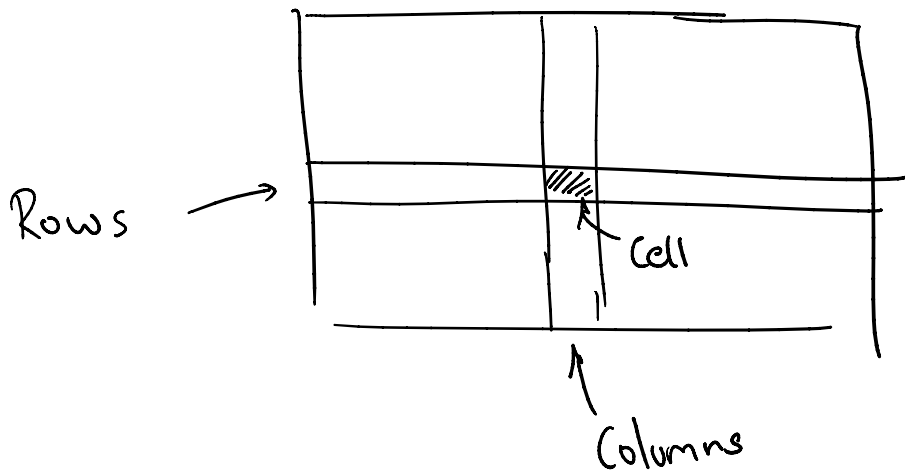
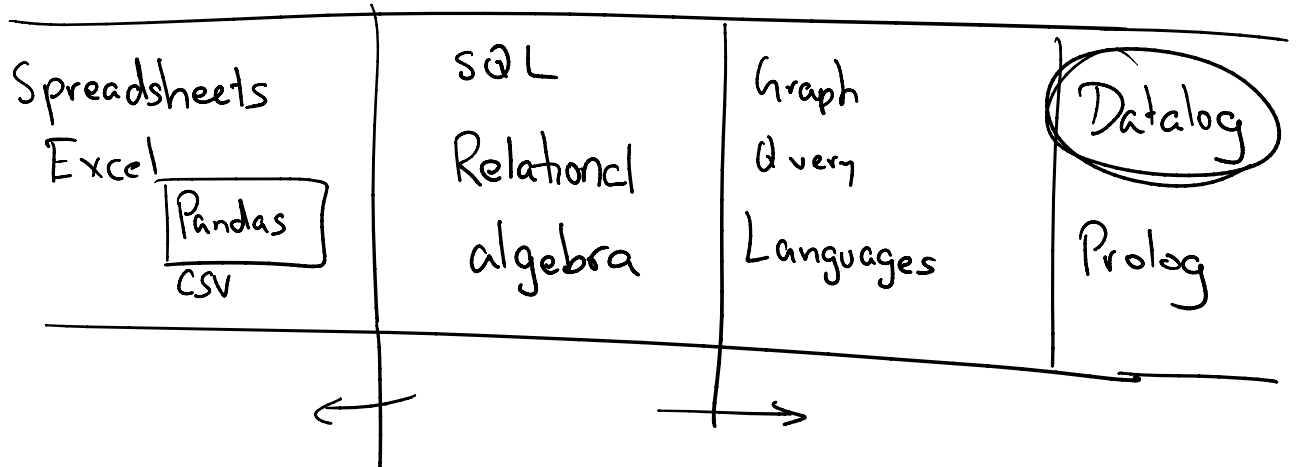
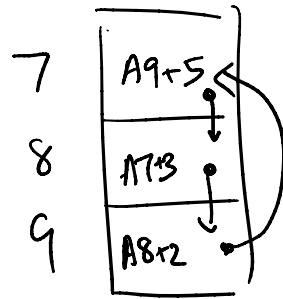


Unit 3 : Programming with Relations



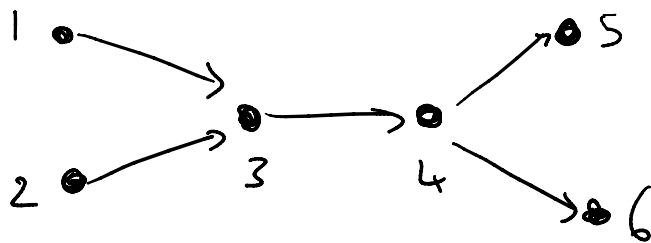
Circular references

A



A spreadsheet is well-formed if the graph of induced dependences is a DAG.

Topological sort ← Directed Acyclic Graph



3 → 4 here

1 2 3 4 5 → 6

2 1 3 4 6 5

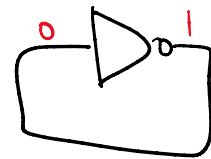
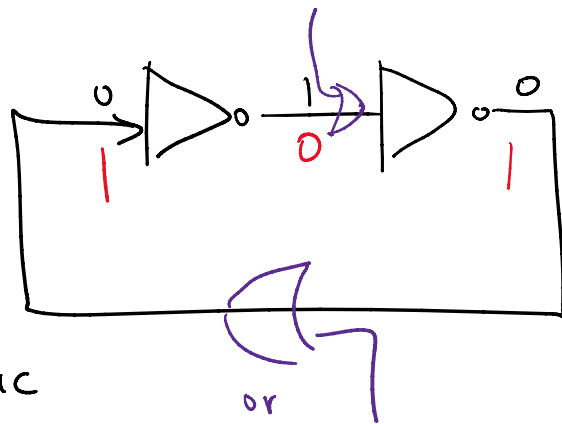
⋮

So, 3 before 4

here

;

DAGs
↓
Stateless electronic
circuits / Combinatorial logic



DAGs → Spreadsheets

Makefiles / Build systems

Combinatorial circuits

What are (the equivalent of) pointers?

$\text{int } * \text{ ptr} = \& x$ Finding addresses ②

$* \text{ ptr}$ Dereferencing ①

*ptr — Dereferencing ①

① Dereferencing

- Here is a cell.
- In it, you will find the address of some other cell
- Get me the value of that other cell.

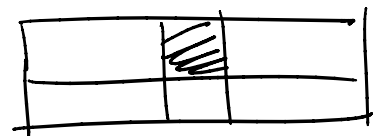
INDIRECT

ADDRESS: Convert row number & column number into alphanumeric index

VLOOKUP



HLOOKUP



MATCH: Find index of cell

MATCH : Find index of cell

Building VLOOKUP using match

VLOOKUP ("John", A12:B14, 2)

= let index =
MATCH ("John", A12:A14, 0) (1)

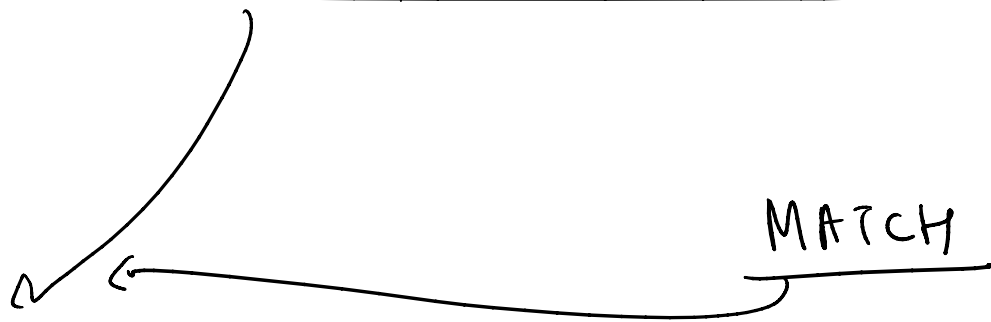
let base Row = Row (A12)

let base Col = Column (A12)

INDIRECT (ADDRESS (base Row + index - 1
base Col + 2 - 1
4))

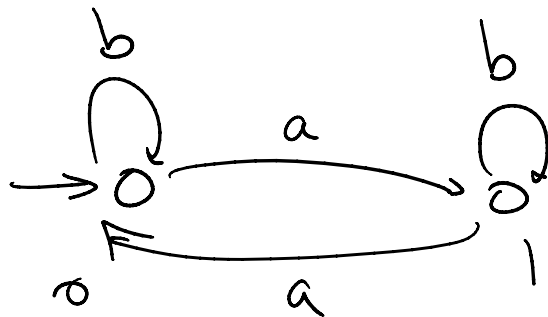
ADDRESS / INDIRECT / Row / COLUMN

ADDRESS / INDIRECT / ROW / COLUMN



VLOOKUP / HLOOKUP

Performing computations in a spreadsheet



	A	B
19		$\text{If } (A_{19} = 'a', \text{abs}(B_{18} - 1), B_{18})$
20		$\text{If } (A_{20} = 'a', \text{abs}(B_{19} - 1), B_{19})$

If ($B_{20} = 'a'$, $abs(C_{19}-1)$,
 C_{19})

