

Closed Sets of Functions and Closed Sets of Relations

Matthew Cook



The goal of this talk:

To explain (with proof) the Galois connection between closed sets of functions and closed sets of relations.

About Relations:

1. Introduce relations and how we will think about them.
2. Explain how we combine relations into networks.
3. Define what we mean by a closed set of relations.
4. Testing implementability: The *Full Network*.

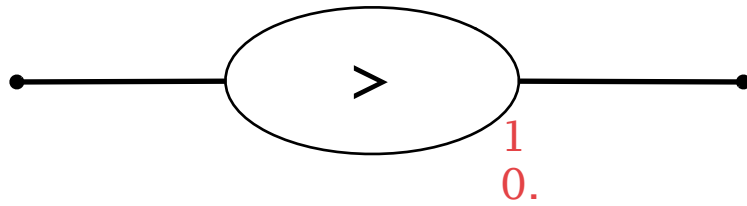
About Functions:

1. Define what we mean by a closed set of functions.

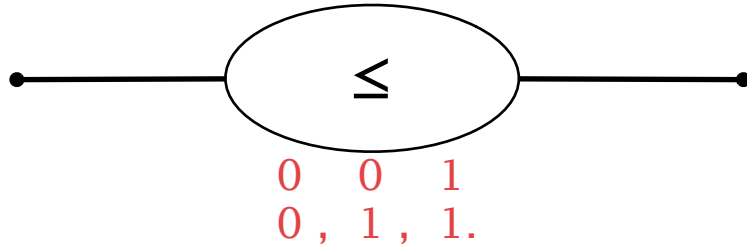
The Connection:

1. Explain what *preservation* is.
2. Show how preservation connects closed sets of functions with closed sets of relations.

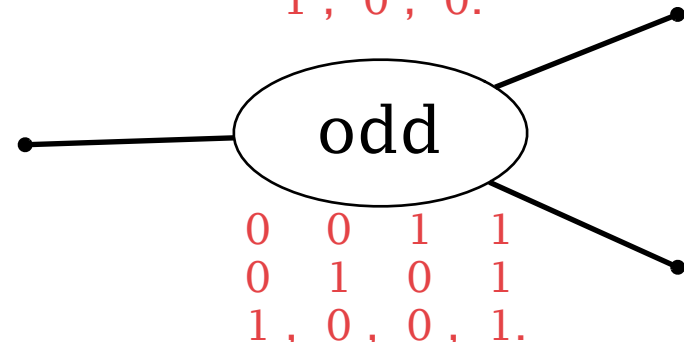
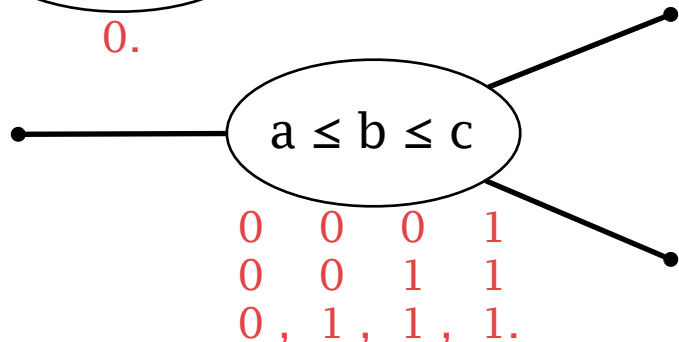
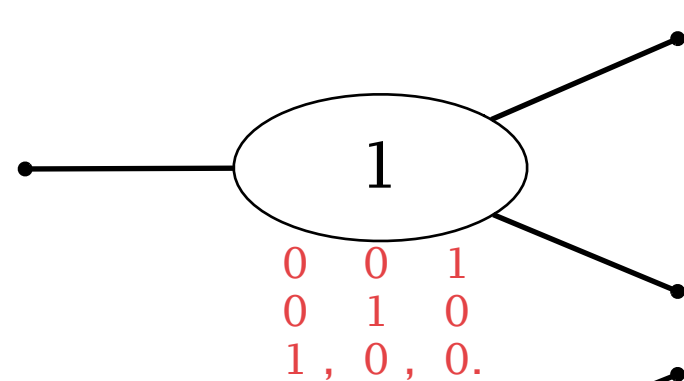
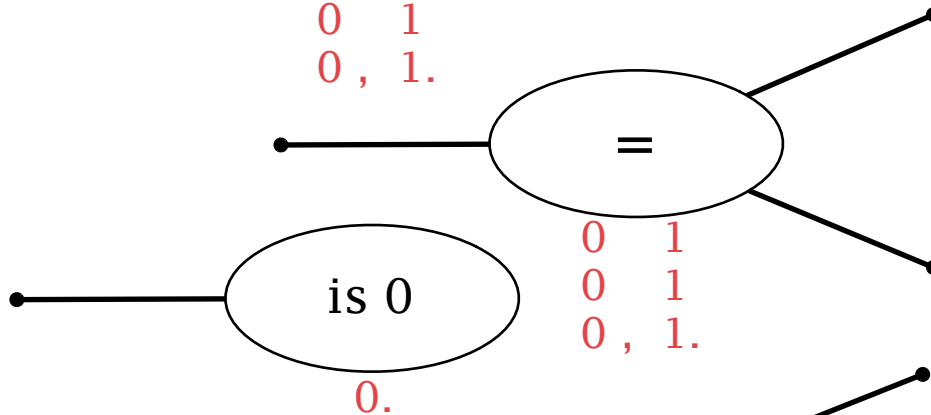
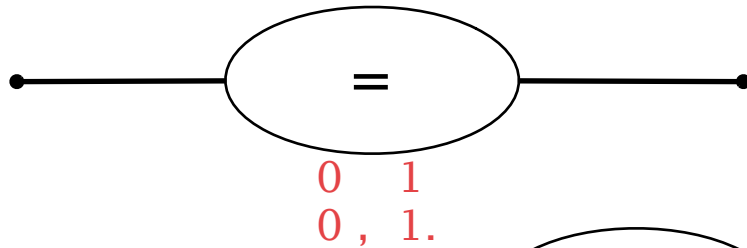
What Are Relations?



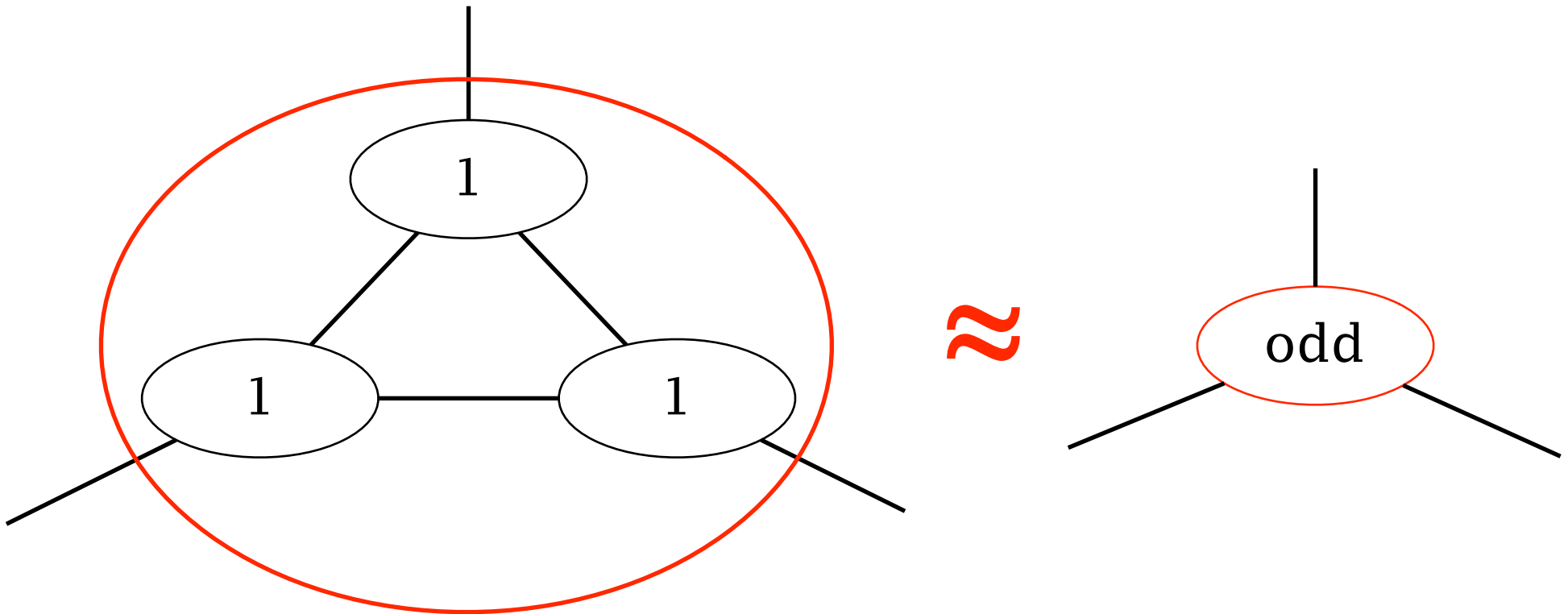
Some combinations of values satisfy the relation. Some do not.



A relation can be defined as the set of all combinations of values that satisfy it.

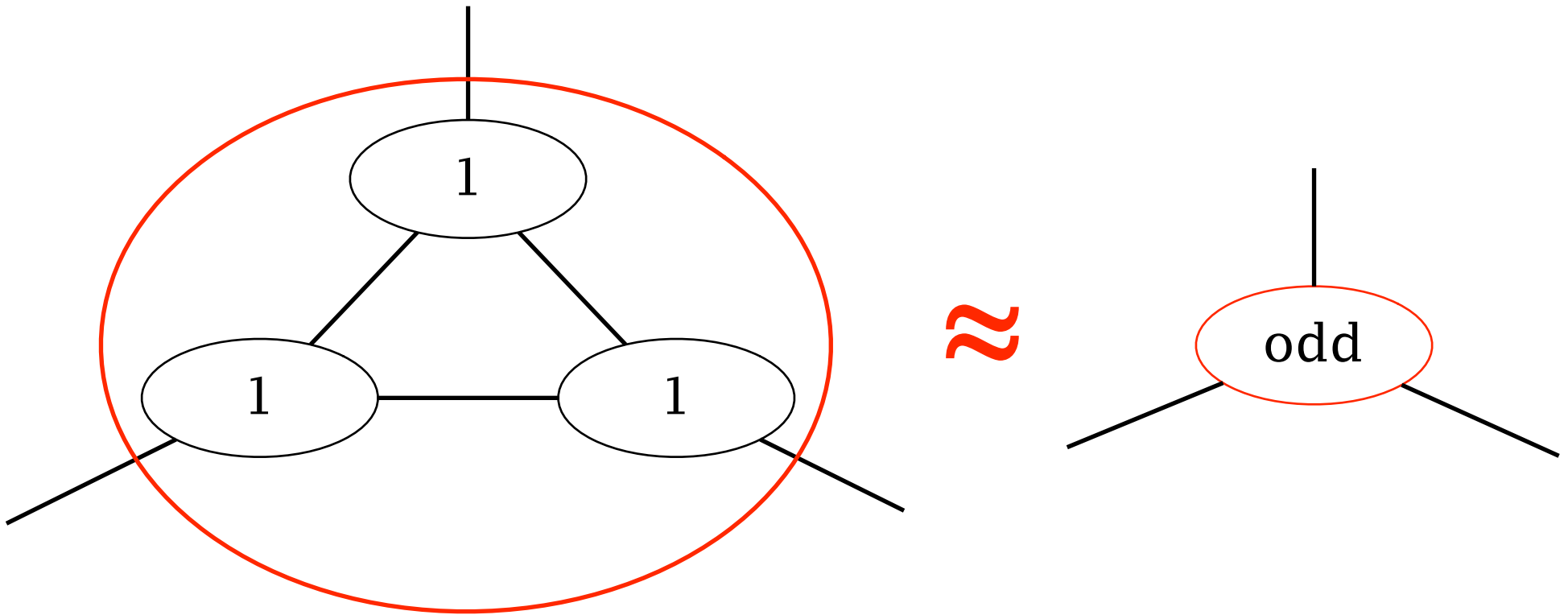


Composition of Relations



A ***network of relations*** (or circuit of relations) is a graph whose vertices represent relations and whose edges represent variables. “Internal edges” have each end connected to a vertex, while “external (dangling) edges” have one end available to the external world.

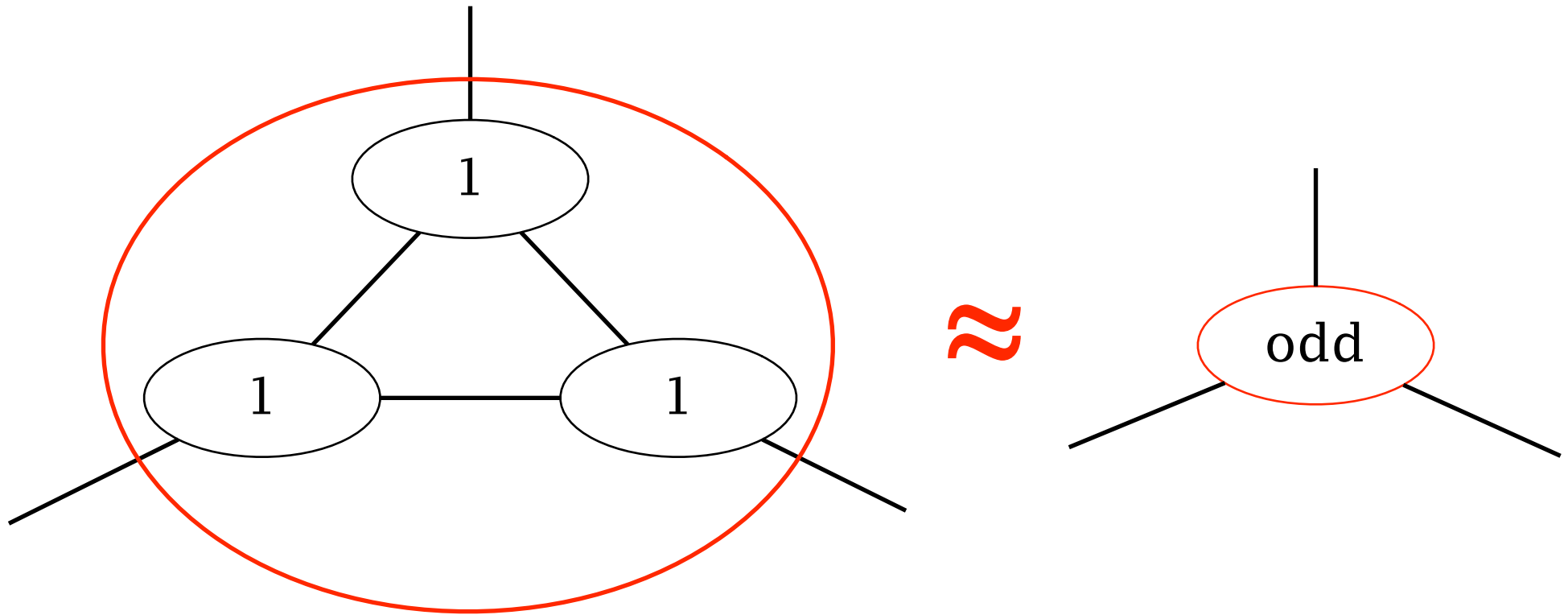
Composition of Relations



What does it mean for a set of values on the external edges to ***satisfy*** a network of relations?

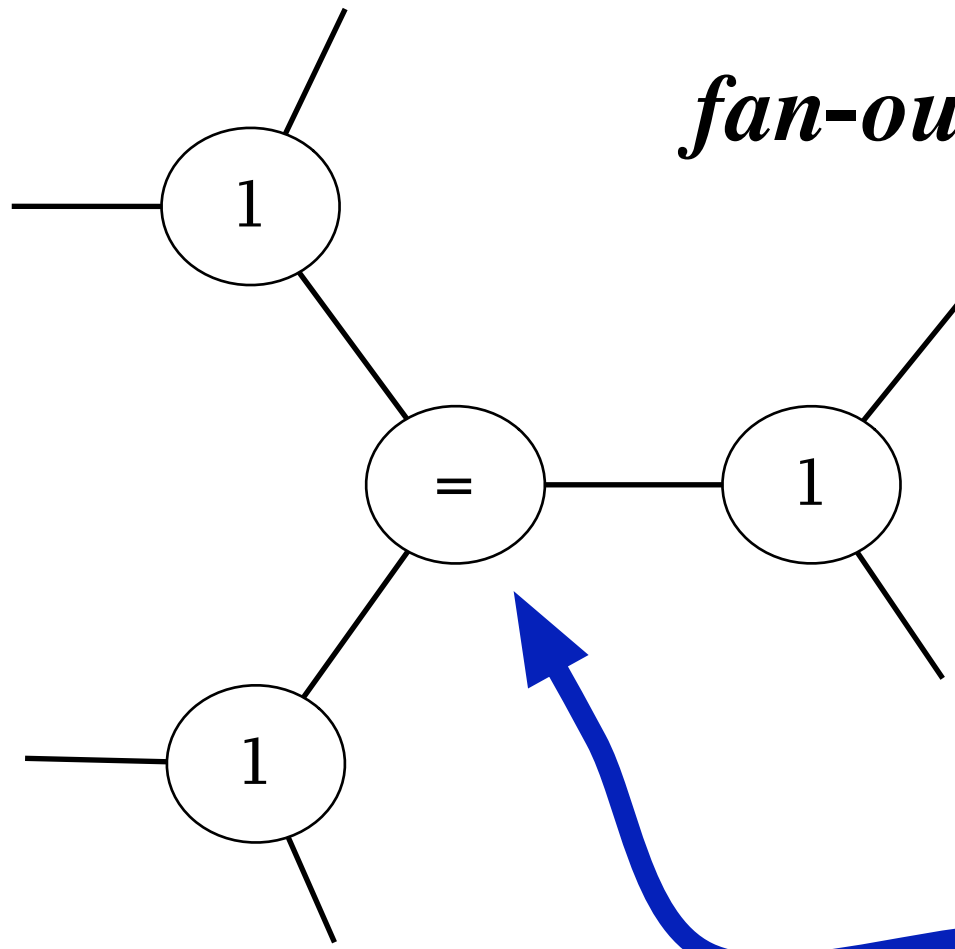
It means that there is some assignment of values to the internal edges for which every relation is satisfied.

Composition of Relations



A network of $\textcircled{1}$ relations can **implement** an $\textcircled{\text{odd}}$ relation.
Can a net of $\textcircled{\text{odd}}$ relations implement a $\textcircled{1}$ relation?

No, because no matter what the network is, it will accept the triple $\langle 1, 1, 1 \rangle$ with every internal wire being 1.



fan-out ... is the equality relation.

What is this?

It is a dot, indicating that the wires are “tied together.”

That is, the same value must be on all three wires.

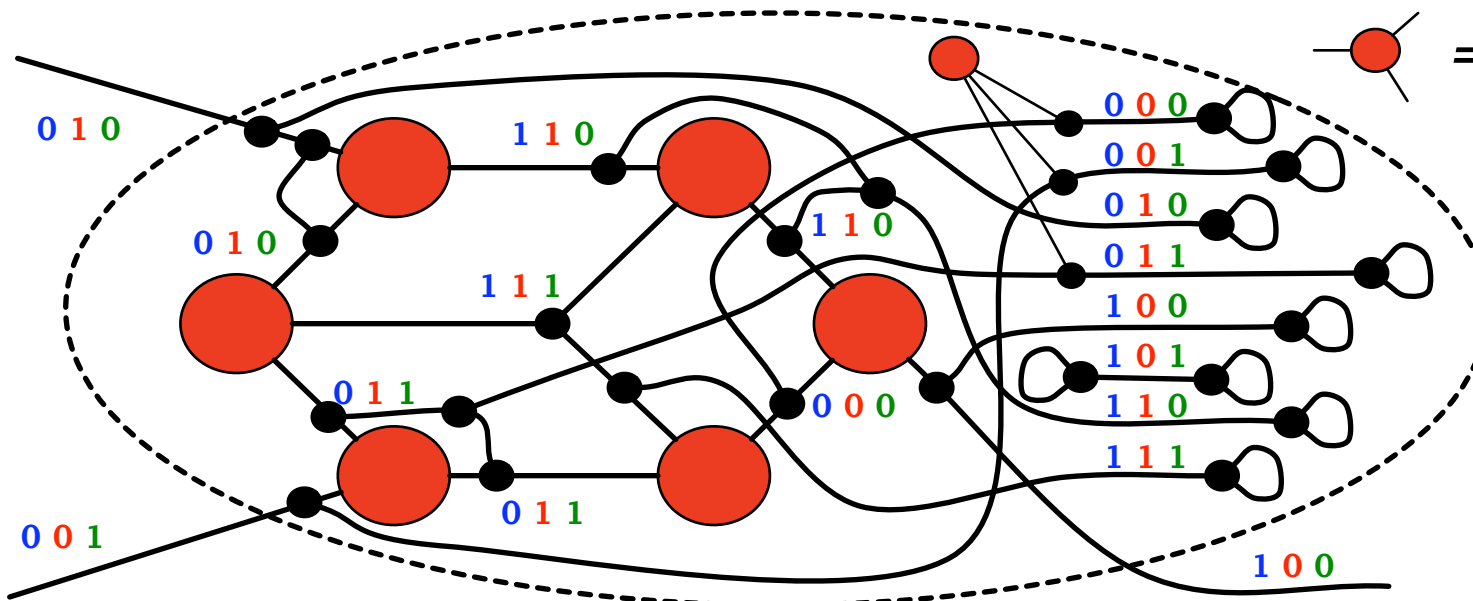
It is an equality relation.

Fan-Out Enables Implementability Testing

Suppose we are given a set of relations, including fan-out, and we want to know whether they can implement a particular target relation T .

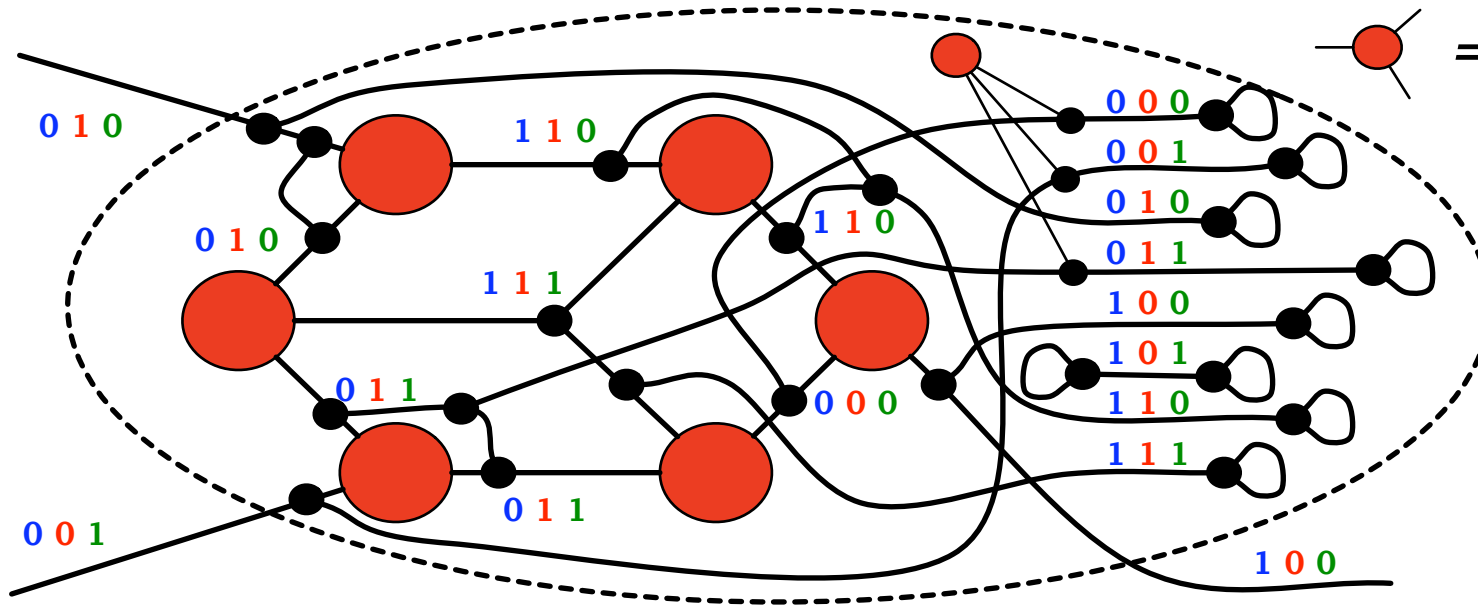
*It turns out that
there is a canonical network that implements T , if anything can.
Let's see why...*

Suppose there is a network that implements T . Then, for each vector acceptable to T , there must be at least one setting of all the wires in the network that satisfies every relation in the network. We will pick such a “certificate” solution for each acceptable vector. Other solutions (besides certificate solutions) are called “extra solutions”.



1. Connect Wires by Label.
2. Eliminate duplicate relations.
3. Add consistent relations.

Fan-Out Enables Implementability Testing

 = $a \leq b \leq c$

1. Connect Wires by Label.
2. Eliminate duplicate relations.
3. Add consistent relations.

To check whether a target T (accepting k vectors) can be implemented:

1. List all the extra solutions alongside the certificate solutions.

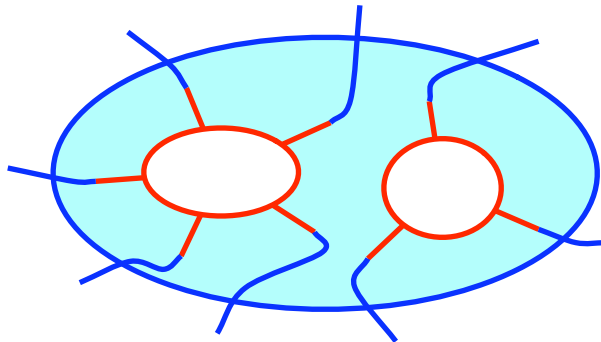
[illegible]

2. Consider the wires that correspond to T 's vectors.
3. Do any of the extra solutions allow tuples T should reject?

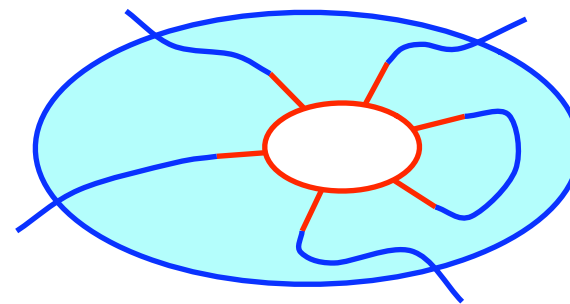
Closed Sets of Relations

- Given some relations (call them *generators*), what can you build?
- The set of all relations implementable with the generators is a *closed set* (closed under composition of relations), because anything implementable with members of the set is also in the set. This set is the *closure* of the generators.
- Note: This definition of a closed set does not depend on generators.
- A Question: Could there be a closed set $\mathbf{R}$ that is not generated by some set of generators? Answer: No, because if we use all the relations in $\mathbf{R}$ as generators, we see that their closure is $\mathbf{R}$.
- By *composition* of relations, we mean any sequence of the following operations:

Juxtaposition



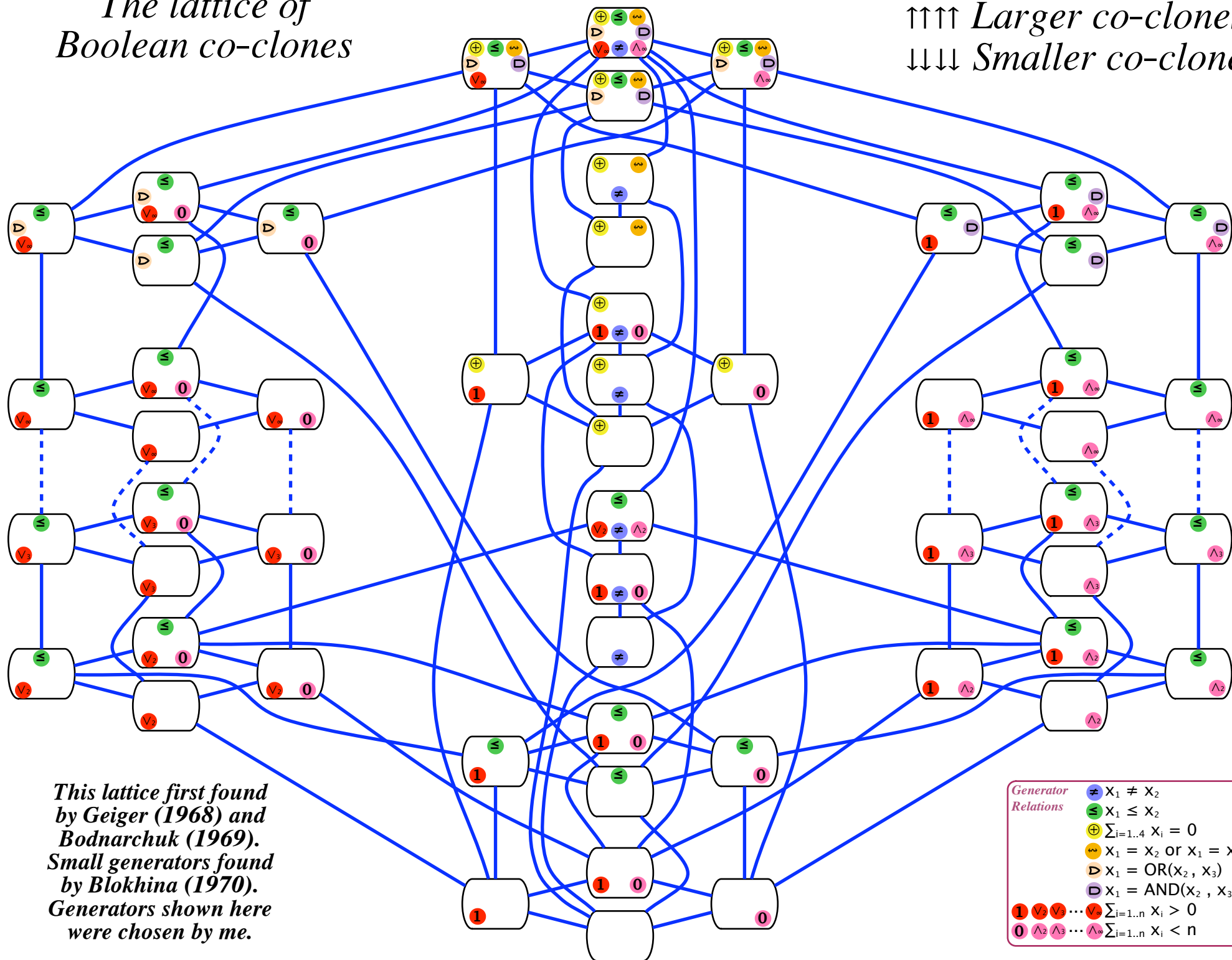
Jumper



- A *co-clone* is a closed set of relations that includes $\bullet$.

The lattice of Boolean co-clones

↑↑↑↑ Larger co-clones
↓↓↓↓ Smaller co-clones



This lattice first found by Geiger (1968) and Bodnarchuk (1969). Small generators found by Blokhina (1970). Generators shown here were chosen by me.

Generator	$\neq$ $x_1 \neq x_2$
Relations	$\leq$ $x_1 \leq x_2$
	$\oplus$ $\sum_{i=1..4} x_i = 0$
	$\approx$ $x_1 = x_2$ or $x_1 = x_3$
	$\triangleright$ $x_1 = \text{OR}(x_2, x_3)$
	$\sqcap$ $x_1 = \text{AND}(x_2, x_3)$
	$\mathbf{1}$ $\vee_2 \vee_3 \dots \vee_n$ $\sum_{i=1..n} x_i > 0$
	$\mathbf{0}$ $\wedge_2 \wedge_3 \dots \wedge_n$ $\sum_{i=1..n} x_i < n$

Closed Sets of Functions

- Given some functions, what other functions can you build?
- Given some gates, what feed-forward circuits can you build?
- A set of functions can be closed under *composition*.

What do we mean by composition?

$$f'(w, x, y, z) = f_0(y, y, f_1(w, y), f_2(x, y, z), z) \quad (\text{Sense used by Post})$$

$\uparrow \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow$

each argument is either a variable or a function

- A *clone* is a closed set of functions that includes all *projection functions*.
- A *projection function* is a function that just returns one of its arguments.

$$f(x_1, x_2, \dots, x_n) = x_k$$

- If projection functions are present (as is the case for clones), then composition can be defined in a more stereotyped way:

$$f'(x_1, x_2, \dots, x_n) = f_0(f_1(x_1, \dots, x_n), f_2(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)) \quad (m \text{ need not equal } n)$$

- Projection functions take care of single variable arguments of previous definition.
- What about arguments that were functions of a subset of the variables?

If we need $f'(w, x, y, z) = f_0(y, y, x)$, again just use projection functions.

Some Examples of Closed Sets of Functions

- Monotonic functions are functions where increasing the arguments can only increase the output.

Any composition of monotonic functions yields another monotonic function.

- “0 at 0” functions are functions where if all the arguments are 0, then the output is 0 too.

Any composition of “0 at 0” functions yields another “0 at 0” function.

- Self-dual functions are functions where if we flip all arguments from 0 to 1 and from 1 to 0, then the output flips too.

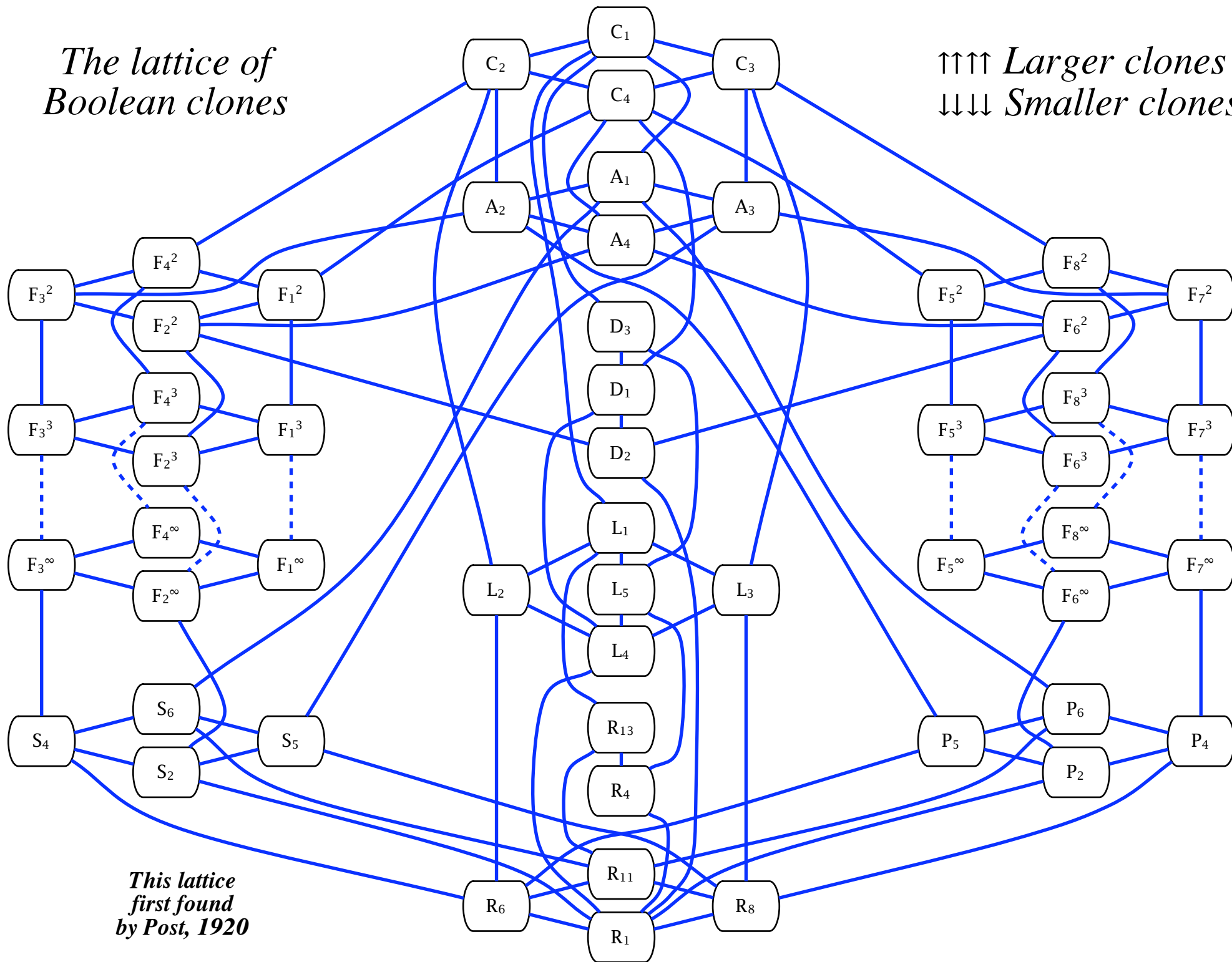
Any composition of self-dual functions yields another self-dual function.

- Linear functions are functions that are a linear combination (mod 2) of some of their arguments.

Any composition of linear functions yields another linear function.

*The lattice of
Boolean clones*

↑↑↑↑ *Larger clones*
↓↓↓↓ *Smaller clones*



*This lattice
first found
by Post, 1920*

Preservation

- A function can *preserve* a relation.

This means that if we go through a sequence of function evaluations where the sequence of values for any given argument satisfies the relation, then sequence of values of the output of the function will also satisfy the relation.

$$\begin{array}{ccccccc} f(x_1, x_2, \dots, x_n) & = & a \\ f(y_1, y_2, \dots, y_n) & = & b \\ f(z_1, z_2, \dots, z_n) & = & c \\ \uparrow & \uparrow & \dots & \uparrow & \uparrow \end{array}$$

- The function has the property that if every column in the arguments ($\langle x_1, y_1, z_1 \rangle$ through $\langle x_n, y_n, z_n \rangle$) satisfies the relation, then the final output column ($\langle a, b, c \rangle$) will also satisfy the relation.
- If $\langle a, b, c \rangle$ is not accepted by the relation, then there must be some k for which $\langle x_k, y_k, z_k \rangle$ is not accepted by the relation.
- The function preserves the relation.

(Those are three ways of saying the same thing.)

Preservation

- Monotonic functions preserve $\leq$.

$$f(x_1, x_2, \dots, x_n) = a$$

$$f(y_1, y_2, \dots, y_n) = b$$

$$\begin{array}{ccccccc} \uparrow & \uparrow & \dots & \uparrow & & \uparrow & \end{array}$$

If $x_1 \leq y_1$ and $x_2 \leq y_2$ and $\dots$ and $x_n \leq y_n$, then $a \leq b$.

- “0 at 0” functions preserve the “is 0” relation.

$$f(x_1, x_2, \dots, x_n) = a$$

$$\begin{array}{ccccccc} \uparrow & \uparrow & \dots & \uparrow & & \uparrow & \end{array}$$

If x_1 is 0 and x_2 is 0 and $\dots$ and x_n is 0, then a is 0.

Preservation

- Self-dual functions preserve $\neq$.

$$\begin{array}{ccccccc} f(x_1, x_2, \dots, x_n) & = & a \\ f(y_1, y_2, \dots, y_n) & = & b \\ \uparrow & \uparrow & \dots & \uparrow & & \uparrow \end{array}$$

If $x_1 \neq y_1$ and $x_2 \neq y_2$ and $\dots$ and $x_n \neq y_n$, then $a \neq b$.

- Linear functions preserve $(\sum_{i=1..n} x_i \bmod 2 = 0)$.

$$\begin{array}{ccccccc} f(w_1, w_2, \dots, w_n) & = & a \\ f(x_1, x_2, \dots, x_n) & = & b \\ f(y_1, y_2, \dots, y_n) & = & c \\ f(z_1, z_2, \dots, z_n) & = & d \\ \uparrow & \uparrow & \dots & \uparrow & & \uparrow \end{array}$$

Changing some inputs of W changes the output iff for any other inputs Y , changing the same inputs also changes the output.

Preservation and Closed Sets

- A set of functions defined by what they preserve is a clone.

$\{ f \mid \forall r \in \mathbf{R}, f \text{ preserves } r \}$ is closed under composition and includes all projection functions.

Some examples:

The set of all monotone functions (functions preserving $\leq$) is a clone.

The set of all zero-preserving functions (preserving “is 0”) is a clone.

The set of all self-dual functions (preserving $\neq$) is a clone.

The set of all self-dual monotone functions (preserving $\leq$ and $\neq$) is a clone.

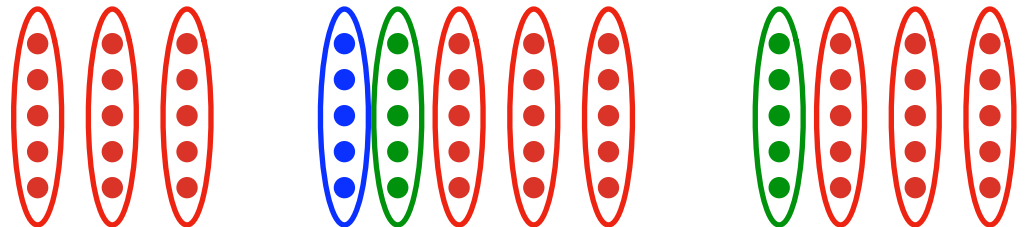
Proof:

Set includes all projection functions? Yes.

$$\begin{array}{ccccccc}
 f(x_1, x_2, \dots, x_n) & = & a \\
 f(y_1, y_2, \dots, y_n) & = & b \\
 f(z_1, z_2, \dots, z_n) & = & c \\
 \uparrow & \uparrow & \dots & \uparrow & & \uparrow \\
 r, & r, & \dots, & r & \Rightarrow & r
 \end{array}$$

Suppose the f_i are in the set... Will f' be in it?

$$f'(x_1, \dots, x_n) = f_0(f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n))$$



Set includes functions generatable (from other f in the set) by composition?

Preservation and Closed Sets

- A set of relations defined by functions preserving them is a co-clone.

$\{ r \mid \forall f \in \mathbf{F}, f \text{ preserves } r \}$ is closed under composition and includes $\neg \bullet \swarrow$.

Example:

$\{ \text{Relations preserved by } \text{MAJ}(x,y,z) \}$ is a co-clone. It includes $=, \neq, \leq, <, \dots$

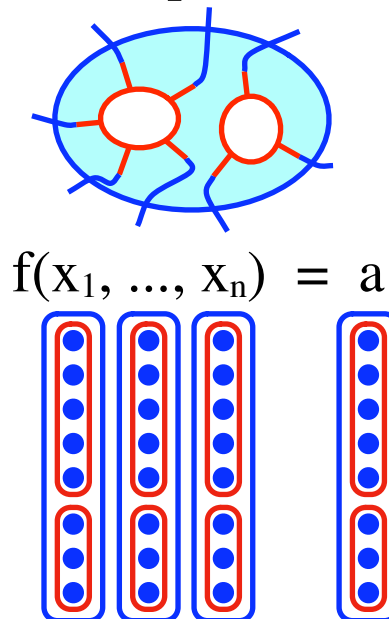
Proof:

Set includes $\neg \bullet \swarrow$? Yes.

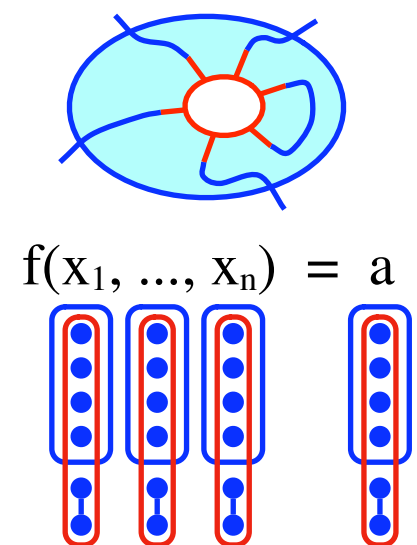
Set includes relations generatable, from other r in the set, by composition?

$$\begin{array}{rcl}
 f(x_1, x_2, \dots, x_n) & = & a \\
 f(y_1, y_2, \dots, y_n) & = & b \\
 f(z_1, z_2, \dots, z_n) & = & c \\
 \uparrow \quad \uparrow \quad \dots \quad \uparrow & & \uparrow \\
 r, \quad r, \quad \dots, \quad r & \Rightarrow & r
 \end{array}$$

Juxtaposition



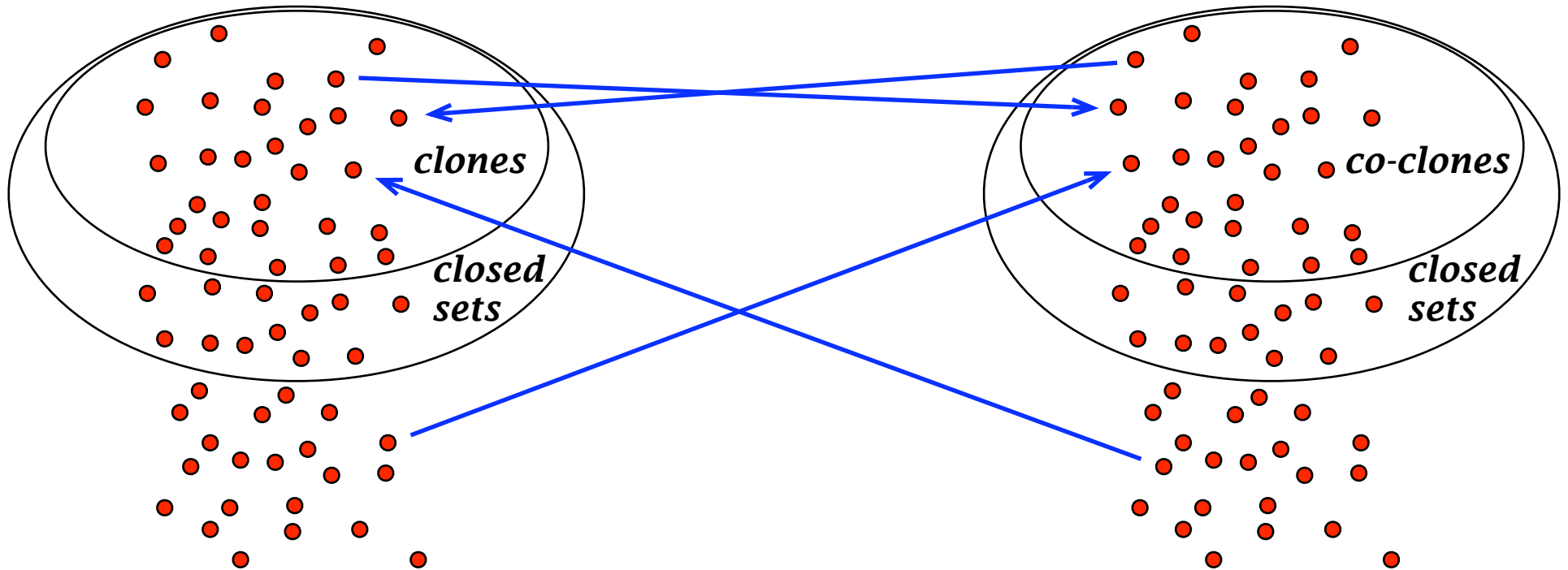
Jumper



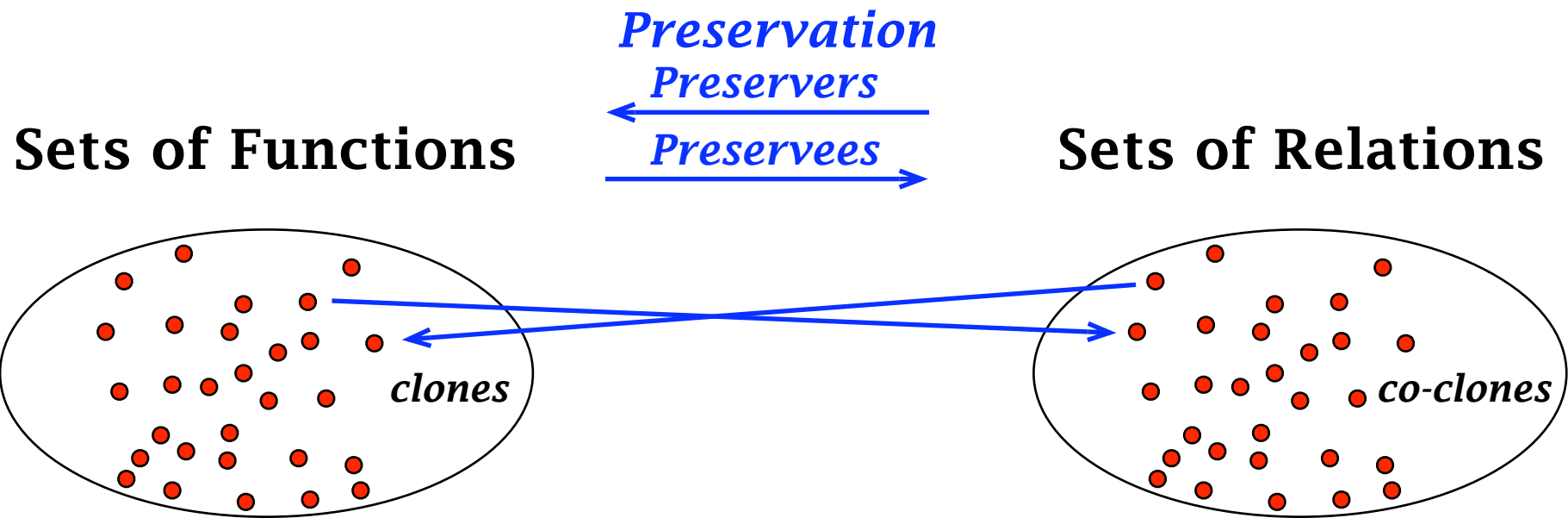
Preservation
Preservers ←
→ *Preserves*

Sets of Functions

Sets of Relations

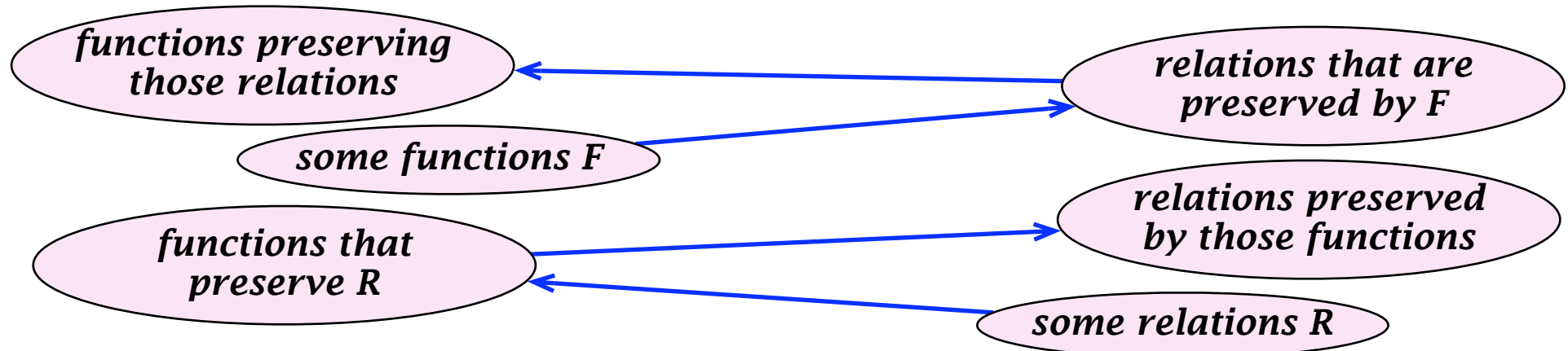


• We want to show that these mappings give a bijection between clones and co-clones. •



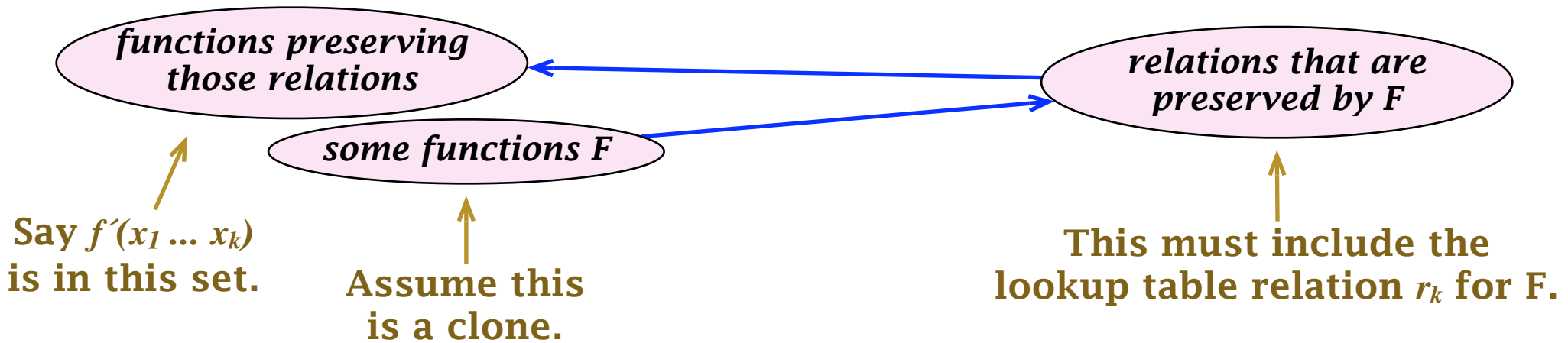
• We want to show that these mappings give a bijection between clones and co-clones. •

Easy Part: When we go back and forth, the set cannot lose any elements.



Harder Part: Why can't the set gain new elements?

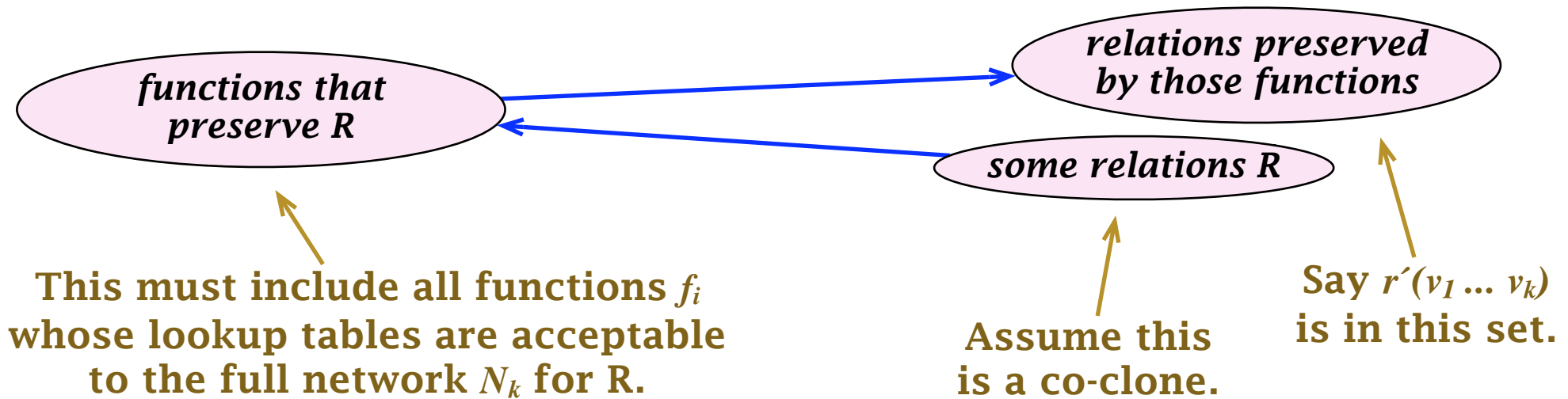
Harder Part: Why can't the set gain new elements?



Can we show that f' must be in the clone F ?

Yes, because f' preserves r_k (Preserving r_k means you're in the clone.)
and
 r_k is preserved by all f in F . (If it doesn't, then F is not a clone.)

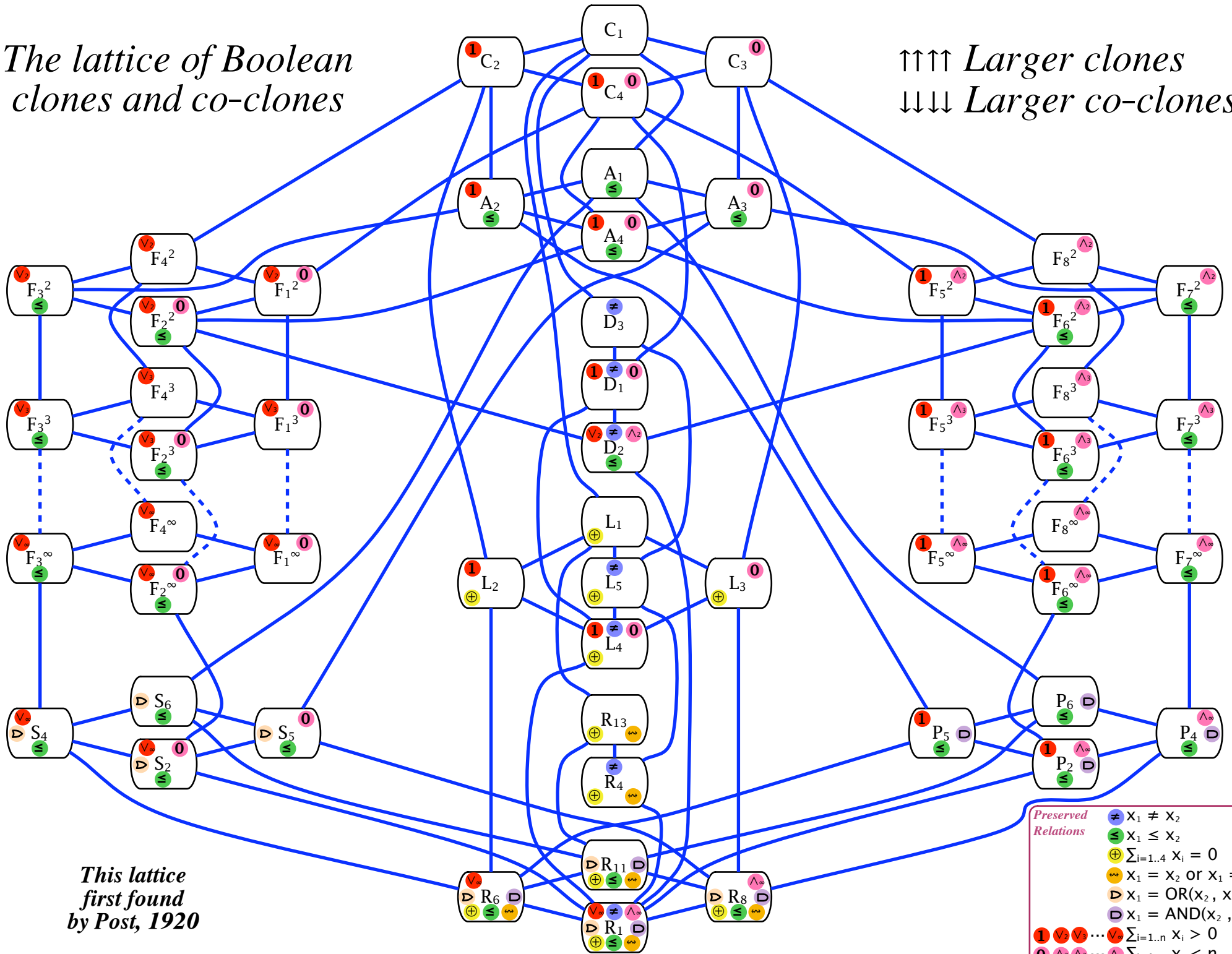
Harder Part: Why can't the set gain new elements?



Can we show that r' must be in the co-clone R ? Yes, because r' is preserved by the functions whose lookup table is acceptable to N_k .

The lattice of Boolean clones and co-clones

↑↑↑↑ Larger clones
↓↓↓↓ Larger co-clones



This lattice first found by Post, 1920