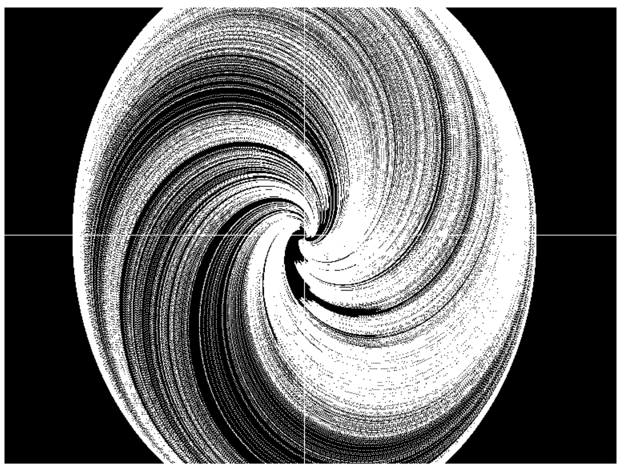
Reasonable Decimal Arithmetic

Alain Schremmer

Reasonable Decimal Arithmetic

(Lean Edition)



FreeMathTexts.org

Version 4.1 — Tuesday 25th August, 2020

[1]Educologists who deem the footnotes “inflammatory” need only uncomment “renewcommand-footnote” in the ADJUSTMENTS TO DOCUMENT to turn them off.

To Alain and Julien.

Contents

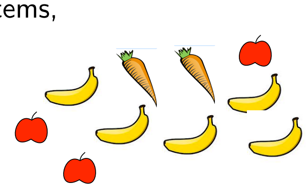
Decimals for Numbers of Items

We begin by introducing and discussing, in the simplest possible setting, the *concepts* that are at the heart of the DECIMAL SYSTEM, the modern symbolic-system for counting numbers and we then construct the DECIMAL SYSTEM itself.

1 Representing A Collection

1.

EXAMPLE 0.1.
The following real world items,



are not all of the *same kind* and so do *not* make a *collection*.

LANGUAGE 0.1
It will be very convenient to show *collections* as follows:

- When using *pictures* as substitutes for the real world items, we will enclose the *pictures* within a line.
- When using *words* as substitutes for the real world items, we will

picture
stand
number phrase
information
specify
qualitative information
denominator
quantitative information
numerator
count
progression

enclose the *words*, separated by **commas**, between **braces**.

EXAMPLE 0.2.
The real world items make up the collection In other words, the real world items APPLE APPLE APPLE make up the collection {APPLE, APPLE, APPLE}

2. In order to *represent* on paper a real world *collection*, we write a **number phrase** that includes the **information** necessary to **specify** that *collection*:

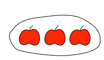
- The *kind* of items in the collection (**qualitative information**) which we represent on paper by a **denominator**[2] that is a name usually associated with the items.

EXAMPLE 0.3.
Given the following collection of real world items,

we can use for *denominator* the name of the President whose picture is on all of them, namely **Washington**.

EXAMPLE 0.4.

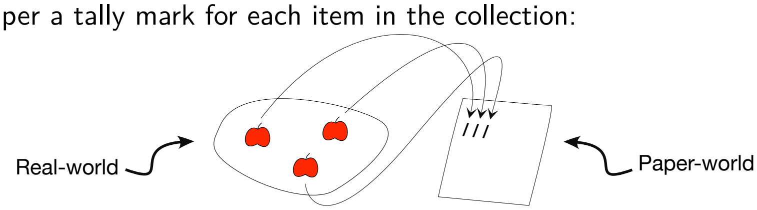
- The *number* of items in the collection (**quantitative information**) which we represent by a **numerator** which we get by **counting** the items in the collection: we point at each and every item in the collection while reciting the **progression** that we memorized when we were children. The number of items in the collection is what we say when we point at the last item in the collection.[3]

EXAMPLE 0.5.
To count the items in the collection  we point successively at each and every  while reciting the progression: $\xrightarrow{\text{one, two, three}}$.
Since what we said last is "three", the number of items is THREE and

[2] This is precisely the point where, by ignoring denominators in the name of "abstraction", Educologists give birth to "math anxiety".
[3] Educologists will surely agree that this is not the place to introduce *equipotence*.

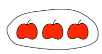
number-phrase
sort
bunch of collections
bunch-list
square brackets
[ ]

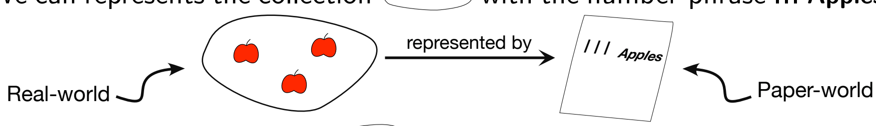
children would represent the *number* of items with a *numerator* by writing on paper a tally mark for each item in the collection:

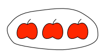


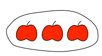
This is in fact essentially what the Babylonians started doing about 3000 BC (See http://en.wikipedia.org/wiki/Babylonian_numerals.)

Any *collection* of real world items can thus be represented by a **number-phrase**, that is by a *numerator* together with a *denominator*.

Example 0.6.
We can represents the collection  with the number-phrase **III Apples**:



In Thailand, the collection  would be represented by the *number-phrase*
๓ แอปเปิล where ๓ is the *numerator* and แอปเปิล is the *denominator*.

In Vietnam, the collection  would be represented by the *number-phrase*
ba táo where **ba** is the *numerator* and **táo** is the *denominator*.

3. When the real

i. We **sort** the bunch according to the *different* kinds of real world items that are in the bunch and we then collect the items of the *same* kind into collections. We will say that we have sorted the original bunch of real world items into a **bunch of collections**.

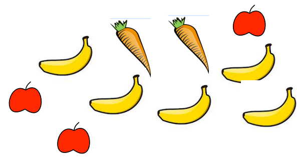
ii. We can then represent each collection by a number-phrase,

iii. Now, we can represent the *bunch of collections* by a **bunch-list**[4], that is by enclosing the number-phrases that represent each collection, separated by *commas*, within two **square brackets**, [ ].

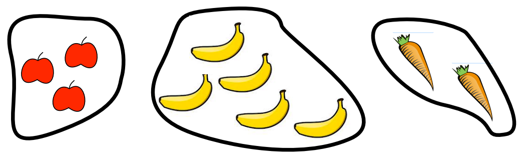
[4]No doubt, Educologists will have realized that we cannot use the term *vector* here because bunch-lists can have different lengths.

basic numerators
digit

Example 0.7.
Represent the bunch of real world items



i. We sort the real world items into the following *bunch* of *collections*:



ii. Using tally marks to write the numerators and English nouns to write the denominators, we can represent each collection by a number phrase:

III Apples, **IIIII Bananas**, **II Carrots**

iii. Now we can represent the original bunch of real world items by the *bunch-list*

[III Apples, IIIII Bananas, II Carrots]

2 Representing Large Collections

Representing collections involving a large number of real world items is a major problem if only because, when the number of items in the collection gets to be large, so does the number of tally marks on paper and it gets more and more difficult to distinguish number-phrases.

Example 0.8.
Using *tally marks* for the numerator, it is not easy to see how many items there are in the collection represented by the number-phrase

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII Apples

and how different, if at all, it is from the collection represented by the number-phrase

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII Apples

The modern approach is up front a number of **basic numerators**, usually called **digits**, and then somehow make smaller collections that can each be represented with a *digit*.

This, though, will require an extra digit, will of course use the symbol 0 which, in fact, was a late invention. See https://en.wikipedia.org/wiki/zero Altogether, the number of declared digits *plus one* (for *zero*) is called the **system-base**.

0
system-base
bundle
leftover item
postfix

1. More precisely, say we have some large collection of items and only a small number of *digits including* 0, call that number N. Then:

i. We **bundle** the *items* at the rate of N *items* per **bundle**. In other words,

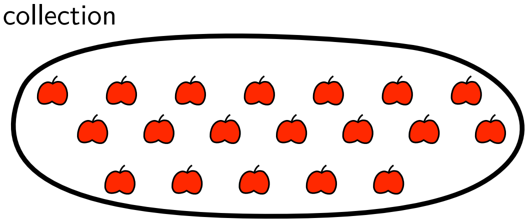
$$\boxed{\text{Bundling-rate} = \text{System-base}}$$

Once we are done bundling the items into *bundles of* N *items*, we are left with less than N **leftover items** and we can therefore use a *digit* as numerator in the number phrase that represents the collection of leftover items.

ii. We then deal with the *bundles of* N *items* exactly the same way: we bundle the *bundles of* N *items* and we can then *represent* the *number* of *leftover bundles of* N *items* with a *digit*. For a denominator, we use a **postfix** which consists of as many times the *system base* as we bundled.

iii. And so on until there is nothing more to bundle.

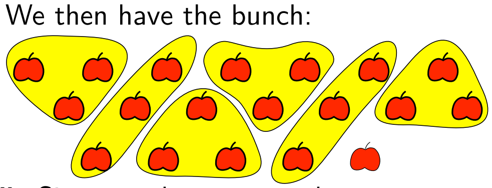
EXAMPLE 0.9.
To represent the collection



using only the THREE digits, 0, 1, 2.

i. Since we have more than TWO APPLES, we must bundle the APPLES at the rate of THREE APPLES for ONE BUNDLE OF THREE APPLES.

We then have the bunch:



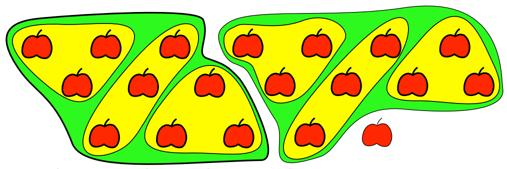
namely the bunch consisting of:

- ONE leftover APPLES
- SIX BUNDLES OF THREE APPLES

ii. Since we have more than TWO BUNDLES OF THREE APPLES, we must bundle the BUNDLES OF THREE APPLES at the rate of THREE BUNDLES OF THREE APPLES for ONE BUNDLE OF THREE BUNDLES OF THREE APPLES.

bundle-numerators
list-phrase

We then have the bunch



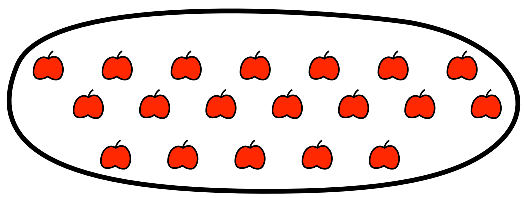
namely the bunch consisting of:

- ONE leftover APPLE
- NO leftover BUNDLE OF THREE APPLES
- TWO BUNDLES OF THREE BUNDLES OF THREE APPLES

➠ The number phrase that represents NO leftover BUNDLE OF THREE APPLES uses the postfix THREE- and the number phrase is 0 THREE-**Apples**

➠ The number phrase that represents the leftover TWO BUNDLES OF THREE BUNDLES OF THREE APPLES uses the postfix THREE-THREE- and is 2 THREE-THREE-**Apples**

iii. Since there is nothing more to bundle, we can represent the original collection



by the *bunch list*

[2 THREE-THREE-**Apples**, 0 THREE-**Apples**, 1 SINGLE-**Apple**]

2. Since the number-phrases in the *bunch-list* all involve the same *denominator* we can write the name of the items just once to the right of the remaining list of **bundle-numerators** and use what we will call a **list-phrase**.

EXAMPLE 0.10.
Instead of writing the *bunch-list*

[2 **THREE-THREE-Apple**, 0 **THREE-Apple**, 1 **SINGLE-Apple**]

we will write the *list-phrase*

$\underbrace{[\ 2\ \textbf{Three-Three},\ 0\ \textbf{Three},\ 1\ \textbf{Single}\ ]}_{\text{Numerator}}\ \underbrace{\textbf{Apples}}_{\text{Denominator}}$

A *list-phrase* is a much better representation of a collection with a large number of items in that it separates *quantitative* information from *qualitative* information.

3 Base TEN

TEN
decimal system

In the examples of the previous section, we used THREE as system-base because the purpose there was only to demonstrate how things worked[5].

1. Historically, a variety of system-bases were invented (See http://en.wikipedia.org/wiki/Gross_(unit)) but eventually the entire world ended up using the Hindu-Arabic system-base **TEN**.(See http://en.wikipedia.org/wiki/History_of_the_Hindu-Arabic_numeral_system.)

Tally marks	Digits
I	1
II	2
III	3
IIII	4
IIIII	5
IIIIII	6
IIIIIII	7
IIIIIIII	8
IIIIIIIII	9

xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx system also known as the **decimal system** (See http://en.wikipedia.org/wiki/Decimal).

EXAMPLE 0.11.

aaaaaaaaaaa	bbbbbb bbbbbb bbbbbbbbbbb

2. yyyyyyyy

3. zzzzzzz

[5]Educologists of course know that Z. P. Dienes always used to *start* his workshops with THIRD graders, base-THREE ARITHMETIC BLOCKS and the digits 0, 1, 2.

What is important is the real-world, that is physics, but it can be explained only in mathematical terms.

Dennis Serre[1]

[...] mathematics as a precise language for expressing relationships among quantities in the real-world [...].

Carver Mead[2]

margin
index
real-world
entity
relationship
information
action

Chapter 1

Symbolic Systems

Language 1.1 Just as in any Science, Technology and Engineering, the answer in Mathematics to the question "You know what I mean?" is always "No!" because we need to agree, precisely and exactly, on *what* we are talking about. In particular, the reader will have to know the meaning of the words that are boldfaced as they are introduced in this text. To help the reader who, later on, might not be too sure about such a word, the page where it was introduced can always be found in the **index** at the end of the boook and the word will also appear at the top of the **margin** on the page where it was first introduced.

Contrary to what most people think, Mathematics in general, and Arithmetic in particular, originate in the **real-world** and deal with real-world **entities** and the **relationships** among real-world entities. However, in order to:

- Communicate about the real-world, that is share **information** about the real world,
- Investigate how the real-world *works*,
- Plan in advance our **actions** on the real-world (because acting on the

[1]Bulletin of the American Mathematical Society, Vol 47 Number 1 Pages 139-144
[2]Foreword to *Street-Fighting Mathematics* by Sanjoy Mahajan, The MIT Press.

language
represent
word
vocabulary
noun
verb
phrase
noun phrase
verb phrase

real-world without thinking ahead usually has very bad consequences), we need a **language** to **represent** the real world on *paper*.

And, indeed, there are all sorts of languages: anywhere between 3000 and 8000 spoken languages, hundreds of sign-languages, etc. There are written languages, pictorial languages, secret and non-secret codes, bar codes, computer languages, etc. (See http://en.wikipedia.org/wiki/Language.)

Nevertheless, as we will see, MATHEMATICS requires a language of its own.

1 What Languages Are Made Of

Very, very roughly, most languages are constructed as follows:

1. The building blocks of the language are **words** listed in a **vocabulary** that must be available to all users of the language.

EXAMPLE 1.1. Of the following:
- word, wood
- woord, wod

the first two are English words but the other two are not.

The two main kinds of *words* that we will be using are:
- **Nouns** to represent real-world *entities*,
- **Verbs** to represent *relationships* among real-word entities.

EXAMPLE 1.2. Of the following:
- apple, triangle
- eats, intersects

the first two are *nouns* and the other two are *verbs*.

2. Quite often, though, we do not have single ready-made words to represent real-word entities or to represent a relationship among real-word entities and we will have to use a **phrase**, that is "a small group of words standing together as a conceptual unit" (http://www.oxforddictionaries.com/us/definition/american_english/phrase). More precisely, we will use:
- **Noun phrases**, that is groups of words that work the same way single nouns do, to represent real-world entities,
- **Verb phrases**, that is groups of words that work the same way single verbs do, to represent relationships among real-word entities.

We will place **hyphens** between the words that make up a phrase to emphasize the fact that these words are to be read together as a whole.

hyphen
sentence
grammatical rule
grammar
state
statement
meaning
meaningless
TRUE
occur
FALSE

EXAMPLE 1.3. Of the following:
- ten-dollar-bill, mathematics-teacher
- takes-apart, is-the-daughter-of

the first two are *noun phrases* and the other two are *verb phrases*.

3. In order to communicate we use **sentences**, that is groups of *words* or *phrases* constructed according to **grammatical rules** listed in a **grammar** that must be available to all users of the language.

EXAMPLE 1.4. Of the following:
- The hamburger paints the ocean with the mountains.
- Dog fox lazy quick brown the jumps the over.

the first is an English *sentence* but the second is not an English sentence.

4. However, a *sentence* may or may not be **stating** something about the real-world.
A *sentence* that states something about the real-world is called a **statement** and what it states about the real-world is called the **meaning** of the sentence.
A sentence that does not state something about the real-world is said to be **meaningless**.

EXAMPLE 1.5. Of the following two *sentences*:
- The spaghetti sauce is too hot to be eaten with a spoon.
- The rose understands the navy with a table.

the first is a *statement* (about the real-world) but the second is *meaningless* since it is *not* a statement (about the real-world).

5. A sentence is then:
– **TRUE** if what it states about the real-world **occurs** in the real-world,
– **FALSE** if what it states about the real-world does *not* occur in the real-world.

EXAMPLE 1.6. The following are both (famous) *statements* about the real-world:

procedure
simulate
process
native language
object language
metalanguage

- The moon is made of green cheese.
- Humans are the only featherless bipeds.

However,

> The first statement is FALSE because, as Armstrong checked on July 20, 1969, the moon is *not* made of green cheese.

while

> The second statement is TRUE because, even though there are many species that are featherless and many species that use only two legs for walking, it so happens that humans are the only ones to be both featherless and bipedal.

6. Our main goal in this text will be, using *sentences*, to develop **procedures** to **simulate** on paper real-world **processes**.

Example 1.7. Suppose we have to carry out the real-world *process* of handing out seventeen dollars to each one of two thousand six hundred and forty eight people. A preliminary issue then is whether we have the cash to carry out this real-world process.
To find out, we *represent* on paper both how many dollars per person and how many persons and we then use the paper-world *procedure* called *multiplication* to figure out on paper how many dollars we will need in order to carry out the *process* in the real world.

2 Learning Languages

Learning a language is nowhere as simple as one might think.

1. The language that each one of us first learned as a child, our **native language**, we learned by directly associating *nouns* with the real-world *entities* that the nouns represent. But how children learn *verbs* to represent real-world *relationships* among real-world entities still isn't fully understood.

2. A second language is usually not learned the same way as a native language, that is not by direct association with the real-world, but through the language that we already know. So, learning a second language automatically involves *two* languages which we will need to keep separate:

- The **object language**, which is the language that we want to learn,
- The **metalanguage**, which is the language we already know and which we use to learn the object language.

pictures
substitute

EXAMPLE 1.8. For an American learning Spanish, the *object language* is Spanish and English is the *metalanguage* but for a Spaniard learning English, the *object language* is English and Spanish is the *metalanguage*.

3. In a book, the *metalanguage* is used for several different purposes:

- To describe and discuss the *real-world* to be to be represented on paper with the object language. In particular, we will use the *metalanguage* to describe and discuss:
  - The real-world *entities* to be represented on paper,
  - The real-world *relationships* among real-world entities to be represented on paper;

but also

- To describe and discuss features of the *object language* itself,

and, more generally,

- To discuss matters.

3 In This Book

We will be using English as our metalanguage.

1. Since we will not be able to exhibit the real-world items that we will want to represent on paper, we will usually use **pictures** as **substitutes** for the real-world entities. However, as even using pictures will not always be possible, we will also use as substitutes English words which will then be printed in BOLD-FACED ITALICS. So, in this book, words printed in bold-faced italics will belong to the metalanguage.

EXAMPLE 1.9. Say we intend to discuss the paper-representation of *real-world* one-dollar bills. Since we cannot exhibit these *real-world* one-dollar bills, we will use *pictures* as a *substitute* for these real-world one-dollar bills, for instance

However we will also use as *substitutes* for these real-world one-dollar bills the English words (in bold-faced italics),

ONE-DOLLAR BILL ONE-DOLLAR BILL ONE-DOLLAR BILL

shorthand
boldfaced
margin
index
natural language

NOTE 1.1
It is very important to distinguish *nouns* from *substitutes*:

- *nouns* are intended to *represent on paper* the real-world items.

while

- *substitutes* are intended to *stand for* the real-world items and will be used only when we cannot use *pictures* to do so.

2. When there will be too many real-world entities for us to use pictures or English words as substitutes, we will use English numbers in SMALL CAPS as a **shorthand**. So, in this book, numbers spelled out in small caps belong to the metalanguage.

EXAMPLE 1.10. We will use FIVE APPLES as a shorthand for APPLE APPLE APPLE APPLE APPLE as well as FIVE  as a shorthand for .

3. In order to make it easier to find in this book the place where words, both in the object language and in the metalanguage, appear for the first time and are explained, this book uses a standard method: These words are

- **boldfaced** the first time they appear in this book and are explained,
- printed in the **margin** of the page where they appear for the first time and are explained,
- listed in the **index** at the end of the book along with the number of the page where they appear for the first time and are explained.

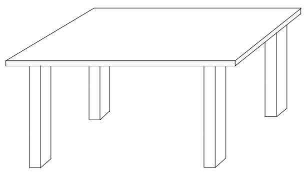
4 Natural Languages

A **natural language** is the kind of language we use most of the time, routinely, and rather loosely. Natural languages depend on place and time.

EXAMPLE 1.11.

specialized languages

The real-world entity whose *picture* is



is represented by different *words* depending on the natural language we are using:

Language	Word
Chinese	表
English	table
Finnish	taulukko
French	table
German	Tisch
Greek	τραπέζι
Italian	tavola
Japanese	食卓
Latin	tabula
Russian	стол
Spanish	mesa
Vietnamese	bān

How natural languages were born and how they then evolved and interacted with one another is buried in time and investigated inETYMOLOGY (See http://en.wikipedia.org/wiki/Etymology).

EXAMPLE 1.12.

5 Specialized Languages

While natural languages usually serve us tolerably well for general communication, they have severe limitations.

1. People working in any trade need **specialized languages** because:

– Workers need to know exactly what they are to do,

EXAMPLE 1.13. We cannot just tell an architect "Design me a building". Architects need *specifications*. In turn, architects cannot just give pictures to construction workers. Construction workers need at least *blueprints*.

– Workers need to communicate with each other while doing the work.

EXAMPLE 1.14. Imagine the likely consequences of two persons working with some dangerous machinery and one telling the other: "Push the make-the-case

gizmo which is on the whatchamacallit next to the doodad on the other side of the doohickey."

2. Sometimes, specialized languages use made-up words but most of the time they use common words with a meaning special to the trade.

EXAMPLE 1.15. When an *electrician* asks for a pancake, that's not because s/he is hungry but because s/he needs a junction box the thickness of drywall. When a *carpenter* asks for *drywall*, s/he is not likely to be asking for a wall that is not wet. When a *mason* asks for a *hawk*, s/he is not likely to be asking for a bird.

EXAMPLE 1.16. "[The actor] must learn the theatre's special vocabulary. Partly technical, partly slang, much of it is standardized on the English-speaking stage. As a working actor, you must be familiar with this language, just as a mechanic must know the names of his tools or a surgeon the names of her instruments." (From *Acting is Believing* by Charles McGaw, Kenneth L. Stilson, Larry D. Clark.)

EXAMPLE 1.17. "If you look up the meaning of the word "theory" in the dictionary, it is described as being a synonym for words like "proposition", "hypothesis", or even "speculation". In contrast, a scientific theory is an established body of knowledge about a certain subject, supported by observable facts, repeatable experiments, and logical reasoning. A theory in science is a formal explanation of some aspect of the natural world, tested and verified by careful observation and experimentation. A good theory is one that also produces accurate and useful predictions." (From https://plus.maths.org/content/evolution-its-real-gravity?nl=0.)

6 Mathematical Language

In the mathematical trade, the situation is exactly the same as in any other trade except even more so because, in mathematics, just as if we were in a court of law, we *always* have to **make-the-case** that:

– The *mathematical sentences* we write are TRUE, that is are statements about the real-world whose meaning occurs in the real-world,

– The *mathematical procedures* we develop are **correct**, that is simulate the real-world processes.
– **Noun-symbols** to represent real-world *entities*,
– **Verb-symbols** to represent *relationships* among real-world entities,

together with

– **Logical-symbols** to represent in the symbolic-system the meaning of the English words "and", "or", "not".[3]. We will use &, $\vee$, $\neg$

correct
noun-symbol
verb-symbol
&
$\vee$
$\neg$
symbolic-sentence
code

We can then assemble these symbols to write **symbolic-sentences**. These symbolic sentences will then be TRUE or FALSE depending on whether they make statements about the real-world whose meaning occurs or does not occur in the real-world. We will then say that we have **coded** the information about the real-world into the *symbolic-system*.

EXAMPLE 1.18. Let's say we want to *code* the following real-world situation:

- The entities are three persons: ANDY, BETH and CATHY.
- The relationship is LOVES/HATES as determined by the following table:

LOVES/HATES ↑	ANDY	BETH	CATHY
ANDY	LOVES	LOVES	LOVES
BETH	LOVES	HATES	LOVES
CATHY	HATES	LOVES	HATES

So, for instance, in the real-world according to the above table, we have that ANDY LOVES CATHY, CATHY HATES ANDY, etc.

i. In order to *code* the information about this real-world situation, we may, for instance, use the following symbolic-system:

- The *vocabulary* consists of the following symbols:
  - Noun-symbols: a to represent ANDY, b to represent BETH, c to represent CATHY.
  - Verb-symbol: L to represent LOVES,
  - Logical-symbol: & to represent AND.
- The *grammar* says that the noun-symbols should be on either sides of the verb-symbol—just as in English. (But, for another possible *grammar*, see http://en.wikipedia.org/wiki/Reverse_Polish_notation.)

ii. Then, for instance:

[3]Educologists will surely object to our excluding "if ... then ...". The reason is that, instead of using implication (a binary operation among sentences), we find it more natural to say that a sentence *yields* another sentence (a binary relation among sentences) to represent on paper *logical consequence*.

comment

- The symbolic-sentence

$$bLa$$

means BETH LOVES ANDY and since, in the real-world according to the above table, BETH LOVES ANDY, the symbolic-sentence bLa is TRUE.
- The symbolic-sentence

$$cLa$$

means CATHY LOVES ANDY and since, in the real-world according to the above table, CATHY HATES ANDY, the symbolic-sentence cLa is FALSE.
- The symbolic-sentence

$$cLb \;\&\; cLa$$

means that CATHY LOVES BETH AND CATHY LOVES ANDY and since, in the real-world according to the above table, CATHY LOVES BETH but CATHY HATES ANDY, the symbolic-sentence $cLb \;\&\; cLa$ is FALSE.

NOTE 1.2
In *this* symbolic-system, we would not be able to write a symbolic-sentence to mean that CATHY LOVES BETH OR CATHY LOVES ANDY because the vocabulary in *this* symbolic-system does not include a logical-symbol to represent OR.

Aside from the need to compute with symbols, we will also want to be able to make **comments** in the metalanguage in order to describe, explain and discuss what it is that we are doing within the *symbolic-system*.

EXAMPLE 1.19. Software engineers will tell you that any computer code *must* include comments about what the code is supposed to do that are intended to be read by *humans*—while being ignored by the *computer*. Indeed, in the absence of comments, the code rapidly becomes unusable because after a while the way it works cannot be understood anymore and therefore the code cannot be maintained. So, every computer language includes a way for there to be *comments*.

Finally, we have to agree to write in the *standard* mathematical language. This is because we are trying to write sentences that are TRUE and the only way to make sure that our sentences are TRUE is to submit to others the *case* we made for their truth. And of course, if we do not write in the standard mathematical language, nobody is going to look at, even less debate, our case and so we will never know if the sentences we wrote are TRUE or FALSE. See http://en.wikipedia.org/wiki/Peer_review

declared
quantity
item
number

However, mathematicians do allow deviations from and additions to the standard mathematical language because:

i. A symbolic-system for dealing with all of mathematics would be so big as to be totally unusable,

and so,

ii. Given a particular purpose, parts of the *standard* mathematical language have to be re-cycled and, more generally, deviations are often unavoidable.

Of course, as in computer programming, any deviation from and/or addition to the standard mathematical language has to be **declared** *up front*. And a look at any *real* mathematical text will show how often mathematicians use the word *definition*.

7 Symbolic Systems for Arithmetic

As opposed to GEOMETRY which deals with the *shape* of entities, KINEMATICS which deals with the *speed* of entities, DYNAMICS which deals with the way entities react to *forces*, THERMODYNAMICS which deals with the temperature of entities, etc, ARITHMETIC deals with **quantities** of entities.

1. What immediately complicates matters, though, is that, there are two very different kinds of real world *entities* and, correspondingly, two very different kinds of *quantities* [4]:

entities	*quantities*
items	**numbers**
substances	**amounts**

- *Items* are distinct entities we can deal with *individually* and instead of speaking of a quantity of *items* we speak of a *number*[5] of *items*.

> **LANGUAGE 1.2** While it is often possible in the real world to exhibit the real world elements to be dealt with, this is impossible in a book. So, to signal that we are dealing with *real world* elements, we will

[4] Here, Educologists would of course speak of *multitude* versus *magnitude*. The trouble, though, is that both terms usually connote "big".

[5] Educologists will surely agree with the Greeks that, when all is said and done, the only *real* numbers are the *counting* numbers.

picture
stand
substance
amount

use **pictures** and/or WORDS IN BOLD-FACED ITALICS to **stand** for these elements.

EXAMPLE 1.20.
The pictures  stand for real world elements that can each be dealt with one at a time.

EXAMPLE 1.21. APPLES are *items* we can deal with one at a time and we speak of a *number* of APPLES.

EXAMPLE 1.22. The word STEPS stands for parts in a *procedure* that are elements because they can each be dealt with one at a time.

- A *substance* is an entity we can deal with only *in bulk* and instead of speaking of a quantity of *substance* we speak of an *amount* of *substance*.

EXAMPLE 1.23. SOUP is a *substance* we can deal with only in bulk and we speak of an *amount* of SOUP.

2. Natural languages often use different words when dealing with *numbers* and when dealing with *amounts* but not always.

EXAMPLE 1.24. In English, the distinction between *numbers* and *amounts* is not always made:

	Numbers	Amounts
	Too *many* APPLES	Too *much* SOUP
	A *few* APPLES	A *little* SOUP
	Fewer APPLES than BANANAS	*Less* SOUP than MILK
but:	*More* APPLES than BANANAS	*More* SOUP than MILK
in spite of:	*Many more* APPLES than BANANAS	*Much more* SOUP than MILK

3. As we will see, the distinction between *numbers* and *amounts* is an absolutely fundamental one, especially in sciences and technologies[6]. (See http://www.differencebetween.net/science/mathematics-statistics/

[6]Still, the difference can be difficult to pin down. As Educologists well know, while the

difference-between-number-and-amount/ and http://www.youtube.com/watch?v=F8Do9bDfd1Y).

ancient Greeks had the concept of *number*, they were unable to deal with *amounts* other than via 'ratios and proportions' which effectively stopped the development of ARITHMETIC for many centuries. Which in turn makes one wonder why Educologists should keep insisting on teaching "ratios and proportions".

[...] clay tablets that represented millions of sheep, tonnes of barley, stacks of iron and nearly all the [gold coins] ever minted.

Gore Vidal[1]

kind
collect
collection

Chapter 2

Decimals for Numbers of Items

We begin by introducing and discussing, in the simplest possible setting, the *concepts* that are at the heart of the DECIMAL SYSTEM, the modern symbolic-system for counting numbers and we then construct the DECIMAL SYSTEM itself.

1 Representing A Collection

Real world *items* to be dealt with may or may not be all of the same **kind**.

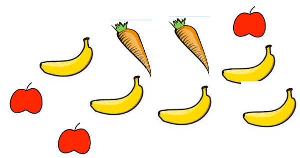
1. Real world *items* that are all of the same kind are easy to deal with because we can **collect** them into a **collection**[2] of *items*.

EXAMPLE 2.1. The following real world items,

[0] *Creation*, Ballantine Fiction, Page 326

[2] As Educologist well know, the only good sets are *collections*.

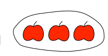
comma
{ }
braces
number phrase
information
specify
qualitative information
denominator



are not all of the *same kind* and so do *not* make a *collection*.

LANGUAGE 2.1 It will be very convenient to show *collections* as follows:

- When using *pictures* as substitutes for the real world items, we will enclose the *pictures* within a line.
- When using *words* as substitutes for the real world items, we will enclose the *words*, separated by **commas**, between **braces**, that is between **{ }**

EXAMPLE 2.2. The real world items  make up the collection . In other words, the real world items APPLE APPLE APPLE make up the collection {APPLE, APPLE, APPLE}

1. In order to *represent* on paper a real world *collection*, we write a **number phrase** that gives all the **information** necessary to **specify** that *collection*:

- The *kind* of items in the collection (**qualitative information**) which we represent on paper by a **denominator**[2] that is a name usually associated with the items.

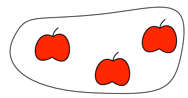
EXAMPLE 2.3. Given the following collection of real world items,

we can use for *denominator* the name of the President whose picture is on all of them, namely **Washington**.

[2]This is precisely the point where, by ignoring denominators in the name of "abstraction", Educologists give birth to "math anxiety".

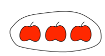
quantitative information
numerator
count
progression
number-phrase

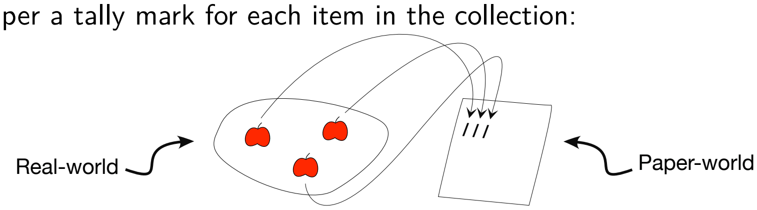
EXAMPLE 2.4. The *nature* of the collection



is that it is a collection of APPLES.

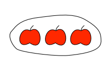
- The *number* of items in the collection (**quantitative information**) which we represent by a **numerator** which we get by **counting** the items in the collection: we point at each and every item in the collection while reciting the **progression** that we memorized when we were children. The number of items in the collection is what we say when we point at the last item in the collection.[3]

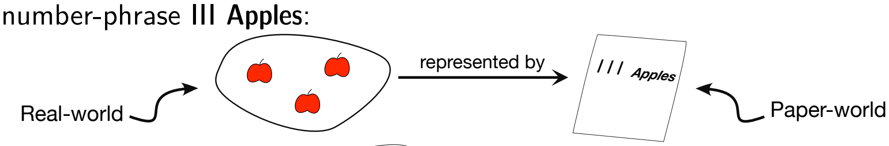
EXAMPLE 2.5. To count the items in the collection  we point successively at each and every  while reciting the progression: $\xrightarrow{\text{one, two, three}}$. Since what we said last is "three", the number of items is THREE and children would represent the *number* of items with a *numerator* by writing on paper a tally mark for each item in the collection:

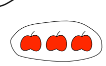


This is in fact essentially what the Babylonians started doing about 3000 BC (See http://en.wikipedia.org/wiki/Babylonian_numerals.)

Any *collection* of real world items can thus be represented by a **number-phrase**, that is by a *numerator* together with a *denominator*.

EXAMPLE 2.6. We can represents the collection  with the number-phrase III **Apples**:

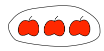


In Thailand, the collection  would be represented by the *number-phrase*

[3]Educologists will surely agree that this is not the place to introduce *equipotence*.

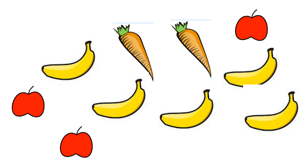
bunch
sort
bunch of collections
bunch-list
square brackets
[ ]

๓ แอปเปิล where ๓ is the *numerator* and แอปเปิล is the *denominator*.

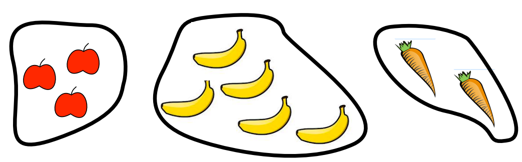
In Vietnam, the collection  would be represented by the *number-phrase*
ba táo where **ba** is the *numerator* and **táo** is the *denominator*.

2. When the real world *items* to be dealt with are *not* all of the same kind, we cannot *collect* them and we will just say that they make up a **bunch** of items. In order to represent on paper a *bunch*, we proceed as follows:
i. We **sort** the bunch according to the *different* kinds of real world items that are in the bunch and we then collect the items of the *same* kind into collections. We will say that we have sorted the original bunch of real world items into a **bunch of collections**.
ii. We can then represent each collection by a number-phrase,
iii. Now, we can represent the *bunch of collections* by a **bunch-list**[4], that is by enclosing the number-phrases that represent each collection, separated by *commas*, within two **square brackets**, [ ].

EXAMPLE 2.7. Represent the bunch of real world items



i. We sort the real world items into the following *bunch* of *collections*:



ii. Using tally marks to write the numerators and English nouns to write the denominators, we can represent each collection by a number phrase:

III Apples, IIIII Bananas, II Carrots

iii. Now we can represent the original bunch of real world items by the *bunch-list*

[III Apples, IIIII Bananas, II Carrots **]**

[4]No doubt, Educologists will have realized that we cannot use the term *vector* here because bunch-lists can have different lengths.

shorthand

> **NOTE 2.1 When + means &:** *Bunch lists* are often written with just + between the number phrases where however + is to be understood as meaning "and". A better way thus is to use & instead of +.

EXAMPLE 2.8.

[III Apples, IIIII Bananas, II Carrots]

is often written

III Apples + IIIII Bananas + II Carrots

where here + is to be understood as meaning "and". So, we would better write

III Apples & IIIII Bananas & II Carrots

2 Representing Large Collections

Representing collections involving a large number of real world items is a major problem if only because, when the number of items in the collection gets to be large, so does the number of tally marks on paper and it gets more and more difficult to distinguish number-phrases.

EXAMPLE 2.9. Using *tally marks* for the numerator, it is not easy to see how many items there are in the collection represented by the number-phrase

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII Apples

and how different, if at all, it is from the collection represented by the number-phrase

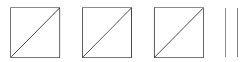
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII Apples

.

1. The first idea that comes to mind is of course to make the numerators more readable by introducing **shorthands** as the strings of tally marks get longer and longer.

EXAMPLE 2.10. When confronted with the problem that eighteen tally marks are not very readable, children readily come up with something like

declare
basic numerators
digit
zero
0
system-base
bundle
leftover item



And something like that is in fact what the Romans did:

String	Roman shorthand
IIIII	V
VV	X
XXXXX	L
LL	C
CCCCC	D
DD	M

EXAMPLE 2.11. In the Roman System, the shorthand for

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII **Apples**

is LV **Apples** and the shorthand for

IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII **Apples**

is LIV **Apples**

(See http://en.wikipedia.org/wiki/Roman_numerals and http://en.wikipedia.org/wiki/Numeral_system).

However, even though the Roman system is fairly efficient for many *communication* purposes, it does not lend itself very well to *procedures* because it requires an ever-increasing number of *new* symbols. The modern approach is to **declare** up front a number of **basic numerators**, usually called **digits**, and then somehow make smaller collections that can each be represented with a *digit*. This, though, will require an extra digit, namely **zero**. We will of course use the symbol **0** which, in fact, was a late invention. See https://en.wikipedia.org/wiki/zero Altogether, the number of declared digits *plus one* (for *zero*) is called the **system-base**. More precisely, say we have some large collection of items and only a small number of *digits including* 0, call that number N. Then:

i. We **bundle** the *items* at the rate of N *items* per **bundle**. In other words,

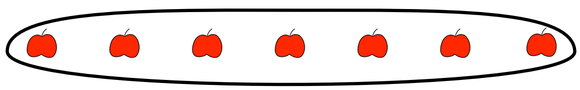
$$\boxed{\text{Bundling-rate} = \text{System-base}}$$

Once we are done bundling the items into *bundles of* N *items*, we are left with less than N **leftover items** and we can therefore use a *digit* as numerator in the number phrase that represents the collection of leftover items.

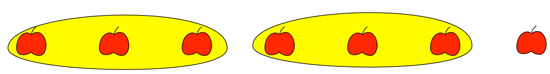
ii. We then deal with the *bundles of N items* exactly the same way: we bundle the *bundles of N items* and we then *represent* the *number* of *leftover bundles of N items* with a *digit* and the kind of bundle with a **postfix** which consists of the *system base* written as many times as we bundled. postfix

iii. And so on until there is nothing more to bundle.

EXAMPLE 2.12. To represent the collection



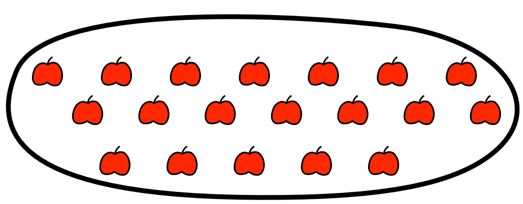
using only the THREE digits, 0, 1 and 2, we must bundle the APPLES at the rate of THREE APPLES for ONE BUNDLE OF THREE APPLES. We then have the bunch:



which we can represent by the *bunch list*:

[2 THREE-**Apples**, 1 **Apple**]

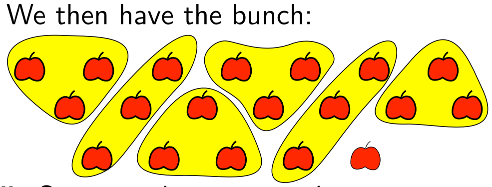
EXAMPLE 2.13. To represent the collection



using only the THREE digits, 0, 1, 2.

i. Since we have more than TWO APPLES, we must bundle the APPLES at the rate of THREE APPLES for ONE BUNDLE OF THREE APPLES.

We then have the bunch:



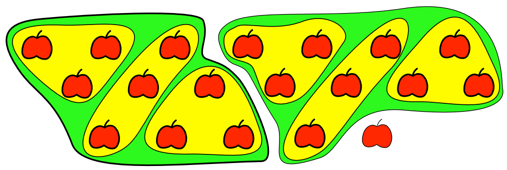
namely the bunch consisting of:

- ONE leftover APPLES
- SIX BUNDLES OF THREE APPLES

ii. Since we have more than TWO BUNDLES OF THREE APPLES, we must bundle the BUNDLES OF THREE APPLES at the rate of THREE BUNDLES OF

THREE APPLES for ONE BUNDLE OF THREE BUNDLES OF THREE APPLES.

We then have the bunch



namely the bunch consisting of:

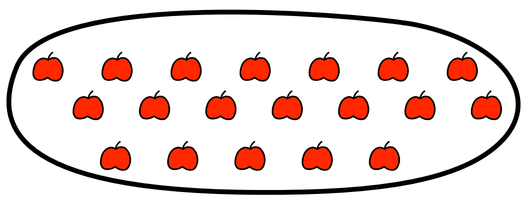
- ONE leftover APPLE
- NO leftover BUNDLE OF THREE APPLES
- TWO BUNDLES OF THREE BUNDLES OF THREE APPLES

➠ The number phrase that represents NO leftover BUNDLE OF THREE APPLES uses the postfix THREE- and the number phrase is 0 THREE-**Apples**

➠ To represent NO leftover BUNDLE OF THREE APPLES we use the postfix THREE- and the number phrase is 0 THREE-**Apples**

➠ The number phrase that represents the leftover TWO BUNDLES OF THREE BUNDLES OF THREE APPLES uses the postfix THREE-THREE- and is 2 THREE-THREE-**Apples**

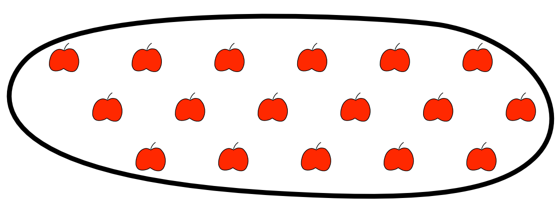
iii. Since there is nothing more to bundle, we can represent the original collection



by the *bunch list*

[2 THREE-THREE-**Apples**, 0 THREE-**Apples**, 1 SINGLE-**Apple**]

EXAMPLE 2.14. Suppose we are only using TWO digits, 1 and 2. Then, in order to rerpresent the collection

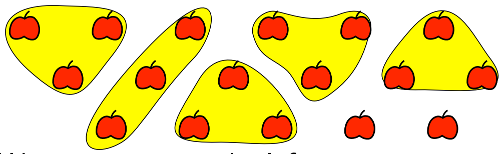


we proceed as follows:

i. Since we have more than TWO APPLES, we must bundle the APPLES at the rate of THREE APPLES for ONE COLLECTION OF THREE APPLES.

list-phrase

We then have the bunch:



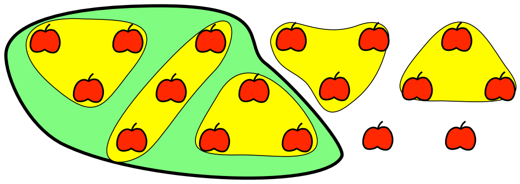
that is the bunch consisting of:

- TWO APPLES
- FIVE COLLECTIONS OF THREE APPLES

We can represent the leftover TWO APPLES by the number phrase 2 **Apples**

ii. Since we have more than TWO COLLECTIONS OF THREE APPLES, we must collect the COLLECTIONS OF THREE APPLES at the rate of THREE COLLECTIONS OF THREE APPLES for ONE COLLECTION OF THREE COLLECTIONS OF THREE APPLES.

We then have the bunch:

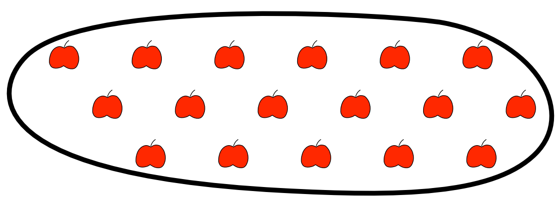


that is the bunch consisting of:

- TWO APPLES
- TWO COLLECTIONS OF THREE APPLES
- ONE COLLECTION OF THREE COLLECTIONS OF THREE APPLES

We can represent the leftover TWO COLLECTIONS OF THREE APPLES by the number phrase 2 THREE-**Apples**

iii. Since there is nothing more to bundle, we can represent the original collection



by the bunch list

[1 THREE-THREE-**Apples**, 2 THREE-**Apples**, 1 **Apple** **]**

2. Since the number-phrases in the *bunch-list* all involve the same *denominator* we can write the name of the items just once to the right of numerators and use what we will call a **list-phrase**.

- A *numerator* which is the list of the *bundle-numerators* in the bunch-list.
- A *denominator* which is the *denominator* common to all the number-phrases in the bunch-list, that is the denominator of the real world items in the original collection.

TEN

EXAMPLE 2.15. Instead of writing the *bunch-list*

[2 **THREE-THREE-Apple**, 0 **THREE-Apple**, 1 **SINGLE-Apple** **]**

we will write the *list-phrase*

$$\underbrace{\left[\ 2\ \textbf{THREE-THREE},\ 0\ \textbf{THREE},\ 1\ \textbf{SINGLE}\ \right]}_{\text{Numerator}}\ \underbrace{\textbf{Apples}}_{\text{Denominator}}$$

A *list-phrase* is a much better representation of a collection with a large number of items in that it separates *quantitative* information from *qualitative* information.

3 Base TEN

In the example of the previous section, we used THREE as system-base because the purpose was only to demonstrate how things worked[5].

1. Historically, a variety of system-bases were invented (See http://en.wikipedia.org/wiki/Gross_(unit)) but eventually the entire world ended up using the Hindu-Arabic system-base **TEN**.(See http://en.wikipedia.org/wiki/History_of_the_Hindu-Arabic_numeral_system.)

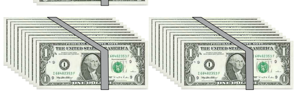
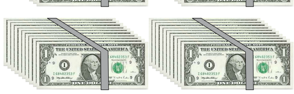
Tally marks	Digits
I	1
II	2
III	3
IIII	4
IIIII	5
IIIIII	6
IIIIIII	7
IIIIIIII	8
IIIIIIIII	9

So, with collections involving more than NINE items we have to bundle at the rate of TEN to ONE and thus the Hindu-Arabic system is a base TEN

[5]Educologists of course know that Z. P. Dienes always used to *start* his workshops with THIRD graders, base-THREE ARITHMETIC BLOCKS and the digits 0, 1, 2.

system also known as the **decimal system** (See http://en.wikipedia.org/wiki/Decimal). decimal system

EXAMPLE 2.16. Using **Washington** as a denominator for  and the decimal system to write numerators, we have:

Real-world DOLLAR BILLS	Paper-world List-phrases
	1 SINGLE **Washington**
	2 SINGLE **Washingtons**
... ...	...
	9 SINGLE **Washingtons**
	1 TEN **Washingtons**
 	[1 TEN, 1 Single] **Washingtons**
 	[1 TEN, 2 SINGLE] **Washingtons**
... ...	...
 	[1 TEN, 9 Single] **Washingtons**
	2 TEN **Washingtons**
 	[2 TEN, 1 SINGLE] **Washingtons**
... ...	...

2. The choice of TEN as base is nothing more than a historical accident as there is nothing logically necessary about the choice of TEN as base. It just happened and other bases could be—and actually *are*—used in exactly the same manner.

Example 2.17.

Example 2.18. Deep down, computers use a base TWO symbolic system, called **machine language** (See http://en.wikipedia.org/wiki/Machine_code), that involves only the digit 1 together with the symbol 0, because any electronic device is either *on* or *off*.
At intermediate levels, computers use

- either a base EIGHT symbolic system with the symbols
  0, 1, 2, 3, 4, 5, 6, 7
  (See http://en.wikipedia.org/wiki/Octal)
- or a base SIXTEEN symbolic system with the symbols
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f
  (See http://en.wikipedia.org/wiki/Hexadecimal)

The reason is that both bases make it easy to translate to and from the *machine language*.

The important point is that all that can be done with TEN digits can be done exactly the same way in any base, that is with *any* number of digits.

3. But while a base TEN system developed like the base THREE system we developed in the previous chapter would of course work perfectly well, it too would quickly become overly cumbersome with collections involving large numbers of items because it's not only the *digits*. So, we will now introduce the system of shorthands for large number that makes it the system we know and love. However, there are two other systems of shorthands which work equally well and which are in fact those used in science and technology and which we will introduce after we have introduced *array-phrases*.

4 Common System

This is the system of shorthands that we use in everyday life.

1. In real life, we use *shorthands* for strings of postfixes and **Single** often goes without saying.

- **Hundred** as shorthand for **Ten-Ten**,
- **Thousand** as shorthand for **Ten-Ten-Ten**,
- After that we use again **Single**, **Ten** and **Hundred** followed by **Thousand**

- After that we use **Single**, **Ten** and **Hundred** followed by one of the **scale-postfixes** of the short scale system of shorthands, that is **Million**, **Billion**, **Trillion**, etc, each of which shorthands three more copies of **Ten** than the previous one. (See http://en.wikipedia.org/wiki/Long_and_short_scales) Etc scale-postfix array-phrase

  In practice, though, **Single** often goes without saying. So, for instance,

instead of	we say
Ten-Ten	Hundred
Ten-Ten-Ten	Thousand
Ten-Ten-Ten-Ten	Ten-Thousand
Ten-Ten-Ten-Ten-Ten	Hundred-Thousand
Ten-Ten-Ten-Ten-Ten-Ten	Million
Ten-Ten-Ten-Ten-Ten-Ten-Ten	Ten-Million
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Hundred-Million
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Billion
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Ten-Billion
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Hundred-Billion
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Trillion
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Ten-Trillion
Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten-Ten	Hundred-Trillion

Example 2.19. When, in English, we speak of, say, a COLLECTION OF

SEVEN THOUSAND TWO HUNDRED AND FORTY SIX APPLES,

whether we realize it or not, we are using the *list-phrase*

[7 Ten-Ten-Ten, 2 Ten-Ten, 4 Ten, 6 Single,] Apples

Numerator Denominator

5 Array-Phrases

6 Decimal-phrases

1. More precisely, and especially for the purpose of *computations*, it will be much more convenient to use **array-phrases** in which the *postfixes* and the *digits* are separated:

header
decimal-phrase
selected postfix
decimal pointer

- The *denominator* remains the same,

but

- The *numerator* consists of:
  - A **header** which is the list of *all* the postfixes,
  - The *digits* in the list-phrase, each placed under the corresponding postfix.

EXAMPLE 2.20. Instead of writing the *list*-phrase

[7 THOUSAND, 2 HUNDRED, 0 TEN, 3 SINGLE] (Numerator) Apples (Denominator)

we can use the decimal header

THOUSAND	HUNDRED	TEN	SINGLE

and then write the *list-phrase* as the *array-phrase*

THOUSAND	HUNDRED	TEN	SINGLE
7	2	0	3

Numerator — Apples (Denominator)

2. But then the heading can go "without saying" and we can just write a **decimal-phrase** which consists of:

- A *denominator* which is still just the name of the *items*
- A *numerator* which consists of *all* the digits in the *array-phrase* (including the zeros if any) followed by a **selected postfix** with the digit corresponding to that selected postfix marked with a **decimal pointer**.

Decimal-phrases are extremely flexible in that we can select either

- The *postfix*,

or

- The *pointed digit*.

EXAMPLE 2.21. Instead of writing the *array-phrase*

THOUSAND	HUNDRED	TEN	SINGLE	TENTH	HUNDREDTH	THOUSANDTH
7	2	0	3			

Numerator — Apples (Denominator)

- We can choose a *postfix*, for instance HUNDRED, and then write the *decimal* *phrase*

$$\underbrace{72.03\ \textsc{Hundred}}_{\text{Numerator}}\quad \underbrace{\textbf{Apples}}_{\text{Denominator}}$$

or

- We can choose a *digit*, for instance 0, and then write the *decimal phrase*

$$\underbrace{720.03\ \textsc{Ten}}_{\text{Numerator}}\quad \underbrace{\textbf{Apples}}_{\text{Denominator}}$$

> **NOTE 2.2**
> The use of a *period* as decimal pointer is not universal and, for instance, many languages use a *comma* instead of a *point*. (See http://en.wikipedia.org/wiki/Decimal_mark.)

EXAMPLE 2.22. In order to convey the information contained in the list-phrase:

$$\underbrace{[\ 8\ \textsc{Hundred},\ 0\ \textsc{Ten},\ 3\ \textsc{Single}\ ]}_{\text{Numerator}}\quad \underbrace{\textbf{Apples}}_{\text{Denominator}}$$

i. We rewrite the list-phrase as an *array-phrase*:

THOUSAND	HUNDRED	TEN	SINGLE
0	8	0	3

Numerator — **Apples** (Denominator)

ii. We can then *select* a postfix, for instance HUNDRED so that the digit to be *pointed* is 8 and the *decimal-phrase* then is:

$$\underbrace{08.03\ \textsc{Hundred}}_{\text{Numerator}}\quad \underbrace{\textbf{Apples}}_{\text{Denominator}}$$

3. As with all phrases, there is no loss of information with *decimal-phrases*.

EXAMPLE 2.23. Receiving the decimal-phrase

08.03 HUNDRED Apples.
Numerator Denominator

the recipient will print the header and place the pointed digit 8 under the postfix **Hundred**:

THOUSAND	HUNDRED	TEN	SINGLE	
	8			**Apples**

and will then place the other digits, 0 8 03, under the heading according to their position relative to the pointed digit 8:

THOUSAND	HUNDRED	TEN	SINGLE	
0	8	0	3	**Apples**

From this array, the recipient can then reconstitute the list-phrase

[8 HUNDRED , 3 SINGLE] **Apples**

4. A given list-phrase can be replaced by several different decimal-phrases depending on which postfix we decide to *select*:

EXAMPLE 2.24. In order to convey the information contained in the *list-phrase*:

[4 THOUSAND , 6 TEN , 1 SINGLE] **Apples**

we could use any one of the following *decimal-phrases*

4061. SINGLE **Apples**

406.1 TEN **Apples**

40.61 HUNDRED **Apples**

4.061 THOUSAND **Apples**

Indeed, any of these decimal-phrases corresponds to the following array-phrase:

THOUSAND	HUNDRED	TEN	SINGLE	
4	0	6	1	**Apples**
Numerator				Denominator

from which the recipient would then be able to recover

[4 THOUSAND, 0 HUNDRED, 6 TEN, 1 SINGLE] (Numerator) **Apples** (Denominator)

7 Flexibility

select denominator

In addition to their efficiency, decimal-phrases have the advantage of giving us a lot of flexibility.

1. At any moment, we can change either the *postfix* or *point* at another digit. In both cases, all we need to adjust matters is to replace the decimal-phrase under the heading and then:

i. If we changed the *postfix*, we adjust the *numerator* by pointing now at the digit that is below the new postfix in the header,

========================== Begin EXAMPLE

EXAMPLE 2.25. If we want to change the *postfix* of the decimal-phrase

4.061 **Thousand Apples**

by using, for instance, the postfix **Ten** instead of the postfix **Thousand** then, in order to find which digit we must point in the new numerator, we place the original decimal-phrase under a header:

Thousand	Hundred	Ten	Single	
4	0	6	1	**Apples**

which shows that we need to point at 6. We thus get the decimal-phrase

406.1 **Ten Apples**

==================================== End EXAMPLE

ii. If we changed the digit being *pointed*, we adjust the numerator by using the postfix in the header that is above the pointed digit.

========================== Begin EXAMPLE

EXAMPLE 2.26. If we want to change the *numerator* of the decimal-phrase

4.061 **Thousand Apples**

by pointing, for instance, at

0.

then, in order to find the new postfix, we place the original decimal-phrase under a header:

Thousand	Hundred	Ten	Single	
4	0	6	1	**Apples**

which shows that we need to use the postfix **Hundred** We thus get the decimal-phrase

$$40.61$$ **Hundred Apples**

======================================= End EXAMPLE

2. To summarize, we have several ways to represent on paper a real-world collection:

- A list
- A list-phrase
- An array
- Any number of decimal-phrases

EXAMPLE 2.27. The collection described in English as

{SEVEN THOUSANDS TWO HUNDRED THREE APPLES}

can be represented on paper by any of the following:

- The list
  [7 **Thousand Apples**, 2 **Hundred Apples**, 3 **Single Apples]**
  which corresponds closely to the bundling that was done
- The list-phrase
  [7 **Thousand** , 2 **Hundred** , 3 **Single] Apples**
  which corresponds closely to the way we describe it in English
- The array

Thousand	Hundred	Ten	Single	
7	2	0	3	**Apples**

  which, as we will see, is very convenient to carry out certain procedures.
- Any one of the decimal-phrases:

precise decimal phrase
approximation
too small to matter

7203. **Single** **Apples**
720.3 **Ten** **Apples**
72.03 **Hundred** **Apples**
7.203 **Thousand** **Apples**

which, as we will see, is also a very convenient to deal with amounts of stuff.

Each representation has its advantages and disadvantages. Which we will use will depend on the circumstances but it will pay to be able to switch easily from one to any of the others.

8 Procedurability

As we will see, it turns out that arrays will be extremely efficient for *computational procedures* because we can enter the numerators of several list-phrases under the same header rather than using several arrays.

EXAMPLE 2.28. Using the following header

Thousand	Hundred	Ten	Single

we can rewrite, for example, the following factored list-phrases

[3 **Thousand** , 1 **Ten** , 7 **Single**] **Apples**

[7 **Thousand** , 9 **Hundred** , 3 **Ten**] **Apples**

[4 **Hundred** , 2 **Single**] **Apples**

under a single header:

Thousand	Hundred	Ten	Single	
3	0	1	7	**Apples**
7	9	3	0	**Apples**
0	4	0	2	**Apples**

9 Approximations

In many situations, using an **precise decimal phrase** is not realistic and can even be misleading and it is often better to give an **approximation** +[...] with [...] standing for some number **too small to matter** metric system prefixes

EXAMPLE 2.29. The Census Bureau says that the population of Philadelphia in 2010 was 1,526,006. But, knowing that the population changes, the Census Bureau did give a date, April 1, but, even so, the precision of the figure is misleading in that there must have been deaths and births on April 1, 2010. So, it would have been better to say that the population of Philadelphia on April 1, 2010 was 1.526,000 +[...] with [...] standing for some number *too small to matter*.

The flexibility of the decimal system makes it completely trivial to approximate a decimal phrase: just replace the rightmost digit(s) by zeros and add [...] to indicate that the figure with the zeros is only an approximation.

EXAMPLE 2.30. Depending on the circumstances, we could approximate 7361. SINGLE Apples by:

7360. +[...] SINGLE Apples

7300. +[...] SINGLE Apples

7000. +[...] SINGLE Apples

==================================== End EXAMPLE

10 Metric Prefixes

Often, using for the *denominator* the name of the items is neither convenient nor realistic because this can force the number of items then to be either too large or too small for comfort. So, the **metric system** uses the **prefixes**

KILO	HECTO	DEKA	—	DECI	CENTI	MILLI

to create denominator that are larger or smaller than the name of the items and thus more appropriate to what is being discussed.

EXAMPLE 2.31. When discussing the price of cars, people often say that a car sold for 13 **K**—by which they mean 13 **KILO Dollars**—rather than say that the car sold for 13 000 **Dollars**

EXAMPLE 2.32. When speaking about computer files, people often say that a file is 13 **MEG**—by which they mean 13 **MEGA Bytes**—rather than to say that the file is 13 000 000 **Bytes**

EXAMPLE 2.33.

Thousand	Hundred	Ten	Single

The metric system is used for denominators in every country in the world except the U.S., Liberia and Myanmar and to some extent Great Britain where it is nevertheless used for *money* as well as in the *sciences* and *technologies*. (See https://en.wikipedia.org/wiki/Metric_system.)

EXAMPLE 2.34. If we choose **Franklins** as our primary denominator, then, instead of using the header

Cleveland	Franklin	Hamilton	Washington	Roosevelt

we can use the header

DEKA Franklins	Franklins	DECI Franklins	CENTI Franklins	MILLI Franklins

so that, for instance,

- instead of writing 3.2 **Clevelands** we can write 3.2 **DEKAFranklins**,
- instead of writing 7.45 **Hamiltons** we can write 7.45 **DECIFranklins**,
- instead of writing 0.83 **Washingtons** we can write 0.83 **CENTIFranklins**,
- instead of writing 53.46 **Roosevelts** we can write 7.45 **MILLIFranklins**.

But if we choose **Washingtons** as our primary denominator, then instead of using the header

Clevelands	Franklins	Hamiltons	Washingtons	Roosevelts

we can use the header

KILO Washingtons	HECTO Washingtons	DEKA Washingtons	Washingtons	DECI Washingtons

so that, for instance,

– instead of writing 3.2 **Clevelands** we can write 3.2 **Kilo****Washingtons**,
– instead of writing 23.77 **Franklins** we can write 23.77 **Hecto****Washingtons**,
– instead of writing 7.45 **Hamiltons** we can write 7.45 **Deka****Washingtons**.
– instead of writing 53.46 **Roosevelts** we can write 7.45 **Deci****Washingtons**.

11 Default Rules

There are two default rules.

1. When the decimal point is to the right of the rightmost digit, it is customary not to write the decimal point at all. The corresponding default rule is:

When there is no decimal point, it goes without saying that the decimal point is to the *right* of the rightmost digit.

Example 2.35.

204 **Ten Apples**

stands for

204. **Ten Apples**

2. When the postfix is **Single**, it is customary *not* to write it. The corresponding default rule is:

When there is *no* postfix, it goes without saying that the postfix is **Single**. This, though, is *extremely dangerous* because it depends on us *remembering* what the array denominator is that was picked earlier on[6].

Example 2.36.

72. **Apples**

[6]This is perhaps one more occasion to remind Educologists that memory is the weakest part of the mind, that it is the first to go and that, so far, humans can be defined as thinking entities, that is entities amenable to logic, rather than mere memorizing ones.

stands for

72.. **Single Apples**

3. In the U.S., instead of writing, for instance,

0.56 **Thousand Apples**.

it is usual to write

.56 **Thousand Apples**

We will absolutely *not* do so in this book, if only because we don't like the idea of a pointer pointing, at least apparently, at nothing.

4. By placing 85.7 **Ten Apples** under a header,

Thousand	Hundred	Ten	Single	
0	8	5	7	Apples

we see that the decimal-phrase 85.7 **Ten Apples** could just as well be replaced by any of the following decimal-phrases:

857.	**Single Apples**
	Ten Apples
8.57	**Hundred Apples**
0.857	**Thousand Apples**

Any of the above represents the same combination as the list-phrase

[8 **Hundred**, 5 **Ten**, 7 **Single**] **Apples**

5. As we already saw, we can select as array denominator *any* denominator *we* want.

Example 2.37. A rich person might choose **Thousand Apples** as her/his array denominator while the rest of us would probably choose **Single Apples** as our array denominator.

6. Not only does the use of decimal-phrases allow us to choose up-front any *array denominator* we want but it also allows us to change our mind later and, at any time, to choose a different denominator. All we need to do is to place the array numerator back under the header, choose the denominator we want to use now as array denominator and to “move” the array point accordingly.

7. Similarly, if, for whatever reason, we want to have a smaller or a larger *array numerator*, all we need to do is to place the array numerator back under the header, choose the digit we want to be pointed and use as array denominator the denominator that corresponds to the newly pointed digit.

There are only four systems that bridge all linguistic barriers:
–The complete set of mathematical symbols,
–The International System of Units,
–The symbols for chemical elements,
–The way of writing notes for music.

Anders J. Thor[1]

lump
length
area
volume

Chapter 3

Decimals for Amounts of Substance

While *items* come in as *collections*, *substances* come in as **lumps** but natural languages use specialized words instead of the generic word "lump". For instance, in English:

- A 1-dimensional *lump* is usually called a **length**. (See http://en.wikipedia.org/wiki/Length.)
- A 2-dimensional *lump* is usually called an **area**. (See http://en.wikipedia.org/wiki/Area.)
- A 3-dimensional *lump* is usually called a **volume**. (See http://en.wikipedia.org/wiki/Volume.)

EXAMPLE 3.1.
- A *length* of rope.
- An *area* of grass.
- A *volume* of water.

In order to *represent* a *lump* on paper, we will again write a *number phrase* that gives all the *information* necessary to specify that *lump*:

- The *kind* of substance in the lump (*qualitative* information) which we represent on paper by a *denominator* that is a name usually associated

[0]Chairman of IEC TC 25

reduce
measure
discretize
unit lump
measured part
measure

with the substance,

- The *amount* of substance in the lump (*quantitative* information) which we will represent by a *numerator*. In the case of *lumps*, though, getting the numerator will be a bit more complicated than in the case of *collections* where we just had to *count* the items.

EXAMPLE 3.2.

1 Measuring Amounts

To deal with the problem of how to represent *amounts of substances* we will use a standard approach in mathematics which is to **reduce** a new problem to a problem we already solved.

1. In the present case, we will reduce the problem of representing an *amount of substance* to the problem of representing a *number of items*, which we already solved in chapter 2. We will do that by first **measuring** the *lump of substance* which we do as follows:

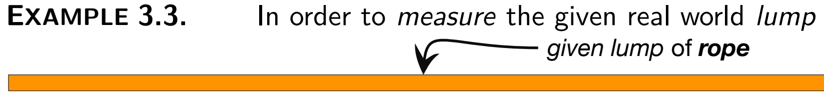
i. We **discretize** the lump of substance, that is:

– We choose a **unit lump** of substance.

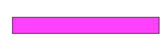
– We determine the **measured part** of the given lump, that is the largest collection of *unit lumps* contained in the given *lump*, where by largest we mean that, with one more *unit lump*, we would have more substance than in the given *lump*.

ii. The **measure** of the given lump is then the number of *unit lumps* in the *measured part*.

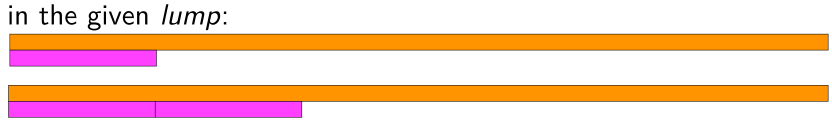
EXAMPLE 3.3. In order to *measure* the given real world *lump* of ROPE,



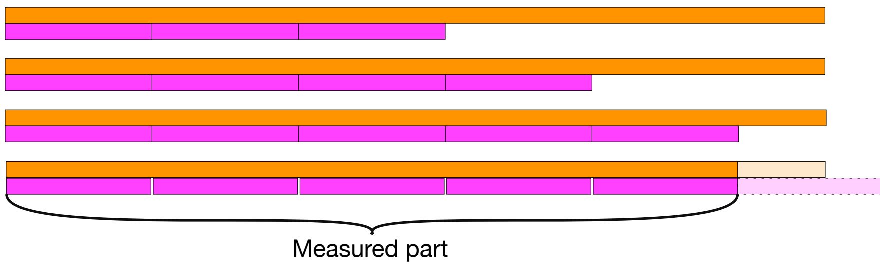
i. We discretize the given real world *lump* of ROPE, that is:

– We choose a *unit lump* of rope, for instance 

– We determine the *measured part* of the given lump which is the largest collection of *unit lumps* contained in the given *lump*, where by largest we mean that, with one more *unit lump*, we would have more substance than in the given *lump*:



remainder



ii. The *measure* of the given lump of ROPE is therefore FIVE *unit lumps* of ROPE.

2. Representing lumps involving a *large* amount of real world substance is not a problem: since the *measured part* of a lump is a *collection* of *unit lumps*, the procedure for denoting a *measured part* with a *large number of unit lumps* will be exactly the same as the procedure we used in section 2 to represent a collection of a *large number of items*.

EXAMPLE 3.4. The *measured part* of a large length of ROPE might be represented on paper by a *number phrase* like:

[7 THOUSAND, 5 HUNDRED, 8 TEN, 3 SINGLE] Sticks of Rope

Numerator *Denominator*

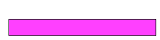
and therefore, as we saw in section 6, by the decimal phrase

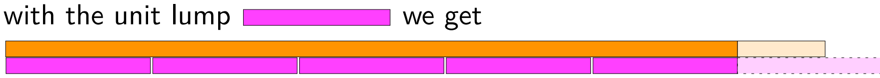
7583. SINGLE Sticks of Rope

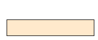
Numerator Denominator

3. However, it is most important to realize that, no matter what *unit lump* we choose, the chances that a *given lump* will contain an *exact* number of *unit lumps* are strictly zero and that there will *always* be some **remainder**.

EXAMPLE 3.5. When we measure the given real world *lump* of ROPE,

with the unit lump  we get



where  is the *remainder*.

EXAMPLE 3.6.

[...]
sub-unit lump

EXAMPLE 3.7.

2 Real Number Phrases

What makes *lumps of substance* more difficult to represent than *collections of items* is that we have to deal with the *remainder*

1. There is no point representing a lump of substance by the number phrase that represents its *measured part* if the remainder is not small enough that we can ignore it. But when the remainder *is* small enough that we can ignore it, then we will represent the remainder with the symbol **[...]**.

EXAMPLE 3.8. Suppose we have a lump whose *measured part* is represented by

83 (Numerator) Sticks of Rope (Denominator)

and suppose that the *remainder* is small enough for our purpose. Then we will represent the remainder by

[...] (Numerator) Sticks of Rope (Denominator)

so that we can represent the lump by

83 (Numerator) Sticks of Rope (Denominator) + [...] (Numerator) Sticks of Rope (Denominator)

2. When the remainder is *not* small enough that we can ignore it, then what we do is to measure just the remainder with a **sub-unit lump**, that is a smaller lump such that **TEN** of these smaller lumps amount to the same as the *unit-lump*. And in fact the metric prefix **DECI** which we saw in section 10 provides us automatically with a name for this sub-unit lump.

EXAMPLE 3.9. Suppose we have a lump whose *measured part* is represented by

83 (Numerator) Sticks of Rope (Denominator)

and suppose that the remainder is in fact too large to be represented by [...]. In that case, we measure the *remainder* with the *sub-unit length* TEN of which amount to the same as the original unit-length and whose name will therefore real numerator

be DeciSticks of Rope and we might then be able to represent the given lump by, for instance,

83 Sticks of Rope + 7 DeciSticks of Rope + [...] DeciSticks of Rope

where [...] *might* now represent a remainder "too small to matter".

If the remainder is still *not* small enough not to matter, then we just repeat the above maneuver with a *sub-sub-unit lump* and we might get something like

83 Sticks of Rope + 7 DeciSticks of Rope + 4 CentiSticks of Rope + [...] CentiSticks of Rope

where [...] *might* now represent a remainder "too small to matter".

If not, etc.

3. What we will usually do is to use *decimal phrases* to represent the *measured part*.

Example 3.10. In the previous example, using *decimal phrases* for the measured parts, we would write:

$\underbrace{83.}_{\text{Numerator}}\ \underbrace{\text{Sticks of Rope}}_{\text{Denominator}}$

or

$\underbrace{83.7}_{\text{Numerator}}\ \underbrace{\text{Sticks of Rope}}_{\text{Denominator}}$

or

$\underbrace{83.74}_{\text{Numerator}}\ \underbrace{\text{Sticks of Rope}}_{\text{Denominator}}$

4. To represent the given lump itself , we will use **real numerators**, that is decimal phrase followed by [...].

Example 3.11. In the previous example, using *real numerators*, we would write for the given length of rope:

$\underbrace{83. + [...]}_{\text{Numerator}}\ \underbrace{\text{Sticks of Rope}}_{\text{Denominator}}$

or

$\underbrace{83.7 + [...]}_{\text{Numerator}}\ \underbrace{\text{Sticks of Rope}}_{\text{Denominator}}$

or

$$\underbrace{83.74 + [...]}_{\text{Numerator}}\ \underbrace{\textbf{Sticks}\text{ of }\textbf{Rope}}_{\text{Denominator}}$$

EXAMPLE 3.12.

EXAMPLE 3.13.

EXAMPLE 3.14.

EXAMPLE 3.15.

EXAMPLE 3.16.

EXAMPLE 3.17.

EXAMPLE 3.18.

3 The U.S. Customary System of Units

In the United States, though, as we will now see, the *units of stuff* are set by the U.S. Customary System of Units (See http://www.ndt-ed.org/GeneralResources/Units/USCustomarySystem.htm) and have none of the features we used above.

1. The bundling rates do not stay the same as we keep on bundling at higher levels.

EXAMPLE 3.19. The U.S. Customary System of Units sets the bundling rates for *lengths* as:

Bundle	Single item
Bundle of twelve inches	one foot
Bundle of three feet	one yard
Bundle of six hundred sixty yards	one mile

2. The bundling rates are different depending on the *kind* of stuff being dealt with.

EXAMPLE 3.20. The U.S. Customary System of Units sets the bundling rates for *liquids* as:

Bundle	Single item
BUNDLE OF EIGHT OUNCES	ONE CUP
BUNDLE OF TWO CUPS	ONE PINT
BUNDLE OF TWO PINTS	ONE QUART
BUNDLE OF FOUR QUARTS	ONE GALLON

but sets the bundling rates for *weights* as

Bundle	Single item
BUNDLE OF SIXTEEN OUNCES	ONE POUND
BUNDLE OF HUNDRED POUNDS	ONE HUNDREDWEIGHT
BUNDLE OF TWENTY HUNDREDWEIGHT	ONE TON

3. It is often impossible to find a unit of stuff that is *commensurate* with the amount of stuff to be measured.

EXAMPLE 3.21.
- A MILE, the largest U.S. Customary Unit of length, is still not large enough to be commensurate with, say, the distance between the sun and the star Alpha Centauri because that would give 25689592880812.363 Miles which is not really informative.
- An INCH, the smallest U.S. Customary Unit of length, is still not small enough to be *commensurate* with the distance between two atoms because we would get 0 **inches** $+$ [...] which is not really informative either.

4. The units can be too far apart which then makes the approximation difficult to control.

EXAMPLE 3.22. The jump from YARD to MILE is very large which can make dealing with remainders difficult. When measuring the length of, say, a small road, we may run into the following catch-22 situation:
- If we use MILES as units of length, the remainder can be up to ONE MILE which for a small road may be too much because a small road may not even

metric system
base unit

be ONE MILE long in which case the remainder would be all there is and therefore certainly not too-small-to-matter-here.

- If we use YARDS as units of length, this allows the remainder to be up to ONE YARD which, even for a small road, may be too small because the number of YARDS is likely to depend on how the road was surveyed and thus the remainder does not really correspond to the approximation.

The U.S. Customary System of Units does provide units of length between YARDS and MILES such as FURLONG (See http://en.wikipedia.org/wiki/Furlong) but nobody seems to use FURLONGS anymore. For instance, interstate roads are marked in TENTHS OF A MILE rather than in FURLONGS.

5. In fact, the U.S.A., Liberia and Myanmar (AKA Burma) are the only countries to use the U.S. System of Customary Units.

6. However, in the U.S., the money world, as well as the scientific and military worlds, that is finances, sciences and technologies—use the International System of Units (AKA). (See http://en.wikipedia.org/wiki/Metrication_in_the_United_States.)

4 The International System of Units

Contrary to the U.S. Customary System of Units, the International System of Units does *not* provide a list of units because there would then be in that list both a largest unit with no guarantee that this largest unit would always be large enough and a smallest unit with no guarantee that this smallest unit would always be small enough.

What the International System of Units does is to provide a mechanism for creating and naming *systematically* any number of units, no matter what the kind of substance, including, in any particular situation, units as large as we need and units as small as we need. (See http://en.wikipedia.org/wiki/International_System_of_Units.)

1. More precisely, the International System of Units specifies a single **base unit** of substance for each kind of substance but the way it does it depends on the kind of substance:

i. For each one of the following seven kinds of substance, the International System of Units specifies the base unit by way of a *real-world process* (See for instance http://en.wikipedia.org/wiki/Metre):

Kind of stuff	Base Unit
Length	meter
Mass	gram
Time	second
Electric current	ampere
Temperature	kelvin
Luminous Intensity	candela
Amount Of Substance	mole

base␣unit,␣fundamental
base␣unit,␣derived
secondary␣units

We will refer to these base units as **fundamental base units**.[2]

ii. For all the other substances, the International System of Units specifies a base unit, referred to here as **derived base unit**, by way of a *paper-world procedure* involving only multiplication and division of any of the seven fundamental base units and/or any of the base units that have already been derived.

Example 3.23. Since the area of a rectangle is its length times its width, the (derived) base unit for areas is derived from the (fundamental) base unit for length as meter×meter AKA square meter.

Example 3.24. Since we get the average speed of a vehicle traveling a distance by dividing that *distance* by the *time* it took the vehicle to travel that distance, the (derived) base unit for speed is *meter/second*.

2. In order to create **secondary units** from the base unit, the International System of Units uses the *metric prefixes* which we already saw in section 10:

Kilo	Hecto	Deka	—	Deci	Centi	Milli

which work exactly the same as

Thousand	Hundred	Ten	Single	Tenth	Hundredth	Thousandth

[2]Educologists are of course well aware that the law adopted by the National Constituting Assembly of the French Revolution on March 30, 1791 mandated a "natural" definition of the primary units, that is a definition that would be neither anthropocentric nor nationalistic.

EXAMPLE 3.25. When the substance is *length*, then the base unit is **Meter** and secondary units are

KILO	HECTO	DEKA		DECI	CENTI	MILLI
Meter	Meter	Meter	Meter	Meter	Meter	Meter

so that, for instance, instead of writing 32.6 **Meters** we can write:

0.0326 **KILOMeters**,
0.326 **HECTOMeters**,
3.26 **DEKAMeters**,

$326.$ **DECIMeters**,
$3260.$ **CENTIMeters**,
$32600.$ **MILLIMeters**,

compare
one-to-one matching
leftover item
is larger than

Chapter 4

Comparisons

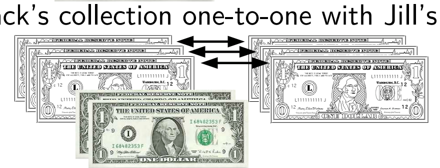
1 Comparing Collections

Two real world collections can **compare** in three different ways depending on whether, after we have done a **one-to-one matching**, there are **leftover items** in the *first* collection or in the *second* collections or *no* leftover item whatsoever:

- When the leftover items are in the *first* collection, we will say that the *first* collection **is larger than** the second collection.

EXAMPLE 4.1.

To compare Jack's  with Jill's  in the real world, we match Jack's collection one-to-one with Jill's collection:



and we have that

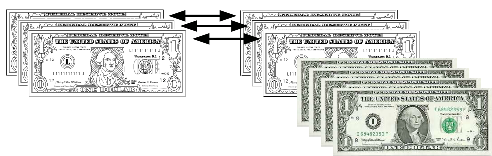
is smaller than
is the same as

Jack's collection *is larger than* Jill's collection

- When the leftover items are in the *second* collection, we will say that the *first* collection **is smaller than** the second collection.

EXAMPLE 4.2.

To compare Jack's  with Jill's  in the real world, we match Jack's collection one-to-one with Jill's collection:



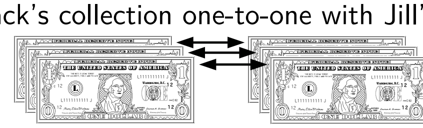
and we have that

Jack's collection *is smaller than* Jill's collection

- When there is *no* leftover item, we will say that the first collection **is the same in size as** the second collection.

EXAMPLE 4.3.

To compare Jack's  with Jill's  in the real world, we match Jack's collection one-to-one with Jill's collection:



and we have that

Jack's collection *is the same as* Jill's collection

EXAMPLE 4.4.

EXAMPLE 4.5.

EXAMPLE 4.6.

comparison sentences
$<$
is smaller than
is less than
$>$
is larger than
is more than

EXAMPLE 4.7.

EXAMPLE 4.8.

EXAMPLE 4.9.

EXAMPLE 4.10.

NOTE 4.1

2 Comparison Sentences: Equalities and Strict Inequalities

1. Given two real world *collections* and the *number phrases* that represent then, we now want to write on paper **comparison sentences**, that is *sentences* (TRUE or FALSE) that state whether the first collection is *larger than*, *smaller than* or *the same as* the second collection. For that we introduce the following *comparison verbs*:

- We will use the *comparison verb* $<$ to represent the comparison *is smaller than* and we will read $<$ either as **is smaller than** or as **is less than**.

EXAMPLE 4.11.

Given Jack's  and Jill's , the *comparison sentence* which states that

Jack's collection *is smaller than* Jill's collection

is

9 **Dollars** < 5 **Dollars**

- We will use the *comparison verb* $>$ to represent the relationship *is larger than* and we will read $>$ either as **is larger than** or as **is more than**.

EXAMPLE 4.12.

=
is equal to
strict␣inequality
equality

Given Jack's  and Jill's , the *comparison sentence* which states that

Jack's collection *is larger than* Jill's collection

is

$$5 \text{ Dollars} > 3 \text{ Dollars}$$

- We will use the *comparison verb* $=$ to represent the comparison *is the same as* and we will read $=$ as **is equal to**.

EXAMPLE 4.13.

Given Jack's  and Jill's , the *comparison sentence* which states that

Jack's collection *is the same as* Jill's collection

is

$$3 \text{ Dollars} = 9 \text{ Dollars}$$

> **NOTE 4.2**
> We use the same words for the real-world *comparisons* and for the *comparison verbs* we write on paper because, even though we need to distinguish the real-world from the paper-world, that is the way it goes and, anyway, nobody could remember which is to be used when talking about the real world and which is to be used when talking about the paper world. So, we will have to use other ways to keep in mind whether we are working in the *real world* or in the *paper world*.

2. *Sentences* involving the *comparison verbs* $>$ or $<$ are called **strict inequalities** while sentences involving the *verb* $=$ are called **equalities**.

EXAMPLE 4.14.

$13 \text{ Dollars} < 7 \text{ Dollars}$	and	$8 \text{ Dollars} > 2 \text{ Dollars}$	are *strict inequalities*
		$3 \text{ Dollars} = 3 \text{ Dollars}$	is an *equality*

3. In order to be able to decide whether a comparison sentence is TRUE or FALSE, we first need to extend the concept of *counting* to counting from any **start-number** to any **end-number** so that we have to be able to count in either one of two **directions**, namely **count up** or **count down**. Either way, though

start-number
end-number
direction
count up
count down

i. We start counting *after* the start-number. (Just as we started counting *after* 0.)
ii. We stop counting *after* the end-number.

EXAMPLE 4.15.
To count from the start-number 3 to the end-number 7, we start counting *up* after 3, that is 4, and stop after 7:

$$\xrightarrow{4,\ 5,\ 6,\ 7}$$

EXAMPLE 4.16.
To count from the start-number 37 to the end-number 12, we start counting *down* after 37, that is at 36, and stop after 12:

$$\xrightarrow{36,\ 35,\ 34,\ \dots\ ,\ 14,\ 13,\ 12}$$

4. While the real-world *real-world process* of comparing two real-world collections by *matching one-to-one* the two collections can be very painful with large collections, once the collections are represented by *number-phrases*, getting the TRUE comparison sentence is easy:

- If we need to count *up*, then the TRUE sentence must use the comparison verb $<$
- If we need to count *down*, then the TRUE sentence must use the comparison verb $>$
- If we don't need to count at all, then the TRUE sentence must use the comparison verb $=$

EXAMPLE 4.17.
The sentence

$$5 \textbf{ Dollars} < 3 \textbf{ Dollars}$$

is FALSE because to count from 5 to 3 we must count *down*. (And, indeed, in the real world, after a *one-to-one matching* of  and

, the leftover items would be in the *first* collection.)

EXAMPLE 4.18.
The sentence

$$5 \text{ Dollars} > 3 \text{ Dollars}$$

is TRUE because to count from 5 to 3 we must count *down*. (And, indeed, in the real world, after a *one-to-one matching* of  and , the leftover items would be in the *first* collection.)

EXAMPLE 4.19.
The sentence

$$5 \text{ Dollars} = 3 \text{ Dollars}$$

is FALSE because to count from 5 to 3 we *must* count. (And, indeed, in the real world, after a *one-to-one matching* of  and , there *would be* leftover items.)

EXAMPLE 4.20.

EXAMPLE 4.21.

3 Comparison Sentences: Weak Inequalities

1. Most of the time, though, we do not want to speak FALSE and we use:

AGREEMENT 4.1
When no indication of TRUE or FALSE is given, comparison sentences will always be understood to be TRUE and this will go *without saying*

So, instead of writing a comparison sentence that is FALSE and having to say that it is FALSE, we will write the **negation** of the FALSE comparison sentence which will be a TRUE comparison sentence whose truth will therefore "go without saying".

negation
slash
mutually exclusive

To write the *negation* of a comparison sentence, we just **slash** the *comparison verb* in the comparison sentence.

EXAMPLE 4.22.
Instead of writing

$$5 \textbf{ Washingtons} < 3 \textbf{ Washingtons} \text{ is FALSE}$$

we write

$$5 \textbf{ Washingtons} \not< 3 \textbf{ Washingtons}$$

which we read as

FIVE Washingtons is *no less than* THREE Washingtons.

and where the fact that it is TRUE goes without saying.

EXAMPLE 4.23.
Instead of writing

$$5 \textbf{ Washingtons} > 7 \textbf{ Washingtons} \text{ is FALSE}$$

we write

$$5 \textbf{ Washingtons} \not> 7 \textbf{ Washingtons}$$

which we read as

FIVE Washingtons is *no more than* SEVEN Washingtons.

and where the fact that it is TRUE goes without saying.

EXAMPLE 4.24.
Instead of writing

$$3 \textbf{ Washingtons} = 7 \textbf{ Washingtons} \text{ is FALSE}$$

we write

$$3 \textbf{ Washingtons} \neq 7 \textbf{ Washingtons}$$

which, naturally, we read as

FIVE Washingtons is *not equal to* SEVEN Washingtons.

and where the fact that it is TRUE goes without saying.

2. The comparison verbs $<$, $>$, and $=$ are **mutually exclusive**: given three comparison sentences involving the same number-phrases and the comparison verbs *is larger than*, *is smaller than*, and *is the same as*, as soon as

weak inequality
$\leqq$

we know that one of the three comparison sentences is TRUE, we automatically know that the other two comparison sentences must be FALSE.

EXAMPLE 4.25.
Once we know that 5 **Dollars** $>$ 3 **Dollars** is TRUE, we automatically know that both 5 **Dollars** $<$ 3 **Dollars** and 5 **Dollars** $=$ 3 **Dollars** must be FALSE.

Another way to put it is that as soon as we know that one of the three comparison sentences is FALSE, then one of the other two comparison sentences *must* be TRUE.

EXAMPLE 4.26.
Once we know that 4 **Dollars** $>$ 7 **Dollars** is FALSE, we automatically know that one of 4 **Dollars** $<$ 7 **Dollars** and 4 **Dollars** $=$ 4 **Dollars** must be TRUE.

3. In fact, while mathematicians do not mind writing negation of *equalities*, they don't like writing negations of *strong inequalities* and prefer to make use of the observation we just made that the three comparison verbs, $<$, $>$, and $=$, are mutually exclusive: if a strong inequality is FALSE, then either the *other strong inequality* or the *equality* must be TRUE. So, they prefer to write **weak inequalities** that say just that. More precisely,

- Instead of writing a comparison sentence with the verb $\ngtr$, read as "no more than", we actually write a *weak inequality* with the comparison verb $\leqq$, read as "less than or equal to".

  EXAMPLE 4.27.
  Instead of writing
  $$3 \text{ **Washingtons**} \ngtr 7 \text{ **Washingtons**}$$
  read as
  THREE Washingtons is no more than SEVEN Washingtons.
  mathematicians prefer to write "the other two" comparison sentences
  3 **Washingtons** $<$ 7 **Washingtons** or 3 **Washingtons** $=$ 7 **Washingtons**
  in the shape of the weak inequality
  $$3 \text{ **Washingtons**} \leqq 7 \text{ **Washingtons**}$$
  which, naturally, we read as
  THREE Washingtons is less than or equal to SEVEN Washingtons.
  and where the fact that it is TRUE goes without saying.

- Instead of writing a comparison sentence with the verb $\nless$, read as "no less than", we actually write a *weak inequality* with the comparison verb

$\geqq$, read as "more than or equal to". $\geqq$ inequality

Example 4.28.
Instead of writing

$$5 \text{ Washingtons} \nless 3 \text{ Washingtons}$$

read as

FIVE Washingtons is no less than THREE Washingtons.

mathematicians prefer to write "the other two" comparison sentences

$$5 \text{ Washingtons} \geqq 3 \text{ Washingtons}$$

which, naturally, we read as

FIVE Washingtons is more than or equal to THREE Washingtons.

and where the fact that it is TRUE goes without saying.

Finally, the word **inequality** will cover both *strong inequalities* and *weak inequalities*.

4 Comparison Sentences For Large Collections

In order to compare *large* collections, we represent them by *decimal* number-phrases but, both on paper and in our mind, we use in fact *tabular* number-phrases under a single heading as this by-passes the issue of whether or not the two decimal number-phrases have the same select-denominator.

There are two cases:

1. The *leftmost* digit of the *tabular* number-phrases are under *different* denominators. Then, *regardless of the other digits*, we can write either that:

- the number-phrase with the leftmost digit *left* of the other leftmost digit is *larger than* the other number-phrase.

or that

- the number-phrase with the leftmost digit *right* of the other leftmost digit is *smaller than* the other number-phrase.

Example 4.29.
Jack's collection is represented by 5.87 **DekaWashingtons** and Jill's collection is represented by 2341.6 **DeciWashingtons**.

i. The tabular number-phrases are:

	Kilo Washingtons	Hecto Washingtons	Deka Washingtons	Washingtons	Deci Washingtons	Centi Washingtons
Jack			5	8	7	
Jill		2	3	4	1	6

ii. Since Jill has **HectoWashingtons** while Jack doesn't have any, and even though she has fewer **DekaWashingtons**, fewer **Washingtons** and fewer **DeciWashingtons** than Jack, we can write that

5.87 **DekaWashingtons** < 2341.6 **DeciWashingtons**

or that

2341.6 **DeciWashingtons** > 5.87 **DekaWashingtons**

2. The *leftmost* digit of the tabular number-phrases are under the *same* denominator. Then, *regardless of the other digits*, we can write either that:

- the number-phrase with the *larger* leftmost digit is *larger* than the other number-phrase

or that

- the number-phrase with the *smaller* leftmost digit is *smaller* than the other number-phrase

Example 4.30.
Jack's collection is represented by 54.37 **DekaWashingtons** and Jill's collection is represented by 3689.2 **DeciWashingtons**.
i. The tabular number-phrases are:

	Kilo Washingtons	Hecto Washingtons	Deka Washingtons	Washingtons	Deci Washingtons	Centi Washingtons
Jack		5	4	3	7	
Jill		3	6	8	9	2

ii. Since Jack has more **HectoWashingtons** than Jill has, and even though he has fewer **DekaWashingtons**, fewer **Washingtons**, fewer **DeciWashingtons** and fewer **CentiWashingtons** than Jill, we can write that

54.37 **DekaWashingtons** > 3689.2 **DeciWashingtons**

or that

3689.2 **DeciWashingtons** < 54.37 **DekaWashingtons**

5 Graphing Sets of Number Phrases

graph
ruler
label
arrowhead
tick mark
origin
solid dot
hollow dot

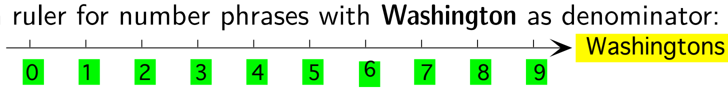
Once we have *represented* a collection with a number phrase, we will often also want to **graph** that number phrase.

1. For that purpose, we will use **rulers** which are straight lines with a few **labels**:

- an **arrowhead** to indicate the way *numerators* go *up*,
- **tick marks** to be labeled with *numerators*,
- the *denominator* next to the arrowhead.

AGREEMENT 4.2
Rulers will always have to include an **origin**, that is a tickmark labeled 0.

EXAMPLE 4.31.
Here is a ruler for number phrases with **Washington** as denominator:



2. Then, we will *graph* a number phrase with a dot which will be either:

A **solid dot** ● to graph a numerator which IS in the set.

or

A **hollow dot** ○ to graph a numerator which IS NOT in the set,

EXAMPLE 4.32.
The *graph* for the number phrase 3 **Washingtons** could be either:

if the number phrase 3 IS in the set we want to graph,

or

if the number phrase 3 IS NOT in the set we want to graph.

comparison problem
data set
requirement
specifying form
box
solutions
solution subset
comparison formula
unspecified numerator

6 Meeting A Requirement

1. Very often, we have to solve **comparison problems**, that is given a collection of items, called the **data set**, we need to get those items which meet a **requirement** given in the shape of a **specifying form**, that is a sentence with an empty **box** in which to fill in names of items. Those items whose name turn the specifying form into a TRUE sentence are called **solution** and make up the **solution subset** for the given comparison problem.

EXAMPLE 4.33.
Given the *data set*
{George W. Bush, Hillary Clinton, John Kennedy, Barack Obama }
and the *specifying form*

[ ] is a past President of the United States.

- Since the sentence

  [George W. Bush] is a past President of the United States.

  is TRUE, George W. Bush is a *solution*.
- Since the sentence

  [Hillary Clinton] is a past President of the United States.

  is FALSE, Hillary Clinton is *not* a *solution*.
- Etc

So, the *solution subset* of the comparison problem is:
{George W. Bush, John Kennedy}

2. Usually, instead of using *comparison forms*, we will use **comparison formulas** that is, instead of a *box*, we will use:

- the letter x to work as an **unspecified numerator**.

followed by

- the *denominator* of the number phrases in the data set,

EXAMPLE 4.34.
Instead of writing the comparison *form*

[ ] $>$ 5 Apples,

we will write the comparison *formula*

x Apples $>$ 5 Apples,

3. However, in order to focus, we will usually **declare** the denominator up front and then we will deal with just the *numerators*.

declare
basic comparison problems
gauge number phrase

EXAMPLE 4.35.
Instead of writing the comparison *form*

$$\boxed{} > 5 \text{ Apples},$$

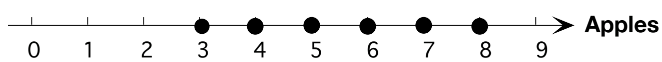
we *declare* that the denominator is Apples and the comparison formula is:

$$x > 5$$

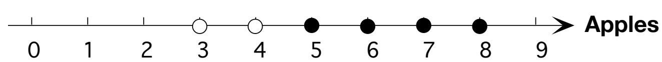
Declaring the denominator makes writing comparison problems a lot simpler.

EXAMPLE 4.36.
After declaring that the denominator is **Apples**, we just write:

- Data set: $\{3, 4, 5, 6, 7, 8\}$
  which we can graph as:



- Specifying formula: $x \geqq 5$
- Solution subset: $\{5, 6, 7, 8\}$
  which we can graph as:



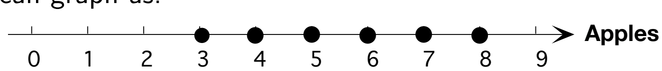
4. In this course, we will limit ourselves to **basic comparison problems**, that is, given a data set of number phrases, we will just compare them to a **gauge number phrase**.

EXAMPLE 4.37.
Find the solution subset of the comparison problem where the denominator is **Apples** and whose

- Data set is

$$\{3, 4, 5, 6, 7, 8\}$$

  which we can graph as:



- Comparison formula is

$$x \geqq 5$$

  where 5 is the gauge numerator.

inequation
equation

However, we will emphasize the distinction between:

- **Inequations**[1] which are comparison formulas with any one of the five comparison verbs: $>$ $<$ $\neq$ $\leqq$ $\geqq$
- **Equations** which are comparison formulas with the comparison verb $=$

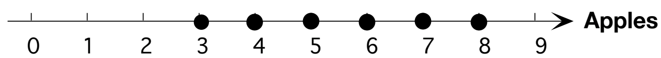
5. Finding the solution subset of basic comparison problem is straightforward: We just try each number phrase in the data set.

EXAMPLE 4.38.

We declare that we are dealing with **Apples**. Given the data set

$$\{3, 4, 5, 6, 7, 8\},$$

which we can graph as:



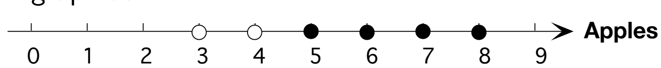
and the comparison formula

$$x \geqq 5$$

we try each numerator in the data set:

$3 \geqq 5$ which is FALSE,
$4 \geqq 5$ which is FALSE,
$5 \geqq 5$ which is TRUE,
$6 \geqq 5$ which is TRUE,
$7 \geqq 5$ which is TRUE,
$8 \geqq 5$ which is TRUE,

So, the solution subset is
{5, 6, 7, 8}
that is:
5 **Apples**, 6 **Apples**, 7 **Apples**, 8 **Apples**
which we can graph as:



[1]Although supposedly concerned with the relevance of mathematics to the "real world", Educologists are strangely indifferent to the fact that, *in real life*, inequations are vastly more prevalent than equations. Not to mention that mixing up (in)equa*tions* and (in)equa*lities* cannot help.

initial state
final state
agent of change
initial collection
final collection
action

Chapter 5

Addition - Subtraction

1 States And Actions

In the previous Chapter, we compared given collections to a given gauge collection. Much more often, though, we compare collections *not* in their **initial state**, that is as given, but only *after* they have been changed to a **final state** by some **agent of change**. For short, we will often say **initial collection** instead of collection in the initial state and **final collection** instead of collection in the final state.

Then, the **action** of an *agent of change* is:

$$\text{Initial collection} \xrightarrow{\textit{Agent of Change}} \text{Final collection}$$

EXAMPLE 5.1.
The sun is the agent that changes APPLES from being in a green state to being in a ripe state. In other words, the *action* of the sun is:

$$\text{Collection of } \textit{green} \text{ APPLES} \xrightarrow{\textit{Sun}} \text{Collection of } \textit{ripe} \text{ APPLES}$$

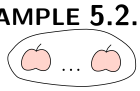
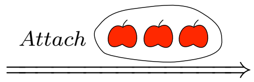
attach
add-to collection
detach
take-from collection

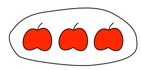
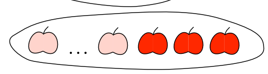
2 Attaching And Detaching A Collection

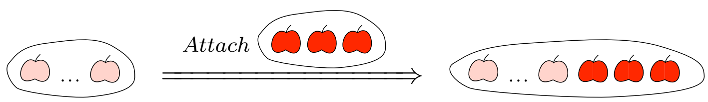
In this chapter, we will deal only with two related kinds of agents of change.

1. The first kind of agents of change **attaches** a given **add-to collection** *to* a collection in the initial state to get the final state of that collection.

Example 5.2.

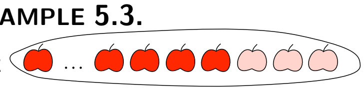
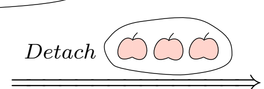
Let  be the *initial state* some collection is in. After using the agent of change 

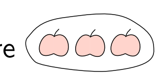
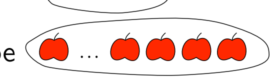
, where  is the *add-to collection*, the *final state* of the collection will be . In short, the *action* is:

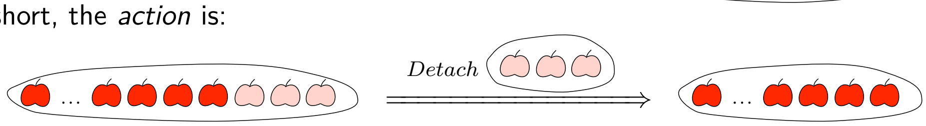


2. The second kind of agents of change **detaches** a given **take-from collection** *from* a collection in the initial state to get the final state of that collection.

Example 5.3.

Let  be the *initial state* some collection is in. After using the agent of change 

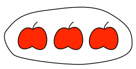
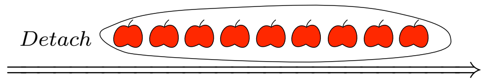
, where  is the *take-from collection*, the *final state* of the collection will be . In short, the *action* is:



3. Of course, contrary to what happens with *attaching* agents of change, *detaching* agents of change do not always work since we cannot detach items that are not already in the initial state of the collection.

Example 5.4.

function
input-output rule
unspecified input
specific inputs
function name
output specifying code
specific outputs
adding function
addition
+

Let  be the *initial state* a collection is in. Obviously, we cannot use the agent of change .

3 Translations

Real world *agents of change* are represented on paper by paper world **functions** which we specify with an **input-output rule** that consists of:

i. An **unspecified input** eventually to be replaced by **specific inputs**, that is the number phrases that represent the *initial collections*.

ii. A **function name**, that is the name of the function that represents the agent of change

iii. The **output specifying code** which is the code that specifies the *output* of the function in terms of the input. The **specific outputs** are the number phrases that represent the final collections.

Thus, the *input-output rule* of a function

$$\underbrace{\text{Unspecified input}}_{\text{To be replaced by specific inputs}} \xrightarrow{\text{Name of function}} \underbrace{\text{Output-specifying code}}_{\text{Specifies what to do with the inputs to get the outputs}}$$

represents on paper the real world *action* of an agent of change

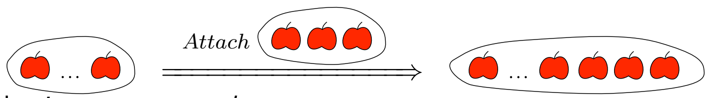
$$\text{Initial collection} \xRightarrow{\textit{Agent of Change}} \text{Final collection}$$

We now look at the functions that represent the agents of change which *attach* or *detach* a given collection.

1. An **adding function**, often called **addition** for short, is a function that represents an agent that *attaches* a given add-to collection to an initial collection. The *function name* for *adding functions* consists of the symbol + to represent *attaching* followed by the number phrase that represents the *add-to collection*.

Example 5.5.
To represent on paper the real world *action*

subtracting function
subtraction
–



we write the *input-output rule*

$$\underbrace{x \ \text{Apples}}_{\textit{Unspecified input}} \xrightarrow{+\,3 \ \text{Apples}} \underbrace{\big(x + 3\big) \ \text{Apples}}_{\textit{Output specifying code}}$$

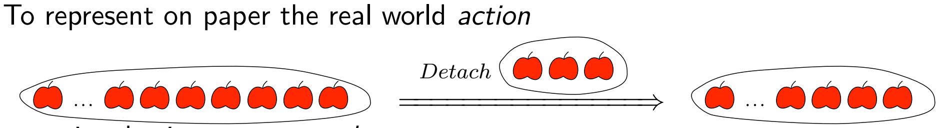
or, if the denominator Apples has been previously *declared*, just

$$\underbrace{x}_{\textit{Unspecified input}} \xrightarrow{+3} \underbrace{x + 3}_{\textit{Output specifying code}}$$

> **LANGUAGE 5.1** **T** his use of the symbol $+$ to represent *attachment* is only the first of many different uses of the symbol $+$ and this will create decoding difficulties. We will deal with these difficulties one at a time, as we encounter each new use of the symbol $+$.

2. A **subtracting function**, often called **subtraction** for short, is a function that represents an agents that *detaches* a given take-from collection from an initial collection. The *function name* for subtracting functions consists of the symbol $-$ to represent *detaching* followed by the number phrase that represents the *take-from collection*.

EXAMPLE 5.6.
To represent on paper the real world *action*



we write the *input-output rule*

$$\underbrace{x \ \text{Apples}}_{\textit{Unspecified input}} \xrightarrow{-\,3 \ \text{Apples}} \underbrace{\big(x - 3\big) \ \text{Apples}}_{\textit{Output specifying code}}$$

execute
long addition
carryover

or, if the denominator **Apples** has been previously *declared*, just

$$\underbrace{x}_{\textit{Unspecified input}} \xrightarrow{-3} \underbrace{x - 3}_{\textit{Output specifying code}}$$

> **LANGUAGE 5.2** This use of the symbol $-$ to represent *detachment* is only the first among of many different uses of the symbol $-$ and this will create decoding difficulties. We will deal with these difficulties one at a time, as we encounter each new use of the symbol $-$.

4 Procedures For Additions and Subtractions

We now describe the procedures involved in **executing** the output specifying code, once we have specified the input, that is once we have replaced x by a specific input numerator.

1. For *addition*, we use either one of two procedures depending on whether the add-to numerator is *basic* or *large*:

i. When the add-to numerator is *basic*, the procedure is just to count *up* from the input numerator by a number equal to the *add-to numerator*.

EXAMPLE 5.7.
In order to *add* 5 **Meters** *to* 13 627.48 **Meters** we count 5 *up* from 13 627.48:

$$\xrightarrow{13\,628.48,\;\; 13\,629.48,\;\; 13\,630.48,\;\; 13\,631.48,\;\; 13\,632.48}$$

so that:

$$\Big[13\,627.48 + 5\Big] \textbf{ Meters} = 13\,632.48 \textbf{ Meters}$$

ii. When the add-to numerator is *large* the procedure is to turn the *decimal-phrases* back into *array-phrases* that is to place the numerators under a header as in section 6 Decimal-phrases and then to use **long addition**, the procedure we learned in elementary school and which is really nothing more than *counting up* with "**carryover**".

long subtraction
borrowing

EXAMPLE 5.8.
In order to add 526.003 **Meters** *to* 4 627.47 **Meters** we place both numerators under a *decimal header*:

THOUSAND	HUNDRED	TEN	SINGLE	TENTH	HUNDREDTH	THOUSANDTH
4	6	2	7	4	7	
	5	2	6	0	0	3

and then we do the *long addition* under the header:

THOUSAND	HUNDRED	TEN	SINGLE	TENTH	HUNDREDTH	THOUSANDTH
1		1				
4	6	2	7	4	7	
	5	2	6	0	0	3
5	1	5	3	4	7	3

Which gives us the *decimal number-phrase*:

5 153.473 **Meters**

so that:

$$[4\,627.47 + 526.003]\ \textbf{Meters} = 5\,153.473\ \textbf{Meters}$$

2. For *subtraction*, we use either one of two procedures depending on whether the subtract-from numerator is *basic* or *large*:

i. When the take-from numerator is *basic*, the procedure is just to count *down* from the input numerator by a number equal to the *take-from numerator*.

EXAMPLE 5.9.
In order to subtract 3 **Meters** from 13 627.48 **Meters** we can count 3 *down* from 13 627.48:

$$\xrightarrow{13\,626.48,\quad 13\,625.48,\quad 13\,624.48}$$

so that:

$$\Big[13\,627.48\ -\ 3\Big]\ \textbf{Meters} =\ 13\,624.48\ \textbf{Meters}$$

ii. When the take-from numerator is *large* the procedure is to turn the *decimal-phrases* back into *array-phrases* that is to place the numerators under a header as in section 6 Decimal-phrases and then to use **subtraction addition**, the procedure we learned in elementary school and which is really nothing more than *counting down* with "**borrowing**".

translating function
translated set

Example 5.10.
In order to subtract 627.48 Meters *from* 5 796.3 Meters we place both numerators under a *decimal header*:

Thousand	Hundred	Ten	Single	Tenth	Hundredth	Thousandth
5	7	9	6	3	0	
	6	2	7	4	8	

and then we do the *long subtraction* under the header:

Thousand	Hundred	Ten	Single	Tenth	Hundredth	Thousandth
		8	5	2		
5	7	~~9~~	~~6~~	~~3~~	10	
	6	2	7	4	8	
5	1	6	8	8	2	

Which gives us the *decimal number-phrase*:

5 168.82 Meters

so that:

$[5\,796.3 - 627.48]$ Meters $=$ 5 168.82 Meters

3. While most of what we do with subtraction looks very much like what we did with addition, there is a most important difference between addition and subtraction: while we can *always* perform an *addition*, we can perform a *subtraction* only when the take-from numerator is *no more than* the *input*.

Example 5.11.

Example 5.12.

Example 5.13.

Example 5.14.

5 Translations

Both adding functions and subtracting functions are **translating functions** in that, graphically, they slide the whole *data set* by the add-to or take-from number phrase into the **translated set**.

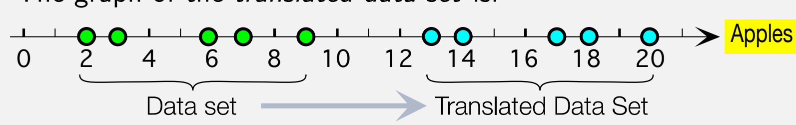
Example 5.15.
In order to translate
the data set: $\underbrace{\{2, 3, 9, 6, 7\}}_{\text{Numerator Set}}$ $\underbrace{\text{Apples}}_{\text{Common Denominator}}$

with the adding function: $\underbrace{x}_{\text{Unspecified input}} \xrightarrow{+11} \underbrace{x + 11}_{\text{Output specifying code}}$

i. We *specify* x to be each and every one of the numerators in the numerator set:

- $x \xrightarrow{+11} x + 11 \Big|_{x \leftarrow 2}$ gives $2 \xrightarrow{+11} 13$ so the *output* is 13.
- $x \xrightarrow{+11} x + 11 \Big|_{x \leftarrow 3}$ gives $3 \xrightarrow{+11} 14$ so the *output* is 14.
- $x \xrightarrow{+11} x + 11 \Big|_{x \leftarrow 9}$ gives $9 \xrightarrow{+11} 20$ so the *output* is 20.
- $x \xrightarrow{+11} x + 11 \Big|_{x \leftarrow 6}$ gives $6 \xrightarrow{+11} 17$ so the *output* is 17.
- $x \xrightarrow{+11} x + 11 \Big|_{x \leftarrow 7}$ gives $7 \xrightarrow{+11} 18$ so the *output* is 18.

ii. The *translated data set* is therefore $\{13, 14, 20, 17, 18\}$ Apples.
iii. The graph of the *translated data set* is:



Example 5.16.
In order to translate
the data set: $\underbrace{\{18, 13, 19, 15, 17\}}_{\text{Numerator Set}}$ $\underbrace{\text{Apples}}_{\text{Common Denominator}}$

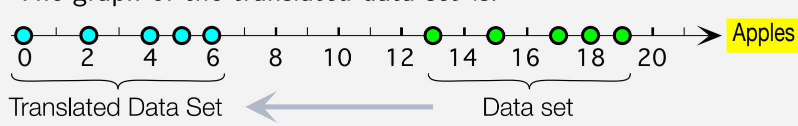
with the subtracting function: $\underbrace{x}_{\text{Unspecified input}} \xrightarrow{-13} \underbrace{x - 13}_{\text{Output specifying code}}$

i. We *specify* x to be each and every one of the numerators in the numerator set:

- $x \xrightarrow{-13} x - 13 \Big|_{x \leftarrow 18}$ gives $18 \xrightarrow{-13} 5$ so the *output* is 5.
- $x \xrightarrow{-13} x - 13 \Big|_{x \leftarrow 13}$ gives $13 \xrightarrow{-13} 0$ so the *output* is 0.
- $x \xrightarrow{-13} x - 13 \Big|_{x \leftarrow 19}$ gives $19 \xrightarrow{-13} 6$ so the *output* is 6.
- $x \xrightarrow{-13} x - 13 \Big|_{x \leftarrow 15}$ gives $15 \xrightarrow{-13} 2$ so the *output* is 2.
- $x \xrightarrow{-13} x - 13 \Big|_{x \leftarrow 17}$ gives $17 \xrightarrow{-13} 4$ so the *output* is 4.

ii. The *translated set* is therefore $\{5, 0, 6, 2, 4\}$ **Apples**.

iii. The graph of the *translated data set* is:



EXAMPLE 5.17.

EXAMPLE 5.18.

EXAMPLE 5.19.

EXAMPLE 5.20.

reverse

6 Reversing A Translation

A very important feature of *translating functions* is that they can be **reversed**, that is, given a translating function of one kind, we can always find a translating function of the other kind which "undoes" the first one in that if we input the output of the first function into the second function, the output of the second function will be what we inputed in the first function.

- Given an adding function, there is a subtracting function which, whatever the input to the adding function, will take the output of the adding function as input and return as its output the original input to the adding function.

EXAMPLE 5.21.

Given the adding function $\xrightarrow{+7 \text{ Apples}}$, the subtraction function $\xrightarrow{-7 \text{ Apples}}$ undoes the action of the adding function on x **Apples**:

$$x \text{ Apples} \xrightarrow{+7 \text{ Apples}} x \text{ Apples} + 7 \text{ Apples}$$

$$x \text{ Apples} + 7 \text{ Apples} \xrightarrow{-7 \text{ Apples}} x \text{ Apples}$$

- Given an subtracting function, there is an adding function which, whatever the input to the subtracting function, will take the output of the subtracting function as input and return as its output the original input to the subtracting function.

EXAMPLE 5.22.

Given the subtracting function $\xrightarrow{-12 \text{ Apples}}$, the addition function $\xrightarrow{+12 \text{ Apples}}$ undoes the action of the subtracting function on x **Apples**:

$$x \text{ Apples} \xrightarrow{-12 \text{ Apples}} x \text{ Apples} - 12 \text{ Apples}$$

$$x \text{ Apples} - 12 \text{ Apples} \xrightarrow{+12 \text{ Apples}} x \text{ Apples}$$

EXAMPLE 5.23.

EXAMPLE 5.24.

reverse problem

EXAMPLE 5.25.

EXAMPLE 5.26.

EXAMPLE 5.27.

7 Reverse Problems

We now want to represent on paper what we do when we compare a collection to a given gauge collection *after* it has been changed to a *final state* by attaching or detaching a given collection.

So, on paper, given a *data set* and a *function*, a **reverse problem** will be a comparison problem in which it is the *outputs* of the function which we want to compare to the given *gauge numerator*.

EXAMPLE 5.28.
Given the *reverse* addition problem

- *Data set*: $\underbrace{\{2, 3, 9, 6, 7\}}_{\text{Numerator Set}}$ $\underbrace{\text{Apples}}_{\text{Common Denominator}}$
- *Function*: $\underbrace{x}_{\text{Unspecified input}} \xrightarrow{+4} \underbrace{x+4}_{\text{Output specifying code}}$
- *Comparison formula*: $\underbrace{x+4}_{\text{Output specifying code}} \leqq \underbrace{7}_{\text{Gauge Numerator}}$

  where $\leqq$ is the *comparison verb*.

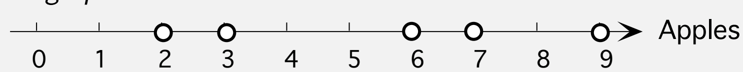
In order to solve this comparison problem,

i. We *specify* x to be each and every numerator in the numerator set:

- $x + 4 \leqq 7 \Big|_{x \leftarrow 2}$ gives the *comparison sentence* $6 \leqq 7$ which is TRUE
- $x + 4 \leqq 7 \Big|_{x \leftarrow 3}$ gives the *comparison sentence* $7 \leqq 7$ which is TRUE
- $x + 4 \leqq 7 \Big|_{x \leftarrow 9}$ gives the *comparison sentence* $13 \leqq 7$ which is FALSE
- $x + 4 \leqq 7 \Big|_{x \leftarrow 6}$ gives the *comparison sentence* $10 \leqq 7$ which is FALSE
- $x + 4 \leqq 7 \Big|_{x \leftarrow 7}$ gives the *comparison sentence* $11 \leqq 7$ which is FALSE

ii. The *solution subset* is therefore $\{2, 3\}$ **Apples**.

iii. The *graph* of the solution subset is



Example 5.29.

Example 5.30.

Example 5.31.
Given the *reverse* subtraction problem

- *Data set*: $\underbrace{\{2, 3, 9, 6, 7\}}_{\text{Numerator Set}}$ $\underbrace{\textbf{Apples}}_{\text{Common Denominator}}$
- *Function*: $\underbrace{x}_{\text{Unspecified input}} \xrightarrow{-4} \underbrace{x - 4}_{\text{Output specifying code}}$
- *Comparison formula*: $\underbrace{x - 4}_{\text{Output specifying code}} \leqq \underbrace{8}_{\text{Gauge numerator}}$

where $\leqq$ is the *comparison verb*

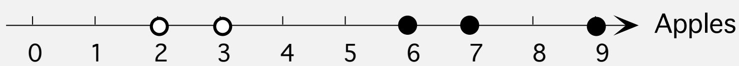
In order to solve this comparison problem,

i. We *specify* x to be each and every numerator in the numerator set:

- $x - 4 \leqq 7 \Big|_{x \leftarrow 2}$ gives the *comparison sentence* $2 - 4 \leqq 7$ but since we cannot execute $2 - 4$, the sentence is FALSE.
- $x - 4 \leqq 7 \Big|_{x \leftarrow 3}$ gives the *comparison sentence* $3 - 4 \leqq 7$ but since we cannot execute $3 - 4$, the sentence is FALSE.
- $x - 4 \leqq 7 \Big|_{x \leftarrow 9}$ gives the *comparison sentence* $5 \leqq 7$ which is TRUE.
- $x - 4 \leqq 7 \Big|_{x \leftarrow 6}$ gives the *comparison sentence* $2 \leqq 7$ which is TRUE.
- $x - 4 \leqq 7 \Big|_{x \leftarrow 7}$ gives the *comparison sentence* $3 \leqq 7$ which is TRUE.

ii. The *solution subset* is therefore $\{9, 6, 7\}$ **Apples**.

iii. The *graph* of the solution subset is



Example 5.32.

Example 5.33.

Chapter 6

Multiplication - Division

Multiplication and division are very different from addition and subtraction in several different ways.

1 Can Money Be Multiplied By Money?

A major way in which *multiplication* differs from both *addition* and *subtraction* is that while we could add and/or subtract number phrases (with a common denominator) and get as a result a number phrase (with that same common denominator), we usually cannot multiply number phrases and get as a result a number phrase.

Example 6.1. Given the number phrase 7 **Dimes**, we can add or subtract the number phrase 2 **Dimes** to represent attaching or detaching a collection but what could multiplying 7 **Dimes** by 3 **Dimes** possibly represent?

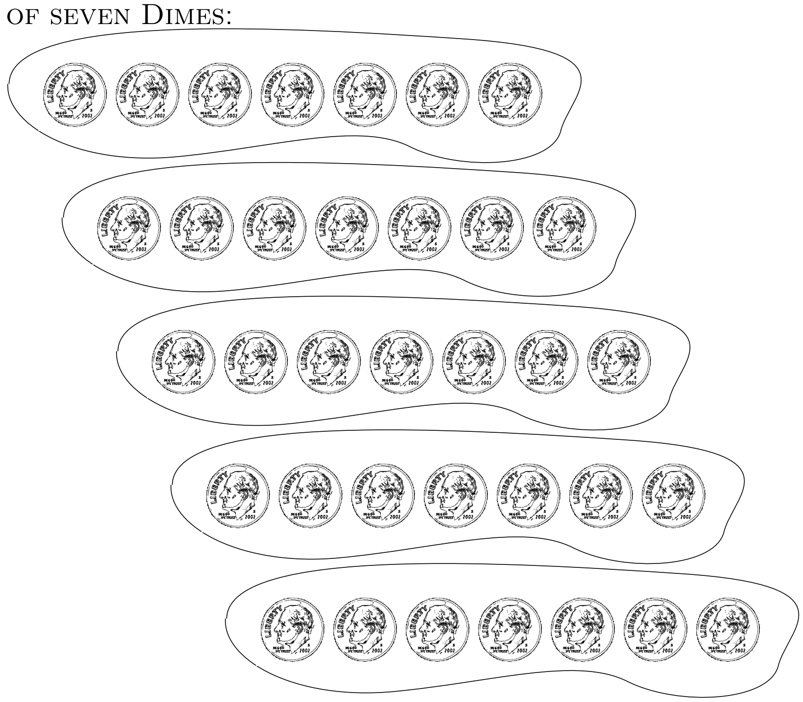
2 Repeated Addition

A number phrase in which the *denominator* is itself a *number phrase* represents a *collection of collections*.

repeated addition of number phrases
multiply

EXAMPLE 6.2. The number phrase 7 **Dimes** represents the collection

and the number phrase 5[7 **Dimes**] represents a collection of FIVE COLLECTIONS OF SEVEN DIMES:



We get the *total number of items* by **repeated addition of number phrases**, that is by adding the number phrases which represent the identical collections being collected. But then it is usual to say that this is what **multiplying** the number of collections being collected by the number of items in the identical collections being collected does.

EXAMPLE 6.3. The total number of DIMES in the previous example is represented by:

$$\begin{aligned} 7\,\textbf{Dimes} + 7\,\textbf{Dimes} + 7\,\textbf{Dimes} + 7\,\textbf{Dimes} + 7\,\textbf{Dimes} &= [7+7+7+7+7]\,\textbf{Dimes} \\ &= 35\,\textbf{Dimes} \end{aligned}$$

and then it is usual to that this is

$$= [5 \times 7]\,\textsf{Dimes}$$

However:

i. 5×7 gets the result of a *repeated addition of number phrases* but, as already mentioned at the outset, does *not* get the result of a multiplication of number phrases the way addition and subtraction are addition and subtraction *of number phrases*.

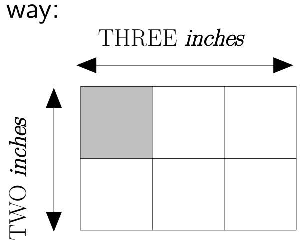
ii. Moreover, even looking at multiplication as replacing repeated addition only works *when the numerator is a* counting number but this does not extend to when the numerator is a *decimal number*.

EXAMPLE 6.4. While $5[7\,\textsf{Dimes}]$ represents a collection of FIVE COLLECTIONS OF SEVEN DIMES, what collection of COLLECTIONS OF SEVEN DIMES could $5.0384[7\,\textsf{Dimes}]$ possibly represent?

3 Computing Areas

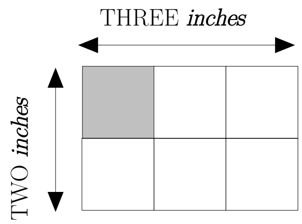
In certain rare cases, a multiplication *of number phrases* does make sense but the denominator of the *result* is *different* from the denominator of the original number phrases.

EXAMPLE 6.5. $2\,\textsf{Inches} \times 3\,\textsf{Inches} = [7 \times 3]\,\textsf{SquareInches}$
which represents the *area* of a TWO-BY-THREE RECTANGLE, that is a **rectangle** that is TWO INCHES LONG one way and THREE INCHES LONG the other way:



Indeed, if we want to tile this rectangle with ONE-INCH-BY-ONE-INCH MOSAICS we get

worth
unit-worth
exchange
exchange rate
co-number phrase
co-multiply



Counting the MOSAICS shows that we will need SIX ONE-INCH-BY-ONE-INCH MOSAICS.

But this type of multiplication *does* extend to *decimal numbers*.

4 Co-multiplication

We seldom deal with a collection without wanting to know what the (money?) **worth** of the collection is, that is how much money the collection could be exchanged for.

EXAMPLE 6.6. Given a collection of FIVE APPLES, and given that the *worth* of an APPLE is SEVEN CENTS, the real-world *process* for finding the *worth* of the collection is to exchange each APPLE for SEVEN CENTS. Altogether, we end up exchanging the whole collection for THIRTY-FIVE CENTS which is therefore the total *worth* of the collection.

1. A **unit-worth** of a given substance is the amount of another kind of substance we can **exchange** for one unit of the given substance. The real world **exchange rate** is then represented on paper by a **co-number phrase** in the shape of a "fraction".

EXAMPLE 6.7. Let the substance be GASOLINE. Then, if we can exchange each GALLONS OF GAS for 3.149 **Dollars**, we will represent this *exchange rate* by the co-number phrase $3.149\,\frac{\textsf{Dollars}}{\textsf{Gallon of Gas}}$ which we read 3.149 **Dollars** per **Gallon of Gas**

2. We obtain the worth of an amount of substance at a given exchange rate by **co-multiplying** the number phrase that represents the amount of substance by the co-number phrase that represents the unit worth of that substance. And of course exactly the same goes for collections of items.
The procedure for co-multiplying is quite simple:

i. multiply the *numerators*
ii. multiply the *denominators* with cancellation.

percentage

EXAMPLE 6.8. Given a collection of FIVE APPLES and given that the *worth* of ONE APPLE is SEVEN CENTS, the real-world *process* for finding the *worth* of the collection is to exchange each APPLE for SEVEN CENTS. Altogether, we end up exchanging the whole collection for THIRTY-FIVE CENTS which is therefore the total *worth* of the collection.
On paper, we write

$$5 \text{ Apples} \times 7 \frac{\text{Cents}}{\text{Apple}} = (5 \times 7) \left(\cancel{\text{Apples}} \times \frac{\text{Cents}}{\cancel{\text{Apple}}}\right)$$
$$= 35 \text{ Cents}$$

3. Co-multiplication is at the heart of a part of mathematics called DIMENSIONAL ANALYSIS (See https://en.wikipedia.org/wiki/Dimensional_analysis) that is much used in sciences such as PHYSICS, MECHANICS, CHEMISTRY and ENGINEERING where people have to "cancel" denominators all the time.

EXAMPLE 6.9.
$$5 \text{ Hours} \times 7 \frac{\text{Miles}}{\text{Hour}} = (5 \times 7) \left(\cancel{\text{Hours}} \times \frac{\text{Miles}}{\cancel{\text{Hour}}}\right) = 35 \text{ Miles}$$

EXAMPLE 6.10.
$$5 \text{ Square-Inches} \times 7 \frac{\text{Pound}}{\text{Square-Inch}} = (5 \times 7) \left(\cancel{\text{Square-Inches}} \times \frac{\text{Pound}}{\cancel{\text{Square-Inch}}}\right) = 35 \text{ Pounds}$$

Co-multiplication is also central to a part of mathematics called LINEAR ALGEBRA that is itself of major importance both in many other parts of mathematics and for all sort of applications in sciences such as ECONOMICS.

More modestly, *co-multiplication* also arises in **percentage** problems:

EXAMPLE 6.11.
$$5 \text{ Dollars} \times 7 \frac{\text{Cents}}{\text{Dollar}} = (5 \times 7) \left(\cancel{\text{Dollars}} \times \frac{\text{Cents}}{\cancel{\text{Dollar}}}\right) = 35 \text{ Cents}$$

assign
round
share
leftover

5 Sharing In The Real World

We first look at the *real-world process* and then we look at the corresponding *paper-world procedure*. In the real world, we often encounter situations in which we have to **assign** (equally) the items in a first collection to the items of another collection.

The *process* is to make **rounds** during each of which we *assign* one item of the first collection to each one of the items in the second collection. The process comes to an end when, after a round has been completed,

- there are items left unassigned but not enough to complete another round. The **share** is then the collection of items from the first collection that have been assigned to each item of the second collection and the **leftovers** are the collection of items from the first collection left unassigned after the process has come to an end.

Example 6.12. In the real world, say we have a collection of seven dollar-bills which we want to assign to each and every person in a collection of three person. We want to know how many dollar-bills we will assign to each person and how many dollar-bills will be left-over.
i. We make a *first round* during which we hand-out one dollar-bill to each and every person in the collection. This uses three dollar-bills and leaves us with four dollar-bills after the first round.
ii. We make a *second round*, we hand-out one dollar-bill to each and every person in the collection. This uses another three dollar-bills and leaves us with one dollar-bill after the second round.
iii. If we try to make a *third round*, we find that we cannot complete the third round.
So, the *share* is two dollar-bills and the *leftovers* is one dollar-bill.

or,

- there is no item left unassigned. The *share* is again the collection of items from the first collection that have been assigned to each item of the second collection and there are no *leftovers*.

Example 6.13. In the real world, say we have a collection of eight dollar-bills which we want to assign to each and every person in a collection of four person. We want to know how many dollar-bills we will assign to each person and how many dollar-bills will be left-over.

i. We make a *first round* during which we hand-out one dollar-bill to each and every person in the collection. This uses four dollar-bills and leaves us with four dollar-bills after the first round.
ii. We make a *second round*, we hand-out one dollar-bill to each and every person in the collection. This uses another four dollar-bills and leaves us with no dollar-bill after the second round.
iii. So, we cannot make a *third round*.
So, the *share* is two dollar-bills and there are no leftovers.

division
dividend
divisor
quotient
remainder
trial and error
try
partial product

6 Division On Paper

The paper *procedure* that corresponds to the real-world process is called **division**.

1. *Division* will involve the following language:

- The number-phrase that represents the first collection, that is the collections of items *to be assigned* to the items of the second collection, is called the **dividend**,
- The number-phrase that represents the second collection, that is the collection of items *to which* the items of the first collection are to be assigned, is called the **divisor**,
- The number-phrase that represents the *share* is called the **quotient**,
- The number-phrase that represents the *leftovers* is called the **remainder**.

EXAMPLE 6.14. Given a real-world situation with a collection of eight dollar-bills to be assigned to each and every person in a collection of four persons,

- The *dividend* is 7 **Dollars**
- The *divisor* is 3 **Persons**
- The *quotient* is $2\frac{\textbf{Dollars}}{\textbf{Person}}$
- The *remainder* is 1 **Dollar**

2. The *division procedure* taught in elementary schools is a **trial and error** procedure which follows the real-world process closely inasmuch as each *round* is represented by a **try** in which:

i. We use the *multiplication procedure* to find the **partial product** which represents how many items *have been used* by the end of the corresponding *real-world round*.

partial remainder

ii. We use the *subtraction procedure* to find the **partial remainder** which represents how many items, if any, are *left over* by the end of the corresponding *real-world round*.

EXAMPLE 6.15. In order to divide 987 by 321, we go through the following *tries*:

First try:

i. We multiply the *divisor* 321 by 1 which gives the *partial product* 321:

$$\begin{array}{r} 1 \\ 321 \;)\; \overline{987} \\ 321 \end{array}$$

ii. We subtract the *partial product* 321 from the *dividend* 987:

$$\begin{array}{r} 1 \\ 321 \;)\; \overline{987} \\ \underline{321} \\ 666 \end{array}$$

which leaves the *partial remainder* 666 which is *larger* than 321 therefore too large.

Second try:

i. We multiply the *divisor* 321 by 2 which gives the *partial product* 642:

$$\begin{array}{r} 2 \\ 321 \;)\; \overline{987} \\ 642 \end{array}$$

ii. We subtract the *partial product* 642 from the *dividend* 987:

$$\begin{array}{r} 2 \\ 321 \;)\; \overline{987} \\ \underline{642} \\ 345 \end{array}$$

which leaves the *partial remainder* 345 which is *larger* than 321 therefore too large.

Third try:

i. We multiply the *divisor* 321 by 3 which gives the *partial product* 963:

$$\begin{array}{r|l} & 3 \\ 321 & 987 \\ & 963 \end{array}$$

ii. We subtract the *partial product* 963 from the *dividend* 987:

$$\begin{array}{r|r} & 3 \\ 321 & 987 \\ & 963 \\ \hline & 24 \end{array}$$

which leaves the *partial remainder* 24 which is *smaller* than 321 so this is it! **Fourth try** (Just to check that this is really it.)

i. We multiply the *divisor* 321 by 4 which gives the *partial product* 1284:

$$\begin{array}{r|l} & 4 \\ 321 & 987 \\ & 1284 \end{array}$$

ii. We cannot subtract the *partial product* 1284 from the *dividend* 987:

$$\begin{array}{r|r} & 4 \\ 321 & 987 \\ & 1284 \\ \hline \end{array}$$

and indeed we cannot complete the fourth try and must go back to the last complete try, that is the third try, and we get that the *quotient* is 3 and the *remainder* 24.

This procedure, though, has two severe shortcomings:

- All these *full multiplications* require a lot of work.
- This procedure will *not* extend to *polynomials*

Example 6.16.

Example 6.17.

plain item
plain number phrase
plain item

Chapter 7

Signed Numbers – Comparisons

LANGUAGE 7.1 To make it clear which kind of collections we are talking about, from now on we will refer to the collections which were introduced in chapter 2 as collections of **plain items** and to the number phrases which represent them as **plain number phrases**

There are two issues with *plain* number phrases:

- While we can count *up* as far as we want we cannot always count *down* as far as we want.
- Number phrases can represent collections because all the items in a collection are of *one* kind but there are many situations in which items can come in either one of *two* kinds.

EXAMPLE 7.1.

– 3056.38 **Dollars** does not say if this was a *deposit* or a *withdrawal*,

– 37 800 **Dollars** does not say if a business is **in the red** (*owes* that money) or **in the black** (*has* that money).

– 62 **Dollars** does not say if a gambler is **ahead of the game** (has won more than s/he has lost) or **in the hole** (has lost more than s/he has won).

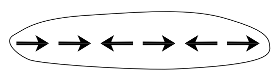
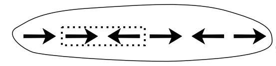
oriented items
orientation
opposite orientations
cancel
directed action
sided state
benchmark

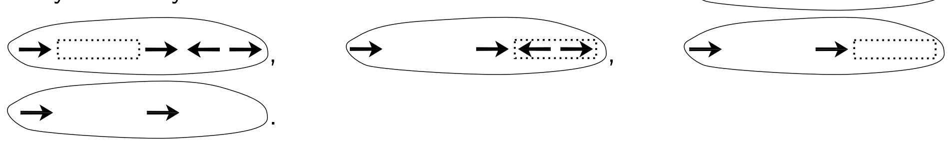
– 2 **Feet** does not say if a point is to the *left* or to the *right* of a benchmark.
– 5 **Inches** does not say if a point is *above* or *below* a baseline.

We will now develop a symbolic system which can do all that.

1 Oriented Items

1. In the real world, there are many situations where we have to deal with collections of **oriented items**, that is items with *either one of two* **orientations** but where items with **opposite orientations cancel** each other so that collections of oriented items can only involve items that are all oriented the same way.

EXAMPLE 7.2. The collection  reduces automatically to only items with the same orientation: ,



2. In the real-world, *oriented items* generally fall into either one of two categories:

- Items called **directed actions** which are "moves" of one kind or another but that can go either in *this-direction* or *that-direction*.

  EXAMPLE 7.3.
  – a businesswoman may *deposit* three thousand dollar on a bank account or may *withdraw* three thousand dollars from a bank account.
  – a gambler may *win* sixty-two dollars or may *lose* sixty-two dollars.
  – on a horizontal line, a point can be moved two feet *leftward* or two feet *rightward*
  – on a vertical line, a point can be moved five inches *upward* or five inches *downward*

- Items called **sided states** which can be either on *this-side* or *that-side* of some **benchmark**.

declare
standard direction
opposite direction
standard side
opposite side
signed numerator
sign
positive numerator

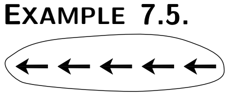
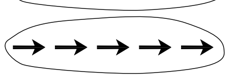
EXAMPLE 7.4.

- a business may be three thousand dollars *in the red* or three thousand dollars *in the black*
- a gambler may be sixty-two dollars *ahead of the game* or sixty-two dollars *in the hole*.
- on a horizontal line with some benchmark, a point may be two feet *to the left* of the benchmark or two feet *to the right* of the benchmark
- on a vertical line with some baseline, a point may be five inches *above* the benchmark or five inches *below* the benchmark

2 Signed Number Phrases

LANGUAGE 7.2
To make it clear which kind of numerator we are talking about, we will refer to the numerators which were introduced in Chapter 2 as **plain numerators**.

To *represent* collections of oriented items on paper, we could just use plain numerators and two denominators, one for each orientation.

EXAMPLE 7.5. We could represent the collection of oriented items  by the plain number phrase 5 **Left Arrows** and the collection  by the plain number phrase 5 **Right Arrows**

However, even though orientation is arguably a *qualitative* issue, representing the *orientation* as part of the *numerator* instead of as part of the *denominator* will enormously facilitate *computations*.

1. The first thing we have to do is to **declare**

- which direction is to be the **standard direction** and which direction is to be the **opposite direction**,
- which side of the benchmark is going to be the **standard side** and which side is to be the **opposite side**,

2. Then, a **signed numerator** will consists of:

- A **sign** for which, traditionally, we use + to indicate one orientation and − to indicate the other orientation. Numerators signed with + are called **positive numerators** and numerators signed with − are

negative numerator
opposite sign
size
signed number-phrase

called **negative numerators**. Thus, **opposite signs** represent opposite orientations.

- A **size** which is a *plain numerator*.

EXAMPLE 7.6.

> **NOTE 7.1** From now on, the symbol $+$ will thus be used for two purposes as $+$ can now be:
> - The *symbol* for addition of plain numerators as in Chapter 5,
> - The *sign* of a positive numerator as now.

In order to make it always clear for which purpose $+$ is being used, we use

> **AGREEMENT 7.1 I** n this text, $+$ as *sign* of a positive numerator will *never* "go without saying".

Indeed, with **AGREEMENT 7.3** we will always know whether $+$ stands for the *symbol* for plain addition or for the *sign* of a positive numerator.

EXAMPLE 7.7. In $2 + 5$, the symbol $+$ cannot be the sign of 5 because, by **AGREEMENT 7.3**, 2 has to be a *plain* numerator and what could a *plain* numerator followed by a *signed* numerator possibly represent?

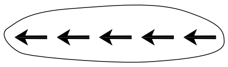
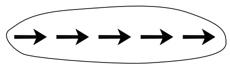
But the price for that is that if we want to write a *positive* numerator, we *must* write the sign $+$ as, otherwise, by **AGREEMENT 7.3**, with no $+$, the numerator will be seen as being *plain*.

EXAMPLE 7.8. If we want to talk about the opposite of -5, we *must* write $+5$ because if we write just 5 this will be seen as a *plain* numerator which is not the opposite of anything and has no opposite.

3. Then, a **signed number-phrase** will consist of:

- a signed-numerator
- a denominator

signed ruler

Example 7.9. We declare that + will represent *right* steps and therefore that − will represent *left* steps. Then, to represent the collection  we will use the signed number phrase -5 **Arrows** and to represent the collection  we will use the signed number phrase $+5$ **Arrows**

Example 7.10.
We declare that the *standard* direction is to *win* money so that to *lose* money is the *opposite* direction. Then,

When a *real-world* gambler:	We write on *paper*:
• *wins* sixty-two dollars	$+62$ **Dollars**
• *loses* sixty-two dollars	-62 **Dollars**

in which $+62$ is a *positive* signed-numerator and -62 is a *negative* signed-numerator.

Example 7.11.
We declare that the *standard* side is *in-the-black* so that *in-the-red* is the *opposite* side. Then,

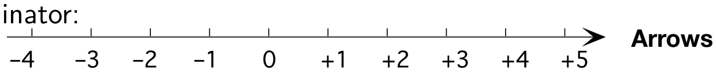
When a *real-world* business is:	We write on *paper*:
• three thousand dollars *in-the-black*	$+3000$ **Dollars**
• three thousand dollars *in-the-red*	-3000 **Dollars**

in which $+3000$ is a *positive* signed-numerator and -3000 is a *negative* signed-numerator.

3 Graphing Signed Number Phrases

1. To *graph* signed number phrases, we use **signed rulers**.

Example 7.12. Here is a signed ruler for signed numerators with **Arrows** as denominator:



Just as with plain number phrases, we will use *solid dots* and *hollow dots* to graph signed number phrases.

plain counting

2. From the *graphic* viewpoint:

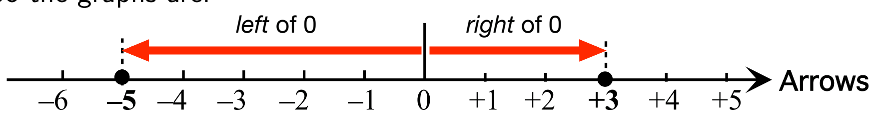
- The *sign* of a signed numerator codes *which side* of 0 the graph of the signed numerator is.

Example 7.13. Since

Sign of -5 $= -$, the signed numerator -5 is left of 0.

Sign of $+3$ $= +$, the signed numerator $+3$ is right of 0.

So the graphs are:

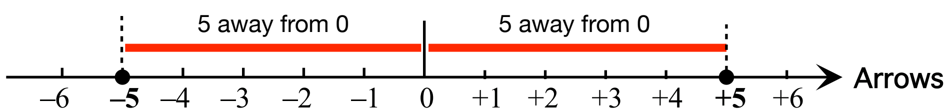


- The *size* of a signed numerator codes *how far away* from 0 the signed numerator is on a signed ruler.

Example 7.14. Since

Size of -5 is 5, the signed numerator -5 is 5 away from 0,

Size of $+5$ is 5, the signed numerator $+5$ is 5 away from 0.



4 Signed Counting

1. In Chapter 2, we used **plain counting** and:

- Starting from a given numerator, we could count *up* any number of steps.

Example 7.15.

$$\xrightarrow{21,22,23,\ldots,134,135,136,\ldots}$$

but

- Starting from a given numerator, we could count *down* only a certain number of steps,

Example 7.16. Starting from 42, we could count down *at most* 41 steps.

$$\xrightarrow{41,\ 40,\ 39,\ \ldots\ ,\ 4,\ 3,\ 2,\ 1,\ 0}$$

2. With *signed* numerators, the situation is much simpler: starting from *any* signed numerator, we can count *any* number of steps *up* or *down*. smaller

EXAMPLE 7.17. From $+7$, we can count *up* 17 to $+24$:

$$\xrightarrow{\text{+8, +9, +10, +11, ... , +23, +24}}$$

EXAMPLE 7.18. From $+20$, we can count *down* 7 to $+13$:

$$\xrightarrow{\text{+19, +18, ... , +14, +13}}$$

EXAMPLE 7.19. From $+8$, we can count *down* 13 to -5:

$$\xrightarrow{\text{+7, +6, ... , +2, +1, 0, -1, -2, -3, -4, -5}}$$

EXAMPLE 7.20. From -7, we can count *up* 13 to $+6$:

$$\xrightarrow{-7,-6,-5,...,-2,-1,0,+1,+2,...,+5,+6}$$

EXAMPLE 7.21. From -4, we can count *down* 5 to -9:

$$\xrightarrow{\text{+52, +51, +50, +49, ... , +2, +1, 0, -1, -2, -3, ... , -22, -23}}$$

5 Comparisons Of Signed Number Phrases

We can compare *signed* number phrases from two different viewpoints depending on whether or not we take the *sign* into consideration:

1. When we compare the signed numerators *themselves*, that is when we *do* take the sign of the numerators into consideration:

- In this text, we will use the same comparison verbs we introduced for plain numerators but inside an "o" as a reminder that we need to take the "o"rientation of the items into consideration:
- We *count*, up or down as the arrowhead on a signed ruler goes.

EXAMPLE 7.22. In order to go from -5 **Arrows** to $+3$ **Arrows** we must count *up* so we write

$$-5 \textbf{ Arrows} \textcircled{<} +3 \textbf{ Arrows}$$

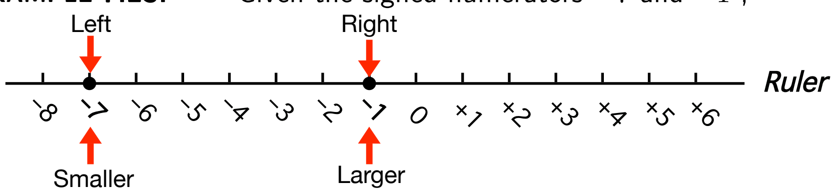
which we read as: Minus FIVE Arrows is oless than Plus THREE Arrows.

- Graphically, regardless of the signs of the two signed numerators,
  - The **smaller** signed numerator is to the *left*

larger
closer
farther
size-comparisons

– The **larger** signed numerator is to the *right*.

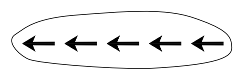
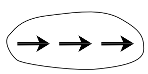
EXAMPLE 7.23. Given the signed numerators -7 and -1 ,



and we have $-7 \, (<) - 1$ -1*isrightmost*, -1*isthelargeroneandsince* -7*isleftmost*, -7*isthesmallerone* : $-1 > -7$ or $-7 < -1$

2. Often though, we will only want to know which of the two numerators is **closer** to the *origin* and/or which numerator is **farther** from the origin. Then we must *not* take the *sign* of the numerators into consideration. [1]

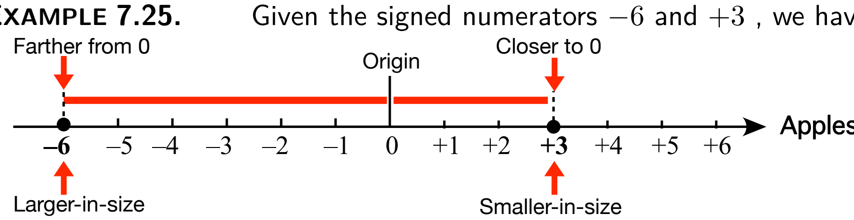
- Unfortunately, there are no symbols for **size-comparisons** and we will just have to write the words.
- We must compare only the *sizes* of the collections, just the way we compared the size of collections of *plain* items.

EXAMPLE 7.24. Given the collections  and  we write:

-5 **Arrows** *is larger in size than* $+3$ **Arrows**

- Graphically, size-comparison means comparing *how far* from the *origin* the two signed number phrases are.

EXAMPLE 7.25. Given the signed numerators -6 and $+3$, we have

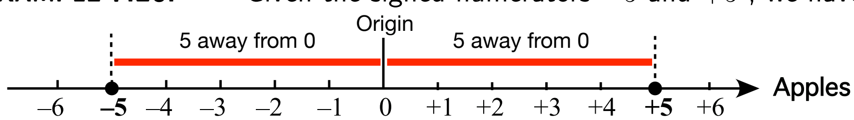


and since -6 is *farther away from* 0 than $+3$, -6 is *larger in size than* $+3$.

[1] Educologists will surely ask why not use **absolute values**? The answer is that *we* want to compare the *signed* numerators *themselves* while the Educologists would compare something entirely different, namely the *plain* numerators that are their sizes.

boundary
test
ray, solid
ray, hollow

Example 7.26. Given the signed numerators -5 and $+5$, we have



and so, since -5 and $+5$ are equally far from the origin, the signed numerator -5 and $+5$ have the same *size*, namely 5.

6 Meeting a Signed Decimal Requirement

When solving comparison problems where the data set is *counting* numerators, we just tried each numerator in the data set by *counting up* or *counting down* to the gauge numerator. With decimal numbers, though, we cannot count up or down.

1. The gauge will be the **boundary** of the solution subset in the sense that all solutions will be on one side or the other of the gauge. What the side is depends of course on the comparison verb but we have only to **test** one numerator on each side of the gauge. Then:

- If the tested numerator is a *solution*, then all the numerators on that side of the *boundary* are also *solutions*.
- If the tested numerator is a *non-solution* then all the numerators on that side of the *boundary* are also *non-solutions*.

2. In order to represent a *solution subset*,

i. We graph the *boundary* of the solution subset exactly the same way as we graphed *counting* number-phrases that is we use

- a *solid dot* ● to graph a boundary that is a *solution*.
- a *hollow dot* ○ to graph a boundary that is a *non-solution*:

ii. • We graph the *solution subset* with a **solid ray**

because this is what we would get if we were to draw a whole lot of *solid dots* right next to each other to graph all the *decimal* numerators that are solutions:

- We graph the part of the data set that is not the *solution subset* with a **hollow ray** because this is what we would get if we were to draw a whole lot of *hollow dots* right next to each other to graph all the *decimal* numerators that are non-solutions:

EXAMPLE 7.27. Given the comparison problem in **Dollars** in which

- the *data set* consists of all possible signed decimal numbers.
- the *comparison formula* is

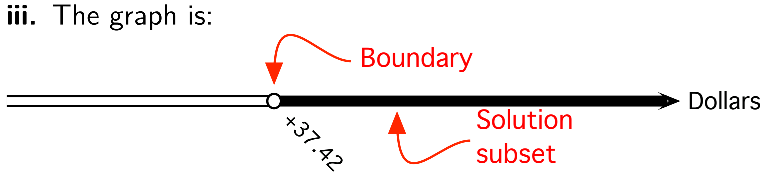
$$x > +37.42$$

i. The *boundary* of the solution subset is $+37.42$ but since $+37.42 > +37.42$ is FALSE, the boundary is a *non-solution*.

ii. To determine on which side of the boundary the *solution subset* is, we pick and test some number on each side of the boundary:

- We pick a numerator to the *left* of $+37.42$, for instance -50, and we try it in the comparison formula:
  $-50 > +37.42$ is FALSE
  so *all* the numerators to the *left* of $+37.42$ are *non-solutions*.
- We pick a numerator to the *right* of $+37.42$, for instance $+100$, and we try it in the comparison formula:
  $+100 > +37.42$ is TRUE
  so *all* the numerators to the *right* of $+37.42$ are *solutions*.

iii. The graph is:



EXAMPLE 7.28. Given the comparison problem in **Dollars** in which

- the *data set* consists of all possible signed decimal numbers.
- the *comparison formula* is

$$x \leqq -358.13$$

i. The *boundary* of the solution subset is -358.13 and since $-358.13 \leqq -358.13$ is TRUE, the boundary is a *solution*.

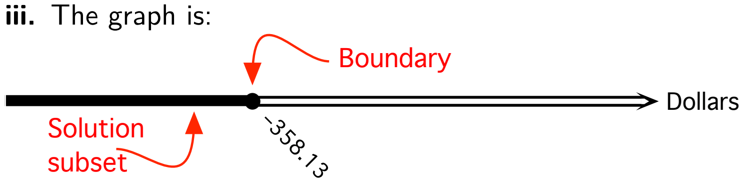
ii. To determine on which side of the boundary the *solution subset* is, we pick and test some number on each side of the boundary:

- We pick a numerator to the *left* of -358.13, for instance -500, and we try it in the comparison formula:

$-500 \leqq -358.13$ is TRUE
so *all* the numerators to the *left* of -358.13 are *solutions*.

- We pick a numerator to the *right* of -358.13, for instance 0, and we try it in the comparison formula:
  $0 \leqq -358.13$ is FALSE
  so *all* the numerators to the *right* of -358.13 are *non-solutions*.

iii. The graph is:



add-on collection of *oriented* items

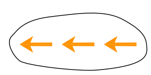
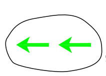
Chapter 8

Signed Addition

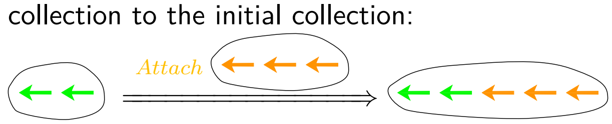
1 Attaching Collections of Oriented Items

Attaching an **add-on collection of *oriented* items** to an initial collection of *oriented* items is almost as simple as attaching an add-on collection of *plain* items to an initial collection of *plain* items. The only difference is that we must check the orientation of the items in the add-on collection against the orientation of the items in the initial collection.

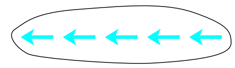
- When the orientation of the items in the add-on collection is *the same as* the orientation of the items in the initial collection, the attachment process is exactly the same as with collections of plain items.

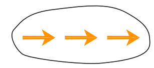
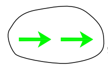
EXAMPLE 8.1. To perform the *action* of attaching  to the *initial collection* , that is:

we attach the add-on collection to the initial collection:

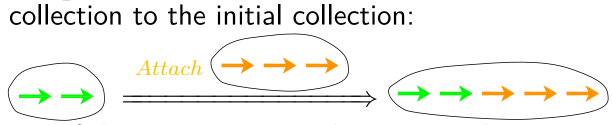


and, since the orientation of the initial items is *the same as* the orientation of the attached items, there is no cancellation and the *final collection* is

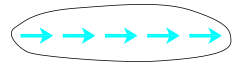
.

EXAMPLE 8.2. To perform the *action* of attaching  to the *initial collection* , that is:

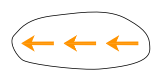
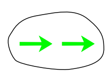
we attach the add-on collection to the initial collection:

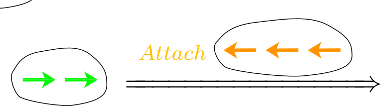


and, since the orientation of the initial items is *the same as* the orientation of the attached items, there is no cancellation and the *final collection* is

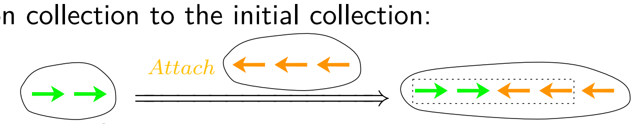
.

- When the orientation of the items in the add-on collection is *the opposite of* the orientation of the items in the initial collection, there is going to be *cancellation* of the items of *opposite* orientation.

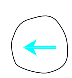
EXAMPLE 8.3. To perform the *action* of attaching  to the *initial collection* , that is:



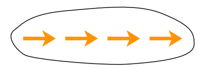
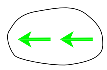
we attach the add-on collection to the initial collection:

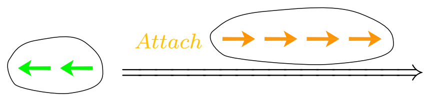


but, since the orientation of the initial items collection is *the opposite of* the orientation of the attached items, there are two cancellations and the *final collection* is

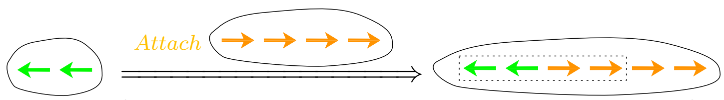
.

function
input-output rule
unspecified input
specific inputs
function name
output specifying code
specific outputs

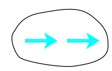
Example 8.4. To perform the *action* of attaching  to the *initial collection* , that is:



we attach the add-on collection to the initial collection:



but, since the orientation of the initial items collection is *the opposite of* the orientation of the attached items, there are two cancellations and the *final collection* is

.

2 Adding Signed Number Phrases

1. We saw in Chapter 5 that real world *agents of change* are represented on paper by **functions** which we specify with an **input-output rule** that consists of:

i. An **unspecified input** eventually to be replaced by **specific inputs**, that is the number phrases that represent the *initial collections*.

ii. A **function name**, that is the name of the function that represents the agent of change

iii. The **output specifying code** which is the code that specifies the *output* of the function in terms of the input. The **specific outputs** are the number phrases that represent the final collections.

Thus, the real world *action*

Initial collection $\xrightarrow{\textit{Agent of Change}}$ Final collection

is represented on paper by the *input-output rule*

Unspecified input $\xrightarrow{\text{Name of function}}$ Output-specifying code

To be replaced by specific inputs — Specifies what to do with the inputs to get the outputs

2. It will be most important to know what we are talking about:

plain adding
⊕
oplussing
signed adding function
signed addition

LANGUAGE 8.1 (Plain versus Signed Adding) To make it clear which *adding* we are talking about, we will refer to the adding of plain numerators which were introduced in Chapter 5 as **plain adding**.
In fact, to keep things clear, we will use for signed addition the symbol $\oplus$, read "oplus", and, in the future, instead of saying that we are "*adding* a signed numerator", we will say that we are "**oplussing** that numerator".

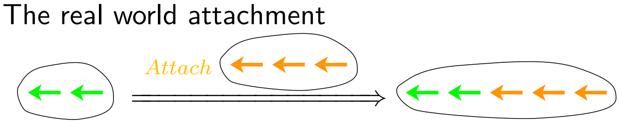
3. Just as with plain addition, we represent the action of attaching an add-on collection of *oriented* items to a collection of *oriented* items by the input-output rule of a **signed adding function**, often called **signed addition** for short.

Input signed numerator $\xrightarrow{\oplus \text{ signed numerator}}$ Output numerator =
Input signed numerator ⊕ *signed* numerator

4. Just like, in the real world , we must check if the orientation of the items in the add-on collection is the same as or the opposite of the orientation of the items in the initial collection, in the paper world we must check if the sign of the add-on numerator is the same as or the opposite of the sign of the input numerator.

- When the sign of the add-on numerator is the same as the sign of the input numerator, then:
  - The *sign* of the output numerator is the sign common to the input numerator and the add-on numerator
  - The *size* of the output numerator is the plain addition of the size of the add-on numerator to the size of the input numerator

EXAMPLE 8.5. The real world attachment

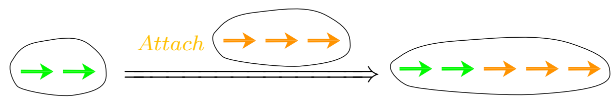


is represented in the paper world by

$$
\begin{aligned}
-2\,\textsf{Arrows} \xrightarrow{\oplus -3\,\textsf{Arrows}} & -2\,\textsf{Arrows} \oplus -3\,\textsf{Arrows} \\
&= -(2+3)\,\textsf{Arrows} \\
&= -5\,\textsf{Arrows}
\end{aligned}
$$

where $2+3=5$ is the *plain* addition of the sizes.

Example 8.6. The real world attachment



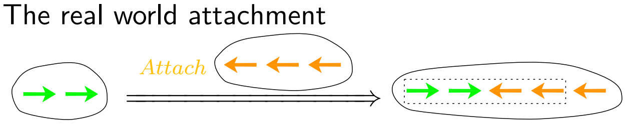
is represented in the paper world by

$$\begin{aligned} +2\,\textsf{Arrows} \xrightarrow{\oplus +3\,\textsf{Arrows}} &+2\,\textsf{Arrows} \oplus +3\,\textsf{Arrows} \\ &= +(2+3)\,\textsf{Arrows} \\ &= +5\,\textsf{Arrows} \end{aligned}$$

where $2 + 3 = 5$ is the *plain* addition of the sizes.

- When the sign of the add-on numerator is the *opposite of* the sign of the input numerator, we must *compare* the *size* of the add-on numerator to the *size* of the input numerator in order to know which way to *plain subtract* the sizes:
  - The *sign* of the output numerator will be the sign of the numerator with the *larger size*,
  - The *size* of the output numerator will be the result of *plain subtracting* the numerator with *smaller size* from the numerator with *larger size*.

Example 8.7. The real world attachment

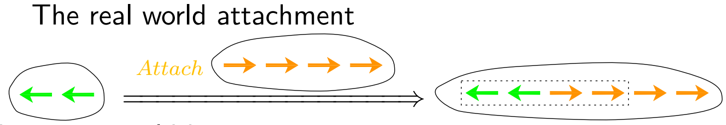


is represented in the paper world by

$$\begin{aligned} +2\,\textsf{Arrows} \xrightarrow{\oplus -3\,\textsf{Arrows}} &+2\,\textsf{Arrows} \oplus -3\,\textsf{Arrows} \\ &= -(3-2)\,\textsf{Arrows} \\ &= -1\,\textsf{Arrows} \end{aligned}$$

where $3 - 2 = 1$ is the *plain* subtraction of the smaller size from the larger size..

Example 8.8. The real world attachment



is represented in the paper world by

prior knowledge

$$-2\,\text{Arrows} \xrightarrow{\oplus +4\,\text{Arrows}} -2\,\text{Arrows} \oplus +4\,\text{Arrows}$$
$$= +(4-2)\,\text{Arrows}$$
$$= +2\,\text{Arrows}$$

where $4-2=1$ is the *plain* subtraction of the smaller size from the larger size..

5. Here are a few examples that focus on the *numerators*.

EXAMPLE 8.9. $-3 \oplus -5 = -(3+5) = -8$
where, since the two numerators have the *same* signs:
- the *sign* of the result, $-$, is the common sign,
- the *size* of the result, 8, is the the *plain addition*, $3+5$, of the sizes

EXAMPLE 8.10. $+3 \oplus +5 = +(3+5) = +8$
where, since the two numerators have the *same* signs:
- the *sign* of the result, $+$, is the common sign,
- the *size* of the result, 8, is the the *plain addition*, $3+5$, of the sizes

EXAMPLE 8.11. $-3 \oplus +5 = +(5-3) = +2$
where, since the two numerators have *opposite* signs:
- the *sign* of the result, $+$, is the sign of the numerator with the *larger* size,
- the *size* of the result, 2, is the *plain* subtraction, $5-3$, of the smaller size from the larger size.

EXAMPLE 8.12. $+3 \oplus -5 = -(5-3) = -2$
where, since the two numerators have *opposite* signs:
- the *sign* of the result, $-$, is the sign of the numerator with the *larger* size,
- the *size* of the result, 2, is the *plain* subtraction, $5-3$, of the smaller size from the larger size.

3 Architecture

Dealing with *signed* numerators was based on a **prior knowledge** of *plain* numerators. (To help us focus, we will deal here with *counting* numerators

tool
forked

but things are exactly the same with *decimal* numerators.)

1. In order to represent collections of *plain items*, we used:

i. *Plain numerators*, $1, 2, 3, \ldots$

ii. In order to *compare* a first plain numerator to a second plain numerator, the procedure was to count from the first plain numerator to the second plain numerator and if we had to *count up* then the verb was $<$ and if we had to *count down* the verb was $>$.

iii. In order to *add* to a first plain numerator a second plain numerator, the symbol was $+$ and the procedure was, starting from the first plain numerator, to count *up* on our fingers the second plain numerator. Then the plain numerator we arrive at was the result.

iv. In order to *subtract* from a first plain numerator a second plain numerator, the symbol was $-$ and the procedure was, starting from the first plain numerator, to count *down* on our fingers the second plain numerator. Then, *if we could do it*, the plain numerator we arrived at was the result.

So, our **tool** was *counting*.

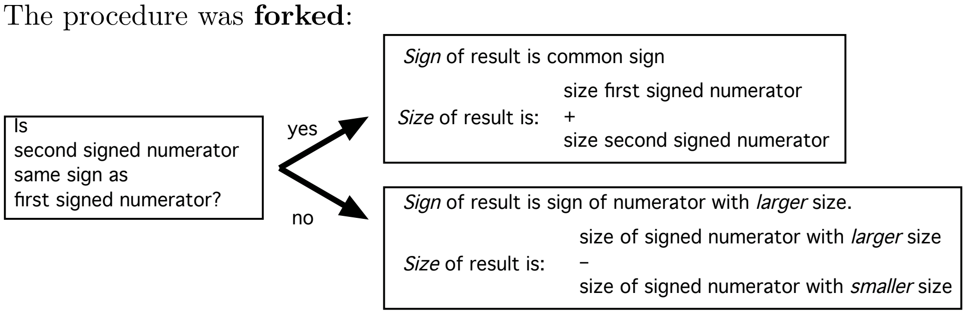
2. In order to represent collection of *oriented items*, we used:

i. *Signed numerators* whose *sign* was one of two symbols, $+$ or $-$ (unfortunately already used above) and whose *size* was a plain numerator.

ii. In order to *compare* a first signed numerator to a second signed numerator, the procedure was still to count from the first signed numerator to the second signed numerator and if we had to *count up* then the verb was $\textcircled{<}$ and if we had to *count down* the verb was $\textcircled{>}$. Note, though, that this was *signed counting* since we also had to be able to count on *negative* numerators.

iii. In order to *oplus* to a first signed numerator a second signed numerator, the procedure was quite a bit more complicated because:

- The procedure was **forked**:



but also because

- We had to compare, add, subtract *plain numerators*, namely the *sizes* of the signed numerators.

Altogether then, *oplussing signed numerator* is *based* on prior knowledge of *plain numerators*.

take-from

Chapter 9

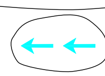
Signed Subtracting

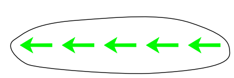
1 Detaching Collections of Oriented Items

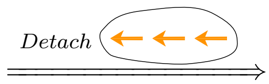
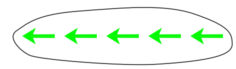
1. Detaching a **take-from** collection of *oriented* items collection of oriented items from an initial collection of *oriented* items is quite different from detaching a take-from collection of *plain* items from an initial collection of *plain* items.

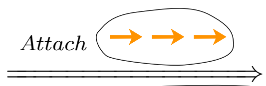
- When the orientation of the items in the take-from collection is *the same as* the orientation of the items in the initial collection, up-front, the detachment process still looks pretty much the same as with collections of plain items.
  - If the take-from collection is *smaller in size* than the initial collection, things work exactly as with collection of plain items.

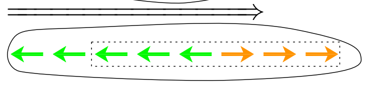
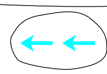
EXAMPLE 9.1. 

and the *final collection* is .

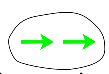
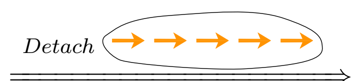
Observe, though, that instead of 

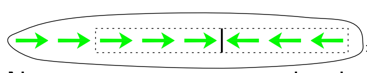
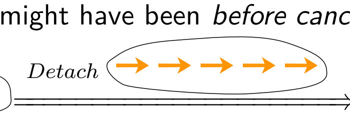
 we could have done 



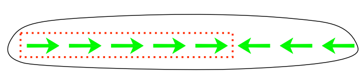
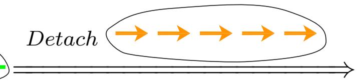
 with, after cancellations, the same final collection .

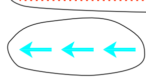
– If the take-from collection is *larger in size* than the initial collection, up front, we cannot carry out the detachment and we must look at the initial collection as it might have been *before cancellations*. We can then detach the items in the take-from collection from the items in the initial collection with the same orientation.

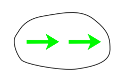
EXAMPLE 9.2.   ? We look at the initial collection as it might have been *before cancellation*:

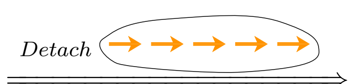
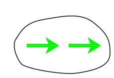
  ?

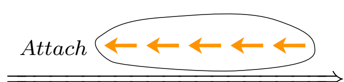
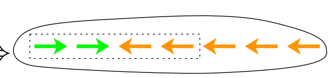
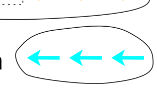
Now we can carry out the detachment and get the final collection:



Observe, though, that instead of 

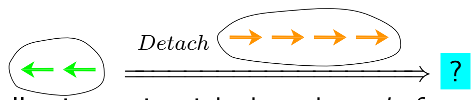
 we could have done 

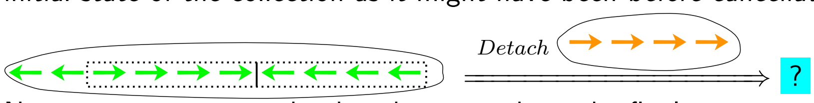
  with, after cancellations, the same final collection .

- When the orientation of the items in the take-from collection is *the opposite of* the orientation of the items in the initial collection, things are totally unlike the plain case and we must look at the initial collection as

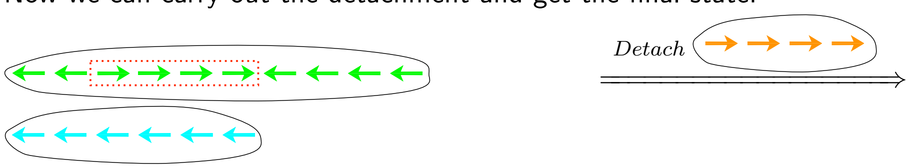
function
input-output rule
unspecified input
specific inputs
function name
output specifying code
specific outputs

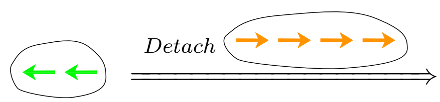
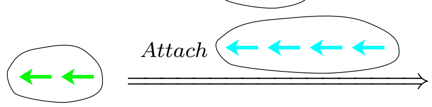
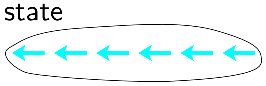
it might have been *before cancellations*

EXAMPLE 9.3.  We look at the initial state of the collection as it might have been *before cancellations*.



Now we can carry out the detachment and get the final state:



Observe, though, that instead of  we could have done  with the same final state  .

2. Indeed, as we saw in the above **EXAMPLES**, in the real world, instead of detaching a take-from collection of oriented items, we can always attach an add-on collection of items with the opposite orientation because the oriented items in the add-on collection will cancel the oriented items in the initial collection.

2 Subtracting Signed Number Phrases

1. We saw in Chapter 5 that real world *agents of change* are represented on paper by **functions** which we specify with an **input-output rule** that consists of:
i. An **unspecified input** eventually to be replaced by **specific inputs**, that is the number phrases that represent the *initial collections*.
ii. A **function name**, that is the name of the function that represents the agent of change
iii. The **output specifying code** which is the code that specifies the *output* of the function in terms of the input. The **specific outputs** are the number phrases that represent the final collections.

plain subtracting
⊖
ominussing
signed subtracting function
signed subtraction

Thus, the real world *action*

Initial collection $\xrightarrow{\textit{Agent of Change}}$ Final collection

is represented on paper by the *input-output rule*

Unspecified input (To be replaced by specific inputs) $\xrightarrow{\text{Name of function}}$ Output-specifying code (Specifies what to do with the inputs to get the outputs)

2. It will be most important to know what we are talking about:

LANGUAGE 9.1 (Plain versus Signed Subtracting) To make it clear which *subtracting* we are talking about, we will refer to the subtracting of plain numerators which were introduced in Chapter 5 as **plain subtracting**.
In fact, to keep things clear, we will use for signed subtracting the symbol ⊖, read "ominus", and, in the future, instead of saying that we are "*subtracting* a signed numerator", we will say that we are "**ominussing** that numerator".

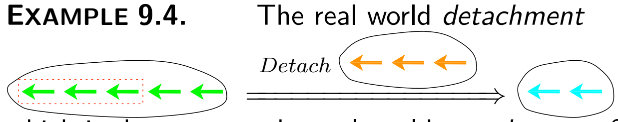
3. Just as with plain subtraction, we represent the action of detaching a take-from collection of *oriented* items from a collection of *oriented* items by the input-output rule of a **signed subtracting function**, often called **signed subtraction** for short.

Input signed numerator $\xrightarrow{\ominus \text{ signed numerator}}$ Output numerator = Input signed numerator ⊖ *signed* numerator

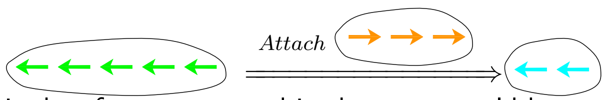
4. But, while the *addition* of two signed number phrases was a bit complicated, the *subtraction* of two signed number phrases is quite simple:

THEOREM 9.1 Ominus To ominus a number phrase, oplus the opposite of the number phrase

EXAMPLE 9.4. The real world *detachment*



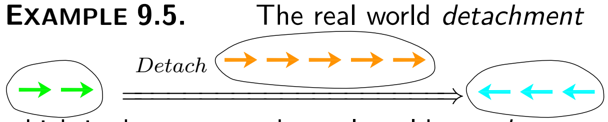
which is the same as the real world *attachment of the opposite*



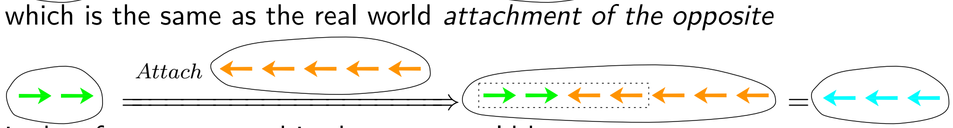
is therefore represented in the paper world by

$$
\begin{aligned}
-5 \text{ Arrows} \xrightarrow{\ominus -3 \text{ Arrows}} & -5 \text{ Arrows} \ominus -3 \text{ Arrows} \\
&= -5 \text{ Arrows} \oplus +3 \text{ Arrows} \\
&= -(5-3) \text{ Arrows} \\
&= -2 \text{ Arrows}
\end{aligned}
$$

Example 9.5. The real world *detachment*



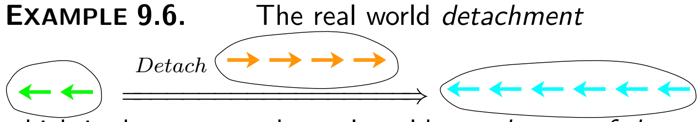
which is the same as the real world *attachment of the opposite*



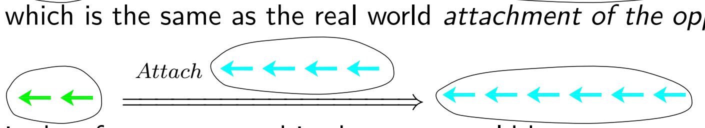
is therefore represented in the paper world by

$$
\begin{aligned}
+2 \text{ Arrows} \xrightarrow{\ominus +5 \text{ Arrows}} & +2 \text{ Arrows} \ominus +5 \text{ Arrows} \\
&= +2 \text{ Arrows} \oplus -5 \text{ Arrows} \\
&= -(5-2) \text{ Arrows} \\
&= -3 \text{ Arrows}
\end{aligned}
$$

Example 9.6. The real world *detachment*



which is the same as the real world *attachment of the opposite*



is therefore represented in the paper world by

$$
\begin{aligned}
-2\,\textsf{Arrows} \xrightarrow{\ominus +4\,\textsf{Arrows}} & -2\,\textsf{Arrows} \ominus +4\ \textsf{Arrows} \\
& = -2\,\textsf{Arrows} \oplus -4\ \textsf{Arrows} \\
& = -(2+4)\,\textsf{Arrows} \\
& = -6\,\textsf{Arrows}
\end{aligned}
$$

Here are a few examples that focus on the numerators:

EXAMPLE 9.7. In order to get $(+3) \ominus (+5)$,

i. We compute $(+3) \oplus$ Opposite $(+5)$ that is we compute $(+3) \oplus (-5)$

ii. Which, using the Chapter on signed addition, gives -2 .

EXAMPLE 9.8. In order to get $(+3) \ominus (-5)$,

i. We compute $(+3) \oplus$ Opposite (-5) that is we compute $(+3) \oplus (+5)$

ii. Which, using the Chapter on signed addition, gives $+8$.

EXAMPLE 9.9. In order to get $(-3) \ominus (+5)$,

i. We compute $(-3) \oplus$ Opposite $(+5)$ that is we compute $(-3) \oplus (-5)$

ii. Which, using the Chapter on signed addition, gives -8.

EXAMPLE 9.10. In order to get $(-3) \ominus (-5)$,

i. We compute $(-3) \oplus$ Opposite (-5) that is we compute $(-3) \oplus (+5)$

ii. Which, using the Chapter on signed addition, gives $+2$.

EXAMPLE 9.11.

EXAMPLE 9.12.

3 Bank Accounts

A very important application of signed numerators is that they allow us to monitor real world bank accounts.

state
balance
in the black
in the red
action
initial state
final state
deposit
withdrawal
change
gain

1. The standard terminology for monitoring bank accounts is as follows:
a. The real world **state** of a bank account at any time is represented on paper by a *signed* numerator called the **balance** and:

- If the balance is *positive*, we say that the real world bank account is **in the black** (Ahead of the game.).
- If the numerator which represents the balance is *negative*, we say that the real world bank account is **in the red** (In the hole).

b. A real world **action** on a bank account changes the bank account from an **initial state** to a **final state**.

$$\text{Initial State} \xRightarrow{\textit{Action}} \text{Final State}$$

An action on a bank account can be either a **deposit** represented on paper by a positive numerator or a **withdrawal** represented on paper by a negative numerator.
c. The **change** from an initial state to a final state is:
i. the final state,
from which we detach
ii. the initial state.
The reason is that each state is the result of *all prior* actions from the very beginning. So, by subtracting the *initial* state from the *final state*, we eliminate the effect of all the actions that resulted in the *initial* state leaving only the last one, namely the effect of the *last action*.
So, the change between an initial state and a final state is represented on paper by the *signed difference*

$$\text{final balance} \ominus \text{initial balance}$$

2. The *change* from an initial state to a final state can be:

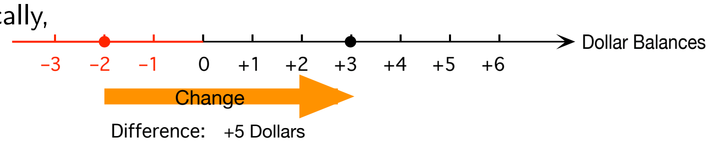
- *up* in which case the change is called a **gain** and, on paper, the *difference* which represents a gain is a *positive* numerator.

EXAMPLE 9.13.
– On Monday, Jill's balance was TWO DOLLARS in the *red*
– On Wednesday, Jill's balance was THREE DOLLARS in the *black*
So, from Monday to Wednesday, Jill's balance went *up* by FIVE DOLLARS and, correspondingly, on paper the *difference* which represent the *change* is

$$\begin{aligned}\text{Difference} &= \textbf{Final Balance} \ominus \textbf{Initial Balance}\\ &= +3\ \textbf{Dollars} \ominus -2\ \textbf{Dollars}\\ &= +3\ \textbf{Dollars} \oplus +2\ \textbf{Dollars}\\ &= +5\ \textbf{Dollars}\end{aligned}$$

loss

Graphically,



or can be

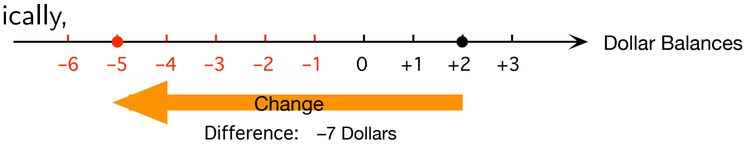
- *down* in which case the change is called a **loss** and, on paper, the *difference* which represents a loss is a *negative* numerator,

EXAMPLE 9.14.

– On Monday, Jack's balance was TWO DOLLARS in the *black*
– On Wednesday, Jack's balance was FIVE DOLLARS in the *red*
So, from Monday to Wednesday, Jack's balance went *down* by SEVEN DOLLARS and, correspondingly, on paper the *difference* which represent the *change* is

$$\begin{aligned}\text{Difference} &= \textbf{Final Balance} \ominus \textbf{Initial Balance}\\ &= -5 \textbf{ Dollars} \ominus +2 \textbf{ Dollars}\\ &= -5 \textbf{ Dollars} \oplus -2 \textbf{ Dollars}\\ &= -7 \textbf{ Dollars}\end{aligned}$$

Graphically,



EXAMPLE 9.15.

Chapter 10

Multiplication - Division of Signed Numbers

Just like multiplication and division of plain numbers were very different from addition and subtraction of plain numbers, multiplication and division of signed numerators are very different from addition and subtraction of signed numbers.

i. Multiplication of signed numbers cannot be repeated *oplussing*.

Example 10.1. What repeated *oplussing* could -3 **Arrows** multiplied by $+5$ **Arrows** possibly stand for?

ii. Multiplication of signed numbers cannot give an area.

Example 10.2. What area could -3 **Feet** multiplied by $+5$ **Feet** possibly be? The denominator would of course have to be **SquareFeet** but what if the result of the multiplication turned out *negative*? What could a *negative* area be?

1 Signed Co-multiplication

This is where multiplication of signed numerators, for which we will use the symbol $\otimes$, is very useful.

signed co-multiplication

1. We begin by looking at the real-world. As before, we want to investigate the *change* in a given state, *gain* or *loss*, that results from a given transaction, "in" or "out" as before but with *oriented collections* of "good" items or "bad" items.

EXAMPLE 10.3. Consider a store where collections of APPLES can either GET *into* THE STORE or GET *out of* THE STORE. Moreover, the collections are collections of *oriented* items in that the apples can be either GOOD APPLES—inasmuch as they will generate a *profit* when they are sold—or BAD APPLES—inasmuch as they will generate a *loss* because they will have to be disposed of at a cost.

2. We now look at the way we will represent things on paper.

a. We will represent

- Collections of items getting "in" by *positive* number-phrases,
- Collections of items getting "out" by *negative* number-phrases,

EXAMPLE 10.4. In EXAMPLE 10.3 , we would represent

- Collections of APPLES getting *into* the store by *positive* number-phrases,
- Collections of APPLES getting *out of* the store by *negative* number-phrases,

b. We will represent

- The unit-worth of "good" items by a *positive* number-phrase,
- The unit-worth of "bad" items by *negative* number-phrases,

EXAMPLE 10.5. In EXAMPLE 10.3 , we would represent

- The UNIT-VALUE of GOOD APPLES that is APPLES that will generate a sales *profit* of seven cents per apples by the co-number-phrase $+7\ \frac{\text{Cents}}{\text{Apple}}$
- The UNIT-VALUE of BAD APPLES that is APPLES that will generate a disposal *cost* of seven cents per apple by the co-number-phrase $-7\ \frac{\text{Cents}}{\text{Apple}}$

3. We can now write the procedure for **signed co-multiplication** for which we will use the symbol $\otimes$:

i. multiply the *denominators* (with cancellation).

ii. *Otime* the *numerators* according to the way the result is a "good" change or a "bad" change:

- A collection of "good" items getting "in" makes for a "good" change so $+ \otimes + = +$.

EXAMPLE 10.6.

Three apples get *in* the store. $+3$ **Apples**

The apples have a unit-value of seven cents-per-apple *gain*. $+7\ \frac{\textsf{Cents}}{\textsf{Apple}}$

The specifying phrase is $[+3\ \textsf{Apples}] \otimes \left[+7\ \frac{\textsf{Cents}}{\textsf{Apple}}\right]$

We co-multiply $[(+3) \otimes (+7)]\ \left[\cancel{\textsf{Apples}} \times \frac{\textsf{Cents}}{\cancel{\textsf{Apple}}}\right]$

We get a twenty-one cent *gain*. $= +21$ **Cents**

- A collection of "good" items getting "out" makes for a "bad" change so $+ \otimes - = -$.

EXAMPLE 10.7.

Three apples get *in* the store. $+3$ **Apples**

The apples have a unit-value of seven cents-per-apple *loss*. $-7\ \frac{\textsf{Cents}}{\textsf{Apple}}$

The specifying phrase is $[+3\ \textsf{Apples}] \otimes \left[-7\ \frac{\textsf{Cents}}{\textsf{Apple}}\right]$

We co-multiply $[(+3) \otimes (-7)]\ \left[\cancel{\textsf{Apples}} \times \frac{\textsf{Cents}}{\cancel{\textsf{Apple}}}\right]$

We get a twenty-one cent *loss*. $= -21$ **Cents**

- A collection of "bad" items getting "in" makes for a "bad" change so $- \otimes + = -$.

EXAMPLE 10.8.

Three apples get *out* of the store. -3 **Apples**

The apples have a unit-value of seven cents-per-apple *gain*. $+7\ \frac{\textsf{Cents}}{\textsf{Apple}}$

The specifying phrase is $[-3\ \textsf{Apples}] \otimes \left[+7\ \frac{\textsf{Cents}}{\textsf{Apple}}\right]$

We co-multiply $[(-3) \otimes (+7)]\ \left[\cancel{\textsf{Apples}} \times \frac{\textsf{Cents}}{\cancel{\textsf{Apple}}}\right]$

We get a twenty-one cent *loss*. $= -21$ **Cents**

- A collection of "bad" items getting "out" makes for a "good" change so $- \otimes - = +$.

EXAMPLE 10.9.

Three apples get *out* of the store.	-3 **Apples**
The apples have a unit-value of seven cents-per-apple *loss*.	$-7\ \frac{\textbf{Cents}}{\textbf{Apple}}$
The specifying phrase is	$[-3\ \textbf{Apples}] \otimes \left[-7\ \frac{\textbf{Cents}}{\textbf{Apple}}\right]$
We co-multiply	$[(-3) \otimes (-7)]\ \left[\cancel{\textbf{Apples}} \times \frac{\textbf{Cents}}{\cancel{\textbf{Apple}}}\right]$
We get a twenty-one cent *gain*.	$= +21$ **Cents**

In other words, the rule for the *multiplication of signs* is:

THEOREM 10.1 Multiplication of Signs

	+	−
+	+	−
−	−	+

2 Signed Division

The rule of signs for the *division of signs* is the same as the rule for the *multiplication of signs*:

THEOREM 10.2 Division of Signs

	+	−
+	+	−
−	−	+

EXAMPLE 10.10. If we know that the store has incurred a TWENTY-ONE CENTS loss with THREE APPLES moving out, what was the unit-worth of the APPLES?
Dividing -21 **Cents** by -3 **Apples** gives us $+7\frac{\textbf{Cents}}{\textbf{Apple}}$
(Indeed, if APPLES moving *out* resulted in a *loss*, then the APPLES must have been *good*!)

EXAMPLE 10.11. If we know that the store has incurred a TWENTY-ONE CENTS loss with GOOD APPLES worth SEVEN CENTS each, how many APPLES moved and which way did they move?

Dividing -21 **Cents** by $+7\frac{\textbf{Cents}}{\textbf{Apple}}$ gives us -3 **Apples**
(Indeed, if GOOD APPLES moving resulted in a *loss*, then the APPLES must have moved *out*!)

EXAMPLE 10.12. If we know that the store has incurred a TWENTY-ONE CENTS gain with THREE APPLES moving out, what is the unit-worth of the APPLES?
Dividing $+21$ **Cents** by -3 **Apples** gives us $-7\frac{\textbf{Cents}}{\textbf{Apple}}$
(Indeed, if APPLES moving *out* resulted in a *gain*, then the APPLES must have been *bad*!)

EXAMPLE 10.13. If we know that the store has incurred a TWENTY-ONE CENTS gain with the unit-worth of the BAD APPLES worth SEVEN CENTS each, how many APPLES moved and which way did they move?
Dividing $+21$ **Cents** by $-7\frac{\textbf{Cents}}{\textbf{Apple}}$ gives us -3 **Apples**
(Indeed, if BAD APPLES moving resulted in a *gain*, then the APPLES must have moved *out*!)

Chapter 11

Coding Multiple Operations

(See http://www.devmathrevival.net/?p=2628#comment-204871)

Up until now, we followed the standard practice in the *sciences* and *technologies* in which numerators always come *with* denominators so that there never could be any doubt as to what was to be done. In this chapter, we will look at the difficulties created by the practice standard in *mathematics*, namely the use of numerators *without* denominators.

1 Plain numbers $\mathbb{N}$

1. With *plain* numbers, each symbol can only stand for *one* action.

a. With *plain* numbers, the symbol + can only stand for *add*.

Example 11.1. In $2+3$, the symbol $+$ stands for *add*.
So we count *up* $2 \xrightarrow{3,4,5} 5$, and the result is 5.

Example 11.2. In $72.18 + 31.04$, $+$ stands for *add*.
So we set up

$$\begin{array}{r} \scriptstyle 1 \\ 7\,2.1\,8 \\ +\quad 3\,1.0\,4 \\ \hline 1\,0\,3.2\,2 \end{array}$$

and the result is 103.22.

b. With *plain* numbers, the symbol $-$ can only stand for *subtract*.

EXAMPLE 11.3. In $5-3$, the symbol $-$ stands for *subtract*.
So we count *down* $5 \xrightarrow{4,3,2} 2$, and the result is 2.

EXAMPLE 11.4. In $3-5$, the symbol $-$ stands for *subtract*.
So we count *down* $3 \xrightarrow{2,1,0,?} ?$ and the subtraction cannot be done.

EXAMPLE 11.5. In $31.05-17.22$, the symbol $-$ stands for *subtract*.

We set up
$$\begin{array}{r} 3\ 1.0\ 5 \\ -\ 1\ 7.2\ 2 \\ \hline 1\ 3.8\ 3 \end{array}$$
and the result is 13.83.

EXAMPLE 11.6. In $17.22-31.05$, the symbol $-$ stands for *subtract*.
But we cannot do $17.22-31.05$.

c. With *plain* numbers, the symbol $\times$ can only stand for *multiply*.

EXAMPLE 11.7. In both 5×3 and 3×5, the symbol $\times$ stands for *multiply*.
Then the multiplication tables give us both $5\times3=15$ and $3\times5=15$

EXAMPLE 11.8. In 53.04×30.27, the symbol $\times$ stands for *multiply*.

We set up
$$\begin{array}{r} 5\ 3.0\ 4 \\ \times\ 3\ 0.2\ 7 \\ \hline 3\ 7\ 1\ 2\ 8 \\ 1\ 0\ 6\ 0\ 8\ \ \\ 1\ 5\ 9\ 1\ 2\ \cdot\ \ \ \ \\ \hline 1\ 6\ 0\ 5.5\ 2\ 0\ 8 \end{array}$$
and the result is $1\,605.5208$

d. With *plain* numbers, the symbol $\div$ can only stand for *divide*.

EXAMPLE 11.9. In both $15\div3$ and $17\div5$, the symbol $\times$ stands for *divide*.
Then the multiplication tables give us both $15\div3=5$ and $17\div5=3$ with a remainder of 2

EXAMPLE 11.10. In $523.14\div32.07$, the symbol $\div$ stands for *divide*.

We set up
$$\begin{array}{r} 3 \\ 32.1\ \big)\ \overline{98.7} \\ 96.3 \\ \hline 2.4 \end{array}$$
and the result is $3+[...]$

2. Generally, we *operate* the same way we read and write, that is from left to right. However this may not be the case when using *more than one* operation and we then need to think again about *denominators*.

EXAMPLE 11.11. $7+4+2$ can only come from something like 7 **Apples** $+$ 4 **Apples** $+$ 2 **Apples** and so we do:

$$\begin{aligned} 7+4+2 \\ 11+2 \\ 13 \end{aligned}$$

EXAMPLE 11.12. $7-4-2$ can only come from something like 7 **Apples** $-$ 4 **Apples** $-$ 2 **Apples** and so we do, *if we can*:

$$\begin{aligned} 7-4-2 \\ 3-2 \\ 1 \end{aligned}$$

EXAMPLE 11.13. $7-5-3$ can only come from something like 7 **Apples** $-$ 5 **Apples** $-$ 3 **Apples** which we cannot do because 7 **Apples** $-$ 5 **Apples** $=$ 2 **Apples** and we cannot do 2 **Apples** $-$ 3 **Apples**

EXAMPLE 11.14. $2+3\times 5$ can only come from either:

- Something like 2 **SqFeet** $+$ 3 **Feet** $\times$ 5 **Feet**

or from

- Something like 2 **Dollars** $+$ 3 **Apples** $\times$ $5\,\frac{\textbf{Dollars}}{\textbf{Apple}}$

Either way, if we try to do the addition first we are looking at

- 2 **SqFeet** $+$ 3 **Feet**

or

- 2 **Dollars** $+$ 3 **Apples**

which make no sense.

If we try to do the *multiplication first* we are looking at:

- 3 **Feet** $\times$ 5 **Feet** $=$ 15 **SqFeet**

and

- 3 **Apples** $\times$ $5\,\frac{\textbf{Dollars}}{\textbf{Apple}}$ $=$ 15 **Dollars**

which make perfect sense. So, altogether, we do:

parentheses

- $2\,\textsf{SqFeet} + 3\,\textsf{Feet} \times 5\,\textsf{Feet} = 2\,\textsf{SqFeet} + 15\,\textsf{SqFeet} = 17\,\textsf{SqFeet}$
- $2\,\textsf{Dollars} + 3\,\textsf{Apples} \times 5\frac{\textsf{Dollars}}{\textsf{Apple}} = 2\,\textsf{Dollars} + 15\,\textsf{Dollars} = 17\,\textsf{Dollars}$

So, even without denominators, we *must* do the *multiplication first*:

$$2 + 3 \times 5 = 2 + 15 = 17$$

3. Same with *division*: we must do division ahead of addition and subtraction.

4. If, for whatever reason, somebody wants an opearation to be done first, this person *must* say so by using **parentheses**.

EXAMPLE 11.15. In $(2 + 3) \times 5$, the *parentheses* say we *must* do the addition first:

$$(2 + 3) \times 7 = 5 \times 7 = 35$$

2 Signed numbers $\mathbb{Z}$

With *signed* numbers, things get complicated because each symbol can now have *several* meanings and because even when dealing with *signed* numbers, we still need to use *plain* operations for the *sizes* of these signed numbers.

- The symbol $+$ can now mean:
  - *positive*
  - *signed addition* (for which, until now, we used the symbol $\oplus$)
  - *plain addition* (Still used for the *sizes* of signed numbers.)
- The symbol $-$ can now mean:
  - *negative*
  - *signed subtraction* (for which, until now, we used the symbol $\ominus$)
  - *plain subtraction* (Still used for the *sizes* of signed numbers.)
- The symbol $\times$ can now mean:
  - *signed multiplication* (for which, until now, we used the symbol $\otimes$)
  - *plain multiplication* (Still used for the *sizes* of signed numbers.)

The symbol $\div$ can now mean:
  - *signed division* (for which, until now, we used the symbol ⨸)
  - *plain division* (Still used for the *sizes* of signed numbers.).

It was to make things easier that until now we used the symbols $\oplus$, $\ominus$. $\otimes$, and ⨸ for the *signed* operations. We will now see how to use just $+$, $-$, $\times$, and $\div$.

1. The first difficulty is that writers often let the *positive symbol go without saying*.

EXAMPLE 11.16. $7 - (-3)$ stands for $+7 \ominus -3$ and therefore we do $+7 \oplus +3 = +10$

2. The second difficulty comes from the fact that writers often let *both* the *positive symbol* and the *signed addition symbol* go without saying.

EXAMPLE 11.17. $3 - 7$
i. If we are sure that the writer intended *plain numbers*, then $3 - 7$ can't be done
ii. If we are sure that the writer intended *signed numbers* then $3 - 7$ can stand for either:

- $+3 \oplus -7$ if the writer intended the $\oplus$ to go without saying and therefore the $-$ to be the sign going with the 7

or

- $+3 \ominus +7$ if the writer intended the $-$ to mean $\ominus$ and the sign of 7 to go without saying therefore to be $+$.

But, since $\ominus$ is done as $\oplus$ *opposite*, both compute to -4 and which the writer intended does not matter.

EXAMPLE 11.18. With $-4 \cdot 6 - 7$ we know the writer intended *signed numbers* because of the -4.
But then the writer intended $-4 \otimes +6 \oplus -7$.
So, we do the *multiplication first*:

$$-4 \otimes +6 \oplus -7 =$$

EXAMPLE 11.19.

EXAMPLE 11.20.

EXAMPLE 11.21.

EXAMPLE 11.22.

Example 11.23.

Example 11.24.

Example 11.25.

Example 11.26.

Example 11.27.

Chapter 12

Fractions – Comparison

There are two main aspects to fractions

- Fractions can be a historical remnant from before the invention of the decimal-metric system
- Fractions can be *code* for division.

but unfortunately fractions are also confused very often with

- scores.

1 Fractions as Historical Leftovers

1. Imagine someone who knows only

- *counting numerators* namely $1, 2, 3, \ldots$
- and that the *denominator* **Dollar** represents a real world 

Suppose now that this person were to ask what  is.
The usual responses are, more or less:

- "A **Quarter**". This, though, just says what the *denominator* for this item is. It does not say *what* the real world item itself *is*.

- "25 **Cents**". Since the person knows how to count, s/he understands the *numerator* 25 but not the *denominator* **Cents**. So, this does not say *what* the real world item itself *is*.
- "$\frac{1}{4}$ **Dollar**". Since the person knows what a **Dollar** is s/he understands the *denominator* but since s/he only knows how to *count* s/he doesn't understand what $\frac{1}{4}$ means. So, this does not say *what* the real world item itself *is*.

The only things that will explain to this person what a quarter is is to say that four such items can be exchanged for a dollar:

In other words, we can only write:

$$4 \textbf{ Quarters} = 1 \textbf{ Dollar}$$

2. The difficulty though is that this does not allow us to use "quarter" as a *denominator*.

EXAMPLE 12.1. After we have explained what a LINCOLN is in terms of what a WASHINGTON is by writing:

$$1 \textbf{ Lincoln} = 5 \textbf{ Washingtons}$$

we can explain what a number of LINCOLNS is by writing:

$$2 \textbf{ Lincoln} = 10 \textbf{ Washingtons}$$
$$3 \textbf{ Lincoln} = 15 \textbf{ Washingtons}$$
$$4 \textbf{ Lincoln} = 20 \textbf{ Washingtons}$$
$$5 \textbf{ Lincoln} = 25 \textbf{ Washingtons}$$

etc

EXAMPLE 12.2. But after we have explained what a QUARTER is in terms of what a WASHINGTON is by writing:

$$4 \textbf{ Quarters} = 1 \textbf{ Dollar}$$

we still cannot explain what most numbers of QUARTERS are:

2 **Quarters** =?

3 **Quarters** =?

4 **Quarters** = 1 **Dollar**

5 **Quarters** =?

etc

because the 1 in what we wrote to explain what a QUARTER is in terms of what a WASHINGTON is on the wrong side.

3. The only way out is to use a standard *linguistic trick*.

EXAMPLE 12.3. Use **Quarter** as a shorthand for:

of-which-4-can-be-exchanged-for-1-Dollar

Then, we can explain what all numbers of QUARTERS are:

2 **Quarters** = 2 **of-which-4-can-be-exchanged-for-1-Dollar**

3 **Quarters** = 3 **of-which-4-can-be-exchanged-for-1-Dollar**

4 **Quarters** = 1 **Dollar**

5 **Quarters** = 5 **of-which-4-can-be-exchanged-for-1-Dollar**

etc

4. An immediate advantage is that it also makes it will make it easier to compare *fractions* with *counting numerators*.

EXAMPLE 12.4. We can see that

5 **of-which-4-can-be-exchanged-for-1-Dollar**

is the same amount of money as

1 **Dollar** & 1 **of-which-4-can-be-exchanged-for-1-Dollar**

fraction bar

and then we have:

2 **Quarters** = 2 **of-which-4-can-be-exchanged-for-1-Dollar**

3 **Quarters** = 3 **of-which-4-can-be-exchanged-for-1-Dollar**

4 **Quarters** = 1 **Dollar**

5 **Quarters** = 1 **Dollar** & 1 **of-which-4-can-be-exchanged-for-1-Dollar**

6 **Quarters** = 1 **Dollar** & 2 **of-which-4-can-be-exchanged-for-1-Dollar**

7 **Quarters** = 1 **Dollar** & 3 **of-which-4-can-be-exchanged-for-1-Dollar**

8 **Quarters** = 2 **Dollars**

9 **Quarters** = 2 **Dollar** & 1 **of-which-4-can-be-exchanged-for-1-Dollar**

etc

5. However, the above is of course not the way we usually write fractions because it would make it awkward to develop procedures for calculating with fractions and we are now going to see how we came about to write, and what is involved when we *write*, say,

$$\frac{3}{4}\ \textbf{Dollar}$$

Starting from

$$3\ \textbf{Quarters}$$

we already saw above that this is

$$3\ \textbf{of-which-4-can-be-exchanged for-1-Dollar}$$

Now we write the denominator **of-which-4-can-be-exchanged for-1-Dollar** symbolically as $\boxed{4 \rightarrow 1\ \textbf{Dollar}}$ so that we now have

$$3\ \boxed{4 \rightarrow 1\ \textbf{Dollar}}$$

What happened at this point is that, instead of writing the *denominator* next to the *numerator*, the *denominator* came to be written under the *numerator*:

$$\frac{3}{\boxed{4 \rightarrow 1\ \textbf{Dollar}}}$$

and then of course the box disappeared since the **fraction bar** was enough of a separator:

$$\frac{3}{4 \rightarrow 1\ \textbf{Dollar}}$$

But then, part of the denominator went back to the right of the numerator which required the arrow to be bent improper fraction

$$\frac{3}{4} \xrightarrow{\;1\ \textsf{Dollar}\;}$$

and, as usual, the 1 then went without saying

$$\frac{3}{4} \xrightarrow{\;\textsf{Dollar}\;}$$

and finally the arrow disappeared which gives us what we usually write

$$\frac{3}{4}\ \textsf{Dollar}$$

If we look at this as "3", "bar", "4", "**Dollars**" in that order, then the fraction bar continues to act as a separator between "3", the *numerator*, and "4", "**Dollars**", the *denominator* which, to make sense, must still be read as "4 for a **Dollars**", in other words, we have

3 **Items-at-**4**-for-a-Dollar**

Thus, while 3 is truly a *numerator*, 4 is only *part* of the *denominator* and, if we forget this, then it becomes impossible to deal with fractions on the basis of common sense.

Altogether, it is crucial to see

$$\frac{3}{4}\ \textsf{Dollar}$$

as just a *shorthand* for

3 **Items-at-**4**-for-a-Dollar**

2 Mixed-Numbers and Improper Fractions

For some reason, by now lost in time, number-phrases such as

4 **Quarters**

5 **Quarters**

6 **Quarters**

etc

are called in school language **improper fractions** even though there is nothing wrong with them.

1. Converting "improper fractions", for instance $\frac{9}{4}$ **Dollar**, to something presumably more "proper" is a favorite school exercise but is absolutely straightforward if we keep in mind that

- the *numerator* is 9
- the *denominator* is **Of-which-**4**-can-be-exchanged-for-**1**-Dollar**

mixed number

Then,

$$\frac{9}{4}\ \textsf{Dollar} = 9\ \textsf{Of-which-4-can-be-exchanged for-1-Dollar}$$
$$= 2\ \textsf{Dollars}\ \&\ 1\ \textsf{Of-which-4-can-be-exchanged for-1-Dollar}$$

which can be coded as

$$= 2\ \textsf{Dollar}\ \&\ \frac{1}{4}\ \textsf{Dollar}$$

so that, since the denominators are the same,

$$= \left[2 + \frac{1}{4}\right]\ \textsf{Dollars}$$

which in fact is often written

$$= 2\tfrac{1}{4}\ \textsf{Dollars}$$

where $2\frac{1}{4}$ is called a **mixed number** and where the fact that the fraction is written with smaller digits is supposed to warn that the missing operation sign is a $+$.

2. Converting "mixed-numbers" to "improper fractions" is an equally popular exercise in school and just as straightforward as the former one. For instance,

$$2\tfrac{1}{4}\ \textsf{Dollars} = \left[2 + \frac{1}{4}\right]\ \textsf{Dollars}$$
$$= 2\ \textsf{Dollar}\ \&\ \frac{1}{4}\ \textsf{Dollar}$$
$$= 2\ \textsf{Dollars}\ \&\ 1\ \textsf{Of-which-4-can-be-exchanged for-1-Dollar}$$

which tells us at what rate to change the 2 **Dollars**

$$= 8\ \textsf{Of-which-4-can-be-exchanged for-1-Dollar}\ \&\ 1\ \textsf{Of-which-4-can-be-exchanged for-1-Dollar}$$

which we can code as

$$= \frac{8}{4}\ \textsf{Dollar} + \frac{1}{4}\ \textsf{Dollar}$$

and, since the denominators are the same,

$$= \frac{9}{4}\ \textsf{Dollar}$$

3 Fractions as Code for Division

These days, other than in the schools and the use of "half", "quarter"[1], "eighth" etc in construction, fractions are now mostly used as code for *divi-*

[1] Notice by the way the demise of "half", "quarter" brought about by digital watches: who still says "a quarter to two"?

sion.

4 Comparison of Fractions

The safest and usually fastest way to *compare* fractions is to carry out both divisions to enough digits that the quotients become different.

EXAMPLE 12.5. To *compare* $\frac{5}{7}$ and $\frac{17}{25}$, do both divisions until the quotients become differents:
i. Doing the divisions to the "ones":
- $5 \div 7 = 0. + [...]$
- $17 \div 25 = 0. + [...]$

Therefore continue the divisions
ii. Doing the divisions to the "tenths":
- $5 \div 7 = 0.7 + [...]$
- $17 \div 25 = 0.6 + [...]$

Therefore $\frac{5}{7} > \frac{17}{25}$

Chapter 13

Comparisons

Chapter 14

Multiplication

Chapter 15

Division

Chapter 16

Division

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the

Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright ©YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Index