Automated Reasoning I, 2019/20
Endterm Exam, Sample Solution

Assignment 1

Part (a) The statement holds: If $\{b\} \succ_{\text{mul}} S_1$, then by definition $S_1 = (\{b\} - X) \cup Y$ for multisets X and Y such that $\emptyset \neq X \subseteq \{b\}$ and such that for each $y \in Y$ there is an $x \in X$ with $x \succ y$. Clearly X must equal $\{b\}$, and therefore $Y = S_1$. Thus we have $b \succ y$ for each $y \in S_1$. Analogously, we can show that $b \succ y$ for each $y \in S_2$. Therefore $b \succ y$ for each $y \in S_1 \cup S_2$, which implies $\{b\} \succ_{\text{mul}} S_1 \cup S_2$.

Part (b) The statement does not hold: Let $S = \{b_1, b_2\}$, then obviously $\{b_1, b_2\} \succ_{\text{mul}} \{b_1\}$ and $\{b_1, b_2\} \succ_{\text{mul}} \{b_2\}$, but not $\{b_1, b_2\} \succ_{\text{mul}} \{b_1, b_2\}$.

Assignment 2

(1) **true:** in particular, it has models with arbitrarily large universes.
(2) **true:** $\forall x\, P(x) \models \forall x\, P(f(x))$.
(3) **false:** the formula is unsatisfiable, so it has no models at all.
(4) **true:** take $U_\mathcal{A} = \{1, 2\}$, $b_\mathcal{A} = 1$, $c_\mathcal{A} = 1$, $f_\mathcal{A} : x \mapsto 2$, $P_\mathcal{A} = \{2\}$.
(5) **false:** take $U_\mathcal{A} = \{1, 2\}$, $b_\mathcal{A} = 1$, $c_\mathcal{A} = 1$, $f_\mathcal{A} : x \mapsto 1$, $P_\mathcal{A} = \{1\}$.
(6) **true:** in fact all Herbrand interpretations over Σ have the same infinite universe $\{b, c, f(b), f(c), f(f(b)), f(f(c)), \dots\}$.
(7) **false:** $P(b) \vee P(c)$ has infinitely many Herbrand models over Σ, which differ in the interpretation of P on ground terms different from b and c.
(8) **true:** the interpretation of P on all ground terms with f at the root is fixed, but P can be either true or false for b and either true or false for c; this leaves four combinations.

Grading scheme: 5th, 6th, 7th, 8th correct answer: 2, 3, 3, 2 points.

Assignment 3

We construct a strict tableau for (1)–(3):

$$P(b) \wedge \neg P(d) \quad (1)$$
$$P(c) \vee \big(P(b) \wedge P(d)\big) \quad (2)$$
$$P(c) \rightarrow \neg\big(P(b) \vee P(d)\big) \quad (3)$$
$$P(b) \quad (4)$$
$$\neg P(d) \quad (5)$$

Left branch:

$$P(c) \quad (6)$$

- Left sub-branch: $\neg P(c)$ (10)
- Right sub-branch: $\neg\big(P(b) \vee P(d)\big)$ (11), $\neg P(b)$ (12), $\neg P(d)$ (13)

Right branch:

$$P(b) \wedge P(d) \quad (7)$$
$$P(b) \quad (8)$$
$$P(d) \quad (9)$$

We start with α-expansion of (1), this yields (4) and (5), then β-expansion of (2) yields (6) and (7), and α-expansion of (7) yields (8) and (9). The rightmost branch is now closed.

We continue with β-expansion of (3), this yields (10) and (11). The leftmost branch is now also closed.

Finally, α-expansion of (11) yields (12) and (13), so that the middle branch is closed as well.

Since every path is now closed, the set of input formulas is unsatisfiable.

Grading scheme: tableau: 8 points; explanation: 2 points.

Assignment 4

$$\dfrac{\dfrac{\dfrac{\dfrac{E \vdash f\,g\,f\,b \approx b}{E \vdash g\,f\,g\,f\,b \approx g\,b}}{E \vdash f\,g\,f\,g\,f\,b \approx f\,g\,b}}{E \vdash f\,g\,b \approx f\,g\,f\,g\,f\,b} \qquad E \vdash f\,g\,f\,g\,f\,b \approx b}{E \vdash f\,g\,b \approx b}$$

Note that the *Instance* rule (which is used to derive the two leaf formulas) does not have a premise.

Assignment 5

We start with the given equations (1)–(3).

$f(x,x) \approx f(x,b)$	(1)	$f(x,x) \to g(x)$	(4)
$f(x,x) \approx f(c,x)$	(2)	$f(x,b) \to g(x)$	(7)
$f(x,x) \approx g(x)$	(3)	$f(c,x) \to g(x)$	(8)
$g(x) \approx f(x,b)$	(5)	$g(b) \to g(c)$	(12)
$g(x) \approx f(c,x)$	(6)		
$g(b) \approx g(b)$	(9)		
$g(c) \approx g(c)$	(10)		
$g(b) \approx g(c)$	(11)		

Equations (1) and (2) cannot be oriented, so we apply "Orient" to replace (3) by (4). Now we can use "Simplify-Eq" twice with (4) to replace equation (1) by (5) and to replace equation (2) by (6). By applying "Orient" twice, we replace (5) and (6) by the corresponding rewrite rules (7) and (8). Using the critical pair between rules (4) and (7), the "Deduce" rule adds equation (9), which is trivial and gets eliminated using "Delete". Using the critical pair between rules (4) and (8), the "Deduce" rule adds equation (10), which is also trivial and gets eliminated using "Delete". Finally, using the critical pair between rules (7) and (8), the "Deduce" rule adds equation (11); this equation is replaced by the corresponding rewrite rule (12) using "Orient". Since all critical pairs between persisting rules have been computed and all equations have been eliminated, we can stop now; the final rewrite system is $\{(4), (7), (8), (12)\}$.

Grading scheme: starting with orienting the right input equation, i.e., (3): 4 points; simplifying the remaining equations afterwards: 4 points; computing the critical pair between (7) and (8): 4 points; computing the other critical pairs: 2 points; additional errors: −2 points.

Assignment 6

Part (a) Let R be the one-rule rewrite system $\{\, f(f(x)) \to f(g(f(x))) \,\}$. As shown in the lecture, R is not contained in any simplification ordering, so it is in particular not contained in any LPO. The system R has only one dependency pair, namely $f^\sharp(f(x)) \to f^\sharp(g(f(x)))$, and therefore only one node in the dependency graph. Since $\mathrm{ren}(\mathrm{cap}(f^\sharp(g(f(x))))) = f^\sharp(g(y))$ is not unifiable with $f^\sharp(f(x))$, the approximated dependency graph has no edges.

Part (b) Let R be the one-rule rewrite system $\{\, f(g(x)) \to f(x) \,\}$. Since $f(x)$ is embedded in $f(g(x))$, R is contained in *every* simplification ordering, so it is in particular contained in every LPO. The system R has only one dependency pair, namely $f^\sharp(g(x)) \to f^\sharp(x)$, and therefore only one node in the dependency graph. Since $\mathrm{ren}(\mathrm{cap}(f^\sharp(x))) = f^\sharp(y)$ is unifiable with $f^\sharp(g(x))$, the approximated dependency graph has an edge between the only node and itself.