Theorem 0: Every well-formed formula starts with a propositional variable or an opening bracket.

Theorem 1: Every well-formed formula has an equal number of opening and closing brackets.

Theorem 2: Every proper prefix of a well-formed formula has more opening brackets than closing brackets.

**Theorem: There is a unique way to construct every well-formed formula.**

Proof by structural induction:

Let x be a well-formed formula. We want to prove that there is a unique way to construct x as a well-formed formula.

Base case: x is a propositional variable.

Can we construct x as (Øa) for a well-formed formula a by applying negation as the last step?

Can we construct x as (a\*b) for well-formed formulas a and b by applying a binary connective \* as the last step?

Induction step:

Case 1: x is (Øa) for a well-formed formula a.

Induction hypothesis: Assume that there is a unique way to construct a. We need to prove that there is a unique way to construct (Øa).

We already know one way to construct (Øa): construct a, and apply negation as the last step. We need to show that there is no other way to construct (Øa).

Can we construct (Øa) as a propositional variable?

Can we construct (Øa) as (c\*d) for well-formed formulas c and d by applying a binary connective \* as the last step?

Case 2: x is (a\*b) for well-formed formulas a and b where \* is one of Ù, Ú, ®, and «.

Induction hypothesis: Assume that there is a unique way to construct a and b respectively. We need to prove that there is a unique way to construct (a\*b).

We already know one way to construct (a\*b): construct a and b separately, and apply \* as the last step. We need to show that there is no other way to construct (a\*b).

Can we construct (a\*b) as a propositional variable?

Can we construct (a\*b) as (Øc) for well-formed formula c by applying negation as the last step?

Can we construct (a\*b) as (c@d) for well-formed formulas c and d by applying a binary connective @ that is different from \* as the last step?

QED