Conversionless Mathematical Operations on Roman Numerals

Gustek

October 17, 2025

I. Introduction

This paper presents algorithms to compute the result of basic mathematical operations on Roman numerals without converting them to (and then from) base 10 numbers.

II. Addition

Let us define a function $M = \{(\text{IV}, \text{IIII}); (\text{IX}, \text{VIIII}); (\text{XL}, \text{XXXX}); (\text{XC}, \text{LXXXX}); (\text{CD}, \text{CCCC}); (\text{CM},\text{DCCCC}))$ and $E = (\text{IIIII}, \text{V}); (\text{VV}, \text{X}); (\text{XXXXX}, \text{L}); (\text{LL}, \text{C}); (\text{CCCCC}, \text{D}); (\text{DD}, \text{M})\}$ with $M(x) = x$ and $E(x) = x$ for all other values. $\tilde{M}$ is defined as $\tilde{M}(x) = x$ if $M(x) = x$ and $\tilde{M}(x) = \tilde{M}(M(x))$ else. $\tilde{E}$ is defined in a similar fashion.

The algorithm to compute $z = x + y$ is as follows:

1 $x' \leftarrow \tilde{M}(x)$
2 $y' \leftarrow \tilde{M}(y)$
3 $z' \leftarrow group(cat(x, y))$
4 $z \leftarrow \tilde{M}^{-1}(\tilde{E}(z'))$

Where cat is the concatenation function and $group$ groups individual components of a number together. We describe the operation of the algorithm for the computation of $z = \text{XLIX} + \text{LXI}$ (49 + 61):

operation	line
$x' \leftarrow M(\text{XLIX}) = \text{XXXXVIIII}$	1
$y' \leftarrow M(\text{LXI}) = \text{LXI}$	2
$cat(\text{XXXXVIIII}, \text{LXI}) = \text{XXXXVIIII.LXI}$	3
$z' \leftarrow group(\text{XXXXVIIII.LXI}) = \text{LXXXXXVIIIII}$	3
$E(\text{LXXXXXVIIIII}) = \text{LLVV}$	4
$z \leftarrow E(\text{LLVV}) = \text{CX}$	4

We thus get $z \leftarrow \text{CX}$ (110).

III. Substraction

The algorithm for substraction is slightly more complex than the one for addition, as it involves more conditional expansions/reductions. The algorithm to compute $z = x - y$ where $x > y$ (where $>$ is the natural order on Roman numerals) is setup by defining x' and y' to be respectively $\tilde{M}(x)$ and $\tilde{M}(y)$. Let the function $v(x)$ turn a Roman numeral $d_1 d_2 ... d_n$ into the vector $(d_1, d_2, ..., d_n)$. We also define $\overline{x} = v(x')$ and $\overline{y} = v(y')$. The core of the algorithm is as follows:

1 while $\overline{y} \neq ()$
2 — $m_x \leftarrow \min(\overline{x})$
3 — $m_y \leftarrow \min(\overline{y})$
4 — if $m_y \in \overline{x}$
5 —— $\overline{y} \leftarrow \overline{y} - m_y$
6 —— $\overline{x} \leftarrow \overline{x} - m_y$
7 — else if $m_y < m_x$
8 —— $\overline{x} \leftarrow E^{-1}(\overline{x})$
9 — else
10 —— if $E(\overline{x}) \neq \overline{x}$
11 ——— $\overline{x} \leftarrow E(\overline{x})$
12 —— else
13 ——— $\overline{y} \leftarrow E^{-1}(\overline{y})$
14 $z \leftarrow \tilde{M}^{-1}(\tilde{E}(\overline{x}))$

We describe the operation of the algorithm for the computation of $z = \text{XX} - \text{XVIII}$ (20 - 18). We thus have $\overline{x} = (\text{X}, \text{X})$ and $\overline{y} = (\text{X}, \text{V}, \text{I}, \text{I}, \text{I})$. We write the vectors as standard numerals for the sake of conciseness. The algorithms runs as follows:

operation	m_x	m_y	line
$\overline{x} \leftarrow E^{-1}(\text{XX}) = \text{VVVV}$	X	I	8
$\overline{x} \leftarrow E^{-1}(\text{VVVV}) = 20\text{I}$	V	I	8
$\overline{y} \leftarrow \text{XVII}$	I	I	5
$\overline{x} \leftarrow 19\text{I}$	I	I	6
$\overline{y} \leftarrow \text{XVI}$	I	I	5

operation	m_x	m_y	line
$\overline{x} \leftarrow$ 18I	I	I	6
$\overline{y} \leftarrow$ XV	I	I	5
$\overline{x} \leftarrow$ 17I	I	I	6
$\overline{x} \leftarrow$ VVVII	I	V	11
$\overline{y} \leftarrow$ X	V	V	5
$\overline{x} \leftarrow$ VVII	V	V	6
$\overline{x} \leftarrow$ XII	V	X	11
$\overline{y} \leftarrow ()$	X	X	5
$\overline{x} \leftarrow$ II	X	X	6
$z \leftarrow$ II	I		14

We thus get $z \leftarrow$ II (2).

IV. Multiplication

Roman numerals multiplication can be achieved through matrix multiplication of digits vectors. The algorithm to compute $z = xy$ is as follows:

1 $x' \leftarrow \tilde{M}(x)$
2 $y' \leftarrow \tilde{M}(y)$
3 $\overline{x} \leftarrow v(x')$
4 $\overline{y} \leftarrow v(y')$
5 $\overline{z} \leftarrow \overline{x} \times \overline{y}^{\perp}$
6 $z' \leftarrow group(cat(\overline{z}))$
7 $z \leftarrow \tilde{M}^{-1}\left(\tilde{E}(z')\right)$

The matrix product on line 5 uses the following π function for component multiplication:

	I	V	X	L	C	D	M
I	I	V	X	L	C	D	M
V	V	XXV	L	CCL	D	MMD	5M
X	X	L	C	D	M	5M	10M
L	L	CCL	D	MMD	5M	25M	50M
C	C	D	M	5M	10M	50M	100M
D	D	MMD	5M	25M	50M	250M	500M
M	M	5M	10M	50M	100M	500M	1000M

We describe the operation of the algorithm for the computation of $z =$ XLIII · XII (43 · 12):

operation	line
$x' \leftarrow M(\text{XLIII}) =$ XXXXIII	1
$y' \leftarrow M(\text{XII}) =$ XII	2
$\overline{x} \leftarrow (\text{X}, \text{X}, \text{X}, \text{X}, \text{I}, \text{I}, \text{I})$	3
$\overline{y} \leftarrow (\text{X}, \text{I}, \text{I})$	4
$\overline{z} \leftarrow \overline{y}^{\perp} \times \overline{x} = \begin{pmatrix} c & c & c & c & x & x & x \\ x & x & x & x & i & i & i \\ x & x & x & x & i & i & i \end{pmatrix}$	5
$cat(\overline{z}) =$ CCCCXXX.XXXXIII.XXXXIII	6
$z' \leftarrow$ $group$(CCCCXXX.XXXXIII.XXXXIII) = CCCCXXXXXXXXXXXXXIIIIII	6
E(CCCCXXXXXXXXXXXXXIIIIII) = CCCCLLXVI	7
E(CCCCLLXVI) = CCCCCXVI	7
$z \leftarrow E$(CCCCCXVI) = DXVI	7

We thus get $z \leftarrow$ DXVI (516).

V. Division

We do not define a specific algorithm for (Euclidean) division here, as it can be reduced to substractions and the introduction of a trivial order on Roman numerals.