Special Case: sets of equations $E$ without variables

Here, it will turn out that $E$ is decidable.

Applications:
- compiler construction: optimization of code
  e.g. Common Subexpression problem (identify subexpressions that are "equal" w.r.t. their occurrence in the program). Then these expressions only have to be evaluated once
- program verification
  corresponds to the question of validity of universally quantified formulas of p.d. logic

etc.

**Def 3.2.1 (Ground Identity)**
An equation $s \equiv t$ is a ground identity iff $\overline{G(s)} = \overline{G(t)} = \emptyset$.

**Ex 3.2.2** Imperative program with variables

$i, j, k, l, m$ variables for numbers
$f, g$ variables for arrays

\[
\begin{align*}
  i & := k; \\
  k & := l; \\
  f[i] & := g[k];
\end{align*}
\]

if $(j == f[j])$

\[
\begin{align*}
  f & \left[ l \right] \\
  i & := j; \\
  k & := l; \\
  m & := 8
\end{align*}
\]
\[ m = g \text{ [I]} \]

\[ \left\{ \begin{array}{c}
\text{Does } f \text{ [m]} = g \text{ [I]} \text{ hold here?} \\
(\text{Yes})
\end{array} \right. \]

So we want to find out whether \( f(m) \equiv E g(k) \) holds for
\( E = \{ x = e, k = l, f(x) \equiv g(k), d = f(x), m = g(k) \} \).

Here, \( x, y, \ldots \) are not variables from \( S \), but constants from \( S_0 \).

By Birkhoff’s Theorem, instead of \( f(m) \equiv E g(k) \), we can check \( f(m) \leftarrow E g(k) \).

**Goal:** Develop an efficient decision procedure that checks \( S \equiv E \) if \( E \) doesn’t contain variables.

---

**Idea:** Given \( E \), compute all equations \( m \equiv v \)
that are entailed by \( E \) (i.e., all equations with \( m \equiv v \)). Then check whether \( S \equiv t \) is among these equations.

To compute all equations entailed by \( E \), consider:
- reflexivity
- symmetry
- transitivity
- congruence/monotonicity

---

**Def 323 (Direct Consequences of \( E \))**

For a set of ground identities \( E \) over a signature \( \Sigma \) we define:

\[ R, \Sigma, \equiv \]
\[ R = \{ t = t \mid t \in \Sigma(\Sigma) \} \]
\[ S(\Sigma) = \{ t = s \mid s = t \in \Sigma \} \]
\[ T(\Sigma) = \{ s = r \mid \text{there exists a } t \in \Sigma(\Sigma) \text{ such that } s = t, t = r \in \Sigma \} \]
\[ C(\Sigma) = \{ f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n) \mid f \in \Sigma, s_i = t_i \in \Sigma \text{ for all } 1 \leq i \leq n \} \]

**Ex 324**

Let \( \Sigma = \{ i = i, k = k, f(i) = g(k), j = f(i), m = g(k) \} \)
\[ R = \{ i = i, j = j, f(i) = f(i), f(g(k)) = f(g(k)), \ldots \} \]
\[ S(\Sigma) = \{ i = i, k = k, g(k) = f(i), f(j) = j, g(k) = m \} \]
\[ T(\Sigma) = \{ i = f(i) \} \]
\[ C(\Sigma) = \{ f(i) = f(j), g(k) = g(j), \ldots \} \text{ is 10 equations} \]

Clearly, \( \Sigma \cup R \cup S(\Sigma) \cup T(\Sigma) \cup C(\Sigma) \) only contains equations that are entailed by \( \Sigma \), but it does not yet contain all of these equations.

\( \Rightarrow \) One has to repeat the application of \( S, T, \) and \( C \) (potentially infinitely often).

\( \Rightarrow \) This results in the congruence closure of \( \Sigma \).
Def 325 (Congruence Closure)
For a set of ground identities $E$ over a signature $\Sigma$, we define:

\[ E_0 = E \cup R \]
\[ E_{i+1} = E_i \cup S(E_i) \cup T(E_i) \cup C(E_i) \]
for all $i \in \mathbb{N}$.

The congruence closure of $E$ is the "limit" of the sequence $E_0, E_1, E_2, \ldots$, i.e.,
\[ CC(E) = \bigcup_{i \in \mathbb{N}} E_i \]

To check whether $s \equiv_{E} t$ holds, we now start computing the congruence closure $CC(E)$. If $s \equiv_{E} t \in CC(E)$, we return "yes" and otherwise, we return "no".

Questions
1. Is this correct (i.e., does $s \equiv_{E} t$ iff $s \equiv t \in CC(E)$ hold?) Yes, see Thm 327

2. In general, the iteration $E_0, E_1, E_2, \ldots$ does not terminate (i.e., $E_0 \not\subseteq E_1 \not\subseteq E_2 \not\subseteq \ldots$). Therefore, this procedure doesn't terminate if $s \equiv t \not\in CC(E)$. 

TES 2015-16 Seite 4
Ex 326: \( f(m) \equiv g(n) \in \mathcal{E}_g \subseteq \mathcal{C}(E) \)

Thus 327 (Congruence Closure is Sound + Complete)
Let \( E \) be a set of ground identities over \( E \), let \( s, t \in \mathcal{E}(E) \). Then: \( s \equiv_E t \) iff \( s \equiv t \in \mathcal{C}(E) \).

Proof

\( \subseteq \) (Soundness)
We have to show that \( s \equiv t \in \mathcal{E}_i \) implies \( s \equiv \mathcal{E}_i t \) for all \( i \in \mathbb{N} \). We use induction on \( i \).

Ind. Base: \( i = 0 \)
Since \( \mathcal{E}_0 = E \cup R \), this follows from reflexivity of \( \equiv_E \) (Lemma 3.1.10).

Ind. Step: \( s \equiv t \in \mathcal{E}_{i+1} = \mathcal{E}_i \cup S(\mathcal{E}_i) \cup T(\mathcal{E}_i) \cup C(\mathcal{E}_i) \)
By the induction hypothesis, we have \( s \equiv \mathcal{E}_i t \) for all \( M \mathcal{E}_i N \in \mathcal{E}_i \). Then we also have \( s \equiv \mathcal{E}_i t \), since \( \equiv_E \) is a congruence relation (Lemma 3.1.10).

\( \supseteq \) (Completeness): We postpone this proof, because we will show a stronger statement in Thm 3.2.17.
Problem of Question 2 remains:

In general, $E_i \subset E_{i+1}$ and thus, the iteration to compute $CC(E)$ does not stop.

$i \equiv i \in E_0$

$f(i) \equiv f(i) \in E_1 \setminus E_0$

$f(f(i)) \equiv f(f(i)) \in E_2 \setminus E_1$

\vdots

$f^n(i) \equiv f^n(i) \in E_n \setminus E_{n-1}$