What does it mean for a unification algorithm to be correct?
What is the relationship between the input and output?
   ▶ Input: a set of equations.
   ▶ Output: a solution which maps type variables to types (it is a substitution).
Does the algorithm terminate?
A substitution $S$ is a *unifier of an equation* $T_1 \equiv T_2$ if $S(T_1) = S(T_2)$, i.e., the results of substitution are syntactically equal.

Example: $\{\alpha \mapsto \text{bool}, \beta \mapsto \text{int}\}$ unifies $\alpha \rightarrow \text{int} \equiv \text{bool} \rightarrow \beta$.

A *unifier of a set of equations* $E$ is a substitution $S$ that unifies every equation in $E$. 
The unification algorithm takes one step at a time, simplifying a set of equation $E$ to a new set $E'$. We write $E \rightarrow E'$ for one of these steps.

The unification algorithm ends when none of the rules applies to the current equation set, i.e. $\neg \exists E'. E \rightarrow E'$.

We can read off a solution from a set of equations $E$ if it is on solved form:

1. All the equations have the form $\alpha = T$.
2. If a variable occurs on the left of an equation, it does not occur anywhere else.

$$\{\alpha_1 = T_1, \ldots, \alpha_n = T_n\} \rightarrow \{\alpha_1 \mapsto T_1, \ldots, \alpha_n \mapsto T_n\}$$
Lemma

If $\neg \exists E'. E \rightarrow E'$ then $E$ is in solved form.
The output $S$ is obviously a unifier for the final set of equations, call it $E_f$.

$E_f = \{ \alpha_1 = T_1, \ldots, \alpha_n = T_n \}$

$S = \{ \alpha_1 \mapsto T_1, \ldots, \alpha_n \mapsto T_n \}$

$S(E_f) = \{ T_1 = T_1, \ldots, T_n = T_n \}$

But is $S$ a unifier for the initial set of equations, $E_0$?

$(E_0 \xrightarrow{n} E_f)$

How can we prove that it is?
Is the output a unifier of the equations?

**Lemma**

If \( E \rightarrow E' \) and \( S \) unifies \( E' \) then \( S \) unifies \( E \).

**Theorem (Soundness of the unification algorithm)**

If \( E \rightarrow^* E' \) and \( S \) unifies \( E' \) then \( S \) unifies \( E \).
If there is a solution, will the algorithm find it?

▶ Suppose there is a solution $S$ that unifies $E$. When we run the algorithm, will it stop with a set $E_f$ that is in solved form?
▶ What is the relationship between $E_f$ and $S$?
A *most general unifier* of $E$ is a substitution $S$ that unifies $E$ and, for any other substitution $R$ that unifies $E$, there exists $U$ such that $U \circ S = R$.

Example: let $E$ be $\{\alpha \doteq \beta, \gamma \doteq \alpha \rightarrow \beta\}$. Then $S = \{\alpha \mapsto \beta, \gamma \mapsto \beta \rightarrow \beta\}$ is a most general unifier of $E$.

Another solution of $E$ is $R = \{\alpha \mapsto \text{int}, \beta \mapsto \text{int}, \gamma \mapsto \text{int} \rightarrow \text{int}\}$ but have $\{\beta \mapsto \text{int}\} \circ S = \{\alpha \mapsto \text{int}, \beta \mapsto \text{int}, \gamma \mapsto \text{int} \rightarrow \text{int}\} = R$.

Another solution of $E$ is $R = \{\alpha \mapsto \text{bool}, \beta \mapsto \text{bool}, \gamma \mapsto \text{bool} \rightarrow \text{bool}\}$ but have $\{\beta \mapsto \text{bool}\} \circ S = \{\alpha \mapsto \text{bool}, \beta \mapsto \text{bool}, \gamma \mapsto \text{bool} \rightarrow \text{bool}\} = R$. 
Lemma

If $S$ is a unifier of $E$, then either $E$ is in solved form or there is an $E'$ such that $E \rightarrow E'$ and $S$ is a unifier of $E'$.

Lemma

If $S$ is a unifier of $E$ and $E$ is in solved form, then the solution $S'$ read from $E$ is more general than $S$: there is an $R$ such that $R \circ S' = S$.

Theorem (Completeness)

If $S$ is a unifier of $E$ then there exists an $E_f$ such that $E \rightarrow^* E_f$ such that $E_f$ is in solved form, and the solution $S_f$ read from $E_f$ is the most general unifier.
Most proofs of termination associate a number with all the state used by an algorithm, and show that this number shrinks with each step of the algorithm.

This association is called a measure function.

We need to come up with a measure function $m$ on a set of equations and prove the following lemma.

**Lemma**

If $E \rightarrow E'$ then $m(E) < m(E')$. 
Measure function

\[ m(E) = (n_1, n_2, n_2) \]

- \( n_1 \) is the number of variables in \( E \) that do not occur only once as the left-hand side of some equation.
- \( n_2 \) is the total size of all the equations in \( E \).
- \( n_3 \) is the number of equations of the form \( \alpha = \alpha, \text{int} = \text{int}, \) and \( T = \alpha \).

The ordering relation \( < \) that we use to compare tuples is the lexicographical ordering:

\[
(n_1, n_2, n_2) < (n'_1, n'_2, n'_2) = n_1 < n'_1 \lor ((n_1 = n'_1 \land n_2 < n'_2) \\
\lor (n_1 = n'_1 \land n_2 = n'_2 \land n_3 < n'_3))
\]
Termination

Theorem

For any $E$, there exists an $n$ and $E'$ such that $E \rightarrow^n E'$ and
$\neg \exists E''. E' \rightarrow E''$. 