SHEFFER STROKE

William Bricken

June 1987

The single operator notation of Sheffer stroke explodes the representation of a logical form, and therefore enjoys none of the benefits of Losp.

The problem is that stroke forces the unary operator NOT to be binary.

Here's a Sheffer-Losp map. From it, it's easy to see where Sheffer went wrong.

Logic	*Sheffer*	*/Losp*
NOT a	a | a (a a)	= (a)
a OR b	(a | b) | (a | b) ((a b) (a b))	= ((a b)) = a b
a AND b	(a | a) | (b | b) ((a a) (b b))	= ((a)(b))
a IMPLIES b	((a | a) | b) | ((a | a) | b) (((a a) b) ((a a) b))	= (((a) b)((a) b)) = (((a) b)) = (a) b
a IFF b	(((a|a)|(b|b)) | (a|b)) | (((a|a)|(b|b)) | (a|b)) ((((a a) (b b)) (a b)) (((a a) (b b)) (a b))) = ((((a) (b)) (a b)) (((a) (b)) (a b))) = ((((a) (b)) (a b))) = ((a) (b)) (a b)	

The transcription rule is to erase the stroke and leave the parentheses, which then become boundary operators. Usually the outermost stroke needs an outermost parenthesis added. Then remove all the redundancy introduced by the stroke. This shows up as duplicate representations of the same expression.

Stroke is a binary operator, and therefore requires brackets to maintain precedence. Note how this simple restriction introduces extreme representational redundancy.