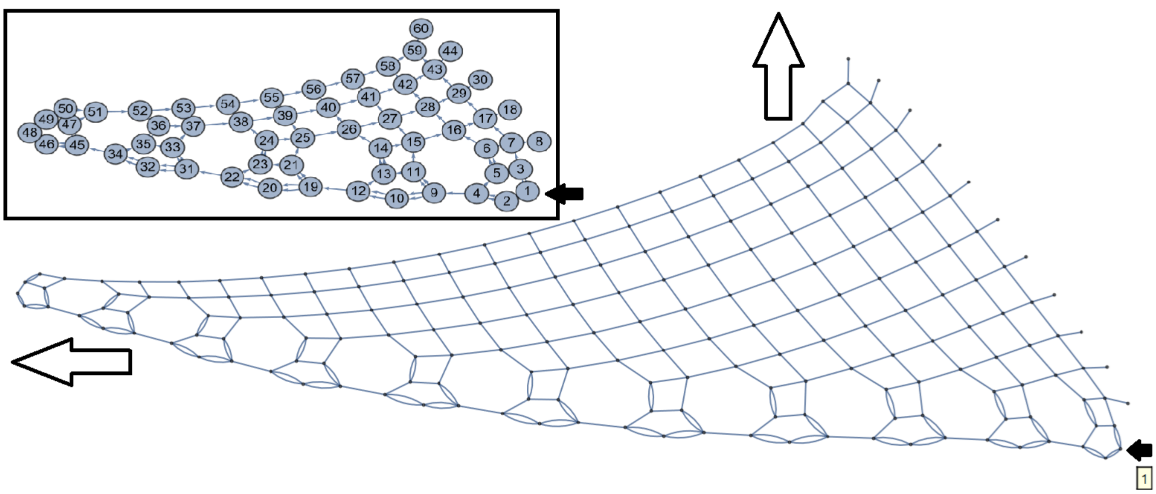
FAQ : Indexed Concatenation

An Indexed Concatenation is a mathematical notation designed to represent complex datasets in a more concise form. This is typically done by combining repeated elements of the dataset to create a more compact list.

Basics of using Indexed Concatenation to summarize a network.

Here is the example graph used in the SIMULTECH2025 Paper, presented by Rhys Sharpe in June 2025, in Bilbao, Spain. Clear patterns are visible:



Out[#]=

```
{1 → 2, 1 → 2, 1 → 3, 1 → 3, 2 → 4, 2 → 4, 3 → 5, 3 → 7, 4 → 5, 4 → 9, 5 → 6, 5 → 6,
 6 → 7, 6 → 16, 7 → 8, 7 → 17, 9 → 10, 9 → 10, 9 → 11, 9 → 11, 10 → 12, 10 → 12, 11 → 13,
 11 → 15, 12 → 13, 12 → 19, 13 → 14, 13 → 14, 14 → 15, 14 → 26, 15 → 16, 15 → 27,
 16 → 17, 16 → 28, 17 → 18, 17 → 29, 19 → 20, 19 → 20, 19 → 21, 19 → 21, 20 → 22,
 20 → 22, 21 → 23, 21 → 25, 22 → 23, 22 → 31, 23 → 24, 23 → 24, 24 → 25, 24 → 38, 25 → 26,
 25 → 39, 26 → 27, 26 → 40, 27 → 28, 27 → 41, 28 → 29, 28 → 42, 29 → 30, 29 → 43, 31 → 32,
 31 → 32, 31 → 33, 31 → 33, 32 → 34, 32 → 34, 33 → 35, 33 → 37, 34 → 35, 34 → 45, 35 → 36,
 35 → 36, 36 → 37, 36 → 52, 37 → 38, 37 → 53, 38 → 39, 38 → 54, 39 → 40, 39 → 55, 40 → 41,
 40 → 56, 41 → 42, 41 → 57, 42 → 43, 42 → 58, 43 → 44, 43 → 59, 45 → 46, 45 → 46, 45 → 47,
 45 → 47, 46 → 48, 46 → 48, 47 → 49, 47 → 51, 48 → 49, 48 → 61, 49 → 50, 49 → 50}
```

The data listed above represents the first 100 edges of the graph. 1 -> 2 means a connection between vertices 1 and 2. We note that the repetitive patterns in the graph are not at all visible in the edge list. But grouping edges by “out-vertex” and taking the differences of the vertex numbers for each edge gives the edge difference set list (EDSL), in which patterns reappear.

This shows how the graph edges correspond to sets in the EDSL. Even in the first few sets we begin to see some repetitions:

Out[∘]=

Edges:	{1 → 2, 1 → 2, 1 → 3, 1 → 3}	{2 → 4, 2 → 4}	{3 → 5, 3 → 7}	{4 → 5, 4 → 9}	{5 → 6, 5 → 6}	{6 → 7, 6 → 16}	{7 → 8, 7 → 17}	{}	{9 → 10, 9 → 10, 9 → 11, 9 → 11}	…
EDSL:	{1, 1, 2, 2}	{2, 2}	{2, 4}	{1, 5}	{1, 1}	{1, 10}	{1, 10}	{}	{1, 1, 2, 2}	…

For this graph, the first 100 entries in the EDSL are given below. Runs of duplicates are highlighted:

{{1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 5}, {1, 1}, {1, 10}, {1, 10}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 7}, {1, 1}, {1, 12}, {1, 12}, {1, 12}, {1, 12}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 9}, {1, 1}, {1, 14}, {1, 14}, {1, 14}, {1, 14}, {1, 14}, {1, 14}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 11}, {1, 1}, {1, 16}, {1, 16}, {1, 16}, {1, 16}, {1, 16}, {1, 16}, {1, 16}, {1, 16}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 13}, {1, 1}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {1, 18}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 15}, {1, 1}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {1, 20}, {}, {1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 17}, {1, 1}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {1, 22}, {}, {1, 1, 2, 2}, {2, 2}}

An EDSL of a network having some intrinsic pattern typically contains some duplicate sets, which provides an obvious first step in condensing the network down. In this EDSL we see 2 adjacent copies of {1,10}, 4 adjacent copies of {1,12}, 6 copies of {1,14}, 8 copies of {1,16}, etc., which we represent $\overset{2}{€}$[{1, 10}], $\overset{4}{€}$[{1, 12}], $\overset{6}{€}$[{1, 14}], $\overset{8}{€}$[{1, 16}], etc. By reducing the graph down like this, we can notice further patterns. Here is the EDSL above, with exact duplicates summarized, separated onto different lines to showcase the next step.

$$\Big\{\{1, 1, 2, 2\}, \{2, 2\}, \{2, 4\}, \{1, 5\}, \{1, 1\}, \overset{2}{€}[\{1, 10\}], \{\},$$
$$\{1, 1, 2, 2\}, \{2, 2\}, \{2, 4\}, \{1, 7\}, \{1, 1\}, \overset{4}{€}\{1, 12\}\Big], \{\},$$
$$\{1, 1, 2, 2\}, \{2, 2\}, \{2, 4\}, \{1, 9\}, \{1, 1\}, \overset{6}{€}[\{1, 14\}], \{\},$$
$$\{1, 1, 2, 2\}, \{2, 2\}, \{2, 4\}, \{1, 11\}, \{1, 1\}, \overset{8}{€}[\{1, 16\}], \{\}, \ldots\Big\}$$

After we do this, there is no further reduction that can be done based on exact duplicates of adjacent sets, but we can see that each 7-element row has a common pattern, with only 3 numbers changing in each row. In the first row these numbers are 5, 2, and 10; and in the second row these numbers are 7, 4, and 12; and in general, in the k^{th} row, they are $2k + 3$, $2k$, and $2k + 8$. Before moving on to the details of the Indexed Concatenation Notation, the fully reduced EDSL is listed below.

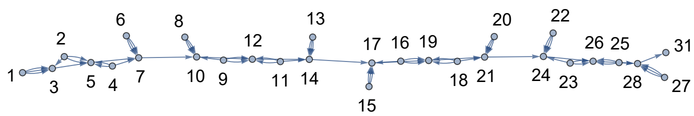
Out[∘]=

$$\left\{\underset{k=1}{\overset{18}{€}}\left[\{1, 1, 2, 2\}, \underset{j=1}{\overset{2}{€}}[\{2, 2j\}], \{1, 3+2k\}, \{1, 1\}, \underset{i=1}{\overset{2k}{€}}[\{1, 8+2k\}], \{\}\right]\right\}$$

Note that index variables, *i, j,* and *k*, have been introduced. Just as one would expect by analogy to an indexed summation, *i, j,* and *k* take on the initial value of 1 and are incremented until the specified final values are reached. In this case it’s 2k, 2 and 18, but the latter can be replaced by ∞. Assuming the pattern has been well established and will continue indefinitely, we have succeeded in summarizing the entire infinite network’s EDSL in one line! Similarly, most causal networks generated by a Sessie (Sequential Substitution System) Ruleset can be summarized by this notation.

Some other examples of I-CATs of networks

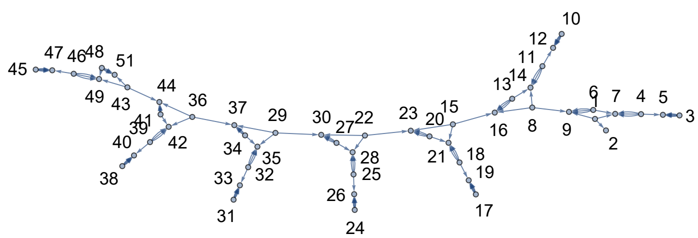
Out[•]=



Out[•]=

$$\left\{\overset{4}{\text{€}}[\{2, 2, 2\}, \{1, 3, 3\}, \{2\}, \{1, 1, 3\}, \{2\}, \{1, 1, 1\}, \{3\}]\right\}$$

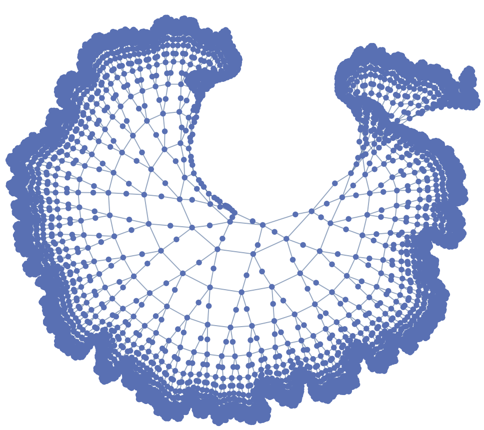
Out[•]=



Out[•]=

$$\left\{\overset{7}{\text{€}}\left[\{1, 6, 8\}, \{\}, \left\{\overset{4}{\text{€}}[2]\right\}, \overset{2}{\text{€}}\left[\left\{1, \overset{3}{\text{€}}[3]\right\}, \{\}\right]\right]\right\}$$

Out[•]=



Out[•]=

$$\left\{\underset{k=1}{\overset{10}{\text{€}}}\left[\{-1+2^k, 2^k, 1+2^k\}, \overset{1+2^k}{\text{€}}\left[\{1+2^k\}\right], \{1\}, \underset{j=1}{\overset{-1+2^k}{\text{€}}}\left[\{1, -1+2^k+j, 2^k+j\}\right]\right]\right\}$$

Examples of I-CATs summarizing patterns in decimal expansions:

Out[#]=

Fraction	Decimal Representation	Indexed Concatenation	Summation
1/3	0.33333…	$0.\overset{\infty}{\text{€}}3$	$0+\sum_{k=1}^{\infty}(3\cdot 10^{-k})$
1/9	0.11111…	$0.\overset{\infty}{\text{€}}1$	$0+\sum_{k=1}^{\infty}(1\cdot 10^{-k})$
1/11	0.0909090909…	$0.\overset{\infty}{\text{€}}(09)$	$0+\sum_{k=1}^{\infty}(9\cdot 10^{-k})$
1/7	0.142857142857142857142857…	$0.\overset{\infty}{\text{€}}(142857)$	$0+\sum_{k=1}^{\infty}(1428571\cdot 10^{-6\,k})$
1/13	0.076923076923076923…	$0.\overset{\infty}{\text{€}}(076923)$	$0+\sum_{k=1}^{\infty}(076923\cdot 10^{-6\,k})$
1/17	0.05882352941176470588…	$0.\overset{\infty}{\text{€}}(0588235294117647)$	$0+\sum_{k=1}^{\infty}(0588235294117647\cdot 10^{-16\,k})$
NA	0.010010001…	$0.\mathop{\text{€}}\limits_{i=1}^{\infty}((\mathop{\text{€}}\limits_{j=1}^{i}0)1)$	$0+\sum_{i=1}^{\infty}(1\cdot 10^{-i\,(i+3)/2})$
NA	0.112123123412345…	$0.\mathop{\text{€}}\limits_{i=1}^{\infty}((\mathop{\text{€}}\limits_{j=1}^{i}j)$	? (tricky when i≥10,100,etc.)
NA	0.1223334444…	$0.\mathop{\text{€}}\limits_{i=1}^{\infty}((\mathop{\text{€}}\limits_{j=1}^{i}i)$	? (tricky when i≥10,100,etc.)
NA	0.1123581321…	$0.\mathop{\text{€}}\limits_{i=1}^{\infty}(\phi^{n}-(-\phi)^{-n})/\sqrt{5})$	? (tricky when fib_n≥10,100,etc.)

Examples of I-CATs with Lists/Sequences

A few manual constructions:

$$\mathop{\text{€}}\limits_{i=1}^{3}\{i,\ i^{2}\} = \{1,\ 1\} \mathbin{+\!\!+} \{2,\ 4\} \mathbin{+\!\!+} \{3,\ 9\} = \{1,\ 1,\ 2,\ 4,\ 3,\ 9\}$$

$$\mathop{\text{€}}\limits_{i=1}^{5}\{i,\ 10-i\} =$$

$$\{1,\ 9\} \mathbin{+\!\!+} \{2,\ 8\} \mathbin{+\!\!+} \{3,\ 7\} \mathbin{+\!\!+} \{4,\ 6\} \mathbin{+\!\!+} \{5,\ 5\} = \{1,\ 9,\ 2,\ 8,\ 3,\ 7,\ 4,\ 6,\ 5,\ 5\}$$

$$\left\{\mathop{\text{€}}\limits_{i=1}^{3}\mathop{\text{€}}\limits_{j=1}^{3}[j\,i]\right\} = \left\{\mathop{\text{€}}\limits_{i=1}^{3}[i,\ 2\,i,\ 3\,i]\right\} =$$

$$\{[1,\ 2,\ 3] \mathbin{+\!\!+} [2,\ 4,\ 6] \mathbin{+\!\!+} [3,\ 6,\ 9]\} = \{1,\ 2,\ 3,\ 2,\ 4,\ 6,\ 3,\ 6,\ 9\}$$

$$\left\{\mathop{\text{€}}\limits_{i=1}^{3}\mathop{\text{€}}\limits_{j=1}^{3}\mathop{\text{€}}\limits_{k=1}^{3}[i\,j\,k]\right\} =$$

$$\left\{\mathop{\text{€}}\limits_{i=1}^{3}\mathop{\text{€}}\limits_{j=1}^{3}[ij,\ 2\,ij,\ 3\,ij]\right\} = \left\{\mathop{\text{€}}\limits_{i=1}^{3}([i,\ 2\,i,\ 3\,i] \mathbin{+\!\!+} [2\,i,\ 4\,i,\ 6\,i] \mathbin{+\!\!+} [3\,i,\ 6\,i,\ 9\,i]\right\}$$

$$= \left\{\mathop{\text{€}}\limits_{i=1}^{3}[i,\ 2\,i,\ 3\,i,\ 2\,i,\ 4\,i,\ 6\,i,\ 3\,i,\ 6\,i,\ 9\,i]\right\}$$

= {[1, 2, 3, 2, 4, 6, 3, 6, 9]
⧺ [2, 4, 6, 4, 8, 12, 6, 12, 18] ⧺ [3, 6, 9, 6, 12, 18, 9, 18, 27]}
= {1, 2, 3, 2, 4, 6, 3, 6, 9, 2, 4,
6, 4, 8, 12, 6, 12, 18, 3, 6, 9, 6, 12, 18, 9, 18, 27}

$$\left\{\mathop{€}_{i=1}^{4}\mathop{€}_{j=1}^{5}[j+i,\ j*i]\right\} = \left\{\mathop{€}_{i=1}^{4}[1+i,\ 1\,i,\ 2+i,\ 2\,i,\ 3+i,\ 3\,i,\ 4+i,\ 4\,i,\ 5+i,\ 5\,i]\right\}$$

{2, 1, 3, 2, 4, 3, 5, 4, 6, 5, 3, 2, 4, 4, 5, 6, 6, 8, 7, 10,
4, 3, 5, 6, 6, 9, 7, 12, 8, 15, 5, 4, 6, 8, 7, 12, 8, 16, 9, 20}

Out[#]=

Compressed Form of Pascal's Triangle:

$$\mathop{€}_{n=0}^{5}\left[\left\{\mathop{€}_{k=0}^{n}\left[\frac{n!}{k!\,(-k+n)!}\right]\right\}\right]$$

Expanded Form:

{1}
{1, 1}
{1, 2, 1}
{1, 3, 3, 1}
{1, 4, 6, 4, 1}
{1, 5, 10, 10, 5, 1}

Examples of I-CATs with Matrices

An unspecified *m* × *n* matrix:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{n} \\ a_{21} & a_{22} & \cdots & a_{2\,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

IC form :

$$\mathop{€}_{i=1}^{m}\left\{\mathop{€}_{j=1}^{n}\{a_{ij}\}\right\}$$

Cantor’s enumeration of the rational numbers:

(This section is a near verbatim excerpt from the SIMULTECH 2025 paper.)

A famous matrix was used in the first enumeration of the rationals, published in (Cantor, 1874). Georg Cantor, the father of transfinite mathematics, published his proof of the countability of the rationals in 1874, over 150 years ago. One might presume that in the intervening time someone would have come up with a concise, mathematically precise way to describe the process. Sadly, that expectation would be incorrect. Words to describe Cantor’s method are easy:

1. Imagine an infinite table in which the first element of the first row is 1/1, and then moving one column to the right always increases the numerator by 1, while moving one row down increases the denominator by 1.

2. Following any (infinite) row or column would mean never getting to the next one; instead, we traverse the array one (finite) diagonal at a time, starting from the upper left corner. Note that for all elements n/d on a given diagonal, $n+d$ is a constant, and in fact, on diagonal number *diag*, $n+d = diag + 1$.

Cantor actually used a “snake-like” traversal, winding back-and-forth on alternate diagonals, but for our purposes a “trace and retrace” pattern is preferable.

Figure 2: A traversal producing Cantor’s enumeration of fractions.

Out[#]=

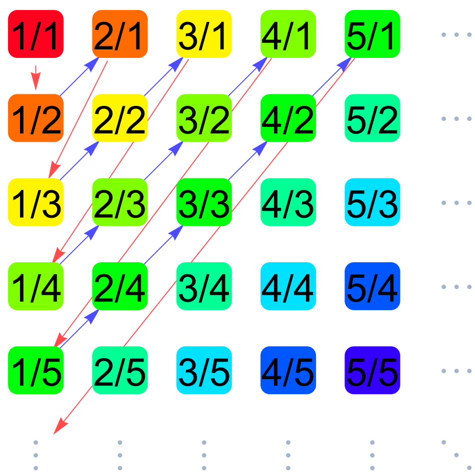


Figure 2 first appeared in (Caviness, 2011), and the code to generate it was updated in (Nachbar, 2023).

Note: The set of all fractions in the table is $\left\{\frac{p}{q} \,\middle|\, p, q \in \mathbb{Z}^+\right\}$, with $\frac{p}{q}$ appearing in row q, column p.

The following IC constructs the diagonals in the order Cantor visits them, but traverses each diagonal from left-to-right (by increasing numerator), as shown by the small blue arrows, before advancing to the next diagonal (longer red arrows). We do this by two nested IC objects, the outer specifying the diagonal $diag$, the inner giving the n^{th} element on the diagonal. If the nested list structure is not desired, the inner set of curly braces could be replaced by the disappearing square brackets, generating sequences that are simply spliced into a single list.

Out[#]=

List of diagonals:

Out[#]//DisplayForm=

$$\left\{\left\{\frac{1}{1}\right\}, \left\{\frac{1}{2}, \frac{2}{1}\right\}, \left\{\frac{1}{3}, \frac{2}{2}, \frac{3}{1}\right\}, \left\{\frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}\right\}, \left\{\frac{1}{5}, \frac{2}{4}, \frac{3}{3}, \frac{4}{2}, \frac{5}{1}\right\}, \ldots\right\}$$

Out[#]=

IC reduction:

$$\underset{diag=1}{\overset{\infty}{\text{€}}} \left\{ \underset{n=1}{\overset{diag}{\text{€}}} \left\{ \frac{n}{1+diag-n} \right\} \right\}$$

This enumeration cannot be easily produced by conventional mathematical notation, such as using set-builder notation. For example,

$\left\{\frac{p}{q} \mid p, q \in \mathbb{Z}^{+}\right\}$

produces all fractions, but loses Cantor's diagonal ordering.

$\left\{\frac{n}{diag-n+1} \mid diag \in \mathbb{Z}^{+},\ n \in \{1,\ \ldots,\ diag\}\right\}$

again produces all fractions, but assumes the user will not sort the resulting list (i.e., non-standard treatment for sets), and has no interest in grouping entries by diagonal.

The Indexed Concatenation form practically writes itself, while set-builder notation requires significant mathematical gymnastics to produce the same result.

The Indexed Concatenation Notation can represent infinite matrices of quantum mechanics – a, $a^{\dagger}$:

Out[ ]=

$$\begin{pmatrix} 0 & \sqrt{1} & 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & \sqrt{3} & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & \sqrt{4} & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \sqrt{5} & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{6} & \ldots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \ldots \\ \sqrt{1} & 0 & 0 & 0 & 0 & 0 & \ldots \\ 0 & \sqrt{2} & 0 & 0 & 0 & 0 & \ldots \\ 0 & 0 & \sqrt{3} & 0 & 0 & 0 & \ldots \\ 0 & 0 & 0 & \sqrt{4} & 0 & 0 & \ldots \\ 0 & 0 & 0 & 0 & \sqrt{5} & 0 & \ldots \\ 0 & 0 & 0 & 0 & 0 & \sqrt{6} & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Out[ ]=

$$\left\{\overset{\infty}{\underset{k\vDash 1}{€}}\left[\left\{\overset{k}{€}[0],\ \sqrt{k},\ \overset{\infty}{€}[0]\right\}\right]\right\},\quad \left\{\overset{\infty}{\underset{k\vDash 1}{€}}\left[\left\{\overset{k-2}{€}[0],\ \sqrt{k-1},\ \overset{\infty}{€}[0]\right\}\right]\right\}$$

Read the rows of a this way:

On row k, start with a sequence of 0s (k copies), followed by $\sqrt{k}$, then an infinite sequence of 0s.

Note: $\left\{\overset{5}{€}[0]\right\} = \left\{\overset{5}{\underset{j\vDash 1}{€}}[0]\right\} == \{0,\ 0,\ 0,\ 0,\ 0\}$, (i.e., a subsequence of 5 copies of 0 inserted into the list (an ordered multiset).

Read the rows of $a^{\dagger}$ this way:

On row k, start with a sequence 0s ($k-2$ copies), followed by $\sqrt{k-1}$, then an infinite sequence of 0s.

Tricky point: when $k = 1$ or 2, what does it mean to concatenate $k-2$ copies of 0? (!)

Let's break this down.

Although the index variable isn't explicitly given, $\overset{k-2}{€}[0]$ actually means something like $\overset{k-2}{\underset{j\vDash 1}{€}}[0]$, so what are the meanings of $\overset{0}{\underset{j\vDash 1}{€}}[0]$ and $\overset{-1}{\underset{j\vDash 1}{€}}[0]$? The first clearly means a sequence of 0 copies of 0, so it's an empty subsequence, and adds no entries to that row. But we also consider $\overset{-1}{\underset{j\vDash 1}{€}}[0]$ to be an empty subsequence, since "j starts at 1 and goes up to -1" shouldn't even "start the loop", we've already passed the termination condition!

Other work:

- How to "reduce" (or compress) a network list without using ToNetDifferenceSets? *Unknown!*
  - ○ We need this in order to continue on the work started by Jeanna Toulouse and Chris Trana
  - ○ How will these two reductions compare? Can we move directly from one to the other?
  - ○ BREAKTHROUGH!

An algorithm has been developed to convert an EDSL I-Cat (even multiply nested!) into an I-Cat of the original graph’s edges.

The EDSL I–Cat is easier to generate, and then we use it to come back to the “Trana Form” of the graph’s edges. Topic of a future paper.

- Can the dimension of the network be read directly from the IC reduced form? *Unknown!*
- Can IC compress the complete "Evolution" or even the TEvolution (tagged evolution) of the Sessie? *Unknown!*
- Can IC be applied in general to repeating strings? *Unknown, but probably.*
- Can IC be applied to musical notation? "What is the IC reduced form of Beethoven's Ninth?" (Or of one of Bach's Concertos?!)

Perhaps more fundamentally,

- How can the ReduceSetList algorithm be made more reliable?
  - ○ Last year's version only worked about half the time, this new version does better (~70%), but needs improvement still.
- We need to step through the algorithm in our meeting, and ask everyone to suggest improvements.
  - ○ NOT the code, just the logic of the steps involved.
  - ○ **A couple nice demos showing the steps of the reduction algorithm are desperately needed.**

FAQ Sessie Intro, 2025.09.09, Kenneth Caviness and Colton Edelbach