How to Program an Abacus

Baptiste Mélès 梅乐思

Archives Henri Poincaré – Université de Lorraine (Nancy)

SAW
Cultures of computation and quantification
March 25-29, 2013

Question

- What functions can be computed with different kinds of non-mechanical calculating tools?
- Examples of calculating tools: abacus, counting-rods, logarithm tables, slide rule, counting on paper...
- Examples of limitations: logarithm tables and slide rule have no addition.
- What can I compute with my abacus? systems of linear equations? logarithms?

Question

- What functions can be computed with different kinds of non-mechanical calculating tools?
- Examples of calculating tools: abacus, counting-rods, logarithm tables, slide rule, counting on paper...
- Examples of limitations: logarithm tables and slide rule have no addition.
- What can I compute with my abacus? systems of linear equations? logarithms?

Question

- What functions can be computed with different kinds of non-mechanical calculating tools?
- Examples of calculating tools: abacus, counting-rods, logarithm tables, slide rule, counting on paper...
- Examples of limitations: logarithm tables and slide rule have no addition.
- What can I compute with my abacus? systems of linear equations? logarithms?

Question

- What functions can be computed with different kinds of non-mechanical calculating tools?
- Examples of calculating tools: abacus, counting-rods, logarithm tables, slide rule, counting on paper...
- Examples of limitations: logarithm tables and slide rule have no addition.
- What can I compute with my abacus? systems of linear equations? logarithms?

Classical answers

- Two possible answers:
  - *A posteriori* (history of mathematics): "just read your classics."
    - Cheng Dawei 程大位 (1533–1606), *Suanfa Tongzong* 算法統宗 (1592).
  - *A priori* (computer science): "just make a machine."
    - Alan Turing (1912–1954), "On Computable Numbers" (1936): the definition of "Turing machines" begins with the behaviour of a human "computer" (i.e. the calculating *man*);
    - Joachim Lambek (born 1922), "How to Program an Infinite Abacus" (1961).

Classical answers

- Two possible answers:
  - *A posteriori* (history of mathematics): "just read your classics."
    - Cheng Dawei 程大位 (1533–1606), *Suanfa Tongzong* 算法統宗 (1592).
  - *A priori* (computer science): "just make a machine."
    - Alan Turing (1912–1954), "On Computable Numbers" (1936): the definition of "Turing machines" begins with the behaviour of a human "computer" (i.e. the calculating *man*);
    - Joachim Lambek (born 1922), "How to Program an Infinite Abacus" (1961).

Classical answers

- Two possible answers:
  - *A posteriori* (history of mathematics): "just read your classics."
    - Cheng Dawei 程大位 (1533–1606), *Suanfa Tongzong* 算法統宗 (1592).
  - *A priori* (computer science): "just make a machine."
    - Alan Turing (1912–1954), "On Computable Numbers" (1936): the definition of "Turing machines" begins with the behaviour of a human "computer" (i.e. the calculating *man*);
    - Joachim Lambek (born 1922), "How to Program an Infinite Abacus" (1961).

Limits

- *A posteriori* (history of mathematics): "just read your classics".
  - But the classics only describe what *can* be done, and not what *cannot* be done.
- *A priori* (computer science): "just make a machine".
  - But we lose the cultural side of the problem: the role of the human hand, the pedagogical techniques... And infinite abaci do not exist.

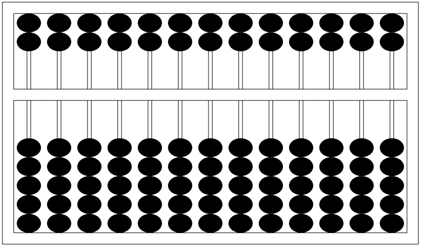
Methodology

Some concepts of computer science may help us to understand how computing tools work:

- data structures: the objects of the calculus, with which we compute;
- functions or procedures: the acts of the calculus, performed by human hands.

See Karine Chemla, *Les Neuf Chapitres*, which uses Knuth's computing model (*The Art of Computer Programming*).

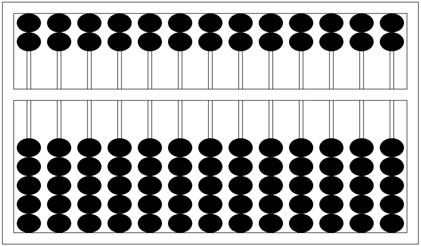
Data structure of the abacus



What are abaci?

There are a lot of different kinds of abacus:

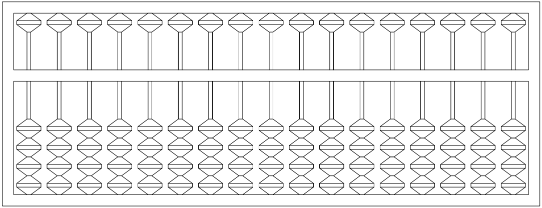
1. Chinese *suanpan* 算盤:



- 13 or more rods;
- 2 quinary beads on each rod;
- 5 unary beads on each rod.

What are abaci?

2. Japanese *soroban* 算盤:



- from 13 to 21 rods;
- 1 quinary bead on each rod;
- 4 unary beads on each rod.

What are abaci?

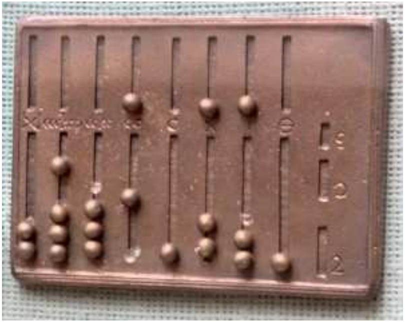
3. Russian *schoty* (счёты):
   - 7 or more rods;
   - 10 unary beads on each rod (no quinary bead);
   - 4 beads for quarters of rubles;
   - 4 beads for quarters of kopeks (until 1916).

What are abaci?

4. French abacus (*boulier*):
   - 10 unary beads on each rod.

What are abaci?

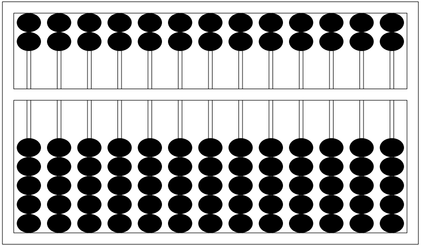
5. One particular kind of Roman abacus:



- 1 quinary bead on each column;
- 4 unary beads on each column;
- units are sometimes fixed.

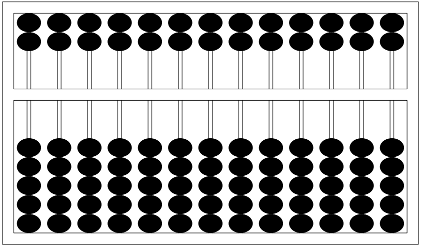
How to count on an abacus?

First question: what are the data structures (the objects of computation) on the abacus?



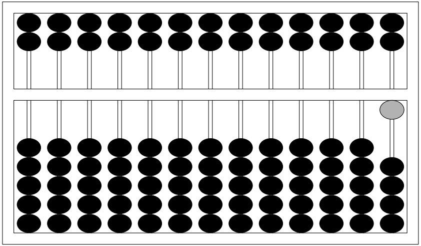
How to count on an abacus?

How to count on the *suanpan*:



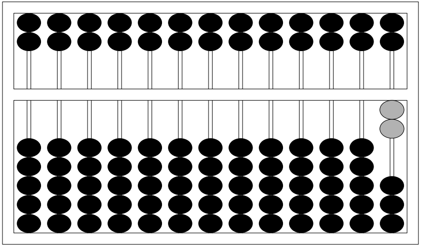
How to count on an abacus?

How to count on the *suanpan*:



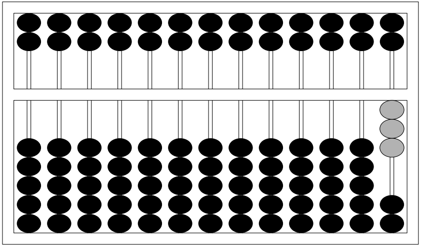
How to count on an abacus?

How to count on the *suanpan*:



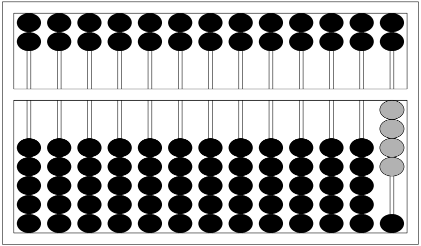
How to count on an abacus?

How to count on the *suanpan*:



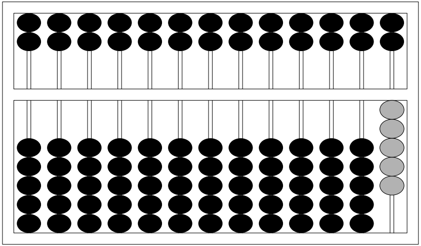
How to count on an abacus?

How to count on the *suanpan*:



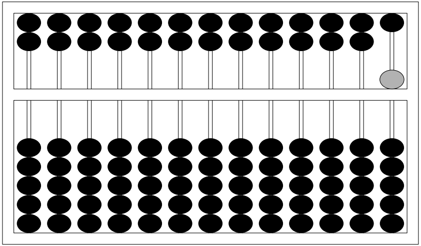
How to count on an abacus?

How to count on the *suanpan*:



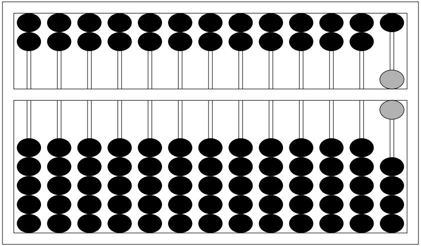
How to count on an abacus?

How to count on the *suanpan*:



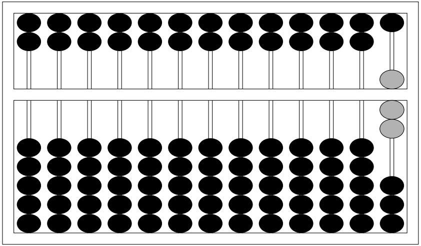
How to count on an abacus?

How to count on the *suanpan*:



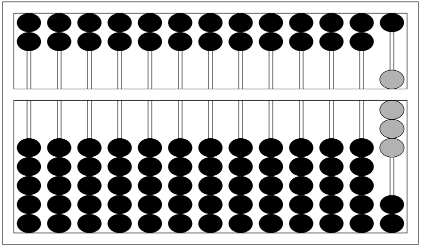
How to count on an abacus?

How to count on the *suanpan*:



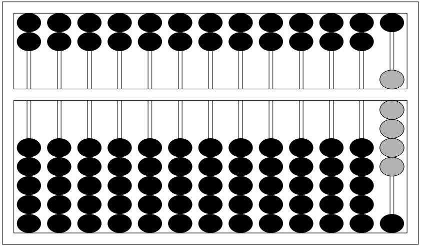
How to count on an abacus?

How to count on the *suanpan*:



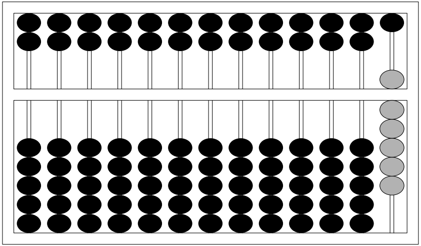
How to count on an abacus?

How to count on the *suanpan*:



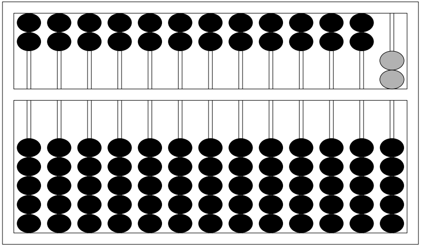
How to count on an abacus?

How to count on the *suanpan*:



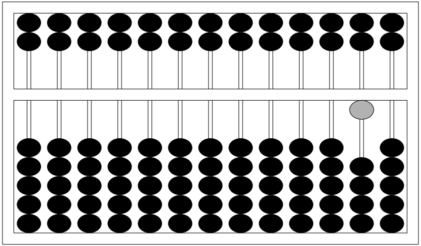
How to count on an abacus?

How to count on the *suanpan*:



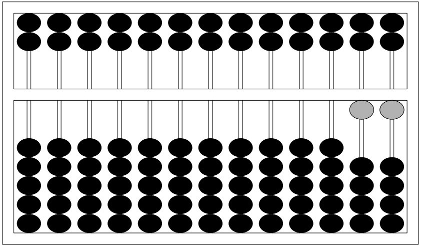
How to count on an abacus?

How to count on the *suanpan*:



How to count on an abacus?

How to count on the *suanpan*:



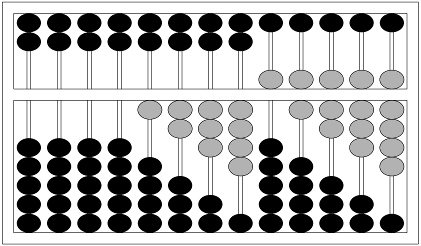
What is a number?

What is a number on an abacus?
It depends on your abacus:

- *soroban*: decimal numbers;
- *boulier*: decimal numbers;
- *schoty*: absolute decimal numbers;
- Greek and Roman abaci: (floating or absolute) decimal numbers;
- *suanpan*:
  - decimal numbers;
  - hexadecimal numbers for 1 *jin* 斤 = 16 *liang* 兩.

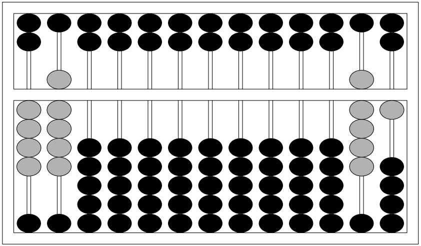
How many numbers can we write on an abacus?

We can encode **one** number, for example 123 456 789, on the *suan-pan*:



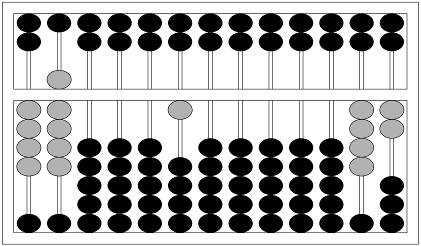
How many numbers can we write on an abacus?

Sometimes, it can be useful to write down **two** numbers, typically for the greatest common divisor (for instance of 49 and 91):



How many numbers can we write on an abacus?

And sometimes even **three** numbers, as for "euclidian" division (divisor, quotient, dividend or rest), as for $91 = 49 \times 1 + 42$:



Data structure of the abacus

1. Data structure: one register containing a list of a few numbers.
2. Unlike the counting rods, space is *very* limited: one can hardly write more than one or two numbers. This will be a problem, since most of our arithmetical operations (addition, multiplication...) are binary.

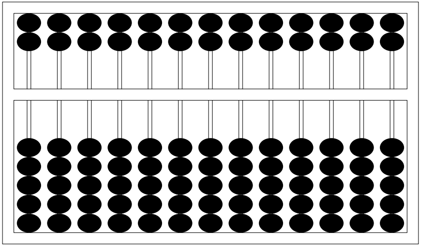
Basic operations

Second question: what are the basic operations on an abacus?

Addition without carry

Let us begin with an addition without carry: $12 + 81$.

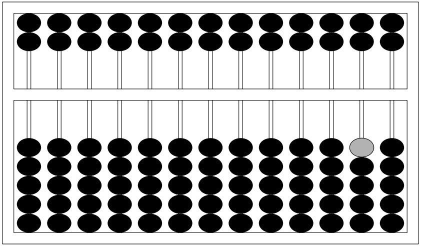
1. Reset the abacus:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

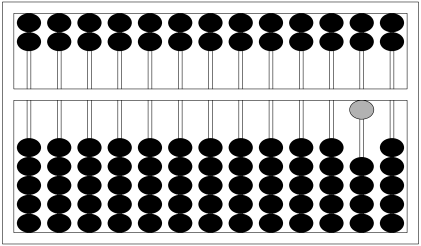
2. Encode the first number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

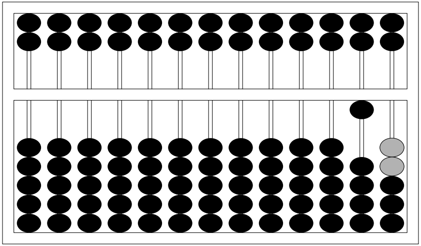
2. Encode the first number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

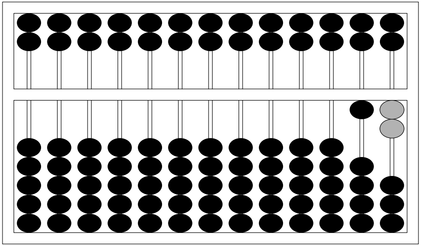
2. Encode the first number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

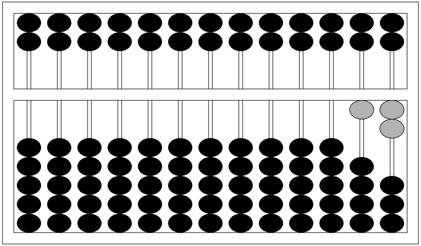
2. Encode the first number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

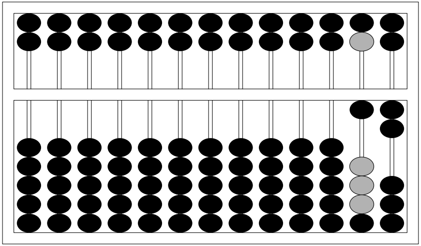
2. Encode the first number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

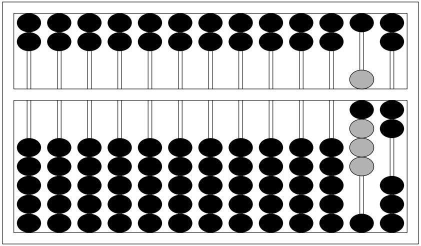
3. Add the second number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

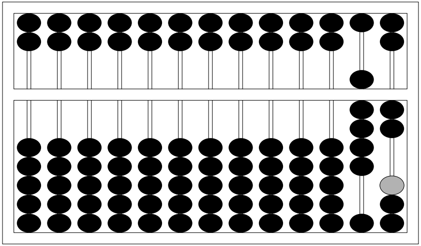
3. Add the second number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

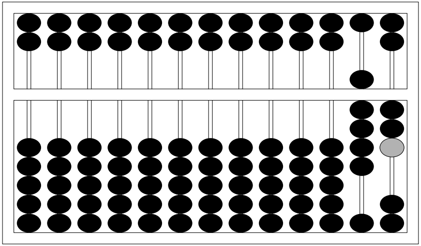
3. Add the second number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

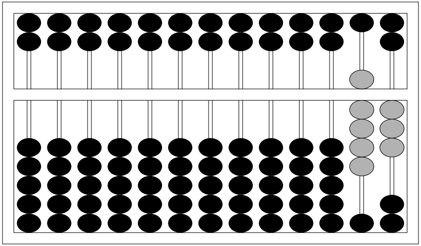
3. Add the second number:



Addition without carry

Let us begin with an addition without carry: $12 + 81$.

4. Read the result:



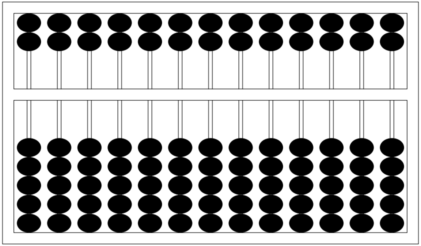
Theorem

$12 + 81 = 93.$

Addition with carry

Let us now compute an addition with carry: $93 + 234$.

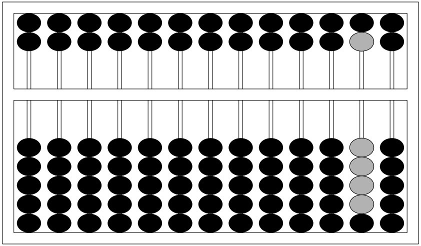
1. Reset the abacus:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

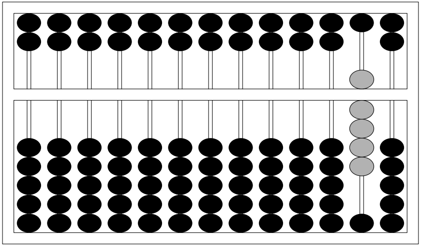
2. Encode the first number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

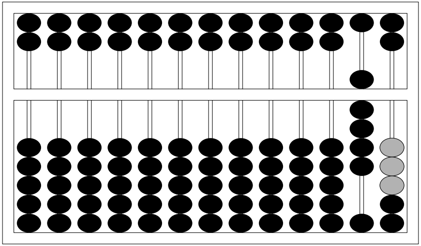
2. Encode the first number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

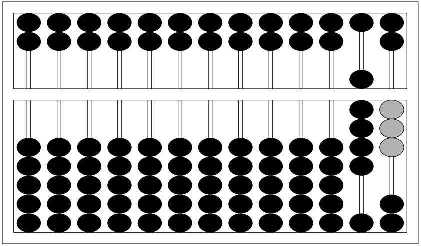
2. Encode the first number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

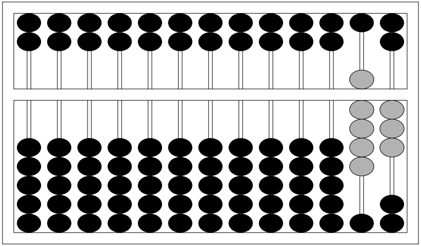
2. Encode the first number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

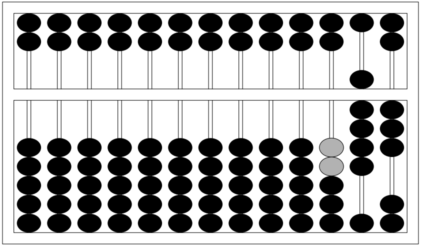
2. Encode the first number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

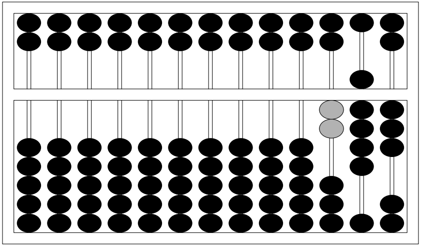
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

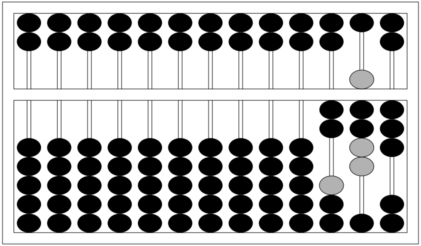
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

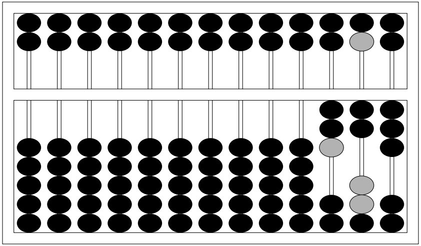
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

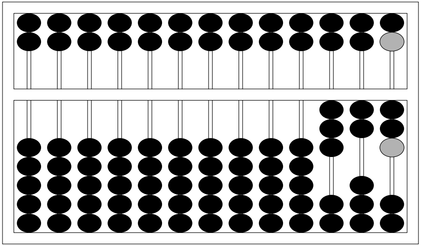
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

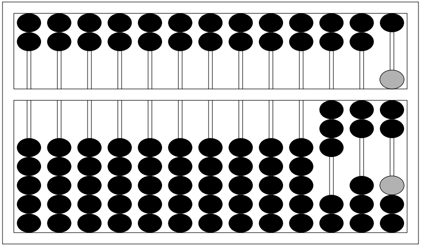
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

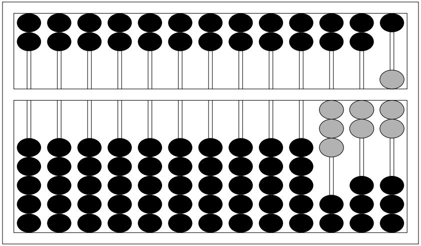
3. Add the second number:



Addition with carry

Let us now compute an addition with carry: $93 + 234$.

4. Read the result:

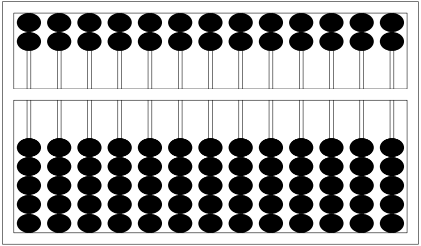


Theorem

$93 + 234 = 327.$

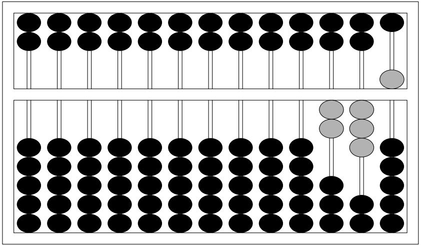
Multiplication

Let us compute 235×14.



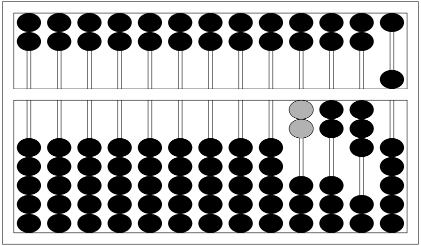
Multiplication

Let us compute 235×14.



Multiplication

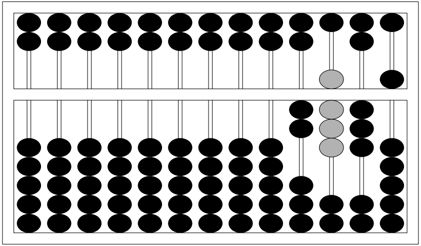
Let us compute 235×14.



because $2 \times 1 = 2$

Multiplication

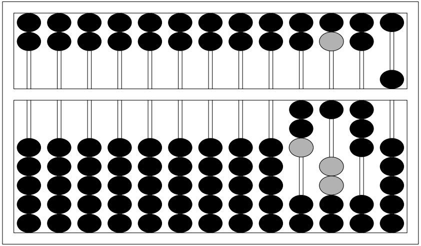
Let us compute 235×14.



because $2 \times 4 = 8$

Multiplication

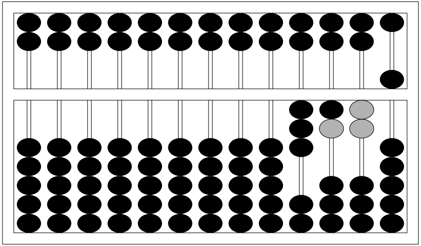
Let us compute 235×14.



because $3 \times 1 = 3$

Multiplication

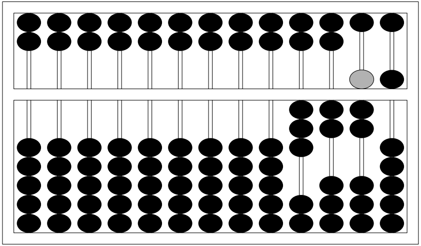
Let us compute 235×14.



because $3 \times 4 = 12$

Multiplication

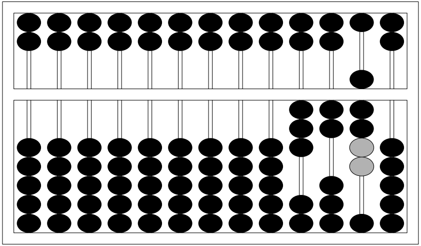
Let us compute 235×14.



because $5 \times 1 = 5$

Multiplication

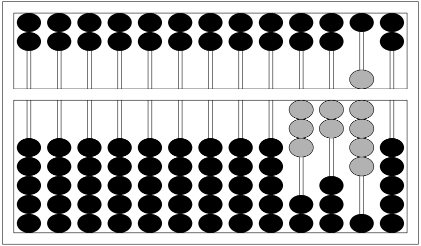
Let us compute 235×14.



because $5 \times 4 = 20$

Multiplication

Let us compute 235×14.



Theorem

$235 \times 14 = 3290.$

Unary operations

- What can we observe? At each step of our computations, there was *one and only one* number on the abacus.

Theorem

Addition and multiplication are unary operations.

Unary operations

- What can we observe? At each step of our computations, there was *one and only one* number on the abacus.

Theorem

Addition and multiplication are unary operations.

Some words about curryfication

How is it possible? How is it that we can transform a binary operation into an unary operation?

Definition

Curryfication (Schönfinkel, Curry) = transformation of an *n-ary* function into a mere composition of *unary* functions.

Some words about curryfication

Definition

Curryfication (Schönfinkel, Curry) = transformation of an *n*-ary function into a composition of unary functions (provided that their respective "values" can themselves be functions).

Example

Without curryfication	After curryfication
$\begin{tikzcd} 12 \arrow[dr] & & \\ & (12,81) \arrow[r, "+"] & 93 \\ 81 \arrow[ur] & & \end{tikzcd}$	$\begin{tikzcd} & 81 \arrow[d, "\mathit{curryfied}+"] & \\ 12 \arrow[r, dash] & {} \arrow[r, "+81"'] & 93 \end{tikzcd}$

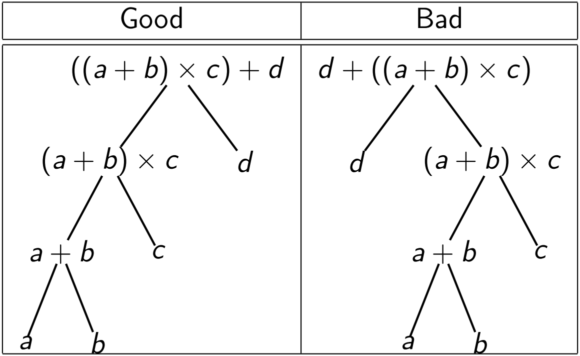
Some words about curryfication

When we compute $12 + 81$:

- the operands play asymmetrical roles:
  - 12 is a number, a passive object, represented on the tool;
  - 81 is part of a function; it is an act of my hand;
- the result does not occupy a distinct place: it results from the transformation of the input number;
- there is no "variable assignment" (numbers put in memory during the computation). This is an other difference with the counting rods (see Karine Chemla, *Les Neuf Chapitres*).

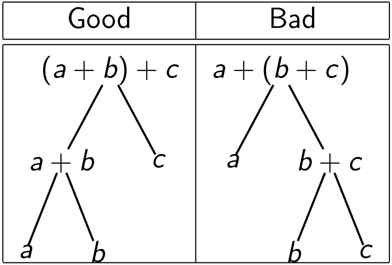
Computational consequences

Since there is no variable assignment, intermediary results can not be memorized. As a consequence, some operations are not possible because of their very syntactic tree.



Computational consequences

- $(a + b) + c$ is computable,
- $a + (b + c)$ is not computable.

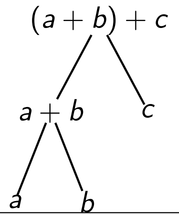
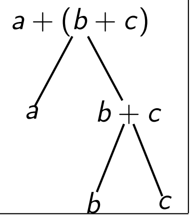


Theorem

Addition is not associative.

Computational consequences

- $(a+b)+c$ is computable,
- $a+(b+c)$ is not computable.

Good	Bad
	

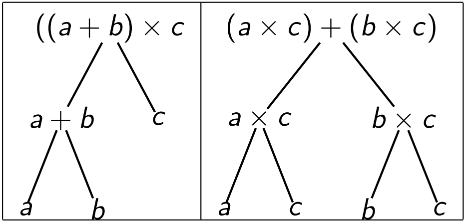
Theorem

Addition is not associative.

Computational consequences

- $(a + b) \times c$ is computable,
- $(a \times c) + (b \times c)$ is not computable.

Good	Bad

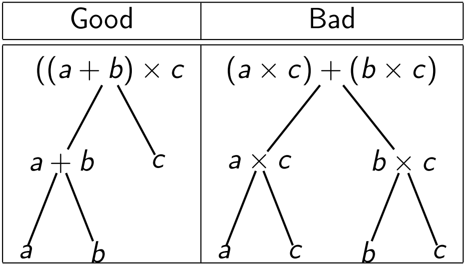


Theorem

Multiplication is not distributive.

Computational consequences

- $(a + b) \times c$ is computable,
- $(a \times c) + (b \times c)$ is not computable.



Theorem

Multiplication is not distributive.

Computational consequences

Can we at least save the commutativity of addition?

Fibonacci's commutative addition

2	and	2	make	4
2		3		5
2		4		6
2		5		7
2		6		8
2		7		9
2		8		10
2		9		11
2		10		12

	Key	*for*	*Three*	
3	*and*	3	*make*	6
3		4		7
3		5		8
3		6		9
3		7		10
3		8		11
3		9		12
3		10		13

	Key	*for*	*Four*	
4	*and*	4	*make*	8
4		5		9
4		6		10
4		7		11
4		8		12
4		9		13
4		10		14

	Key	*for*	*Five*	
5	*and*	5	*make*	10
5		6		11
5		7		12
5		8		13
5		9		14
5		10		15

	Key	*for*	*Six*	
6	*and*	6	*make*	12
6		7		13
6		8		14
6		9		15
6		10		16

	Key	*for*	*Seven*	
7	*and*	7	*make*	14
7		8		15
7		9		16
7		10		17

	Key	*for*	*Eight*	
8	*and*	8	*make*	16
8		9		17
8		10		18

	Key	*for*	*Nine*	
9	*and*	9	*make*	18
9		10		19

	Key	*for*	*Ten*	
10	*and*	10	*make*	20
20	*and*	20	*make*	40

Fibonacci's commutative addition

Triangular addition table:

+	1	2	3	⋯
1	2			
2	3	4		
3	4	**5**	6	
⋮	⋮	⋮	⋮	⋱

Theorem

$3 + 2 = 2 + 3.$

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:	$27 + 14$:
$\begin{array}{rrr} & 1 & 4 \\ + & 2 & 7 \\ \hline \end{array}$	$\begin{array}{rrr} & 2 & 7 \\ + & 1 & 4 \\ \hline \end{array}$

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:	$27 + 14$:
$\begin{array}{rrr} & 1 & \\ & 1 & 4 \\ + & 2 & 7 \\ \hline & & 1 \end{array}$	$\begin{array}{rrr} & 1 & \\ & 2 & 7 \\ + & 1 & 4 \\ \hline & & 1 \end{array}$

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:	$27 + 14$:
$$\begin{array}{ccc} & 1 & \\ & 1 & 4 \\ + & 2 & 7 \\ \hline & 4 & 1 \end{array}$$	$$\begin{array}{ccc} & 1 & \\ & 2 & 7 \\ + & 1 & 4 \\ \hline & 4 & 1 \end{array}$$

Theorem

$14 + 27 = 27 + 14$ *(and for the same reasons).*

Cheng Dawei's instructions for addition

Chinese text	Instructions	Operation
下五除三	(move) down (one) quinary (bead) (and) remove three (unary beads)	$4+2=4+5-3$
下五除一	(move) down (one) quinary (bead) (and) remove one (unary bead)	$2+4=2+5-1$

Theorem

$2+4 \neq 4+2.$

Chinese text	Instructions	Operation
下五除一	(move) down (one) quinary (bead) (and) remove one (unary bead)	$2+4=2+5-1$
下五除一	(move) down (one) quinary (bead) (and) remove one (unary bead)	$3+4=3+5-1$

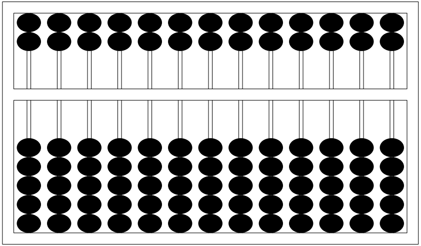
Theorem

$2+4=3+4.$

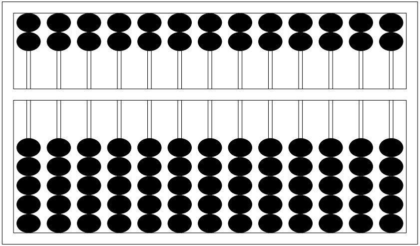
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Reset the abacus:



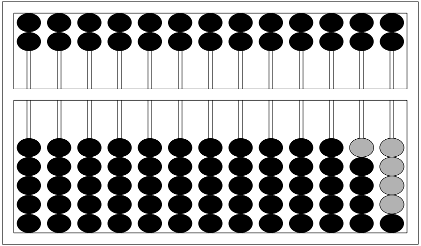
$27 + 14$:
Reset the abacus:



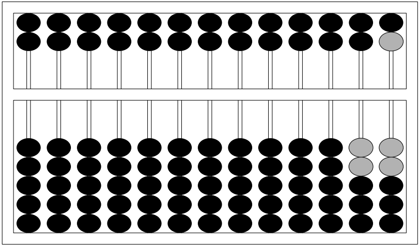
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Encode the first number:



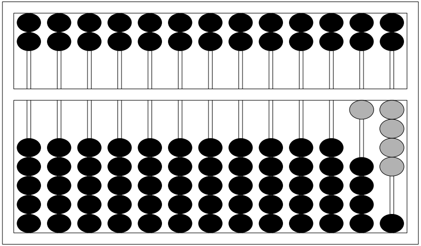
$27 + 14$:
Encode the first number:



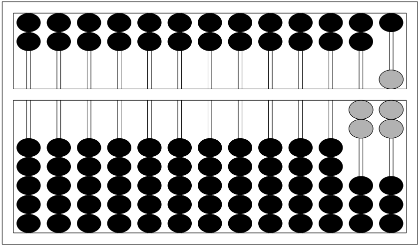
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Encode the first number:



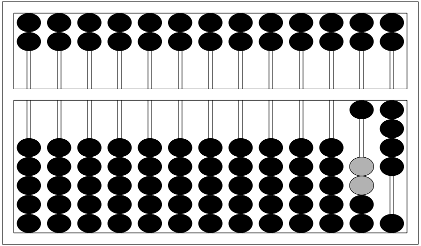
$27 + 14$:
Encode the first number:

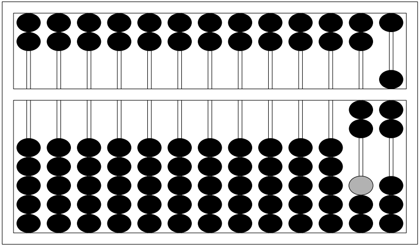


Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:	$27 + 14$:
Add the second number:	Add the second number:

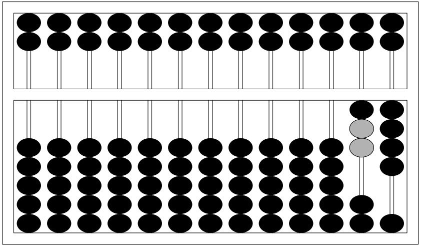




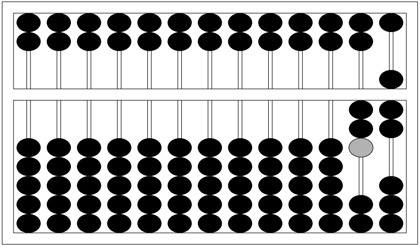
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Add the second number:



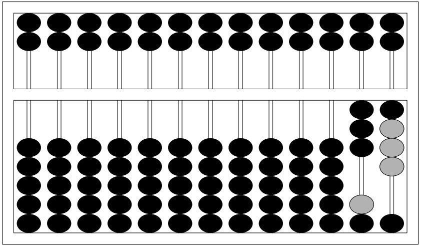
$27 + 14$:
Add the second number:



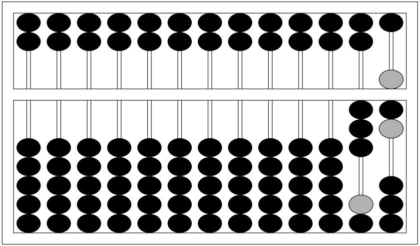
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Add the second number:



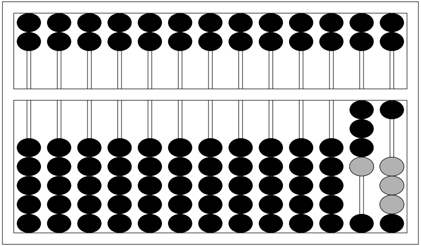
$27 + 14$:
Add the second number:



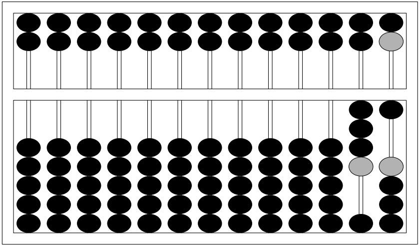
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Add the second number:



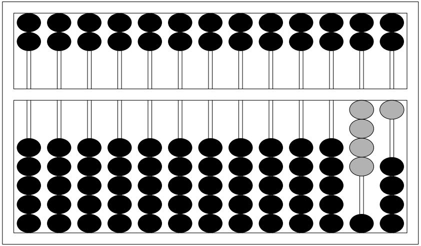
$27 + 14$:
Add the second number:



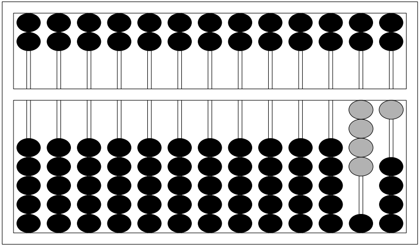
Cheng Dawei's instructions for addition

Let us compare $14 + 27$ and $27 + 14$.

$14 + 27$:
Add the second number:



$27 + 14$:
Add the second number:



Theorem

$14 + 27 = 41$ *and* $27 + 14 = 41$ *(but not for the same reasons).*

Cheng Dawei's instructions for addition

Addition on an abacus is:

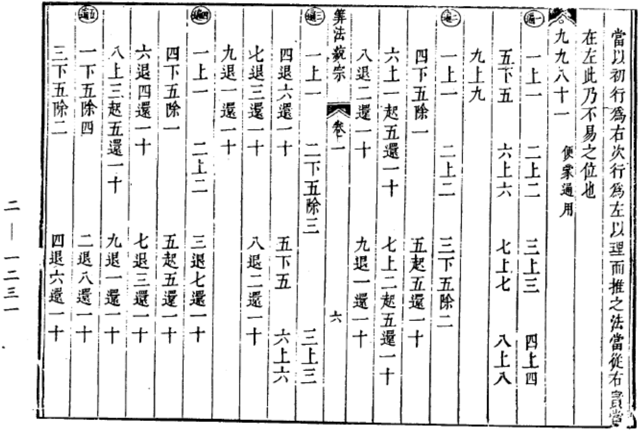
- *denotationally* commutative (the result is the same),
- but *operationally* not commutative (the operations are different).

$$\begin{tikzcd}
 & & 0 \arrow[dl, "+14"'] \arrow[dr, "+27"] & \\
\text{operational semantics} & 14 \arrow[dr, "+27"'] & & 27 \arrow[dl, "+14"] \\
\text{denotational semantics} & & 41 &
\end{tikzcd}$$

Cheng Dawei's instructions for addition

Can we even speak of an addition table in Cheng Dawei's book?

Cheng Dawei's instructions for addition



當以初行爲右次行爲左以理而推之法當從右實當
在左此乃不易之位也
九九八十一 便蒙通用
一上一 二上二 三上三 四上四
五下五 六上六 七上七 八上八
九上九
一上一 二上二 三下五除二
四下五除一 五起五還一十
六上一起五還一十 七上二起五還一十
八退二還一十 九退一還一十
算法統宗 卷一 六
一上一 二下五除三 三上三
四退六還一十 五下五 六上六
七退三還一十 八退二還一十
九退一還一十
一上一 二上二 三退七還一十
四下五除一 五起五還一十
六退四還一十 七退三還一十
八上三起五還一十 九退一還一十
一下五除四 二退八還一十
三下五除二 四退六還一十

Cheng Dawei's instructions for addition

中國科學技術典籍通彙 數學卷

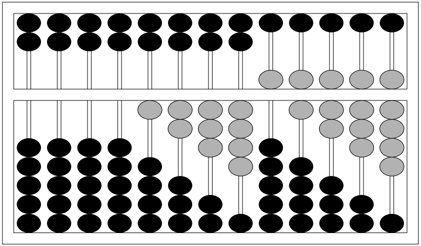
五下五 六上一起五還一十 七上七
八退二還一十 九退一還一十
(三六) 一上一 二上二 三退七還一十
四下五除一 五起五還一十
六上六 七退三還一十
八退二還一十 九上四起五還一十
(三七) 一上一 二下五除三 三上三
四退六還一十 五下五
六退四還一十 七上二起五還一十
八退二還一十 九退一還一十

筭法統宗 卷一 七

(三八) 一上一 二上二 三下五除二
四下五除一 五起五還一十
六上一起五還一十 七退三還一十
八退二還一十 九退一還一十
(三九) 一上一 二上二 三上三 四上四
五下五 六上六 七上七 八上八
九退一還一十

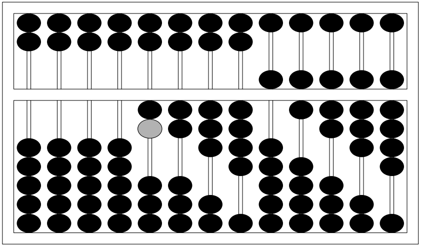
九九合數 乘除加減皆呼此數 故呼小數在上大數在下
○一一如一 ○一二如二 二二如四
○一三如三 二三如六 三三如九

Cheng Dawei's instructions for addition



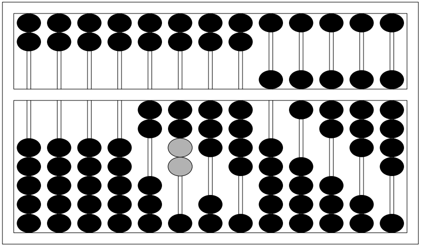
123456789

Cheng Dawei's instructions for addition



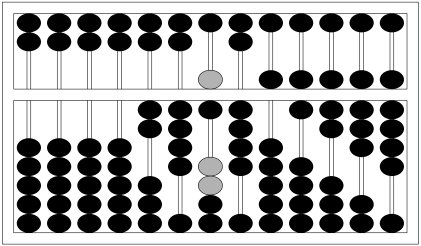
$$\begin{array}{r} 123456789 \\ +100000000 \\ \hline 223456789 \end{array}$$

Cheng Dawei's instructions for addition



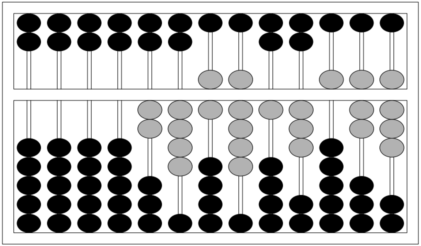
$$
\begin{array}{r}
123456789 \\
+100000000 \\
+20000000 \\
\hline
243456789
\end{array}
$$

Cheng Dawei's instructions for addition



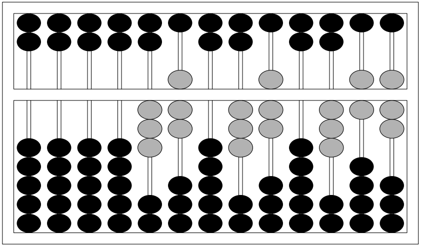
$$\begin{array}{r} 123456789 \\ +100000000 \\ +20000000 \\ +3000000 \\ \hline 246456789 \end{array}$$

Cheng Dawei's instructions for addition



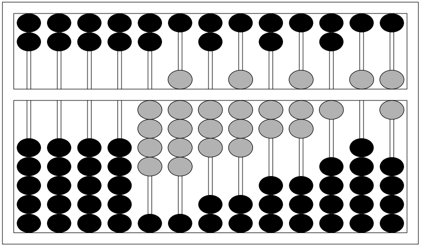
$$\begin{array}{r} 123456789 \\ +123456789 \\ \hline 246913578 \end{array}$$

Cheng Dawei's instructions for addition



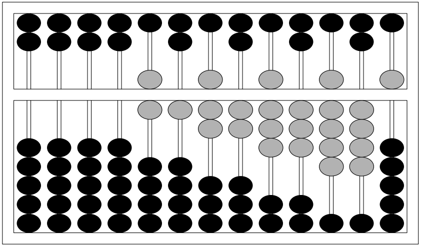
$$\begin{array}{r} 123456789 \\ +123456789 \\ +123456789 \\ \hline 370370367 \end{array}$$

Cheng Dawei's instructions for addition



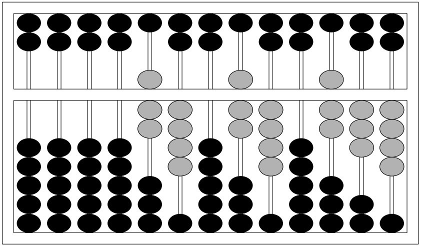
$$
\begin{array}{r}
123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
\hline
493827156
\end{array}
$$

Cheng Dawei's instructions for addition



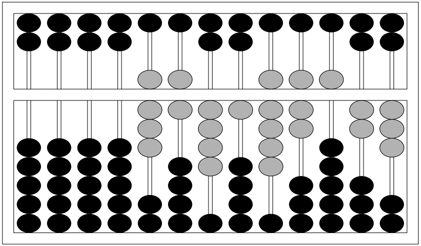
$$
\begin{array}{r}
123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
\hline
617283945
\end{array}
$$

Cheng Dawei's instructions for addition



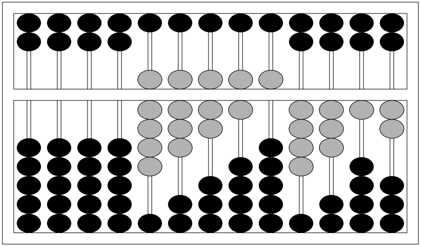
$$
\begin{array}{r}
123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
\hline
740740734
\end{array}
$$

Cheng Dawei's instructions for addition



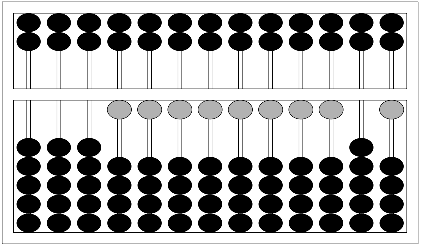
$$\begin{array}{r} 123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ \hline 864197523 \end{array}$$

Cheng Dawei's instructions for addition



$$
\begin{array}{r}
123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
+123456789 \\
\hline
987654312
\end{array}
$$

Cheng Dawei's instructions for addition



$$\begin{array}{r} 123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ +123456789 \\ \hline 1111111101 \end{array}$$

It is not a table of addition, but an exercise.

Cheng Dawei's instructions for addition

Cheng Dawei's "table of addition" is not even complete:

+	0	1	2	3	4	5	6	7	8	9
0		1	1	3	1	1	3	1	1	1
1		1	1		1	1		1	1	
2		1	1		1	1			1	1
3		1		2	1		2	2	1	1
4		1	2	1	1	1	1		1	1
5						2		2	1	1
6		1	1	1	1		1		1	1
7		1	1	2	1	1	2	2	1	1
8		1	1		1	1			1	1
9		1	1		1	1		1		1

Cheng Dawei's instructions for addition

It is not complete, even up to commutativity:

+	0	1	2	3	4	5	6	7	8	9
0		1	1	3	1	1	3	1	1	1
1		1	2	1	2	1	1	2	2	1
2			1		3	1	1	1	2	2
3				2	2		3	4	1	1
4					1	1	2	1	2	2
5						2		3	2	2
6							1	2	1	1
7								2	1	2
8									1	1
9										1

This "table" does not teach the list of all possible additions, but just the set of elementary techniques. It is an *operational pedagogy*.

Conclusion

On an abacus,

- "numbers" are not always numbers: they may be acts;
- neither addition nor multiplication is a binary operation;
- addition is neither associative nor commutative;
- addition does not require a table;
- multiplication is not distributive.

Conclusion

When we study a computing tool, we have to forget what we know about these operations.
All mathematical properties are not always effective on computing tools.
Some concepts of computer science (data structures, curryfication, variable assignment, semantics of programming languages) can help us describe the properties which are effective on computing tools.

Bibliography

- Cheng Dawei 程大位, *Suanfa Tongzong* 算法統宗, 1592.
- Karine Chemla and Guo Shuchun, *Les Neuf Chapitres*, 2004.
- Chen Yifu, *Les Différents Modes de manipulation des boules du boulier chinois à l'époque des Ming en Chine*, 2007.
- Kojima Takashi, *The Japanese Abacus: its Use and Theory*, 1954; *Advanced Abacus: Theory and Practice*, 1963.
- Joachim Lambek, "How to Program an Infinite Abacus", 1961.
- Alain Schärlig, *Compter avec des cailloux*, 2001.