On-Line Multiplication in Real and Complex Base

Christiane Frougny

LIAFA, CNRS and Université Paris 8

http://www.liafa.jussieu.fr/~cf/

Athasit Surarerks

Chulalongkorn University, Bangkok

http://www.cp.eng.chula.ac.th/faculty/athasit.html

On-line computability

To pipe-line additions/subtractions, multiplications and divisions, computations are to be done Most Sigificant Digit First, *i.e.* from left to right.

Additional requirement: deterministic processing and, after a certain delay δ of latency, for one input digit there is one output digit.

To generate the jth digit of the result, it is necessary and sufficient to have the first $(j + \delta)$ digits of the input available.

[Ercegovac and Trivedi, 77]

A and B two finite digit sets, $A^{\mathbb{N}}$ set of infinite sequences of elements of A.

$$\begin{aligned} \varphi : A^{\mathbb{N}} &\rightarrow B^{\mathbb{N}} \\ (a_j)_{j\geq 1} &\mapsto (b_j)_{j\geq 1} \end{aligned}$$

φ is on-line computable with delay δ if there exists δ such that, for each $j \geq 1$ there exists

$$\Phi_j : A^{j+\delta} \rightarrow B$$

such that

$$b_j = \Phi_j(a_1 \cdots a_{j+\delta})$$

$A^{j+\delta}$ is the set of sequences of length $j + \delta$ of elements of A.

Multiplication

D a finite digit set. Multiplication is on-line computable with delay δ in base β on D if there exists a function

$$\begin{aligned} \mu : D^{\mathbb{N}} \times D^{\mathbb{N}} &\to D^{\mathbb{N}} \\ ((x_j)_{j\geq 1}, (y_j)_{j\geq 1}) &\mapsto (p_j)_{j\geq 1} \end{aligned}$$

such that

$$\sum_{j\geq 1} p_j \beta^{-j} = \sum_{j\geq 1} x_j \beta^{-j} \times \sum_{j\geq 1} y_j \beta^{-j}$$

which is on-line computable with delay δ.

In the following, the operands begin with a run of δ zeroes. This allows to ignore the delay inside the computation.

Beta-Representations

D a finite digit set of real or complex digits.

Base β a real or complex number such that $|\beta| > 1$.

β-representation on D of x real or complex is a sequence $(x_j)_{j\geq 1}$ with $x_j \in D$ such that

$$x = \sum_{j\geq 1} x_j \beta^{-j}$$

Signed-digit number system

Base β integer > 1

signed-digit set $S = \{-a, \ldots, a\}$, $\beta/2 \leq a \leq \beta - 1$.

Redundancy

Addition can be performed in constant time in parallel, and is computable by an on-line finite automaton

[Avizienis 1961, Chow and Robertson 1978, Muller 1994]

Negative base numeration system

Base β a negative integer < -1,
canonical digit set $A = \{0, \ldots, |\beta| - 1\}$.

On a signed-digit set $T = \{-a, \ldots, a\}$, with $|\beta|/2 \leq a \leq |\beta| - 1$ the representation is redundant and addition can be performed in constant time in parallel, and is computable by an on-line finite automaton [Frougny 1999]

Representation in real base

Base β a real number > 1.

$x \in [0,1]$ can be represented in base β by a greedy algorithm [Rényi 1957]:
$r_0 = x$ and for $j \geq 1$ let $x_j = \lfloor \beta r_{j-1} \rfloor$ and $r_j = \{\beta r_{j-1}\}$. Thus $x = \sum_{j \geq 1} x_j \beta^{-j}$.

x_j is in the canonical digit set $A_\beta = \{0, \ldots, \lfloor \beta \rfloor\}$ if $\beta \notin \mathbb{N}$, $A_\beta = \{0, \ldots, \beta - 1\}$ if $\beta \in \mathbb{N}$.

When $\beta \notin \mathbb{N}$, x may have several different β-representations on A_β : this system is naturally redundant.

Example $\beta = \frac{1+\sqrt{5}}{2}$, $A_\beta = \{0,1\}$.

$$\begin{aligned} 3-\sqrt{5} &=_\beta 10010^\omega \\ &=_\beta 01110^\omega \\ &=_\beta 100(01)^\omega \end{aligned}$$

Addition in real base is on-line computable.

A Pisot number is an algebraic integer > 1 such that all its algebraic conjugates are less than 1 in modulus.

The natural integers and the golden ratio are Pisot numbers.

If β is a Pisot number addition is computable by an on-line finite state automaton [Frougny 2001]

Knuth number system

Base β a complex number of the form $\beta = i\sqrt{r}$, r an integer ≥ 2.

Canonical digit set $A = \{0, \ldots, r-1\}$.

Since $\beta^2 = -r$

$$z = \sum_{j \geq 1} a_j \beta^{-j} = \sum_{k \geq 1} a_{2k}(-r)^{-k} + i\sqrt{r} \sum_{k \geq 0} a_{2k+1}(-r)^{-k-1}$$

$$\Re(z) = x = \sum_{k \geq 1} a_{2k}(-r)^{-k}$$

$$\Im(z) = y = \sqrt{r} \sum_{k \geq 0} a_{2k+1}(-r)^{-k-1}$$

The β-representation of z can be obtained by intertwinning the $(-r)$-representation of x and the $(-r)$-representation of $y/\sqrt{r}$.

Signed-digit set $R = \{-a, \ldots, a\}$, $r/2 \leq a \leq r-1$: redundancy, addition is computable in constant time in parallel [Nielsen and Muller 1996, McIlhenny and Ercegovac 1998, McIlhenny 2002]

Addition is computable by an on-line finite state automaton [Frougny 1999]

Classical on-line multiplication algorithm

[Trivedi and Ercegovac 1977]

Multiplication of two numbers represented in integer base $\beta > 1$ with digits in $S = \{-a, \ldots, a\}$, $\beta/2 \leq a \leq \beta - 1$, is computable by an on-line algorithm with delay δ, where δ is the smallest positive integer such that

$$\frac{\beta}{2} + \frac{2a^2}{\beta^\delta(\beta - 1)} \leq a + \frac{1}{2}.$$

If $\beta = 2$ and $a = 1$, $\delta = 2$.
If $\beta = 3$ and $a = 2$, $\delta = 2$.
If $\beta = 2a \geq 4$ then $\delta = 2$.
If $\beta \geq 4$ and if $a \geq \lfloor \beta/2 \rfloor + 1$, $\delta = 1$.

Classical on-line multiplication algorithm

Input: $x = (x_j)_{j \geq 1}$ and $y = (y_j)_{j \geq 1}$ in $S^{\mathbb{N}}$ such that $x_1 = \cdots = x_\delta = 0$ and $y_1 = \cdots = y_\delta = 0$.

Output: $p = (p_j)_{j \geq 1}$ in $S^{\mathbb{N}}$ such that $\sum_{j \geq 1} p_j \beta^{-j} = \sum_{j \geq 1} x_j \beta^{-j} \times \sum_{j \geq 1} y_j \beta^{-j}$.

begin

$p_1 \leftarrow 0, \ldots, p_\delta \leftarrow 0$

$W_\delta \leftarrow 0$

$j \leftarrow \delta + 1$

while $j \geq \delta + 1$ **do**

$\{W_j \leftarrow \beta(W_{j-1} - p_{j-1}) + y_j X_j + x_j Y_{j-1}$

$p_j \leftarrow \text{round}(W_j)$

$j \leftarrow j + 1\}$

end

$X_j = \sum_{1 \leq i \leq j} x_i \beta^{-i}$.

For $n \geq \delta$, $X_n Y_n - P_n = \beta^{-n}(W_n - p_n)$

$|W_n - p_n| \leq \frac{1}{2}$,

$$|X_n Y_n - P_n| \leq \frac{\beta^{-n}}{2}$$

and the algorithm is convergent.

The sequence $p_1 \cdots p_n$ is a β-representation of the most significant half of the product $X_n Y_n$.

Digits p_j's are in digit set S if $|W_j| \leq a + \frac{1}{2}$.

Line 5 and $|X_j| < \frac{a}{\beta^\delta(\beta-1)}$ and $|Y_{j-1}| < \frac{a}{\beta^\delta(\beta-1)}$ imply that

$$|W_j| < \frac{\beta}{2} + \frac{2a^2}{\beta^\delta(\beta-1)} \leq a + \frac{1}{2}$$

by hypothesis on delay δ.

On-line multiplication algorithm in negative base

Multiplication of two numbers represented in negative base $\beta < -1$ and digit set $T = \{-a, \ldots, a\}$, $|\beta|/2 \leq a \leq |\beta| - 1$, is computable by the classical on-line algorithm with delay δ, where δ is the smallest positive integer such that

$$\frac{|\beta|}{2} + \frac{2a^2}{|\beta|^\delta(|\beta| - 1)} \leq a + \frac{1}{2}.$$

On-line multiplication algorithm in real base

$D = \{0, \ldots, d\}$ with $d \geq \lfloor \beta \rfloor$.

Multiplication of two numbers represented in base β with digits in D is computable by an on-line algorithm with delay δ, where δ is the smallest positive integer such that

$$\beta + \frac{2d^2}{\beta^\delta(\beta - 1)} \leq d + 1.$$

Real base on-line multiplication algorithm

Input: $x = (x_j)_{j \geq 1}$ and $y = (y_j)_{j \geq 1}$ in $D^{\mathbb{N}}$ such that $x_1 = \cdots = x_\delta = 0$ and $y_1 = \cdots = y_\delta = 0$.

Output: $p = (p_j)_{j \geq 1}$ in $D^{\mathbb{N}}$ such that $\sum_{j \geq 1} p_j \beta^{-j} = \sum_{j \geq 1} x_j \beta^{-j} \times \sum_{j \geq 1} y_j \beta^{-j}$.

begin

$p_1 \leftarrow 0, \ldots, p_\delta \leftarrow 0$

$W_\delta \leftarrow 0$

$j \leftarrow \delta + 1$

while $j \geq \delta + 1$ **do**

$\{W_j \leftarrow \beta(W_{j-1} - p_{j-1}) + y_j X_j + x_j Y_{j-1}$

$\quad p_j \leftarrow \lfloor W_j \rceil$

$\quad j \leftarrow j + 1\}$

end

Example $\beta = \frac{1+\sqrt{5}}{2}$. Multiplication on $\{0,1\}$ is on-line computable with delay $\delta = 5$.
$x = y = .0^5 10101$, $x \times y = p = .0^{10}101000100001$

j	$(W_j)_{\frac{1+\sqrt{5}}{2}}$	p_j
6	.000001	0
7	.00001	0
8	.0010001001	0
9	.010001001	0
10	.101000100001	0
11	1.01000100001	1
12	.1000100001	0
13	1.000100001	1
14	.00100001	0
15	.0100001	0
16	.100001	0
17	1.00001	1
18	.0001	0
19	.001	0
20	.01	0
21	.1	0
22	1.0	1

Application to carry-save representation

Carry-save representation : β an integer > 1, and digit set $D = \{0, \ldots, \beta\}$.

Redundancy.

Real base on-line multiplication algorithm :

$\beta = 2$ on $\{0, 1, 2\}$, delay $\delta = 3$.

$\beta \geq 3$, on $D = \{0, \ldots, \beta\}$, delay $\delta = 2$.

Internal additions and multiplications by a digit can be performed in parallel.

On-line multiplication algorithm in the Knuth number system

Multiplication of two complex numbers represented in base $\beta = i\sqrt{r}$, with r an integer ≥ 2, and digit set $R = \{-a, \ldots, a\}$, $r/2 \leq a \leq r-1$, is computable by an on-line algorithm with delay δ, where δ is the smallest odd integer such that

$$\frac{r}{2} + \frac{4a^2}{r^{\frac{\delta-1}{2}}(r-1)} \leq a + \frac{1}{2}. \qquad (1)$$

If $r = 2$ and $a = 1$, $\delta = 7$.
If $r = 8$ or $r = 9$ and $a = r - 1$, $\delta = 3$.
If $r = 10$ and $a \geq 7$, $\delta = 3$.
In the other cases, for $r \leq 10$ the delay is $\delta = 5$.

Complex base on-line multiplication algorithm

Input: $x = (x_j)_{j\geq 1}$ and $y = (y_j)_{j\geq 1}$ in $R^{\mathbb{N}}$ such that $x_1 = \cdots = x_\delta = 0$ and $y_1 = \cdots = y_\delta = 0$.

Output: $p = (p_j)_{j\geq 1}$ in $R^{\mathbb{N}}$ such that $\sum_{j\geq 1} p_j \beta^{-j} = \sum_{j\geq 1} x_j \beta^{-j} \times \sum_{j\geq 1} y_j \beta^{-j}$.

begin

1. $p_1 \leftarrow 0, \ldots, p_\delta \leftarrow 0$
2. $W_\delta \leftarrow 0$
3. $j \leftarrow \delta + 1$
4. **while** $j \geq \delta + 1$ **do**
5. $\{W_j \leftarrow \beta(W_{j-1} - p_{j-1}) + y_j X_j + x_j Y_{j-1}$
6. $p_j \leftarrow \text{sign}\,(\Re(W_j)) \lfloor |\Re(W_j)| + \frac{1}{2} \rfloor$
7. $j \leftarrow j + 1\}$

end

Digit p_j is in R if $|\Re(W_j)| < a + \frac{1}{2}$.

By Line 6

$$\Re(|W_j - p_j|) \leq \frac{1}{2} \text{ and } \Im(W_j - p_j) = \Im(W_j).$$

By Line 5

$$|\Re(W_j)| \leq \sqrt{r}|\Im(W_{j-1})| + a(|\Re(X_j) + \Re(Y_{j-1})|)$$

and

$$|\Im(W_j)| \leq \frac{\sqrt{r}}{2} + a(|\Im(X_j) + \Im(Y_{j-1})|).$$

Suppose that δ is odd. Then

$$|\Re(X_j)| < \frac{a}{r^{\frac{\delta-1}{2}}(r-1)} \text{ and } |\Im(X_j)| < \sqrt{r}\frac{a}{r^{\frac{\delta+1}{2}}(r-1)}$$

and the same holds true for Y_{j-1}.

Thus

$$|\Re(W_j)| \leq \frac{r}{2} + \frac{4a^2}{r^{\frac{\delta-1}{2}}(r-1)} < a + \frac{1}{2}.$$

Suppose now that a better even delay δ' could be achieved. Then

$$|\Re(X_j)| < \frac{a}{r^{\frac{\delta'}{2}}(r-1)} \quad \text{and} \quad |\Im(X_j)| < \sqrt{r}\frac{a}{r^{\frac{\delta'}{2}}(r-1)}$$

thus

$$|\Re(W_j)| < \frac{r}{2} + \frac{2a^2(r+1)}{r^{\frac{\delta'}{2}}(r-1)}.$$

This delay will work if

$$\frac{r}{2} + \frac{2a^2(r+1)}{r^{\frac{\delta'}{2}}(r-1)} \leq a + \frac{1}{2}. \tag{2}$$

Suppose that the delay in (1) is of the form $\delta = 2k + 1$ and the delay in (2) is of the form $\delta' = 2k'$, and set

$$C = \frac{(r-1)(2a+1-r)}{4a^2}.$$

Then k is the smallest positive integer such that

$$k > \frac{\log(2/C)}{\log(r)}$$

and k' is the smallest positive integer such that

$$k' > \frac{\log((r+1)/C)}{\log(r)}$$

and obviously $k < k'$.

For $n \geq \delta$, $X_nY_n - P_n = \beta^{-n}(W_n - p_n)$

$$|\Re(W_n - p_n)| \leq 1/2$$

and

$$|\Im(W_n - p_n)| = |\Im(W_n)| \leq \frac{\sqrt{r}}{2} + \sqrt{r}\frac{2a^2}{r^{\frac{\delta+1}{2}}(r-1)}$$

thus the algorithm is convergent, and $p_1 \cdots p_n$ is a β-representation of the most significant half of X_nY_n.

Example $\beta = 2i$ and $R = \{\bar{2}, \bar{1}, 0, 1, 2\}$. $\delta = 5$.
$x = .0^5 1\bar{2}0\bar{1}201$ and $y = .0^5 1\bar{1}00121$.
$x \times y = p = .0^{10} 1111\bar{1}1\bar{1}2\bar{1}\bar{1}\ldots$

j	$(W_j)_{2i}$	p_j
6	$.000001$	0
7	$.0001112$	0
8	$.001112$	0
9	$.01112\bar{1}1$	0
10	$.1111000\bar{1}2$	0
11	$1.1110120\bar{2}$	1
12	$1.11\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
13	$1.1\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
14	$1.\bar{1}1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
15	$\bar{1}.1\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
16	$1.\bar{1}2\bar{1}\bar{1}\bar{1}\bar{1}21$	1
17	$\bar{1}.2\bar{1}\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
18	$2.\bar{1}\bar{1}\bar{1}\bar{1}21$	2
19	$\bar{1}.\bar{1}\bar{1}\bar{1}21$	$\bar{1}$
20	$\bar{1}.\bar{1}\bar{1}21$	$\bar{1}$