# Automated Reasoning I, 2015
# Final Exam, Sample Solution

## Assignment 1

We first define for every $\Sigma$-algebra $\mathcal{A}$ an algebra $\mathcal{A}'$ by $U_{\mathcal{A}'} = U_{\mathcal{A}}$, $f_{\mathcal{A}'} = f_{\mathcal{A}}$ for every $f/n \in \Omega$, and $P_{\mathcal{A}'} = U_{\mathcal{A}}^m \setminus P_{\mathcal{A}}$ for every $P/m \in \Pi$.

In the next step we prove the lemma that $\mathcal{A}(\beta)(\mathrm{neg}(F)) = \mathcal{A}'(\beta)(F)$ for every formula $F$ and every assignment $\beta$.

We use induction over the structure of formulas. Clearly $\mathcal{A}(\beta)(\mathrm{neg}(\bot)) = \mathcal{A}(\beta)(\bot) = 0 = \mathcal{A}'(\beta)(\bot)$ and $\mathcal{A}(\beta)(\mathrm{neg}(\top)) = \mathcal{A}(\beta)(\top) = 1 = \mathcal{A}'(\beta)(\top)$.

Since all function symbols are interpreted in the same way in the algebras $\mathcal{A}$ and $\mathcal{A}'$, we get $\mathcal{A}(\beta)(t) = \mathcal{A}'(\beta)(t)$ for every term $t$; therefore $\mathcal{A}(\beta)(\mathrm{neg}(P(t_1, \ldots, t_m))) = \mathcal{A}(\beta)(\neg P(t_1, \ldots, t_m)) = 1$ iff $(\mathcal{A}(\beta)(t_1), \ldots, \mathcal{A}(\beta)(t_m)) \notin P_{\mathcal{A}}$ iff $(\mathcal{A}'(\beta)(t_1), \ldots, \mathcal{A}'(\beta)(t_m)) \in P_{\mathcal{A}'}$ iff $\mathcal{A}'(\beta)(P(t_1, \ldots, t_m)) = 1$.

By structural induction we now obtain $\mathcal{A}(\beta)(\mathrm{neg}(F \wedge G)) = \mathcal{A}(\beta)(\mathrm{neg}(F) \wedge \mathrm{neg}(G)) = \min\{\mathcal{A}(\beta)(\mathrm{neg}(F)), \mathcal{A}(\beta)(\mathrm{neg}(G))\} = \min\{\mathcal{A}'(\beta)(F), \mathcal{A}'(\beta)(G)\} = \mathcal{A}'(\beta)(F \wedge G)$ and $\mathcal{A}(\beta)(\mathrm{neg}(\neg F)) = \mathcal{A}(\beta)(\neg(\mathrm{neg}(F))) = 1 - \mathcal{A}(\beta)(\mathrm{neg}(F)) = 1 - \mathcal{A}'(\beta)(F) = \mathcal{A}(\beta)(\neg F)$ and $\mathcal{A}(\beta)(\mathrm{neg}(\exists x\, F)) = \mathcal{A}(\beta)(\exists x\, (\mathrm{neg}(F))) = \max_{a \in U_{\mathcal{A}}}\{\mathcal{A}(\beta[x \mapsto a])(\mathrm{neg}(F))\} = \max_{a \in U_{\mathcal{A}'}}\{\mathcal{A}'(\beta[x \mapsto a])(F)\} = \mathcal{A}'(\beta)(\exists x\, F)$.

Using the lemma, we now see that if $F$ is valid, then for every $\mathcal{A}$ and $\beta$ we get $\mathcal{A}(\beta)(\mathrm{neg}(F)) = \mathcal{A}'(\beta)(F) = 1$, which implies that $\mathrm{neg}(F)$ is valid as well.

*Note:*

- The subformulas of valid (or unsatisfiable) formulas are in general neither valid nor unsatisfiable. E.g., $(\exists x\, P(x)) \vee (\forall y\, \neg P(y))$ is valid, and $(\exists x\, P(x)) \wedge (\forall y\, \neg P(y))$ is unsatisfiable, but $(\exists x\, P(x))$ and $(\forall y\, \neg P(y))$ are neither valid nor unsatisfiable. For this reason, using some lemma is unavoidable. Any attempt to prove properties of valid (or unsatisfiable, or satisfiable) formulas directly by induction must fail, since the induction hypothesis is not applicable to the subformulas.

## Assignment 2

**Part (a)** In (1), $P(b, x)$ and $R(f(x), x)$ are not maximal since $P(g(x), x) \succ P(b, x)$ and $P(g(x), x) \succ R(f(x), x)$. In (3), $\neg R(y, z)$ is not maximal since $\neg P(z, h(y)) \succ \neg R(y, z)$. In (4), $Q(z)$ is not maximal since $\neg P(z, b) \succ Q(z)$. In (5), $\neg R(f(x), x)$ is not maximal since $Q(x) \succ \neg R(f(x), x)$. The remaining literals are maximal in their clauses: (1)1; (2)1; (3)1; (4)1, (4)2, (4)4; (5)1, (5)2. This yields the following inferences:

Res. (1)1, (3)1: mgu: $\{x \mapsto h(y),\ z \mapsto g(h(y))\}$
$P(b, h(y)) \vee R(f(h(y)), h(y)) \vee \neg R(y, g(h(y)))$

Res. (1)1, (4)1: mgu: $\{x \mapsto c,\ y \mapsto g(c)\}$
$P(b, c) \vee R(f(c), c) \vee \neg P(z, b) \vee \neg Q(z) \vee R(z, g(c))$

Res. (1)1, (4)2: mgu: $\{x \mapsto b,\ z \mapsto g(b)\}$
$P(b, b) \vee R(f(b), b) \vee \neg P(y, c) \vee \neg Q(g(b)) \vee R(g(b), y)$

Fact. (5)1, (5)2: mgu: $\{x \mapsto b\}$
$Q(b) \vee \neg R(f(b), b)$

**Part (b)** Resolution inferences with $P(g(x), x)$ in (1) can only be prevented by selecting the negative literals $\neg R(y, z)$ in (3) and $\neg Q(z)$ in (4). By selecting $\neg Q(z)$ in (4), a resolution inference between (4) and (5) becomes possible; to avoid this resolution inference and the ordered factorization inference with (5), the negative literal $\neg R(f(x), x)$ must be selected in (5).

## Assignment 3

(1) **true:** $[c] = \{c, d\}$.
(2) **false:** $f(y, c) \leftrightarrow_E b \leftrightarrow_E f(c, c)$ implies $f(y, c) \in [f(c, c)]$.
(3) **true:** the universe of $\mathcal{T} = \mathrm{T}_\Sigma(X)/E$ is the set of all $E$-congruence classes of terms in $\mathrm{T}_\Sigma(X)$, so it includes $[x]$.
(4) **false:** an $E$-congruence class contains *all* terms in $\mathrm{T}_\Sigma(X)$ that are $E$-equal to each other, so the $E$-congruence class of $b$ and $f(x, c)$ contains, e.g., $f(c, c)$ and $f(f(y, y), c)$ as well.

(5) **true:** $f(c,b) \leftrightarrow_E f(d,b)$ implies $f(c,b) \in [f(d,b)]$.
(6) **true:** $f(y,d) \leftrightarrow_E f(y,c) \leftrightarrow_E b \leftrightarrow_E f(z,c)$ implies $f_{\mathcal{T}}([y],[d]) = [f(y,d)] = [f(z,c)]$.
(7) **true:** $\mathcal{T}(\beta)(y) = [c] = [d] = \mathcal{T}(\beta)(d)$, so $\mathcal{T}(\beta)(y \approx d) = 1$.
(8) **false:** for the modified assignment $\gamma = \beta[x \mapsto [b]]$, $\mathcal{T}(\gamma)(z) = [b] \neq [c] = \mathcal{T}(\gamma)(c)$.

## Assignment 4

Since every application of a rule in $R$ reduces the size of the term by 1, the rewrite system $R$ is obviously terminating. It has no critical pairs, so it is locally confluent and, by termination, confluent. By Birkhoff's Theorem, $R \models \forall \vec{x}\,(s \approx t)$ if and only if $s \leftrightarrow_R^* t$. As $R$ is confluent, $s \leftrightarrow_R^* t$ if and only if $s \to_R^* u \leftarrow_R^* t$ for some $u$. Since every $R$-rewrite step reduces the size of the term by 1, the derivation $s \to_R^* u$ can consist of at most $|s| - 1$ steps and the derivation $u \leftarrow_R^* t$ can consist of at most $|t| - 1$ steps; so we get a derivation $s \leftrightarrow_R^* t$ with at most $(|s| - 1) + (|t| - 1)$ rewrite steps.

## Assignment 5

The relation $\succ_{\mathrm{do}}$ is not stable under substitutions. For instance $s = f(x, g(g(y))) \succ_{\mathrm{do}} f(g(x), g(y)) = t$ since $\mathrm{depth}(s) = 3$ and $\mathrm{depth}(t) = 2$, but if $\sigma = \{x \mapsto h(h(z))\}$, then we get $s\sigma = f(h(h(z)), g(g(y))) \prec_{\mathrm{do}} f(g(h(h(z))), g(y)) = t\sigma$ since $\mathrm{depth}(s\sigma) = 3$ and $\mathrm{depth}(t\sigma) = 4$.

## Assignment 6

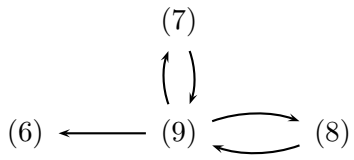
**Part (a)** The set of defined symbols is $D = \{f, g, h\}$, therefore $R$ has four dependency pairs:

$$f^\sharp(x, h(x)) \to h^\sharp(k(x)) \qquad (6)$$
$$f^\sharp(h(x), y) \to g^\sharp(x, g(h(x), x)) \qquad (7)$$
$$f^\sharp(h(x), y) \to g^\sharp(h(x), x) \qquad (8)$$
$$g^\sharp(x, x) \to f^\sharp(x, x) \qquad (9)$$

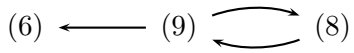
*Note:*

- There is no dependency pair $f^\sharp(h(x), y) \to h^\sharp(x)$ derived from (2), since $h(x)$ is a proper subterm of the left-hand side of (2).

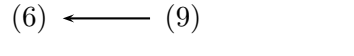
**Part (b)** The approximated dependency graph for $R$ is




The only SCC has the node set $\{(7), (8), (9)\}$. We use a simple projection $\pi$ with $\pi(f^\sharp) = 1$ and $\pi(g^\sharp) = 1$. For (7), $\pi(f^\sharp(h(x), y)) = h(x) \rhd x = \pi(g^\sharp(x, g(h(x), x)))$, for (8), $\pi(f^\sharp(h(x), y)) = h(x) = \pi(g^\sharp(h(x), x))$, and for (9), $\pi(g^\sharp(x, x)) = x = \pi(f^\sharp(x, x))$. So we can delete node (7) from the dependency graph and obtain




Now the only SCC has the node set $\{(8), (9)\}$. We use a simple projection $\pi$ with $\pi(f^\sharp) = 1$ and $\pi(g^\sharp) = 2$. For (8), $\pi(f^\sharp(h(x), y)) = h(x) \rhd x = \pi(g^\sharp(h(x), x))$ and for (9), $\pi(g^\sharp(x, x)) = x = \pi(f^\sharp(x, x))$. Therefore, we can now delete (8) from the dependency graph and obtain




Since there are no more SCCs left, the TRS $R$ is terminating.

*Note:*

- An SCC is a *maximal* subgraph in which there is a non-empty path from every node to every node. The subgraphs with the node sets $\{(7), (9)\}$ and $\{(8), (9)\}$ are *not* SCCs of the original dependency graph, since they are proper subgraphs of the subgraph with the node set $\{(7), (8), (9)\}$ in which there is a non-empty path from every node to every node.

**Part (c)** The exact dependency graph for $R$ contains an edge from a dependency pair $s \to t$ to a dependency pair $u \to v$ if $t\sigma \to_R^* u\tau$ for some instances $t\sigma$ and $u\tau$. For the dependency pairs (9) and (6), this condition is satisfied: Let $\sigma = \{x \mapsto h(b)\}$ and $\tau = \{x \mapsto b\}$, then $f^\sharp(x, x)\sigma = f^\sharp(h(b), h(b))$ rewrites to $f^\sharp(x, h(x))\tau = f^\sharp(b, h(b))$ using rule (5).