Number, Writing, and Computation

Alex Sanchez

November 2023

1 Number

“Just two more feet to the left.” “I need forty cc’s of Pancuronium, stat!” “The consumer price index rose by half a percent during January.” “That’s a long pass from the Broncos thirty yard line to the Dolphins twenty yard line.” “Three cheers for Robin Hood!”

Most familiarly, people use *numbers* to facilitate precise communication. In particular, numbers refine and, indeed, enable the notion of measurement.

1.1 Measurement

Different kinds of objects want for different kinds of measurements–and different kinds of measurements want for different kinds of numbers! For instance[1],

- If you want to count the number of individual M&M’s candy pieces in a bowl, then (nonnegative) *natural numbers*, like $0, 1, 2, 3, \ldots$ etc., are the appropriate tools. Sometimes 0 is considered as a natural number and sometimes it is not; there is no agreement on this convention. So, I’ll generally specify “(positive) natural numbers” to mean that 0 is **not** considered a natural number or “(nonnegative) natural numbers” to mean that 0 is considered a natural number.

- If you want to measure a person’s height, then you might report their approximate height as a natural number of inches, like 67 inches[2]. However, if you wanted to measure the diameter of an individual M&M’s candy piece, then rounding to the nearest inch would probably be “about 0 inches”, which doesn’t tell you much! The problem, of course, is that inches are too large, or *imperfect*, to measure individual M&M’s candy pieces. Instead, you might want to use millimeters, which are much smaller. Of course, “about 11 millimeters” is a much more useful measurement!

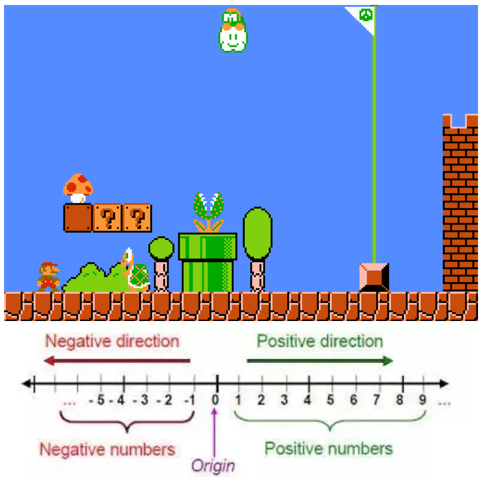
[1]Don’t worry about understanding all the details for the following examples; they are only meant to make clear the idea that different number systems are designed to handle different kinds of measurements.

[2]Which is, equivalently (and more commonly), reported as 5’7”.

(Generated with Midjourney; prompt by Tucker Forbes[3])

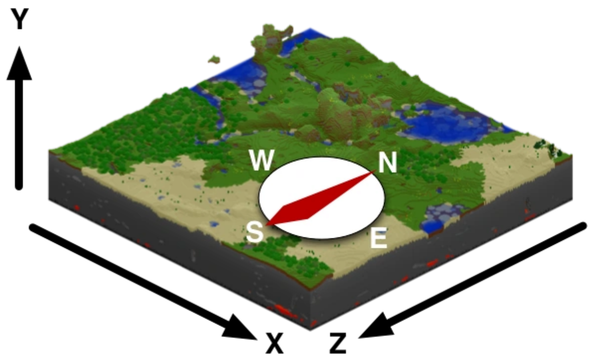
For those people who lack a prior physical intuition for the length of a millimeter but are familiar with inches, it would be useful to relate inches to millimeters. Indeed, we can say that 127 millimeters is the same[4] length as 5 inches. Using the language of nonnegative *rational numbers*[5], we can say that a millimeter is equal to $\frac{5}{127} \approx 0.039$ inches. Indeed, the use of rational numbers allows us to make measurements at *arbitrary* precision.

- The natural numbers are also insufficient to measure the position of character in an old-school 2D platformer, like the Mario surface, even though the positions are all *discrete*[6], and so there is no issue with precision like we had with measuring physical lengths. Instead, the issue is that the natural numbers only extend infinitely in one direction–but our character can walk off in two directions: left and right! We remedy this by introducing "negative" numbers, which are nothing more than another copy of the naturals (without zero) going in the other direction. The resulting numbers are called the *integers*. By convention, the positive integers are written on the right and the negative numbers go off to the left, like $\ldots, -2, -1, 0, 1, 2, \ldots$.



- If, instead of Mario, we want to measure the position of a character in Minecraft, we need to use (positive and negative) rational numbers because of the precision concerns from before (since players can move continuously in Minecraft, enabling parkour maneuvers).

[3]Check out his YouTube channel, @AutodidactOfficial!
[4]It turns out that this actually the *definition* of an inch, so this ratio is **exact**.
[5]These are more commonly known as "fractions".
[6]In these games, characters reside in "spaces", like (abstract) chess pieces in squares, at any given time, rather than sliding freely and *continuously*, like (physical) chess pieces on a flat board or a hockey puck on ice.



However, Minecraft is also 3D, so we need to use 3 numbers, called *(vector) coordinates*, to describe the position–one number for each direction: east/west, up/down, and north/south.

- There are even extremely intricate number systems for recording things as abstract as the configuration state of a Rubik's cube using the idea of symmetry groups. Of course, the details of such a construction are beyond the scope of this discussion. However, it should be noted that mathematicians have dreamed up numbers for all kinds of occasions, including ones you might care about–but that you probably won't see in school.

Exercise: write down some things you might measure and what kind of qualities you'd want for number systems to represent those measurements.

1.2 Natural Numbers

But let's not get ahead of ourselves. We start by considering only the (positive) natural numbers, $1, 2, 3, \ldots$ and some of their basic properties. There is only one basic idea we need at this stage: **The natural numbers count the number of 1's which they're equal to.** To illustrate,

- $\underbrace{1}_{1} = 1$ This one[7] is almost too simple...but the pattern will become clear in the following cases.
- $\underbrace{1+1}_{2} = 2$
- $\underbrace{1+1+1}_{3} = 3$
- $\underbrace{1+1+1+1}_{\mathbf{4}} = 4$ This bold number (under the brace) is just counting the number of 1's in the sum on the left-hand-side of the equation.
- $\underbrace{1+1+1+1+1}_{5} = \mathbf{5}$ This bold number is numerical *value* of the sum.

$\vdots$

- $\underbrace{1+1+1+1+1+1+1 \cdots + 1}_{\mathbf{100}} = \mathbf{100}$ In particular, no matter how many ones there are, the two bold numbers will always be the same (here, we see the case for 100); that is, the number of 1's a sum of this kind will always equal that sum's value.

[7]Ba dum tss

$\vdots$

In principle, this idea of successive enumeration is really all there is to the natural numbers. However, as you're probably aware, it's not that easy in practice. A great bit of complexity occurs in at least two ways:

1. There is complexity in how we represent/compute with numbers.
2. There are deep mathematical complexities in the structure of the ways in which numbers multiply together.

For now, we focus on the former.

2 Writing

The original purpose of writing, in the first place, was to communicate with people who are not present in space or time. The concept of the (phonetic) alphabet allowed people to reconstruct some version of spoken language directly from the written language; rather than having to memorize independent glyphs for each word/idea in a language, the letters of an alphabet allow the reader to "compute" the spoken word. Since languages have many more words than sounds, this reduces the memory demands on the reader. The situation is more extreme with numbers: there are infinitely many numbers, so memorizing a unique glyph for each number is doomed to failure. Throughout the ages, the inventors of notational systems for numbers have understood this problem. We will look at a few different notational systems.

2.1 Tally Marks

2.1.1 The Basics

To my knowledge, all civilizations that developed a *numeral* system at all developed a system of tally marks. Tally marks use the idea about natural numbers that we saw before, that natural numbers count 1's. Here, we present a tally system based on the "$*$" symbol:

- "one" $= *$
- "two" $= **$
- "three" $= * * *$

  etc.

We quickly see the problem with such a naïve tally system. The number "forty-two" looks like

$$* **$$

in our system, as it stands. That is an unreadable and unwieldy numeral for a fairly modest number. Let alone trying to write down (or read!) the number "one thousand" (or worse)...

Exercise: write down the following numbers in the tally system:

- "four"
- "twelve"
- "twenty"

2.1.2 New Symbols

To accommodate large numbers, there are multiple options that we can use. The first one we'll explore is memorizing new symbols to represent collections of old symbols. So, we use the following new tally system:

- As before, "one" $= *$, "two" $= **$, etc.
- Now, we write "ten" $= \triangle = * * * * * * * * * *$.

  For instance, "eleven" $= \triangle *$, "twenty-one" $= \triangle\triangle *$, and "twelve" $= \triangle * *$.

This fixes the problem of making "forty-two" readable: indeed, $\triangle\triangle\triangle\triangle **$ is pretty manageable. However, "one thousand" remains an issue. No problem, we can just keep adding symbols:

$$\text{write "one hundred"} = \square = \triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle.$$

Now, even "one thousand" is semi-manageable to read/write: $\square\square\square\square\square\square\square\square\square\square$.

This system of numbers which we have constructed is more or less the same idea as that of *Roman numerals*, the system of numbers used by the Roman Empire–except[8] that the Romans added even more symbols to their numeral system...

> Exercise: write down the following numbers in the *new* tally system:
>
> - "four"
> - "fourty-two"
> - "one hundred eleven"

2.1.3 Exponential Scaling

However, we can see that we're fighting a losing battle by adding more and more symbols: since numbers get infinitely big, we're either going to have to add an infinite number of symbols to our numeral system or we're going to be unable to deal with very large numbers, like 1000000, etc. To that end, we explore a different idea than adding a list of new symbols altogether: we put boxes around symbols to indicate that those symbols represent 10 times their original value. This will be easier to understand with some examples:

- $\boxed{*} = * * * * * * * * * * = \text{"ten"} = \triangle$.
- $\boxed{\boxed{*}} = \boxed{* * * * * * * * * *} = \boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*}\boxed{*} = \text{"one hundred"} = \square = \triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle\triangle$.
- $\boxed{\boxed{\boxed{*}}} = \text{"one thousand", etc.}$
- Numbers like "one thousand two hundred thirty-four" are written as $\boxed{\boxed{\boxed{*}}}\boxed{\boxed{**}}\boxed{* * *} * * * *$.
- We can (barely) handle really, really big numbers, like "one million" $= \boxed{\boxed{\boxed{\boxed{\boxed{\boxed{*}}}}}}$.

[8] The Roman numeral system also has a way of subtracting depending on order, but I consider that a minor detail for the purposes of this development.

This system is very powerful. In fact, for most pedagogical purposes (especially before multiplication), it's **conceptually superior** to the standard way of writing numbers. However, it's slightly less economical (since you need to do a lot more physical writing) and less practical (since writing more than 3 nested boxes is a pain). Then again, it's perfectly good for most modestly sized numbers and it more transparently demonstrates many of the true properties natural numbers (especially with respect to addition).

Exercise: write down the following numbers in the **new** *new* tally system:

- "four"
- "two thousand one hundred eleven"
- "sixty thousand"

2.1.4 Best of Both Worlds: Final Tally System

Now, it's kind of annoying to write $\ast\ast\ast\ast\ast\ast\ast\ast\ast$ for "nine". So, instead of symbols like $\triangle$ and $\square$, which are now superseded by $\boxed{\ast}$ and $\boxed{\boxed{\ast}}$, respectively, we (re)introduce the familiar Arabic Numerals:

- $1 = \ast =$ "one"
- $2 = \ast\ast =$ "two"

  $\vdots$
- $9 = \ast\ast\ast\ast\ast\ast\ast\ast\ast =$ "nine"

Now, note that this system is **still a tally system**, not a positional system. That means that, in this tally system, the numeral 11 does *not* represent the number "eleven", but actually represents the number "two"; since $\ast = 1$, we have that $\ast\ast = 11$, and so

$$11 = \ast\ast = 2 = \text{“two”}.$$

If, instead, we really want to write the number "eleven", then we should probably use the boxes:

$$\boxed{1}1 = \boxed{\ast}\ast = \ast\ast\ast\ast\ast\ast\ast\ast\ast\ast\ast = \text{“eleven”}$$

This system is getting pretty close to the standard Hindu-Arabic numeral system. Indeed, we can write numbers like "one thousand two hundred thirty-four" as $\boxed{\boxed{\boxed{1}}}\boxed{\boxed{2}}\boxed{3}4$.

Exercise: write down the following numbers in the `new` **new** *new* tally system:

- "four"
- "fifty-five thousand two hundred sixty-eight"
- "nine hundred eighty-seven thousand six hundred fifty-four"

2.2 Positional Systems

At this point, the main difference between our **Final Tally system, or FTS for short,** and the standard, positional **Hindu-Arabic Numeral System, or HANS for short,** is that our FTS admits multiple valid numerals for each number, while the HANS only admits one valid numeral for each number. For instance,

- In HANS, the numeral 1234 is the only way to write the number "one thousand two hundred thirty-four".
- However, in FTS, we can write this number with any of the following numerals:

$$\boxed{\boxed{\boxed{1}}}\boxed{\boxed{2}}\boxed{3}4 = 4\boxed{3}\boxed{\boxed{2}}\boxed{\boxed{\boxed{1}}} = \boxed{\boxed{1}}3\boxed{2}\boxed{\boxed{1}}1\boxed{\boxed{\boxed{1}}}\boxed{1} = 43\boxed{\boxed{8}}7\boxed{5}58\boxed{\boxed{2}}9\boxed{76}99.$$

  Obviously, recognizing these different numerals as representing the same number is practically impossible at worst and extremely laborious at best. That is one practical reason why HANS has dominated. (However, there are advantages to having some flexibility in the way we write our numerals, also, which we will see shortly...)

It turns out that it's always possible to write any natural number with one glyph (digit 1-9) per box, with at most one box of each size. Choosing this convention as the standard allows us to pick $\boxed{\boxed{\boxed{1}}}\boxed{\boxed{2}}\boxed{3}4$ as the official "standard form" in FTS–and this is what HANS does, too.

At this point, there is only one thing preventing us from simply omitting the boxes from the standard form of FTS and obtaining HANS–consider the numbers and their standard form FTS numerals:

- "eleven" $= \boxed{1}1$. If we were to simply omit the boxes, then we obtain the correct HANS numeral, 11. Nice.
- "one hundred one" $= \boxed{\boxed{1}}1$. However, if we were to simply omit the boxes here, then **would not** obtain the correct HANS numeral; instead, we would also obtain 11.
- "one hundred ten" $= \boxed{\boxed{1}}\boxed{1}$. Similarly, simply omitting the boxes here still yields 11, the wrong HANS numeral.
- "one thousand one" $= \boxed{\boxed{\boxed{1}}}1$. Likewise...

  $\vdots$

The thing we are missing is the *positional placeholder*. That is, we should allow empty boxes. That is, instead of writing, for example $\boxed{\boxed{1}}1$ for "one hundred one", we should instead write $\boxed{\boxed{1}}\square 1$, with an empty box. The symbol we use in HANS for this placeholder is 0. Thus, we have the following corrections to the above numerals:

- "eleven" $= \boxed{1}1$. If we were to simply omit the boxes, then we obtain the correct HANS numeral, 11. Nice.
- "one hundred one" $= \boxed{\boxed{1}}\square 1$. Now, when we omit the boxes, we get 101, the correct HANS numeral for "one hundred one", as desired!
- "one hundred ten" $= \boxed{\boxed{1}}\boxed{1}$. In this case, there is no empty box at the end, so we must add the placeholder for the missing unboxed glyph at the end of this FTS numeral to obtain the correct HANS numeral, 110.
- "one thousand one" $= \boxed{\boxed{\boxed{1}}}\boxed{\square}\square 1$. Likewise...

  $\vdots$

Therefore, **we can understand positional number systems, including HANS, as a kind of standardized tally system (in this case, FTS), together with positional placeholders.** HANS is more useful for communication with *others* since there is one **and only one** numeral for each number, whereas there are multiple possible numerals for each number in FTS...

3 Computation

However, as any writer will tell you: When we write, we are in dialogue with *ourself*, and writing facilitates that dialogue. Analogously, good mathematical notation facilitates computations. And FTS is more useful than HANS in this respect because we have flexibility to write numerals in a convenient form. To this end, we study the addition of numbers utilizing our FTS system.

3.1 Addition in a Tally System

What is $*$ plus $*$? It's just $**$, of course! The main advantage of a tally system is that it makes addition trivial. What is $**$ plus $**$? $****$. Easy!

What is $\boxed{*}*$ plus $\boxed{*}*$? Well, it's $\boxed{*}*\boxed{*}*$. That's technically correct. However, we want to put that in standardized form. So, we rewrite it!

$$\boxed{*}* + \boxed{*}* = \boxed{*}*\boxed{*}* = \boxed{*}\boxed{*}** = \boxed{**}**.$$

Using the correspondence between FTS and HANS, we can read off a basic arithmetical fact: "eleven plus eleven" is $\boxed{1}1 + \boxed{1}1 = \boxed{*}* + \boxed{*}* = \boxed{**}** = \boxed{2}2$ is "twenty-two"; in particular, writing the same numerical fact in HANS, we have that $11 + 11 = 22$.

This tally system has a number of *rewriting rules*, which are fairly intuitive; I haven't explicitly mentioned them up until this point, but I will now.

3.2 Rewriting Numbers in FTS

We have already seen instances representing all of the essentially different kinds of rewrites we need to consider for FTS, so this section is merely an aside to review and elucidate the underlying general principles.

1. We can use numerals representing the same number interchangeably; this is the most fundamental rewrite rule. Let's see some examples to see how to apply it:

   - We have seen that, in FTS, $55 = \boxed{1}$. So, in particular, we have that $\boxed{1}1 = 551$, and $\boxed{\boxed{2}}\boxed{1}4 = \boxed{\boxed{2}}554$, etc.

   - We have seen that, in FTS, $11 = 2$. So, in particular, we have that $\boxed{1}2 = \boxed{1}11$ and $\boxed{\boxed{2}}\boxed{1}4 = \boxed{\boxed{11}}\boxed{1}4$, etc.

   - We can use multiple instances of this rewrite rule to note that $\boxed{1}2 = 5511$ and $\boxed{\boxed{2}}\boxed{1}4 = \boxed{\boxed{11}}554$.

   - For that matter, we have that $\boxed{55} = \boxed{\boxed{1}}$ and $\boxed{11} = \boxed{2}$, etc.

2. The order of tallies doesn't matter, since it ultimately just boils down to a long list of $*$ symbols. In particular, you can write the individual glyphs in a numeral in any order. Let's see some examples to see how to apply this rule:

- In FTS, we have that $\boxed{1}1 = 1\boxed{1}$. Indeed, both numerals represent the number “eleven”.
- In FTS, we have that $\boxed{\boxed{2}}\boxed{1}4 = \boxed{\boxed{2}}4\boxed{1} = \boxed{1}\boxed{\boxed{2}}4 = \boxed{1}4\boxed{\boxed{2}}\boxed{1} = 4\boxed{\boxed{2}}\boxed{1} = 4\boxed{1}\boxed{\boxed{2}}$. Indeed, all of these numerals represent the number “two hundred fourteen”.

3. Applying a box to a numeral is the same as apply a box to each tally in the numeral individually. For instance, $\boxed{***} = \boxed{*}\boxed{*}\boxed{*}$. Let’s see some more examples to see how to apply this rule in FTS:

   - In FTS, we have that $\boxed{2} = \boxed{11} = \boxed{1}\boxed{1}$.
   - Likewise, $\boxed{5}\boxed{5} = \boxed{55} = \boxed{\boxed{1}}$.
   - And also, $\boxed{\boxed{1}}\boxed{\boxed{1}} = \boxed{\boxed{1}\boxed{1}} = \boxed{\boxed{11}} = \boxed{\boxed{2}}$.

Exercise: identify the numbers represented by each of the above FTS numerals.

3.2.1 A Glimpse into the Future

The typical approach for stating rules that apply in various situations, such as rewrite rules, is to use *variables*. The concept of a variable, when it is first being learned, is often presented as somewhat mysterious or vague, but that need not be the case; we will treat variables in detail later. For now, though, I will say something about their utility.

Whereas before we stated our rewrite rules by analogy/example, variables allow us to capture many–potentially infinitely many–cases in a single formula. In this way, we can be more precise about exactly which situations a given rule applies. An approach based on variables has all the specificity of forming an explicit list of situations, like “$\boxed{1}2 = 2\boxed{1}$, $\boxed{3}4 = 4\boxed{3}$”, while covering an arbitrarily large set of possibilities–something which would be infeasible without variables.

That’s enough of a digression. Back to addition...

3.3 An Algorithm for Addition

Our strategy for adding two numbers represented by HANS numerals is:

1. Write each number, represented as HANS numerals, as FTS numerals.
2. Add the numbers in FTS by sticking the two numerals together, a process called *concatenation*.
3. Rewrite the sum as a FTS numeral in standard form.
4. Convert the resulting standardized FTS numeral into a HANS numeral.

We have already seen how to do all these steps except for the rewriting step; we will explore that step now.

3.3.1 An Instructive Example

To that end, we consider an example, "twelve thousand three hundred forty-five" plus "six thousand seven hundred eighty-nine" is

$$\boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{2}}}\,\boxed{\boxed{3}}\,\boxed{4}5\,\boxed{\boxed{\boxed{6}}}\,\boxed{\boxed{7}}\,\boxed{8}9 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{2}}}\,\boxed{\boxed{\boxed{6}}}\,\boxed{\boxed{3}}\,\boxed{\boxed{7}}\,\boxed{4}\,\boxed{8}59 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{26}}}\,\boxed{\boxed{37}}\,\boxed{48}59$$

$$= \boxed{\boxed{\boxed{\boxed{*}}}}\,\boxed{\boxed{\boxed{**\quad ******}}}\,\boxed{\boxed{***\quad *******}}\,\boxed{****\quad ********}*****\quad *********$$

$$= \boxed{\boxed{\boxed{\boxed{*}}}}\,\boxed{\boxed{\boxed{********}}}\,\boxed{\boxed{\boxed{*}}}\,\boxed{\boxed{*}**}\,\boxed{*}**** = \boxed{\boxed{\boxed{\boxed{*}}}}\,\boxed{\boxed{\boxed{*********}}}\,\boxed{\boxed{*}}\,\boxed{***}****$$

$$= \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{9}}}\,\boxed{\boxed{1}}\,\boxed{3}4$$

is "nineteen thousand one hundred thirty-four". Or, in HANS, $12345 + 6789 = 19134$. The procedure used in this example is pretty straight-forward:

1. Rearrange to bring together boxes of the same size.
2. Add up numerals within boxes of the same size, starting with the smallest:
   (a) Revert to "$*$" symbols.
   (b) Count.
   (c) Express totals as numerals using digits: $1, 2, 3, 4, 5, 6, 7, 8, 9$.
3. Collect the *overflow* into the next-highest box.
4. Repeat for boxes of all all sizes.

In this example, I didn't assume any arithmetical facts whatsoever–not even the standard addition tables in single digits. Instead, I converted numerals back into "$*$" symbols and simply counted. Indeed, if you can count to "ten" (for instance, on your fingers), then you don't really *need* to memorize addition tables.[9] However, many people have gone through the trouble of memorizing how to add single digit numbers quickly. So, if you have that information, then you can skip some counting when you add numbers.

I will repeat the same computation, but this time instead of using "$*$" symbols, I will only use the $1, 2, 3, 4, 5, 6, 7, 8, 9$ digits:

$$\boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{2}}}\,\boxed{\boxed{3}}\,\boxed{4}5\,\boxed{\boxed{\boxed{6}}}\,\boxed{\boxed{7}}\,\boxed{8}9 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{2}}}\,\boxed{\boxed{\boxed{6}}}\,\boxed{\boxed{3}}\,\boxed{\boxed{7}}\,\boxed{4}\,\boxed{8}59 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{26}}}\,\boxed{\boxed{37}}\,\boxed{48}59$$

$$= \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{8}}}\,\boxed{\boxed{\boxed{1}}}\,\boxed{\boxed{1}2}\,\boxed{1}4 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{81}}}\,\boxed{\boxed{1}}\,\boxed{21}4 = \boxed{\boxed{\boxed{\boxed{1}}}}\,\boxed{\boxed{\boxed{9}}}\,\boxed{\boxed{1}}\,\boxed{3}4$$

Indeed, this is just the standard garden-variety addition algorithm. You will notice that it requires being able to add single digit numbers. For instance, you need to know that $4 + 8 = \boxed{1}2$ (or 12 in HANS). Of course, the benefit is that you get to do step (2) of the above algorithm in one step instead of three sub-steps.

[9] This, perhaps, runs contrary to popular belief. Oh well.

3.3.2 Getting Comfortable: Comparison to the Familiar Algorithm

In some sense elaborated above, the algorithm we discovered is actually essentially identical to the standard addition algorithm except for cosmetic changes (where/how to write numbers). However, the conceptual basis is different since we are relying upon a chain of understanding down to the level of tally systems in order to motivate our steps–whereas, the standard addition algorithm is presented purely as a symbolic manipulation (since its purpose is to efficiently compute correct answers, rather than to explain what's happening). Moreover, cosmetic changes–especially changes as profound as the ones we've made here–can be disorienting; I will do a small example with a side-by-side comparison.

Standard addition algorithm:

$$\begin{array}{rrr} & 1 & \\ & 6 & 6 \\ + & 6 & 6 \\ \hline & & 2 \end{array} \Longrightarrow \begin{array}{rrr} 1 & 1 & \\ & 6 & 6 \\ + & 6 & 6 \\ \hline 1 & 3 & 2 \end{array}$$

The purpose of this algorithm is to efficiently compute the sum.

Digits-only tally-algorithm:

$$\begin{aligned} \boxed{6}6 + \boxed{6}6 &= \boxed{6}6\boxed{6}6 \\ &= \boxed{6}\boxed{6}66 \\ &= \boxed{66}66 \\ &= \boxed{66}\boxed{1}2 \\ &= \boxed{661}2 \\ &= \boxed{\boxed{1}3}2 \\ &= \boxed{\boxed{1}}\boxed{3}2 \end{aligned}$$

The purpose of this algorithm is to be a bridge between the other two algorithms.

"$*$" symbol tally-algorithm:

$$\begin{aligned} \boxed{6}6 + \boxed{6}6 &= \boxed{6}6\boxed{6}6 \\ &= \boxed{6}\boxed{6}66 \\ &= \boxed{66}66 \\ &= \boxed{66} \ast\ast\ast\ast\ast\ast\ \ast\ast\ast\ast\ast\ast \\ &= \boxed{66}\boxed{\ast}\ast\ast \\ &= \boxed{66}\boxed{1}2 \\ &= \boxed{661}2 \\ &= \boxed{\ast\ast\ast\ast\ast\ast\ \ast\ast\ast\ast\ast\ast\ \ast}2 \\ &= \boxed{\boxed{\ast}\ast\ast\ast}2 \\ &= \boxed{\boxed{1}3}2 \\ &= \boxed{\boxed{1}}\boxed{3}2 \end{aligned}$$

The purpose of this algorithm is to make clear what's happening to the numbers and how our written mathematical notation reflects that numerical reality.

When you compute sums, by all means, use the standard algorithm. However, understanding the reasoning behind the "$*$" symbol tally-algorithm is helpful on a conceptual level, get a feeling for what natural numbers are and how they work.

Compute the following sums (which are all written in HANS):

- $1 + 2$
- $10 + 20$
- $10 + 2$
- $987 + 76$
- $1234 + 567800$