Abstract Syntax
- an expression is a tree with every node labeled with a character string

Concrete Syntax
- a character string

Variables
- strings are divided into "variable" and "non-variables".

A inference rule has
1) A sequence of "antecedent"
2) A conclusion expression

\[ R = \{ f(x), f(g(x,x)) \} \]
\( \text{rule(ants(f(x)), f(g(x,x)))} \)
\( \vdash_R f(a) \)
\( \vdash_R f(g(a,a)) \)
\( \vdash_R f(g(g(a,a), g(a,a))) \)

A substitution is a mapping from variables to expressions.
If \( \sigma \) is a substitution, then we write \( \sigma(x) \) for the expression \( \sigma \) assigns to variable \( x \).
If \( e \) is an expression, then \( \sigma(e) \) is the result of replacing each variable \( x \) in \( e \) with \( \sigma(x) \).

Consider a rule

\[
\begin{array}{c}
A_1 \\
\vdots \\
A_n \\
\hline \\
C
\end{array}
\]

consider a substitution \( \sigma \).
If

\[
\begin{array}{c}
A_1 \\
\vdots \\
A_n \\
\hline \\
C
\end{array}
\]

is in \( R \) and \( \sigma \) is a substitution and \( \vdash_R \sigma(A_1) \ldots \sigma(A_n) \), then \( \vdash_R \sigma(C) \).
ex) \( \vdash_R f(a) \quad \sigma(x) = a \quad \sigma(x) = g(a, a) \)
\( \vdash_R f(g(a, a)) \quad \vdash_R f(g(g(a, a), g(a, a))) \)