Exercise 24 (Equivalence Classes)

Let $\Sigma = \{f, g\}$ and
$$E = \{f(f(x)) \approx f(x), \ g(f(x)) \approx f(x), \ f(g(x)) \approx g(x), \ g(g(x)) \approx g(x)\}$$

a) Describe the equivalence classes of $\approx_E$.

b) For each equivalence class $[t]_{\approx_E}$, determine a shortest term $\hat{t}$ in $[t]_{\approx_E}$.

c) Give a model for $E$ that has more than one element.

Exercise 25 (Congruence Closure)

Let $\Sigma = \{f, a, b\}$ and $G = \{f(a, b) \approx a\}$. Using congruence closure, decide whether the terms $s$ and $t$ are equivalent with respect to the equation set $G$. Use the abstract algorithm on equation sets.

a) $s = f(f(a, b), b), \ t = a$

b) $s = f(f(a, b), b), \ t = b$

Exercise 26 (Unification)

Let $u, x, y, z$ be variables. Use the unification algorithm recalled in the lecture to solve the following two unification problems:

a) $S_1 = \{f(h(x), g(x, u)) \approx f(z, g(f(y, y), z))\}$

b) $S_2 = \{h(x, g(x, y), y) \approx h(x, g(a, y), y), \ z \approx h(x, g(x, b), b)\}$

Homework 27 (Programming Assignment)

You are assigned to implement congruence closure in Haskell. Additional instructions can be found on the lecture web page.

Homework 28 (More General Substitutions)

Let $\sigma$ and $\sigma'$ be substitutions such that each one is an instance of the other: Formally, we have both $\sigma \preceq \sigma'$ and $\sigma \succeq \sigma'$. Show that in this case, there must exist a renaming $\rho$ (i.e., an injective substitution where $\text{Ran}(\rho) \subseteq V$) such that $\sigma = \rho\sigma'$. 