A T-Sequence Language

Introduction

This section now assumes t-sequences have been defined and discussed in the early parts of the book. The section has been renamed to distinguish t-sequences from a language that describes them.

This section describes a language that can be used for constructing and manipulating t-sequences. The formalism that is introduced here allows t-sequences to be described precisely and compactly and provides conceptual focus. There is no mathematics, per se, just as these is no mathematics in the notation used for weaving drafts. Like draft notation, the t-sequence notation must be understood to be useful. The t-sequence language uses many “special” characters to describe operations concisely. Because of this, the t-sequence language may appear to be daunting. But the ideas are simple.

Terminology and Notational Conventions

The term sequence implies linear order. The terms in a sequence come one after another. There is a first term, a second term, and so on.

T-sequence terms may be explicit, as in 1, 4, 6, and so on, or they may be given as variables that take on different values in different contexts. Variables are indicated by lowercase italic letters, such as \(i, j\), and \(k\). Subscripts may be used to distinguish different term variables, such as \(i_1, i_2, j_3\), and so on.

Sequences may be given explicitly by enclosing their terms in square brackets, as in

\[
[1, 2, 3, 4, 3, 2, 3, 4, 5, 6, 7, 8, 7, 6, 5, 4]
\]

\[
[i_1, i_2, i_3, \ldots, i_6, j_3]
\]

Ellipses are used to indicate one or more terms in a sequence that are not given explicitly, as in

\[
[1, 2, 3, 4, 5, \ldots, 15, 16, 15, \ldots, 1]
\]
Variables are used to name sequences so that they can be referred to without specifying their terms. Sequence variables are indicated by uppercase italic letters, such as $S$, $T$, and $U$. Sequence variables also may have subscripts to distinguish different sequences in a common context. Examples are $S_1$, $S_2$, and $T_5$.

A specific sequence can be given a name. This is called assignment and is indicated by a colon followed by an equal sign, as in

$$S := [1, 2, 3, 4, 3, 2, 3, 4, 5, 6, 7, 8]$$

Then $S$ can be used to refer to this sequence without giving all the terms.

Two t-sequences are identical, denoted by $S = T$, if they are the same, term by term.

**Graphic Representation**

Patterns in t-sequences usually are easier to detect in graphical representations than by examining sequences of integers.

In this section, the values in grid plots increase upward:

In the grid plots used in this section, the axes usually are not marked, since such markings tend to distract the human visual system and interfere with pattern recognition.

The bottom row corresponds to the value 1 and the left column corresponds to the first value in the sequence.

**T-Sets and T-Numbers**

Sometimes it is useful to specify the particular shafts/treadles used in a t-sequence. This is called the *t-set* of the t-sequence. Braces are used to denote t-sets, as in $\{1, 2\}$.

In many cases, all the shafts and treadles are used, as illustrated in the example above. For example, the t-set for the plot above is

$$\{1, 2, 3, 4, 5, 6, 7, 8\}.$$

Finally, to avoid having to say shaft/treadle numbers repeatedly, *t-numbers* is used to cover both.
Sequence Metrics

There are three important properties associated with a sequence: its length, its minimum value (usually 1), and its maximum value, called its bound. These are given by functions whose names are lowercase Greek letters:

- $\lambda(S)$: length
- $\mu(S)$: minimum value
- $\gamma(S)$: maximum value (bound)

For the sequence $S$ in the preceding section, $\lambda(S) = 12$, $\mu(S) = 1$, and $\gamma(S) = 8$.

Extension

Concatenation

The most fundamental operation on t-sequences is appending one to another to form a longer one. This is called concatenation.

Concatenation of t-sequences is denoted by $S \mid T$ in which the result is a new sequence consisting of the terms of $S$ followed by the terms of $T$.

For example if

$S = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2]$ and $T = [1, 2, 3, 2, 1, 2, 3, 4, 5, 6, 7, 8]$ then

$S \mid T = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1, 2, 3, 2, 1, 2, 3, 4, 5, 6, 7, 8]$.

Here is the graphic representation:
The empty sequence $\Theta$ is the identity with respect to concatenation. That is,

$$(S \mid \Theta) = (\Theta \mid S) = S$$

for all $S$.

Often many $t$-sequences are concatenated, one after the other. To handle such cases conveniently, the notation

$$\{S_1, S_2, \ldots, S_n\}$$

denotes the concatenation of $S_1, S_2, \ldots, S_n$.

**Repetition**

*Repetition* is one of the most common operations on $t$-sequences. Repetition consists of concatenating a sequence with itself, perhaps several times.

Repetition is denoted by

$$S \times i$$

where $i$, an integer $\geq 0$, specifies the number of repetitions.

For example, if $S$ is as given in the preceding section, then

$$(S \times 3) = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1, 2, 3, 4, 5, 6, 5, 4, 3, 2]$$

Here is what it looks like as a grid plot:

$$(S \times 1) = S \text{ and } (S \times 0) = \Theta, \text{ the empty sequence for all } S.$$  

**Extension**

It sometimes is desirable to repeat a sequence to a specific length that is not
an even multiple of the length of the sequence.

This operation is called *extension* and is denoted by $S \Rightarrow i$, where $i \geq 0$ is the length of the new sequence.

For example, if $T$ is as given previously,

$$(T \Rightarrow 23) = [1, 2, 3, 2, 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 2, 1, 2, 3, 4, 5, 6, 7]$$

Here is what it looks like as a grid plot:

![Grid plot](image)

The extension length $i$ may be less than $\lambda(S)$, in which case truncation at the right occurs. For example,

$$(T \Rightarrow 9) = [1, 2, 3, 4, 5, 6, 5, 4, 3]$$

Of course, $(S \Rightarrow 0) = \Theta$ for all $S$.

### Duplicate Terms

Although concatenation and its two specialized forms, repetition and extension, are simple and fundamental operations, problems may arise if the last term in a sequence is the same as the first term in the sequence appended to it. Such duplicate terms may appear as undesirable artifacts of the concatenation and in some weaving contexts may cause structural problems.

For example, if

$$S = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1]$$

and duplicates at the boundaries of concatenation are not removed, $S \times 3$ would be as shown as:

![Grid plot](image)

If duplicate terms at the boundaries of concatenation are removed, however, the result is as shown here:
Whether or not duplicates that result from concatenation should be removed is a matter of context and not a property of the sequences involved. More often than not, duplicate removal is desired, so the operations of concatenation, repetition, and extension do that.

There are alternative versions of these operations that do not remove duplicates. These are denoted by $S \mathbin{|+} T$, $S \times i$, and $S \Rightarrow i$. For example, for the sequence $S$ given above,

$$(S \times 3) = [1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1, 1, 2, 3, 4, 5, 6, 5, 4, 3, 2, 1]$$

Note: Any duplicates within a sequence are unaffected by any of the concatenation operations.

**Runs**

Runs — integers in numerical sequence — occur very frequently in t-sequences.

A simple run consists of integers in order from a starting value to an ending value. If the starting value is less than the ending value, the run is up, else it is down. Here is an up run followed by a down run.

**Simple Runs**

There are two different kinds of runs that are composed of simple runs: connected runs and disconnected runs.
A T-Sequence Language

Disconnected Runs

Simple Runs

A simple run is denoted by

\[ i \rightarrow j \]

For example,

\[ (2 \rightarrow 8) = [2, 3, 4, 5, 6, 7, 8] \]

and

\[ (5 \rightarrow 1) = [5, 4, 3, 2, 1] \]

Connected Runs

In a connected run, runs go up and down (or down and up) between a beginning point, inflection points, and an ending point with no gaps. Beginning points, inflection points, and ending points collectively are called anchor points.

Connected runs can be constructed using simple runs and concatenation (with duplicate removal [1]). For example, the connected run shown in Figure 2 can be constructed by

\[ (1 \rightarrow 8) \mid (8 \rightarrow 2) \mid (2 \rightarrow 6) \mid (6 \rightarrow 1) \]

where parentheses are used to make the grouping of operations unambiguous.

Constructing a connected run using concatenation is unnecessarily cumbersome, however, since the run is completely described by its anchor points: the sequence \([1, 8, 2, 6, 1]\). This is emphasized in the figure below, where the anchor points are set off by color. The sequence of anchor points follows.

Highlighted Anchor Points
The operation for constructing a connected run from an anchor-point sequence \( S \) is denoted by \( \rightarrow S \). Note that the operator symbol is in prefix position before its operand as opposed to the same symbol used to denote simple runs, which is in infix position between its operands.

For example,

\[
\rightarrow [1, 8, 2, 6, 1]
\]

produces the same connected run as the concatenation of simple runs shown earlier.

**Disconnected Runs**

In a disconnected run, there are breaks in the numerical sequence of values. Disconnected runs can, of course, be constructed by concatenating simple runs. For example, the disconnected run shown in Figure 3 can be constructed by

\[
(1 \rightarrow 5) \mid (2 \rightarrow 6) \mid (3 \rightarrow 7) \mid (4 \rightarrow 8) \mid (6 \rightarrow 3) \mid (5 \rightarrow 2) \mid (4 \rightarrow 1)
\]

This also is unnecessarily cumbersome, since the runs are completely characterized by pairs of beginning and ending points: the sequences

\[
[1, 2, 3, 4, 6, 5, 4]
\]

and

\[
[5, 6, 7, 8, 3, 2, 1]
\]

This picture shows the disconnected runs with the end points highlighted. Following two pictures show the sequences of beginning and end points.
The operation for constructing disconnected runs from sequences of end points is denoted by $S \div T$, where $S$ is the sequence of beginning points and $T$ is the sequence of ending points.

For example, the disconnected runs shown in Figure 6 can be constructed by

$$[1, 2, 3, 4, 6, 5, 4] / [5, 6, 7, 8, 3, 2, 1]$$

It is worth noting that the sequences of beginning and ending points are concatenations of simple runs, so the same result can be obtained by

$$((1 \rightarrow 4) / (6 \rightarrow 4)) / ((5 \rightarrow 8) / (3 \rightarrow 1))$$

Although this form is more complicated than the one using explicit sequences, it reveals underlying structure in this disconnected run. Another form is perhaps more revealing:

$$((1 \rightarrow 4) / (5 \rightarrow 8)) / ((6 \rightarrow 4) / (3 \rightarrow 1))$$

This is the concatenation of a sequence of upward disconnected runs with a sequence of downward connected runs. This is, of course, evident in the grid plot.

**Symmetries**

Symmetry is one of the most powerful tools for producing aesthetically pleasing patterns. In t-sequences, the main use of symmetry is in concatenating a sequence and its reversal to produce a palindrome. Geometrically, reversal is horizontal reflection.

**Horizontal Reflection**

Horizontal reflection reverses the order of the terms in a sequence left to right. Horizontal reflection is denoted by $\leftrightarrow S$. For example, if
210

A T-Sequence Language

\[ S = [2, 4, 6, 8, 1, 3, 5, 7, 1, 2, 3, 1, 2, 3] \]

then

\[ \leftrightarrow S = [3, 2, 1, 3, 2, 1, 7, 5, 3, 1, 8, 6, 4, 2] \]

Vertical Reflection

It is also possible to reflect a sequence vertically by reversing the values, so that the largest becomes 1, the next-to-largest becomes 2, and so on. If this operation is denoted by \( \nu(i) \), then

\[ \nu(i) = \gamma(i) - i + 1 \]

The operation of vertical reflection is denoted by \( \Downarrow S \). For example, if

\[ S = \rightarrow[1, 3, 1, 6, 2, 8] \]

then

\[ \Downarrow S = \rightarrow[8, 6, 8, 3, 7, 1] \]
A palindrome is a sequence that is the same forwards and backwards. A palindrome is created by concatenating a sequence with its horizontal reflection (reversal):

\[ S \ construed S \]

This operation is so important that it has its own notation: \( \cap S \). For example, if

\[ S = \rightarrow[1, 3, 1, 6, 2, 8] \]

then

\[ \cap S = \rightarrow[1, 3, 1, 6, 2, 8, 2, 6, 1, 3, 1] \]

Note that the duplicate value at the middle is removed, as it is with concatenation. In the case that duplicate removal is not desired,

\[ S \ construed S \]

can be used.

“Palinforms”

The coined the word “palinforms” refers to concatenations of a sequence with one of its reflections other than the horizontal one.

There are two reflections other than horizontal that can be used to create palinforms: vertical and combined horizontal and vertical. Consider

\[ S = \rightarrow[1, 3, 1, 6, 2, 8] \]
Motifs along Paths

Some of the most interesting patterns in t-sequences come from placing a (usually) short sequence, called a motif, at successive points along a path. Here is an example:

The motif is

and the path is a straight draw:
The operation of placing a motif $M$ along a path $P$ is denoted by

$$M @ P$$

Placing a motif along a path is concatenation with an offset. Adjacent duplicates may arise, and as for other forms of concatenation [1], duplicate values at boundaries are removed by default. The operation

$$M @_2 P$$

does not remove duplicates that arise at boundaries.

It is worth noting that if the path is a constant sequence (all terms the same), a motif along the path simply is a repeat. In other words, the concept of a

**Summary**

- $S \upharpoonright T$ concatenation
- $|(S_1, S_2, \ldots, S_n)$ concatenation
- $S \times i$ repetition
- $S \Rightarrow i$ extension
- $i \rightarrow j$ simple run
- $\rightarrow S$ connected run
- $S / T$ disconnected runs
- $\leftrightarrow S$ horizontal reflection
- $\Cap S$ vertical reflection
- $\cap S$ palindrome formation
- $M @ P$ motif along a path

Concatenation operations without duplicate removal:

- $S \upharpoonright_2 T$ concatenation
- $|(S_1, S_2, \ldots, S_n)$ concatenation
- $S \times_2 i$ repetition
- $S \Rightarrow_2 i$ extension
- $M @_2 P$ motif along a path