# Term Rewriting Systems

Florian Frohn

13. November 2015

# Exercise 1

The program

```
a = c;
d = f[f[c]];
f[b] = f[c];
if ( b == f[a] ) {
   (*)
}
```

has been translated to

$$\begin{aligned} a &\equiv c \\ d &\equiv f(f(c)) \\ f(b) &\equiv f(c) \\ b &\equiv f(a) \end{aligned}$$

Show via $\leftrightarrow^*_{\mathcal{E}}$ that $d \equiv_{\mathcal{E}} f(c)$ holds.

# Exercise 1

$$
\begin{array}{rcl}
a & \equiv & c \\
d & \equiv & f(f(c)) \\
f(b) & \equiv & f(c) \\
b & \equiv & f(a)
\end{array}
$$

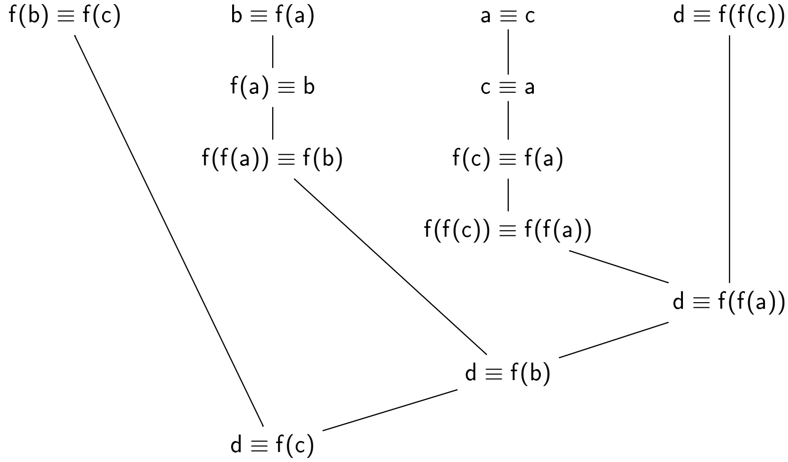
Show via $\leftrightarrow_{\mathcal{E}}^*$ that $d \equiv_{\mathcal{E}} f(c)$ holds.

$$
\begin{array}{ll}
 & \underline{d} \\
\to_{\mathcal{E}} & f(f(\underline{c})) \\
\leftarrow_{\mathcal{E}} & f(\underline{f(a)}) \\
\leftarrow_{\mathcal{E}} & \underline{f(b)} \\
\to_{\mathcal{E}} & f(c)
\end{array}
$$

# Exercise 1

$$a \equiv c, d \equiv f(f(c)), f(b) \equiv f(c), b \equiv f(a)$$

Show via congruence closure that $d \equiv_{\mathcal{E}} f(c)$ holds.



# Exercise 1

```
1. a = c;
2. d = f[f[c]];
3. f[b] = f[c];
4. if ( b == f[a] ) {
5.   (*)
6. }
```

Give initial values for the variables `a`, `b`, `c`, `d` and for the array `f`, such that at the position `(*)` the value of `d` is not equal to that of `f[c]`.
initial valuation: `f=[1,2,3,4], a=0, b=2, c=1, d=0`

1. `f=[1,2,3,4], a=1, b=2, c=1, d=0`
   - `f[c] = 2, f[f[c]] = 3`
2. `f=[1,2,3,4], a=1, b=2, c=1, d=3`
   - `f[c] = 2`
3. `f=[1,2,2,4], a=1, b=2, c=1, d=3`
   - `f[a] = 2 = b`

4./5. `f=[1,2,2,4], a=1, b=2, c=1, d=3`

# Exercise 2

$$\{\mathsf{a} \equiv \mathsf{b}, \mathsf{c} \equiv \mathsf{f}(\mathsf{c}), \mathsf{f}(\mathsf{b}) \equiv \mathsf{g}(\mathsf{b}), \mathsf{d} \equiv \mathsf{c}, \mathsf{f}(\mathsf{g}(\mathsf{b})) \equiv \mathsf{f}(\mathsf{d})\}$$

Check $\mathsf{g}(\mathsf{c}) \equiv_{\mathcal{E}} \mathsf{f}(\mathsf{f}(\mathsf{a}))$ using the Algorithm CONGRUENCE_CLOSURE.

1. Let $S = Subterms(\mathcal{E}) \cup Subterms(s) \cup Subterms(t)$ and $L = \{\{s, t\} | s \equiv t \in \mathcal{E}\} \cup \{\{s\} | s \in S\}$
2. Join all sets $M_1, M_2 \in L$ where $M_1 \cap M_2 \neq \emptyset$
3. Let $K = L \cup \{\{\mathsf{f}(s_1, \ldots, s_n), \mathsf{f}(t_1, \ldots, t_n)\} | f \in \Sigma, \exists M_i \in L : s_i, t_i \in M_i, \mathsf{f}(s_1, \ldots, s_n), \mathsf{f}(t_1, \ldots, t_n) \in S\}$
4. Join all sets $M_1, M_2 \in K$ where $M_1 \cap M_2 \neq \emptyset$
5. If $K \neq L$ set $L = K$ and go to 3.
6. If $\exists M \in L : s \in M \wedge t \in M$, return True, otherwise return False

# Exercise 2

$$\{a \equiv b, c \equiv f(c), f(b) \equiv g(b), d \equiv c, f(g(b)) \equiv f(d)\}$$
$$g(c) \equiv_{\mathcal{E}} f(f(a))$$

1. Let $S = \textit{Subterms}(\mathcal{E}) \cup \textit{Subterms}(s) \cup \textit{Subterms}(t)$ and $L = \{\{s,t\} | s \equiv t \in \mathcal{E}\} \cup \{\{s\} | s \in S\}$
2. Join all sets $M_1, M_2 \in L$ where $M_1 \cap M_2 \neq \emptyset$
3. Let $K = L \cup \{\{f(s_1, \ldots, s_n), f(t_1, \ldots, t_n)\} | f \in \Sigma, \exists M_i \in L : s_i, t_i \in M_i, f(s_1, \ldots, s_n), f(t_1, \ldots, t_n) \in S\}$
4. Join all sets $M_1, M_2 \in K$ where $M_1 \cap M_2 \neq \emptyset$
5. If $K \neq L$ set $L = K$ and go to 3.
6. If $\exists M \in L : s \in M \wedge t \in M$, return True, otherwise return False

1. $S = \{a, b, c, d, f(a), f(b), f(c), f(d), f(f(a)), f(g(b)), g(b), g(c)\}$,
   $L = \{\{a, b\}, \{c, f(c)\}, \{f(b), g(b)\}, \{d, c\}, \{f(g(b)), f(d)\}\} \cup \ldots$
2. $L = \{\{a, b\}, \underline{\{c, d, f(c)\}}, \{f(b), g(b)\}, \{f(g(b)), f(d)\}\} \cup \ldots$
3. $K = L \cup \{\underline{\{f(a), f(b)\}}, \underline{\{f(c), f(d)\}}\} \cup \ldots$
4. $L = \{\{a, b\}, \underline{\{c, d, f(c), f(d), f(g(b))\}}, \underline{\{f(a), f(b), g(b)\}}, \} \cup \ldots$
3. $K = L \cup \{\{f(a), f(b)\}, \{f(c), f(d)\}, \underline{\{f(f(a)), f(g(b))\}}\} \cup \ldots$
4. $L = \{\{a, b\}, \underline{\{c, d, f(c), f(d), f(g(b)), f(f(a))\}}, \{f(a), f(b), g(b)\}\} \cup \ldots$
3. $K = L \cup \{\{f(a), f(b)\}, \{f(c), f(d)\}, \{f(f(a)), f(g(b))\}\} \cup \ldots$
4. $K = L$
5. return False

## Exercise 3

The set $\mathcal{F}$ of all formulas in propositional logic with equalities and uninterpreted functions is the smallest set such that:

- if $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$, then $s \equiv t \in \mathcal{F}$
- if $\varphi, \psi \in \mathcal{F}$, then $\varphi \wedge \psi \in \mathcal{F}$ and $\varphi \vee \psi \in \mathcal{F}$
- if $\varphi \in \mathcal{F}$, then $\neg\varphi \in \mathcal{F}$

Given $\varphi \in \mathcal{F}$, develop a procedure that decides if there is an interpretation $I = (\mathcal{A}, \alpha, \beta)$ with $I \models \varphi$.

1. Let $\varphi' = \varphi\sigma$ with $\sigma = \{x/c_x | x \in \mathcal{V}(\varphi)\}$
   - If $I \models \varphi$, then $(\mathcal{A}, \alpha') \models \varphi'$ where $\alpha'(c_x) = \beta(x)$
2. Let $\varphi_1 \vee \cdots \vee \varphi_n = \text{DNF}(\varphi')$
   - If $(\mathcal{A}, \alpha') \models \varphi'$, then $(\mathcal{A}, \alpha') \models \varphi_1 \vee \cdots \vee \varphi_n$
3. Check each $\varphi_i$ separately
   - $(\mathcal{A}, \alpha') \models \varphi = \varphi_1 \vee \cdots \vee \varphi_n$ implies $\exists i : (\mathcal{A}, \alpha') \models \varphi_i$
4. Let $\varphi_i = \varphi_i^+ \wedge \varphi_i^-$
5. Check each $s \not\equiv t \in \varphi_i^-$ separately
   - $\varphi_i = \varphi_i^+ \wedge \varphi_i^-$ unsat implies $\exists s \not\equiv t \in \varphi_i^- : \varphi_i^+ \implies s \equiv t$

# Exercise 3

The set $\mathcal{F}$ of all formulas in propositional logic with equalities and uninterpreted functions is the smallest set such that:

- if $s, t \in \mathcal{T}(\Sigma, \mathcal{V})$, then $s \equiv t \in \mathcal{F}$
- if $\varphi, \psi \in \mathcal{F}$, then $\varphi \wedge \psi \in \mathcal{F}$ and $\varphi \vee \psi \in \mathcal{F}$
- if $\varphi \in \mathcal{F}$, then $\neg\varphi \in \mathcal{F}$

Given $\varphi \in \mathcal{F}$, develop a procedure that decides if there is an interpretation $I = (\mathcal{A}, \alpha, \beta)$ with $I \models \varphi$.

1. Let $\varphi' = \varphi\sigma$ with $\sigma = \{x/c_x | x \in \mathcal{V}(\varphi)\}$
2. Let $\varphi_1 \vee \cdots \vee \varphi_n = \mathrm{DNF}(\varphi')$
3. For each $i \in \{1, \ldots, n\}$
   - Let $\varphi_i = s_1 \equiv t_1 \wedge \cdots \wedge s_m \equiv t_m \wedge \varphi_i^-$
   - Let $\mathcal{E} = \{s_1 \equiv t_1, \ldots, s_m \equiv t_m\}$
   - Check $\exists s \not\equiv t \in \varphi_i^- : s \equiv_{\mathcal{E}} t$ using congruence closure
   - If there is no such $s \not\equiv t$, return sat
4. return unsat

# Exercise 3

1. Let $\varphi' = \varphi\sigma$ with $\sigma = \{x/c_x | x \in \mathcal{V}(\varphi)\}$
2. Let $\varphi_1 \vee \cdots \vee \varphi_n = \mathrm{DNF}(\varphi')$
3. For each $i \in \{1, \ldots, n\}$
   - Let $\varphi_i = s_1 \equiv t_1 \wedge \cdots \wedge s_m \equiv t_m \wedge \varphi_i^-$
   - Let $\mathcal{E} = \{s_1 \equiv t_1, \ldots, s_m \equiv t_m\}$
   - Check $\exists s \not\equiv t \in \varphi_i^- : \mathcal{E} \models s \equiv t$ using congruence closure
   - If there is no such $s \not\equiv t$, return sat
4. return unsat

Run your algorithm on the formula

$$\neg(\neg(x_1 \equiv x_2) \vee \neg(x_2 \equiv x_3)) \wedge x_4 \equiv x_5 \wedge \neg(\mathrm{f}(x_1) \equiv \mathrm{f}(x_2)) \wedge \neg(x_5 \equiv x_1)$$

- 1. yields $\neg(\neg(c_{x_1} \equiv c_{x_2}) \vee \neg(c_{x_2} \equiv c_{x_3})) \wedge c_{x_4} \equiv c_{x_5} \wedge \neg(\mathrm{f}(c_{x_1}) \equiv \mathrm{f}(c_{x_2})) \wedge \neg(c_{x_5} \equiv c_{x_1})$
- 2. yields $c_{x_1} \equiv c_{x_2} \wedge c_{x_2} \equiv c_{x_3} \wedge c_{x_4} \equiv c_{x_5} \wedge \mathrm{f}(c_{x_1}) \not\equiv \mathrm{f}(c_{x_2}) \wedge c_{x_5} \not\equiv c_{x_1}$
  - $\varphi_1^- = \mathrm{f}(c_{x_1}) \not\equiv \mathrm{f}(c_{x_2})) \wedge c_{x_5} \not\equiv c_{x_1}$
  - $\mathcal{E} = \{c_{x_1} \equiv c_{x_2}, c_{x_2} \equiv c_{x_3}, c_{x_4} \equiv c_{x_5}\}$
  - $\mathrm{f}(c_{x_1}) \equiv_{\mathcal{E}} \mathrm{f}(c_{x_2})$ by congruence closure
- return unsat

# Exercise 3

Give a formula $\varphi \in \mathcal{F}$ and a *finite* carrier $\mathcal{A}$ such that $\varphi$ is satisfiable for all infinite carriers, but there is no interpretation $I = (\mathcal{A}, \alpha, \beta)$ such that $I \models \varphi$.

- Let $\mathcal{A} = \{\mathcal{O}\}$
- Let $\varphi = \neg(x \equiv y)$
- Infinite carrier: $\varphi$ is satisfied if $\beta(x) \neq \beta(y)$
- But: unsatisfiable with carrier $A$