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

Chapter 5

Addition - Subtraction

5.1 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 5.5.1. 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:

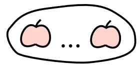
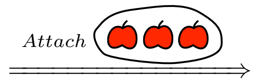
$$\text{Collection of } \textit{green } \textbf{\textit{apples}} \xRightarrow{Sun} \text{Collection of } \textit{ripe } \textbf{\textit{apples}}$$

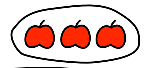
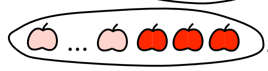
5.2 Attaching And Detaching A Collection

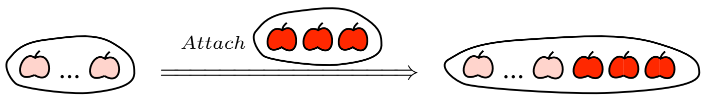
In this chapter, we will deal only with two related kinds of agents of change.

attach
add-to collection
detach
take-from collection
function
input-output rule
unspecified input
specific inputs

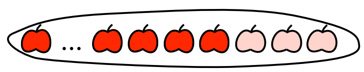
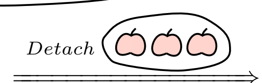
1. The first kind of agents of change **attaches** a given **add-to collection** *to* a collection in the initial state to get the final state of that collection.

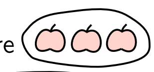
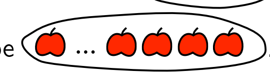
Example 5.5.2. Let  be the *initial state* some collection is in. After using the agent of change 

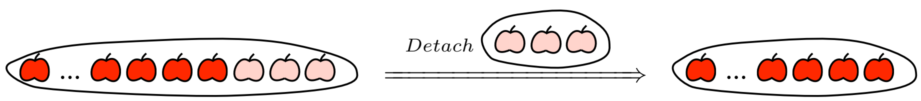
, where  is the *add-to collection*, the *final state* of the collection will be . In short, the *action* is:



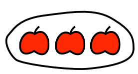
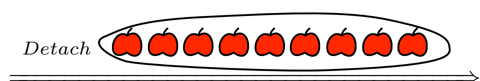
2. The second kind of agents of change **detaches** a given **take-from collection** *from* a collection in the initial state to get the final state of that collection.

Example 5.5.3. 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 5.5.4. Let  be the *initial state* a collection is in. Obviously, we cannot use the agent of change 

.

5.3 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.

function name
output specifying code
specific outputs
adding function
addition
+

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

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

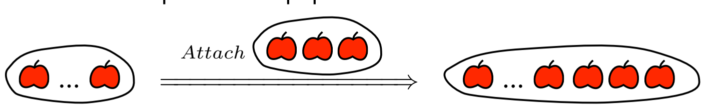
represents on paper the real world *action* of an agent of change

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

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 5.5.5. To represent on paper the real world *action*



we write the *input-output rule*

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

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

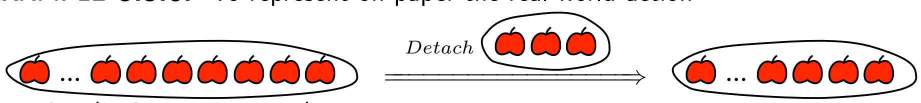
$$\underbrace{x}_{\text{Unspecified input}} \xrightarrow{+3} \underbrace{x+3}_{\text{Output specifying code}}$$

subtracting function
subtraction
–
execute

LANGUAGE 5.1. This 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 5.5.6. To represent on paper the real world *action*



we write the *input-output rule*

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

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

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

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 $-$.

5.4 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*:

long addition
carryover

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

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

so that:

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

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 **?? ??** and then to use **long addition**, the procedure we learned in elementary school and which is really nothing more than *counting up* with "**carryover**".

EXAMPLE 5.5.8. 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*:

$$5\,153.473\ \text{Meters}$$

so that:

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

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

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

long subtraction
borrowing
translating function
translated set

so that:

$$\left[13\,627.48 - 3\right] \text{Meters} = 13\,624.48\,\text{Meters}$$

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 **?? ??** and then to use **subtraction addition**, the procedure we learned in elementary school and which is really nothing more than *counting down* with "**borrowing**".

EXAMPLE 5.5.10. 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	$\cancel{9}$	$\cancel{6}$	$\cancel{3}$	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:

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

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*.

5.5 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 5.5.11. In order to translate

the data set: $\underbrace{\{2, 3, 9, 6, 7\}}_{\textit{Numerator Set}}\ \underbrace{\text{Apples}}_{\textit{Common Denominator}}$

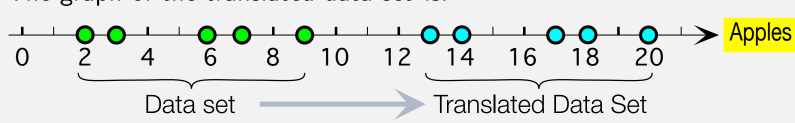
with the adding function: $\underbrace{\boxed{x}}_{\text{Unspecified input}} \xrightarrow{+11} \underbrace{\boxed{x}+11}_{\text{Output specifying code}}$

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 5.5.12. In order to translate

the data set: $\underbrace{\{18, 13, 19, 15, 17\}}_{\text{Numerator Set}}$ $\underbrace{\textbf{Apples}}_{\text{Common Denominator}}$

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

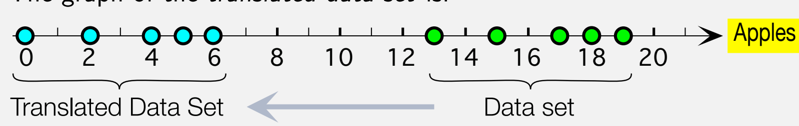
reverse

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:



5.6 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 5.5.13. Given the adding function $\xrightarrow{+7 \text{ Apples}}$, the subtraction function reverse problem

$\xrightarrow{-7\text{ Apples}}$ 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 5.5.14. 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:

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

5.7 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 5.5.15. Given the *reverse* addition problem

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

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

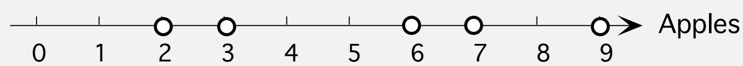
In order to solve this comparison problem,

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 5.5.16. Given the *reverse* subtraction problem

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

  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:

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

Index