Chapter 8

Operators

In order to go about getting qualitative global graphs for plain polynomial functions, that is functions that are “combinations” of non-negative-exponent power functions., we first need to develop a bit of “machinery” that we will use in all the rest of this text, starting next chapter.

8.1 Chaining Functions

The functions we have seen so far in this text each represent on paper a fairly simple real world process while, quite often, we will want to investigate much more complicated real world processes.

1. Before anything else, though, we must discuss something about the way we write functions because here it is going to complicate matters a bit. This is the fact that
   • In the arrow diagram that pictures a function, the input is written to the left of the function while
   • in the output, we write the input inside a parenthesis which is to the right of the name of the function

**Example 1.** Given the function \( TOBY \) and the input \(+13\), we write

\[
+13 \rightarrow_{TOBY} TOBY(+13)
\]
in which:

- the input $+13$ is to the left of the arrow that pictures the function $TOBY$
  
  $+13 \xrightarrow{TOBY} TOBY(+13)$

while

- in the output, the parenthesis in which we write the input $+13$ is to the right of $TOBY$
  
  $+13 \xrightarrow{TOBY} TOBY(+13)$

There was of course no logical reason to do so and, in the output, we could have written the parenthesis in which we write the input to the left of the name of the function. And, in fact, since the relatively recent advent of the use of arrow diagrams to picture functions, writing the parenthesis in which we write the input to the left of the name of the function has often become the practice among mathematicians.

**Example 2.** Given the function $TOBY$ and the input $+13$, mathematicians using arrow diagrams would likely write

$+13 \xrightarrow{TOBY} (+13)\ TOBY$

However and as it happens, writing the parenthesis in which we write the input to the right of the name of the function does not create any confusion as long as we look at functions one at a time which is what we have done so far and which we will do after this chapter.

So, since the “elementary textbooks” currently on the market have yet to use arrow diagrams to picture functions, the choice was thus made for REASONABLE ALGEBRAIC FUNCTIONS to conform with commercial textbooks in this matter since, at least for the time being, a commercial textbook is likely what the reader will have to use after this one.

Nevertheless, writing the input to the right of the name of the function certainly does not help and one has to be very careful until accustomed when dealing with several functions at the same time as we will have to do next and as the reader will also have to do in subsequent courses.

2. Indeed, quite often we can break a complicated real world process into a succession of real world processes for which we already have functions to represent them on paper. What we then do is to chain these functions, that is have each of these functions use at its input the output of the preceding function.

**Example 3.** Say we have a complicated real world process that can be broken into a succession of three processes that can be represented by the functions $f_1, f_2, f_3$ in
8.1. CHAINING FUNCTIONS

that order. The real world process can then be represented by the chain:

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

a. It is when when we follow what happens to an input to the chain throughout the chain that we see how writing the parenthesis to the right creates a problem: the order of the names of the functions in the output is the opposite of the order of the functions in the chain.

**EXAMPLE 4.** Given the chain

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

we follow what happens throughout the chain when we input \(-7\):

\[
\begin{align*}
-7 & \rightarrow f_1 \rightarrow f_1(-7) \\
 & \rightarrow f_2 \rightarrow f_2(f_1(-7)) \\
 & \rightarrow f_3 \rightarrow f_3(f_2(f_1(-7)))
\end{align*}
\]

The annoying fact is that in the outputs, the names of the functions, \(f_1, f_2, \) and \(f_3\), appear in the order opposite from the order in which they appear in the chain.

b. It is interesting to see that when we write in the output the parenthesis in which we write the input to the left of the name of the function, the order in which the names of the functions appear in the chain and in the output are then the same.

**EXAMPLE 5.** Given the chain

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

we follow what happens throughout the chain when we input \(+13\):

\[
\begin{align*}
+13 & \rightarrow f_1 \rightarrow (+13)f_1 \\
 & \rightarrow f_2 \rightarrow ((+13)f_1)f_2 \\
 & \rightarrow f_3 \rightarrow (((+13)f_1)f_2)f_3
\end{align*}
\]

The names of the functions, \(f_1, f_2, \) and \(f_3\), appear in the output in exactly the same order as in the chain.
c. In fact, it also happens that when writing outputs in this manner there is no need to enclose the inputs within parentheses when writing outputs. This is called the **Reverse Polish Notation**.\footnote{It is worth noting that Educologists, always in search of “new approaches” to facilitate matters for the students, have never even considered the matter.}

**Example 6.** Given the chain

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

we follow what happens throughout the chain when we input +13:

\[ +13 \rightarrow f_1 \rightarrow +13f_1 \]
\[ +13f_1 \rightarrow f_2 \rightarrow +13f_1 f_2 \]
\[ +13f_1 f_2 \rightarrow f_3 \rightarrow +13f_1 f_2 f_3 \]

The names of the functions, \( f_1 \), \( f_2 \), and \( f_3 \), appear in the output in exactly the same order as in the chain.

3. Since we start with an input and end up with an output, a chain of “simple” functions works in fact like a single “complicated” function which is called the **composite function** of the chain of functions.

**Example 7.** Given the chain

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

we have the single, equivalent, composite function.

\[ f_2 \rightarrow f_3 \rightarrow \text{Composite Function} \]

a. The way we will code the *composite* of a chain in terms of the names of the functions will be to write the functions in the order they appear in the chain and separate them by the *chaining symbol* \( \circ \) which is read “followed by”.

\[ f_1 \circ f_2 \circ f_3 \]
However, as a result of our writing in the output the parenthesis in which we write the input to the right of the name of the function, the order in which the names of the functions appear in the composite is again the opposite of the order of the names of the functions in the output.

**Example 8.** Given the chain

\[ f_1 \rightarrow f_2 \rightarrow f_3 \]

when we follow the outputs we get

\[ f_3(f_2(f_1(x))) \]

in which the order in which the names of the functions in the end output is the opposite of the order in which the names of the functions appear in the composite.

**c.** The way we code the composite function of a chain of functions is not the one most commonly used in “elementary” textbooks. Here is why we do not conform with “common” usage:

There are three things involved in a chain:

1. The order of the functions in the chain itself
2. The order of the names of the functions as they appear in the composite of the chain.
3. The order in which the names of the functions appear in the output of the chain.

The problem is of course the one we discussed above, namely that once we write the parenthesis in which we write the input to the right of the name of the function then the order in i. and iii. are automatically going to be opposite and the only thing we can decide is what to chose for ii.:

- Most “elementary” textbooks being “answer oriented” side with iii.
- This text being “process oriented” chose to side with i.

Note though that, had we written the parenthesis in which we write the input to the left of the name of the function, then the order in i. and iii. would have been the same and then there would have been no question as what to choose for ii.

**4.** Depending on the kind of functions that are in the chain, the order of the functions in the chain may or may not matter as to what the output of the chain will be. So, it is advisable not to be sloppy and make sure not to change the order whenever we are copying something.
8.2 Operators

When given an input function an operator returns an output function just like, when given an input number, a function returns an output number. In other words, operators are to functions exactly the same as what functions are to numbers.

Here, though, we will only be concerned with chain operators, that is with operators that chain the input function with a given function associated with the chain operator and which we will call the operating function. There is therefore going to be two kinds of chain operators depending on the place of the operating function relative to the input function:
• pre-chaining operators in which the operating function comes in the chain before the input function,
• post-chaining operators in which the operating function comes in the chain after the input function.

However, since the only operators we will be using in this text will be post-chaining operators, we will tend to use the word “operator” as a shorthand for “post-chaining operator”. In fact, we will often say that the operating function operates on the input function to give the output function.

We now look at increasingly sophisticated kinds of operators, the first few to familiarize ourselves with operators but the last ones because we will be using operators of that kind all the time in the rest of this text.

8.3 Constant Operators

The simplest possible operators are constant operators, that is post-chaining operators whose operating function is a constant function.

Let the the operating function of the constant operator be the constant function \( CONSTANT_a \)

\[
x \xrightarrow{CONSTANT_a} CONSTANT_a(x) = a
\]

we want to investigate how it operates on a given input function \( F \)

\[
x \xrightarrow{F} F(x)
\]

i. We have
ii. Now since, no matter what its input is, the function $\text{CONSTANT}_a$ will return the number $a$, we have

$$F(x) \xrightarrow{\text{CONSTANT}_a} \text{CONSTANT}_a(F(x)) = a$$

iii. Therefore, no matter what $x$ is,

$$x \xrightarrow{F} F(x) \xrightarrow{\text{CONSTANT}_a} a$$

iv. That is the chain $F \xrightarrow{\text{CONSTANT}_a} \text{CONSTANT}_a$ is functionally equal to the single function $\text{CONSTANT}_a$ which we can write in three different ways:

- By way of a functional equality:

$$F \xrightarrow{\text{CONSTANT}_a} = \text{CONSTANT}_a$$

- By way of a commutative (arrow) diagram:

- By way of a composite function:

$$F \circ \text{CONSTANT}_a = \text{CONSTANT}_a$$

v. Altogether, we have found that

$$x \xrightarrow{F} F(x) \xrightarrow{\text{CONSTANT}_a} a$$
EXAMPLE 9. Let the operating function of the constant operator be the constant function $CONSTANT_{-3}$

$$x \xrightarrow{CONSTANT_{-3}} CONSTANT_{-3}(x) = -3$$

Then, given the input function $LUKE$ specified by the table:

<table>
<thead>
<tr>
<th>$x$</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LUKE(x)$</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

we have:

<table>
<thead>
<tr>
<th>$x$</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LUKE(x)$</td>
<td>+1</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

From the graphic viewpoint, we have
In other words, no matter what the input is, the output of a constant operator is a constant function in exactly the same manner that the output of a constant function is a constant number.

As a result, from the graphic viewpoint, a constant operator replaces whatever the graph of the input function is by the graph of a constant function, that is by a horizontal line.
8.4 Translation Operators

This is the next simpler kind of operator but its definition requires that we first define a new kind of function to be used as operating function.

1. A translation function is a function which adds to the input a given signed number $a$.

   **Example 10.** Let $HANK_{-3}$ be the translation function specified by the input-output rule

   \[
   x \xrightarrow{HANK_{-3}} HANK_{-3}(x) = x + (-3)
   \]

   Then a table for $HANK_{-3}$ is

   \[
   \begin{array}{cccccccccccc}
   x & -6 & -5 & -4 & -3 & -2 & -1 & 0 & +1 & +2 & +3 & +4 & +5 & +6 \\
   \hline
   HANK_{-2}(x) & -9 & -8 & -7 & -6 & -5 & -4 & -3 & -2 & -1 & 0 & +1 & +2 & +3
   \end{array}
   \]

   and, from the graphic viewpoint, a plot of $HANK_{-2}$ is:

   ![Plot of HANK_{-2}]

2. We now define a translation operator as a post-chaining operator whose operating function is a translation function $TRANSLATION_a$

   \[
   x \xrightarrow{TRANSLATION_a} TRANSLATION_a(x) = x + a
   \]

   We want to investigate how it operates on a given input function $F$

   \[
   x \xrightarrow{F} F(x)
   \]

   i. We have
ii. Now since, no matter what its input is, the function $\text{TRANSLATION}_a$ will add the number $a$ to the input, we have

$$F(x) \xrightarrow{\text{TRANSLATION}_a} \text{TRANSLATION}_a(F(x)) = F(x) + a$$

iii. Therefore, no matter what $x$ is,

$$x \xrightarrow{F} F(x) \xrightarrow{\text{TRANSLATION}_a} F(x) + a$$

iv. Altogether

So, contrary to what happened with constant operators, the output function is \textit{not functionally equal} to any function that we have already encountered.

**Example 11.** Let the operating function of the translation operator be the translation function $\text{TRANSLATION}_{-2}$

$$x \xrightarrow{\text{TRANSLATION}_{-2}} \text{TRANSLATION}_{-2}(x) = x + (-2)$$

Then, given the input function \textit{LUKE} specified by the table:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$-6$</th>
<th>$-5$</th>
<th>$-4$</th>
<th>$-3$</th>
<th>$-2$</th>
<th>$-1$</th>
<th>$0$</th>
<th>$+1$</th>
<th>$+2$</th>
<th>$+3$</th>
<th>$+4$</th>
<th>$+5$</th>
<th>$+6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{LUKE}(x)</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$+1$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$-1$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-2$</td>
</tr>
</tbody>
</table>

we have:
From the graphic viewpoint, we have
8.5. DILATION OPERATORS

In other words, in order to create the graph of the output function, a translation operator just shifts the graph of the input function, up/down according to the sign of \( a \) and of course by an amount equal to the size of \( a \).

8.5 Dilation Operators

A dilation operator is a post-chaining operator whose operating function is a dilation function \( \text{DILATION}_a \)

\[
\begin{align*}
x & \overset{\text{DILATION}_a}{\rightarrow} \text{DILATION}_a(x) = ax
\end{align*}
\]

We want to investigate how it operates on a given input function \( F \)

\[
\begin{align*}
x & \overset{F}{\rightarrow} F(x)
\end{align*}
\]

i. We have

\[
\begin{array}{c|c}
\text{Input function:} & \begin{align*}x & \overset{F}{\rightarrow} F(x)\end{align*} \\
\text{Output function:} & \begin{align*}x & \overset{F}{\rightarrow} F(x) \overset{\text{DILATION}_a}{\rightarrow} \text{DILATION}_a(F(x))\end{align*}
\end{array}
\]

ii. Now since, no matter what its input is, the function \( \text{DILATION}_a \) will multiply the number \( a \) by the input, we have

\[
\begin{align*}
F(x) & \overset{\text{DILATION}_a}{\rightarrow} \text{DILATION}_a(F(x)) = a \cdot F(x)
\end{align*}
\]

iii. Therefore, no matter what \( x \) is,

\[
\begin{align*}
x & \overset{F}{\rightarrow} F(x) \overset{\text{DILATION}_a}{\rightarrow} a \cdot F(x)
\end{align*}
\]

iv. Altogether

\[
\begin{array}{c|c}
\text{Input function:} & \begin{align*}x & \overset{F}{\rightarrow} F(x)\end{align*} \\
\text{Output function:} & \begin{align*}x & \overset{F}{\rightarrow} a \cdot F(x)\end{align*}
\end{array}
\]
While dilation operators are not particularly complicated, in order to make sure of what they do graphically we now look at the following three cases.

1. We begin by looking at positive dilation operators, that is (post-)dilation operators where the operating dilation function has a positive coefficient. In that case we find that the corresponding graphic procedure is just a vertical stretch from the 0-output level line of the plot, that is the plot of the input function is stretched vertically away from the 0-output level line as if it had been drawn on a sheet of rubber glued to the 0-output level line.

**Example 12.** Given the function $LUKE$ specified by the table:

| $x$  | $-6$ | $-5$ | $-4$ | $-3$ | $-2$ | $-1$ | $0$  | $1$  | $2$  | $3$  | $4$  | $5$  | $6$  |
|------|------|------|------|------|------|------|------|------|------|------|------|------|
| $LUKE(x)$ | $+2$ | $+1$ | $0$  | $+1$ | $+2$ | $+2$ | $+1$ | $0$  | $-1$ | $-2$ | $-2$ | $-2$ |

whose plot is

![Output Ruler](image)

and given the dilation function $DILATION_{+3}$ specified by the input-output rule

$$x \xrightarrow{D_{+3}} DILATION_{+3}(x) = +3x$$

we want to see the effect of the corresponding dilatation operator:

![Dilation operator](image)

that is we want to see the table and the plot of the output function $LUKE \circ D_{+3}$.
With not too much work, we find that:

- The table of $LUKE \circ DILATION_{-3}$:

<table>
<thead>
<tr>
<th>$x$</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LUKE(x)$</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>+3$LUKE(x)$</td>
<td>+6</td>
<td>+3</td>
<td>0</td>
<td>+3</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
<td>+3</td>
<td>0</td>
<td>-3</td>
<td>-6</td>
<td>-6</td>
<td>-6</td>
</tr>
</tbody>
</table>

- The plot of $LUKE \circ DILATION_{-3}$:
2. We continue by looking at the opposite operator, that is the (post-)dilation operator whose coefficient is $-1$ because multiplying by $-1$ amounts to just taking the opposite. In that case we find that the graphic procedure is just a vertical flip of the plot, that is the plot of the input function is flipped around the 0-output level line.

**Example 13.** Given the function $LUKE$ specified by the table:

<table>
<thead>
<tr>
<th>$x$</th>
<th>$-6$</th>
<th>$-5$</th>
<th>$-4$</th>
<th>$-3$</th>
<th>$-2$</th>
<th>$-1$</th>
<th>$0$</th>
<th>$+1$</th>
<th>$+2$</th>
<th>$+3$</th>
<th>$+4$</th>
<th>$+5$</th>
<th>$+6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LUKE(x)$</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$+1$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$-1$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-2$</td>
</tr>
</tbody>
</table>

opposite operator
vertical flip
whose plot is

![Diagram of a plot]

and given the dilation function $DILATION_{-1}$ specified by the input-output rule

$$x \xrightarrow{DILATION_{-1}} DILATION_{-1}(x) = -1x$$

we want to see the effect of the corresponding dilatation operator, that is of the $OPPOSITE$ operator:

![Flowchart showing the effect of the $OPPOSITE$ operator]

that is we want to see the table and the plot of the output function $LUKE \circ OPPOSITE$:

![Diagram showing the effect of the $OPPOSITE$ operator on $LUKE(x)$]

Without too much work, we find:

- The table of $LUKE \circ OPPOSITE$:
### CHAPTER 8. OPERATORS

<table>
<thead>
<tr>
<th>$x$</th>
<th>$-6$</th>
<th>$-5$</th>
<th>$-4$</th>
<th>$-3$</th>
<th>$-2$</th>
<th>$-1$</th>
<th>$0$</th>
<th>$+1$</th>
<th>$+2$</th>
<th>$+3$</th>
<th>$+4$</th>
<th>$+5$</th>
<th>$+6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$LUKE(x)$</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$+1$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+1$</td>
<td>$0$</td>
<td>$-1$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-2$</td>
</tr>
</tbody>
</table>

### Opposite operator

<table>
<thead>
<tr>
<th>$x$</th>
<th>$-6$</th>
<th>$-5$</th>
<th>$-4$</th>
<th>$-3$</th>
<th>$-2$</th>
<th>$-1$</th>
<th>$0$</th>
<th>$+1$</th>
<th>$+2$</th>
<th>$+3$</th>
<th>$+4$</th>
<th>$+5$</th>
<th>$+6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(-1) \cdot LUKE(x)$</td>
<td>$-2$</td>
<td>$-1$</td>
<td>$0$</td>
<td>$-1$</td>
<td>$-2$</td>
<td>$-2$</td>
<td>$-1$</td>
<td>$0$</td>
<td>$+1$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+2$</td>
<td>$+2$</td>
</tr>
</tbody>
</table>
3. We finally look at negative dilations, that is (post-)dilations with a negative dilation coefficient. In that case we find that the graphic action is a vertical stretch followed by a vertical flip ... or a vertical flip followed by a vertical stretch.

**Example 14.** Given the function \( LUKE \) specified by the table:

<table>
<thead>
<tr>
<th>( x )</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( LUKE(x) )</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
</tbody>
</table>

whose plot is

and given the dilation function \( DILATION_{-3} \) specified by the input-output rule

\[
x \mapsto DILATION_{-3} \quad DILATION_{-3}(x) = -3x
\]

we want to see the effect of the corresponding dilatation operator:

that is we want to see the table and the plot of the output function \( LUKE \circ DILATION_{+3} \):
In order not to use anything more than what we already did, what we do, though, is either

\[
\begin{align*}
& x \rightarrow \text{LUKE} \rightarrow \text{DILATION}_{+3} \rightarrow \text{OPPOSITE} \\
& x \rightarrow \text{OPPOSITE} \rightarrow \text{DILATION}_{-3} \rightarrow \text{LUKE}(x)
\end{align*}
\]

Both ways work because we have both \((-3) = (-1) \cdot (+3)\) and \((-3) = (+3) \cdot (-1)\).

For no particular reason, we choose the second route and with just a little bit more work than before we get:

- The table for \(\text{LUKE} \circ \text{DILATION}_{-3}\):

<table>
<thead>
<tr>
<th>(x)</th>
<th>(-6)</th>
<th>(-5)</th>
<th>(-4)</th>
<th>(-3)</th>
<th>(-2)</th>
<th>(-1)</th>
<th>(0)</th>
<th>(+1)</th>
<th>(+2)</th>
<th>(+3)</th>
<th>(+4)</th>
<th>(+5)</th>
<th>(+6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{LUKE}(x))</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>x</td>
<td>(-6)</td>
<td>(-5)</td>
<td>(-4)</td>
<td>(-3)</td>
<td>(-2)</td>
<td>(-1)</td>
<td>(0)</td>
<td>(+1)</td>
<td>(+2)</td>
<td>(+3)</td>
<td>(+4)</td>
<td>(+5)</td>
<td>(+6)</td>
</tr>
<tr>
<td>((-1) \cdot \text{LUKE}(x))</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>+1</td>
<td>+2</td>
<td>+2</td>
<td>+2</td>
</tr>
<tr>
<td>((-3) \cdot \text{LUKE}(x))</td>
<td>-6</td>
<td>-3</td>
<td>0</td>
<td>-3</td>
<td>-6</td>
<td>-6</td>
<td>-3</td>
<td>0</td>
<td>+3</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
<td>+6</td>
</tr>
</tbody>
</table>

- The plot for \(\text{LUKE} \circ D_{-3}\):
8.5. DILATION OPERATORS

---

**Opposite operator**

---

**Dilation operator**

---

**Stretch**

---
4. One aspect of dilation operators is that they can be seen as “de-normalizers”, that is as undoing what we did when we normalized power functions as far as the coefficient is concerned.

**THEOREM 1.** Any constant function \( \text{CONSTANT}_a \)  
\[
x \xrightarrow{\text{CONSTANT}_a} \text{CONSTANT}_a(x) = a
\]
can be obtained:
- from the function \( \text{UNIT}_+ \)  
  
  \[
x \xrightarrow{\text{UNIT}_+} \text{UNIT}_+(x) = +1
\]
  
  by a post-dilation with coefficient \( a \):
  
  \[
  \text{UNIT}_+(x) = +1
  \]
  
  or
  
  \[
  x \xrightarrow{\text{CONSTANT}_a} a = a \cdot \text{UNIT}_+(x) = a \cdot (+1)
  \]

- from the function \( \text{UNIT}_- \)  
  
  \[
x \xrightarrow{\text{UNIT}_-} \text{UNIT}_-(x) = -1
\]
  
  by a post-dilation with coefficient opposite \( a \):
  
  \[
  \text{UNIT}_-(x) = -1
  \]
  
  or
  
  \[
  x \xrightarrow{\text{CONSTANT}_a} a = \text{opposite } a \cdot \text{UNIT}_-(x) = -a \cdot -1
  \]

**THEOREM 2.** Any dilation function \( \text{DILATION}_a \)  
\[
x \xrightarrow{\text{DILATION}_a} \text{DILATION}_a(x) = ax
\]
can be obtained:
- from the function \( \text{IDENTITY} \)  
  
  \[
x \xrightarrow{\text{IDENTITY}} \text{IDENTITY}(x) = x
\]
  
  by a post-dilation with coefficient \( a \)
8.5. DILATION OPERATORS

\[
\begin{align*}
\text{IDENTITY}(x) &= x \\
 x &\xrightarrow{\text{DILATION}_a} ax = a \cdot \text{IDENTITY}(x) = a \cdot x
\end{align*}
\]

or

- from the function \text{OPPOSITE}

\[
\begin{align*}
 x &\xrightarrow{\text{OPPOSITE}} \text{OPPOSITE}(x) = -x \\
x &\xrightarrow{\text{DILATION}_a} ax = \text{opposite } a \cdot \text{OPPOSITE}(x) = -a \cdot -x
\end{align*}
\]

\[
\begin{align*}
\text{OPPOSITE}(x) &= -x \\
x &\xrightarrow{\text{DILATION}_a} ax = \text{opposite } a \cdot \text{OPPOSITE}(x) = -a \cdot -x
\end{align*}
\]

**Example 15.** Given the power function \text{BLIP} specified by the input-output rule

\[
x \xrightarrow{\text{BLIP}} \text{BLIP}(x) = (\frac{-82.6}{x})^{x+13}
\]

normalizing the coefficient gives

\[
x \xrightarrow{\text{BLIP}} \text{BLIP}(x) = (\frac{-1}{x})^{x+13}
\]

Letting the dilation function \text{DILATION}_{+82.6}

\[
x \xrightarrow{\text{DILATION}_{+82.6}} \text{DILATION}_{+82.6}(x) = +82.6x
\]

operate on the normalized power function

\[
x \xrightarrow{\text{BLIP}} \text{BLIP}(x) = (\frac{-1}{x})^{x+13}
\]

gives

\[
x \xrightarrow{\text{BLIP}} (\frac{-1}{x})^{x+13} \xrightarrow{\text{DILATION}_{+82.6}} +82.6 \cdot (\frac{-1}{x})^{x+13}
\]

that is the original power function

\[
x \xrightarrow{\text{BLIP}} -82.6x^{x+13}
\]

As such, post-dilation operators will remain in the background of what we will be doing which will essentially be to use normalized power functions.
8.6 Add-On Operators

So far, the operators that we saw did the same thing to each and every output number of the input function to produce the output number of the output function:

- Constant operators substituted the number $a$ to each and every output number of the input function,
- Translation operators added the number $a$ to each and every output number of the input function,
- Dilation operators multiplied by the number $a$ each and every output number of the input function.

In contrast, bundle operators can do something different to each and every output number of the input function to produce the output number of the output function. In particular, add-on operators, which we will be using all the time in the rest of this text, add to each and every output number of the input function the output number of the operating function for the same input number. In the case of add-on operators, we will refer to the operating function as the add-on function.

In other words, given the add-on function $A$

\[ x \xrightarrow{A} A(x) \]

and given the input function $F$

\[ x \xrightarrow{F} F(x) \]

the add-on $A(x)$ operator returns the output function $[F + A]$, that is the function specified by the global input-output rule

\[ x \xrightarrow{[F+A]} [F + A](x) = F(x) + A(x) \]

So, translation operators are just a special kind of add-on operators in which the add-on function is a constant function.

We now discuss and illustrate how we add-on a function depending on whether the input function and the add-on function are specified by:

- Input-output tables
- Global input-output rules
- Quantitative plots

1. When the input function and the add-on function are specified by input-output tables, things are completely straightforward. For instance, it is often the case that when we buy something, some surcharge is applied
and, even though there are two kinds of surcharge, we never have any trouble figuring things out.

- The surcharge can be a “fixed” surcharge such as a “shipping cost” and then the operator that gives the net cost is a translation operator.

**Example 16.** Given the following “price lists”, that is a function whose inputs are kinds of merchandise rather than numbers but whose outputs are numbers, namely dollar amounts,

<table>
<thead>
<tr>
<th>Books</th>
<th>List price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>52.45</td>
</tr>
<tr>
<td>English</td>
<td>47.80</td>
</tr>
<tr>
<td>History</td>
<td>62.75</td>
</tr>
<tr>
<td>Biology</td>
<td>74.50</td>
</tr>
<tr>
<td>Poetry</td>
<td>64.28</td>
</tr>
</tbody>
</table>

and if the shipping charge is $4.25, then the surcharge is given by the translation function $\text{SHIPPING CHARGE}_{4.25}$ and the Shipping Charge translation operator gives the net-prices as output function:

<table>
<thead>
<tr>
<th>Books</th>
<th>List price</th>
<th>Net price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>52.45</td>
<td>56.70 ( = 52.45 + 4.25)</td>
</tr>
<tr>
<td>English</td>
<td>47.80</td>
<td>52.05 ( = 47.80 + 4.25)</td>
</tr>
<tr>
<td>History</td>
<td>62.75</td>
<td>67.00 ( = 62.75 + 4.25)</td>
</tr>
<tr>
<td>Biology</td>
<td>74.50</td>
<td>78.75 ( = 74.50 + 4.25)</td>
</tr>
<tr>
<td>Poetry</td>
<td>64.28</td>
<td>68.53 ( = 64.28 + 4.25)</td>
</tr>
</tbody>
</table>

- The surcharge can be a “prorated” surcharge such as a “sales tax” and then the operator that gives the net cost is an add-on operator whose add-on function is a dilation function.

**Example 17.** Given the following “price lists”, that is a function whose inputs are kinds of merchandise rather than numbers but whose outputs are numbers, namely dollar amounts,
CHAPTER 8. OPERATORS

<table>
<thead>
<tr>
<th>Books</th>
<th>List price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>52.45</td>
</tr>
<tr>
<td>English</td>
<td>47.80</td>
</tr>
<tr>
<td>History</td>
<td>62.75</td>
</tr>
<tr>
<td>Biology</td>
<td>74.50</td>
</tr>
<tr>
<td>Poetry</td>
<td>64.28</td>
</tr>
</tbody>
</table>

and if the sales tax is 6%, then the surcharge is given by the dilation function \( \text{SALESTAX}_6\% \) and the Sales Tax \textit{add-on operator} gives the net-prices as output function:

<table>
<thead>
<tr>
<th>Books</th>
<th>List price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math</td>
<td>52.45 (= 52.45 + 0.06 \cdot 52.45)</td>
</tr>
<tr>
<td>English</td>
<td>50.67 (= 47.80 + 0.06 \cdot 52.45)</td>
</tr>
<tr>
<td>History</td>
<td>66.52 (= 62.75 + 0.06 \cdot 52.45)</td>
</tr>
<tr>
<td>Biology</td>
<td>78.97 (= 74.50 + 0.06 \cdot 52.45)</td>
</tr>
<tr>
<td>Poetry</td>
<td>68.14 (= 64.28 + 0.06 \cdot 52.45)</td>
</tr>
</tbody>
</table>

2. When the input function and the add-on function are specified by \textit{global input-output rules}, things are actually quite simple because we know how to add in algebra.

\textbf{Example 18.} The add-on operator \( ADD-ON_{MINT} \) is the add-on operator that corresponds to the power function \( MINT \).

If, for instance, the function \( MINT \) is specified by the global input-output rule

\[ x \rightarrow_{MINT} MINT(x) = -12.82x^4 \]

then, given any input-function \( f \),

\[ x \rightarrow f(x) \]

the add-on operator \( ADD-ON_{MINT} \) outputs the function \( [f + MINT] \) that is the function specified by the global input-output rule

\[ x \rightarrow_{[f+MINT]} [f + MINT](x) = f(x) - 12.82x^4 \]
In other words, the add-on operator \( \text{ADD-ON}_{\text{MINT}} \) adds the output of \( \text{MINT} \) to the output of \( f \).
For instance, if the input function to the operator \( \text{ADD-ON}_{\text{MINT}} \) is the power function \( \text{TEA} \) specified by the global input-output rule
\[
 x \xrightarrow{\text{TEA}} \text{TEA}(x) = +49.28x^7
\]
then the output function from the operator \( \text{ADD-ON}_{\text{MINT}} \) is the function \([\text{TEA} + \text{MINT}]\)
\[
 x \xrightarrow{\text{TEA}} \text{TEA}(x) \\
 \xrightarrow{\text{ADD-ON}_{\text{MINT}}} \\
 x \xrightarrow{[\text{TEA} + \text{MINT}]} [\text{TEA} + \text{MINT}](x)
\]
where the function \([\text{TEA} + \text{MINT}]\) is specified by the global input-output rule
\[
 x \xrightarrow{[\text{TEA} + \text{MINT}]} [\text{TEA} + \text{MINT}](x) = \text{TEA}(x) + \text{MINT}(x) \\
 = +49.28x^7 - 12.82x^4
\]

3. When the input function and the add-on function are specified by quantitative plots, we must stack pointwise the plot of the add-on function on top of the plot of the input function. In other words, on each input-level line, we “count the output of the add-on function from the output of the input function”.

a. A frequent situation will be one where we have the graph of a constant function and we want to add-on the graph of a dilation function.

**Example 19.** Let \( JILL \) be the constant function whose plot (blue) is:

and let \( NATE \) be the dilation function whose plot (green) we want to add-on:
i. We count the outputs of \textit{NATE} (green):

![Output Ruler](image)

ii. We recount them from the outputs of \textit{JILL} (blue):

![Output Ruler](image)

iii. This gives us the plot of \textit{JILL}+\textit{NATE} (black):

![Output Ruler](image)
8.6. ADD-ON OPERATORS

Altogether, we have:

\[
\begin{align*}
\text{Input} & \quad \text{Output} \\
\hline
+6 & \quad +6 \\
+2 & \quad +2 \\
+4 & \quad +4 \\
-6 & \quad -6 \\
-2 & \quad -2 \\
-4 & \quad -4 \\
0 & \quad 0 \\
\end{align*}
\]

b. Another frequent situation will be one where we have the graph of a dilation function and we want to add-on the graph of a squaring function.

**Example 20.** Let *RENEE* be the dilation function whose plot (blue) is:
and let \( JANE \) be the \textit{squaring} function whose plot (green) we want to add-on:

i. We count the outputs of \( JANE \) (green). For instance, the plot point for the input +2 is on the +1-output level line.

ii. We recount the outputs for \( JANE \) from the outputs of \( RENEE \) (blue). For instance, the plot point for the input +2 is now on the +2-output level line.
iii. This gives us the plot of $RENEE + JANE$ (black):

Altogether, we have: