Formal Definition of Table Operations
Version 1.2

Table operations are categorized into *primitive* and *derived* operations. A primitive operation is an operation that cannot be decomposed. A derived operation, on the other hand, is a composition of a sequence of primitive operations. This article describes the mathematical concept of each primitive table operation. For convenience, a table is represented as a matrix with homogeneous elements. For example, an $m \times n$ table T is

$$T = \begin{bmatrix} t_{11} & \dots & t_{1j} & \dots & t_{1n} \\ \dots & \dots & \dots & \dots & \dots \\ t_{i1} & \dots & t_{ij} & \dots & t_{in} \\ \dots & \dots & \dots & \dots & \dots \\ t_{m1} & \dots & t_{mj} & \dots & t_{mn} \end{bmatrix}, \tag{1}$$

where t_{ij} is the element at the i^{th} row and j^{th} column. As a convention, the letters A and B are used to denote source tables; R is used to denote a result table; lower case letters are used to denote the elements in a table (e.g., a_{ij}, b_{ij}, and r_{ij} are the elements in A, B, and R, respectively); and T_m and T_n denote the number of rows and columns of T, respectively.

There are ten primitive operations, namely `Select`, `Contract`, `Expand`, `Merge`, `Match`, `Group`, `Nesting`, `Cascade`, `Layout`, and `Caption`. These operations can be used to derive other operations. The operations `Select`, `Merge`, `Contract`, `Expand`, and `Match` perform table operations for flat tables (non-nested tables). The rest of the operations are `Group`, `Nesting`, `Cascade`, `Layout`, and `Caption`, which are designed to support nested report layouts, i.e., layouts with tables consisting of tables (subtables). The following sections give the mathematical definition of each primitive operation.

1 Select operation

A `Select` operation filters a subset of elements out of a source table. Given a table A, a list of rows with m elements $\langle \alpha_1, \alpha_2, \dots, \alpha_m \rangle$ and a list of columns with n elements $\langle \beta_1, \beta_2, \dots, \beta_n \rangle$ can be selected from A by the following `Select` operation:

$$R = \texttt{Select}_{\,rows\,=\,\langle \alpha_1,\, \alpha_2,\, ...,\, \alpha_m \rangle,\; columns\,=\,\langle \beta_1,\, \beta_2,\, ...,\, \beta_n \rangle}\,(A), \tag{2}$$

where R has $m \times n$ elements defined by $r_{ij} = a_{\alpha_i \beta_j}$. For example:

$$\texttt{Select}_{\,rows\,=\,\langle 1,\, 3 \rangle,\; columns\,=\,\langle 2,\, 4 \rangle} \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} \right) = \begin{bmatrix} x_{12} & x_{14} \\ x_{32} & x_{34} \end{bmatrix} \tag{3}$$

$$\texttt{Select}_{\ rows\ =\ \langle 2\rangle,\ columns\ =\ \langle 3\rangle} \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} \right) = \begin{bmatrix} x_{23} \end{bmatrix} \tag{4}$$

The lists of rows and columns can be expressed as either $\langle l_1, l_2, \ldots, l_n \rangle$ to denote a list of target rows or columns, $\langle * \rangle$ to denote all rows or all columns, or $\langle start, end\colon\ step \rangle$ to denote $\langle start, start + step, \ldots, end \rangle$. For example:

$$\texttt{Select}_{\ rows\ =\ \langle 1,\ 3\rangle,\ columns\ =\ \langle * \rangle} \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} \right) = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} \tag{5}$$

$$\texttt{Select}_{\ rows\ =\ \langle 1,\ 3\rangle,\ columns\ =\ \langle 1{,}4{:}2 \rangle} \left(\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \end{bmatrix} \right) = \begin{bmatrix} x_{11} & x_{13} \\ x_{31} & x_{33} \end{bmatrix} \tag{6}$$

2 Merge operation

Given two tables A and B and two positive integers dx and dy, a `Merge` operation transfers the location of B to $(dx + 1, dy + 1)$ and merges B with A. The `Merge` operation is defined as:

$$R\ =\ \texttt{Merge}_{\ x\ =\ dx,\ y\ =\ dy}\ (A,\ B), \tag{7}$$

where R has $\max(A_m, B_m + dy) \times \max(A_n, B_n + dx)$ elements defined by $r_{ij} = \alpha \cup \beta$, where α and β are

$$\alpha = \begin{cases} a_{ij} & \text{if } 1 \le i \le A_m \text{ and } 1 \le j \le A_n \\ \emptyset & \text{else} \end{cases} \tag{8}$$

$$\beta = \begin{cases} b_{ij} & \text{if } 1 \le i - dy \le B_m \text{ and } 1 \le j - dx \le B_n \\ \emptyset & \text{else} \end{cases} \tag{9}$$

For example, a table B can be put into the right side of another table A by the following operation:

$$\texttt{Merge}_{\ dx\ =\ 2,\ dy\ =\ 0} \left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \right) = \begin{bmatrix} x_{11} & x_{12} & y_{11} & y_{12} \\ x_{21} & x_{22} & y_{21} & y_{22} \end{bmatrix}, \tag{10}$$

Similarly, a table can be put on the left side, on the top, and on the bottom of another table by the merge operation.

Sometimes, it is useful to concatenate the elements of a table B to the elements of another table A (e.g., adding a suffix to every element of A). This can be done by a merge operation that produces elements with composite values. For example:

$$\texttt{Merge}_{\ dx\ =\ 0,\ dy\ =\ 0} \left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \right) = \begin{bmatrix} x_{11},\ y_{11} & x_{12},\ y_{12} \\ x_{21},\ y_{21} & x_{22},\ y_{22} \end{bmatrix}, \tag{11}$$

$$\texttt{Merge}_{\ dx\ =\ 1,\ dy\ =\ 0} \left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \right) = \begin{bmatrix} x_{11} & x_{12},\ y_{11} & y_{12} \\ x_{21} & x_{22},\ y_{21} & y_{22} \end{bmatrix}, \tag{12}$$

Note that "," is used to denote an element with multiple values. It is also possible that the resulting table contains empty elements. For example:

$$\texttt{Merge}_{\,dx\,=\,1,\;dy\,=\,1}\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}, \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}\right) = \begin{bmatrix} x_{11} & x_{12} & \emptyset \\ x_{21} & x_{22},\ y_{11} & y_{12} \\ \emptyset & y_{21} & y_{22} \end{bmatrix}, \tag{13}$$

3 Match operation

A `Match` operation can perform either 1D or 2D matching of two tables. A 1D matching is similar to a database join operation and a 2D matching is a 2-dimensional version of the matching. A `Match` operation is defined as:

$$R = \texttt{Match}_{\,dimension\,=\,d}\,(A, B), \tag{14}$$

where d is either 1 or 2. Let's look at 2D matching first. Table A defines a 2D layout using $A_m - 1$ vertical keys $(a_{21}, a_{31}, \dots, a_{A_m 1})$ and $A_n - 1$ horizontal keys $(a_{12}, a_{13}, \dots, a_{1A_n})$. Each row of table B contains a pair of keys (the first two columns) and a data cell (b_{i3}) to be matched (placed) to the 2D layout. The result R has $(A_m - 1) \times (A_n - 1)$ elements defined by

$$r_{ij} = \{b_{x3} : 1 \le x \le B_m \text{ and } b_{x1} = a_{(i+1)1} \text{ and } b_{x2} = a_{1(j+1)}\}.$$

For example:

$$\texttt{Match}_{\,dimension\,=\,2}\left(\begin{bmatrix} \times & k_{12} & k_{22} & k_{32} \\ k_{11} & \times & \times & \times \\ k_{21} & \times & \times & \times \end{bmatrix}, \begin{bmatrix} k_{11} & k_{12} & d_{13} \\ k_{21} & k_{22} & d_{23} \\ k_{21} & k_{32} & d_{33} \\ k_{11} & k_{22} & d_{43} \\ k_{21} & k_{12} & d_{53} \\ k_{11} & k_{32} & d_{63} \end{bmatrix}\right) = \begin{bmatrix} d_{13} & d_{43} & d_{63} \\ d_{53} & d_{23} & d_{33} \end{bmatrix}, \tag{15}$$

where a "$\times$" denotes a "don't care" value. In this example, the vertical key of d_{33} is k_{21}, which is located in the 3^{th} row of A. Therefore, in R, d_{33} is at the $(3-1)^{th}$ row. The horizontal key of d_{33} is k_{32}, which is located in the 4^{th} column of A. Therefore, in R, d_{33} is at the $(4-1)^{th}$ column. Overall, d_{33} is located at $(2, 3)$ in R.

After a 2D match operation, it is possible that the resulting table contains composite or empty values. For example:

$$\texttt{Match}_{\,dimension\,=\,2}\left(\begin{bmatrix} \times & k_{12} & k_{22} & k_{32} \\ k_{11} & \times & \times & \times \\ k_{21} & \times & \times & \times \end{bmatrix}, \begin{bmatrix} k_{11} & k_{12} & d_{13} \\ k_{11} & k_{22} & d_{23} \\ k_{21} & k_{32} & d_{33} \\ k_{21} & k_{32} & d_{43} \end{bmatrix}\right) = \begin{bmatrix} d_{13} & d_{23} & \emptyset \\ \emptyset & \emptyset & d_{33},\ d_{43} \end{bmatrix}, \tag{16}$$

For 1D matching, A is a table containing a column of keys and B is a table containing data that are to be matched to the location specified by the keys. The result R has $A_m \times (B_n - 1)$ elements defined by

$$r_{ij} = \{b_{x(j+1)} : 1 \le x \le B_m \text{ and } b_{x1} = a_{i1}\}.$$

For example:

$$\texttt{Match}_{\,dimension\,=\,1}\left(\begin{bmatrix} k_{11} \\ k_{21} \\ k_{31} \\ k_{41} \end{bmatrix}, \begin{bmatrix} k_{31} & d_{32} & d_{33} \\ k_{11} & d_{12} & d_{13} \\ k_{41} & d_{42} & d_{43} \\ k_{21} & d_{22} & d_{23} \end{bmatrix}\right) = \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \\ d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix}, \tag{17}$$

In this example, k_{11} is at the first row of A, and the key of $(d_{12}\ d_{13})$ is k_{11}. Therefore, in R, $(d_{12}\ d_{13})$ is moved to the first row.

After a 1D match operation, it is possible that the resulting table contains composite or empty values. For example:

$$\texttt{Match}_{\ dimension\ =\ 1}\left(\begin{bmatrix}k_{11}\\k_{21}\\k_{31}\end{bmatrix},\begin{bmatrix}k_{21}&d_{32}&d_{33}\\k_{11}&d_{12}&d_{13}\\k_{11}&d_{22}&d_{23}\end{bmatrix}\right)=\begin{bmatrix}d_{12},d_{22}&d_{13},d_{23}\\d_{32}&d_{33}\\\emptyset&\emptyset\end{bmatrix}, \tag{18}$$

4 Contract operation

A `Contract` operation contracts either multiple rows or columns into a single one. It is defined as:

$$R=\texttt{Contract}_{\ major\ =\ m,\ size\ =\ s}\ (A), \tag{19}$$

where m is either "*row*" or "*column*," and s is a positive integer indicating that every s rows or columns are to be contracted into a single row or column. If m is "*row*," R has $\lceil \frac{A_m}{s} \rceil \times A_n$ elements defined by

$$r_{ij}=\{a_{xj}:(i-1)\times s<x\leq i\times s\}.$$

For example:

$$\texttt{Contract}_{\ major\ =\ row,\ size\ =\ 2}\left(\begin{bmatrix}d_{11}&d_{12}&d_{13}\\d_{21}&d_{22}&d_{23}\\d_{31}&d_{32}&d_{33}\\d_{41}&d_{42}&d_{43}\end{bmatrix}\right)=\begin{bmatrix}d_{11},d_{21}&d_{12},d_{22}&d_{13},d_{23}\\d_{31},d_{41}&d_{32},d_{42}&d_{33},d_{43}\end{bmatrix}, \tag{20}$$

$$\texttt{Contract}_{\ major\ =\ row,\ size\ =\ 2}\left(\begin{bmatrix}d_{11}&d_{12}&d_{13}\\d_{21}&d_{22}&d_{23}\\d_{31}&d_{32}&d_{33}\\d_{41}&d_{42}&d_{43}\\d_{51}&d_{52}&d_{53}\end{bmatrix}\right)=\begin{bmatrix}d_{11},d_{21}&d_{12},d_{22}&d_{13},d_{23}\\d_{31},d_{41}&d_{32},d_{42}&d_{33},d_{43}\\d_{51}&d_{52}&d_{53}\end{bmatrix}, \tag{21}$$

If m is "*column*," R has $A_m \times \lceil \frac{A_n}{s} \rceil$ elements defined by $r_{ij}=\{a_{ix}:(j-1)\times s<x\leq j\times s\}$. For example:

$$\texttt{Contract}_{\ major\ =\ column,\ size\ =\ 2}\left(\begin{bmatrix}d_{11}&d_{12}\\d_{21}&d_{22}\\d_{31}&d_{32}\\d_{41}&d_{42}\\d_{51}&d_{42}\end{bmatrix}\right)=\begin{bmatrix}d_{11},d_{12}\\d_{21},d_{22}\\d_{31},d_{32}\\d_{41},d_{42}\\d_{51},d_{42}\end{bmatrix}, \tag{22}$$

$$\texttt{Contract}_{\ major\ =\ column,\ size\ =\ 2}\left(\begin{bmatrix}d_{11}&d_{12}&d_{13}\\d_{21}&d_{22}&d_{23}\\d_{31}&d_{32}&d_{33}\\d_{41}&d_{42}&d_{43}\\d_{51}&d_{52}&d_{53}\end{bmatrix}\right)=\begin{bmatrix}d_{11},d_{12}&d_{13}\\d_{21},d_{22}&d_{23}\\d_{31},d_{32}&d_{33}\\d_{41},d_{42}&d_{43}\\d_{51},d_{52}&d_{53}\end{bmatrix}, \tag{23}$$

A `Contract` operation can be used to produce many interesting derived operations. For example, a series of `Contract` and `Expand` operations (explained in the next section) can produce the effects of rotating and resizing tables. However, in practice, when producing a report, a `Contract` operation is seldom used alone. Most of the time, it is the derived operations that are used.

5 Expand operation

A composite value contains a number of *items*. We will define the notation for identifying items belonging to a composite value first. Suppose we have a composite value $v = v_1, v_2, \ldots, v_n$, we use the notation v[i] to indicate the i^{th} item of v, and $|v|$ to indicate the number of items that v has. For example, given the composite value $v = d_{11}, d_{12}$, then $v[1]$ is d_{11}, $v[2]$ is d_{12}, and $|v|$ is 2.

The `Expand` operation is the inverse of the `Contract` operation. It is defined as

$$R = \texttt{Expand}_{\,major\,=\,m,\ size\,=\,s}\,(A), \tag{24}$$

where m is either "*row*" or "*column*," and s is a positive integer indicating that every s items are to be expanded into a row or a column. We will discuss the case that $m = row$ first. Each row is expanded into a number of rows. Assume all $|a_{ij}|$ are the same (every element has the same number of items) and $|a_{ij}|$ is divisible by s. Let $x = |a_{ij}|/s$. Then, every row is expanded into x rows. R has $(x \times A_m)$ rows, and $(x \times A_m)$ by A_n elements. R has elements defined by

$$r_{ij} = \{a_{(\lceil i/x \rceil)j}[y] : (\lceil i/x \rceil - 1) \times s < y \leq \lceil i/x \rceil \times s\}.$$

For example:

$$\texttt{Expand}_{\,major\,=\,row,\ size\,=\,1}\left(\begin{bmatrix} d_{11}, d_{21} & d_{12}, d_{22} & d_{13}, d_{23} \\ d_{31}, d_{41} & d_{32}, d_{42} & d_{33}, d_{42} \end{bmatrix}\right) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \end{bmatrix} \tag{25}$$

$$\texttt{Expand}_{\,major\,=\,row,\ size\,=\,2}\left(\begin{bmatrix} d_{11}, d_{21}, d_{31}, d_{41} & d_{12}, d_{22}, d_{32}, d_{42} \\ d_{51}, d_{61}, d_{71}, d_{81} & d_{52}, d_{62}, d_{72}, d_{82} \end{bmatrix}\right) = \begin{bmatrix} d_{11}, d_{12} & d_{12}, d_{22} \\ d_{31}, d_{41} & d_{32}, d_{42} \\ d_{51}, d_{61} & d_{52}, d_{62} \\ d_{71}, d_{81} & d_{72}, d_{82} \end{bmatrix} \tag{26}$$

It is possible that not all $|a_{ij}|$ are the same and $|a_{ij}|$ is not necessarily divisible by s. The mathematical formula for this case is omitted. For example:

$$\texttt{Expand}_{\,major\,=\,row,\ size\,=\,1}\left(\begin{bmatrix} d_{11}, d_{21} & d_{12}, d_{22} & d_{13}, d_{23} \\ d_{31}, d_{41} & d_{32}, d_{42} & d_{33}, d_{42} \\ d_{51} & d_{52} & d_{53} \end{bmatrix}\right) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \\ d_{51} & d_{52} & d_{53} \end{bmatrix} \tag{27}$$

In case that $m = column$, each column is expanded into a number of columns. Assume all $|a_{ij}|$ are the same (every element has the same number of items) and $|a_{ij}|$ is divisible by s. Let $x = |a_{ij}| \div s$. Then, every column is expanded into x columns. R has $(x \times A_n)$ columns, and A_m by $(x \times A_n)$ elements. R has elements defined by

$$r_{ij} = \{a_{i(\lceil j/x \rceil)}[y] : (\lceil j/x \rceil - 1) \times s < y \leq \lceil j/x \rceil \times s\}.$$

For example:

$$\texttt{Expand}_{\,major\,=\,column,\ size\,=1}\left(\begin{bmatrix} d_{11}, d_{12} & d_{13}, d_{14} \\ d_{21}, d_{22} & d_{23}, d_{24} \\ d_{31}, d_{32} & d_{33}, d_{34} \\ d_{41}, d_{42} & d_{43}, d_{44} \\ d_{51}, d_{52} & d_{53}, d_{54} \end{bmatrix}\right) = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \\ d_{51} & d_{52} & d_{53} & d_{54} \end{bmatrix} \tag{28}$$

$$\texttt{Expand}_{\,major\,=\,column,\;size\,=2}\left(\begin{bmatrix} d_{11}, d_{12}, d_{13}, d_{14} & d_{15}, d_{16}, d_{17}, d_{18} \\ d_{21}, d_{22}, d_{23}, d_{24} & d_{25}, d_{26}, d_{27}, d_{28} \end{bmatrix}\right)$$
$$= \begin{bmatrix} d_{11}, d_{12} & d_{13}, d_{14} & d_{15}, d_{16} & d_{17}, d_{18} \\ d_{21}, d_{22} & d_{23}, d_{24} & d_{25}, d_{26} & d_{27}, d_{28} \end{bmatrix} \tag{29}$$

It is possible that not all $|a_{ij}|$ are the same and $|a_{ij}|$ is not necessarily divisible by s. The mathematical formula for this case is omitted. For example:

$$\texttt{Expand}_{\,major\,=\,column,\;size\,=1}\left(\begin{bmatrix} d_{11}, d_{12} & d_{13} \\ d_{21}, d_{22} & d_{23} \\ d_{31}, d_{32} & d_{33} \\ d_{41}, d_{42} & d_{43} \end{bmatrix}\right) = \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \\ d_{41} & d_{42} & d_{43} \end{bmatrix} \tag{30}$$

Like `Contract` operation, when producing a report, an `Expand` operation is seldom used alone. Most of the time, it is the derived operations that are used.

6 Group operation

The rest of the five primitive table operations are `Group`, `Nesting`, `Cascade`, `Layout`, and `Caption`, which are designed to support nested report layouts, i.e., layouts with tables consisting of tables (subtables).

A nested report layout is considered as a (root) table consisting of a list of tables arranged in a special order. A list of tables is expressed as either $[T_1, T_2, \cdots, T_n]$ or $[T_1\ T_2\ \cdots\ T_n]$. Note that $[T_1, T_2, \cdots, T_n]$ is a table with a single element, which is $T_1, T_2, \cdots T_n$. On the other hand, $[T_1\ T_2\ \cdots\ T_n]$, is a table with n elements, which are $T_1 \cdots T_n$.

By specifying a number of columns (or rows) as the keys, a `Group` operation splits a single table into a list of tables, which have the same key values. A list of subtables R produced by a `Group` operation is defined as

$$R = \texttt{Group}_{\,major\,=\,m,\;key\,=\,\langle \kappa_1,\,\kappa_2,\,...,\,\kappa_m\rangle}\,(A), \tag{31}$$

where m is either "*horizontal*" or "*vertical*," and *key* indicates the list columns or rows that are used as the keys.

We discuss the case that $m = horizontal$ first. The *key* specifies the columns that are used as the keys. In the resulting R, the rows with the same key values are grouped into the same subtable. For example,

$$\texttt{Group}_{\,major\,=\,horizontal,\;key\,=\,\langle 1\rangle}\left(\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{11} & d_{22} & d_{23} \\ d_{21} & d_{32} & d_{33} \\ d_{21} & d_{42} & d_{43} \end{bmatrix}\right) = \left[\begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix}\ \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix}\right] \tag{32}$$

In this example, the first column of A is $\begin{bmatrix} d_{11} \\ d_{11} \\ d_{21} \\ d_{21} \end{bmatrix}$ and is used as the key. Since the first and second rows have the same key value (d_{11}), they are grouped into one subtable. The resulting subtable does not include the key value. Therefore, the first subtable of R is $\begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix}$. Similarly, the third and fourth rows have the same key value (d_{21}) and are grouped into another subtable $\begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix}$.

The following example illustrates the case that multiple columns are use as the key.

$$\texttt{Group}_{\,major\,=\,horizontal,\ key\,=\langle 1,\,2\rangle} \left(\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{11} & d_{22} & d_{23} & d_{24} \\ d_{21} & d_{12} & d_{33} & d_{34} \\ d_{21} & d_{22} & d_{43} & d_{44} \\ d_{11} & d_{12} & d_{53} & d_{54} \\ d_{11} & d_{22} & d_{63} & d_{64} \\ d_{21} & d_{12} & d_{73} & d_{74} \\ d_{21} & d_{22} & d_{83} & d_{84} \end{bmatrix} \right) = \left[\begin{bmatrix} d_{13} & d_{14} \\ d_{53} & d_{54} \end{bmatrix} \begin{bmatrix} d_{23} & d_{24} \\ d_{63} & d_{64} \end{bmatrix} \begin{bmatrix} d_{33} & d_{34} \\ d_{72} & d_{74} \end{bmatrix} \begin{bmatrix} d_{43} & d_{44} \\ d_{83} & d_{84} \end{bmatrix} \right] \tag{33}$$

The source table A is a single (flat, non-nested) table in the above examples. What happens if the source table itself is a list of tables? In this case, the `Group` operation performs grouping for each subtable once. For example:

$$\texttt{Group}_{\,major\,=\,horizontal,\ key\,=\langle 1\rangle} \left(\left[\begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{11} & d_{22} & d_{23} \\ d_{21} & d_{32} & d_{33} \\ d_{21} & d_{42} & d_{43} \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{11} & e_{22} & e_{23} \\ e_{21} & e_{32} & e_{33} \\ e_{21} & e_{42} & e_{43} \end{bmatrix} \right] \right) = \left[\left[\begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} \right] \left[\begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \right] \right] \tag{34}$$

In case that $m = verital$, key specifies the rows that are used as the keys. In the resulting R, the columns with the same key values are grouped into the same subtable. For example:

$$\texttt{Group}_{\,major\,=\,vertical,\ key\,=\langle 1,\,2\rangle} \left(\begin{bmatrix} d_{11} & d_{12} & d_{11} & d_{12} \\ d_{21} & d_{22} & d_{21} & d_{22} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix} \right) = \left[\begin{bmatrix} d_{31} & d_{33} \\ d_{41} & d_{43} \end{bmatrix} \begin{bmatrix} d_{32} & d_{34} \\ d_{42} & d_{44} \end{bmatrix} \right] \tag{35}$$

Note that the inputs of the `Select`, `Merge`, `Match`, `Contract`, and `Expand` operations can also be lists of tables. In this case, the operations also perform transformations for each subtable.

7 Cascade operation

The `Cascade` operation can produce a list of tables by cascading two tables (lists of tables). Given two tables $A = [A_1\ A_2\ \cdots\ A_m]$ and $B = [B_1\ B_2\ \cdots\ B_n]$, the `Cascade` operations is defined as

$$R = \texttt{Cascade}(A, B), \tag{36}$$

where R is $[A_1\ A_2\ \cdots\ A_m\ B_1\ B_2\ \cdots\ B_n]$. For example,

$$\begin{aligned}\texttt{Cascade}\left(\begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix} & \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\right) \\ = \begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix} & \begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix} & \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\end{aligned} \tag{37}$$

8 Nesting operation

The `Nesting` operation can also produce a list of tables by cascading two tables (lists of tables) together. The differences between `Nesting` and `Cascade` operations are that the `Nesting` operation will increase the level of nesting (depth) by 1, and a *direction* option can be specified. The definition of `Nesting` operations is:

$$R = \texttt{Nesting}_{\ direction\ =\ d}(A, B), \tag{38}$$

where d is either "*horizontal*" or "*vertical*."

When $d = vertical$, the resulting R is $\begin{bmatrix}A\\ B\end{bmatrix}$. For example,

$$\begin{aligned}\texttt{Nesting}_{direction\ =\ vertical}\left(\begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix} & \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\right) \\ = \begin{bmatrix}\begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix}\end{bmatrix}\\ \\ \begin{bmatrix}\begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix} & \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\end{bmatrix}\end{aligned} \tag{39}$$

In the case that $d = horizontal$, the resulting $R = [A\ B]$. For example,

$$\begin{aligned}\texttt{Nesting}_{direction\ =\ horizontal}\left(\begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}\begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix}, \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\right) \\ = \begin{bmatrix}\begin{bmatrix}\begin{bmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{bmatrix} & \begin{bmatrix}a_{32} & a_{33}\\ a_{42} & a_{43}\end{bmatrix}\end{bmatrix} & \begin{bmatrix}\begin{bmatrix}b_{12} & b_{13}\\ b_{22} & b_{23}\end{bmatrix} & \begin{bmatrix}b_{32} & b_{33}\\ b_{42} & b_{43}\end{bmatrix}\end{bmatrix}\end{bmatrix}.\end{aligned} \tag{40}$$

Note that this result table (Eq. 40) is very similar to the result table of the `Cascade` example (Eq. 37). However, the structures of these two tables are different — the level of nesting of Eq. 40 is one higher than that of Eq. 37.

9 Layout operation

The structure of a list of tables can be re-arranged by a `Layout` operation. Given an input table $A = [A_1\ A_2\ \cdots\ A_m]$, the `Layout` operation is defined as

$$R = \texttt{Layout}_{layout\ =\ l,\ boundary\ =\ b}(A), \tag{41}$$

where l can be "*ReverseN*" or "*ZigZag*," and b is a positive integer. When $l = ReverseN$, the order of $[A_1\ A_2\ \cdots\ A_m]$ is rearranged into a reverse-N shape, i.e., the result R is a $b \times \lceil \frac{m}{b} \rceil$ table, and $R = \begin{bmatrix} A_1 & A_{b+1} & \cdots \\ A_2 & A_{b+2} & \cdots \\ \cdots & \cdots & \cdots \\ A_b & A_{2b} & \cdots \end{bmatrix}$. For example,

$$\begin{aligned} &\texttt{Layout}_{layout\,=\,ReverseN,\; boundary\,=\,2}\left(\begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} & \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \end{bmatrix}\right) \\ &= \begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} \\ \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \end{bmatrix} \end{aligned} \tag{42}$$

The result of a `Layout` operation may contain null elements when m is not a multiple of b. For example,

$$\begin{aligned} &\texttt{Layout}_{layout\,=\,ReverseN,\; boundary\,=\,3}\left(\begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} & \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \end{bmatrix}\right) \\ &= \begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \\ \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} & \emptyset \\ \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} & \emptyset \end{bmatrix} \end{aligned} \tag{43}$$

When $l = ZigZag$, the order of $[A_1\ A_2\ \cdots\ A_m]$ is rearranged into a ZigZag shape, i.e., the result R is a $\lceil \frac{m}{b} \rceil \times b$ table, and $R = \begin{bmatrix} A_1 & A_2 & \cdots & A_b \\ A_{b+1} & A_{b+2} & \cdots & A_{2b} \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$. For example,

$$\begin{aligned} &\texttt{Layout}_{layout\,=\,ZigZag,\; boundary\,=\,2}\left(\begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} & \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \end{bmatrix}\right) \\ &= \begin{bmatrix} \begin{bmatrix} d_{12} & d_{13} \\ d_{22} & d_{23} \end{bmatrix} & \begin{bmatrix} d_{32} & d_{33} \\ d_{42} & d_{43} \end{bmatrix} \\ \begin{bmatrix} e_{12} & e_{13} \\ e_{22} & e_{23} \end{bmatrix} & \begin{bmatrix} e_{32} & e_{33} \\ e_{42} & e_{43} \end{bmatrix} \end{bmatrix} \end{aligned} \tag{44}$$

10 Caption operation

The `Caption` operation is used to make a table as the caption of another table. Given two input tables A (source table) and B (caption table), the `Caption` operation is defined as

$$R = \texttt{Layout}_{direction\,=\,d,\; level\,=\,l}(A, B), \tag{45}$$

where d is either Top, $Botton$, $Left$, or $Right$, and l is a positive integer specifying the level (depth) of the caption to be applied. For example, the following operation puts the caption c_{11}

on top of the table A (in this case, the level-1 table of A is exactly A itself).

$$\texttt{Caption}_{direction\,=\,Top,\ level\,=\,1}\left(\begin{bmatrix}\begin{bmatrix}d_{12} & d_{13}\\ d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}d_{32} & d_{33}\\ d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}e_{12} & e_{13}\\ e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}e_{32} & e_{33}\\ e_{42} & e_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}c_{11}\end{bmatrix}\right) = \begin{bmatrix} c_{11} & \\ \begin{bmatrix}d_{12} & d_{13}\\ d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}d_{32} & d_{33}\\ d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}e_{12} & e_{13}\\ e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}e_{32} & e_{33}\\ e_{42} & e_{43}\end{bmatrix}\end{bmatrix} \tag{46}$$

The same operation can also be applied to level 2. In this case, the caption c_{11} is put on top of every subtables of A. For example,

$$\texttt{Caption}_{direction\,=\,Top,\ level\,=\,2}\left(\begin{bmatrix}\begin{bmatrix}d_{12} & d_{13}\\ d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}d_{32} & d_{33}\\ d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}e_{12} & e_{13}\\ e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}e_{32} & e_{33}\\ e_{42} & e_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}c_{11}\end{bmatrix}\right) = \begin{bmatrix}\begin{bmatrix}c_{11} & \\ d_{12} & d_{13}\\ d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}c_{11} & \\ d_{32} & d_{33}\\ d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}c_{11} & \\ e_{12} & e_{13}\\ e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}c_{11} & \\ e_{32} & e_{33}\\ e_{42} & e_{43}\end{bmatrix}\end{bmatrix}. \tag{47}$$

When $direction = Left$, the caption is put on the left of table A in the same way. For example,

$$\texttt{Caption}_{direction\,=\,Left,\ level\,=\,2}\left(\begin{bmatrix}\begin{bmatrix}d_{12} & d_{13}\\ d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}d_{32} & d_{33}\\ d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}e_{12} & e_{13}\\ e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}e_{32} & e_{33}\\ e_{42} & e_{43}\end{bmatrix}\end{bmatrix}, \begin{bmatrix}c_{11}\end{bmatrix}\right) = \begin{bmatrix}\begin{bmatrix}c_{11} & d_{12} & d_{13}\\ & d_{22} & d_{23}\end{bmatrix} & \begin{bmatrix}c_{11} & d_{32} & d_{33}\\ & d_{42} & d_{43}\end{bmatrix}\\ \\ \begin{bmatrix}c_{11} & e_{12} & e_{13}\\ & e_{22} & e_{23}\end{bmatrix} & \begin{bmatrix}c_{11} & e_{32} & e_{33}\\ & e_{42} & e_{43}\end{bmatrix}\end{bmatrix}. \tag{48}$$

The rest of the directions, *right* and *bottom*, are similar. We will omit the examples.