.2
There exist double precision numbers $a_h, a_m, a_l \in \mathbb{F}$ such that algorithm 3.1 returns a triple-double number $r_h + r_m + r_l$ which is not normalised.

Proof
It suffices to consider the following double precision numbers:

$$\begin{aligned}a_h &= 1.0\\ a_m &= -2^{-54}\\ a_l &= 2^{-64} + 2^{-107}\end{aligned}$$

During the computations by algorithm 3.1, we will observe the following intermediate values:

$$\begin{aligned}t_{1h} &= a_h \oplus a_m\\ &= 1.0 \oplus -2^{-54}\\ &= 1.0\end{aligned}$$

because $1.0 - 2^{-54}$ is at the exact middle of two floating point numbers for an exponent corresponding to 1.0 and because the mantissa of 1.0 finishes by a 0 [2]. Thus,

$$\begin{aligned}t_{1l} &= a_h + a_m - t_{1h}\\ &= 1.0 - 2^{-54} - 1.0\\ &= -2^{-54}\end{aligned}$$

So we get

$$\begin{aligned}t_2 &= t_{1l} \oplus \left(-2^{-64} + 2^{-107}\right)\\ &= -2^{-54} - 2^{-64}\end{aligned}$$

because with a 53 bit mantissa, 2^{-107} is no longer representable with an exponent corresponding to $2^{-54} + 2^{-64}$ and because 2^{-107} is exactly at half of an ulp of $2^{-54} + 2^{-64}$.
So, clearly, we get

$$\begin{aligned}r_l &= t_{1l} + a_l - t_2\\ &= -2^{-54} - 2^{-64} - 2^{-107} + 2^{-54} + 2^{-64}\\ &= -2^{-107}\end{aligned}$$

So finally, we get

$$\begin{aligned} r_h &= t_{1h} \oplus t_2 \\ &= 1.0 \oplus \left(-2^{-54} - 2^{-64}\right) \\ &= 1.0 - 2^{-53} \end{aligned}$$

because $2^{-54} + 2^{-64}$ is greater than a quarter of an ulp of 1.0 and 1.0 is an exact power of 2. In consequence,

$$\begin{aligned} r_m &= t_{1h} + t_2 - r_h \\ &= 1.0 - 2^{-54} - 2^{-64} - 1.0 + 2^{-53} \\ &= 2^{-55} + 2^{-56} + 2^{-57} + 2^{-58} + 2^{-59} + 2^{-60} + 2^{-61} + 2^{-62} + 2^{-63} + 2^{-64} \end{aligned}$$

So the exponent of r_m is -55 which means that its ulp is at 2^{-107}. In contrast, the exponent of r_l is -107 which means that there is one bit of overlap. ∎

3.2 A second renormalization

Let us now analyse a second renormalization algorithm. We will prove its correctness be a series of lemmas and a final theorem.
Let be the following procedure:

Algorithm 3.3 (Renormalization)
In: $a_h, a_m, a_l \in \mathbb{F}$ *verifying the following preconditions:*
Preconditions:

- *None of the numbers* a_h, a_m, a_l *is subnormal*
- a_h *et* a_m *do not overlap in more than* 51 *bits*
- a_m *et* a_l *do not overlap in more than* 51 *bits*

which means formally:

$$\begin{aligned} |a_m| &\leq 2^{-2} \cdot |a_h| \\ |a_l| &\leq 2^{-2} \cdot |a_m| \\ |a_l| &\leq 2^{-4} \cdot |a_h| \end{aligned}$$

Out: $r_h, r_m, r_l \in \mathbb{F}$

$$\begin{aligned} (t_{1h}, t_{1l}) &\leftarrow \textbf{Add12}\,(a_m, a_l) \\ (r_h, t_{2l}) &\leftarrow \textbf{Add12}\,(a_h, t_{1h}) \\ (r_m, r_l) &\leftarrow \textbf{Add12}\,(t_{2l}, t_{1l}) \end{aligned}$$

Consult also [6] on the subject of this algorithm. Let us give now some lemmas on the properties of the values returned by algorithm 3.3 and on the intermediate ones.

Lemma 3.4 (Exact sum)
For each triple-double number $a_h + a_m + a_l$*, algorithm 3.3 returns a triple-double number* $r_h + r_m + r_l$ *such that*

$$a_h + a_m + a_l = r_h + r_m + r_l$$

Proof
This fact is a trivial consequence of the properties of the **Add12** algorithm. ∎

Lemma 3.5 (Rounding of the middle component)
For each triple-double number $a_h+a_m+a_l$, algorithm 3.3 returns a triple-double number $r_h+r_m+r_l$ such that

$$r_m = \circ(r_m + r_l)$$

The same way, the intermediate and final value will verify the following properties:

$$\begin{aligned} t_{1h} &= \circ(a_m + a_l) \\ r_h &= \circ(a_h + t_{1h}) \\ |t_{1l}| &\leq 2^{-53} \cdot |t_{1h}| \\ |t_{2l}| &\leq 2^{-53} \cdot |r_h| \end{aligned}$$

In particular, r_m will not be equal to 0 if r_l is not equal to 0.

Proof
This fact is a trivial consequence of the properties of the **Add12** algorithm. ∎

Lemma 3.6 (Upper bounds)
For all arguments of algorithm 3.3, the intermediate and final values t_{1h}, t_{1l}, t_{2l} et r_m can be bounded upwards as follows:

$$\begin{aligned} |t_{1h}| &\leq 2^{-1} \cdot |a_h| \\ |t_{1l}| &\leq 2^{-54} \cdot |a_h| \\ |t_{2l}| &\leq 2^{-52} \cdot |a_h| \\ |t_m| &\leq 2^{-51} \cdot |a_h| \end{aligned}$$

Proof

1. **Upper bound for $|t_{1h}|$:**
   We have supposed that

   $$\begin{aligned} |a_m| &\leq 2^{-2} \cdot |a_h| \\ |a_l| &\leq 2^{-4} \cdot |a_h| \end{aligned}$$

   So we can check that

   $$\begin{aligned} |t_{1h}| &\leq |a_m| + |a_l| + 2^{-54} \cdot |a_h| \\ &\leq 2^{-2} \cdot |a_h| + 2^{-4} \cdot |a_h| + 2^{-54} \cdot |a_h| \\ &\leq 2^{-1} \cdot |a_h| \end{aligned}$$

2. **Upper bound for $|t_{1l}|$:**
   Using the properties of the **Add12** algorithm we can get to know that

   $$|t_{1l}| \leq 2^{-53} \cdot |t_{1h}|$$

   which yields finally to

   $$|t_{1l}| \leq 2^{-54} \cdot |a_h|$$

3. **Upper bound for $|t_{2l}|$:**

   $$\begin{aligned} |t_{2l}| &\leq 2^{-53} \cdot |r_h| \\ &\leq 2^{-53} \cdot \circ(|a_h| + |t_{1h}|) \\ &\leq 2^{-53} \cdot |a_h| + 2^{-53} \cdot |t_{1h}| + 2^{-106} \cdot |a_h| + 2^{-106} \cdot |t_{1h}| \\ &\leq 2^{-53} \cdot |a_h| + 2^{-54} \cdot |a_h| + 2^{-106} \cdot |a_h| + 2^{-107} \cdot |a_h| \\ &\leq 2^{-52} \cdot |a_h| \end{aligned}$$

4. **Upper bound for $|r_m|$:**

$$\begin{aligned} |r_m| &\leq |t_{2l} \oplus t_{1l}| \\ &\leq |t_{2l}| + |t_{1l}| + 2^{-53} \cdot |t_{2l} + t_{1l}| \\ &\leq 2^{-52} \cdot |a_h| + 2^{-54} \cdot |a_h| + 2^{-105} \cdot |a_h| + 2^{-107} \cdot |a_h| \\ &\leq 2^{-51} \cdot |a_h| \end{aligned}$$

∎

Lemma 3.7 (Special case for $r_h = 0$)
For all arguments verifying the preconditions of algorithm 3.3, r_h will not be equal to 0 if r_m is not equal to 0.
Formally:

$$r_h = 0 \Rightarrow r_m = 0$$

Proof
Let us suppose that $r_h = 0$ and that $r_m \neq 0$. So we get

$$\begin{aligned} |r_h| &= |\circ (a_h + t_{1h})| \\ &\geq \left|\circ \left(2^{-1} \cdot a_h\right)\right| \\ &= 2^{-1} |a_h| \end{aligned}$$

because we have already shown that

$$|t_{1h}| \leq 2^{-1} \cdot |a_h|$$

So for r_h being equal to 0, a_h must be equal to 0.
In contrast, this yields to $t_{1h} = 0$ because

$$r_h = \circ (0 + t_{1h}) = t_{1h}$$

This implies that $t_{1l} = 0$ because of the properties of the **Add12** procedure. The same way, we get $t_{2l} = 0$.
We can deduce from this that

$$0 \neq r_m = \circ (0 + 0) = 0$$

which is a contradiction. ∎

Lemma 3.8 (Lower bound for $|r_h|$)
For all arguments verifying the preconditions of algorithm 3.3, the final result r_h can be bounded in magnitude as follows:

$$|r_h| \geq 2^{-1} \cdot |a_h|$$

Proof
We have that

$$r_h = a_h \oplus t_{1h}$$

Clearly, if a_h and t_{1h} have the same sign, we get

$$|r_h| \geq |a_h| \geq 2^{-1} \cdot |a_h|$$

Otherwise - a_h and t_{1h} are now of the opposed sign - we have already seen that

$$|t_{1h}| \leq 2^{-1} \cdot |a_h|$$

So, in this case, too, we get

$$|r_h| \geq 2^{-1} \cdot |a_h|$$

∎

Corollary 3.9 (Additional property on the Add12 procedure)
Let be r_h and r_l two double precision floating point numbers returned by the **Add12** *procedure. So*

$$|r_l| \leq \frac{1}{2} \cdot \mathrm{ulp}(r_h)$$

Proof
Using lemma 2.7 and the definition 2.3 of the **Add12** procedure. ∎

Theorem 3.10 (Correctness of the renormalization algorithm 3.3)
For all arguments verifying the preconditions of procedure 3.3, the values returned r_h, r_m and r_l will not overlap unless they are all equal to 0 and their sum will be exactly the sum of the values in argument a_h, a_m et a_l.

Proof
The fact that the sum of the values returned is exactly equal to the sum of the values in argument has already been proven by lemma 3.4.
Without loss of generality, we will now suppose that neither r_h nor r_m will be 0 in which case all values returned would be equal to 0 as we have shown it by lemmas 3.7 and 3.5.
Using lemma 3.5, we know already that r_m and r_l do not overlap. Let us show now that r_h and r_m do not overlap by proving that the following inequality is true

$$|r_m| \leq \frac{3}{4} \cdot \mathrm{ulp}(r_h) < \mathrm{ulp}(r_h)$$

We will than use lemma 2.7 for concluding.
There are two different cases to be treated.

1st case: $t_{2l} = 0$
We know that

$$r_m = \circ(t_{2l} + t_{1l}) = \circ(0 + t_{1l}) = t_{1l}$$

When showing lemma 3.6, we have already proven that

$$|t_{1l}| \leq 2^{-54} \cdot |a_h|$$

Using lemma 3.8, we therefore know that

$$|r_m| \leq 2^{-53} \cdot |r_h|$$

which is the result we wanted to prove.

2nd case: $t_{2l} \neq 0$
Still using lemma 3.6, we have shown that

$$|t_{1h}| \leq 2^{-1} \cdot |a_h|$$

In consequence, when the IEEE 754 [2] addition $r_h = a_h \oplus t_{1h}$ is ported out, the rounding will be done at a bit of weight heigher than one $\mathrm{ulp}(t_{1h})$ because t_{2l} is strictly greater than 0 and because a_h and t_{1h} do not completely overlap. Therefore we can check that

$$|t_{2l}| \geq \mathrm{ulp}(t_{1h})$$

With the result of lemma 3.6 we already mentioned, we can deduce that

$$|t_{2l}| \leq \frac{1}{2} \cdot \mathrm{ulp}(t_{1h}) \leq \frac{1}{2} \cdot |t_{2l}|$$

So one can verify the following upper bound using among others lemma 3.9:

$$\begin{aligned} |r_m| &= |\circ(t_{2l} + t_{1l})| \\ &\leq \circ\left(\frac{3}{2} \cdot |t_{2l}|\right) \\ &\leq \circ\left(\frac{3}{4} \cdot \mathrm{ulp}(r_h)\right) \\ &= \frac{3}{4} \cdot \mathrm{ulp}(r_h) \end{aligned}$$

One remarks that the last simplification is correct here because ulp is always equal to an integer power of 2 and because the precision of a double is greater than 4 bits. ∎

4 Operators on double-double numbers

Since we dispose now of a renormalization procedure which is effective and proven, we can now consider the different addition and multiplication operators we need. They will surely work finally on expansions of size 3, but the double-double format [5] must be analysed, too, because it is at the base of the triple-double format. We already mentioned that on definition and analysis of this operators, we need not care such a lot of the overlap in the components of a triple-double number any more: as long as the overlap does not make us loose a too much of the final accuracy because several bits of the "mantissa" are represented twice, overlap is not of an issue for intermediate values. At the end of triple-double computations, it will be sufficient to apply once the renormalization procedure. In order to measure the consequences of an overlap in the operands on final accuracy and in order to be able to follow the increase of the overlap during computations in triple-double, we will indicate for each operator which produces a triple double result or which takes a triple-double operand not only a bound for relative and absolute rounding errors but also a bound for the maximal overlap of the values returned. All this bounds will be parameterised by a variable representing the maximal overlap of the triple-double arguments.

4.1 The addition operator Add22

Let us analyse first the following addition procedure:

Algorithm 4.1 (Add22)
In: *two double-double numbers,* $a_h + a_l$ *et* $b_h + b_l$
Out: *a double-double number* $r_h + r_l$
Preconditions on the arguments:

$$|a_l| \leq 2^{-53} \cdot |a_h|$$

$$|b_l| \leq 2^{-53} \cdot |b_h|$$

Algorithm:

$t_1 \leftarrow a_h \oplus b_h$
if $|a_h| \geq |b_h|$ **then**

$t_2 \leftarrow a_h \ominus t_1$
$t_3 \leftarrow t_2 \oplus b_h$
$t_4 \leftarrow t_3 \oplus b_l$
$t_5 \leftarrow t_4 \oplus a_l$

else

$t_2 \leftarrow b_h \ominus t_1$
$t_3 \leftarrow t_2 \oplus a_h$
$t_4 \leftarrow t_3 \oplus a_l$
$t_5 \leftarrow t_4 \oplus b_l$

end if
$(r_h, r_l) \leftarrow$ **Add12** (t_1, t_5)

Compare [1] concerning algorithm 4.1.

Theorem 4.2 (Relative error of algorithm 4.1 Add22 without occurring of cancellation)

Let be $a_h + a_l$ and $b_h + b_l$ the double-double arguments of algorithm 4.1 **Add22**.
If a_h and b_h have the same sign, so we know that

$$r_h + r_l = ((a_h + a_l) + (b_h + b_l)) \cdot (1 + \varepsilon)$$

where ε is bounded as follows:

$$|\varepsilon| \leq 2^{-103,5}$$

Proof
Since the algorithm **Add22** ends by a call to the **Add12** procedure, it suffices to show that

$$t_1 + t_5 = ((a_h + a_l) + (b_h + b_l)) \cdot (1 + \varepsilon)$$

Further, since the two branches of the algorithm are symmetrical, we can suppose that $|a_h| \geq |b_h|$ and consider only one branch without loss of generality. Finally, we remark that the following lines of the **Add22** procedure constitute a non-conditional **Add12** with arguments a_h and b_h and the result $t_1 + t_3$:

$$\begin{aligned} t_1 &= a_h \oplus b_h \\ t_2 &= a_h \ominus t_1 \\ t_3 &= t_2 \oplus b_h \end{aligned}$$

Thus, we get

$$\begin{aligned} t_5 &= t_4 \oplus a_l \\ &= (t_4 + a_l) \cdot (1 + \varepsilon_1) \\ &= ((t_3 + bl) \cdot (1 + \varepsilon_2) + a_l) \cdot (1 + \varepsilon_1) \\ &= t_3 + a_l + b_l + \delta \end{aligned}$$

with

$$\delta = t_3 \cdot \varepsilon_2 + b_l \cdot \varepsilon_2 + t_3 \cdot \varepsilon_1 + b_l \cdot \varepsilon_1 + t_3 \cdot \varepsilon_2 \cdot \varepsilon_2 + b_l \cdot \varepsilon_2 \cdot \varepsilon_2 + a_l \cdot \varepsilon_1$$

For giving an upper bound for $|\delta|$, let us first give an upper bound for $|t_3|$, $|a_l|$ and b_l as function of $|a_h + b_h|$ using the following bounds that we know already:

$$\begin{aligned} |a_l| &\leq 2^{-53} \cdot |a_h| \\ |b_l| &\leq 2^{-53} \cdot |b_h| \\ |t_3| &\leq 2^{-53} \cdot |t_1| \end{aligned}$$

We get therefore

$$\begin{aligned} |t_3| &\leq 2^{-53} \cdot |t_1| \\ &= 2^{-53} \cdot |a_h \oplus b_h| \\ &\leq 2^{-53} \cdot |a_h + b_h| + 2^{-106} \cdot |a_h + b_l| \end{aligned}$$

and than

$$\begin{aligned} |a_l| &\leq 2^{-53} \cdot |a_h| \\ &\leq 2^{-53} \cdot |a_h + b_h| \end{aligned}$$

The last bound is verified because we suppose that a_h and b_h have the same sign.
Finally, since $|a_h| \geq |b_h|$,

$$\begin{aligned} |b_l| &\leq 2^{-53} \cdot |b_h| \\ &\leq 2^{-53} \cdot |a_h| \\ &\leq 2^{-53} \cdot |a_h + b_h| \end{aligned}$$

Thus we get for $|\delta|$:

$$\begin{aligned} |\delta| &\leq |a_h + b_h| \cdot \left(2^{-106} + 2^{-159} + 2^{-106} + 2^{-106} + 2^{-159} + 2^{-106} + 2^{-159} + 2^{-212} + 2^{-212} + 2^{-106}\right) \\ &\leq |a_h + b_h| \cdot \left(2^{-104} + 2^{-106} + 2^{-158} + 2^{-159} + 2^{-211}\right) \end{aligned}$$

Let us now give a lower bound for $|a_h + a_l + b_h + b_l|$ as a function of $|a_h + b_h|$ in order to be able to give a relative error bound for the procedure **Add22**. We have that

$$\begin{aligned} |a_l + b_l| &\leq |a_l| + |b_l| \\ &\leq 2^{-53} \cdot |a_h| + 2^{-53} \cdot |b_h| \\ &\leq 2^{-52} \cdot |a_h| \\ &\leq 2^{-52} \cdot |a_h + b_h| \end{aligned}$$

So we can check that

$$|a_h + a_l + b_h + b_l| \geq \left(1 - 2^{-52}\right) \cdot |a_h + b_h|$$

Concerning $|\delta|$, this yields to

$$|\delta| \leq |a_h + a_l + b_h + b_l| \cdot \frac{1}{1 - 2^{-52}} \cdot \left(2^{-104} + 2^{-106} + 2^{-158} + 2^{-159} + 2^{-211}\right)$$

One easily checks that

$$\frac{1}{1 - 2^{-52}} \cdot \left(2^{-104} + 2^{-106} + 2^{-158} + 2^{-159} + 2^{-211}\right) \leq 2^{-103,5}$$

from which one trivially deduces the affirmation. ∎

Theorem 4.3 (Relative error of algorithm 4.1 Add22 with a bounded cancellation)
Let be $a_h + a_l$ and $b_h + b_l$ the double-double arguments of 4.1 **Add22**.
If a_h and b_h are of different sign and if one can check that

$$|b_h| \leq 2^{-\mu} \cdot |a_h|$$

for $\mu \geq 1$
so the returned result will verify

$$r_h + r_l = ((a_h + a_l) + (b_h + b_l)) \cdot (1 + \varepsilon)$$

where ε is bounded as follows:

$$|\varepsilon| \leq 2^{-103} \cdot \frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu} - 2^{-52}} \leq 2^{-102}$$

Proof
Let us reuse the results obtained at the proof 4.1 and let us start by giving an upper bound for $|b_l|$ as a function of $|a_h|$:

$$|b_l| \leq 2^{-53} \cdot |b_h| \leq 2^{-53-\mu} \cdot |a_h|$$

Let us now continue with a lower bound for $|a_h + b_h|$ still as a function of $|a_h|$:

$$|a_h + b_h| \geq |a_h| \cdot \left(1 - 2^{-\mu}\right)$$

We get in consequence

$$|a_l| \leq \frac{2^{-53}}{1 - 2^{-\mu}} \cdot |a_h + b_h|$$

and

$$|b_l| \leq \frac{2^{-53-\mu}}{1 - 2^{-\mu}} \cdot |a_h + b_h|$$

Thus we can check that

$$|\delta| \leq |a_h + b_h| \cdot 2^{-103} \cdot \frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu}}$$

Once again, we must give a lower bound for $|a_h + a_l + b_h + b_l|$ with regard to $|a_h + b_h|$: We know that

$$\begin{aligned} |a_l + b_l| &\leq |a_l| + |b_l| \\ &\leq 2^{-52} \cdot |a_h| \\ &\leq \frac{2^{-52}}{1 - 2^{-\mu}} \cdot |a_h + b_h| \end{aligned}$$

So

$$|a_h + a_l + b_h + b_l| \geq |a_h + b_h| \cdot \frac{1 - 2^{-\mu} - 2^{-52}}{1 - 2^{-\mu}}$$

Thus we get for $|\delta|$

$$\begin{aligned} |\delta| &\leq |a_h + a_l + b_h + b_l| \cdot \frac{1 - 2^{-\mu}}{1 - 2^{-\mu} - 2^{-52}} \cdot 2^{-103} \cdot \frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu}} \\ &= |a_h + a_l + b_h + b_l| \cdot 2^{-103} \cdot \frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu} - 2^{-52}} \end{aligned}$$

So finally the following inequality is verified for the relative error ε:

$$|\varepsilon| \leq 2^{-103} \cdot \frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu} - 2^{-52}}$$

We can still give a less exact upper bound for this term by one that does not depend on μ because $\mu \geq 1$:

$$\frac{1 - 2^{-\mu-1}}{1 - 2^{-\mu} - 2^{-52}} \leq \frac{\frac{3}{4}}{\frac{1}{2} - 2^{-52}} \leq 2$$

so

$$|\varepsilon| \leq 2^{-102}$$

∎

Theorem 4.4 (Absolute error of algorithm 4.1 Add22 (general case))
Let be $a_h + a_l$ and $b_h + b_l$ the double-double arguments of algorithm 4.1 **Add22**.
The result $r_h + r_l$ returned by the algorithm verifies

$$r_h + r_l = (a_h + a_l) + (b_h + b_l) + \delta$$

where δ is bounded as follows:

$$|\delta| \leq \max\left(2^{-53} \cdot |a_l + b_l|, 2^{-102} \cdot |a_h + a_l + b_h + b_l|\right)$$

Proof
Without loss of generality, we can now suppose that

$$\frac{1}{2} \cdot |a_h| \leq |b_h| \leq |a_h|$$

and that a_h and b_h have different signs because for all other cases, the properties we have to show are a direct consequence of theorems 4.2 and 4.3.
So we have

$$\frac{1}{2} \cdot |a_h| \leq |b_h| \leq 2 \cdot |a_h|$$

and

$$\frac{1}{2} \cdot |b_h| \leq |a_h| \leq 2 \cdot |b_h|$$

So the floating point operation

$$t_1 = a_h \oplus b_h$$

is exact by Sterbenz' lemma [11]. In consequence, t_3 will be equal to 0 because, as we have already mentioned, the operations computing t_1 and t_3 out of a_h and b_h constitute a **Add12** whose properties assure that

$$t_3 = a_h + b_h - t_1$$

We can deduce that

$$t_4 = t_3 \oplus b_l = b_l$$

and can finally check that

$$t_5 = t_4 \oplus a_l = (t_4 + a_l) \cdot (1 + \varepsilon^*)$$

with

$$|\varepsilon^*| \leq 2^{-53}$$

So we get

$$r_h + r_l = t_1 + t_5 = (a_h + a_l + b_h + b_l) + \delta$$

with

$$|\delta| \leq 2^{-53} \cdot |a_l + b_l|$$

which yields to the bound to be proven

$$|\delta| = \max\left(2^{-53} \cdot |a_l + b_l|, 2^{-102} \cdot |a_h + a_l + b_h + b_l|\right)$$

∎

Theorem 4.5 (Output overlap of algorithm 4.1 Add22)
Let be $a_h + a_l$ and $b_h + b_l$ the double-double arguments of algorithm 4.1 **Add22**.
So the values r_h and r_l returned by the algorithm will not overlap at all and will verify

$$|r_l| \leq 2^{-53} \cdot |r_h|$$

Proof
The proof of the affirmed property is trivial because the procedure **Add22** ends by a call to sequence **Add12** which assures it. ∎

4.2 The multiplication operator Mul22

Let us now consider the multiplication operator **Mul22**:

Algorithm 4.6 (Mul22)
In: *two double-double numbers,* $a_h + a_l$ *et* $b_h + b_l$
Out: *a double-double number* $r_h + r_l$
Preconditions on the arguments:

$$|a_l| \leq 2^{-53} \cdot |a_h|$$
$$|b_l| \leq 2^{-53} \cdot |b_h|$$

Algorithm:

$$\begin{array}{l}(t_1, t_2) \leftarrow \textbf{Mul12}\,(a_h, b_h)\\ t_3 \leftarrow a_h \otimes b_l\\ t_4 \leftarrow a_l \otimes b_h\\ t_5 \leftarrow t_3 \oplus t_4\\ t_6 \leftarrow t_2 \oplus t_5\\ (r_h, r_l) \leftarrow \textbf{Add12}\,(t_1, t_6)\end{array}$$

Compare also to [1] concerning algorithm 4.6.

Theorem 4.7 (Relative error of algorithm 4.6 Mul22)
Let be $a_h + a_l$ *and* $b_h + b_l$ *the double-double arguments of algorithm 4.6* **Mul22**.
So the values returned r_h *et* r_l *verify*

$$r_h + r_l = ((a_h + a_l) \cdot (b_h + b_l)) \cdot (1 + \varepsilon)$$

where ε *is bounded as follows:*

$$|\varepsilon| \leq 2^{-102}$$

Further r_h *and* r_l *will not overlap at all and verify*

$$|r_l| \leq 2^{-53} \cdot |r_h|$$

Proof
Since algorithm 4.6 ends by a call to the **Add12** procedure, the properties of the latter yield to the fact that r_h and r_l do not overlap at all and that $|r_l| \leq 2^{-53} \cdot |r_h|$.
In order to give upper bounds for the relative and absolute error of the algorithm, let us express t_6 as a function of t_2, a_h, a_l, b_h and b_l joined by the error term δ.
We get

$$\begin{array}{rcl} t_6 & = & t_2 \oplus (a_h \otimes b_l \oplus a_l \otimes b_h)\\ & = & (t_2 + (a_h \cdot b_l \cdot (1+\varepsilon_1) + a_l \cdot b_h \cdot (1+\varepsilon_2)) \cdot (1+\varepsilon_3)) \cdot (1+\varepsilon_4)\end{array}$$

where $|\varepsilon_i| \leq 2^{-53}$, $i = 1, 2, 3, 4$.
Simplifying this expression, we van verify that

$$t_6 = t_2 + a_h \cdot b_l + a_l \cdot b_h + \delta$$

with

$$\begin{array}{rcl} |\delta| & \leq & |a_l \cdot b_l + a_h \cdot b_l \cdot \varepsilon_1 + a_l \cdot b_h \cdot \varepsilon_2 + a_h \cdot b_l \cdot \varepsilon_3 + a_h \cdot b_l \cdot \varepsilon_1 \cdot \varepsilon_3 + a_l \cdot b_h \cdot \varepsilon_3 + a_l \cdot b_h \cdot \varepsilon_1 \cdot \varepsilon_3\\ & & + a_h \cdot b_l \cdot \varepsilon_4 + a_l \cdot b_h \cdot \varepsilon_4 + a_h \cdot b_l \cdot \varepsilon_1 \cdot \varepsilon_4 + a_l \cdot b_h \cdot \varepsilon_2 \cdot \varepsilon_4 + a_h \cdot b_l \cdot \varepsilon_3 \cdot \varepsilon_4\\ & & + a_h \cdot b_l \cdot \varepsilon_1 \cdot \varepsilon_3 \cdot \varepsilon_4 + a_l \cdot b_h \cdot \varepsilon_3 \cdot \varepsilon_4 + a_l \cdot b_h \cdot \varepsilon_2 \cdot \varepsilon_3 \cdot \varepsilon_4|\\ & \leq & |a_h \cdot b_h| \cdot \left(7 \cdot 2^{-106} + 6 \cdot 2^{-159} + 2^{-211}\right)\\ & \leq & |a_h \cdot b_h| \cdot 2^{-103}\end{array}$$

For checking the given bound, we have supposed the following inequalities:

$$|a_l| \leq 2^{-53} \cdot |a_h|$$

and

$$|a_l| \leq 2^{-53} \cdot |a_h|$$

Let us now give a lower bound for $|(a_h + a_l) \cdot (b_h + b_l)|$ as a function of $|a_h \cdot b_h|$. For doing so, we give an upper bound for $|a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l|$.
We verify since:

$$\begin{aligned} |a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l| &\leq |a_h \cdot b_l| + |a_l \cdot b_h| + |a_l \cdot b_l| \\ &\leq 2^{-53} \cdot |a_h \cdot b_h| + 2^{-53} \cdot |a_h \cdot b_h| + 2^{-106} \cdot |a_h \cdot b_h| \\ &\leq 2^{-51} \cdot |a_h \cdot b_h| \end{aligned}$$

This yields to

$$\begin{aligned} |(a_h + a_l) \cdot (b_h + b_l)| &\geq |a_h \cdot b_h| \cdot \left(1 - 2^{-51}\right) \\ &\geq \frac{1}{2} \cdot |a_h \cdot b_h| \end{aligned}$$

from which we deduce that

$$|\delta| \leq 2^{-102} \cdot |(a_h + a_l) \cdot (b_h + b_l)|$$

which gives us as an bound for the relative error

$$r_h + r_l = (a_h + a_l) \cdot (b_h + b_l) \cdot (1 + \varepsilon)$$

with $|\varepsilon| \leq 2^{-102}$. ∎

5 Addition operators for triple-double numbers

5.1 The addition operator Add33

We are going to consider now the addition operator **Add33**. We will only analyse a simplified case where the arguments' values verify some bounds statically known.

Algorithm 5.1 (Add33)
In: *two triple-double numbers,* $a_h + a_m + a_l$ *et* $b_h + b_m + b_l$
Out: *a triple-double number* $r_h + r_m + r_l$
Preconditions on the arguments:

$$\begin{aligned} |b_h| &\leq \frac{3}{4} \cdot |a_h| \\ |a_m| &\leq 2^{-\alpha_o} \cdot |a_h| \\ |a_l| &\leq 2^{-\alpha_u} \cdot |a_m| \\ |b_m| &\leq 2^{-\beta_o} \cdot |b_h| \\ |b_l| &\leq 2^{-\beta_u} \cdot |b_m| \\ \alpha_o &\geq 4 \\ \alpha_u &\geq 1 \\ \beta_o &\geq 4 \\ \beta_u &\geq 1 \end{aligned}$$

Algorithm:

$$
\begin{aligned}
(r_h, t_1) &\leftarrow \mathbf{Add12}\,(a_h, b_h)\\
(t_2, t_3) &\leftarrow \mathbf{Add12}\,(a_m, b_m)\\
(t_7, t_4) &\leftarrow \mathbf{Add12}\,(t_1, t_2)\\
t_6 &\leftarrow a_l \oplus b_l\\
t_5 &\leftarrow t_3 \oplus t_4\\
t_8 &\leftarrow t_5 \oplus t_6\\
(r_m, r_l) &\leftarrow \mathbf{Add12}\,(t_7, t_8)
\end{aligned}
$$

Theorem 5.2 (Relative error of algorithm 5.1 Add33)
Let be $a_h + a_m + a_l$ and $b_h + b_m + b_l$ the triple-double arguments of algorithm 5.1 **Add33** *verifying the given preconditions.*
So the following egality will hold for the returned values r_h, r_m et r_l

$$r_h + r_m + r_l = ((a_h + a_m + a_l) + (b_h + b_m + b_l)) \cdot (1 + \varepsilon)$$

where ε is bounded by:

$$|\varepsilon| \leq 2^{-\min(\alpha_o+\alpha_u, \beta_o+\beta_u)-47} + 2^{-\min(\alpha_o, \beta_o)-98}$$

The returned values r_m and r_l will not overlap at all and the overlap of r_h and r_m will be bounded by the following expression:

$$|r_m| \leq 2^{-\min(\alpha_o, \beta_o)+5} \cdot |r_h|$$

Proof
The procedure 5.1 ends by a call to the **Add12** sequence. One can trivially deduce that r_m and r_l do not overlap at all and verify

$$|r_l| \leq 2^{-53} \cdot |r_m|$$

Further, it suffices that the bounds given at theorem 5.2 hold for t_7 and t_8 because the last addition computing r_m et r_l will be exact. The same way, one can deduce the following inequalities out of the properties of the **Add12** procedure. They will become useful during this proof.

$$|t_1| \leq 2^{-53} \cdot |r_h|$$

$$|t_3| \leq 2^{-53} \cdot |t_2|$$

$$|t_4| \leq 2^{-53} \cdot |t_7|$$

Let us start the proof by giving bounds for the magnitude of r_h with regard to a_h:
We have on the one hand

$$
\begin{aligned}
|r_h| &= |\circ(a_h + b_h)|\\
&= \circ(|a_h + b_h|)\\
&\leq \circ(|a_h| + |b_h|)\\
&\leq \circ\left(|a_h| + \frac{3}{4} \cdot |a_h|\right)\\
&\leq \circ(2 \cdot |a_h|)\\
&= 2 \cdot |a_h|
\end{aligned}
$$

and on the other

$$
\begin{aligned}
|r_h| &= \circ(|a_h + b_h|)\\
&\geq \circ\left(\frac{1}{4} \cdot |a_h|\right)\\
&= \frac{1}{4} \cdot |a_h|
\end{aligned}
$$

So we know that $\frac{1}{4} \cdot |a_h| \leq |r_h| \leq 2 \cdot |a_h|$.
It is now possible to give the following upper bounds for $|t_1|$, $|t_2|$, $|t_3|$, $|t_7|$, $|t_4|$, $|t_6|$ and $|t_6|$:

$$\begin{aligned} |t_1| &\leq 2^{-53} \cdot |r_h| \\ &\leq 2^{-53} \cdot 2^2 \cdot |a_h| \\ &= 2^{-51} \cdot |a_h| \end{aligned}$$

$$\begin{aligned} |t_2| &\leq \circ(|a_m + b_m|) \\ &\leq \circ(|a_m| + |b_m|) \\ &\leq \circ\left(2^{-\alpha_o} \cdot |a_h| + 2^{-\beta_o} \cdot |b_h|\right) \\ &\leq \circ\left(2^{-\alpha_o} \cdot |a_h| + 2^{-\beta_o} \cdot \frac{3}{4} \cdot |a_h|\right) \\ &\leq \circ\left(2^{-\min(\alpha_o,\beta_o)+1} \cdot |a_h|\right) \\ &= 2^{-\min(\alpha_o,\beta_o)+1} \cdot |a_h| \end{aligned}$$

$$\begin{aligned} |t_3| &\leq 2^{-53} \cdot |t_2| \\ &\leq 2^{-53} \cdot 2^{-\min(\alpha_o,\beta_o)+1} \cdot |a_h| \\ &= 2^{-\min(\alpha_o,\beta_o)-52} \cdot |a_h| \end{aligned}$$

$$\begin{aligned} |t_7| &\leq \circ(|t_1 + t_2|) \\ &\leq \circ(|t_1| + |t_2|) \\ &\leq \circ\left(2^{-51} \cdot |a_h| + 2^{-\min(\alpha_o,\beta_o)+1} \cdot |a_h|\right) \\ &\leq \circ\left(2^{-\min(\alpha_o,\beta_o)+2} \cdot |a_h|\right) \\ &= 2^{-\min(\alpha_o,\beta_o)+2} \cdot |a_h| \end{aligned}$$

$$\begin{aligned} |t_4| &\leq 2^{-53} \cdot |t_7| \\ &\leq 2^{-53} \cdot 2^{-\min(\alpha_o,\beta_o)+2} \cdot |a_h| \\ &= 2^{-\min(\alpha_o,\beta_o)-51} \cdot |a_h| \end{aligned}$$

$$\begin{aligned} |t_6| &\leq \circ(|a_l| + |b_l|) \\ &\leq \circ\left(2^{-\alpha_o} \cdot 2^{-\alpha_u} \cdot |a_h| + 2^{-\beta_o} \cdot 2^{-\beta_u} \cdot \frac{3}{4} \cdot |a_h|\right) \\ &\leq \circ\left(2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)+1} \cdot |a_h|\right) \\ &= 2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)+1} \cdot |a_h| \end{aligned}$$

and finally

$$\begin{aligned} |t_5| &\leq \circ(|t_3| + |t_4|) \\ &\leq \circ\left(2^{-\min(\alpha_o,\beta_o)-52} \cdot |a_h| + 2^{-\min(\alpha_o,\beta_o)-51} \cdot |a_h|\right) \\ &\leq \circ\left(2^{-\min(\alpha_o,\beta_o)-50} \cdot |a_h|\right) \\ &= 2^{-\min(\alpha_o,\beta_o)-50} \cdot |a_h| \end{aligned}$$

Using the fact that the addition **Add12** is exact, it is easy to show that we have exactly

$$r_h + t_7 + t_3 + t_4 = a_h + a_m + b_h + b_m$$

Further, we can check

$$\begin{aligned} t_8 &= (t_3 \oplus_2 t_4) \oplus_1 (a_l \oplus_3 b_l) \\ &= t_3 + t_4 + a_l + b_l + \delta \end{aligned}$$

with

$$\begin{aligned} |\delta| \leq\ & |t_3| \cdot \varepsilon_3 + |t_4| \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_3 + |b_l| \cdot \varepsilon_3 + |t_3| \cdot \varepsilon_1 + |t_4| \cdot \varepsilon_1 + |t_3| \cdot \varepsilon_1 \cdot \varepsilon_2 \\ & + |t_4| \cdot \varepsilon_1 \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_1 + |b_l| \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_1 \cdot \varepsilon_3 + |b_l| \cdot \varepsilon_1 \cdot \varepsilon_3 \end{aligned}$$

where for $i \in \{1, 2, 3\}$, ε_i is the relative error bound of the floating point addition $\oplus_i$ and verifies

$$|\varepsilon_i| \leq 2^{-53}$$

So we get immediately

$$r_h + r_m + r_l = r_h + t_7 + t_8 = (a_h + a_m + a_l) + (b_h + b_m + b_l) + \delta$$

Let us now express $|(a_h + a_m + a_l) + (b_h + b_m + b_l)|$ as a function of $|a_h|$:

$$\begin{aligned} |a_h + a_m + a_l| &\leq |a_h| + |a_m| + |a_l| \\ &\leq |a_h| + 2^{-\alpha_o} \cdot |a_h| + 2^{-\alpha_o - \alpha_u} \cdot |a_h| \\ &\leq 2 \cdot |a_h| \end{aligned}$$

and, the same way round,

$$\begin{aligned} |b_h + b_m + b_l| &\leq 2 \cdot |b_h| \\ &\leq \frac{3}{2} \cdot |a_h| \end{aligned}$$

which allows for noting

$$|(a_h + a_m + a_l) + (b_h + b_m + b_l)| \leq 2^2 \cdot |a_h|$$

In order to give a lower bound for this term, let us prove an upper bound for $|b_h + a_m + b_m + a_l + b_l|$ as follows

$$\begin{aligned} |b_h + a_m + b_m + a_l + b_l| &\leq |b_h| + |a_m| + |b_m| + |a_l| + |b_l| \\ &\leq \frac{3}{4} \cdot |a_h| + 2^{-\alpha_o} \cdot |a_h| + 2^{-\beta_o} \cdot \frac{3}{4} \cdot |a_h| + 2^{-\alpha_o - \alpha_u} \cdot |a_h| \\ &\quad + 2^{-\beta_o - \beta_u} \cdot \frac{3}{4} \cdot |a_h| \\ &\leq \frac{7}{8} \cdot |a_h| \end{aligned}$$

So we get

$$|(a_h + a_m + a_l) + (b_h + b_m + b_l)| \geq \frac{1}{8} \cdot |a_h|$$

Using this bounds, we can give upper bounds for the absolute error $|\delta|$ first as a function of $|a_h|$ and than as a function of $|(a_h + a_m + a_l) + (b_h + b_m + b_l)|$ for deducing finally a bound for the relative error.
So we get

$$\begin{aligned} |\delta| \leq\ & |t_3| \cdot \varepsilon_3 + |t_4| \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_3 + |b_l| \cdot \varepsilon_3 + |t_3| \cdot \varepsilon_1 + |t_4| \cdot \varepsilon_1 + |t_3| \cdot \varepsilon_1 \cdot \varepsilon_2 \\ & + |t_4| \cdot \varepsilon_1 \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_1 + |b_l| \cdot \varepsilon_2 + |a_l| \cdot \varepsilon_1 \cdot \varepsilon_3 + |b_l| \cdot \varepsilon_1 \cdot \varepsilon_3 \\ \leq\ & |a_h| \cdot \varepsilon' \end{aligned}$$

with

$$\begin{aligned}
\varepsilon' &\leq 2^{-53}\cdot 2^{-\min(\alpha_o,\beta_o)-52}\\
&+ 2^{-53}\cdot 2^{-\min(\alpha_o,\beta_o)-51}\\
&+ 2^{-53}\cdot 2^{-\alpha_o-\alpha_u}\\
&+ 2^{-53}\cdot 2^{-\beta_o-\beta_u}\\
&+ 2^{-53}\cdot 2^{-\min(\alpha_o,\beta_o)-52}\\
&+ 2^{-53}\cdot 2^{-\min(\alpha_o,\beta_o)-51}\\
&+ 2^{-106}\cdot 2^{-\min(\alpha_o,\beta_o)-52}\\
&+ 2^{-106}\cdot 2^{-\min(\alpha_o,\beta_o)-51}\\
&+ 2^{-53}\cdot 2^{-\alpha_o-\alpha_u}\\
&+ 2^{-53}\cdot 2^{-\beta_o-\beta_u}\\
&+ 2^{-106}\cdot 2^{-\alpha_o-\alpha_u}\\
&+ 2^{-106}\cdot 2^{-\beta_o-\beta_u}\\
&\leq 2^{-\min(\alpha_o,\beta_o)-101}+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)-50}
\end{aligned}$$

This yields to

$$r_h+r_m+r_l=((a_h+a_m+a_l)+(b_h+b_m+b_l))\cdot(1+\varepsilon)$$

with

$$|\varepsilon|\leq 2^{-\min(\alpha_o,\beta_o)-98}+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)-47}$$

In order to finish the prove, it suffices now to give an upper bound for the maximal overlap between r_h and r_m because we have already shown that r_m and r_l do not overlap at all.
So we can check

$$\begin{aligned}
|r_8| &\leq \circ(|t_5|+|t_6|)\\
&\leq \circ\left(2^{-\min(\alpha_o,\beta_o)-50}\cdot|a_h|+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)+1}\cdot|a_h|\right)\\
&\leq \circ\left(2^{-\min(\alpha_o,\beta_o)-48}\cdot|r_h|+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)+3}\cdot|r_h|\right)
\end{aligned}$$

and continue by giving the following upper bound

$$\begin{aligned}
|r_m| &= \circ(|t_7+t_8|)\\
&\leq \circ(|t_7|+|t_8|)\\
&\leq \circ\left(2^{-\min(\alpha_o,\beta_o)+4}\cdot|r_h|+\circ\left(2^{-\min(\alpha_o,\beta_o)-48}\cdot|r_h|+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)+3}\cdot|r_h|\right)\right)\\
&\leq \circ\Big(|r_h|\cdot\Big(2^{-\min(\alpha_o,\beta_u)+4}+2^{-\min(\alpha_o,\beta_o)-48}+2^{-\min(\alpha+\alpha_u,\beta_o+\beta_u)+3}+\\
&\quad 2^{-\min(\alpha_o,\beta_o)-101}+2^{-\min(\alpha_o+\alpha_u,\beta_o+\beta_u)-50}\Big)\Big)\\
&\leq 2^{-\min(\alpha_o,\beta_o)+5}\cdot|r_h|
\end{aligned}$$

This is the maximal overlap bound we were looking for; the proof is therefore finished. ∎

Theorem 5.3 (Special case of algorithm 5.1 Add33)
Let be $a_h+a_m+a_l$ and $b_h+b_m+b_l$ the triple-double arguments of algorithm 5.1 **Add33** *such that*

$$a_h=a_m=a_l=0$$

So the values r_h, r_m and r_l returned will be exactly equal to

$$r_h+r_m+r_l=b_h+b_m+b_l$$

The values r_m and r_l will not overlap at all. The overlap of r_h and r_m must still be evaluated.

Proof
We will suppose that the **Add12** procedure is exact for $a_h = a_m = a_l = 0$ if even we are using its unconditional version. Under this hypothesis, we get thus:

$$\begin{aligned} r_h &= \circ(0 + b_h) = b_h \\ t_1 &= 0 + b_h - b_h = 0 \\ t_2 &= b_m \\ t_3 &= 0 \\ t_7 &= \circ(0 + b_m) = b_m \\ t_4 &= 0 \\ t_6 &= 0 \oplus b_l = b_l \\ t_5 &= 0 \oplus 0 = 0 \\ t_8 &= 0 \oplus b_l = b_l \\ r_m + r_l &= b_m + b_l \end{aligned}$$

In consequence, the following holds for the values returned:

$$r_h + r_m + r_l = b_h + b_m + b_l$$

Clearly r_m et r_l do not overlap because the **Add12** procedure the algorithm calls at its last line assures this property. ∎

5.2 The addition operator Add233

Let us consider now the addition operator **Add233**. We will only analyse a simplified case where the arguments of the algorithm verify statically known bounds.

Algorithm 5.4 (Add233)
In: *a double-double number* $a_h + a_l$ *and a triple-double number* $b_h + b_m + b_l$
Out: *a triple-double number* $r_h + r_m + r_l$
Preconditions on the arguments:

$$\begin{aligned} |b_h| &\leq 2^{-2} \cdot |a_h| \\ |a_l| &\leq 2^{-53} \cdot |a_h| \\ |b_m| &\leq 2^{-\beta_o} \cdot |b_h| \\ |b_l| &\leq 2^{-\beta_u} \cdot |b_m| \end{aligned}$$

Algorithm:

$$\begin{aligned} &(r_h, t_1) \leftarrow \textbf{Add12}\,(a_h, b_h) \\ &(t_2, t_3) \leftarrow \textbf{Add12}\,(a_l, b_m) \\ &(t_4, t_5) \leftarrow \textbf{Add12}\,(t_1, t_2) \\ &t_6 \leftarrow t_3 \oplus b_l \\ &t_7 \leftarrow t_6 \oplus t_5 \\ &(r_m, r_l) \leftarrow \textbf{Add12}\,(t_4, t_7) \end{aligned}$$

Theorem 5.5 (Relative error of algorithm 5.4 Add233)
Let be $a_h + a_l$ *and* $b_h + b_m + b_l$ *the values taken in argument of algorithm 5.4* **Add233**. *Let the preconditions hold for this values.*
So the following holds for the values returned by the algorithm r_h, r_m *et* r_l

$$r_h + r_m + r_l = ((a_h + a_m + a_l) + (b_h + b_m + b_l)) \cdot (1 + \varepsilon)$$

where ε is bounded by

$$|\varepsilon| \leq 2^{-\beta_o-\beta_u-52} + 2^{-\beta_o-104} + 2^{-153}$$

The values r_m and r_l will not overlap at all and the overlap of r_h and r_m will be bounded by:

$$|r_m| \leq 2^{-\gamma} \cdot |r_h|$$

with

$$\gamma \geq \min(45, \beta_o - 4, \beta_o + \beta_u - 2)$$

Proof
We know using the properties of the **Add12** procedure that

$$\begin{aligned} r_h + t_1 &= a_h + b_h \\ t_2 + t_3 &= a_l + b_m \\ t_4 + t_5 &= t_1 + t_2 \\ r_m + r_l = t_4 + t_7 & \end{aligned}$$

Supposing that we dispose already of a term of the following form

$$t_7 = t_5 + t_3 + b_l + \delta$$

with a bounded $|\delta|$, we can note that

$$r_h + r_m + r_l = (a_h + a_l) + (b_h + b_m + b_l) + \delta$$

Let us now express t_7 by t_5, t_3 and b_l:

$$\begin{aligned} t_7 &= t_5 \oplus t_6 \\ &= t_5 \oplus (t_3 \oplus b_l) \\ &= (t_5 + (t_3 + b_l) \cdot (1 + \varepsilon_1)) \cdot (1 + \varepsilon_2) \end{aligned}$$

with $|\varepsilon_1| \leq 2^{-53}$ and $|\varepsilon_2| \leq 2^{-53}$.
We get in consequence

$$t_7 = t_5 + t_3 + b_l + t_3 \cdot \varepsilon_1 + b_l \cdot \varepsilon_1 + t_5 \cdot \varepsilon_2 + t_3 \cdot \varepsilon_2 + b_l \cdot \varepsilon_2 + t_3 \cdot \varepsilon_1 \cdot \varepsilon_2 + b_l \cdot \varepsilon_1 \cdot \varepsilon_2$$

and we can verify that the following upper bound holds for the absolute error δ:

$$\begin{aligned} |\delta| &= |t_3 \cdot \varepsilon_1 + b_l \cdot \varepsilon_1 + t_5 \cdot \varepsilon_2 + t_3 \cdot \varepsilon_2 + b_l \cdot \varepsilon_2 + t_3 \cdot \varepsilon_1 \cdot \varepsilon_2 + b_l \cdot \varepsilon_1 \cdot \varepsilon_2| \\ &\leq 2^{-53} \cdot |t_3| + 2^{-53} \cdot |b_l| + 2^{-53} \cdot |t_5| + 2^{-53} \cdot |b_l| + 2^{-106} \cdot |t_3| + 2^{-106} \cdot |b_l| \\ &\leq 2^{-52} \cdot |t_3| + 2^{-51} \cdot |b_l| + 2^{-53} \cdot |t_5| \end{aligned}$$

Let us get now some bounds for $|t_3|$, $|b_l|$ and $|t_5|$, all as a function of $|a_h|$:

$$|b_l| \leq 2^{-\beta_o-\beta_u} \cdot |b_h| \leq 2^{-\beta_o-\beta_u-2} \cdot |a_h|$$

which can be obtained using the preconditions' hypotheses. Further

$$\begin{aligned} |t_3| &\leq 2^{-53} \cdot |t_2| \\ &= 2^{-53} \cdot |\circ(a_l + b_m)| \\ &\leq 2^{-52} \cdot |a_l + b_m| \\ &\leq 2^{-52} \cdot |a_l| + 2^{-52} \cdot |b_m| \\ &\leq 2^{-105} \cdot |a_h| + 2^{-\beta_o-52} \cdot |b_h| \\ &\leq 2^{-105} \cdot |a_h| + 2^{-\beta_o-54} \cdot |a_h| \end{aligned}$$

and finally

$$\begin{aligned}
|t_5| &\leq 2^{-53} \cdot |t_4| \\
&= 2^{-53} \cdot |\circ(t_1 + t_2)| \\
&\leq 2^{-52} \cdot |t_1 + t_2| \\
&\leq 2^{-52} \cdot |t_1| + 2^{-52} \cdot |t_2| \\
&\leq 2^{-105} \cdot |r_h| + 2^{-52} \cdot |\circ(a_l + b_m)| \\
&\leq 2^{-105} \cdot |\circ(a_h + b_h)| + 2^{-51} \cdot |a_l + b_m| \\
&\leq 2^{-104} \cdot |a_h + b_h| + 2^{-51} \cdot |a_l| + 2^{-51} \cdot |b_m| \\
&\leq 2^{-104} \cdot |a_h| + 2^{-106} \cdot |a_h| + 2^{-104} \cdot |a_h| + 2^{-\beta_o - 53} \cdot |a_h| \\
&\leq |a_h| \cdot \left(2^{-102} + 2^{-\beta_o - 53}\right)
\end{aligned}$$

So we have

$$\begin{aligned}
|\delta| &\leq |a_h| \cdot \left(2^{-157} + 2^{-\beta_o - 106} + 2^{-\beta_o - \beta_u - 53} + 2^{-155} + 2^{-\beta_o - 106}\right) \\
&\leq |a_h| \cdot \left(2^{-\beta_o - \beta_u - 53} + 2^{-\beta_o - 105} + 2^{-154}\right)
\end{aligned}$$

Let us now give a lower bound for $|(a_h + a_l) + (b_h + b_m + b_l)|$ as a function of $|a_h|$ by getting out an upper bound for $|a_l + b_h + b_m + b_l|$ as such a function:

$$\begin{aligned}
|a_l + b_h + b_m + b_l| &\leq |a_l| + |b_h| + |b_m| + |b_l| \\
&\leq |a_h| \cdot \left(2^{-53} + 2^{-2} + 2^{-\beta_o - 2} + 2^{-\beta_o - \beta_u - 2}\right)
\end{aligned}$$

Since $\beta_o \geq 1$, $\beta_u \geq 1$ we can check that

$$|a_l + b_h + b_m + b_l| \leq 2^{-1} \cdot |a_h|$$

In consequence

$$|a_h + (a_l + b_h + b_m + b_l)| \geq \frac{1}{2} \cdot |a_h|$$

Using this lower bound, we can finally give an upper bound for the relative error ε of the considered procedure:

$$r_h + r_m + r_l = ((a_h + a_l) + (b_h + b_m + b_l)) \cdot (1 + \varepsilon)$$

with

$$|\varepsilon| \leq 2^{-\beta_o - \beta_u - 52} + 2^{-\beta_o - 104} + 2^{-153}$$

Last but not least, let us now analyse the additional overlaps generated by the procedure. It is clear that r_m and r_l do not overlap at all because they are computed by the **Add12** procedure. Let us merely examine now the overlap of r_h and r_m.
We begin by giving a lower bound for r_h as a function of a_h:

$$\begin{aligned}
|r_h| &= |\circ(a_h + b_h)| \\
&\geq \circ(|a_h + b_h|) \\
&\geq \circ\left(\frac{3}{4} \cdot |a_h|\right) \\
&\geq \circ\left(\frac{1}{2} \cdot |a_h|\right) \\
&= \frac{1}{2} \cdot |a_h|
\end{aligned}$$

Let us then find an upper bound for $|r_m|$ using also here a term which is a function of $|a_h|$:

$$|r_m| = |\circ(r_m + r_l)|$$

$$\begin{aligned}
&\leq 2 \cdot |r_m + r_l| \\
&= 2 \cdot |t_4 + t_7| \\
&\leq 2 \cdot |t_4| + 2 \cdot |t_5 + t_3 + b_l + \delta| \\
&\leq 2 \cdot |t_4| + 2 \cdot |t_5| + 2 \cdot |t_3| + 2 \cdot |b_l| + 2 \cdot |\delta| \\
&\leq 2 \cdot |t_4| + |a_h| \cdot \left(2^{-101} + 2^{-\beta_o - 52}\right) \\
&\quad + |a_h| \cdot \left(2^{-104} + 2^{-\beta_o - 53}\right) + |a_h| \cdot 2^{-\beta_o - \beta_u - 1} \\
&\quad + |a_h| \cdot \left(2^{-\beta_o - \beta_u - 52} + 2^{-\beta_o - 104} + 2^{-153}\right)
\end{aligned}$$

By bounding finally still $|t_4|$ by a term that is function of $|a_h|$

$$\begin{aligned}
|t_4| &= |\circ (t_1 + t_2)| \\
&\leq 2 \cdot |t_1| + 2 \cdot |t_2| \\
&\leq 2^{-52} \cdot |r_h| + 2 \cdot |\circ (a_l + b_m)| \\
&\leq 2^{-51} \cdot |a_h + b_h| + 4 \cdot |a_l + b_m| \\
&\leq |a_h| \cdot \left(2^{-\beta_o} + 2^{-49}\right)
\end{aligned}$$

we obtain

$$\begin{aligned}
|r_h| &\leq |a_h| \cdot \big(2^{-48} \\
&\quad + 2^{-\beta_o + 1} \\
&\quad + 2^{-101} \\
&\quad + 2^{-\beta_o - 52} \\
&\quad + 2^{-104} \\
&\quad + 2^{-\beta_o - 53} \\
&\quad + 2^{-\beta_o - \beta_u - 1} \\
&\quad + 2^{-\beta_o - \beta_u - 52} \\
&\quad + 2^{-\beta_o - 104} \\
&\quad + 2^{-153}\big) \\
&\leq |a_h| \cdot \left(2^{-47} + 2^{-\beta_o + 2} + 2^{-\beta_o - \beta_u}\right)
\end{aligned}$$

We finally check that we have

$$|r_m| \leq |r_h| \cdot \left(2^{-46} + 2^{-\beta_o + 3} + 2^{-\beta_o - \beta_u + 1}\right)$$

from which we can deduce the following bound

$$|r_m| \leq 2^{-\gamma} \cdot |r_h|$$

with

$$\gamma \geq \min\left(45, \beta_o - 4, \beta_o + \beta_u - 2\right)$$

This finishes the proof. ∎

6 Triple-double multiplication operators

6.1 The multiplication procedure Mul23

Let us go on with an analysis of the multiplication procedure **Mul23**.

Algorithm 6.1 (Mul23)
In: *two double-double numbers* $a_h + a_l$ *and* $b_h + b_l$

Out: *a triple-double number* $r_h + r_m + r_l$
Preconditions on the arguments:

$$\begin{aligned} |a_l| &\leq 2^{-53} \cdot |a_h| \\ |b_l| &\leq 2^{-53} \cdot |b_h| \end{aligned}$$

Algorithm:

$$\begin{aligned} &(r_h, t_1) \leftarrow \textbf{Mul12}\,(a_h, b_h) \\ &(t_2, t_3) \leftarrow \textbf{Mul12}\,(a_h, b_l) \\ &(t_4, t_5) \leftarrow \textbf{Mul12}\,(a_l, b_h) \\ &t_6 \leftarrow a_l \otimes b_l \\ &(t_7, t_8) \leftarrow \textbf{Add22}\,(t_2, t_3, t_4, t_5) \\ &(t_9, t_{10}) \leftarrow \textbf{Add12}\,(t_1, t_6) \\ &(r_m, r_l) \leftarrow \textbf{Add22}\,(t_7, t_8, t_9, t_{10}) \end{aligned}$$

Theorem 6.2 (Relative error of algorithm 6.1 Mul23)
Let be $a_h + a_l$ *and* $b_h + b_l$ *the values taken by arguments of algorithm 6.1* **Mul23**
So the following holds for the values returned r_h, r_m *and* r_l*:*

$$r_h + r_m + r_l = ((a_h + a_l) \cdot (b_h + b_l)) \cdot (1 + \varepsilon)$$

where ε *is bounded as follows:*

$$|\varepsilon| \leq 2^{-149}$$

The values returned r_m *and* r_l *will not overlap at all and the overlap of* r_h *et* r_m *will be bounded as follows:*

$$|r_m| \leq 2^{-48} \cdot |r_h|$$

Proof
Since algorithm 6.1 is relatively long, we will proceed by analysing sub-sequences. So let us consider first the following sequence:

$$\begin{aligned} &(t_2, t_3) \leftarrow \textbf{Mul12}\,(a_h, b_l) \\ &(t_4, t_5) \leftarrow \textbf{Mul12}\,(a_l, b_h) \\ &(t_7, t_8) \leftarrow \textbf{Add22}\,(t_2, t_3, t_4, t_5) \end{aligned}$$

Clearly t_7 and t_8 will not overlap. The same way t_2 and t_3 and t_4 and t_5 will not overlap and we know that we have exactly the following egalities

$$\begin{aligned} t_2 + t_3 &= a_h \cdot b_l \\ t_4 + t_5 &= b_h \cdot a_l \end{aligned}$$

Further we can check that

$$\begin{aligned} |t_3| &\leq 2^{-53} \cdot |\circ\,(a_h \cdot b_l)| \\ &\leq 2^{-52} \cdot |a_h \cdot b_l| \end{aligned}$$

and similarly

$$|t_5| \leq 2^{-52} \cdot |b_h \cdot a_l|$$

So we have on the one side

$$\begin{aligned} 2^{-53} \cdot |t_3 + t_5| &\leq 2^{-53} \cdot |t_3| + 2^{-53} \cdot |t_5| \\ &\leq 2^{-105} \cdot |a_h \cdot b_l| + 2^{-105} \cdot |b_h \cdot a_l| \end{aligned}$$

and on the other

$$\begin{aligned} 2^{-102} \cdot |t_2 + t_3 + t_4 + t_5| &\leq 2^{-102} \cdot |b_h \cdot a_l + a_h \cdot b_l| \\ &\leq 2^{-102} \cdot |b_h \cdot a_l| + 2^{-102} \cdot |a_h \cdot b_l| \end{aligned}$$

Using theorem 4.4 it is possible to note

$$t_7 + t_8 = a_h \cdot b_l + a_l \cdot b_h + \delta_1$$

with

$$|\delta_1| \leq 2^{-102} \cdot |a_h \cdot b_l| + 2^{-102} \cdot |a_l \cdot b_h|$$

Let us now consider the following sub-sequence of algorithm 6.1:

$$\begin{aligned} &(r_h, t_1) \leftarrow \textbf{Mul12}\,(a_h, b_h) \\ &t_6 \leftarrow a_l \otimes b_l \\ &(t_9, t_{10}) \leftarrow \textbf{Add12}\,(t_1, t_6) \end{aligned}$$

Trivially t_9 and t_{10} do not overlap. Additionally, one sees that we have exactly

$$r_h + t_1 = a_h \cdot b_h$$

and, exactly too,

$$t_9 + t_{10} = t_1 + t_6$$

So using

$$t_6 = a_l \otimes b_l = a_l \cdot b_l \cdot (1 + \varepsilon)$$

where $|\varepsilon| \leq 2^{-53}$ we get

$$\begin{aligned} r_h + t_9 + t_{10} &= r_h + t_1 + t_6 \\ &= a_h \cdot b_h + t_6 \\ &= a_h \cdot b_h + a_l \cdot b_l + \delta_2 \end{aligned}$$

with

$$|\delta_2| \leq 2^{-53} \cdot |a_l \cdot b_l|$$

Let us now bound $|t_9|$ with regard to $|a_h \cdot b_h|$:
We have

$$\begin{aligned} |t_9| &\leq \circ\,(|t_1| + |t_6|) \\ &\leq \circ\,(|t_1| + \circ\,(|a_l \cdot b_l|)) \\ &\leq \circ\left(|t_1| + \circ\left(2^{-106} \cdot |a_h \cdot b_h|\right)\right) \\ &\leq \circ\left(2^{-53} \cdot |a_h \cdot b_h| + 2^{-105} \cdot |a_h \cdot b_h|\right) \\ &\leq 2^{-51} \cdot |a_h \cdot b_h| \end{aligned}$$

With inequalities given, we can bound now the absolute and relative error of algorithm **Mul23** 6.1.
We know already that

$$t_7 + t_8 = a_h \cdot b_l + a_l \cdot b_h + \delta_1$$

where

$$|\delta_1| \leq 2^{-102} \cdot |a_h \cdot b_l| + 2^{-102} \cdot |b_h \cdot a_l|$$

and

$$r_h + t_9 + t_{10} = a_h \cdot b_h + a_l \cdot b_l + \delta_2$$

where

$$|\delta_2| \leq 2^{-53} \cdot |a_l \cdot b_l|$$

One remarks that

$$\begin{aligned} |t_8| &\leq 2^{-53} \cdot |t_7| \\ |t_{10}| &\leq 2^{-53} \cdot |t_9| \end{aligned}$$

and easily checks that

$$2^{-53} \cdot |t_{10} + t_8| \leq 2^{-101} \cdot |t_9| + 2^{-101} \cdot |t_7|$$

and that

$$2^{-102} \cdot |t_9 + t_{10} + t_7 + t_8| \leq 2^{-101} \cdot |t_9| + 2^{-101} \cdot |t_7|$$

So by means of the theorem 4.4, we obtain that

$$r_m + r_l = t_9 + t_{10} + t_7 + t_8 + \delta_3$$

where

$$|\delta_3| \leq 2^{-101} \cdot |t_9| + 2^{-101} \cdot |t_7|$$

So finally we get

$$r_h + r_m + r_l = a_h \cdot b_h + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l + \delta$$

where

$$\begin{aligned} |\delta| &= |\delta_1 + \delta_2 + \delta_3| \\ &\leq |\delta_1| + |\delta_2| + |\delta_3| \\ &\leq 2^{-102} \cdot |a_l \cdot b_h| \\ &\quad +2^{-102} \cdot |a_h \cdot b_l| \\ &\quad +2^{-53} \cdot |a_l \cdot b_l| \\ &\quad +2^{-152} \cdot |a_h \cdot b_h| \\ &\quad +2^{-101} \cdot |t_7| \end{aligned}$$

And for $|t_7|$ we obtain the following inequalities

$$\begin{aligned} |t_7| &\leq \circ(|t_7 + t_8|) \\ &\leq 2 \cdot |t_7 + t_8| \\ &\leq 2 \cdot \left(|a_h \cdot b_l| + |a_l \cdot b_h| + 2^{-102} \cdot |a_l \cdot b_h| + 2^{-102} \cdot |a_h \cdot b_l|\right) \\ &\leq 8 \cdot |a_h \cdot b_l| \end{aligned}$$

In consequence we can check

$$|\delta| \leq 2^{-150} \cdot |a_h \cdot b_h|$$

Let us give now an upper bound for $|a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l|$ as a function of $|a_h \cdot b_h|$:
We have

$$\begin{aligned} |a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l| &\leq |a_h \cdot b_l| + |a_l \cdot b_h| + |a_l \cdot b_l| \\ &\leq 2^{-51} \cdot |a_h \cdot b_h| \end{aligned}$$

from which we deduce that

$$\begin{aligned} |a_h \cdot b_h + (a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l)| &\geq |a_h \cdot b_h| \cdot \left(1 - 2^{-51}\right) \\ &\geq \frac{1}{2} \cdot |a_h \cdot b_h| \end{aligned}$$

Thus

$$|\delta| \leq 2^{-149} \cdot |a_h \cdot b_h + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_l|$$

So we can finally give an upper bound for the relative error ε of the multiplication procedure **Mul23** defined by algorithm 6.1:

$$r_h + r_m + r_l = (a_h + a_l) \cdot (b_h + b_l) \cdot (1 + \varepsilon)$$

with

$$|\varepsilon| \leq 2^{-149}$$

Before concluding, we must still analyse the overlap of the different components of the triple-double number returned by the algorithm. It is clear that r_m and r_l do not overlap because the **Add22** brick ensures this. Let us now consider the magnitude of r_m with regard to the one of r_h.
We give first a lower bound for $|r_h|$:

$$\begin{aligned} |r_h| &= |\circ(a_h \cdot b_h)| \\ &\geq \circ(|a_h \cdot b_h|) \\ &\geq \frac{1}{2} \cdot |a_h \cdot b_h| \end{aligned}$$

and then an upper bound for $|r_m|$:

$$\begin{aligned} |r_m| &\leq \circ(|r_m + r_l|) \\ &\leq \circ(|t_7 + t_8| + |t_9 + t_{10}| + \delta_3) \\ &\leq \circ(|a_h \cdot b_l| + |a_l \cdot b_h| + \delta_1 + |t_9| + |t_{10}| + \delta_3) \\ &\leq \circ\left(|a_h \cdot b_h| \cdot \left(2^{-53} + 2^{-53} + 2^{-155} + 2^{-155} + 2^{-51} + 2^{-104} + 2^{-152} + 2^{-151}\right)\right) \\ &\leq 2^{-49} \cdot |a_h \cdot b_h| \end{aligned}$$

From this we can deduce the final bound

$$|r_m| \leq 2^{-48} \cdot |r_h|$$

∎

6.2 The multiplication procedure Mul233

Let us concentrate now on the multiplication sequence **Mul233**.

Algorithm 6.3 (Mul233)
In: *a double-double number* $a_h + a_l$ *and a triple-double number* $b_h + b_m + b_l$
Out: *a triple-double number* $r_h + r_m + r_l$
Preconditions on the arguments:

$$\begin{aligned} |a_l| &\leq 2^{-53} \cdot |a_h| \\ |b_m| &\leq 2^{-\beta_o} \cdot |b_h| \\ |b_l| &\leq 2^{-\beta_u} \cdot |b_m| \end{aligned}$$

with

$$\begin{aligned} \beta_o &\geq 2 \\ \beta_u &\geq 1 \end{aligned}$$

Algorithm:

$$
\begin{array}{l}
(r_h, t_1) \leftarrow \mathbf{Mul12}\,(a_h, b_h)\\
(t_2, t_3) \leftarrow \mathbf{Mul12}\,(a_h, b_m)\\
(t_4, t_5) \leftarrow \mathbf{Mul12}\,(a_h, b_l)\\
(t_6, t_7) \leftarrow \mathbf{Mul12}\,(a_l, b_h)\\
(t_8, t_9) \leftarrow \mathbf{Mul12}\,(a_l, b_m)\\
t_{10} \leftarrow a_l \otimes b_l\\
(t_{11}, t_{12}) \leftarrow \mathbf{Add22}\,(t_2, t_3, t_4, t_5)\\
(t_{13}, t_{14}) \leftarrow \mathbf{Add22}\,(t_6, t_7, t_8, t_9)\\
(t_{15}, t_{16}) \leftarrow \mathbf{Add22}\,(t_{11}, t_{12}, t_{13}, t_{14})\\
(t_{17}, t_{18}) \leftarrow \mathbf{Add12}\,(t_1, t_{10})\\
(r_m, r_l) \leftarrow \mathbf{Add22}\,(t_{17}, t_{18}, t_{15}, t_{16})
\end{array}
$$

Theorem 6.4 (Relative error of algorithm 6.3 Mul233)
Let be $a_h + a_l$ and $b_h + b_m + b_l$ the values in argument of algorithm 6.3 **Mul233** *such that the given preconditions hold.*
So the following will hold for the values r_h, r_m and r_l returned

$$r_h + r_m + r_l = ((a_h + a_l) \cdot (b_h + b_m + b_l)) \cdot (1 + \varepsilon)$$

where ε is bounded as follows:

$$|\varepsilon| \leq \frac{2^{-99-\beta_o} + 2^{-99-\beta_o-\beta_u} + 2^{-152}}{1 - 2^{-53} - 2^{-\beta_o+1} - 2^{-\beta_o-\beta_u+1}} \leq 2^{-97-\beta_o} + 2^{-97-\beta_o-\beta_u} + 2^{-150}$$

The values r_m and r_l will not overlap at all and the following bound will be verified for the overlap of r_h and r_m:

$$|r_m| \leq 2^{-\gamma} \cdot |r_h|$$

where

$$\gamma \geq \min\,(48, \beta_o - 4, \beta_o + \beta_u - 4)$$

Proof
During this proof we will once again proceed by basic bricks that we will assemble in the end. Let us therefore start by the following one:

$$
\begin{array}{l}
(t_2, t_3) \leftarrow \mathbf{Mul12}\,(a_h, b_m)\\
(t_4, t_5) \leftarrow \mathbf{Mul12}\,(a_h, b_l)\\
(t_{11}, t_{12}) \leftarrow \mathbf{Add22}\,(t_2, t_3, t_4, t_5)
\end{array}
$$

Since we have the exact egalities

$$t_2 + t_3 = a_h \cdot b_m$$

and

$$t_4 + t_5 = a_h \cdot b_l$$

and since we know that t_2 and t_3 and t_4 and $_5$ do not overlap, it suffices to apply the bound proven at theorem 4.4. So we can check on the one hand

$$
\begin{array}{rcl}
2^{-53} \cdot |t_3 + t_5| & \leq & 2^{-53} \cdot |t_3| + 2^{-53} \cdot |t_5|\\
& \leq & 2^{-106} \cdot |t_2| + 2^{-106} \cdot |t_4|\\
& \leq & 2^{-105} \cdot |a_h \cdot b_m| + 2^{-105} \cdot |a_h \cdot b_l|\\
& \leq & 2^{-105-\beta_o} \cdot |a_h \cdot b_m| + 2^{-105-\beta_o-\beta_u} \cdot |a_h \cdot b_l|
\end{array}
$$

and on the other

$$
\begin{array}{rcl}
2^{-102} \cdot |t_2 + t_3 + t_4 + t_5| & = & 2^{-102} \cdot |a_h \cdot b_m + a_h \cdot b_l|\\
& \leq & 2^{-102} \cdot |a_h \cdot b_m| + 2^{-102} \cdot |a_h \cdot b_l|\\
& \leq & 2^{-102-\beta_o} \cdot |a_h \cdot b_h| + 2^{-102-\beta_o-\beta_u} \cdot |a_h \cdot b_l|
\end{array}
$$

In consequence, using the mentioned theorem, we obtain

$$t_{11} + t_{12} = a_h \cdot b_m + a_h \cdot b_l + \delta_1$$

with

$$|\delta_1| \leq 2^{-102-\beta_o} \cdot |a_h \cdot b_h| + 2^{-102-\beta_o-\beta_u} \cdot |a_h \cdot b_l|$$

Let us continue with the following part of the algorithm:

$$\begin{array}{l}(t_6, t_7) \leftarrow \mathbf{Mul12}\,(a_l, b_h)\\ (t_8, t_9) \leftarrow \mathbf{Mul12}\,(a_l, b_m)\\ (t_{13}, t_{14}) \leftarrow \mathbf{Add22}\,(t_6, t_7, t_8, t_9)\end{array}$$

We have

$$\begin{array}{rcl} 2^{-53} \cdot |t_7 + t_9| & \leq & 2^{-53} \cdot |t_7| + 2^{-53} \cdot |t_9| \\ & \leq & 2^{-106} \cdot |t_6| + 2^{-106} \cdot |t_8| \\ & \leq & 2^{-105} \cdot |a_l \cdot b_h| + 2^{-105} \cdot |a_l \cdot b_m| \\ & \leq & 2^{-158} \cdot |a_h \cdot b_h| + 2^{-158-\beta_o} \cdot |a_h b_h| \end{array}$$

and

$$\begin{array}{rcl} 2^{-102} \cdot |t_6 + t_7 + t_8 + t_9| & = & 2^{-102} \cdot |a_l \cdot b_h + a_l \cdot b_m| \\ & \leq & 2^{-102} \cdot |a_l \cdot b_h| + 2^{-102} \cdot |a_l \cdot b_m| \\ & \leq & 2^{-155} \cdot |a_h \cdot b_h| + 2^{-155-\beta_o} \cdot |a_h \cdot b_h| \end{array}$$

So we get

$$t_{13} + t_{14} = a_l \cdot b_h + a_l \cdot b_m + \delta_2$$

with

$$|\delta_2| \leq 2^{-155} \cdot |a_h \cdot b_h| + 2^{-155-\beta_o} \cdot |a_h \cdot b_h|$$

Let us now consider the brick that produces t_{15} and t_{16} out of the values in argument. By the properties of the **Add22** procedure, t_{11} and t_{12} and t_{13} and t_{14} do not overlap at all and verify thus the preconditions of the next **Add22** brick that will compute t_{15} and t_{16}. So it suffices to apply once again the absolute error bound of this procedure for obtaining

$$t_{15} + t_{16} = t_{11} + t_{12} + t_{13} + t_{14} + \delta_3$$

with $|\delta_3|$ which remains to be estimated.
So we have on the one hand

$$\begin{array}{rcl} 2^{-53} \cdot |t_{12} + t_{14}| & \leq & 2^{-53} \cdot |t_{12}| + 2^{-53} \cdot |t_{14}| \\ & \leq & 2^{-106} \cdot |t_{11}| + 2^{-106} \cdot |t_{13}| \end{array}$$

– which is an upper bound that can still be estimated by

$$\begin{array}{rcl} |t_{11}| & = & |\circ\,(t_{11} + t_{12})| \\ & \leq & \left|(t_{11} + t_{12}) \cdot \left(1 + 2^{-53}\right)\right| \\ & \leq & 2 \cdot |t_{11} + t_{12}| \\ & \leq & 2 \cdot |a_h \cdot b_m + a_h \cdot b_l + \delta_1| \\ & \leq & 2 \cdot (|a_h \cdot b_m| + |a_h \cdot b_l| + |\delta_1|) \\ & \leq & 2 \cdot \left(2^{-\beta_o} \cdot |a_h \cdot b_h| + 2^{-\beta_o-\beta_u} \cdot |a_h \cdot b_h| + 2^{-\beta_o-102} \cdot |a_h \cdot b_h| + 2^{-\beta_o-\beta_u-102} \cdot |a_h \cdot b_h|\right) \\ & \leq & |a_h \cdot b_h| \cdot \left(2^{-\beta_o+2} + 2^{-\beta_o-\beta_u+2}\right) \end{array}$$

which means, using also the following inequalities that

$$\begin{aligned}
|t_{13}| &= |\circ(t_{13}+t_{14})| \\
&\leq 2\cdot|t_{13}+t_{14}| \\
&\leq 2\cdot|a_l\cdot b_h + a_l\cdot b_m + \delta_2| \\
&\leq 2\cdot(|a_l\cdot b_h| + |a_l\cdot b_m| + |\delta_2|) \\
&\leq 2\cdot\left(2^{-53}\cdot|a_h\cdot b_h| + 2^{-53-\beta_o}\cdot|a_h\cdot b_h| + 2^{-155}\cdot|a_h\cdot b_h| + 2^{-155-\beta_o}\cdot|a_h\cdot b_h|\right) \\
&\leq 2^{-50}\cdot|a_h\cdot b_h|
\end{aligned}$$

we can finally check that we have on the one side

$$2^{-53}\cdot|t_{12}+t_{14}| \leq |a_h\cdot b_h|\cdot\left(2^{-\beta_o-104} + 2^{-\beta_o-\beta_u-104} + 2^{-156}\right)$$

And on the other

$$\begin{aligned}
2^{-102}\cdot|t_{11}+t_{12}+t_{13}+t_{14}| &\leq 2^{-102}\cdot|a_h\cdot b_m + a_h\cdot b_l + a_l\cdot b_h + a_l\cdot b_m + \delta_1 + \delta_2| \\
&\leq 2^{-102}\cdot|a_h\cdot b_h|\cdot \\
&\quad \big(2^{-\beta_o} \\
&\quad +2^{-\beta_o-\beta_u} \\
&\quad +2^{-53} \\
&\quad +2^{-\beta_o-53} \\
&\quad +2^{-\beta_o-102} \\
&\quad +2^{-\beta_o-\beta_u-102} \\
&\quad +2^{-155} \\
&\quad +2^{-\beta_o-155}\big) \\
&\leq |a_h\cdot b_h|\cdot\left(2^{-\beta_o-101} + 2^{-\beta_o-\beta_u-101} + 2^{-154}\right)
\end{aligned}$$

So we know that

$$t_{15}+t_{16} = t_{11}+t_{12}+t_{13}+t_{14}+\delta_3$$

with

$$|\delta_3| \leq |a_h\cdot b_h|\cdot\left(2^{-\beta_o-101} + 2^{-\beta_o-\beta_u-101} + 2^{-154}\right)$$

With this result we can note now

$$t_{15}+t_{16} = a_h\cdot b_m + a_h\cdot b_l + a_l\cdot b_h + a_l\cdot b_m + \delta_4$$

with

$$\begin{aligned}
|\delta_4| &\leq |a_h\cdot b_h|\cdot \\
&\quad \big(2^{-\beta_o-102} \\
&\quad +2^{-\beta_o-\beta_u-102} \\
&\quad +2^{-155} \\
&\quad +2^{-\beta_o-155} \\
&\quad +2^{-\beta_o-101} \\
&\quad +2^{-\beta_o-\beta_u-101}\big) \\
&\leq |a_h\cdot b_h|\cdot\left(2^{-\beta_o-100} + 2^{-\beta_o-\beta_u-100} + 2^{-155}\right)
\end{aligned}$$

Let us give now an upper bound for δ_5 defined by the following expression:

$$r_m + r_l = t_1 + a_l\cdot b_l + t_{15} + t_{16} + \delta_5$$

It is clear that t_{17} and t_{18} do not overlap. In contrast the **Add12** operation which adds t_1 to t_{10} is necessary because t_1 and t_{10} can overlap and even "overtake" each other:

$$|t_1| \geq 2^{-106} \cdot |a_h \cdot b_h| \vee t_1 = 0$$

and

$$|t_{10}| \leq 2^{-\beta_o-\beta_u-52} \cdot |a_h \cdot b_h|$$

The same argument tells us that the **Add12** must be conditional.
So we have

$$t_{17} + t_{18} = t_1 + t_{10}$$

and

$$t_{10} = a_l \cdot b_l + \delta'$$

with

$$|\delta'| \leq 2^{-106-\beta_o-\beta_u} \cdot |a_h \cdot b_h|$$

Let us apply once again the bound for the absolute error of the **Add22** procedure:
So we have on the one hand

$$\begin{aligned} 2^{-53} \cdot |t_{18} + t_{16}| &\leq 2^{-53} \cdot |t_{18}| + 2^{-53} \cdot |t_{18}| \\ &\leq 2^{-106} \cdot |t_{17}| + 2^{-106} \cdot |t_{15}| \end{aligned}$$

We can estimate this by

$$\begin{aligned} |t_{17}| &\leq |\circ(t_1 + t_{10})| \\ &\leq 2 \cdot |t_1 + t_{10}| \\ &\leq 2 \cdot |t_1| + 2 \cdot |t_{10}| \\ &\leq 2^{-52} \cdot |r_h| + 2 \cdot |\circ(a_l \cdot b_l)| \\ &\leq 2^{-52} \cdot |\circ(a_h \cdot b_h)| + 2^2 \cdot |a_l \cdot b_l| \\ &\leq 2^{-51} \cdot |a_h \cdot b_h| + 2^{-51-\beta_o-\beta_u} \cdot |a_h \cdot b_h| \end{aligned}$$

and by

$$\begin{aligned} |t_{15}| &= |\circ(t_{15} + t_{16})| \\ &\leq 2 \cdot |t_{15} + t_{16}| \\ &\leq 2 \cdot |a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_m + \delta_4| \\ &\leq |a_h \cdot b_h| \cdot \left(2^{-\beta_o+1} + 2^{-\beta_o-\beta_u+1} + 2^{-52} + 2^{-\beta_o-52} + 2^{-\beta_o-99} + 2^{-\beta_o-\beta_u-99} + 2^{-154}\right) \\ &\leq |a_h \cdot b_h| \cdot \left(2^{-\beta_o+2} + 2^{-\beta_o-\beta_u+2} + 2^{-51}\right) \end{aligned}$$

So finally, we have on the one hand

$$\begin{aligned} 2^{-53} \cdot |t_{18} + t_{16}| &\leq |a_h \cdot b_h| \cdot \left(2^{-157} + 2^{-157-\beta_o-\beta_u} + 2^{-104-\beta_o} + 2^{-104-\beta_o-\beta_u} + 2^{-157}\right) \\ &\leq |a_h \cdot b_h| \cdot \left(2^{-\beta_o-104} + 2^{-\beta_o-\beta_u-103} + 2^{-156}\right) \end{aligned}$$

And on the other

$$\begin{aligned} 2^{-102} \cdot |t_{17} + t_{18} + t_{15} + t_{16}| &\leq 2^{-102} \cdot |t_1 + a_l \cdot b_l + \delta' + a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h \\ &\quad + a_l \cdot b_m + \delta_4| \\ &\leq 2^{-102} \cdot \left(2^{-53} \cdot |r_h| \right. \\ &\quad + 2^{-53-\beta_o-\beta_u} \cdot |a_h \cdot b_h| \\ &\quad + 2^{-\beta_o} \cdot |a_h \cdot b_h| \\ &\quad \left. + 2^{-\beta_o-\beta_u} \cdot |a_h \cdot b_h| \right. \end{aligned}$$

$$\begin{aligned} & +2^{-53} \cdot |a_h \cdot b_h| \\ & +2^{-53-\beta_o} \cdot |a_h \cdot b_h| \\ & +2^{-106-\beta_o-\beta_u} \cdot |a_h \cdot b_h| \\ & + |a_h \cdot b_h| \cdot \left(2^{-100-\beta_o} + 2^{-100-\beta_o-\beta_u} + 2^{-155}\right)\Big) \\ \leq \quad & |a_h \cdot b_h| \cdot \left(2^{-101-\beta_o} + 2^{-101-\beta_o-\beta_u} + 2^{-153}\right) \end{aligned}$$

which means that we finally obtain the following

$$r_m + r_l = t_1 + a_l \cdot b_l + a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_m + \delta_6$$

with

$$|\delta_6| \leq |\delta_4 + \delta_5|$$

where

$$|\delta_5| \leq |a_h \cdot b_h| \cdot \left(2^{-101-\beta_o} + 2^{-101-\beta_o-\beta_u} + 2^{-153}\right)$$

Thus we can check that

$$\begin{aligned} |\delta_6| \quad & \leq \quad |a_h \cdot b_h| \cdot \left(2^{-100-\beta_o} + 2^{-100-\beta_o-\beta_u} + 2^{-155} + 2^{-101-\beta_o} + 2^{-101-\beta_o-\beta_u} + 2^{-153}\right) \\ & \leq \quad |a_h \cdot b_h| \cdot \left(2^{-99-\beta_o} + 2^{-99-\beta_o-\beta_u} + 2^{-152}\right) \end{aligned}$$

Let us now integrate the different intermediate results:
Since we know that the following egality is exact

$$r_h + t_1 = a_h \cdot b_h$$

we can check that

$$r_h + r_m + r_l = (a_h + a_l) \cdot (b_h + b_m + b_l) + \delta_6$$

We continue by giving a lower bound for $|(a_h + a_l) \cdot (b_h + b_m + b_l)|$ using a term which is a function of $|a_h \cdot b_h|$. We do so for being able to give a relative error bound. We first bound

$$|a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_m + a_l \cdot b_l|$$

by such a term.
We have

$$\begin{aligned} |a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_m + a_l \cdot b_l| \quad & \leq \quad |a_h \cdot b_m| + |a_h \cdot b_l| + |a_l \cdot b_h| + |a_l \cdot b_m| + |a_l \cdot b_l| \\ & \leq \quad |a_h \cdot b_h| \cdot \left(2^{-\beta_o} + 2^{-\beta_o-\beta_u} + 2^{-53}\right. \\ & \qquad \left. +2^{-\beta_o-53} + 2^{-\beta_o-53}\right) \\ & \leq \quad 2^{-\beta_o+1} \cdot |a_h \cdot b_h| + 2^{-\beta_o-\beta_u+1} \cdot |a_h \cdot b_h| \\ & \qquad +2^{-53} \cdot |a_h \cdot b_h| \end{aligned}$$

and we get

$$|(a_h + a_l) \cdot (b_h + b_m + b_l)| \geq |a_h \cdot b_h| \cdot \left(1 - 2^{-53} - 2^{-\beta_o+1} - 2^{-\beta_o-\beta_u+1}\right)$$

from which we deduce (since $\beta_o \geq 2$, $\beta_u \geq 1$)

$$|a_h \cdot b_h| \leq \frac{1}{1 - 2^{-53} - 2^{-\beta_o+1} - 2^{-\beta_o-\beta_u+1}} \cdot |(a_h + a_l) \cdot (b_h + b_m + b_l)|$$

Using this inequality we can finally give a bound for the relative error ε as follows:

$$r_h + r_m + r_l = (a_h + a_l) \cdot (b_h + b_m + b_l) \cdot (1 + \varepsilon)$$

with

$$|\varepsilon| \leq \frac{2^{-99-\beta_o} + 2^{-99-\beta_o-\beta_u} + 2^{-152}}{1 - 2^{-53} - 2^{-\beta_o+1} - 2^{-\beta_o-\beta_u+1}}$$

Let us recall that for this inequality, $\beta_o \geq 2$, $\beta_u \geq 1$ must hold which is the case.
It is certainly possible to estimate $|\varepsilon|$ by a term which is slightly less exact:

$$|\varepsilon| \leq 2^{-97-\beta_o} + 2^{-97-\beta_o-\beta_u} + 2^{-150}$$

because

$$1 - 2^{-53} - 2^{-\beta_o+1} - 2^{-\beta_o-\beta_u+1} \geq \frac{1}{4}$$

for $\forall \beta_o \geq 2, \beta_u \geq 1$.

In order to finish this proof, we must still give an upper bound for the maximal overlap generated by the algorithm 6.3 **Mul233**. Clearly r_m and r_l do not overlap because of the properties of the basic brick **Add22**. Let us give an upper bound for the overlap between r_h and r_m giving a term of the following form:

$$|r_m| \leq 2^{-\gamma} \cdot |r_h|$$

where we constate a lower bound for γ using a term in β_o and β_u.
Let us start by giving a lower bound for r_h as a function of $|a_h \cdot b_h|$.
We have

$$\begin{aligned} |r_h| &= |\circ (a_h \cdot b_h)| \\ &\geq \frac{1}{2} \cdot |a_h \cdot b_h| \end{aligned}$$

Then, let us give an upper bound for $|r_m|$ using a function of $|a_h \cdot b_h|$:

$$\begin{aligned} |r_m| &\leq |\circ (r_m + r_l)| \\ &\leq 2 \cdot |r_m + r_l| \\ &\leq 2 \cdot |t_1 + a_l \cdot b_l + a_h \cdot b_m + a_h \cdot b_l + a_l \cdot b_h + a_l \cdot b_m + \delta_6| \\ &\leq 2 \cdot |a_h \cdot b_h| \cdot \\ &\quad \cdot \left(2^{-52} + 2^{-\beta_o-\beta_u-53} + 2^{-\beta_o} + 2^{-\beta_o-\beta_u} + 2^{-53} + 2^{-\beta_o-53} + \right. \\ &\quad \left. 2^{-\beta_o-99} + 2^{-\beta_o-\beta_u-99} + 2^{-152}\right) \\ &\leq |a_h \cdot b_h| \cdot \left(2^{-50} + 2^{-\beta_o-2} + 2^{-\beta_o-\beta_u+2}\right) \end{aligned}$$

This implies that

$$\begin{aligned} |r_m| &\leq 2 \cdot |r_h| \cdot \left(2^{-50} + 2^{-\beta_o-2} + 2^{-\beta_o-\beta_u+2}\right) \\ &\leq |r_h| \cdot \left(2^{-49} + 2^{-\beta_o-3} + 2^{-\beta_o-\beta_u+3}\right) \end{aligned}$$

From which we can deduce

$$|r_m| \leq 2^{-\gamma} \cdot |r_h|$$

with

$$\gamma \geq \min\left(48, \beta_o - 4, \beta_o - \beta_u - 4\right)$$

This result finishes the proof. ∎

7 Final rounding procedures

The renormalisation sequence and all computational basic operators on triple-double numbers have been presented only for one reason: allowing for implementing efficiently elementary functions in double precision [4, 1, 3]. For obtaining this goal we are still lacking an important basic brick: the final rounding of a triple-double number into a double precision number. This rounding must be possible in each of the 4 known rounding modes [2]. In particular, we will distinguish between the round-to-nearest sequence and the ones for the directed rounding modes.
Let us start this discussion by introducing some notations:
We will notate

- $\circ(x) \in \mathbb{F}$ the rounding to the nearest double precision number of a real number $x \in \mathbb{R}$,
- $\triangle(x) \in \mathbb{F}$ the rounding towards $+\infty$ of a real $x \in \mathbb{R}$ into double precision,
- $\triangledown(x) \in \mathbb{F}$ the rounding towards $-\infty$ of a real $x \in \mathbb{R}$ into double precision and
- $\diamond(x) \in \mathbb{F}$ the rounding towards 0 of a real number $x \in \mathbb{R}$ into a double précision number.

Since the directed rounding modes behave all in a similar fashion we will make a slight abuse of our notations. An unspecified directed rounding mode will be notated also $\diamond(x)$.

Definition 7.1 (Correct rounding procedure)
Let be **A** *a procedure taking a non-overlapping triple-double number $x_h + x_m + x_l$ as argument. This number be such that $x_m = \circ(x_m + x_l)$. Let the procedure* **A** *return a double precision number x'.*
So we will say that **A** *is a correct rounding procedure for round-to-nearest-ties-to-even mode iff for all possible entries*

$$x' = \circ(x_h + x_m + x_l)$$

The same way **A** *is a correct rounding procedure for a directed rounding mode iff for all possible entries*

$$x' = \diamond(x_h + x_m + x_l)$$

In the sequel we will present two algorithms for final rounding – one for round-to-nearest mode, the other one for the directed modes – and we will prove their correctness with regard to definition 7.1.

7.1 Final rounding to the nearest even

Lemma 7.2 (Generation of half an ulp **or a quarter of an** ulp**)**
Let be x a non-subnormal floating point number different from ± 0, $\pm\infty$ and NaN *and such that x^- is not a subnormal number.*
Given the following instruction sequence:

$$\begin{aligned} t_1 &\leftarrow x^- \\ t_2 &\leftarrow x \ominus t_1 \\ t_3 &\leftarrow t_2 \otimes \tfrac{1}{2} \end{aligned}$$

we know that

- *if it exists a $k \in \mathbb{Z}$ such that $x = 2^k$ exactly so*

$$|t_3| = \frac{1}{4} \cdot \mathrm{ulp}(x)$$

- *if it does not exist any $k \in \mathbb{Z}$ such that $x = 2^k$ exactly so*

$$|t_3| = \frac{1}{2} \cdot \mathrm{ulp}(x)$$

Proof
Without loss of generality, we can suppose that x is positive because the definition of x^- and all floating point operations are symmetrical with regard to the sign [2] and because the egalities to be proven ignore it. Additionally since the floating point multiplication by an integer power of 2 is always exact, it suffices to show that $t_2 = \frac{1}{2} \cdot \mathrm{ulp}(x)$ if x is exactly an integer power of 2 and that $t_2 = \mathrm{ulp}(x)$ otherwise.
Let us begin by showing that we have the exact equation

$$t_2 = x - x^-$$

which means that the floating point substraction is exact. This is the case by Sterbenz' lemma [11] if

$$\frac{1}{2} \cdot x \leq x^- \leq 2 \cdot x$$

So let us show this inequality.
Since $x \neq 0$ and since it is not subnormal we know already that $x^- \neq 0$. Additionally $x^- > 0$ because $x > 0$ and x^- is its direct predecessor with regard to $<$. Further by definition 2.5, it is trivial to see that $\forall y \in \mathbb{F}. (y^-)^+ = y = (y^+)^-$.
Since x is positive and since x^- is therefore its predecessor with regard to $<$ we have

$$x^- < x < 2 \cdot x$$

Let us suppose now that

$$x^- < \frac{1}{2} \cdot x$$

Since x is not subnormal et since it is positive, there exist $e \in \mathbb{Z}$ and $m \in \mathbb{N}$ such that

$$x = 2^e \cdot m$$

with

$$2^{p-1} \leq m < 2^p$$

where $p \geq 2$ is the format's precision; in particular, for double precision, $p = 53$.
Given that x^- is not subnormal neither and positive, too, it is the predecessor of x and verifies

$$x^- = \begin{cases} 2^e \cdot (m-1) & \text{if} \quad m - 1 \geq 2^{p-1} \\ 2^{e-1} \cdot (2^p - 1) & \text{otherwise} \end{cases}$$

So two cases must be treated separately:

1st case: $x^- = 2^e \cdot (m-1)$

We get here with the hypotheses supposed

$$\begin{aligned} x^- &< \frac{1}{2} \cdot x \\ 2^e \cdot (m-1) &< \frac{1}{2} \cdot 2^e \cdot m \\ m - 1 &< \frac{1}{2} \cdot m \\ m &< 2 \end{aligned}$$

In contrast $m \geq 2^{p-1}$ and $p \geq 2$; so we have contradiction in this case.

2nd case: $x^- = 2^{e-1} \cdot (2^p - 1)$

We can check

$$\begin{aligned} x^- &< \frac{1}{2} \cdot x \\ 2^{e-1} \cdot \left(2^{p-1} - 1\right) &< \frac{1}{2} \cdot 2^e \cdot m \\ 2^{e-1} \cdot \left(2^{p-1} - 1\right) &< 2^{e-1} \cdot m \\ 2^p - 1 &< m \end{aligned}$$

In contrast $m < 2^p$, thus $2^p - 1 < m < 2^p$. This is a contradiction because the inequalities are strict and $m \in \mathbb{N}$.

So Sterbenz' lemma [11] can be applied and we get the exact equation

$$t_2 = x - x^-$$

It is now important to see that

$$x^+ = \begin{cases} 2^e \cdot (m+1) & \text{if} \quad m+1 < 2^p \\ 2^{e+1} \cdot 2^{p-1} & \text{otherwise} \end{cases}$$

Further, without loss of generality, we can suppose that $x^+ \neq +\infty$ and that therefore

$$\operatorname{ulp}(x) = x^+ - x$$

If ever we could not suppose this, it would suffice to apply definition 2.6 of the ulp function which would only change the exponent e by 1 in the sequel.
So at this stage of the proof, two different cases are to be treated: x is or is not exactly an integer power of 2.

1st case: x is not exactly an integer power of 2

So we get $x = 2^e \cdot m$ with $2^{p-1} < m < 2^p$ from which we deduce that $m - 1 \geq 2^{p-1}$ and that, finally,

$$x^- = 2^e \cdot (m-1)$$

So still two sub-cases present themselves:

case a): $x^+ = 2^e \cdot (m+1)$

We can check that

$$\begin{aligned} \operatorname{ulp}(x) &= x^+ - x \\ &= 2^e \cdot (m+1) - 2^e \cdot m \\ &= 2^e \cdot (m+1-m) \\ &= 2^e \end{aligned}$$

and

$$\begin{aligned} t_2 &= x - x^- \\ &= 2^e \cdot m - 2^e \cdot (m-1) \\ &= 2^e \cdot (m - m + 1) \\ &= 2^e \end{aligned}$$

So we know that

$$t_2 = \mathrm{ulp}\,(x)$$

case b): $x^+ = 2^e \cdot (m+1)$

So in order to get x^+ being equal to $2^e \cdot (m+1)$, we must have $m+1 \geq 2^p$.
In contrast we can show that $m+1 \leq 2^p$ as follows:
Let us suppose that $m+1 > 2^p$. Since $m < 2^p$ because x is not subnormal we get

$$2^p - 1 < m < 2^p$$

In contrast, the inequalities are strict and $m \in \mathbb{N}$, thus contradiction.
We therefore know that

$$m = 2^p - 1$$

So we get

$$\begin{aligned} \mathrm{ulp}\,(x) &= x^+ - x \\ &= 2^{e+1} \cdot 2^{p-1} - 2^e \cdot (2^p - 1) \\ &= 2^e \cdot 2^p - 2^e \cdot 2^p + 2^e \\ &= 2^e \end{aligned}$$

and

$$\begin{aligned} t_2 &= x - x^- \\ &= 2^e \cdot (2^p - 1) - 2^e \cdot (2^p - 2) \\ &= 2^e \cdot 2^p - 2^e - 2^e \cdot 2^p + 2 \cdot 2^e \\ &= 2^e \end{aligned}$$

Thus we have still

$$t_2 = \mathrm{ulp}\,(x)$$

2nd case: x **is exactly an integer power of** 2

So in this case we verify that

$$x = 2^e \cdot 2^{p-1}$$

and therefore $m = 2^{p-1}$.
In consequence,

$$x^+ = 2^e \cdot \left(2^{p-1} + 1\right)$$

because we got $2^{p-1} + 1 < 2^p$ since $p \geq 2$.
The same way

$$x^- = 2^{e-1} \cdot (2^p - 1)$$

because, trivially, $2^{p-1} - 1 < 2^{p-1}$.
So we get

$$\begin{aligned} \mathrm{ulp}\,(x) &= x^+ - x \\ &= 2^e \cdot \left(2^{p-1} + 1\right) - 2^e \cdot 2^{p-1} \\ &= 2^e \cdot 2^{p-1} + 2^e - 2^e \cdot 2^{p-1} \\ &= 2^e \end{aligned}$$

and

$$\begin{aligned} t_2 &= x - x^- \\ &= 2^e \cdot 2^{p-1} - 2^{e-1} \cdot (2^p - 1) \\ &= 2^e \cdot 2^{p-1} - 2^{e-1} \cdot 2^p + 2^{e-1} \\ &= 2^{e-1} \\ &= \frac{1}{2} \cdot 2^e \end{aligned}$$

Thus we can check that

$$t_2 = \frac{1}{2} \cdot \mathrm{ulp}\,(x)$$

∎

Lemma 7.3 (Generation of half an ulp**)**
Let be x a non-subnormal floating point number different from ± 0, $\pm\infty$ and NaN.
Let be the following instruction sequence:

$$\begin{aligned} t_1 &\leftarrow x^+ \\ t_2 &\leftarrow t_1 \ominus x \\ t_3 &\leftarrow t_2 \otimes \tfrac{1}{2} \end{aligned}$$

So the following holds

$$|t_3| = \frac{1}{2} \cdot \mathrm{ulp}\,(x)$$

and one knows that

$$x > 0 \text{ iff } t_3 > 0$$

Proof
In the beginning we will show the first equation; the equivalence of the signs will be shown below. So, without loss of generality, we can suppose that x is positive because the definition of x^+ and all floating point operations are symmetrical with regard to the sign [2] and because the equation to be shown ignorates it. Further since the floating point multiplication by an integer power of 2 is always exact, it suffices to show that $t_2 = \frac{1}{2} \cdot \mathrm{ulp}\,(x)$.
Let us still start the proof by showing that the substraction

$$t_2 \leftarrow t_1 \ominus x$$

is exact by Sterbenz' lemma [11]. We must therefore show that

$$\frac{1}{2} \cdot x \leq x^+ \leq 2 \cdot x$$

Since x is positive and since x^+ is its direct successor in the ordered set of the floating point numbers, we know already that $x < x^+$. So trivially, we get

$$\frac{1}{2} \cdot x < x < x^+$$

Let us suppose now that

$$x^+ > 2 \cdot x$$

Since x is not subnormal and since it is positive, there exist $e \in \mathbb{Z}$ and $m \in \mathbb{N}$ such that

$$x = 2^e \cdot m$$

with

$$2^{p-1} \leq m < 2^p$$

where $p \geq 2$ is the precision of the format.
Further we know that

$$x^+ = \begin{cases} 2^e \cdot (m+1) & \text{if} \quad m+1 < 2^p \\ 2^{e+1} \cdot 2^{p-1} & \text{otherwise} \end{cases}$$

So two different cases show up:

1st case: $x^+ = 2^e \cdot (m+1)$

We have

$$\begin{aligned} x^+ &> 2 \cdot x \\ 2^e \cdot (m+1) &> 2 \cdot 2^e \cdot m \\ m+1 &> 2 \cdot m \\ 1 &> m \end{aligned}$$

In contrast, $m \geq 2^{p-1}$ and $p \geq 2$, thus contradiction.

2nd case: $x^+ = 2^{e+1} \cdot 2^{p-1}$

So in this case, we have $m + 1 \geq 2^p$ and therefore $m = 2^p - 1$ because $m \leq 2^p - 1$ holds since x is not subnormal.
We get thus

$$\begin{aligned} x^+ &> 2 \cdot x \\ 2^{e+1} \cdot 2^{p-1} &> 2 \cdot 2^e \cdot (2^p - 1) \\ 2^e \cdot 2^p &> 2 \cdot 2^e \cdot 2^p - 2 \cdot 2^e \\ 2^p &> 2 \cdot 2^p - 2 \\ 2 &> 2^p \end{aligned}$$

In contrast $p \geq 2$ which is contradictory.
So we can apply Sterbenz' lemma [11] and we get immediately that

$$t_2 = x^+ - x = \text{ulp}\,(x)$$

by the definition of the ulp function
Let us show now that

$$x > 0 \text{ iff } t_3 > 0$$

Let us suppose that x is positive. In consequence x^+ is its successor with regard to $<$ et and we get

$$x^+ - x > 0$$

which means that

$$\begin{aligned} t_2 &= x^+ \ominus x \\ &= \circ\left(x^+ - x\right) \\ &> 0 \end{aligned}$$

because the rounding function is monotonic for positive numbers.
In contrast, if x is negative x^+ is its predecessor with regard to $<$ and we get thus $x^+ - x < 0$.
We conclude in this case in the same way. ∎

Lemma 7.4 (Signs of the generated values)
Let be $x \in \mathbb{F}$ a non-subnormal floating point number different from 0.
Given the following instruction sequence

$$\begin{aligned} t_1 &\leftarrow x^- \\ t_2 &\leftarrow x \ominus t_1 \\ t_3 &\leftarrow t_2 \otimes \tfrac{1}{2} \\ t_4 &\leftarrow x^+ \\ t_5 &\leftarrow t_4 \ominus x \\ t_6 &\leftarrow t_5 \otimes \tfrac{1}{2} \end{aligned}$$

the values t_3 and t_6 have the same sign.

Proof
It is clear that it suffices to show that t_2 and t_5 have the same sign. Because of definition 2.5 of x^+ and x^-, we are obliged to treat two different cases which depend on whether x is positive or not.

1st case: $x > 0$

So x^+ is the successor of x with regard to the order $<$ on the floating point numbers and x^- is its predecessor. Formally we have

$$x^- < x < x^+$$

Thus

$$\begin{aligned} x - x^- &> 0 \\ x^+ - x &> 0 \end{aligned}$$

Due to the monotony of the rounding function, we obtain

$$\begin{aligned} \circ\left(x - x^-\right) &> \circ(0) \\ \circ\left(x^+ - x\right) &> \circ(0) \end{aligned}$$

and thus, since 0 is exactly representable,

$$\begin{aligned} x \ominus x^- &> 0 \\ x^+ \ominus x &> 0 \end{aligned}$$

which is the fact to be shown.

2nd case: $x < 0$

We get in this case that $x^+ < x < x^-$ and we finish the proof in a complete analogous way to the 1st case. ∎

In the sequel, we will use the sign function $\operatorname{sgn}(x)$ which we define as follows:

$$\forall x \in \mathbb{R} \;.\; \operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{otherwise} \end{cases}$$

Lemma 7.5 (Equivalence between a XOR and a floating point multiplication)
Let be $x, y \in \mathbb{F}$ two floating point numbers such that $x \neq 0$, $y \neq 0$.
So,

$$x \otimes y \geq 0$$

implies

$$x > 0 \text{ XOR } y > 0$$

Proof
Clearly $x \otimes y \geq 0$ implies $\circ(x \cdot y) \geq 0$. By monotony of the rounding function, this yields to $x \cdot y \geq 0$. Trivially one sees that this means that $x \geq 0$ XOR $y \geq 0$. Since the equations are not possible by hypothesis, we can conclude. ∎

Lemma 7.6 (Round-to-nearest-ties-to-even of the lower significance parts)
Let be x_m and x_l two non-subnormal floating point numbers such that $\exists e \in \mathbb{Z}\ .\ 2^e = t$ such that $x_m = t^-$ and that $x_l = \frac{1}{4} \cdot \mathrm{ulp}(t)$.
So,

$$x_m \neq \circ(x_m + x_l)$$

Similar, let be x_m and x_l two non-subnormal floating point numbers such that $\exists e \in \mathbb{Z}\ .\ 2^e = t$ such that $x_m = t^+$ and that $x_l = \frac{1}{2} \cdot \mathrm{ulp}(t)$.
So,

$$x_m \neq \circ(x_m - x_l)$$

Proof
In both cases t is representable as a floating point number because x_m is not subnormal. Since t is an integer power of 2, the mantissa of t is even. Therefore the mantissa of x_m is odd in both cases because x_m is either the direct predecessor or the direct successor of t.
Let us show now that $|x_l| = \frac{1}{2} \cdot \mathrm{ulp}(x_m)$. If $x_m = t^-$ then we can deduce

$$\begin{aligned}
\frac{1}{2} \cdot \mathrm{ulp}(x_m) &= \frac{1}{2} \cdot \left(x_m^+ - x_m\right) \\
&= \frac{1}{2} \cdot \left(t - t^-\right) \\
&= -\frac{1}{4} \cdot \left(t^+ - t\right) \\
&= -\frac{1}{4} \cdot \mathrm{ulp}(t) \\
&= -x_l
\end{aligned}$$

using amongst others lemma 2.9. If $x_m = t^+$ then we know that x_m is not an integer power of 2 because t is one and we are supposing that the format's precision p is greater than 2 bits. So it exists $e \in \mathbb{Z}$ and $m = 2^p$ such that $t = 2^e \cdot m$

$$\begin{aligned}
\frac{1}{2} \cdot \mathrm{ulp}(x_m) &= \frac{1}{2} \cdot \left(x_m^+ - x_m\right) \\
&= \frac{1}{2} \cdot \left(2^e \cdot (m+2) - 2^e \cdot (m+1)\right) \\
&= \frac{1}{2} \cdot \left(2^e \cdot (m+1) - 2^e \cdot m\right) \\
&= \frac{1}{2} \cdot \mathrm{ulp}(t) \\
&= x_l
\end{aligned}$$

So, in both cases, $x_m + x_l$ is located exactly at the middle of two floating point numbers that can be expressed with the exponent of x_m or its successor and its predecessor. Since x_m has an odd mantissa the rounding with done away from it. ∎

Algorithm 7.7 (Final rounding to the nearest (even))
In: *a triple-double number $x_h + x_m + x_l$*
Out: *a double precision number x' returned by the algorithm*
Preconditions on the arguments:

- *x_h and x_m as well as x_m and x_l do not overlap*

- $x_m = \circ(x_m + x_l)$
- $x_h \neq 0$, $x_m \neq 0$ *et* $x_l \neq 0$
- $\circ(x_h + x_m) \notin \{x_h^-, x_h, x_h^+\} \Rightarrow |(x_h + x_m) - \circ(x_h + x_m)| \neq \frac{1}{2} \cdot \mathrm{ulp}(\circ(x_h + x_m))$

Algorithm:

$t_1 \leftarrow x_h^-$
$t_2 \leftarrow x_h \ominus t_1$
$t_3 \leftarrow t_2 \otimes \frac{1}{2}$
$t_4 \leftarrow x_h^+$
$t_5 \leftarrow t_4 \ominus x_h$
$t_6 \leftarrow t_5 \otimes \frac{1}{2}$

if $(x_m \neq -t_3)$ **and** $(x_m \neq t_6)$ **then**
  return $(x_h \oplus x_m)$
else
  if $(x_m \otimes x_l > 0.0)$ **then**
    if $(x_h \otimes x_l > 0.0)$ **then**
      return x_h^+
    else
      return x_h^-
    end if
  else
    return x_h
  end if
end if

Theorem 7.8 (Correctness of the final rounding procedure 7.7)
Let be **A** *the algorithm 7.7 said "Final rounding to the nearest (even)". Let be* $x_h + x_m + x_l$ *triple-double number for which the preconditions of algorithm* **A** *hold. Let us notate* x' *the double precision number returned by the procedure.*
So

$$x' = \circ(x_h + x_m + x_l)$$

i.e. **A** *is a correct rounding procedure for round-to-nearest-ties-to-even mode.*

Proof
During this proof we will proceed as follows: one easily sees that the presented procedure can only return four different results which are $x_h \oplus x_m$, x_h, x_h^+ and x_h^-. The choices made by the branches of the algorithm imply for each of this results additional hypotheses on the arguments' values. It will therefore suffice to show for each of this four choices that the rounding of the arguments is equal to the result returned under this hypotheses. In contrast, the one that can be easily deduced from the tests on the branches, which use a floating point multiplication in fact, are not particularly adapted to what is needed in the proof. Using amongst others lemma 7.5, one sees that 9 different simply analysable cases are possible out of which one is a special one and 8 have a very regular form:

1. If the first branch is taken, we know that

$$x_m \neq \mathrm{sgn}(x_h) \cdot \frac{1}{2} \cdot \mathrm{ulp}(x_h)$$

and that

$$x_m \neq -\mathrm{sgn}(x_h) \cdot \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}(x_h) & \text{if } \exists e \in \mathbb{Z} \,.\, 2^e = x_h \\ \frac{1}{2} \cdot \mathrm{ulp}(x_h) & \text{otherwise} \end{cases}$$

as per lemmas 7.2, 7.3 and 7.4. In this case $x_h \oplus x_m$ will be returned.

2. If the first branch is not taken, we know already very well the absolute value of x_m: we can therefore suppose that

$$|x_m| = \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}\,(x_h) & \text{if } \exists e \in \mathbb{Z}\ .\ 2^e = x_h \\ \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) & \text{otherwise} \end{cases}$$

It is thus natural that x_m does not play any role in the following computations of the value to be returned but by its sign. Using 7.5 we know that the two tests that follow are equivalent to **if** $x_m > 0$ XOR $x_l > 0$ and to **if** $x_h > 0$ XOR $x_l > 0$. It is easy to check that the values returned depending on the signs of x_h, x_m and x_l obey to this scheme:

Case	x_h	x_m	x_l	x_m XOR x_l	x_h XOR x_l	Return val. x'	Interpreted val. x'
a.)	+	+	+	+	+	x_h^+	$\mathrm{succ}\,(x_h)$
b.)	+	+	-	-	-	x_h	x_h
c.)	+	-	+	-	+	x_h	x_h
d.)	+	-	-	+	-	x_h^-	$\mathrm{pred}\,(x_h)$
e.)	-	+	+	+	-	x_h^-	$\mathrm{succ}\,(x_h)$
f.)	-	+	-	-	+	x_h	x_h
g.)	-	-	+	-	-	x_h	x_h
h.)	-	-	-	+	+	x_h^+	$\mathrm{pred}\,(x_h)$

We see now that the returned value x' expressed as x_h, $\mathrm{succ}\,(x_h)$ or $\mathrm{pred}\,(x_h)$ in cases a.) through d.) are equivalent to cases h.) through e.). We will consider them thus equivalently; of course, doing so, we will not any longer be able to suppose anything concerning the magnitude and the sign of x_h.

Let us start the proof by showing the correctness of the first case. Since x_h and x_m do not overlap by hypothesis, we know by lemma 2.7 that

$$|x_m| < \mathrm{ulp}\,(x_h)$$

So we can notate the following

$$x_m \in I'_1 \cup I'_2 \cup I'_3 \cup I'_4 \cup I'_5 \cup I'_6$$

with

$$\begin{aligned} I'_1 &= \left]-\mathrm{ulp}\,(x_h)\,; -\frac{3}{4} \cdot \mathrm{ulp}\,(x_h)\right] \\ I'_2 &= \left]-\frac{3}{4} \cdot \mathrm{ulp}\,(x_h)\,; -\frac{1}{2} \cdot \mathrm{ulp}\,(x_h)\right[\\ I'_3 &= \left[-\frac{1}{2} \cdot \mathrm{ulp}\,(x_h)\,; -\tau\right[\\ I'_4 &=]-\tau; 0] \\ I'_5 &= \left[0; \frac{1}{2} \cdot \mathrm{ulp}\,(x_h)\right[\\ I'_6 &= \left]\frac{1}{2} \cdot \mathrm{ulp}\,(x_h)\,; \mathrm{ulp}\,(x_h)\right[\end{aligned}$$

where

$$\tau = \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}\,(x_h) & \text{if } \exists e \in \mathbb{Z}\ .\ 2^e = x_h \\ \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) & \text{otherwise} \end{cases}$$

This is equivalent to claiming

$$x_m \in I_1 \cup I_2 \cup I_3 \cup I_4 \cup I_5 \cup I_6$$

where

$$\begin{aligned}
I_1 &= \left[(-\mathrm{ulp}\,(x_h))^-\,;\left(-\frac{3}{4}\cdot\mathrm{ulp}\,(x_h)\right)^+\right]\\
I_2 &= \left[\left(\frac{3}{4}\cdot\mathrm{ulp}\,(x_h)\right)^-\,;\left(-\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^+\right]\\
I_3 &= \left[\left(-\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^-\,;(-\tau)^+\right]\\
I_4 &= \left[(-\tau)^-\,;0\right]\\
I_5 &= \left[0;\left(\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^-\right]\\
I_6 &= \left[\left(\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^+\,;(\mathrm{ulp}\,(x_h))^-\right]
\end{aligned}$$

because x_m is a floating point number and because all bounds of the intervals are floating point numbers, too. So we can express their predecessors and successors by z^+ et z^-. Thus $\forall i = 1,\ldots,6\ .\ I_i' = I_i$. It is clear that the set of floating point numbers I_3 is empty if $\tau = \frac{1}{2}\cdot\mathrm{ulp}\,(x_h)$. Further we know that x_m and x_l do not overlap and that $x_m = \circ\,(x_m + x_l)$ by hypothesis. This means that

$$|x_l| \leq \frac{1}{2}\cdot\mathrm{ulp}\,(x_m) \leq \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))$$

and we can write

$$x_m + x_l \in (J_1 \cup J_2 \cup J_3 \cup J_4 \cup J_5 \cup J_6)\setminus U$$

with

$$\begin{aligned}
J_1 &= \left[(-\mathrm{ulp}\,(x_h))^- - \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\,;\left(-\frac{3}{4}\cdot\mathrm{ulp}\,(x_h)\right)^+ + \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\right]\\
J_2 &= \left[\left(-\frac{3}{4}\cdot\mathrm{ulp}\,(x_h)\right)^- - \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\,;\left(-\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^+ + \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\right]\\
J_3 &= \left[\left(-\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^- - \frac{1}{2}\cdot\mathrm{ulp}\,(\xi_1)\,;(-\tau)^+ + \frac{1}{2}\cdot\mathrm{ulp}\,(\xi_1)\right]\\
J_4 &= \left[(-\tau)^- - \frac{1}{2}\cdot\mathrm{ulp}\,(\xi_2)\,;0\right]\\
J_5 &= \left[0;\left(\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^- + \frac{1}{2}\cdot\mathrm{ulp}\,(\xi_3)\right]\\
J_6 &= \left[\left(\frac{1}{2}\cdot\mathrm{ulp}\,(x_h)\right)^+ - \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\,;(\mathrm{ulp}\,(x_h))^- + \frac{1}{2}\cdot\mathrm{ulp}\,(\mathrm{ulp}\,(x_h))\right]
\end{aligned}$$

where

$$\begin{aligned}
\xi_1 &= \frac{1}{2}\cdot\mathrm{ulp}\,(x_h) \in I_3\\
\xi_2 &= \tau \in I_4\\
\xi_3 &= \frac{1}{2}\cdot\mathrm{ulp}\,(x_h) \in I_5
\end{aligned}$$

and where U is the set of the impossible cases for $x_m + x_l$. The word "impossible" refers here to the facts caused by the property that $x_m = \circ\,(x_m + x_l)$.

Let us still remark that the intervals J_3, J_4 and J_5 are well defined as per lemma 2.13 and that it is important to see that it does not suffice to estimate their bounds by the less exact inequality that follows:

$$\mathrm{ulp}\left(x_m\right) \leq \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)$$

which would mean that

$$\xi_i = \mathrm{ulp}\left(x_h\right)$$

Since the images of the ulp function are always integer powers of 2, the difference of their predecessors and themselves can be as small as half an ulp of an ulp of x_h which would be a too inexact estimate.

Let us continue now with the simplification of the bounds of the intervals J_i. The purpose of this will be showing that $x_m + x_l$ are always intervals such that one can decide the rounding $\circ\left(x_h + \left(x_m + x_l\right)\right)$ without using the rule of even rounding. Let us remark already that we know that $\forall i = 1, \ldots, 6 \;.\; I_i \subseteq J_i$.

Since $\mathrm{ulp}\left(x_h\right)$ is a non-subnormal floating point number that is positive and equal to an integer power of 2, we get using lemmas 2.8 and 2.9 that

$$\begin{aligned}
\left(-\mathrm{ulp}\left(x_h\right)\right)^- - \frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= -\mathrm{ulp}\left(x_h\right)^- - \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= -\mathrm{ulp}\left(x_h\right) + \left(\mathrm{ulp}\left(x_h\right) - \mathrm{ulp}\left(x_h\right)^-\right) \\
&\quad -\frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= -\mathrm{ulp}\left(x_h\right)
\end{aligned}$$

and similarly

$$\begin{aligned}
\mathrm{ulp}\left(x_h\right)^- + \frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= \mathrm{ulp}\left(x_h\right)^- + \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= \mathrm{ulp}\left(x_h\right)^- + \left(\mathrm{ulp}\left(x_h\right) - \mathrm{ulp}\left(x_h\right)^-\right) \\
&= \mathrm{ulp}\left(x_h\right)
\end{aligned}$$

Further, still analogously to the previous cases and using lemma 2.11,

$$\begin{aligned}
\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^- + \frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)^- + \frac{1}{4}\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) \\
&= \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^- + \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right)\right) \\
&= \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^- + \mathrm{ulp}\left(x_h\right) - \mathrm{ulp}\left(x_h\right)^-\right) \\
&= \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)
\end{aligned}$$

and

$$\begin{aligned}
\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^+ - \frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)^+ - \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)
\end{aligned}$$

Then

$$\begin{aligned}
\left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^- - \frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right) &= -\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)^- - \frac{1}{4}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= \frac{1}{2}\cdot\left(-\mathrm{ulp}\left(x_h\right)^- - \left(\mathrm{ulp}\left(x_h\right) - \mathrm{ulp}\left(x_h\right)^-\right)\right) \\
&= -\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)
\end{aligned}$$

further, using also lemma 2.12,

$$\begin{aligned}\left(-\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)\right)^{+}+\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= -\frac{1}{4}\cdot\left(3\cdot\mathrm{ulp}\left(x_h\right)\right)^{+}+\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= -\frac{1}{4}\cdot\left(3\cdot\mathrm{ulp}\left(x_h\right)^{+}-\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\right)\\ &\quad +\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= -\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)^{+}+\frac{3}{4}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{3}{4}\cdot\left(\mathrm{ulp}\left(x_h\right)^{+}-\mathrm{ulp}\left(x_h\right)-\mathrm{ulp}\left(x_h\right)^{+}\right)\\ &= -\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)\end{aligned}$$

and, still with the same lemmas,

$$\begin{aligned}\left(-\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)\right)^{-}-\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{4}\left(-3\mathrm{ulp}\left(x_h\right)\right)^{-}-\frac{1}{2}\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{1}{4}\cdot\left(3\cdot\mathrm{ulp}\left(x_h\right)-\left(3\cdot\mathrm{ulp}\left(x_h\right)\right)^{-}-3\cdot\mathrm{ulp}\left(x_h\right)\right)\\ &\quad -\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{1}{4}\cdot\left(\left(3\cdot\mathrm{ulp}\left(x_h\right)\right)^{+}-6\cdot\mathrm{ulp}\left(x_h\right)\right)-\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{1}{4}\cdot\left(3\cdot\mathrm{ulp}\left(x_h\right)\right)^{+}-\frac{3}{2}\cdot\mathrm{ulp}\left(x_h\right)-\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{1}{4}\cdot\left(3\cdot\mathrm{ulp}\left(x_h\right)^{+}-\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\right)\\ &\quad -\frac{3}{2}\cdot\mathrm{ulp}\left(x_h\right)-\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\\ &= \frac{3}{4}\cdot\left(\mathrm{ulp}\left(x_h\right)^{+}-2\cdot\mathrm{ulp}\left(x_h\right)-\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)\right)\\ &= \frac{3}{4}\cdot\left(\mathrm{ulp}\left(x_h\right)^{+}-2\cdot\mathrm{ulp}\left(x_h\right)-\mathrm{ulp}\left(x_h\right)^{+}+\mathrm{ulp}\left(x_h\right)\right)\\ &= -\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)\end{aligned}$$

and finally

$$\begin{aligned}\left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^{+}+\frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{2}\cdot\left(-\mathrm{ulp}\left(x_h\right)^{+}+\mathrm{ulp}\left(x_h\right)^{+}-\mathrm{ulp}\left(x_h\right)\right)\\ &= -\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\end{aligned}$$

For each bound that depends on τ we are obliged to treat two different cases.
Let us suppose first that

$$\tau=\frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right)$$

So we get

$$\begin{aligned}\left(-\frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right)\right)^{-}-\frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{4}\cdot\left(-\mathrm{ulp}\left(x_h\right)^{-}-\frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^{+}-\mathrm{ulp}\left(x_h\right)\right)\right)\\ &= \frac{1}{4}\cdot\left(-\mathrm{ulp}\left(x_h\right)^{-}-\left(\mathrm{ulp}\left(x_h\right)^{-}-\mathrm{ulp}\left(x_h\right)\right)\right)\end{aligned}$$

$$\begin{aligned} &= \frac{1}{4}\cdot\left(-\mathrm{ulp}\left(x_h\right)^- - \mathrm{ulp}\left(x_h\right)^- + \mathrm{ulp}\left(x_h\right)\right) \\ &= \frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right) \end{aligned}$$

and

$$\begin{aligned} \left(-\frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right)\right)^+ + \frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{4}\cdot\left(-\mathrm{ulp}\left(x_h\right)^+ + \mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\ &= -\frac{1}{4}\cdot\mathrm{ulp}\left(x_h\right) \end{aligned}$$

Let us suppose now

$$\tau = \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)$$

We get thus

$$\begin{aligned} \left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^- - \frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right) &= \frac{1}{2}\cdot\left(-\mathrm{ulp}\left(x_h\right)^- - \frac{1}{2}\cdot\left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right)\right) \\ &= \frac{1}{2}\cdot\left(-\mathrm{ulp}\left(x_h\right)^- - \left(\mathrm{ulp}\left(x_h\right)^- - \mathrm{ulp}\left(x_h\right)\right)\right) \\ &= \frac{1}{2}\cdot\left(-\mathrm{ulp}\left(x_h\right)^- - \mathrm{ulp}\left(x_h\right)^- + \mathrm{ulp}\left(x_h\right)\right) \\ &= \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right) \end{aligned}$$

and

$$\begin{aligned} \left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^+ + \frac{1}{2}\cdot\mathrm{ulp}\left(\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right) &= \left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^+ + \frac{1}{4}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) \\ &\leq \left(-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right)^+ + \frac{1}{2}\cdot\mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right) \\ &= -\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right) \end{aligned}$$

Finally, for all cases, we observe the following intervals:

$$x_m + x_l \in \left(J_1 \cup J_2 \cup J_3 \cup J_4 \cup J_5 \cup J_6\right) \setminus U$$

with

$$\begin{aligned} J_1 &= \left[-\mathrm{ulp}\left(x_h\right); -\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right)\right] \\ J_2 &= \left[-\frac{3}{4}\cdot\mathrm{ulp}\left(x_h\right); -\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right] \\ J_3 &= \left[-\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right); -\tau\right] \\ J_4 &= [-\tau; 0] \\ J_5 &= \left[0; \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)\right] \\ J_6 &= \left[\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right); \mathrm{ulp}\left(x_h\right)\right] \end{aligned}$$

Let us now consider more precisely the set U if impossible cases due to the property that $x_m = \circ\left(x_m + x_l\right)$ and due to the fact that $x_l \neq 0$:
Let us show that $\frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right) \in U$, i.e.

$$x_m + x_l \neq \frac{1}{2}\cdot\mathrm{ulp}\left(x_h\right)$$

Let us suppose that this would not be the case. We would get

$$x_m + x_l = \frac{1}{2} \cdot \mathrm{ulp}\,(x_h)$$

As $x_m \neq \frac{1}{2} \cdot \mathrm{ulp}\,(x_h)$ as per hypothesis in this branch of the algorithm and because $x_l \neq 0$, there must exist a number $\mu \in \mathbb{R} \setminus \{0\}$ such that $x_m = \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) + \mu$ and that $x_l = -\mu$.
Since $x_l = \mu$ must hold, μ must be a floating point number. Further

$$|\mu| = |x_l| \leq \frac{1}{2} \cdot \mathrm{ulp}\,(x_m)$$

must be justified. So there exist two floating point numbers $\frac{1}{2} \cdot \mathrm{ulp}\,(x_h)$ and x_m such that their difference verifies

$$\begin{aligned} \left| x_m - \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) \right| &= \left| \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) + \mu - \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) \right| \\ &= |\mu| \\ &\leq \frac{1}{2} \cdot \mathrm{ulp}\,(x_m) \end{aligned}$$

which is possible only if x_m is exactly an integer power of 2. In contrast, as $\frac{1}{2} \cdot \mathrm{ulp}\,(x_h)$ is the only one in the interval that is possible for x_m, which is by the way $\left]\frac{1}{4} \cdot \mathrm{ulp}\,(x_h)\,;\mathrm{ulp}\,(x_h)\right[$, we obtain a contradiction.
Using a completely analogous argument, one sees further that

$$-\tau \in U$$

Clearly $0 \in U$ because $x_l \neq 0$ and it is less in magnitude than x_m.
Let us show finally that $-\mathrm{ulp}\,(x_h) \in U$ and that $\mathrm{ulp}\,(x_h) \in U$.
Let us suppose that we would have

$$|x_m + x_l| = \mathrm{ulp}\,(x_h)$$

In contrast we know that $|x_m| < \mathrm{ulp}\,(x_h)$. Since x_m is a floating point number, this means that

$$|x_m| \leq \mathrm{ulp}\,(x_h)^-$$

which yields to

$$\begin{aligned} |x_l| &\geq \mathrm{ulp}\,(x_h) - \mathrm{ulp}\,(x_h)^- \\ &= \frac{1}{2} \cdot \left(\mathrm{ulp}\,(x_h)^+ - \mathrm{ulp}\,(x_h) \right) \\ &= \frac{1}{2} \cdot \mathrm{ulp}\,(\mathrm{ulp}\,(x_h)) \end{aligned}$$

Further we know that $|x_m| \leq \mathrm{ulp}\,(x_h)^-$ and that $|x_l| \leq \frac{1}{2} \cdot \mathrm{ulp}\,(x_m)$. So we can check that

$$\begin{aligned} |x_l| &\leq \frac{1}{2} \cdot \mathrm{ulp}\,(x_m) \\ &= \frac{1}{2} \cdot \left(\left(\mathrm{ulp}\,(x_h)^- \right)^+ - \mathrm{ulp}\,(x_h)^+ \right) \\ &= \frac{1}{2} \cdot \left(\mathrm{ulp}\,(x_h) - \mathrm{ulp}\,(x_h)^- \right) \\ &= \frac{1}{4} \cdot \mathrm{ulp}\,(\mathrm{ulp}\,(x_h)) \end{aligned}$$

We have thus obtained a contradiction to the hypothesis that says that $-\mathrm{ulp}\,(x_h) \notin U$ and that $\mathrm{ulp}\,(x_h) \notin U$.

Let us still show that in the case where x_h is an integer power of 2, i.e. $\exists e \in \mathbb{Z} \,.\, x_h = 2^e$, $-\frac{3}{4} \cdot \mathrm{ulp}(x_h) \in U$. Since $x_l \neq 0$, using a similar argument as the one given above, the problem can be reduced to showing that $x_m = -\frac{3}{4} \cdot \mathrm{ulp}(x_h)$ is impossible if x_h is an integer power of 2. Let us suppose the contrary. Since x_h is an integer power of 2, its mantissa is even. In consequence the mantissa of x_h^- is odd and the one of x_h^{--} is again even. So $\circ(x_h + x_m) = x_h^{--}$ because $x_h + x_m$ is at the exact middle between x_h^{--} and x_h^- and the mantissa of x_h^{--} is even. It follows that $(x_h + x_m) - \circ(x_h + x_m) = \frac{1}{2} \cdot \mathrm{ulp}(\circ(x_h + x_m))$ which is impossible as per hypothesis.

Having shown which numbers are in U, we can rewrite our intervals as follows

$$x_m + x_l \in J'_1 \cup J'_2 \cup J'_3 \cup J'_4 \cup J'_5 \cup J'_6$$

with

$$\begin{aligned}
J'_1 &= \left]-\mathrm{ulp}(x_h)\,; -\frac{3}{4} \cdot \mathrm{ulp}(x_h)\right] \\
J'_2 &= \left]-\frac{3}{4} \cdot \mathrm{ulp}(x_h)\,; -\frac{1}{2} \cdot \mathrm{ulp}(x_h)\right[\\
J'_3 &= \left]-\frac{1}{2} \cdot \mathrm{ulp}(x_h)\,; -\tau\right[\\
J'_4 &=]-\tau; 0] \\
J'_5 &= \left[0; \frac{1}{2} \cdot \mathrm{ulp}(x_h)\right[\\
J'_6 &= \left]\frac{1}{2} \cdot \mathrm{ulp}(x_h)\,; \mathrm{ulp}(x_h)\right[
\end{aligned}$$

One can trivially check that the rounding $\circ(x_h + (x_m + x_l))$ can be decided without using the rule for even rounding. In particular the cases present themselves as follows [2]:

$$\circ(x_h + (x_m + x_l)) = \begin{cases}
x_h^{--} & \text{if } x_m + x_l \in J'_1 \wedge \exists e \in \mathbb{Z} \,.\, 2^e = x_h \\
x_h^{-} & \text{if } x_m + x_l \in J'_1 \wedge \neg\exists e \in \mathbb{Z} \,.\, 2^e = x_h \\
x_h^{-} & \text{if } x_m + x_l \in J'_2 \\
x_h^{-} & \text{if } x_m + x_l \in J'_3 \\
x_h & \text{if } x_m + x_l \in J'_4 \\
x_h & \text{if } x_m + x_l \in J'_5 \\
x_h^{+} & \text{if } x_m + x_l \in J'_6
\end{cases}$$

which can be compared to

$$\circ(x_h + x_m) = \begin{cases}
x_h^{--} & \text{if } x_m \in I'_1 \wedge \exists e \in \mathbb{Z} \,.\, 2^e = x_h \\
x_h^{-} & \text{if } x_m \in I'_1 \wedge \neg\exists e \in \mathbb{Z} \,.\, 2^e = x_h \\
x_h^{-} & \text{if } x_m \in I'_2 \\
x_h^{-} & \text{if } x_m \in I'_3 \\
x_h & \text{if } x_m \in I'_4 \\
x_h & \text{if } x_m \in I'_5 \\
x_h^{+} & \text{if } x_m \in I'_6
\end{cases}$$

Additionally we check that

$$\forall i = 1, \ldots, 6 \,.\, J'_i \subseteq I'_i$$

We would therefore get an immediate contradiction if we supposed that

$$\circ(x_h + (x_m + x_l)) \neq \circ(x_h + x_m)$$

This finishes the proof for the first case.

Let us consider now subcases a.) through d.) of the second main case of the proof. We have already shown that the subcases h.) through e.) are equal to the first ones. Without loss of generality we will only analyse the case where $x_h > 0$. The set of the floating point numbers as well as the rounding function $\circ(\hat{x})$ are symmetrical around 0. We can therefore suppose that

$$x_m = -\begin{cases} \frac{1}{4} \cdot \mathrm{ulp}(x_h) & \text{if } \exists e \in \mathbb{Z} \,.\, 2^e = x_h \\ \frac{1}{2} \cdot \mathrm{ulp}(x_h) & \text{otherwise} \end{cases}$$

or that

$$x_m = \frac{1}{2} \cdot \mathrm{ulp}(x_h)$$

depending on whether x_m is negative or positive.
It is clear that one can suppose that

$$|x_m + x_l| < \mathrm{ulp}(x_h)$$

because otherwise we would have $|x_l| \geq \frac{1}{2} \cdot \mathrm{ulp}(x_h)$ whilst $|x_l| \leq 2^{-53} \cdot \mathrm{ulp}(x_h)$.
Let us treat now the four cases one after another:

a.) We can suppose in this case that $x_m > 0$ and that $x_l > 0$:
So

$$\mathrm{ulp}(x_h) > x_m + x_l > \frac{1}{2} \cdot \mathrm{ulp}(x_h)$$

Thus since the inequalities are strict

$$\circ(x_h + (x_m + x_l)) = x_h^+ = \mathrm{succ}(x_h)$$

which is the number returned by the algorithm.

b.) We have here $x_m > 0$ and $x_l < 0$:
So the same way, we know that

$$x_m + x_l < \frac{1}{2} \cdot \mathrm{ulp}(x_h)$$

Additionally, we know that $x_l \geq -2^{-53} \cdot \mathrm{ulp}(x_h) > -\frac{1}{4} \cdot \mathrm{ulp}(x_h)$. This yields thus to

$$\circ(x_h + (x_m + x_l)) = x_h$$

The correctness of the algorithm is therefore verified also in this case.

c.) In this case one knows that $x_m < 0$ and $x_l > 0$. In consequence

$$x_m = -\tau$$

with

$$\tau = \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}(x_h) & \text{if } \exists e \in \mathbb{Z} \,.\, 2^e = x_h \\ \frac{1}{2} \cdot \mathrm{ulp}(x_h) & \text{otherwise} \end{cases}$$

Thus we get

$$\frac{1}{4} \cdot \mathrm{ulp}(x_h) > 2^{-53} \cdot \mathrm{ulp}(x_h) > x_m + x_l > -\tau$$

mentioning analogous arguments as the ones given above. This yields to

$$\circ(x_h + (x_m + x_l)) = x_h$$

which is the number returned by the algorithm.

d.) Finally if $x_m < 0$ and $x_l < 0$ one checks that

$$-2 \cdot \tau < x_m + x_l < -\tau$$

The lower bound given for $x_m + x_l$ can be explained as follows. If $\tau = \frac{1}{2} \cdot \mathrm{ulp}(x_h)$, it trivially holds due to the bound:

$$|x_m + x_l| < \mathrm{ulp}(x_h)$$

We have already indicated this bound. Otherwise we know that $\tau = \frac{1}{4} \cdot \mathrm{ulp}(x_h)$ and that $x_m = -\tau$. Since $|x_l| \leq 2^{-53} \cdot |x_m|$, one gets

$$x_m + x_l > -\left(\frac{1}{4} + 2^{-55}\right) \cdot \mathrm{ulp}(x_h) > -2 \cdot \tau$$

Thus the bounds obtained for $x_m + x_l$ imply always that

$$\circ(x_h + (x_m + x_l)) = x_h^- = \mathrm{pred}(x_h)$$

Thus in this subcase, too, and therefore in all cases, the algorithm returns a floating point number x' which is equal to $\circ(x_h + x_m + x_l)$.

By this final statement we have finished the proof. ∎

7.2 Final rounding for the directed modes

As we have already mentioned, the three directed rounding modes behave all in a similar fashion. On the one hand we have

$$\forall \hat{x} \in \mathbb{R} \ . \ \diamond(\hat{x}) = \begin{cases} \nabla(\hat{x}) & \text{if } \hat{x} < 0 \\ \triangle(\hat{x}) & \text{otherwise} \end{cases}$$

On the other hand, one can check that

$$\forall \hat{x} \in \mathbb{R} \ . \ \triangle(\hat{x}) = -\nabla(-\hat{x})$$

The given equations are also verified on the set of the floating point numbers $\mathbb{F}$ [4, 2]. We will therefore refrain from treating each directed rounding mode separately but we will consider them all in common. So slightly deriving from our notations, we will notate $\diamond$ the rounding function of all possible three directed rounding modes.

Further we suppose that we dispose of a correct rounding procedure for each directed rounding mode capable of rounding a double number $x_h + x_l$. This procedure will return in fact $\diamond(x_h + x_l)$ [**?**, 4]. For constructing a correct rounding procedure for triple-double precision, we will try to give a reduction procedure for reducing a triple-double number into a double-double number such that the directed rounding of both triple-double and double-double numbers be equal.

Lemma 7.9 (Directed rounding decision)
Let be $x \in \mathbb{F}$ a floating point number.
Let be $\mu, \nu \in \mathbb{R}$ two real numbers such that $|\mu| < \mathrm{ulp}(x)$ and $|\nu| < \mathrm{ulp}(x)$ and that

$$\mathrm{sgn}(\mu) = \mathrm{sgn}(\nu)$$

So the following equation holds

$$\diamond(x + \mu) = \diamond(x + \nu)$$

Proof
We know by definition of the rounding mode, e.g. by the one of rounding $\triangle$ towards $+\infty$ that

$$\forall y \in \mathbb{F}, \mu \in \mathbb{R}, |\mu| < \mathrm{ulp}(y) \ . \ \triangle(y + \rho) = \begin{cases} \mathrm{succ}(y) & \text{if } \rho > 0 \\ y & \text{otherwise} \end{cases}$$

In fact, the rounding result $\triangle\ (y+\rho)$ is the smallest floating point number greater or equal to $y+\rho$.
Since x is a floating point number and as $\mathrm{pred}\,(x) < x+\mu < \mathrm{succ}\,(x)$ and $\mathrm{pred}\,(x) < x+\mu < \mathrm{succ}\,(x)$ because $|\mu| < \mathrm{ulp}\,(x)$ and $|\mu| < \mathrm{ulp}\,(x)$, supposing that $\diamond\,(x+\mu) \neq \diamond\,(x+\nu)$ would yield to an immediate contradiction. ∎

Lemma 7.10 (Disturbed directed rounding)
Let be $\hat{x} \in \mathbb{R}$ a real number and $x = \circ\,(\hat{x}) \in \mathbb{F}$ the (even) floating point number nearest to $\hat{x}$. Let be $\xi\,(\hat{x}) = \hat{x} - x$.
Let be $\delta \in \mathbb{R}$ such that

$$|\delta| < |\xi\,(\hat{x})|$$

So the following equation holds

$$\diamond\,(\hat{x}) = \diamond\,(\hat{x}+\delta)$$

Let us remark still that the inequality in hypothesis – $|\delta| < |\xi\,(\hat{x})|$ – must be assured to be strict.

Proof
We know already that

$$\diamond\,(\hat{x}+\delta) = \diamond\,(x+\xi\,(\hat{x})+\delta)$$

Let us show now that $\xi\,(\hat{x})$ and $\xi\,(\hat{x})+\delta$ have the same sign. Let us therefore suppose that this would not be the case. Without loss of generality, it suffices to consider the case where $\xi\,(\hat{x})$ is positive; the inverse case can be treated completely analogously.
Thus

$$\xi\,(\hat{x}) \geq 0$$

and

$$\xi\,(\hat{x}) + \delta < 0$$

In consequence

$$\xi\,(\hat{x}) < -\delta$$

On the other hand

$$|\delta| < |\xi\,(\hat{x})|$$

Thus

$$0 \leq \xi\,(\hat{x}) < \xi\,(\hat{x})$$

which yields to

$$\xi\,(\hat{x}) = 0$$

In this case we know that

$$\delta = 0$$

as per the hypotheses of the theorem. Thus contradiction and we know that $\xi\,(\hat{x})$ and $\xi\,(\hat{x})+\delta$ have really the same sign.
It is clear that $\xi\,(\hat{x}) \leq \frac{1}{2} \cdot \mathrm{ulp}\,(x)$ because the rounding of $\hat{x}$ towards x is done to the nearest floating point number. In consequence, since $\delta < \xi\,(\hat{x})$ we obtain

$$\xi\,(\hat{x}) + \delta < \mathrm{ulp}\,(x)$$

As x is a floating point number it suffices thus to conclude using lemma 7.9 by putting $\mu = \xi\,(\hat{x})$ and $\nu = \xi\,(\hat{x}) + \delta$. ∎

Lemma 7.11 (Reduction of a triple-double into a double-double – simple case)
Let be $x_h + x_m + x_l \in \mathbb{F} + \mathbb{F} + \mathbb{F}$ a non-overlapping triple-double number such that x_l is not subnormal, such that $x_m = \circ\,(x_m + x_l)$ and such that $|x_m| < \tau$ where

$$\tau = \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}\,(x_h) & \text{if } \exists e \in \mathbb{Z}\ .\ 2^e = |x_h| \wedge \mathrm{sgn}\,(x_m) = -\mathrm{sgn}\,(x_h) \\ \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) & \text{otherwise} \end{cases}$$

Given the instruction sequence below:

$$(t_1, t_2) \leftarrow \mathbf{Add12}\,(x_h, x_m)$$
$$t_3 \leftarrow t_2 \oplus x_l$$

the following equation holds after the execution of the sequence

$$\diamond\,(t_1 + t_3) = \diamond\,(x_h + x_m + x_l)$$

Proof
Due to the hypothesis that $|x_m| < \tau$ we can suppose that $x_h = \circ\,(x_h + x_m)$. In consequence, using the properties of the **Add12** procedure, we know that we have exactly

$$t_1 = \circ\,(x_h + x_m) = x_h$$

and

$$t_2 = x_h + x_m - t_1 = x_m$$

So since as per hypothesis we have $x_m = \circ\,(x_m + x_l)$, we know also that t_3 verifies exactly

$$t_3 = x_m \oplus x_l = \circ\,(x_m + x_l) = x_m$$

Let us put now

$$\delta = x_l$$

and

$$\hat{x} = t_1 + t_3$$

Clearly we get

$$\begin{aligned}
\xi\,(\hat{x}) &= \hat{x} - \circ\,(\hat{x}) \\
&= t_1 + t_3 - \circ\,(t_1 + t_3) \\
&= x_h + x_m - \circ\,(x_h + x_m) \\
&= x_h + x_m - x_h \\
&= x_m
\end{aligned}$$

Let us show now that

$$|\delta| < |\xi\,(\hat{x})|$$

Amongst other by the lemma's hypotheses and due to the fact that $x_m \neq 0$, we can check that

$$\begin{aligned}
|\xi\,(\hat{x})| &= |x_m| \\
&> 2^{-53} \cdot |x_m| \\
&\geq |x_l|
\end{aligned}$$

The inequality the lemma 7.10 asks for in hypothesis is well verified.
So as per the same lemma 7.10 we know that

$$\diamond\,(\hat{x} + \delta) = \diamond\,(\hat{x})$$

This means that

$$\diamond\,(x_h + x_m + x_l) = \diamond\,(t_1 + t_3)$$

which is the equation that was to be shown. ∎

Lemma 7.12 (Reduction of a triple-double into a double-double – difficult case)
Let be $x_h + x_m + x_l \in \mathbb{F} + \mathbb{F} + \mathbb{F}$ a non-overlapping triple-double number such that x_l is not subnormal, such that $x_m = \circ\,(x_m + x_l)$ and that $|x_m| \geq \tau$ where

$$\tau = \begin{cases} \frac{1}{4} \cdot \mathrm{ulp}\,(x_h) & \text{if } \exists e \in \mathbb{Z}\ .\ 2^e = |x_h| \wedge \mathrm{sgn}\,(x_m) = -\mathrm{sgn}\,(x_h) \\ \frac{1}{2} \cdot \mathrm{ulp}\,(x_h) & \text{otherwise} \end{cases}$$

Given the instruction sequence below

$$(t_1, t_2) \leftarrow \mathbf{Add12}(x_h, x_m)$$
$$t_3 \leftarrow t_2 \oplus x_l$$

the following equation holds after the execution of the sequence

$$\diamond(t_1 + t_3) = \diamond(x_h + x_m + x_l)$$

Proof
Without loss of generality, let us suppose in the sequel that $x_h > 0$. This is legitime because the set of the floating point numbers is symmetrical around 0. In fact it suffices to apply lemma 2.8 and definition 2.6 in order to obtain a proof for each case.
In what follows we will proceed as that: we will decompose the problem in several cases and subcases that we will treat one after another. For each of this subcases we will show either directly the desired result or the fact that $|t_2| \geq |x_l|$. In the end we will prove that this fact yields to the correctness of the lemma in each case.
Let us start by considering the case where $\tau = \frac{1}{2} \cdot \mathrm{ulp}(x_h)$. Thus x_h is not an exact integer power of 2.
We therefore get

$$\frac{1}{2} \leq |x_m| < \mathrm{ulp}(x_h)$$

which is equivalent to

$$\frac{1}{2} \leq |x_m| \leq \mathrm{ulp}(x_h)^-$$

because x_m is a floating point number.
We can check now that

$$t_1 = \circ(x_h + x_m) = \begin{cases} x_h^+ & \text{if } x_m > 0 \\ x_h & \text{if } x_m = \frac{1}{2} \cdot \mathrm{ulp}(x_h) \text{ and the mantissa of } x_h \text{ is even} \\ x_h^- & \text{if } x_m < 0 \end{cases}$$

This implies the handling of three different subcases.
Let as treat first the case where $t_1 = x_h$:
We get here

$$\begin{aligned} t_2 &= x_h i + x_m - t_1 \\ &= x_h + x_m - x_h \\ &= x_m \end{aligned}$$

and further

$$\begin{aligned} t_3 &= t_2 \oplus x_l \\ &= x_m \oplus x_l \\ &= \circ(x_m + x_l) \\ &= x_m \end{aligned}$$

as per the hypothesis on the arguments.
So let us put

$$\begin{aligned} \hat{x} &= t_1 + t_3 \\ \delta &= x_l \end{aligned}$$

Thus

$$\begin{aligned} \hat{x} &= x_h + x_m \\ \xi(\hat{x}) &= \circ(\hat{x}) \\ &= \circ(t_1 + t_2) \\ &= t_2 \\ &= t_3 \end{aligned}$$

and

$$|\delta| < |\xi(\hat{x})|$$

because

$$|x_l| \leq 2^{-53} \cdot |x_m|$$

et

$$x_m \neq 0$$

Applying lemma 7.10 we thus obtain

$$\diamond(t_1 + t_3) = \diamond(x_h + x_m + x_l)$$

Let us continue with the case where $t_1 = x_h^+$:
We get here

$$\begin{aligned} t_2 &= x_h + x_m + t_1 \\ &= x_h + x_m - x_h^+ \\ &= -\left(x_h^+ - x_h\right) + x_m \\ &= -\text{ulp}(x_h) + x_m \\ &\leq -\text{ulp}(x_h) + \text{ulp}(x_h)^- \\ &= -\left(\text{ulp}(x_h) - \text{ulp}(x_h)^-\right) \\ &= -\frac{1}{2} \cdot \left(\text{ulp}(x_h)^+ - \text{ulp}(x_h)\right) \\ &= -\frac{1}{2} \cdot \text{ulp}(\text{ulp}(x_h)) \end{aligned}$$

Thus

$$|t_2| \geq \frac{1}{2} \cdot \text{ulp}(\text{ulp}(x_h))$$

In contrast $|x_l| \leq \frac{1}{2} \cdot \text{ulp}(\text{ulp}(x_h))$ as per hypothesis which implies

$$|t_2| \geq |x_l|$$

Let us finally check the properties to be show for the third and last subcase, supposing now that $t_1 = x_h^-$.
Since x_h is not exactly an integer power of 2, we can check the following applying amongst other lemma 2.10:

$$\begin{aligned} t_2 &= x_h + x_m - t_1 \\ &= x_h + x_m - x_h^- \\ &= x_h^+ - x_h + x_m \\ &= \text{ulp}(x_h) + x_m \\ &\geq \text{ulp}(x_h) - \text{ulp}(x_h)^- \\ &= \frac{1}{2} \cdot \text{ulp}(\text{ulp}(x_h)) \end{aligned}$$

We therefore still get

$$|t_2| \geq |x_l|$$

using the same argument as the one given above.

Let us handle now the case where $\tau = \frac{1}{4} \cdot \text{ulp}(x_h)$:
We can suppose in this case that x_h is an exact integer power of 2 and that x_m is negative because

we had already supposed that x_h is positive and because we know that $\operatorname{sgn}(x_h) = -\operatorname{sgn}(x_m)$. We get further the following bounds for x_m:

$$\frac{1}{4} \cdot \operatorname{ulp}(x_h) \leq |x_m| \leq \operatorname{ulp}(x_h)^-$$

which means that

$$-\operatorname{ulp}(x_h)^- \leq x_m \leq -\frac{1}{4} \cdot \operatorname{ulp}(x_h)$$

still because x_m is a floating point number.
Since x_h is an integer power of 2 and as for this reason, its mantissa is even, one can check that

$$t_1 = \circ(x_h + x_m) = \begin{cases} x_h & \text{if } x_m = -\frac{1}{4} \cdot \operatorname{ulp}(x_h) \\ x_h^- & \text{if } -\frac{1}{2} \cdot \operatorname{ulp}(x_h) < x_m < -\frac{1}{4} \cdot \operatorname{ulp}(x_h) \\ x_h^- & \text{if } x_m = -\frac{1}{2} \cdot \operatorname{ulp}(x_h) \\ x_h^- & \text{if } -\frac{3}{4} \cdot \operatorname{ulp}(x_h) < x_m < -\frac{1}{2} \cdot \operatorname{ulp}(x_h) \\ x_h^{--} & \text{if } -\operatorname{ulp}(x_h)^- \leq x_m \leq -\frac{3}{4} \cdot \operatorname{ulp}(x_h) \end{cases}$$

The assertion that $t_1 = x_h^-$ if $x_m = -\frac{1}{2} \cdot \operatorname{ulp}(x_h)$ can be explained as follows:
We have

$$\begin{aligned} x_h + x_m &= x_h - \frac{1}{2} \cdot \operatorname{ulp}(x_h) \\ &= x_h - \frac{1}{2} \cdot \left(x_h^+ - x_h\right) \\ &= x_h - \left(x_h - x_h^-\right) \\ &= x_h^- \end{aligned}$$

Thus

$$\circ(x_h + x_m) = \circ\left(x_h^-\right) = x_h^-$$

because x_h^- is clearly a floating point number. Let us consider now first the cases where we have equations, i.e. the cases where $x_m = -\frac{1}{4} \cdot \operatorname{ulp}(x_h)$ and $x_m = -\frac{1}{2} \cdot \operatorname{ulp}(x_h)$:
Let us commence by the case where $x_m = -\frac{1}{4} \cdot \operatorname{ulp}(x_h)$:
We get here

$$\begin{aligned} t_2 &= x_h + x_m - t_1 \\ &= x_h + x_m - x_h \\ &= x_m \end{aligned}$$

and we can check that the following holds by the hypotheses on the arguments

$$\begin{aligned} t_3 &= x_m \oplus x_l \\ &= \circ(x_m + x_l) \\ &= x_m \end{aligned}$$

Let us put now

$$\begin{aligned} \hat{x} &= t_1 + t_3 \\ \delta &= x_l \\ \xi(\hat{x}) &= \circ(\hat{x}) - \hat{x} = t_3 \end{aligned}$$

So by applying lemma 7.10, we get

$$\diamond(t_1 + t_3) = \diamond(x_h + x_m + x_l)$$

because

$$|x_l| \leq 2^{-53} \cdot |x_m|$$

and $x_m \neq 0$ which is a hypothesis of the lemma to prove.
Let us now handle the second of these particular cases, i.e. the cases where $x_m = -\frac{1}{2} \cdot \mathrm{ulp}(x_h)$: We get

$$\begin{aligned} t_2 &= x_h + x_m - t_1 \\ &= x_h + x_m - x_h^- \\ &= \frac{1}{2} \cdot \mathrm{ulp}(x_h) + x_m \\ &= \frac{1}{2} \cdot \mathrm{ulp}(x_h) - \frac{1}{2} \cdot \mathrm{ulp}(x_h) \\ &= 0 \end{aligned}$$

So in consequence we have

$$\begin{aligned} t_3 &= t_2 \oplus x_l \\ &= 0 \oplus x_l \\ &= x_l \end{aligned}$$

And we thus obtain finally

$$\begin{aligned} \diamond(x_h + x_m + x_l) &= \diamond(t_1 + t_2 + x_l) \\ &= \diamond(t_1 + 0 + x_l) \\ &= \diamond(t_1 + t_3) \end{aligned}$$

Let us now analyse the other principal cases, starting with the case where

$$-\frac{1}{2} \cdot \mathrm{ulp}(x_h) < x_m < -\frac{1}{4} \cdot \mathrm{ulp}(x_h)$$

This inequality bounding x_m is in fact equivalent to the following one because x_m is a floating point number:

$$-\frac{1}{2} \cdot \mathrm{ulp}(x_h)^- \leq x_m \leq -\frac{1}{4} \cdot \mathrm{ulp}(x_h)^+$$

So we can check

$$\begin{aligned} t_2 &= x_h + x_m - t_1 \\ &= x_h + x_m - x_h^- \\ &= \frac{1}{2} \cdot \mathrm{ulp}(x_h) + x_m \\ &\geq \frac{1}{2} \cdot \mathrm{ulp}(x_h) - \frac{1}{2} \cdot \mathrm{ulp}(x_h)^- \\ &= \frac{1}{2} \cdot \left(\mathrm{ulp}(x_h) - \mathrm{ulp}(x_h)^-\right) \\ &= \frac{1}{4}\mathrm{ulp}(\mathrm{ulp}(x_h)) \end{aligned}$$

In contrast we know that

$$|x_l| \leq \frac{1}{2} \cdot \mathrm{ulp}(x_m)$$

Since in the currently treated case the following holds

$$|x_m| \leq \frac{1}{2} \cdot \mathrm{ulp}(x_h)$$

we get as per lemma 2.11

$$|x_l| \leq \frac{1}{4} \cdot \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)$$

which yields to

$$|t_2| \geq |x_l|$$

Let us now consider the second and one but not least case. We suppose here that

$$-\frac{3}{4} \cdot \mathrm{ulp}\left(x_h\right) < x_m < -\frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right)$$

which is equivalent to

$$-\left(\frac{3}{4} \cdot \mathrm{ulp}\left(x_h\right)\right)^- \leq x_m < -\frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right)^+$$

We therefore get

$$\begin{aligned}
t_2 &= x_h + x_m - t_1 \\
&= x_h + x_m - x_h^- \\
&= \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) + x_m \\
&\leq \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) - \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right)^+ \\
&= -\frac{1}{2} \cdot \left(\mathrm{ulp}\left(x_h\right)^+ - \mathrm{ulp}\left(x_h\right)\right) \\
&= -\frac{1}{2} \cdot \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)
\end{aligned}$$

which gives us

$$|t_2| \geq \frac{1}{2} \cdot \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)$$

We can deduce from that, still using the argument that $|x_l| \leq \frac{1}{2} \cdot \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)$, that

$$|t_2| \geq |x_l|$$

Let us finally handle the last case where $-\mathrm{ulp}\left(x_h\right)^- \leq x_m \leq -\frac{3}{4} \cdot \mathrm{ulp}\left(x_h\right)$:
Using the property that x_h^- is an exact integer power of 2 and using further lemma 2.10, we can check now that

$$\begin{aligned}
t_2 &= x_h + x_m - t_1 \\
&= x_h + x_m - x_h^{--} \\
&= x_h + x_m - x_h^{--} + x_h^- - x_h^- \\
&= x_h - x_h^- + x_m + x_h^- - x_h^{--} \\
&= \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) + x_m + \left(x_h^-\right)^+ - x_h^- \\
&= \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) + x_m + x_h - x_h^- \\
&= \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) + x_m + \frac{1}{2} \cdot \mathrm{ulp}\left(x_h\right) \\
&= \mathrm{ulp}\left(x_h\right) + x_m \\
&\geq \mathrm{ulp}\left(x_h\right) + \mathrm{ulp}\left(x_h\right)^- \\
&= \frac{1}{2} \cdot \mathrm{ulp}\left(\mathrm{ulp}\left(x_h\right)\right)
\end{aligned}$$

In consequence we still get the same upper bound for $|x_l|$, i.e.

$$|t_2| \geq |x_l|$$

Since we have now treated all the cases that have been showing up, it suffices now to show that the upper bound already mentioned yields to the property to be proven. Once again, we decompose the problem in cases and subcases.
Let us start by showing the property for the equation $|t_2| = |x_l|$:
If $\mathrm{sgn}(t_2) = \mathrm{sgn}(x_l)$ we get

$$\begin{aligned} t_3 &= t_2 \oplus x_l \\ &= x_l \oplus x_l \\ &= \circ(2 \cdot x_l) \\ &= 2 \cdot x_l \end{aligned}$$

So we have exactly

$$t_1 + t_3 = x_h + x_m + x_l$$

and thus

$$\diamond(t_1 + t_3) = \diamond(x_h + x_m + xl)$$

Otherwise, we have $\mathrm{sgn}(t_2) = -\mathrm{sgn}(x_l)$ and get

$$\begin{aligned} t_3 &= t_2 \oplus x_l \\ &= -x_l \oplus x_l \\ &= 0 \end{aligned}$$

exactly. This means finally that

$$\begin{aligned} \diamond(t_1 + t_3) &= \diamond(t_1) \\ &= \diamond(t_1 + t_2 - t_2) \\ &= \diamond(x_h + x_m + x_l) \end{aligned}$$

In the end let us consider the case where one can suppose that $|t_2| > |x_l|$:
We can suppose here

$$\begin{aligned} t_3 &= t_2 \oplus x_l \\ &= t_2 + x_l + \delta \end{aligned}$$

with

$$|\delta| \leq 2^{-53} \cdot |t_2 + x_l|$$

Let us show now that t_2 and t_3 have the same sign. For doing so let us suppose that this would not be true.
Clearly, t_2 and $t_2 + x_l$ have the same sign because we know that $|t_2| > |x_l|$.
So in order to have $\mathrm{sgn}(t_2) = -1 \cdot \mathrm{sgn}(t_3)$ to hold, we must have

$$|\delta| > |t_2 + x_l|$$

Thus we would obtain

$$2^{-53} \cdot |t_2 + x_l| > |t_2 + x_l|$$

which is not true because

$$t_2 + x_l = 0$$

This would yield to $|t_2| = |x_l|$ which is excluded by hypotheses. Thus, contradiction.
The values t_2 and t_3 have therefore the same sign. By applying lemma 7.9, we get:

$$\diamond(t_1 + t_2) = \diamond(t_1 + t_3)$$

Let us show now that

$$\diamond(t_1 + t_2) = \diamond(x_h + x_m + x_l)$$

in order to be able to conclude.
For doing so, let us put

$$\begin{aligned} \hat{x} &= t_1 + t_2 \\ \delta' &= x_l \\ \xi(\hat{x}) &= t_2 \end{aligned}$$

and let us check that

$$\begin{aligned} |\delta'| &= |x_l| \\ &< |t_2| \\ &= |\xi(\hat{x})| \end{aligned}$$

We can now apply lemma 7.10 and obtain:

$$\diamond(t_1 + t_3) = \diamond(x_h + x_m + x_l)$$

This is the equation to be shown. ∎

Theorem 7.13 (Directed final rounding of a triple-double number)
Let be $x_h + x_m + x_l \in \mathbb{F} + \mathbb{F} + \mathbb{F}$ *a non-overlapping triple-double number.*
Let be $\diamond$ *a directed rounding mode.*
Let be **A** *the following instruction sequence:*

$$\begin{array}{l} (t_1, t_2) \leftarrow \mathbf{Add12}(x_h, x_m) \\ t_3 \leftarrow t_2 \oplus x_l \\ \mathbf{return} \quad \diamond(t_1 + t_3) \end{array}$$

So **A** *is a correct rounding procedure for the rounding mode* $\diamond$.

Proof
Trivial as per lemmas 7.11 and 7.12. ∎

References

[1] CR-Libm, a library of correctly rounded elementary functions in double-precision. http://lipforge.ens-lyon.fr/www/crlibm/.

[2] ANSI/IEEE. Standard 754-1985 for binary floating-point arithmetic (also iec 60559), 1985.

[3] F. de Dinechin, D. Defour, and C. Lauter. Fast correct rounding of elementary functions in double precision using double-extended arithmetic. Technical Report 2004-10, LIP, École Normale Supérieure de Lyon, March 2004.

[4] David Defour. *Fonctions élémentaires: algorithmes et implémentations efficaces pour l'arrondi correct en double précision*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, September 2003.

[5] Theodorus J. Dekker. A floating point technique for extending the available precision. *Numerische Mathematik*, 18(3):224–242, 1971.

[6] Claire Finot-Moreau. *Preuves et algorithmes utilisant l'arithmétique flottante normalisée IEEE*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France, July 2001.

[7] J.-M. Muller. *Elementary Functions, Algorithms and Implementation*. Birkhauser, Boston, 1997.

[8] Jean-Michel Muller. On the definition of ulp(x). Technical Report RR 5504, INRIA, February 2005. Available at ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-5504.pdf.

[9] D. M. Priest. Algorithms for arbitrary precision floating point arithmetic. In *Proceedings of the 10th IEEE Symposium on Computer Arithmetic (Arith-10)*, pages 132–144, Grenoble, France, June 1991. IEEE Computer Society Press.

[10] Jonathan R. Shewchuk. Adaptive precision floating-point arithmetic and fast robust geometric predicates. In *Discrete and Computational Geometry*, volume 18, pages 305–363, 1997.

[11] P. H. Sterbenz. *Floating point computation.* Prentice-Hall, Englewood Cliffs, NJ, 1974.