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:

\[
\text{Initial collection} \overset{\text{Agent of Change}}{\longrightarrow} \text{Final collection}
\]

**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:

\[
\text{Collection of green apples} \overset{\text{Sun}}{\longrightarrow} \text{Collection of ripe apples}
\]

5.2 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 5.5.2.** Let be the initial state some collection is in. After using the agent of change \( \text{Attach} \), 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 \( \text{Detach} \), 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. Thus, the input-output rule of a function represents on paper the real world action of an agent of change.

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

![Diagram of adding function](image)

we write the input-output rule

![Input-output rule diagram](image)

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

![Input-output rule diagram](image)
2. A **subtracting function**, often called **subtraction** for short, is a function that represents an agent 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*

![Diagram of Detach operation]

we write the *input-output rule*

\[
\begin{align*}
\text{Unspecified input} & \quad \text{Apples} \quad \rightarrow \quad (x - 3) \text{ Apples} \\
\text{Output specifying code} & \quad \text{Unspecified input} \quad \rightarrow \quad x - 3
\end{align*}
\]

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

\[
\begin{align*}
\text{Unspecified input} & \quad \rightarrow \quad (x - 3) \\
\text{Output specifying code} & \quad \rightarrow \quad x - 3
\end{align*}
\]

**Language 5.2.** This use of the symbol − to represent *detachment* is only the first among 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*:
5.4. Procedures For Additions and Subtractions

1. 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 13627.48 Meters we count 5 up from 13627.48:

\[
\text{13627.48, 13627.48, 13627.48, 13627.48, 13627.48}
\]

so that:

\[
\left[13627.48 + 5\right] \text{Meters} = 13627.53 \text{Meters}
\]

2. 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 4627.47 Meters we place both numerators under a decimal header:

<table>
<thead>
<tr>
<th>Thousand</th>
<th>Hundred</th>
<th>Ten</th>
<th>Single</th>
<th>Tenth</th>
<th>Hundredth</th>
<th>Thousandth</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

and then we do the long addition under the header:

<table>
<thead>
<tr>
<th>Thousand</th>
<th>Hundred</th>
<th>Ten</th>
<th>Single</th>
<th>Tenth</th>
<th>Hundredth</th>
<th>Thousandth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Which gives us the decimal number-phrase:

\[5153.473 \text{ Meters}\]

so that:

\[\left[4627.47 + 526.003\right] \text{Meters} = 5153.473 \text{ Meters}\]

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

1. 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 13627.48 Meters we can count 3 down from 13627.48:

\[
\text{13627.48, 13627.48, 13627.48}
\]
Chapter 5. Addition - Subtraction

long subtraction
borrowing
translating function
translated set

so that:

\[
\left[ 1362.48 - 3 \right] \text{Meters} = 1362.48 \text{Meters}
\]

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 5796.3 Meters we place both numerators under a decimal header:

<table>
<thead>
<tr>
<th>Thousand</th>
<th>Hundred</th>
<th>Ten</th>
<th>Single</th>
<th>Tenth</th>
<th>Hundredth</th>
<th>Thousandth</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

and then we do the long subtraction under the header:

<table>
<thead>
<tr>
<th>Thousand</th>
<th>Hundred</th>
<th>Ten</th>
<th>Single</th>
<th>Tenth</th>
<th>Hundredth</th>
<th>Thousandth</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>7</td>
<td>9</td>
<td>6</td>
<td>3</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Which gives us the decimal number-phrase:

5168.82 Meters

so that:

\[
[5796.3 - 627.48] \text{Meters} = 5168.82 \text{Meters}
\]

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:

\[
\left\{ 2, 3, 9, 6, 7 \right\}
\]

\text{Apples}

\text{Numerator Set} \quad \text{Common Denominator}
5.5. Translations

with the adding function:

\[ x \rightarrow x + 11 \]

Unspecified input  Output specifying code

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

\begin{itemize}
  \item \( x \rightarrow x + 11 \) gives \( 2 \rightarrow 13 \) so the output is \( 13 \).
  \item \( x \rightarrow x + 11 \) gives \( 3 \rightarrow 14 \) so the output is \( 14 \).
  \item \( x \rightarrow x + 11 \) gives \( 9 \rightarrow 20 \) so the output is \( 20 \).
  \item \( x \rightarrow x + 11 \) gives \( 6 \rightarrow 17 \) so the output is \( 17 \).
  \item \( x \rightarrow x + 11 \) gives \( 7 \rightarrow 18 \) so the output is \( 18 \).
\end{itemize}

ii. The translated data set is therefore \( \{13, 14, 20, 17, 18\} \). Apples.

iii. The graph of the translated data set is:

\[ \text{Data set} \rightarrow \text{Translated Data Set} \]

\[ 0 \quad 2 \quad 4 \quad 6 \quad 8 \quad 10 \quad 12 \quad 14 \quad 16 \quad 18 \quad 20 \]

\[ \text{Apples} \]

EXAMPLE 5.5.12. In order to translate

the data set:

\[ \{18, 13, 19, 15, 17\} \]

Numerator Set  Common Denominator

with the subtracting function:

\[ x \rightarrow x - 13 \]

Unspecified input  Output specifying code
i. We specify $x$ to be each and every one of the numerators in the numerator set:

- $x - 13 \rightarrow x - 13$ gives $18 - 13 \rightarrow 5$ so the output is 5.
- $x - 13 \rightarrow x - 13$ gives $13 - 13 \rightarrow 0$ so the output is 0.
- $x - 13 \rightarrow x - 13$ gives $19 - 13 \rightarrow 6$ so the output is 6.
- $x - 13 \rightarrow x - 13$ gives $15 - 13 \rightarrow 2$ so the output is 2.
- $x - 13 \rightarrow x - 13$ gives $17 - 13 \rightarrow 4$ so the output is 4.

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

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 $+7 \rightarrow$, the subtraction function...
5.7 Reverse Problems

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

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

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:** $\{2, 3, 9, 6, 7\}$
- **Function:** $x \overrightarrow{+4}$
- **Comparison formula:** $x + 4 \leq 7$

where $\leq$ is the comparison verb.
In order to solve this comparison problem,

**EXAMPLE 5.5.16.** Given the reverse subtraction problem

- **Data set:** \{2, 3, 9, 6, 7\}  
  - Numerator Set  
  - Common Denominator  
  - Apples

- **Function:**  
  - Unspecified input \( x \)  
  - Output specifying code \( x - 4 \)  
  
- **Comparison formula:**  
  - Output specifying code \( x - 4 \)  
  - Gauge numerator \( \leq \)  
  - \( 8 \)  

where \( \leq \) is the comparison verb

In order to solve this comparison problem,
5.7. Reverse Problems

i. We specify \( x \) to be each and every numerator in the numerator set:

- \( x - 4 \leq 7 \) gives the comparison sentence \( 2 - 4 \leq 7 \) but since we cannot execute \( 2 - 4 \), the sentence is FALSE.
- \( x - 4 \leq 7 \) gives the comparison sentence \( 3 - 4 \leq 7 \) but since we cannot execute \( 3 - 4 \), the sentence is FALSE.
- \( x - 4 \leq 7 \) gives the comparison sentence \( 5 \leq 7 \) which is TRUE.
- \( x - 4 \leq 7 \) gives the comparison sentence \( 2 \leq 7 \) which is TRUE.
- \( x - 4 \leq 7 \) gives the comparison sentence \( 3 \leq 7 \) which is TRUE.

ii. The solution subset is therefore \( \{9, 6, 7\} \).

iii. The graph of the solution subset is

```
0 1 2 3 4 5 6 7 8 9  Apples
```
## Index

- $+$, 3
- $-$, 4
- action, 1
- add-to collection, 2
- adding function, 3
- addition, 3
- agent of change, 1
- attach, 2
- borrowing, 6
- carryover, 5
- detach, 2
- execute, 4
- final collection, 1
- final state, 1
- function, 2
- function name, 3
- initial collection, 1
- initial state, 1
- input-output rule, 2
- long addition, 5
- long subtraction, 6
- output specifying code, 3
- reverse, 8
- reverse problem, 9
- specific inputs, 2
- specific outputs, 3
- subtracting function, 4
- subtraction, 4
- take-from collection, 2
- translated set, 6
- translating function, 6
- unspecified input, 2