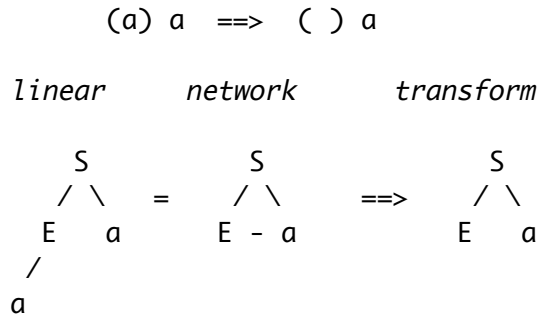


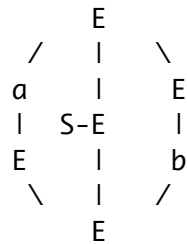
A parallel, graphic representation has only one node for each variable. Since LoF requires two, it is *linear* in its representation. Eg:

Extract:

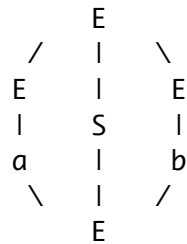


Here's a remarkable consequence: Equivalence has three forms, only two of which are in LoF, because of the linearity of the notation we use.

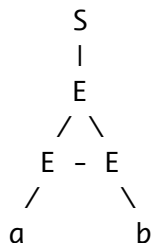
A: (((a) b)((b) a))



B: (a b) ((a) (b))



C: equivalence network



So, the notation is forcing not only duplicate (redundant) variable tokens, it is also forcing *redundant marks*. In form A, if we number the marks:

$$\begin{array}{cccccc} (& (& (a) & b &) & (& (b) & a &) &) \\ 1 & 2 & 3 & & & 4 & 5 & & & \end{array}$$

The network representation shows that marks 2 = 5 and 3 = 4; they are the same distinction.

For a cognitive interpretation:

S: the environment of the structure (handles i/o)

E -- E: a network of *subconscious* processes (no label, no knowledge)

These structures are computable upon using a fully *parallel* process. The order of binding of variables determines whether or not the regime is functional (bound variables) or logical (variable constraints). An imaginary value at a labeled node propagates indeterminate values (errors) through the network.

I don't know whether or not you know these things already, since we are thinking along the same lines.