Lemma 1: Every well-formed formula starts with a propositional variable or an opening bracket.

Lemma 2: Every well-formed formula has an equal number of opening and closing brackets.

Lemma 3: 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.

<u>Can we construct x as (¬a) for a well-formed formula a by applying negation as the last step?</u>

If we construct a formula by applying negation as the last step, then it has to be of the form (¬a) and has to contain at least 3 symbols. However, the formula x only has 1 symbol. Therefore, we cannot construct x by applying negation as the last step.

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

If we construct a formula by applying negation as the last step, then it has to be of the form (¬a) and has to contain at least 3 symbols. However, the formula x only has 1 symbol. Therefore, we cannot construct x by applying negation as the last step.

Induction step:

Case 1: x is (¬a) 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 (¬a).

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

<u>Can we construct (¬a) as a propositional variable?</u>

If we construct a formula as a propositional variable, then it has to have 1 symbol. However, the formula (¬a) has at least 3 symbols. So we cannot construct (¬a) as a propositional variable.

<u>Can we construct (¬a) as (c*d) for well-formed formulas c and d by applying a binary connective * as the last step?</u>

Suppose that we can construct (¬a) as (c*d) for well-formed formulas c and d by applying a binary connective * as the last step. Then the connective * has to be in the formula a. Let a = m*n. Then c = ¬m and d = n. We will argue that c is not a well-formed formula.

m is a proper prefix of the well-formed formula a. By Lemma 3, m has more opening than closing brackets. Thus, c also has more opening than closing brackets. By Lemma 2, c is not a well-formed formula.

Therefore, we cannot construct (¬a) 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 $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.

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).

<u>Can (a*b) be constructed as a propositional variable?</u>

If we construct a formula as a propositional variable, then it has to have 1 symbol. However, the formula (a*b) has at least 3 symbols. So we cannot construct (a*b) as a propositional variable.

<u>Can (a*b) be constructed as ($\neg$c) for well-formed formula c by applying negation as the last step?</u>

Suppose that we can construct (a*b) by applying negation as the last step. Then the binary connective * has to be in c. Let c = m*n. Then a = $\neg$m and b = n. We will argue that a is not a well-formed formula.

m is a proper prefix of the well-formed formula c. By Lemma 3, m has more opening than closing brackets. Thus, a also has more opening than closing brackets. By Lemma 2, a is not a well-formed formula.

<u>Can (a*b) be constructed as (c@d) for well-formed formulas c and d by applying a binary connective @ that is different from * as the last step?</u>

Suppose that we can construct (a*b) by applying a different binary connective @ as the last step. Then the binary connective @ has to be either in a or in b.

If the binary connective @ is in a, then c is a proper prefix of a. By Lemma 3, c has more opening than closing brackets. Thus, c is not a well-formed formula.

If the binary connective @ is in b, then let b = m@n. Then c = a*m and d = n. Let op(x) and cl(x) denote the number of opening and closing brackets in a formula x.

a is a well-formed formula, so op(a) = cl(a) by Lemma 2.
m is a proper prefix of the well-formed formula b, so op(m) > cl(m).
By inspection of c, op(c) = op(a) + op(m) > cl(a) + cl(m) = cl(c).
Thus, c has more opening than closing brackets. By Lemma 3, c is not a well-formed formula.

QED