A LOSP PROOF

William Bricken
March 1985

The following symbolic problem was selected from Klenk, Understanding Symbolic Logic, p. 194. The conclusion can be inferred from the premises in about 20 steps with a firm understanding of standard logic, and around 10 logical inference and transformation rules.

The steps within a Losp solution are:

1. Transcribe the logical expressions into Losp.
2. Apply Losp simplification rules.
3. Interpret the result.

What is surprising is the utter simplicity both of the mapping from Losp to Propositional Calculus, and of the Losp simplification rules.

THE LOSP-PROPOSITIONAL CALCULUS MAP

PROPOSITIONAL CALCULUS	*LOSP EXPRESSIONS*
true	( )
not A	(A)
A or B	A B
A and B	((A) (B))
if A then B	(A) B

THE LOSP RULES OF SIMPLIFICATION

INVOLUTION:	((a))	==>	a
REPLICATION:	a a	==>	a
EXTRACT:	a (a b)	==>	a (b)
DOMINION:	a ( )	==>	( )

These rules are applied in an abstract computational space that is inherently associative, commutative, and n-ary. That is, the space of representation is insensitive to the position and number of expressions.

THE SYMBOLIC PROBLEM

```
Premise 1:        if (A or B) then (C and D)

Premise 2:        if (A and C) then (if D then (not E))

Premise 3:        if F then (not F)

Premise 4:        if (not (E or F)) then (not D)

Conclusion?:      (not A)
```

I will show every step of the calculation in detail.

STEP I: Transcribe into Losp

```
Premise 1:        (A B) ((C) (D))

Premise 2:        ( ((A) (C)) ) (D) (E)

Premise 3:        (F) (F)

Premise 4:        ( (E F) ) (D)

Conclusion:       (A)
```

The relationship between Premise and Conclusion can be expressed by logical operators:

```
if (P1 and P2 and P3 and P4) then Conclusion
```

Transcribing this into Losp:

```
(( (P1) (P2) (P3) (P4) )) Conclusion
```

Finally, substituting the Losp expressions for the Premises and Conclusion:

```
(( ((A B)((C)(D))) ( (((A)(C)))(D)(E) ) ((F)(F)) (((E F))(D)) )) (A)
```

This then is the Losp representation of the logic problem. The primary feature is that it has one explicit operator, the parens.

STEP II: Apply Losp simplification rules

All transformation rules delete components of the initial expression. The evolving forms are aligned in this presentation for ease of seeing what is deleted.

Simplify

```
  (( ((A B)((C)(D))) ( (((A)(C)))(D)(E) ) ((F)(F)) (((E F))(D)) )) (A)   prob
     ((A B)((C)(D))) (   (A)(C)  (D)(E) ) ((F)(F)) (  E F  (D))    (A)    inv
     (     ((C)(D))) (      (C)  (D)(E) ) ((F)(F)) (  E F  (D))    (A)    ext
            (C)(D)   (      (C)  (D)(E) ) ((F)(F)) (  E F  (D))    (A)    inv
            (C)(D)   (           (D)(E) ) ((F)(F)) (  E F  (D))    (A)    inv
            (C)(D)   (              (E) ) ((F)(F)) (  E F     )    (A)    ext
            (C)(D)                   E    ((F)(F)) (  E F     )    (A)    inv
            (C)(D)                   E    ((F)(F)) (    F     )    (A)    ext
            (C)(D)                   E    (      ) (    F     )    (A)    ext
                                          (      )                        dom
```

STEP III: Interpret the result

The value of the expression is parens, which is interpreted for logic as TRUE, therefore the entire expression is a tautology. The Conclusion follows from the Premises.

NOTES

1. If the Conclusion did NOT follow, the expression would *not* simplify to parens. If the original expression was a contradiction, it would literally vanish. If it was a contingency, the expression would simplify to the simplest form of that contingency.

2. The characteristic decision method of alternating INVOLUTION and EXTRACT with DOMINANCE as the termination condition is algorithmically simple.

3. EXTRACT and INVOLUTION are independent, thus they can be applied in any order to yield the same result. Similarly, the order of erasing extractable expressions is irrelevant.

4. The EXTRACT (A) step also extracts tokens in the same space as A, thus, the (A B) is extracted. The formal justification of this is the Losp distributive law (details omitted).

5. Larger expressions could have been extracted, speeding the convergence on the solution. For instance, at the EXTRACT (C) step, the expression (C)(D) could have been extracted.