Now we can mimic $\equiv$ in a purely syntactical way. One only has to take the equations in $E$ and add stability, monotonicity, reflexivity, symmetry, and transitivity.

**Def 3.1.11 (Rewrite Relation, Proof Relation, Derivation)**

For a set of equations $E$, the corresponding rewrite relation $\rightarrow_{E} \subseteq \Sigma(\Sigma, \theta) \times \Sigma(\Sigma, \theta)$ is defined as follows:

$$S \rightarrow_{E} t \iff S_{\pi} = t_{\pi} \quad \text{and} \quad t = S[t_{2} \mid \pi]$$

for some $t_{1} = t_{2} \in E$, some $\tau \in \text{SUB}(\Sigma, \theta)$, some $\pi \in \text{Occ}(S)$.

The relation $\leftrightarrow_{E}^{*}$ is the proof relation of $E$.

We say that an equation $S \equiv t$ can be derived from $E$ ("$E \vdash S \equiv t$") iff $S \leftrightarrow_{E}^{*} t$.

**Goal:** Find out whether $S \equiv t$, i.e., whether $E \vdash S \equiv t$.

This is a semantical question, since one has to check whether all models $A$ of $E$ also satisfy $S \equiv t$.

$\Rightarrow$ difficult to automate

**Solution:** Check instead whether $S \leftrightarrow_{E}^{*} t$, i.e., whether $E \vdash S \equiv t$.

This is a syntactical question:

One has to find a derivation:

$$S = S_{0} \vdash_{E} S_{1} \vdash_{E} S_{2} \vdash_{E} \ldots \vdash_{E} S_{n} = t$$

This is enough, because

$$S \equiv_{E} t \iff S \leftrightarrow_{E}^{*} t$$
(this will be proved in Birkhoff's Theorem, Thm 3.1.14).

Ex. 3.1.12 \( \varepsilon \) = plus-equations

Goal: Find out whether

\[ \text{plus}(\text{succ}(\text{succ}(0)), x) \leftrightarrow \varepsilon \text{ plus}(\text{succ}(0), \text{succ}(x)) \]

This can be done automatically (one has to check whether left- or right-hand sides of equations match subterms).

Requires search, because one does not know which equation of \( \varepsilon \) should be applied in which direction.

To show that \( \varepsilon' \) and \( \varepsilon \rightarrow^* \) are the same, we first prove that the 5 properties (stability, monotonicity, reflexivity, symmetry, transitivity) of \( \varepsilon' \) also hold for \( \varepsilon \rightarrow^* \).

Lemma 3.1.13 (\( \varepsilon \rightarrow^* \) is a stable, monotonic equivalence relation)

For any set of equations \( \varepsilon \), \( \rightarrow^* \) and \( \varepsilon \rightarrow^* \) are stable and monotonic. Moreover, \( \varepsilon \rightarrow^* \) is an equivalence relation.

Proof:

\( \rightarrow^* \) is stable, i.e.: show that \( S^1 \rightarrow^* \varepsilon^1 \) implies

\[ S^0 \rightarrow^* \varepsilon^0 \quad \text{for all substitutions } \theta. \]

\[ S \rightarrow^* t \]

\[ S^|\Pi = t^|\Pi, \quad t = S^|\Pi \varepsilon^|\Pi \text{ for some } \Pi \in \text{Occ}(S), \text{satisfies } \theta, \]

\[ t^0 = t^2 \in \varepsilon \]

Goal: Show that \( S^|\Pi \rightarrow^* \varepsilon^|\Pi. \)

\[ S^|\Pi = S^0 \theta \]

\[ = t_0^0 \theta \]

\[ = t^0 \theta \]
In a similar way, one can show that $\rightarrow_{\varepsilon}$ is closed under contexts (monotonic).

To show that $\rightarrow_{\varepsilon}^{\#$} is stable and monotonic, one can use induction on the length of the derivation. \(\blacksquare\)

Now we can show the main theorem that states that the semantical relation $\equiv_{\varepsilon}$ and the syntactical relation $\rightarrow_{\varepsilon}^{\#}$ are the same.

**Theorem 3.1.15 (Birkhoff's Theorem)**

Let $E$ be a set of equations. Then the relations $\equiv_{\varepsilon}$ and $\rightarrow_{\varepsilon}^{\#}$ are the same. In other words: $E \models s \equiv t$ iff $E + s = t$.

**Proof**

"Soundness" show that $s \rightarrow_{\varepsilon}^{\#} t$ implies $s \equiv_{\varepsilon} t$.

First, we show that $s \rightarrow_{\varepsilon}^{\#} t$ implies $s \equiv_{\varepsilon} t$. 
\[
S \leftrightarrow \varepsilon t
\]

Let \( t_n \equiv t_2 \) or \( t_2 \equiv t_n \) in \( \varepsilon \) such that
\[
S \cong_{\varepsilon} t_n \circ \varepsilon, \quad t = S \varepsilon \varepsilon t_2 \varepsilon t_n \quad \text{for some } \varepsilon, t_n
\]

We have
\[
t_1 \equiv \varepsilon t_2 \quad \text{by symmetry of } \equiv_{\varepsilon} \quad \text{(Lemma 3.1.14)}
\]
\[
t_2 \varepsilon t_n \equiv \varepsilon t_2 \varepsilon t_n \quad \text{by stability of } \equiv_{\varepsilon} \quad \text{(Lemma 3.1.4(d))}
\]
\[
\varepsilon t_2 \varepsilon t_n \equiv \varepsilon S \varepsilon t_2 \varepsilon t_n \quad \text{by monotonicity of } \equiv_{\varepsilon}
\]
\[
\varepsilon S \varepsilon t_2 \varepsilon t_n \equiv \varepsilon t
\]

Now we show that \( S \varepsilon \rightarrow_{\varepsilon} t \) implies \( S \equiv_{\varepsilon} t \):

Let \( S \varepsilon \rightarrow_{\varepsilon} t \) for \( n \in \mathbb{N} \). We use induction on \( n \).

**Ind. Base:** \( n = 0 \)

\[
S \varepsilon \rightarrow_{\varepsilon} t
\]
\[
S = t
\]
\[
S \equiv_{\varepsilon} t \quad \text{by reflexivity of } \equiv_{\varepsilon} \quad \text{(Lemma 3.1.10)}
\]

**Ind. Step:** \( n > 0 \)

\[
S = \varepsilon S_0 \varepsilon S_1 \varepsilon \ldots \varepsilon S_{n-1} \varepsilon S_n = t
\]

By the induction hypothesis:

\[
S \varepsilon S_{n-1} \quad \text{implies } \quad S \equiv_{\varepsilon} S_{n-1}
\]

By the claim above

\[
S_{n-1} \varepsilon t \quad \text{implies } \quad S_{n-1} \equiv_{\varepsilon} t
\]

By transitivity of \( \equiv_{\varepsilon} \)

\[
S \equiv_{\varepsilon} t
\]

(Lemma 3.1.10)

**Completeness:** show that \( S \equiv_{\varepsilon} t \) implies \( S \varepsilon \rightarrow_{\varepsilon} t \)

\( S \equiv_{\varepsilon} t \) means that \( A \equiv \varepsilon \) implies \( A \equiv s \equiv t \)

Idea of the proof:
• Construct a specific $A = (A, \alpha)$ and a specific variable assignment $\beta$ such that for $I = (A, \alpha, \beta)$ we have

$$I \models S \subseteq t \iff S \subseteq^\beta t.$$ 

(3) Then show that $A \models E$.

• This proves the desired claim:

$$S \subseteq t \land A \not\models S \subseteq t \land I \models S \subseteq t \land I \not\models S \subseteq t \land S \subseteq^\beta t.$$ 

(4) Define $I = (A, \alpha, \beta)$ such that

$$I \models S \subseteq t \iff S \subseteq^\beta t.$$ 

First idea: interpret every term as itself,

i.e., choose an interpretation where

$I(t) = t$ for any term $t$.

Then: $A = \Sigma (\Sigma, \theta)$

$\alpha_f = f$ for any $f \in \Sigma$

$\beta(x) = x$ for any $x \in V$

Now indeed $I(t) = t$ for all terms.

But this is not yet what we want.

If $S \subseteq^\beta t$, then we want $I(S) = I(t)$.

Solution: $I$ should not map every term to itself,

but it should map every term to its equivalence class wrt. $\subseteq^\beta$, i.e., $I(t) = [t]_\subseteq^\beta$.
For any equivalence relation, the equivalence class of an object is the set of all objects that are equivalent to \( t \):

\[
\begin{align*}
\llbracket t \rrbracket_\equiv &= \{ s \mid s \xrightarrow{\equiv} t \} \\
\llbracket 0 \rrbracket_\equiv &= \{ 0, \text{plus}(0, 0), \text{plus}(\text{plus}(0, 0), 0), \ldots \} 
\end{align*}
\]

The quotient set \( \mathcal{Q}(\Sigma, \nu) /_\equiv \) is the set of all equivalence classes, i.e.,

\[
\mathcal{Q}(\Sigma, \nu) /_\equiv = \{ \llbracket \pi \rrbracket_\equiv \mid \pi \in \mathcal{Q}(\Sigma, \nu) \}.
\]

E.g.: \( \mathcal{Q}(\Sigma, \nu) /_\equiv = \{ \llbracket 0 \rrbracket_\equiv, \llbracket \text{succ}(0) \rrbracket_\equiv, \ldots, \llbracket x \rrbracket_\equiv, \llbracket y \rrbracket_\equiv, \llbracket \text{plus}(x, y) \rrbracket_\equiv, \ldots \} \)

Now we want to define an interpretation \( I \) which interprets every term as its equivalence class \( \llbracket t \rrbracket_\equiv \). Then:

\[ I(\psi) = I(\xi) \]

\[ I(\llbracket s \rrbracket_\equiv) = \llbracket t \rrbracket_\equiv \]

\[ \llbracket s \rrbracket_\equiv \xrightarrow{\equiv} \llbracket t \rrbracket_\equiv \]

\[ \llbracket s \rrbracket_\equiv \xrightarrow{\equiv} \llbracket t \rrbracket_\equiv \]

\[ I(\phi_{1} \land \ldots \land \phi_{n}) = \phi_{1}(I(t_{1}), \ldots, I(t_{n})) \]

\[ \llbracket \phi_{1} \land \ldots \land \phi_{n} \rrbracket_\equiv \]

\[ \llbracket \phi_{1}(t_{1}, \ldots, t_{n}) \rrbracket_\equiv \]

\[ \llbracket \phi_{1}(t_{1}, \ldots, t_{n}) \rrbracket_\equiv \]

\[ \llbracket \phi_{1}(t_{1}, \ldots, t_{n}) \rrbracket_\equiv \]
Thus: \( \chi(t_n) \rightarrow^* \chi(t_n) \) \( \rightarrow^* \epsilon \).

\( \chi(t_1) \rightarrow^* \chi(t_1) \) \( \rightarrow^* \epsilon \).

\( \chi(t_n) \rightarrow^* \chi(t_n) \) \( \rightarrow^* \epsilon \).

Thus: \( \chi(t_1) \rightarrow^* \chi(t_1) \) \( \rightarrow^* \epsilon \) for all \( t \in T \).

Now indeed: \( \chi(t_1) = \chi(t_1) \rightarrow^* \epsilon \) (can be shown by structural induction).

Next we solve \( (3) \):

We have to show that \( (x, \alpha) \rightarrow^* \epsilon \).

Let \( A \equiv v \in \mathcal{E} \).

We have to show \( A \equiv v \).

Let \( s \) be a variable assignment, let \( J = (x, \alpha, s) \): we have to show \( J \equiv v \), \( J(\alpha) = J(v) \).

For any variable \( x \), let \( s_x \) be a term from the equivalence class \( s(x) \). In other words: \( s(x) = \chi(s_x) \rightarrow^* \epsilon \).

Let \( \sigma \) be the substitution with \( \sigma x = s_x \) for all variables occurring in \( \mathcal{E} \).

Then: \( J(x) = \sigma(x) = \chi(s_x) \rightarrow^* \epsilon \).

By structural induction, one can show that for all terms \( t \), we have

\( J(t) = \chi(t \circ \sigma) \rightarrow^* \epsilon \).

Now one can show that for any \( v \in \mathcal{E} \), we have

\( J(v) = J(v) : \)
\[ M \equiv V \in E \]
\[ \land \quad M \leftrightarrow^* E \]
\[ \land \quad M \rightarrow^* V \Rightarrow (\text{since } \rightarrow^* E \text{ is stable, Lemma 3.1.12})\]
\[ \lor \quad [M]_{\rightarrow^* E} = [V]_{\rightarrow^* E} \]
\[ \land \quad j(M) = j(V) \]

This implies \( A \equiv M \equiv V \) and finishes the proof of \( \Box \).

By Birkhoff's Theorem, we can now try to solve the word problem automatically.

**Ex. 3.1.15**

\( E \) = set of the 3 group axioms

**Goal:** Solve the word problem \( i(i(n)) \equiv E \).

By Birkhoff's Thm., this is equivalent to \( i(i(n)) \rightarrow^* E \).

In general, to prove \( S \rightarrow^* E \) automatically, one can construct a search tree:

```
  S
 /   \
S_1  S_2  S_3  S_4  S_5...
```

where \( S_1, S_2, \ldots \) are all terms that are reachable from \( S \) in one step with \( \rightarrow^* E \).

Then we check if \( t \) occurs in this tree.

- If yes, then \( S \rightarrow^* E \) and therefore \( S \equiv E \).
- If no, then \( S \neq E \) and therefore \( S \neq E \).

**Problems:**

- Paths could be infinitely long (depth could be infinite).
- Node could have infinitely many children
  (This happens if $E$ contains equations $u = v$ with $U(u) \neq U(v)$)
  (breadth could be infinite)

The word problem for equations is not decidable, but semi-decidable:

Given a set of equations $E$ and $s \equiv t$:
- If $s \equiv t$: procedure terminates with "yes"
- If $s \not\equiv t$: procedure might not terminate (or say "no")

Semi-Decision Procedure:
- build up the search tree starting with $s$
- stop as soon as one reaches $t$
- if $U(t_1) = U(t_2)$ for all $t_1 \equiv t_2 \in E$, then one can build up the tree with breadth-first search
- otherwise, one has to construct the tree "by diagonalization" to ensure that eventually one reaches every node of the tree:

```
  1
   / \  \
  2   3   \\
   /   /  \
  3   4   \\
      /   \\
       4 \\
```

2 main drawbacks
- search not goal directed: if $s \equiv t$ holds, it will take very long to find the proof
- not useful to disprove equations: if $s \not\equiv t$ doesn't hold,
then the procedure usually doesn't terminate. Therefore: I identify classes of $\equiv_\varepsilon$ where $\equiv_\varepsilon$ is decidable.

- Develop more efficient procedures for these cases.