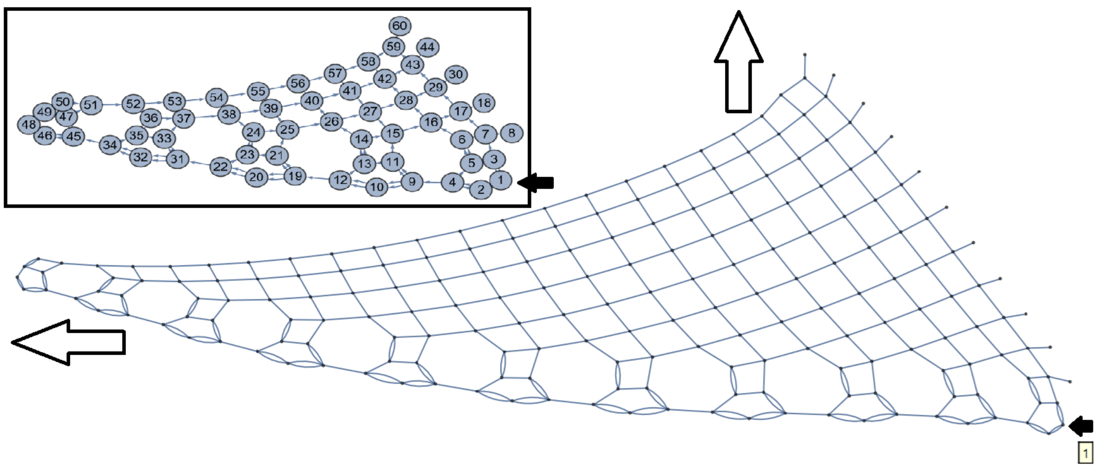
FAQ : Indexed Concatenation

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.



Out[14]=

```
{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. Taking the differences of the vertex numbers for each edge gives the edge difference set list (EDSL).

This shows how the edges correspond to sets in the EDSL.

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.

In[ ]:= **EDSL1⟦1 ;; 100⟧**

```
{{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 4 adjacent copies of {1,12}, 6 copies of {1,14}, and 8 copies of {1,16} we could represent these duplications by €^4[{1, 12}], €^6[{1, 14}], €^8[{1, 16}]. By reducing the graph down like this, we can notice other cases. By applying this to our EDSL above, we get the condensed version of the EDSL.

{{1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 5}, {1, 1}, €^{2}[{1, 10}], {},
{1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 7}, {1, 1}, €^{4}{1, 12}], {},
{1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 9}, {1, 1}, €^{6}[{1, 14}], {},
{1, 1, 2, 2}, {2, 2}, {2, 4}, {1, 11}, {1, 1}, €^{8}[{1, 16}], {}, …}

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 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.

In[15]:= **IC1**

Out[15]=

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

Note that the 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 its 2k, 2 and ∞. Assuming the pattern has been well established and will continue indefinitely, we have succeeded in summarizing the entire infinite network's EDSL in one to two lines! Similarly, most causal networks generated by a Sessie Ruleset can be summarized by this notation.

Still unknown:

• How to "reduce" (or compress) a network list without using ToNetDifferenceSets.
  ○ 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?
• Can the dimension of the network be read directly from the IC reduced form? (Either one)
• Can IC compress the complete "Evolution" or even the TEvolution (tagged evolution) of the Sessie?
• Can IC be applied in general to repeating strings?
• 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, 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 are desperately needed.

FAQ Sessie Intro, 2025.4.2, Kenneth Caviness and Colton Edelbach