

## Appendix

## A. Complete Axiom Derivations

### A.1 Calling Axiom (J1) Proof

**Statement:**  $\langle\langle a \rangle\rangle = a$

**Spatial Interpretation:** Consider a space with form  $a$ . Enclosing  $a$  creates  $\langle a \rangle$ —we are now “outside”  $a$  (inside the boundary around  $a$ ). Enclosing again creates  $\langle\langle a \rangle\rangle$ —we are now “outside” of being “outside”  $a$ , which returns us to  $a$ .

**Algebraic Proof:** Let  $\cdot$  denote truth value evaluation. -

$$\begin{aligned}\langle\langle a \rangle\rangle &= \neg\langle a \rangle \text{ (by enclosure semantics)} \\ - &= \neg\neg a \text{ (by enclosure semantics again)} \\ - &= a \text{ (by double negation)}\end{aligned}$$

Since truth values are preserved and the calculus is sound,

$$\langle\langle a \rangle\rangle = a.$$

### A.2 Crossing Axiom (J2) Proof

**Statement:**  $\langle \rangle \langle \rangle = \langle \rangle$

**Spatial Interpretation:** Two marks side by side both indicate “the marked state.” Indicating the same state twice does not change what is indicated.

**Algebraic Proof:**  $- \langle \rangle \langle \rangle = \langle \rangle \wedge \langle \rangle$  (by juxtaposition semantics) -

$$= \text{TRUE} \wedge \text{TRUE} \text{ (mark is TRUE)} = \text{TRUE} = \langle \rangle$$

## B. Consequence Derivations

### B.1 C1: Position

**Statement:**  $\langle\langle a \rangle b \rangle a = a$

**Derivation:** Consider the Boolean interpretation:  
- LHS =  
 $\neg(\neg a \wedge b) \wedge a$  - =  $(a \vee \neg b) \wedge a$  (De Morgan)  
- =  $a \wedge (a \vee \neg b)$   
(commutative) - =  $a$  (absorption)

### B.2 C3: Generation (Law of Excluded Middle)

**Statement:**  $\langle\langle a \rangle a \rangle = \langle \rangle$

**Derivation:** - LHS =  $\langle\langle a \rangle a \rangle$  - =  $\neg(\neg a \wedge a)$  (Boolean interpretation)  
- =  $\neg(\text{FALSE})$  (contradiction) - = TRUE - =  $\langle \rangle$   
This confirms that  $a \vee \neg a = \text{TRUE}$ .

### B.3 C6: Iteration (Idempotence)

**Statement:**  $aa = a$

**Derivation:** -  $aa = a \wedge a$  (juxtaposition) - =  $a$  (idempotence of AND)

## C. Boolean Algebra Correspondence

### C.1 Complete Translation Table

Boolean	Boundary Form	Reduction
TRUE	$\langle \rangle$	canonical
FALSE	void	canonical
$\neg a$	$\langle a \rangle$	—
$a \wedge b$	$ab$	—
$a \vee b$	$\langle\langle a \rangle\langle b \rangle\rangle$	—
$a \rightarrow b$	$\langle a \langle b \rangle \rangle$	—
$a \leftrightarrow b$	$\langle\langle ab \rangle\langle\langle a \rangle\langle b \rangle\rangle\rangle$	—
$a \oplus b$ (XOR)	$\langle\langle\langle a \rangle b \rangle\langle a \langle b \rangle \rangle \rangle$	—
$a$ NAND $b$	$\langle ab \rangle$	—
$a$ NOR $b$	$\langle\langle\langle a \rangle\langle b \rangle \rangle \rangle$	—

### C.2 NAND Completeness

All Boolean operations expressible via NAND ( $\langle ab \rangle$ ):

- ▶ NOT  $a = a$  NAND  $a = \langle aa \rangle = \langle a \rangle$

## D. Reduction Algorithm Details

### D.1 Pattern Matching

#### Calling Pattern:

Match: Form with `is_marked=True`, `contents=[Form with is_marked=True]`

Result: a

#### Crossing Pattern:

Match: Form with multiple simple marks in contents

Result: Single mark with non-mark contents preserved

### D.2 Trace Format

Each reduction step records:

#### ReductionStep:

- `before`: Form (pre-reduction)
- `after`: Form (post-reduction)
- `rule`: ReductionRule (CALLING | CROSSING | VOID\_ELIMINATE)
- `location`: str (where rule applied)

### D.3 Termination Proof

**Theorem:** The reduction algorithm terminates for all well-formed inputs.

## E. Test Coverage Details

### E.1 Test Categories

Category	Tests	Coverage Target
Unit (forms.py)	36	95%+
Unit (reduction.py)	27	95%+
Unit (algebra.py)	22	90%+
Integration	15	90%+
Theorem verification	12	100%
Edge cases	18	Comprehensive

### E.2 Property-Based Testing

Random form generation tests:

- Depth: 1-6 (uniform)
- Width: 1-4 (uniform)
- Samples: 500 per test run
- Seed: 42 (reproducible)

Verified properties:

- All forms reduce to canonical
- Canonical forms are stable (re-reduction yields same)
- Equivalent forms have equal canonical forms

## F. Notation Reference

---

Symbol	Meaning	LaTeX
$\langle \rangle$	Mark (TRUE)	<code>\langle\!\!\langle \rangle\!\!\rangle</code>
$\emptyset$	Void (FALSE)	<code>\emptyset</code>
$\langle a \rangle$	Enclosure (NOT)	<code>\langle\!\!\langle a \rangle\!\!\rangle</code>
$ab$	Juxtaposition (AND)	<code>ab</code>
$f$	Truth value	<code>\llbracket f \rrbracket</code>
$j$	Imaginary value	<code>j</code>

---

## G. Implementation Reference

### G.1 Module Structure

```
project/src/
    forms.py      # Form class and construction
    reduction.py  # Reduction engine
    algebra.py    # Boolean correspondence
    evaluation.py # Truth value extraction
    theorems.py   # Theorem definitions
    verification.py # Formal verification
    visualization.py # Diagram generation
    __init__.py    # Package exports
```

### G.2 Key APIs

```
# Form construction
make_void() -> Form
make_mark() -> Form
enclose(form: Form) -> Form
juxtapose(*forms: Form) -> Form
```

```
# Reduction
```