Abstracting spreadsheet data flow through hypergraph redrawing

David Birch[1], Nicolai Stawinoga[1], Jack Binks[2], Bruno Nicoletti[2], Paul Kelly[1]

[1]Imperial College London

[2] Filigree Technologies, London

InnovateUK
Grant 104141

What do you see?

	A	B	C	D
1	**Agent**	**Sales (sqft)**	**Sales(m2)**	**Bonus**
2	Fred	5221	485.2952	100
3	Dave	3872	359.9048	0
4	Bob	3651	339.3627	0
5				
6		**Total Sales**	1184.563	
7		**Total Bonus**	100	

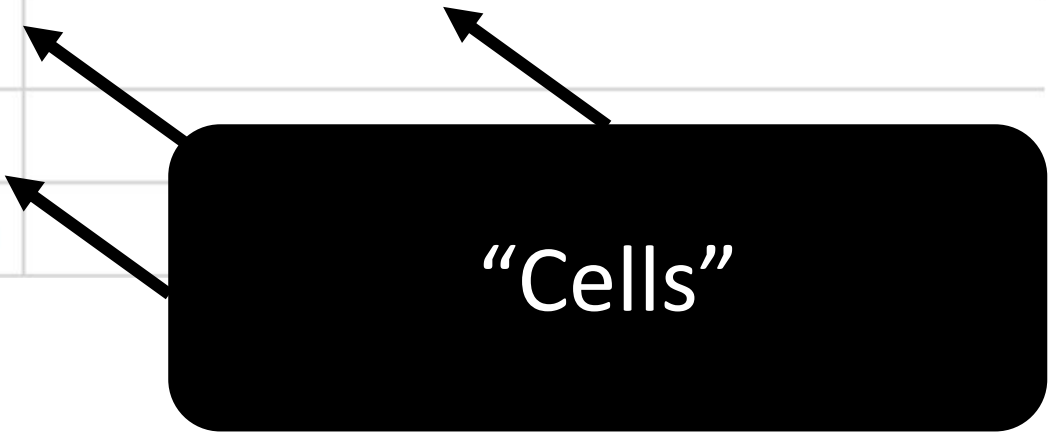
- Words?
- Numbers?
- Maths?
- A table?
- A grid?

How about a prison?

	A	B	C	D
1	**Agent**	**Sales (sqft)**	**Sales(m2)**	**Bonus**
2	Fred	5221	485.2952	100
3	Dave	3872	359.9048	0
4	Bob	3651	339.3627	0
5				
6		Total Sales	1184.563	
7		Total Bonus	100	

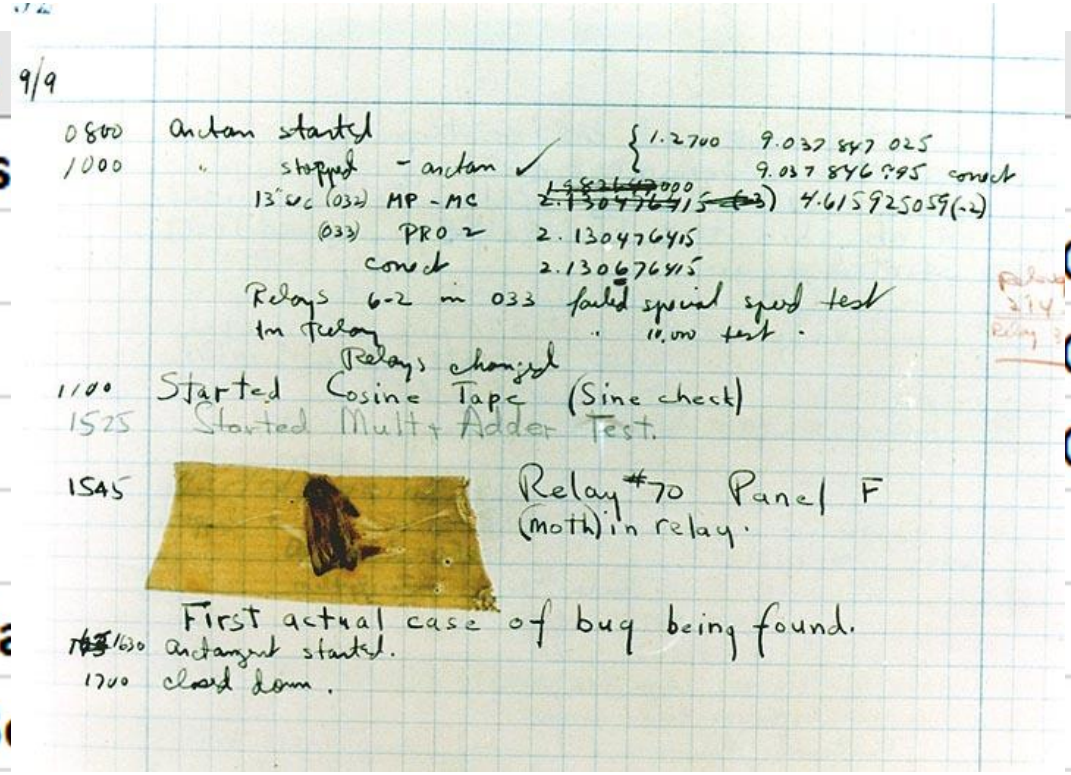
Many friendly numbers and mathematical things are imprisoned :*(

	A	B	C	D
1	**Agent**	**Sales (sqft)**	**Sales(m2)**	**Bonus**
2	Fred	5221	=B2/3.28/3.28	=IF(C2>AVERAGE(C$2:C$4),100,0)
3	Dave	3872	=B3/3.28/3.28	=IF(C3>AVERAGE(C$2:C$4),100,0)
4	Bob	3651	=B4/3.28/3.28	=IF(C4>AVERAGE(C$2:C$4),100,0)
5				
6		**Total Sales**	=SUM(C2:C4)	
7		**Total Bonus**	=SUM(D2:D4)	



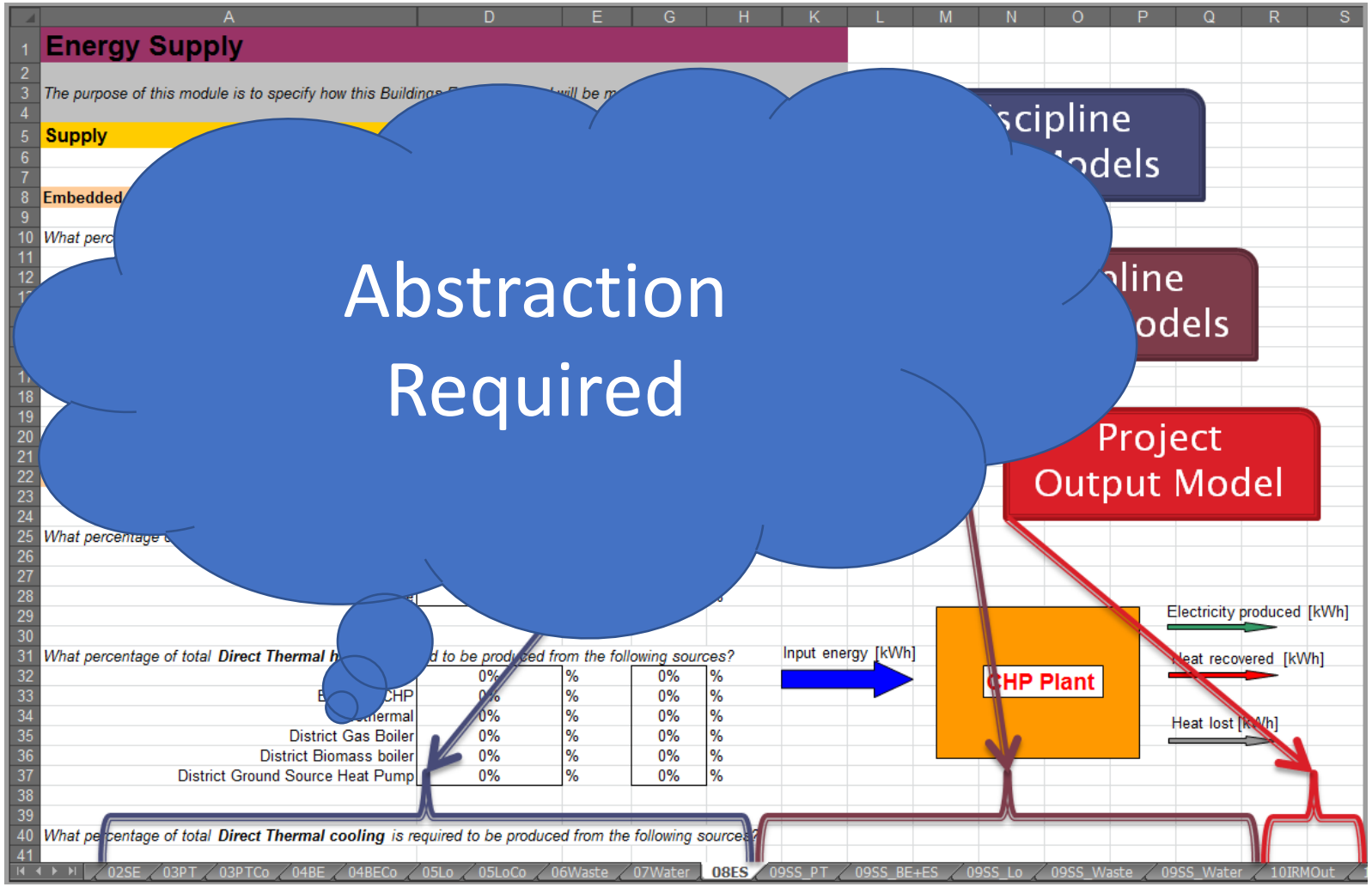
We know there are other things hiding inside this prison

	A	B	D
1	**Agent**	**Sales (s**	
2	Fred	5221	GE(C$2:C$4),100,0)
3	Dave	3872	GE(C$2:C$4),100,0)
4	Bob	3651	GE(C$2:C$4),100,0)
5			
6		**Total Sa**	
7		**Total B**	



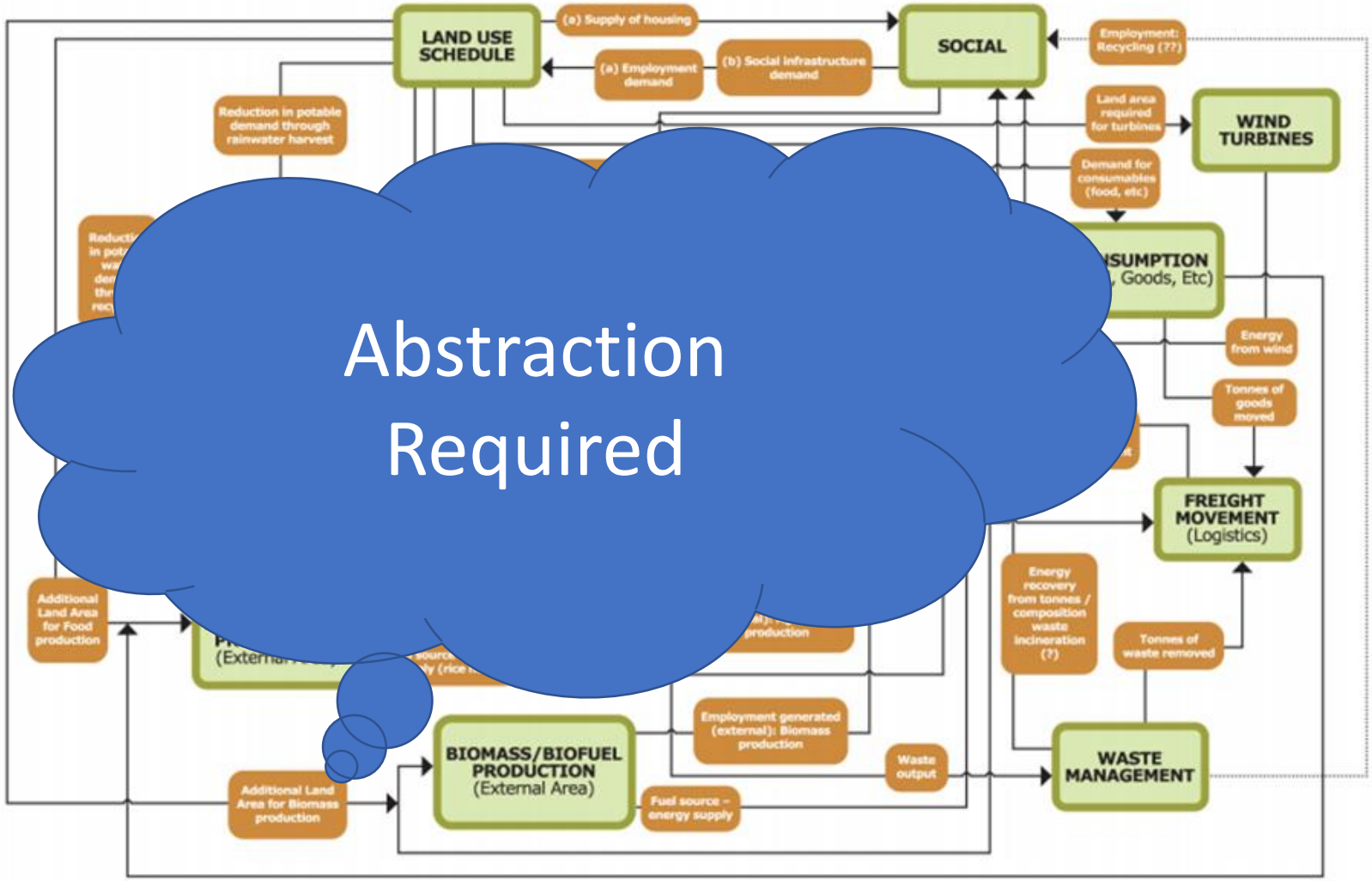
The First "Computer Bug" Moth found trapped between points at Relay # 70, Panel F, of the Mark II Aiken Relay Calculator while it was being tested at Harvard University, 9 September 1947 (public domain)

So how do we know what this prison does?



Page J., Grange N. and Kirkpatrick N. The integrated resource management (IRM) model - guidance tool for sustainable urban design. In *25th Conference on Passive and Low Energy Architecture, PLEA08*, October 2008

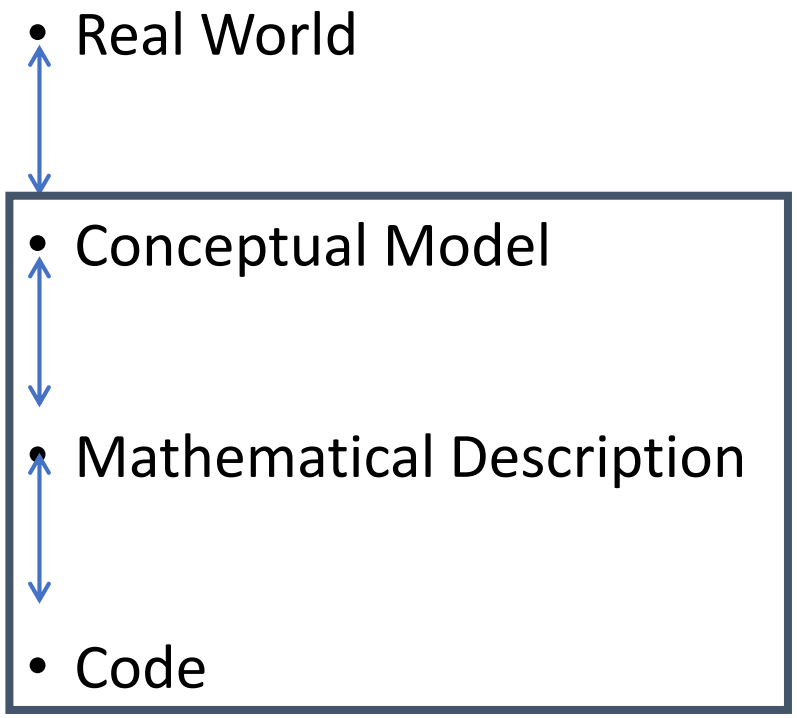
Maybe we have a diagram or mental model?



Page J., Grange N. and Kirkpatrick N. The integrated resource management (IRM) model - guidance tool for sustainable urban design. In *25th Conference on Passive and Low Energy Architecture, PLEA08*, October 2008

Scientific Model Making

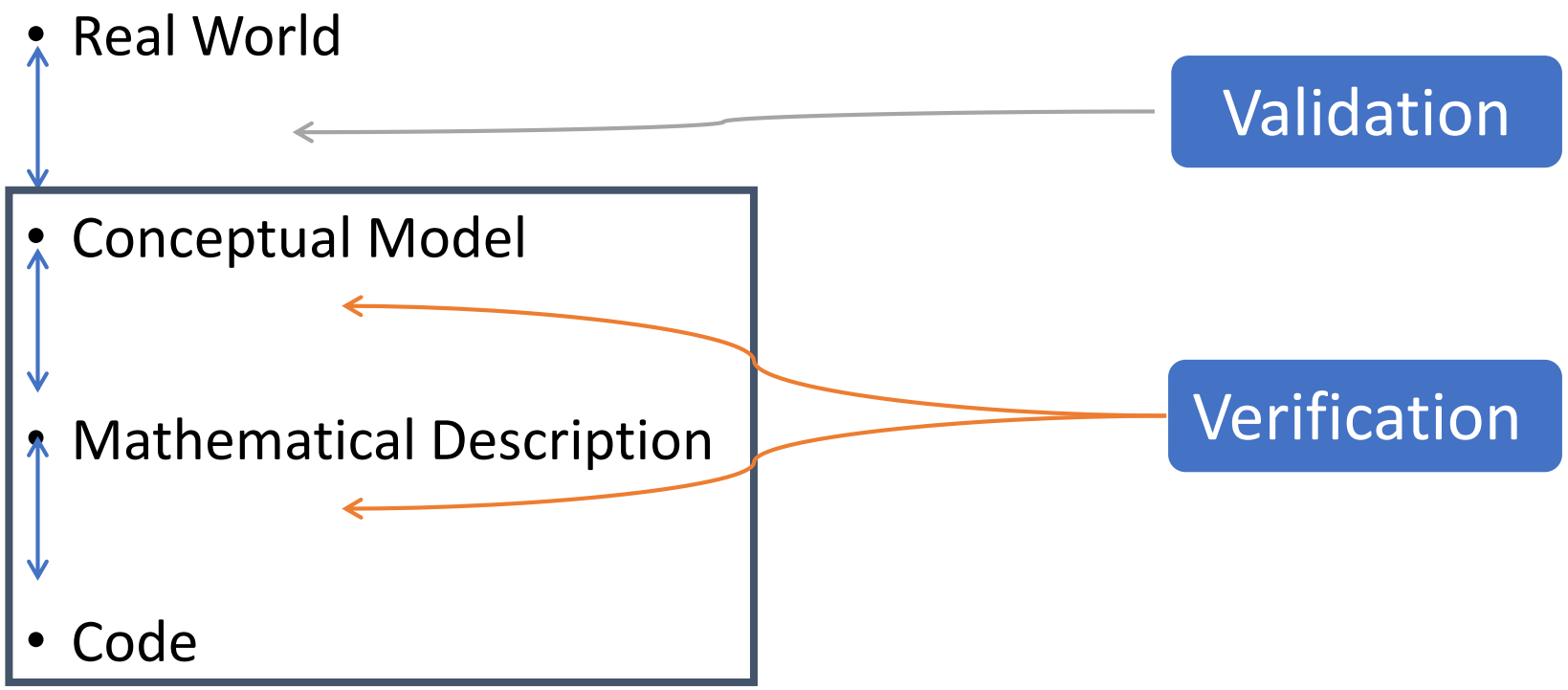
Scientific Model



http://www.easterbrook.ca/steve/?p=2030

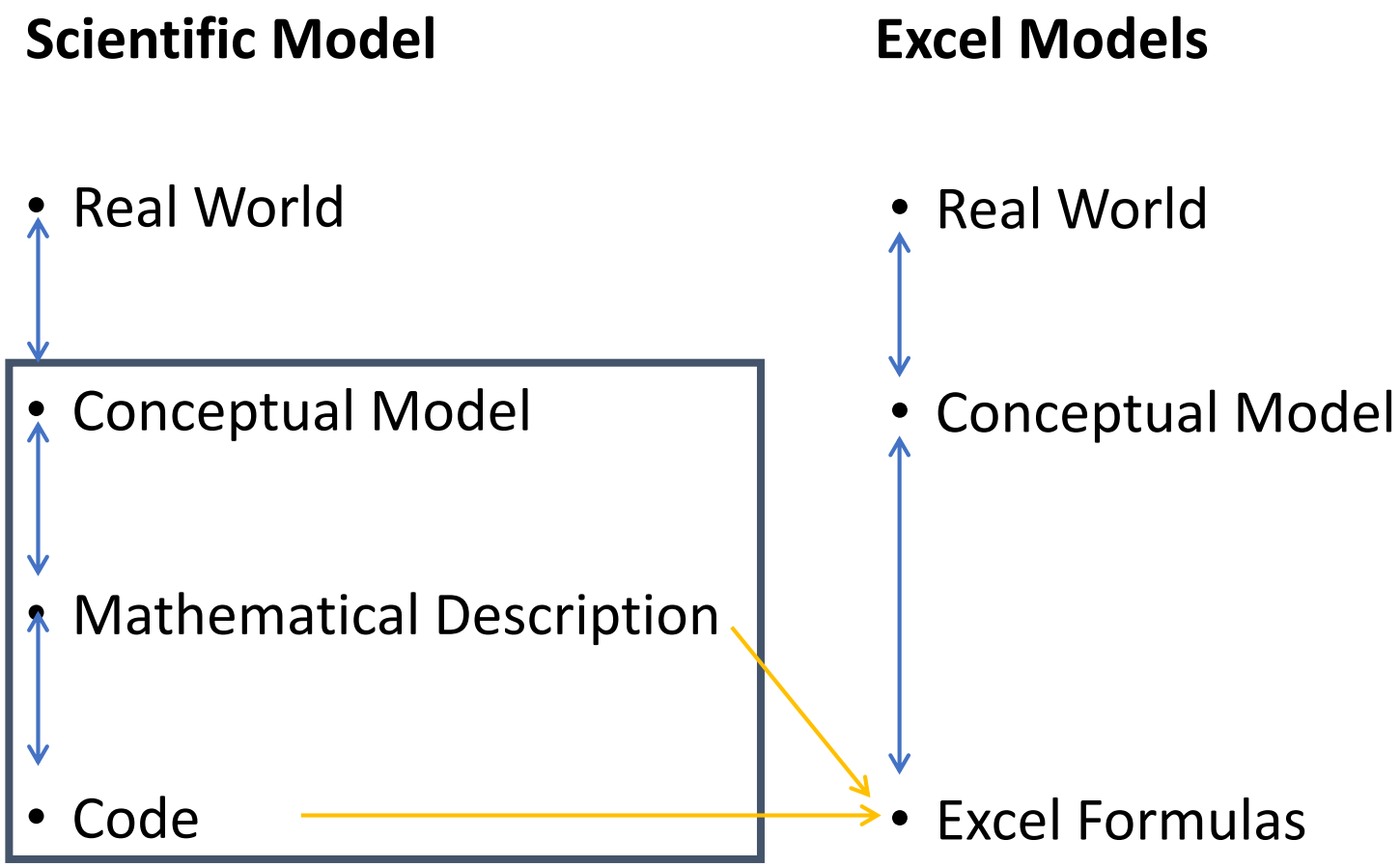
Verification / Validation

Scientific Model

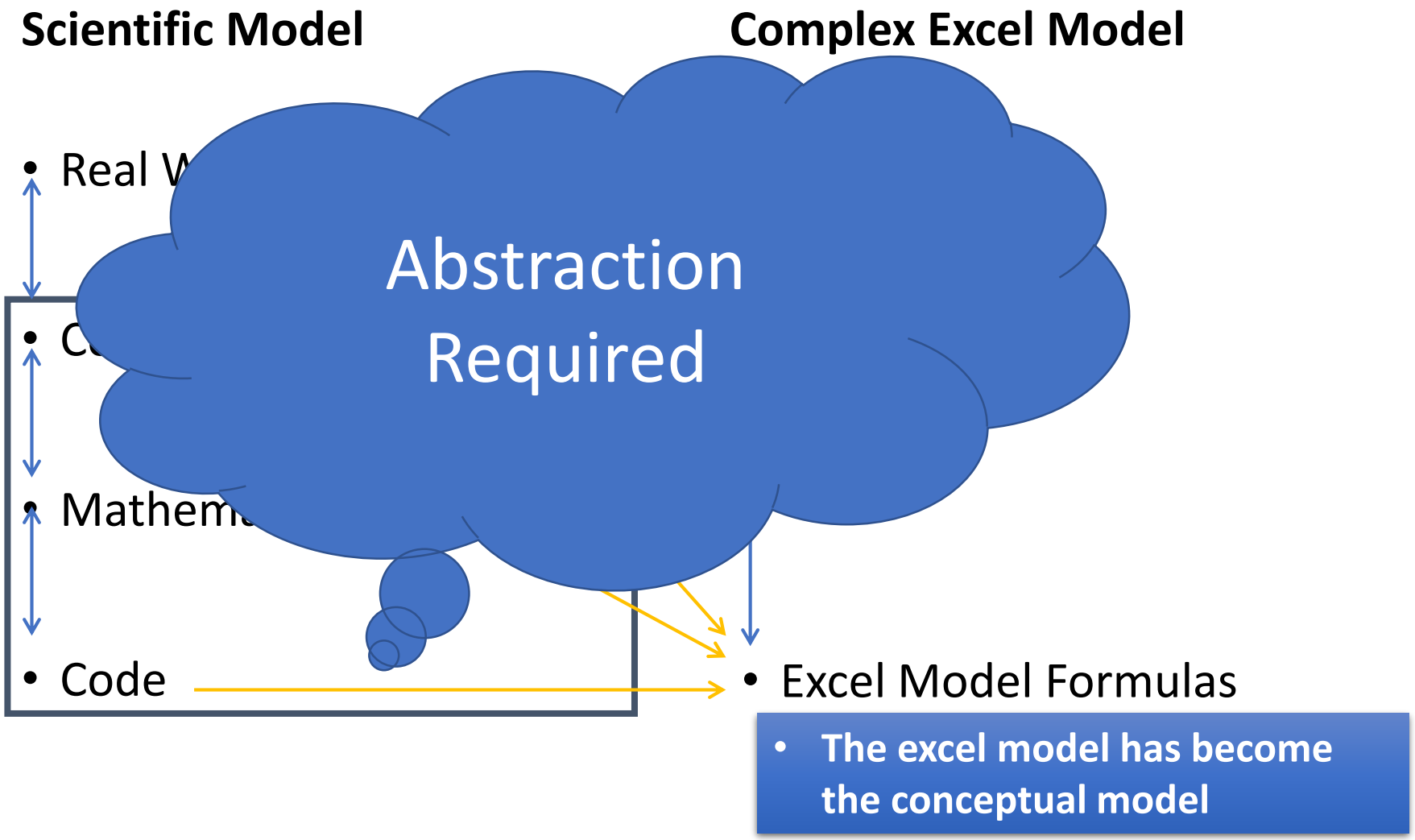


http://www.easterbrook.ca/steve/?p=2030

Spreadsheets



Spreadsheets



Agenda…

1. Just what is a “Cell”?

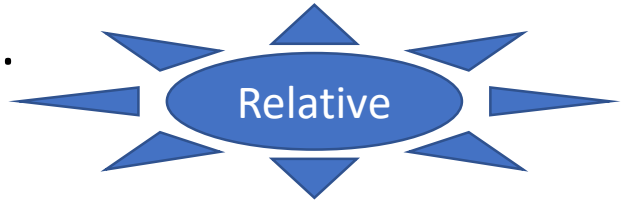
2. How we can abstract “Cells” by redrawing their cells walls?

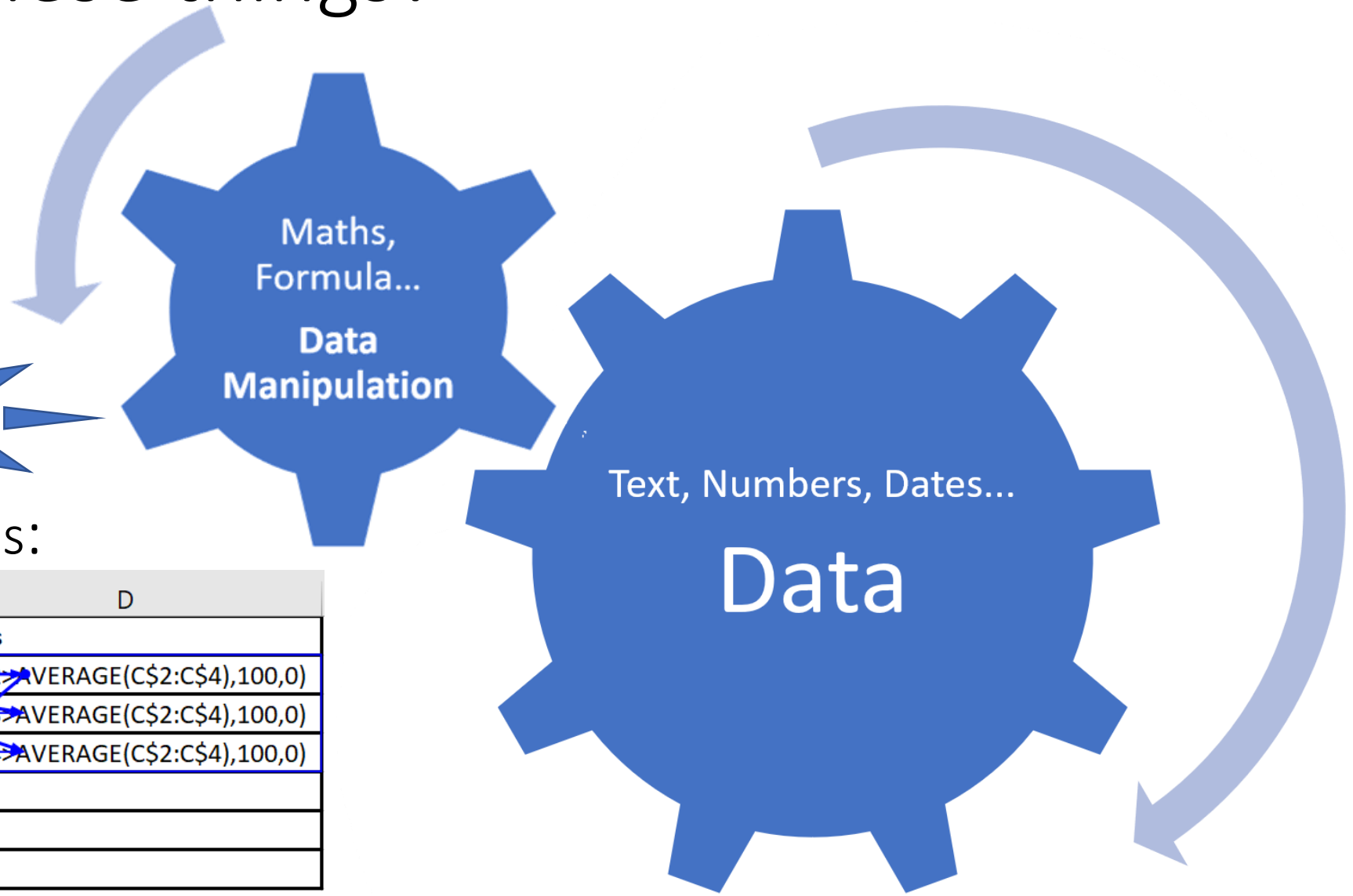
3. Our friends at Filigree Technologies are going to talk to you about a new playground for them

What are these things?

They have position:

- Row
- Column
- Worksheet
- …

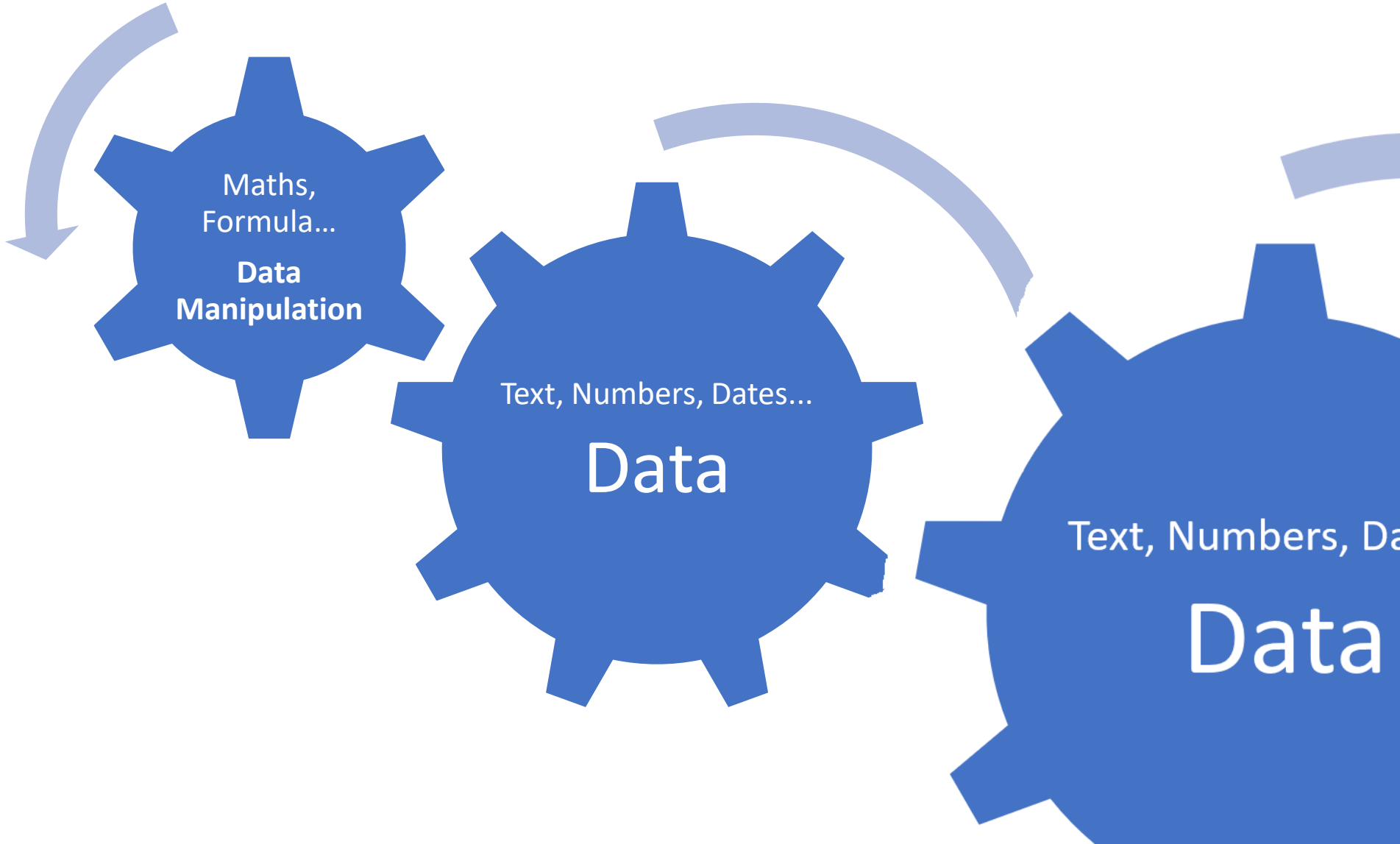




They have relationships:

	A	B	C	D
1	**Agent**	**Sales (sqft)**	**Sales(m2)**	**Bonus**
2	Fred	5221	=B2/3.28/3.28	=IF(C2>AVERAGE(C$2:C$4),100,0)
3	Dave	3872	=B3/3.28/3.28	=IF(C3>AVERAGE(C$2:C$4),100,0)
4	Bob	3651	=B4/3.28/3.28	=IF(C4>AVERAGE(C$2:C$4),100,0)
5				
6		Total Sales	=SUM(C2:C4)	
7		Total Bonus	=SUM(D2:D4)	

What are these things?



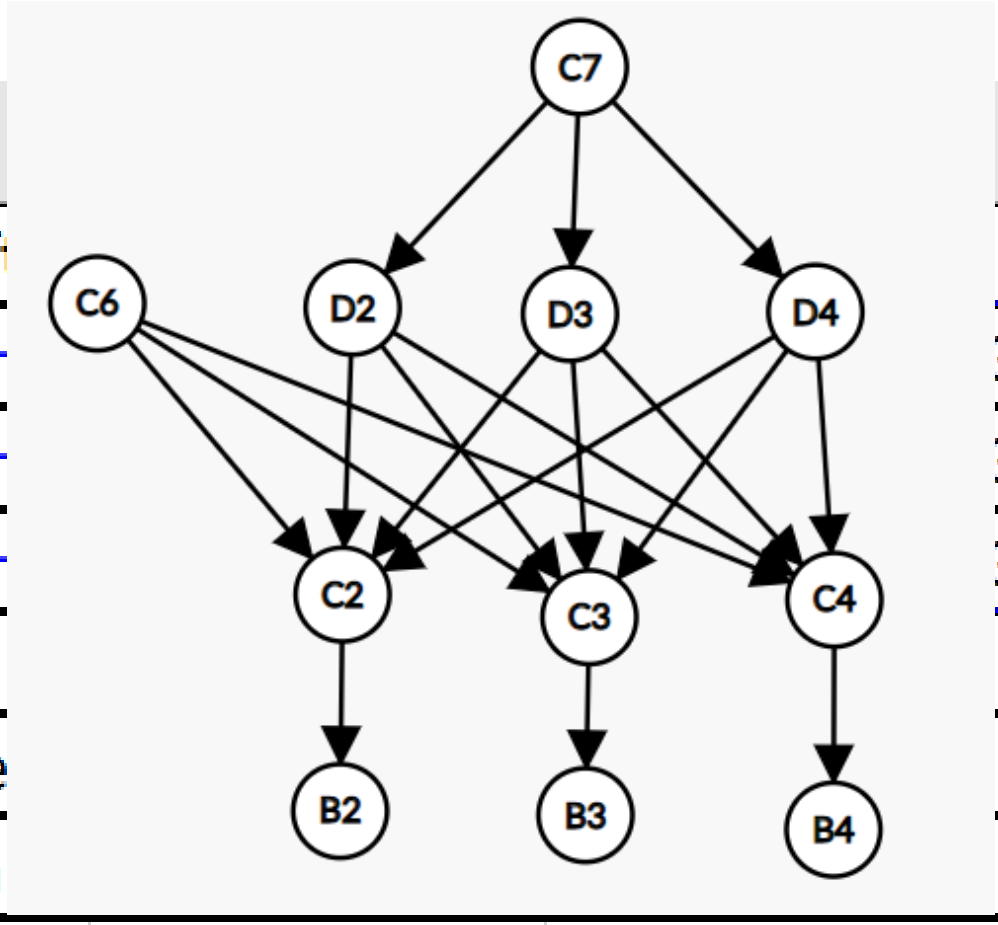
So just what is a cell?

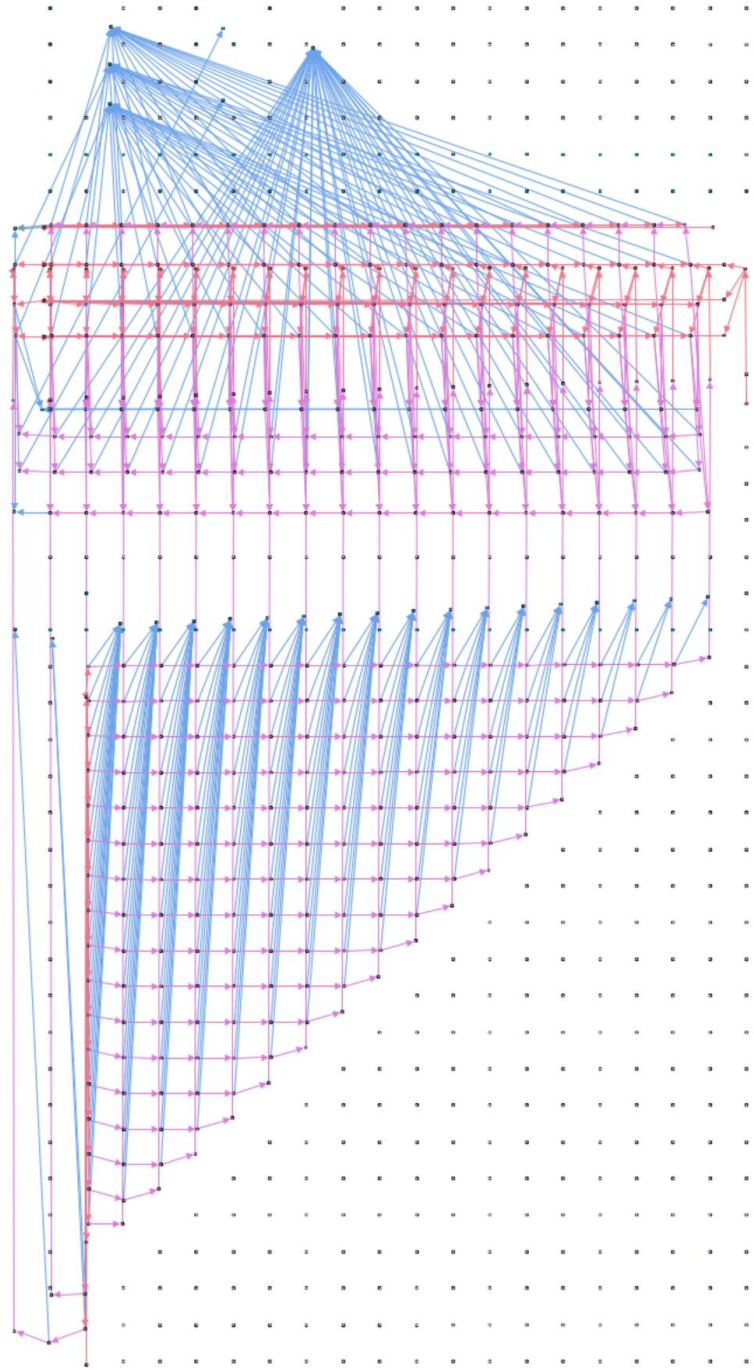
Let’s organise a jail break

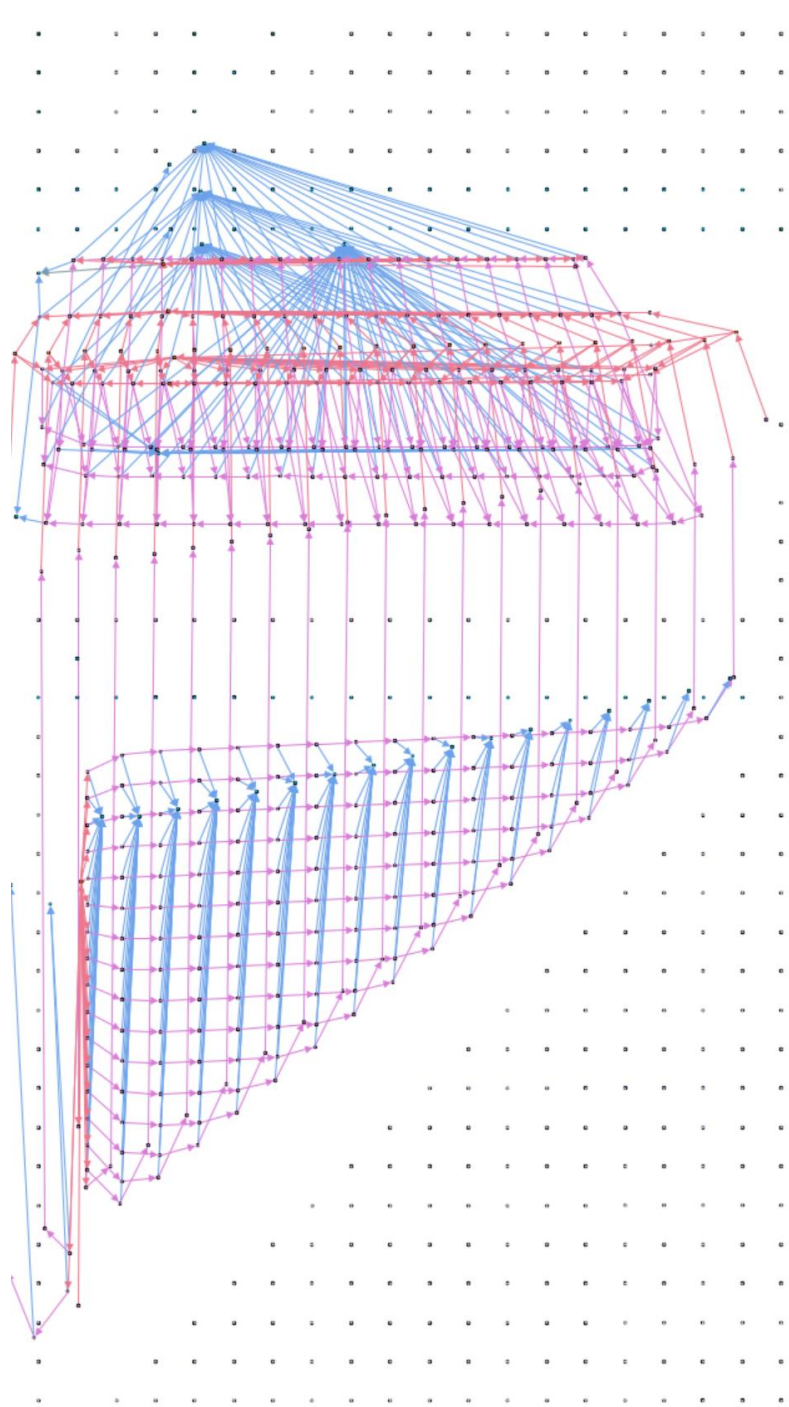
	A	B	C	D
1	**Agent**	**Sales (sqft)**	**Sales(m2)**	**Bonus**
2	Fred	5221	=B2/3.28/3.28	=IF(C2>AVERAGE(C$2:C$4),100,0)
3	Dave	3872	=B3/3.28/3.28	=IF(C3>AVERAGE(C$2:C$4),100,0)
4	Bob	3651	=B4/3.28/3.28	=IF(C4>AVERAGE(C$2:C$4),100,0)
5				
6		Total Sales	=SUM(C2:C4)	
7		Total Bonus	=SUM(D2:D4)	

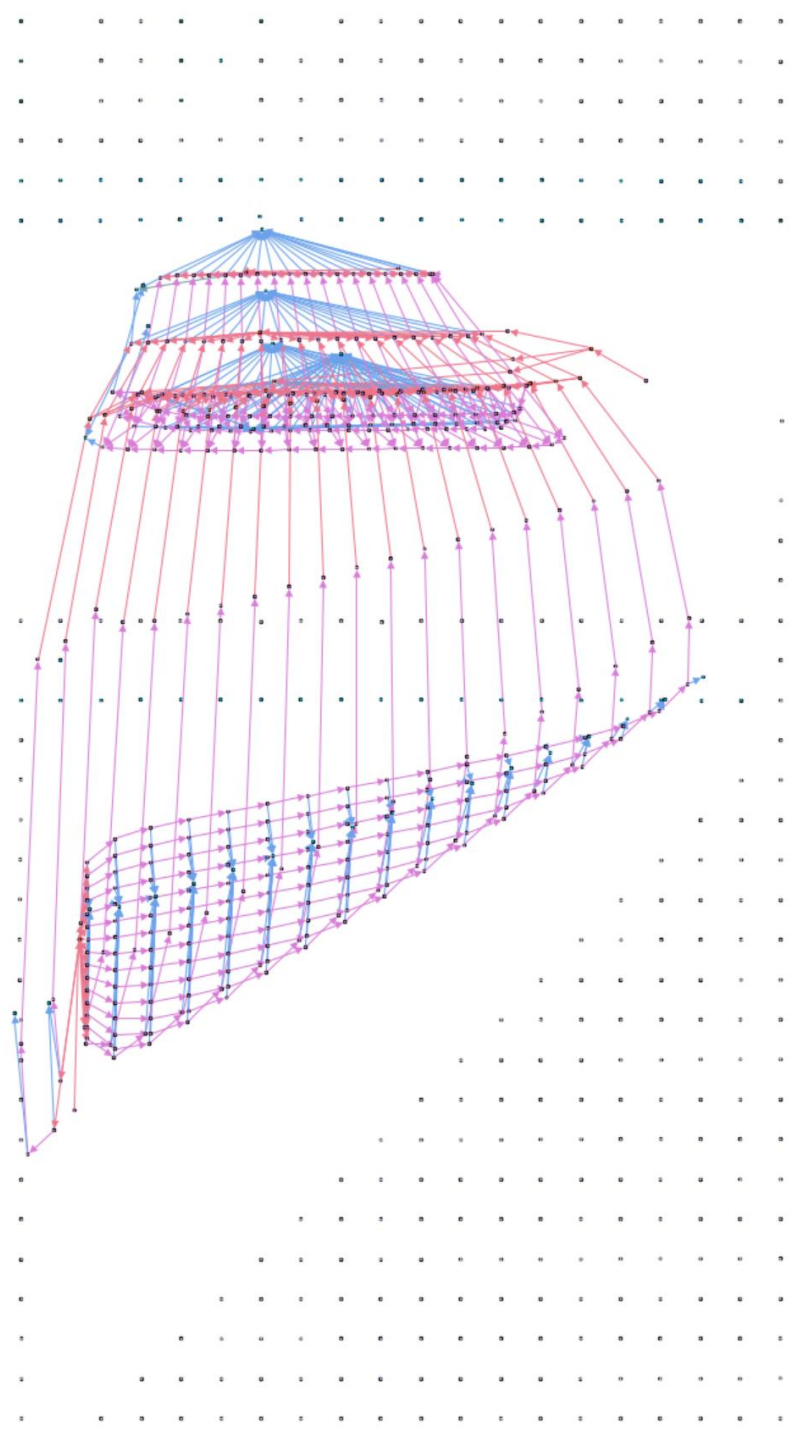
Let’s organise a jail break

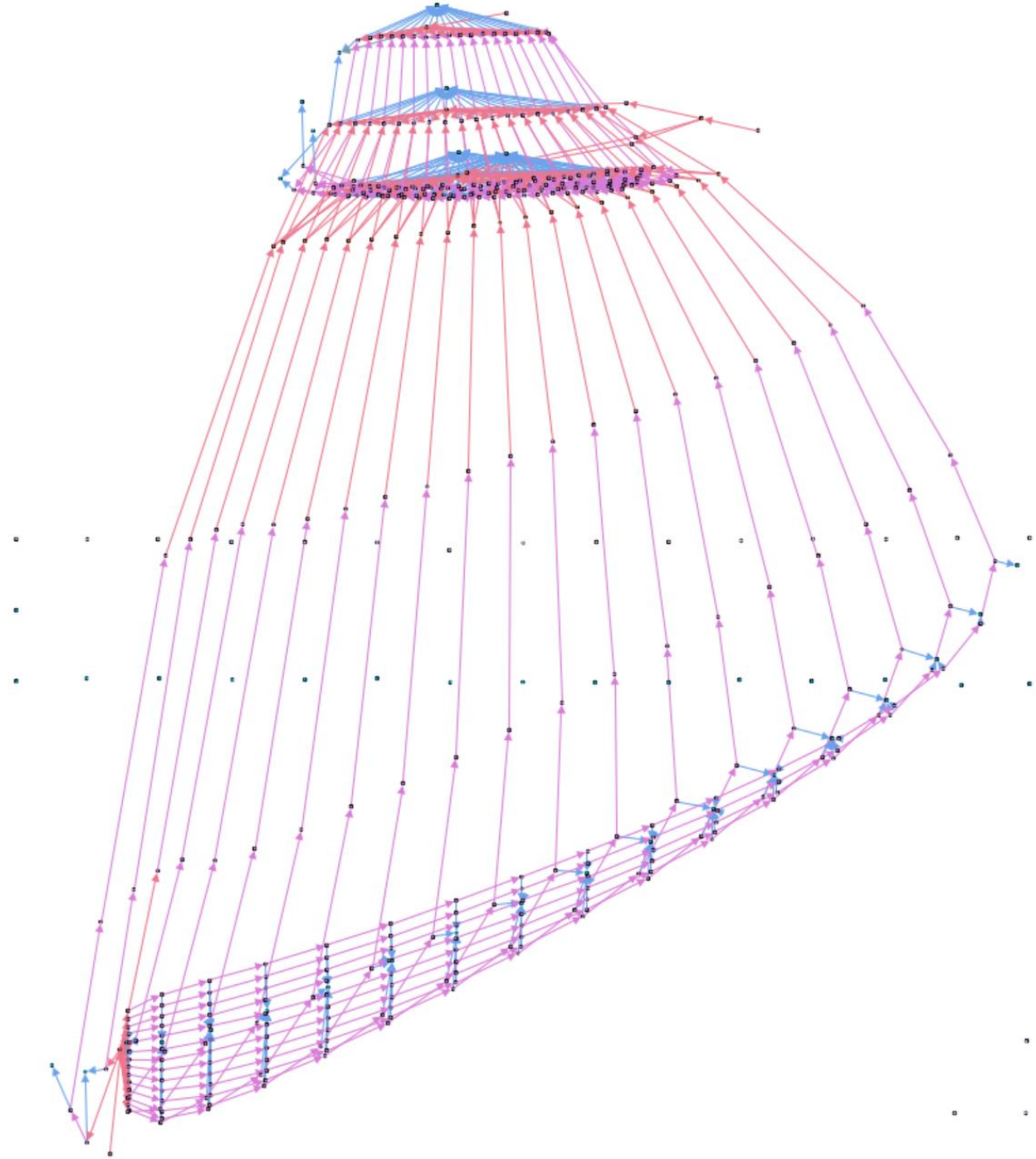
	A	B	D
1	**Agent**	**Sales (sqf**	
2	Fred	5221	GE(C$2:C$4),100,0)
3	Dave	3872	GE(C$2:C$4),100,0)
4	Bob	3651	GE(C$2:C$4),100,0)
5			
6		Total Sale	
7		Total Bon	

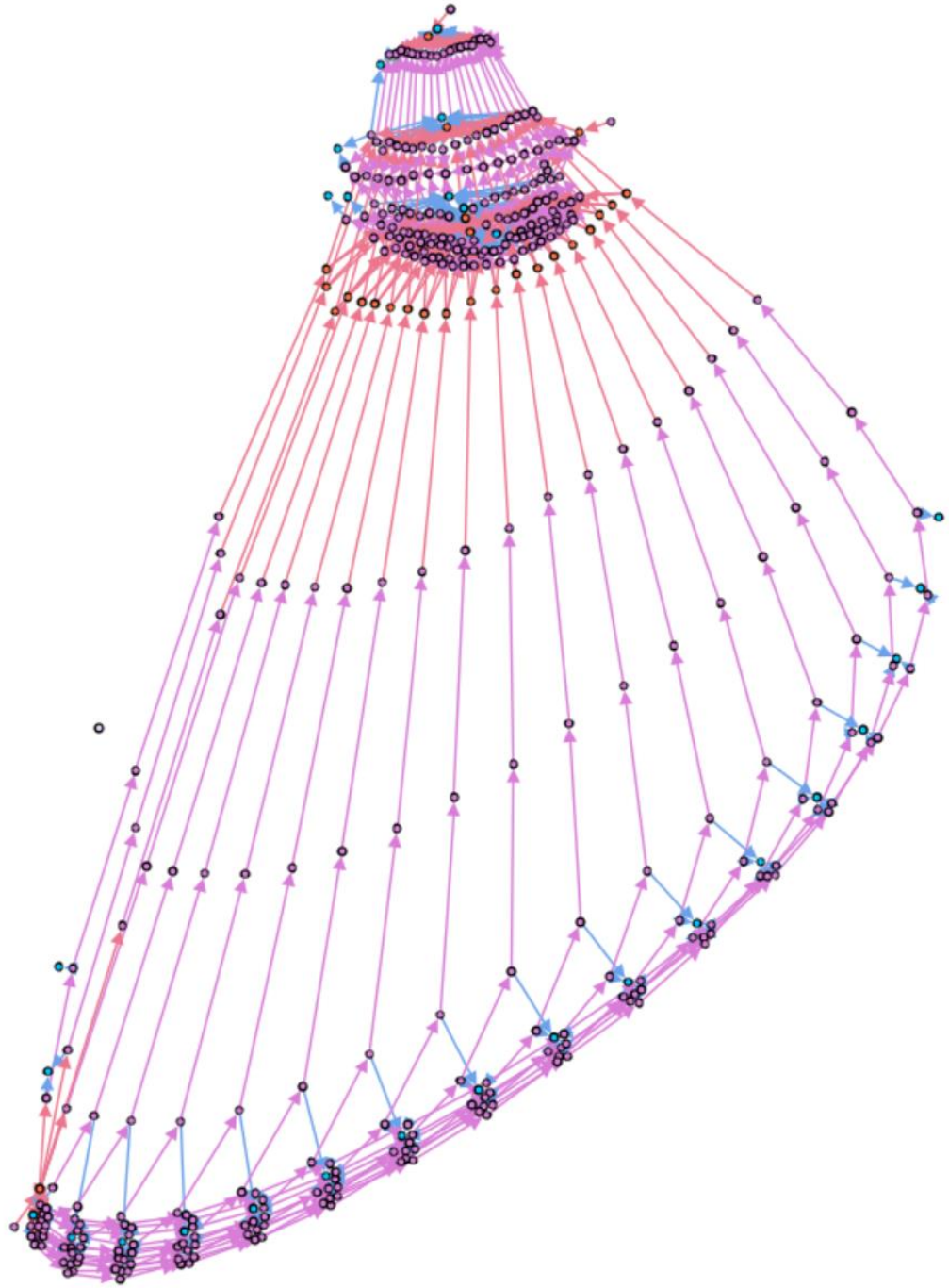


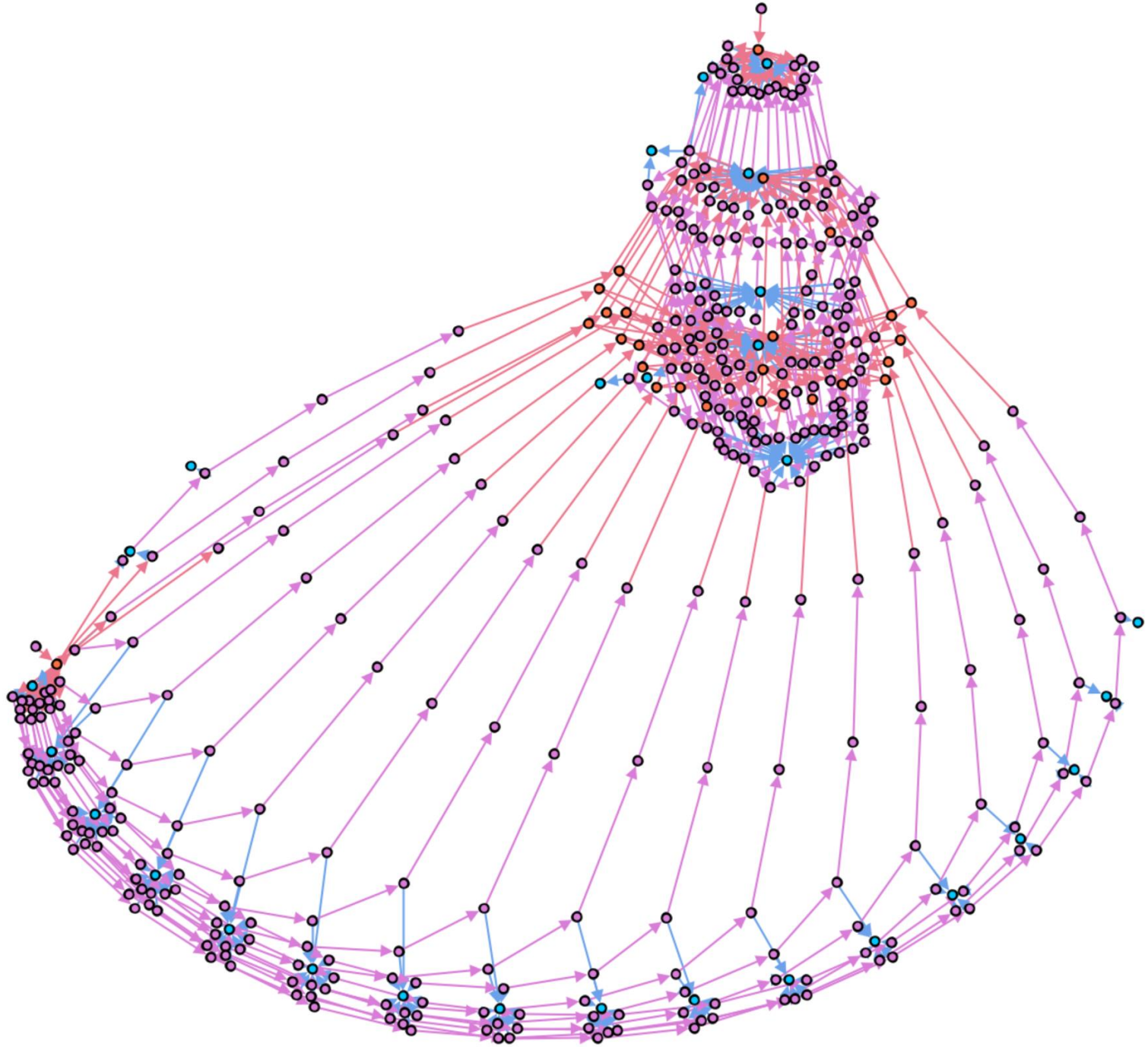


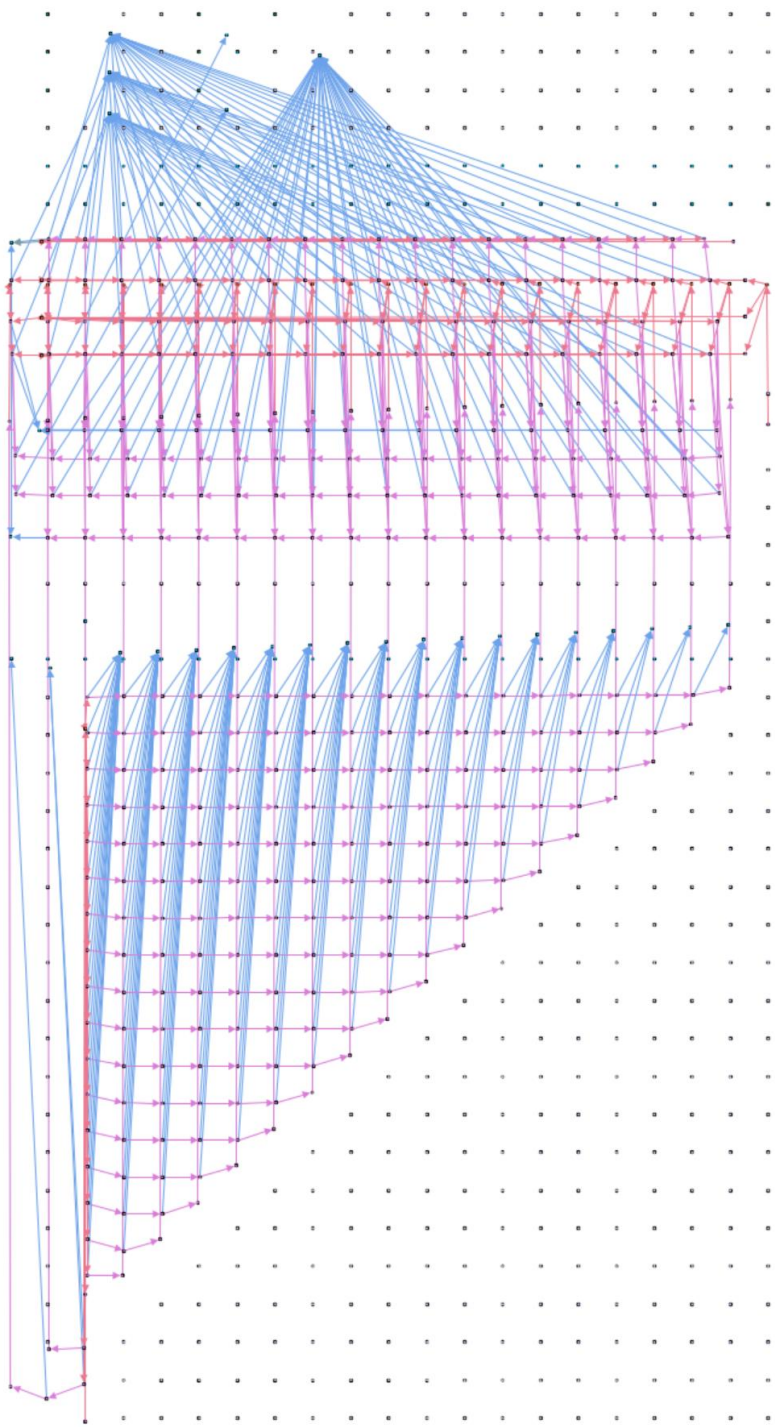


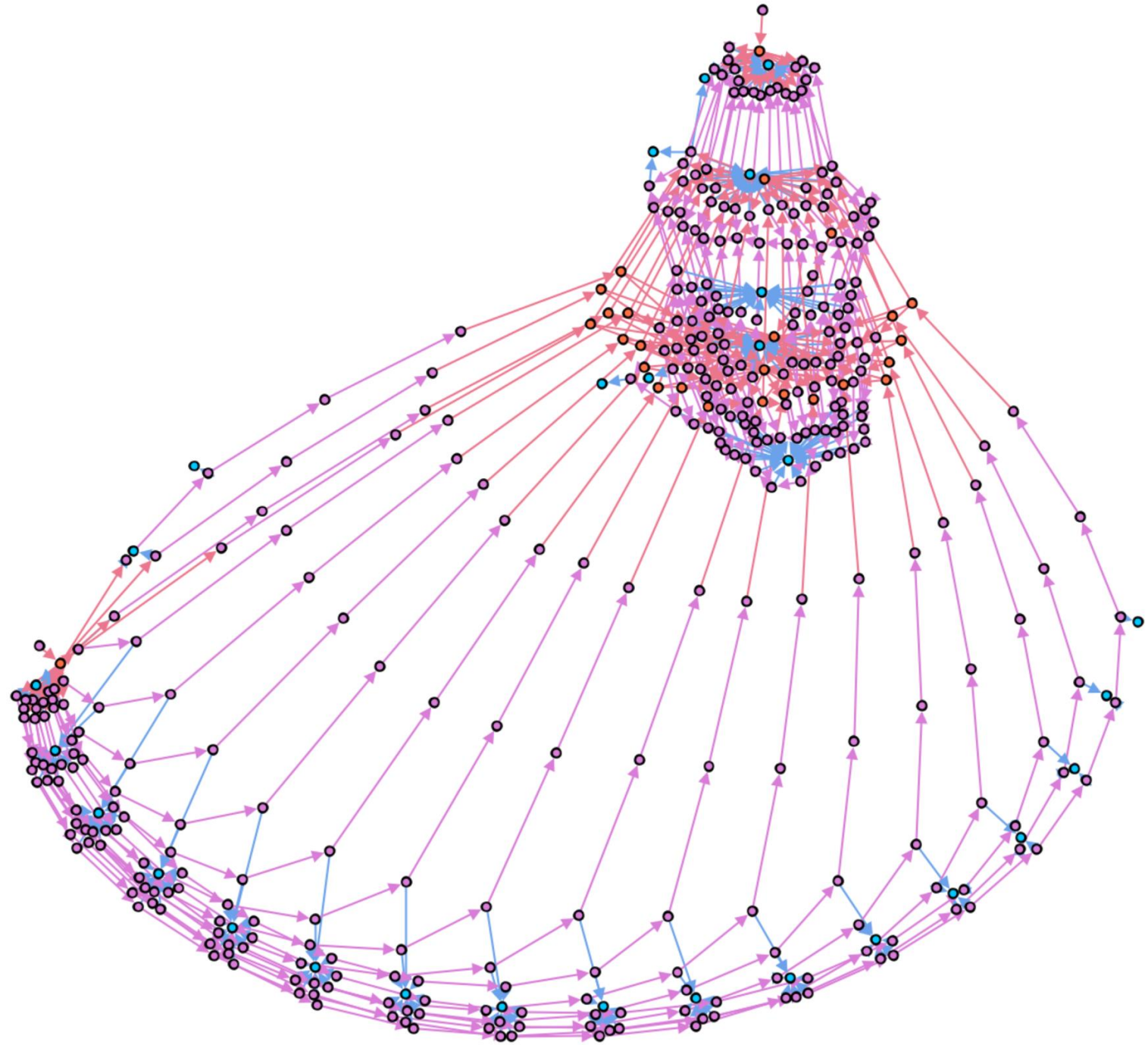




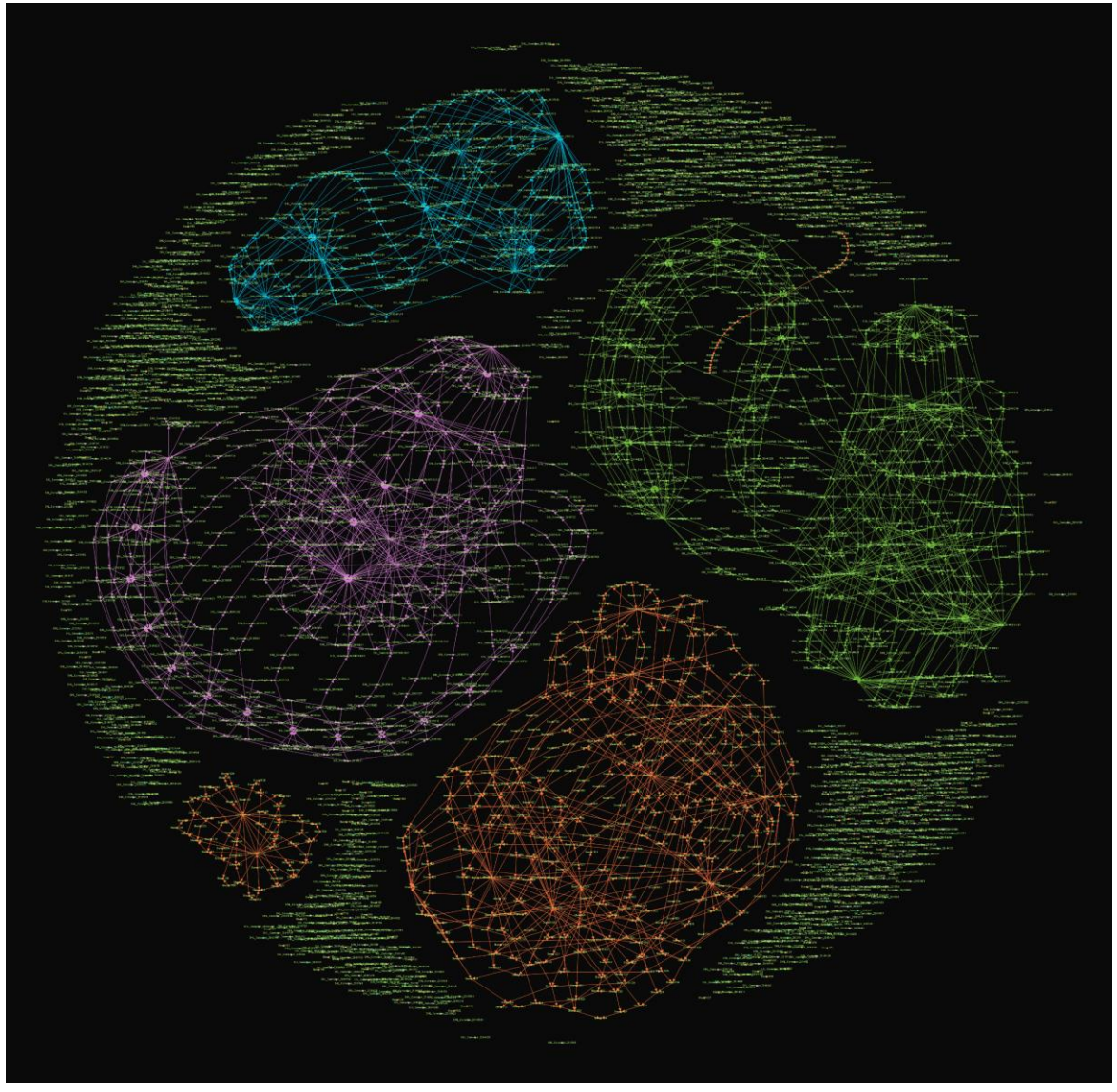




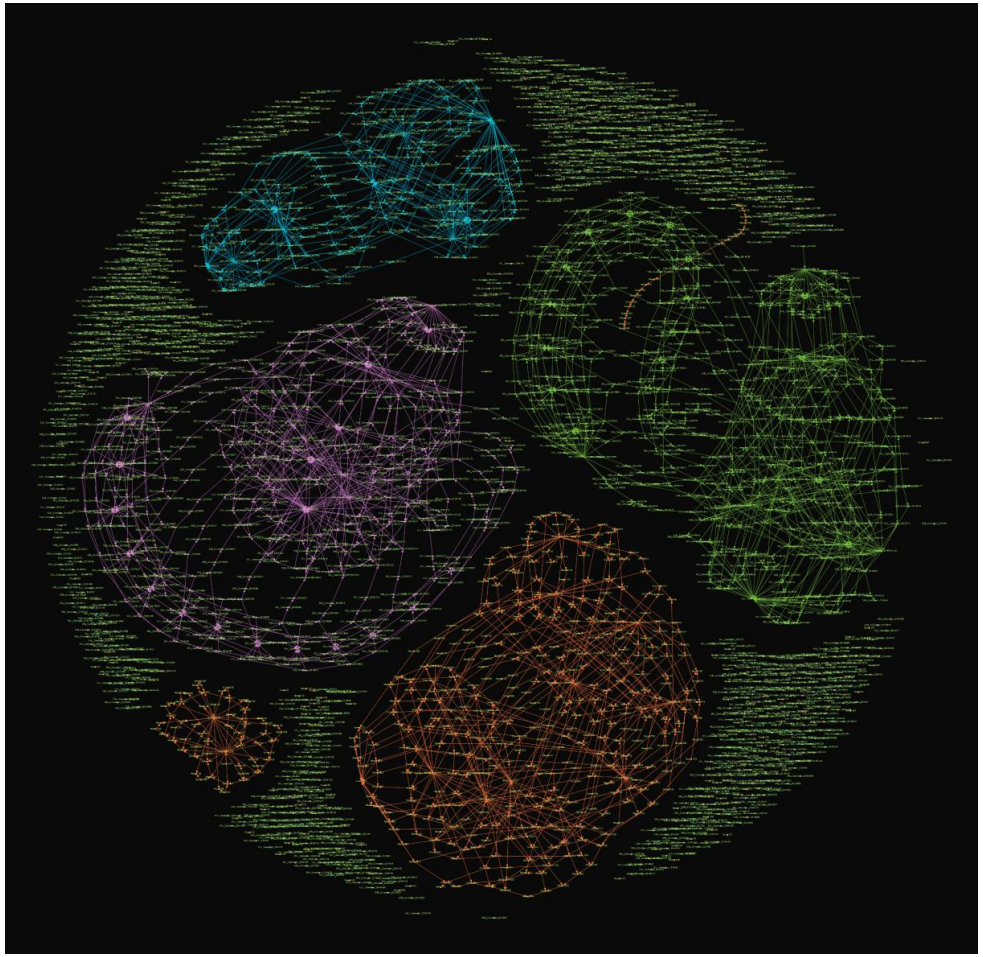


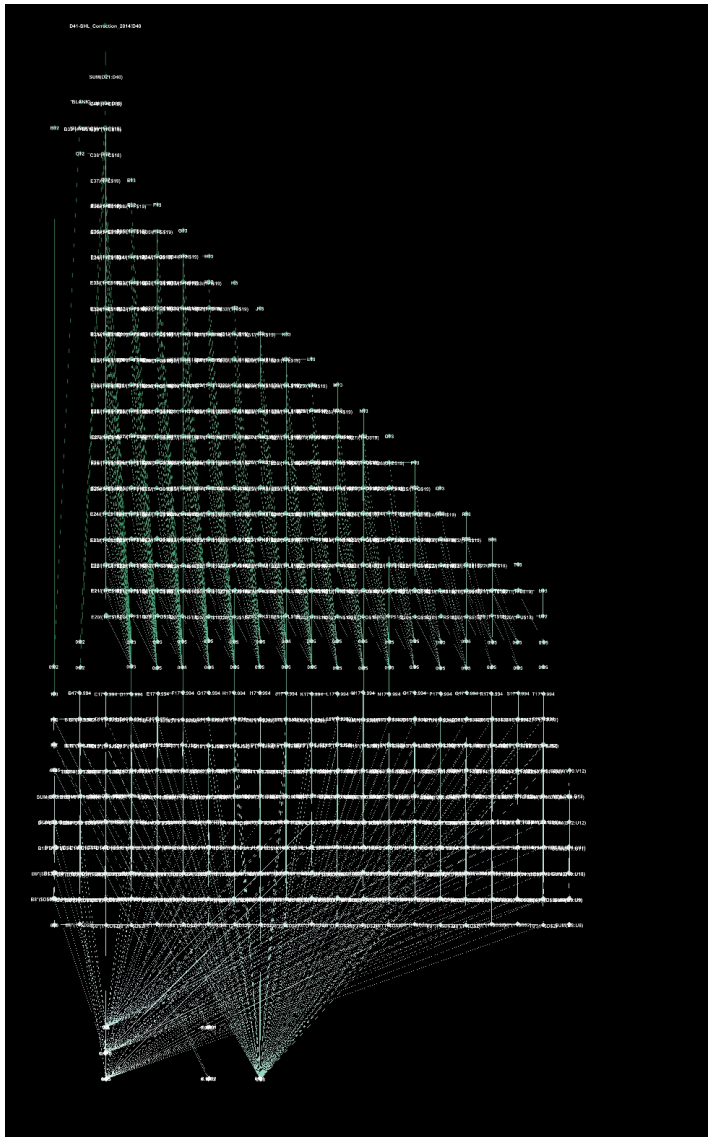


Cells on a Petri dish

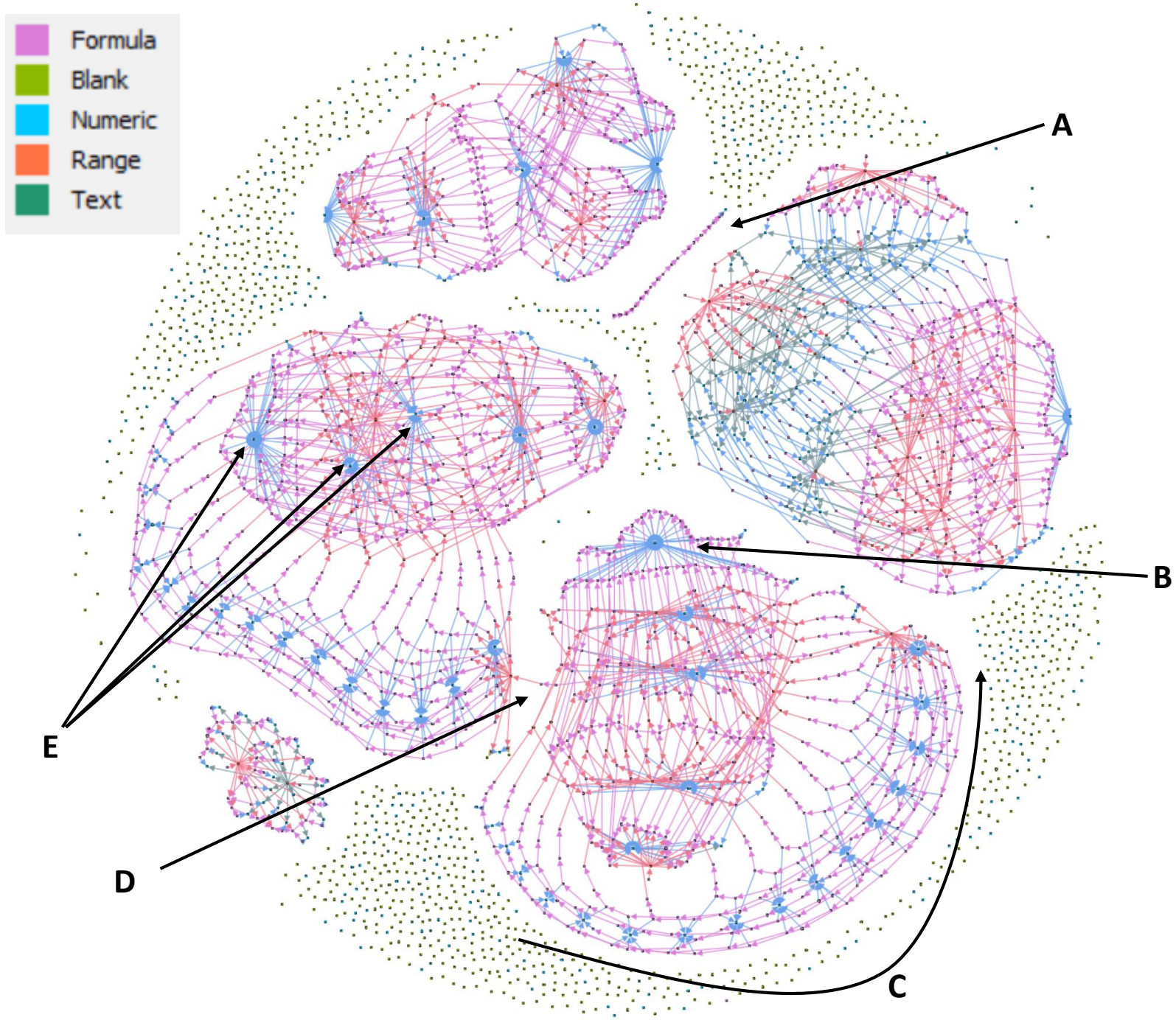


Or we can line them up >MSGFA2 algo



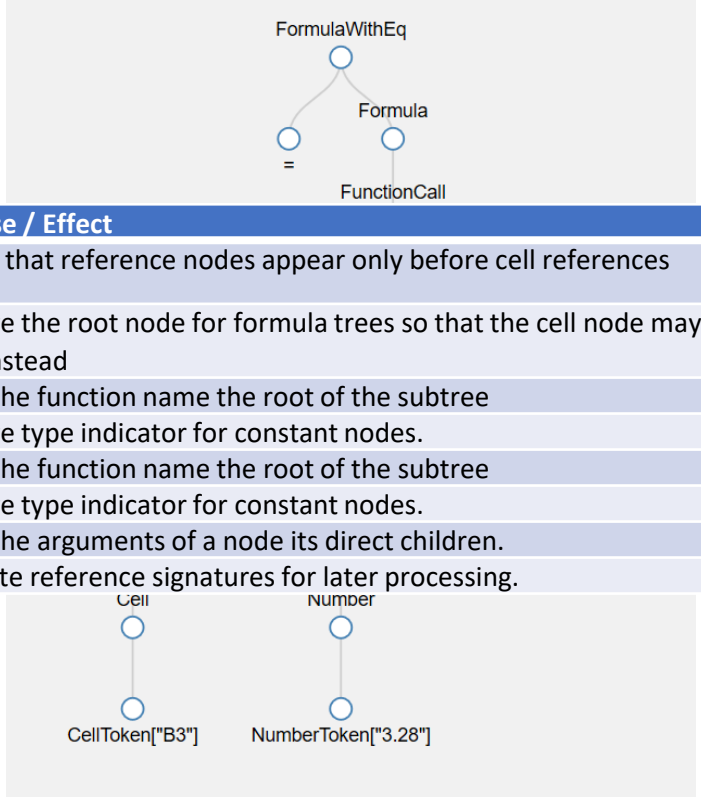


A. A dragged out chain of cells of the form A2=A1*1.10 used to roll a percentage increase over time.
B. A shared and commonly used numeric assumption in the model.
C. A triangular structure of cells calculating net present value of future profit. Each year along the arrow is further into the future and must be projected back one year further, giving the growing complexity of the calculation.
D. A comparison between two business cases. Note the two structurally identical calculation chains being compared differ only by different numeric business assumptions.
E. Examples of the business assumptions being frequently referenced in the calculation model.

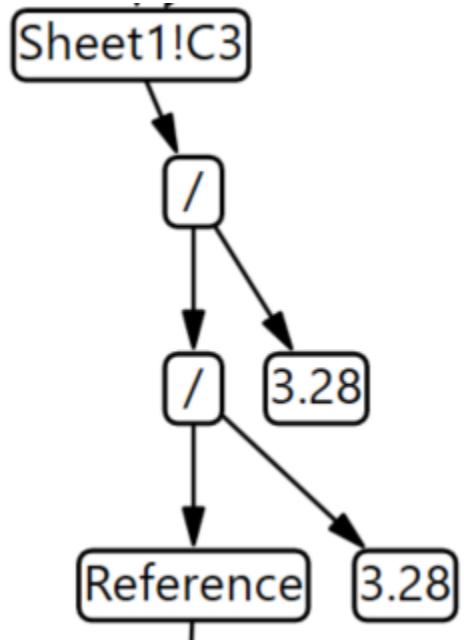


But all we are doing is playing with the prison cells – what is inside?

=B3/3.28/



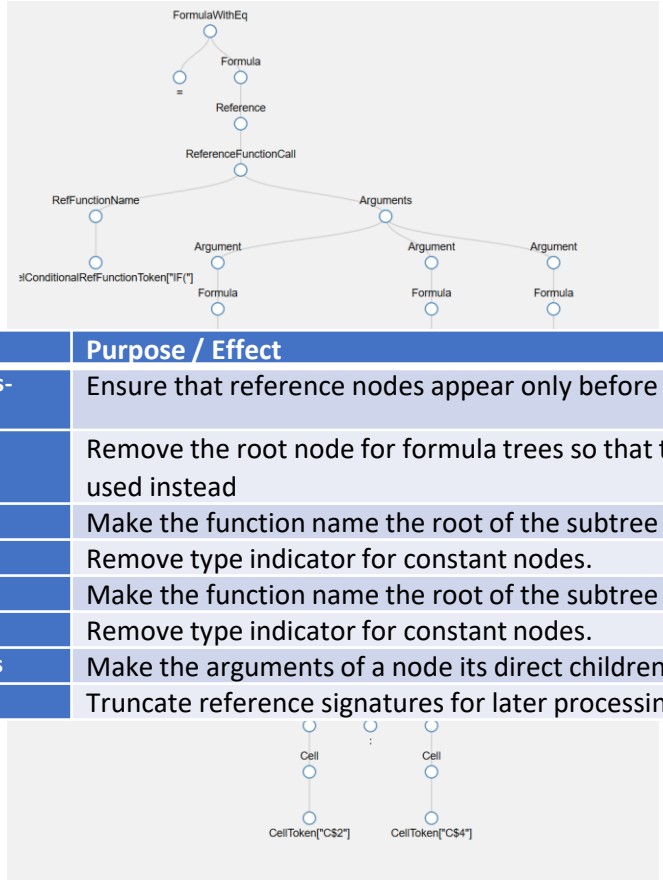
Refactoring	Purpose / Effect
RemoveReferenceNodes-BeforeFunctionCalls	Ensure that reference nodes appear only before cell references
RemoveFormulaEqNode	Remove the root node for formula trees so that the cell node may be used instead
InlineFunctionNames	Make the function name the root of the subtree
RemoveConstantNodes	Remove type indicator for constant nodes.
RemoveFormulaNodes	Make the function name the root of the subtree
RemoveNumberNodes	Remove type indicator for constant nodes.
RemoveArgumentNodes	Make the arguments of a node its direct children.
TruncateReferences	Truncate reference signatures for later processing.



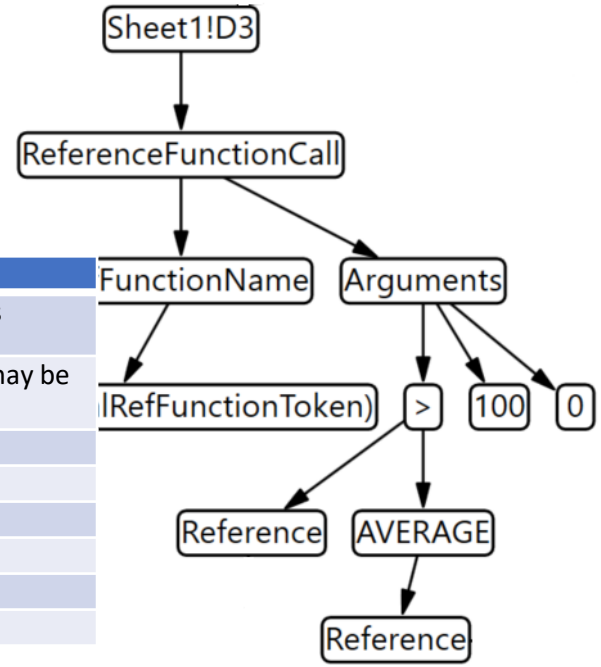
Aivaloglou, E., Hoepelman, D. and Hermans, F., 2015, September. A grammar for spreadsheet formulas evaluated on two large datasets. In *2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)* (pp. 121-130). IEEE.

A more complex example

=IF(C3>AVERAGE(C$2:C$4),100,0)

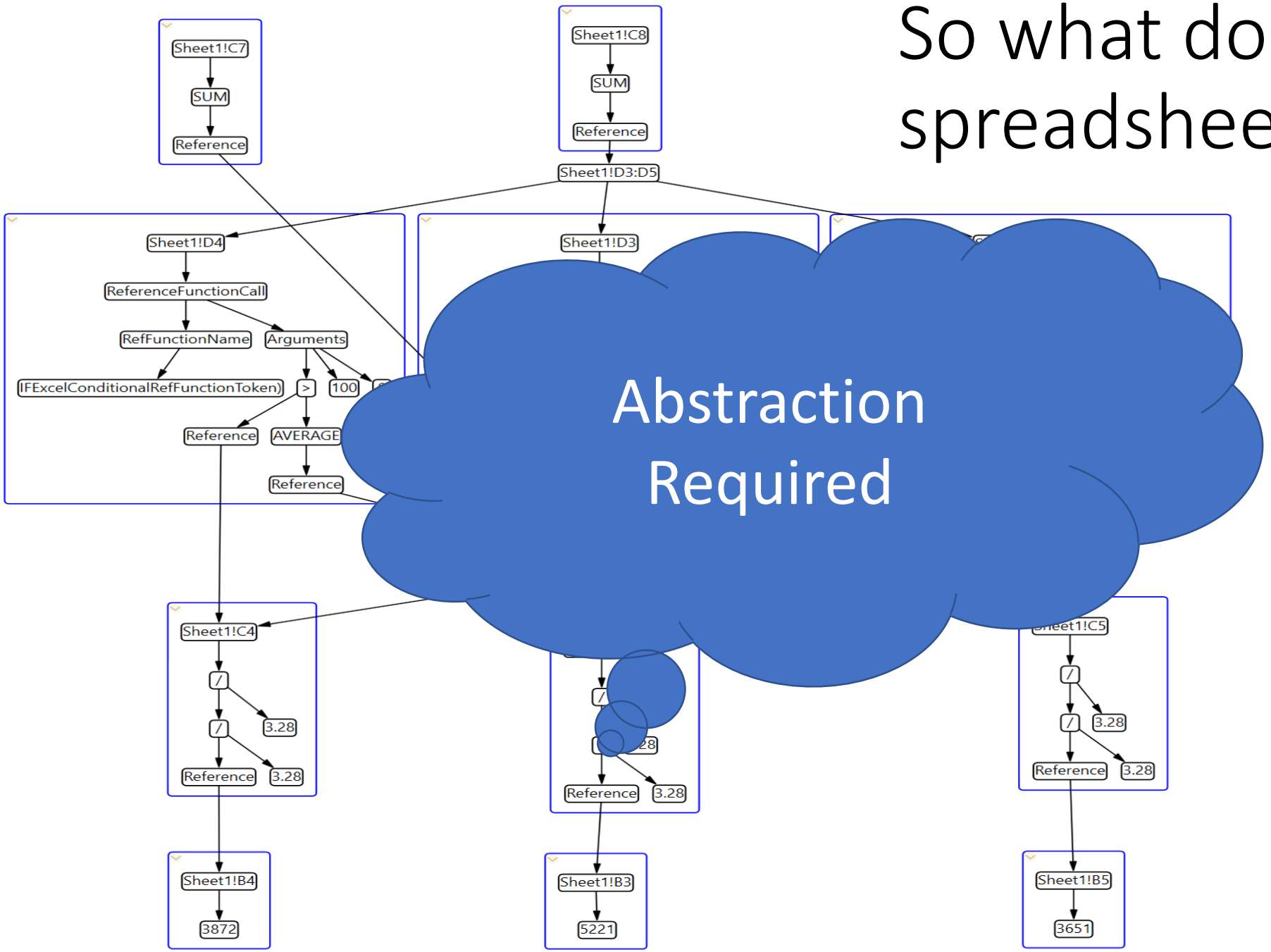


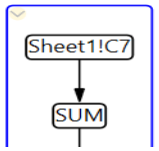
Refactoring	Purpose / Effect
RemoveReferenceNodes-BeforeFunctionCalls	Ensure that reference nodes appear only before cell references
RemoveFormulaEqNode	Remove the root node for formula trees so that the cell node may be used instead
InlineFunctionNames	Make the function name the root of the subtree
RemoveConstantNodes	Remove type indicator for constant nodes.
RemoveFormulaNodes	Make the function name the root of the subtree
RemoveNumberNodes	Remove type indicator for constant nodes.
RemoveArgumentNodes	Make the arguments of a node its direct children.
TruncateReferences	Truncate reference signatures for later processing.

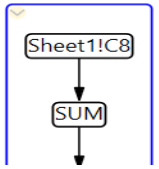


Aivaloglou, E., Hoepelman, D. and Hermans, F., 2015, September. A grammar for spreadsheet formulas evaluated on two large datasets. In *2015 IEEE 15th International Working Conference on Source Code Analysis and Manipulation (SCAM)* (pp. 121-130). IEEE.

So what does a whole spreadsheet look like?







It's a hypergraph

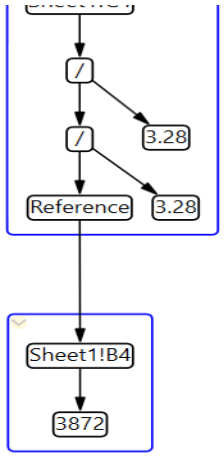
Let $H = (X, E)$ be the hypergraph consisting of vertices

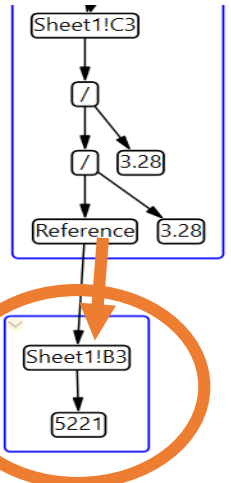
$$X = \{x_i | i \in I_v\},$$

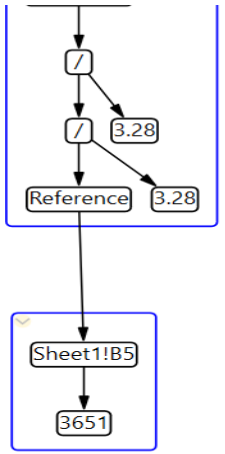
and having *edge set*

$$E = \{e_i | i \in I_e \wedge e_i \subseteq X \wedge e_i \neq \emptyset\},$$

where I_v and I_e are the index sets of the vertices and edges respectively.



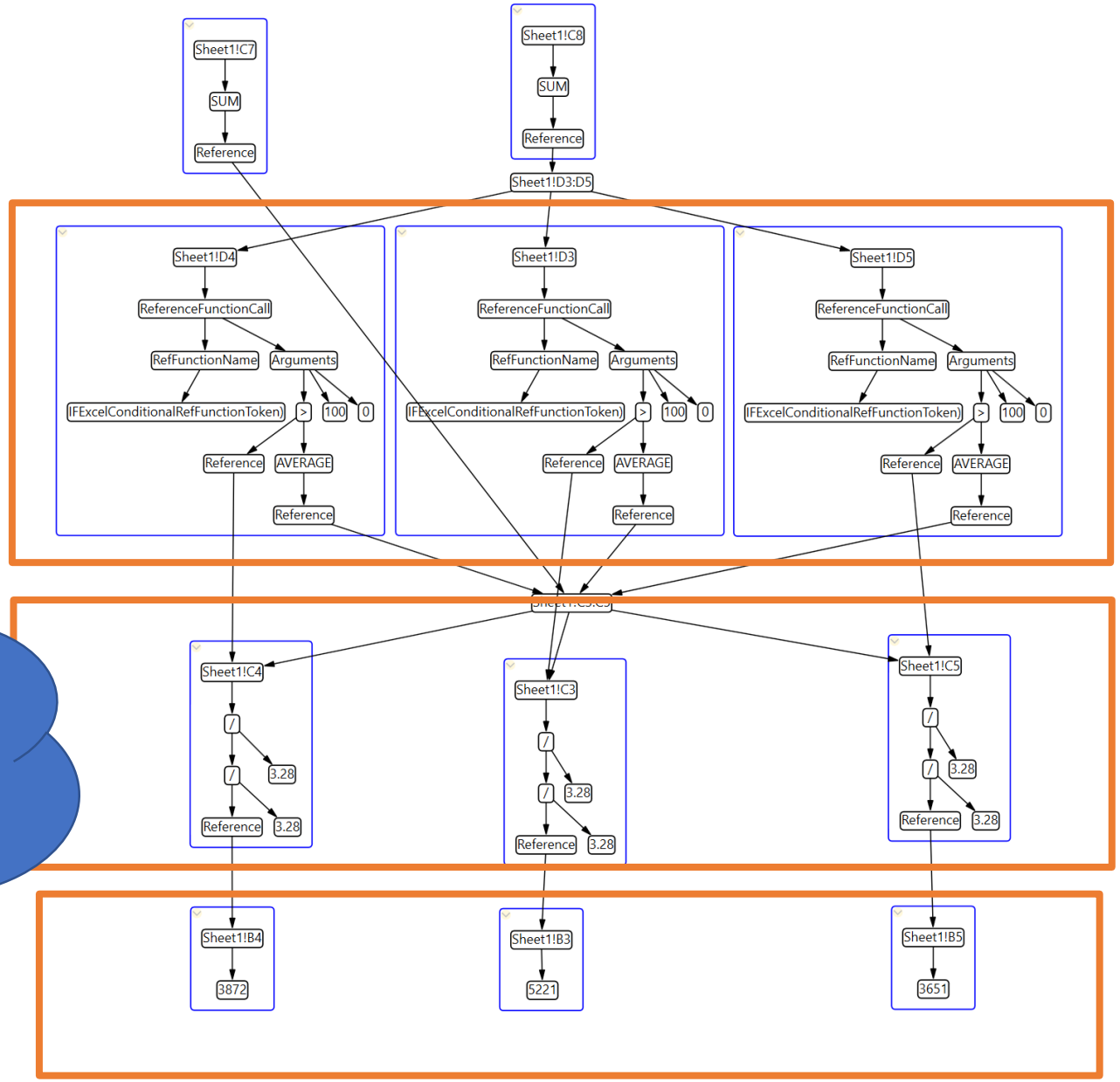




There are patterns here

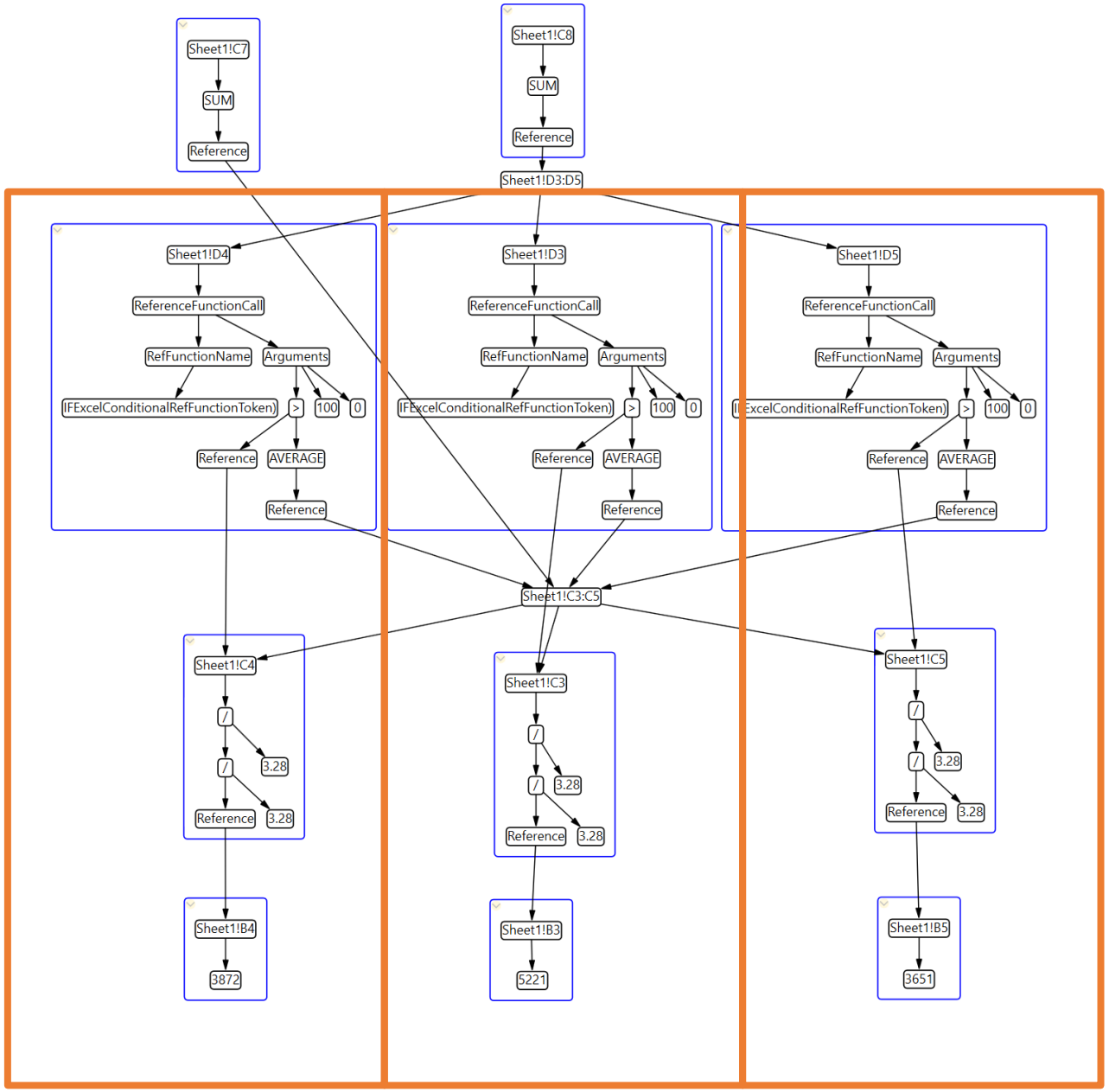
- Similar structures are repeated

Abstraction Required



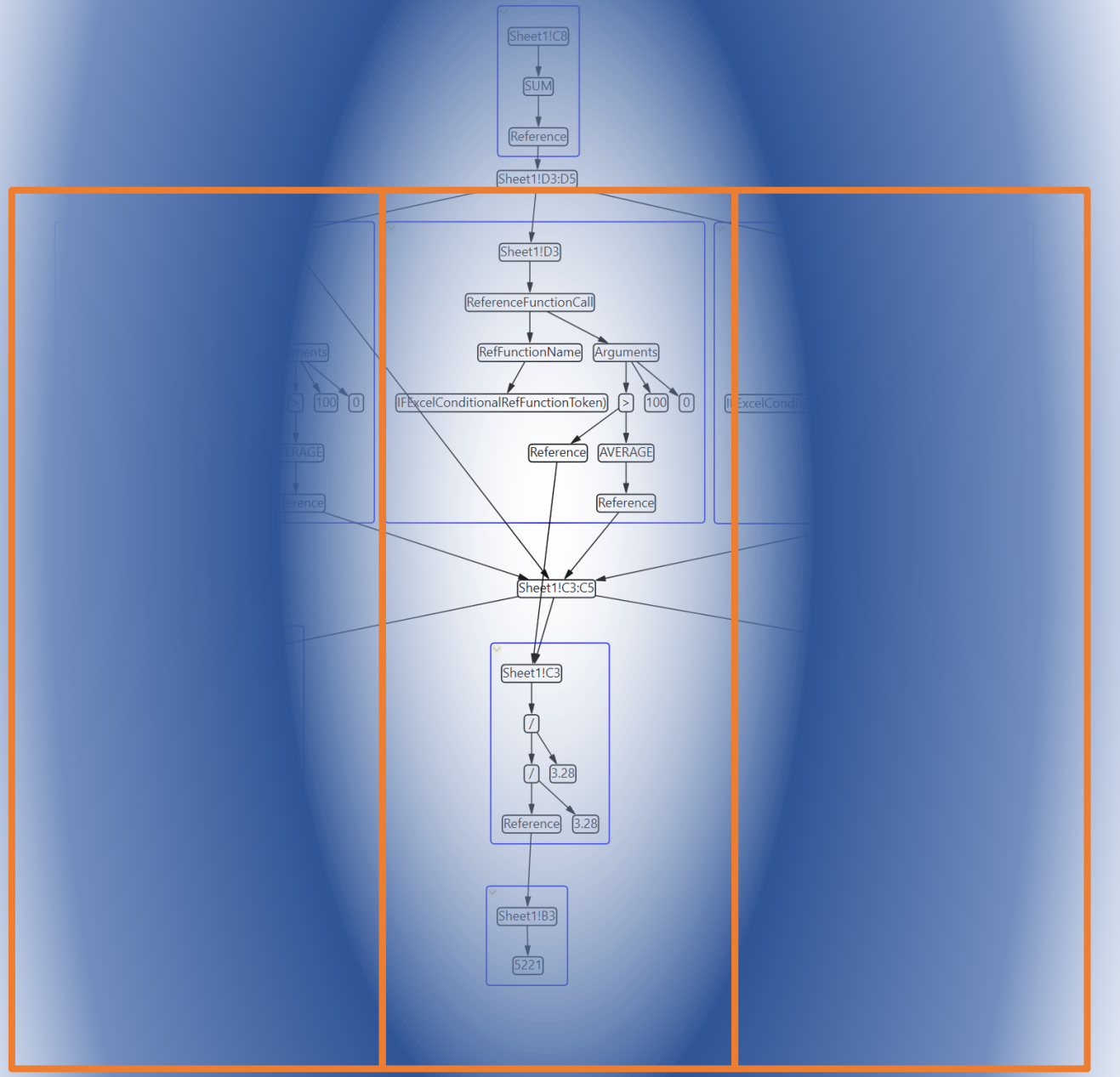
There are patterns here

- Similar structures are repeated



There are patterns here

- Similar structures are repeated
- Data Changes but calculation does not change
- Vectorization!



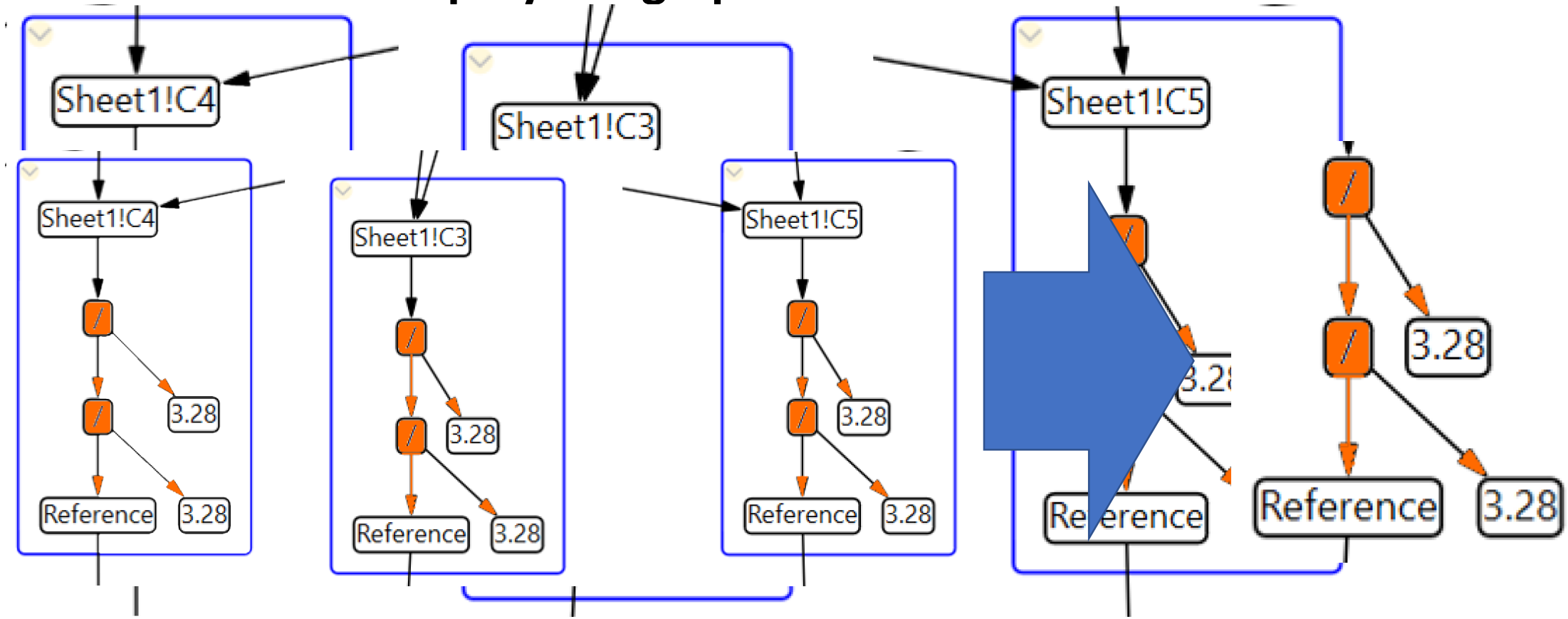
Pattern matching AKA Graph Isomorphisms

- By analysing this hypergraph view of the spreadsheet we seek mechanisms for abstracting the spreadsheet by redrawing the boundaries of cells. To identify opportunities for this process we seek to find repeated structure within the graph which may be replaced by a higher level more abstract “cell”. **In order to do this we explore three abstracting operations:**

1. **We seek table structures of input values.**
2. **We seek vector (array) operations which apply the same operations to related data.**
3. **We seek common sub-expressions within cells which may be factored out to simplify the graph.**

Pattern matching AKA Graph Isomorphisms

1. **We seek common sub-expressions within cells which may be factored out to simplify the graph.**
2.
3.



Step by step

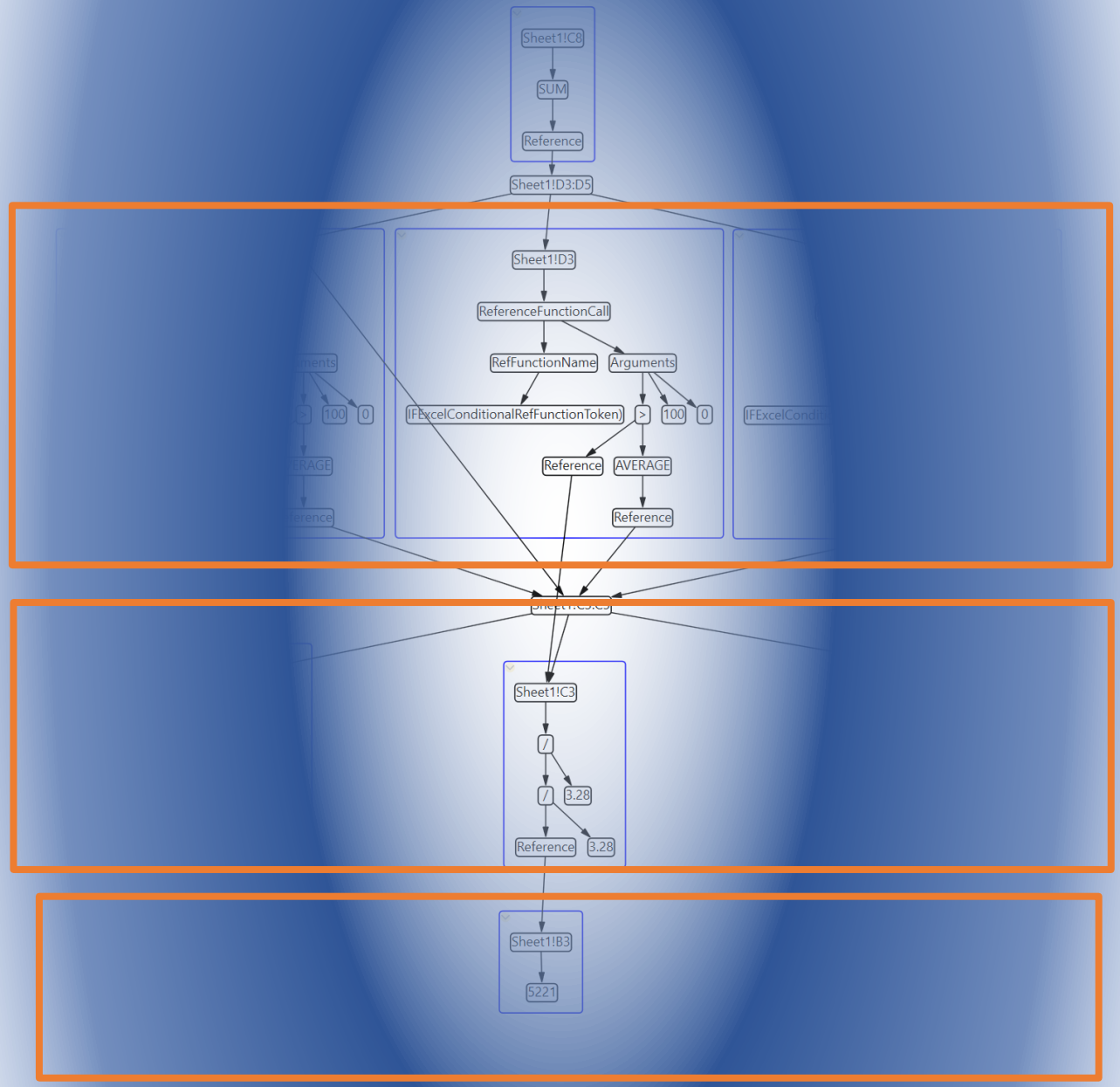
- **Structural isomorphism** – the same operators in the same order
- **Colocation** – located in a contiguous cell space
- **Constant isomorphism** – using the same textual and numeric constants in its calculation
- **Reference isomorphism** – using isomorphic references within its calculation.

Higher Level Abstraction

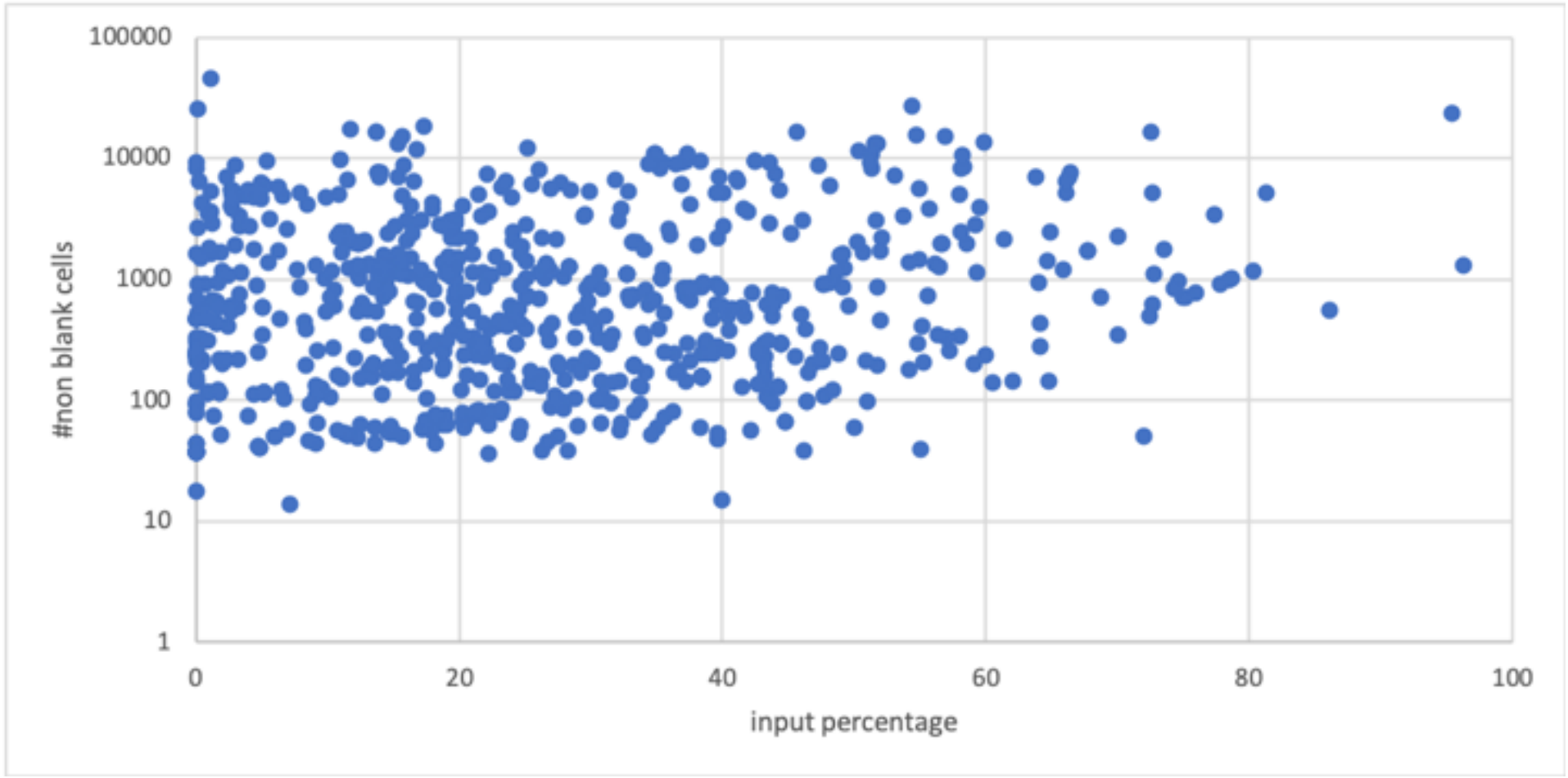
- Compression of spreadsheets

Simplify the hyper graph

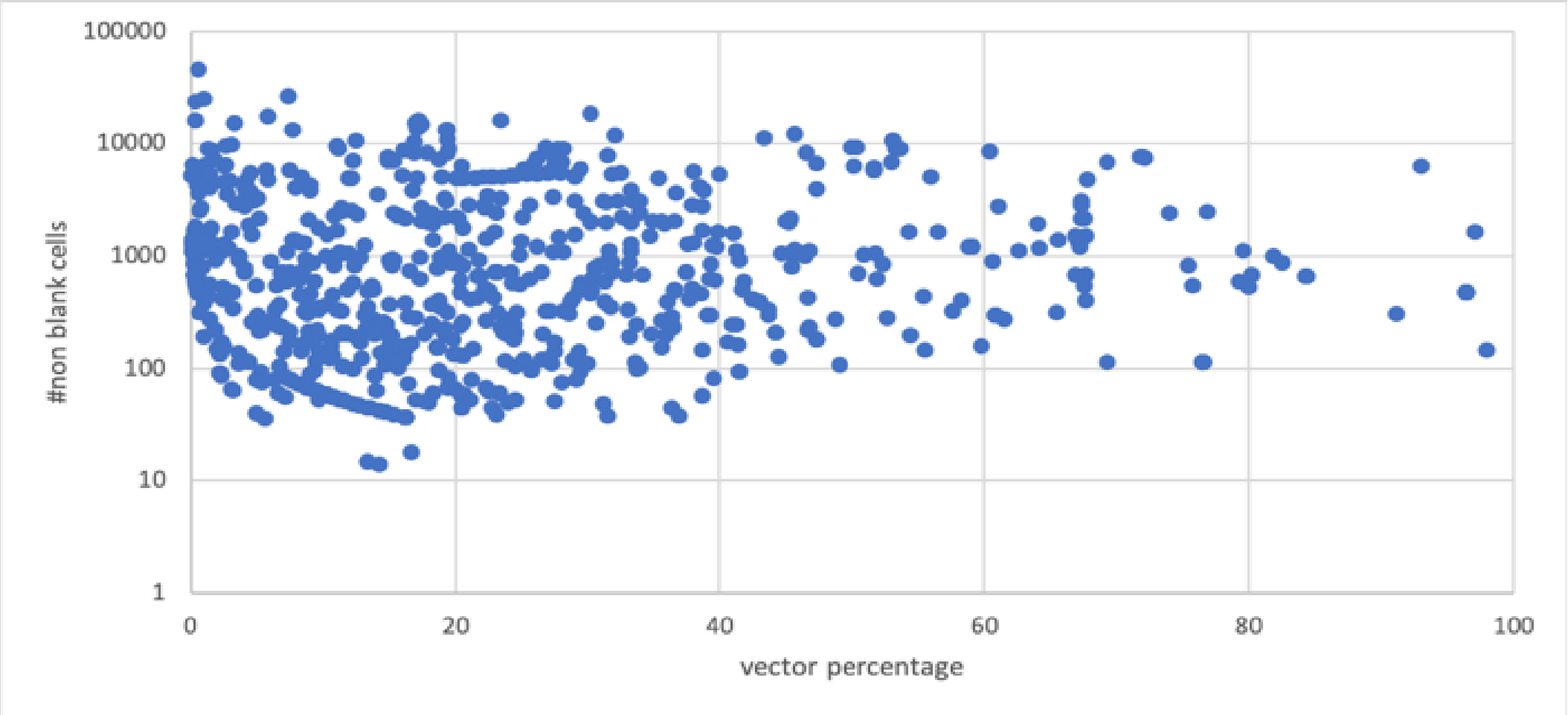
- Find repeated graph structure
- Redraw the cell walls
- More abstract model
  - Each “cell” now operates on array data



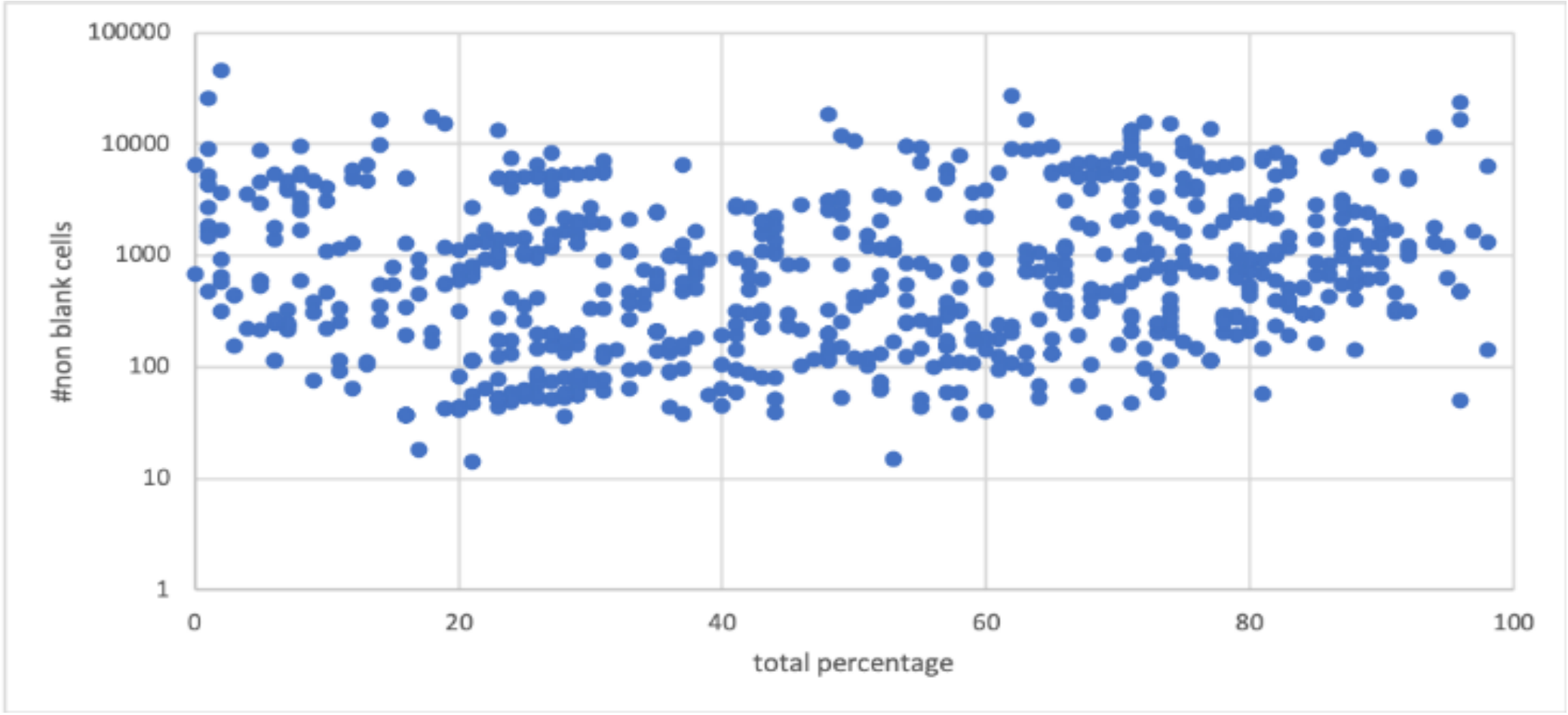
Results on Enron Sample (2124)



Results on Enron Sample (2124)



Results on Enron Sample (2124)



In Summary:

1) Cells > Graph
2) Formula > Graph
3) Cells = hypergraph edges
4) Seek repeated structure
5) Redraw the “cell walls” to create abstraction