Automated Reasoning I, 2017 Re-Exam, Sample Solution

Assignment 1

Part (a) Assume that $\phi(F)$ is satisfiable. Let the valuation $\mathcal{A}$ be a model of $\phi(F)$. We define a valuation $\mathcal{A}'$ by $\mathcal{A}'(P) = \mathcal{A}(P \vee Q)$ and $\mathcal{A}'(R) = \mathcal{A}(R)$ for every propositional variable $R \in \Pi$ different from P.

Now we can show by induction over the structure of formulas that $\mathcal{A}'(G) = \mathcal{A}(\phi(G))$ for every Π-formula G:

If $G = \bot$, then $\mathcal{A}'(\bot) = 0$ and $\mathcal{A}(\phi(\bot)) = \mathcal{A}(\bot) = 0$; analogously, if $G = \top$, then $\mathcal{A}'(\top) = 1$ and $\mathcal{A}(\phi(\top)) = \mathcal{A}(\top) = 1$. If $G = P$, then $\mathcal{A}'(P) = \mathcal{A}(P \vee Q)$ by definition of $\mathcal{A}'$ and $\mathcal{A}(\phi(P)) = \mathcal{A}(P \vee Q)$ by definition of ϕ; if G is a propositional variable R different from P, then $\mathcal{A}'(R) = \mathcal{A}(R)$ and $\mathcal{A}(\phi(R)) = \mathcal{A}(R)$. Finally, if $G = H_1 \vee H_2$, then $\mathcal{A}'(H_1 \vee H_2) = \max\{\mathcal{A}'(H_1), \mathcal{A}'(H_2)\} = \max\{\mathcal{A}(\phi(H_1)), \mathcal{A}(\phi(H_2))\}$ by induction and $\mathcal{A}(\phi(H_1 \vee H_2)) = \mathcal{A}(\phi(H_1) \vee \phi(H_2)) = \max\{\mathcal{A}(\phi(H_1)), \mathcal{A}(\phi(H_2))\}$; analogously, if $G = H_1 \wedge H_2$, then $\mathcal{A}'(H_1 \wedge H_2) = \min\{\mathcal{A}'(H_1), \mathcal{A}'(H_2)\} = \min\{\mathcal{A}(\phi(H_1)), \mathcal{A}(\phi(H_2))\}$ by induction and $\mathcal{A}(\phi(H_1 \wedge H_2)) = \mathcal{A}(\phi(H_1) \wedge \phi(H_2)) = \min\{\mathcal{A}(\phi(H)), \mathcal{A}(\phi(H_2))\}$; and if $G = \neg H$, then $\mathcal{A}'(\neg H) = 1 - \mathcal{A}'(H) = 1 - \mathcal{A}(\phi(H))$ by induction and $\mathcal{A}(\phi(\neg H)) = \mathcal{A}(\neg\phi(H)) = 1 - \mathcal{A}(\phi(H))$.

Since $\mathcal{A}(\phi(F)) = 1$, we conclude that $\mathcal{A}'(F) = 1$, so $\mathcal{A}'$ is a model of F.

Notes:

– It is unavoidable to fix *one* individual model $\mathcal{A}$ of $\phi(F)$ and to construct *one* valuation $\mathcal{A}'$ from $\mathcal{A}$ *before* starting the induction. One cannot replace this by a direct induction: First, the subformulas of a satisfiable formula need not be satisfiable (consider $F = \neg\bot$), so that one cannot apply the induction hypothesis to the subformulas. Second, if the subformulas of a formula are satisfiable, then this does not imply that the formula itself is satisfiable (consider $F = R \vee \neg R$).

Part (b) Let $F = P \vee \neg Q$, then F is not valid, but $\phi(F) = (P \vee Q) \vee \neg Q$ is valid.

Assignment 2

Define $f_{\mathcal{A}}(1) = 2$, $f_{\mathcal{A}}(2) = 1$, $f_{\mathcal{A}}(3) = 3$, and $c_{\mathcal{A}} = 1$.

Assignment 3

Part (a) The following literals are maximal in the clauses (1)–(5):

- Clause (1): literal 1 (literal 2 is smaller than literal 1).
- Clause (2): literals 1 and 2 (literals 3 and 4 are smaller than literal 2).
- Clause (3): literals 1 and 2.
- Clause (4): literals 1 and 2.
- Clause (5): literal 1.

From these, we get the following $\mathrm{Res}^{\succ}_{sel}$ inferences:

Clause (1) literal 1 and clause (2) literal 1 (after renaming x in clause (2) to x' to make the clauses variable-disjoint): $P(h(z), h(z)) \vee \neg P(y, f(f(f(h(z))))) \vee \neg Q(f(h(z))) \vee Q(f(f(h(z))))$ with $\sigma = \{x \mapsto h(z),\ x' \mapsto f(h(z))\}$.

Clause (1) literal 1 and clause (2) literal 2 (after renaming x in clause (2) to x' to make the clauses variable-disjoint): $P(f(x'), f(x')) \vee \neg P(h(z), x') \vee \neg Q(x') \vee Q(f(x'))$ with $\sigma = \{x \mapsto f(x'),\ y \mapsto f(x')\}$.

Clause (3) literal 2 and clause (4) literal 1 (after renaming x and y in clause (4) to x' and y' to make the clauses variable-disjoint): $\neg Q(h(f(x))) \vee Q(g(x'))$ with $\sigma = \{y' \mapsto h(b),\ y \mapsto g(c)\}$.

Grading scheme: 3 points for each of the three required inferences; -1 point for small mistakes; -3 points for each additional (incorrect) inference.

Part (b) Clause (3) is subsumed by clause (5): After applying $\sigma = \{y \mapsto f(x)\}$ to (5), the literals of (5) are a proper submultiset of the literals of (3). By Prop. 3.44, this means that clause (3) is redundant (i. e., every ground instance $\neg Q(h(f(t))) \vee R(h(b), t')$ of (3) is implied by a smaller ground instance $\neg Q(h(f(t)))$ of (5)).

Grading scheme: 3 points for identifying the redundant clause correctly; 2 points for the explanation.

Assignment 4

Part (a) We construct a tableau for the negation of the formula:

$$\begin{array}{c}
\neg((P \to Q) \to ((Q \to R) \to (P \to R))) \\
P \to Q \\
\neg((Q \to R) \to (P \to R)) \\
Q \to R \\
\neg(P \to R) \\
P \\
\neg R
\end{array}$$

$$\begin{array}{ccc}
\neg P & & Q \\
& \neg Q & & R
\end{array}$$

Since all paths are closed, the tableau is closed, so the original formula is valid.

Part (b) We construct a tableau for the negation of the formula:

$$\begin{array}{c}
\neg((R \wedge (R \to P)) \to (P \wedge \neg Q)) \\
R \wedge (R \to P) \\
\neg(P \wedge \neg Q) \\
R \\
R \to P
\end{array}$$

$$\begin{array}{cccc}
\neg R & & P & \\
& \neg P & & \neg\neg Q \\
& & & Q
\end{array}$$

The tableau is maximal and open. By Thm. 3.55 and Thm. 3.51, the negation of the original formula is satisfiable, so the original formula is not valid. (Actually, Thm. 3.55 does not require that the *tableau* is maximal and open, but only that there is one maximal and open *path*.)

Assignment 5

(1) **true:** $g(x) \succ x$ by the subterm property, so $f(g(x)) \succ f(x)$ by compatibility with contexts.
(2) **false:** If $t \succ t'$ in a simplification ordering, then $\mathrm{Var}(t') \subseteq \mathrm{Var}(t)$.
(3) **true:** In an LPO, $f(x) \succ g(x)$ implies $f \succ g$, hence $f(x) \succ g(g(x))$.
(4) **false:** Choose weight 3 for f and weight 2 for g.
(5) **false:** If $t \succ t'$ in a simplification ordering, then $\mathrm{Var}(t') \subseteq \mathrm{Var}(t)$.
(6) **true:** The variable condition for the KBO is satisfied. If f has a positive weight, then the weight of the first term is larger than the weight of the second term; if f has weight 0, then both terms have the same weight, and since $f(x) \succ x$, the argument tuple of the first term is lexicographically larger than the argument tuple of the second term.
(7) **false:** Otherwise we have $f(x) \succ g(f(x)) \succ g(g(f(x))) \succ g(g(g(f(x)))) \succ \ldots$, contradicting well-foundedness.
(6) **true:** The rewrite system $R = \{f(f(x)) \to f(g(f(x)))\}$ is terminating, so $\to_R^+$ is a reduction ordering with the desired property.

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

Assignment 6

Part (a) $R = \{f(b) \to c\}$.

Part (b) $R = \{f(b) \to f(c)\}$, then $DP(R) = \{f^\sharp(b) \to f^\sharp(c)\}$.

Part (c) $R = \{f(b) \to f(f(c))\}$, then $DP(R) = \{f^\sharp(b) \to f^\sharp(f(c)),\ f^\sharp(b) \to f^\sharp(c)\}$.