Automated Reasoning I, 2021/22
Endterm Exam, Sample Solution

Assignment 1

Since the derivation $\varepsilon \parallel N \Rightarrow^*_{\text{CDCL}} M \parallel N$ uses only "Unit Propagate", M does not contain any decision literals. Assume that neither "Unit Propagate" nor "Fail" can be applied to $M \parallel N$. If for some clause $C \in N$ all literals were false in M, then "Fail" would be applicable; if all literals but one were false and one literal were undefined, "Unit Propagate" would be applicable. Therefore, for every clause $C \in N$ either some literal is true in M or at least two literals are undefined in M. In the second case, at least one of these two literals must be negative, since C is a Horn clause. If we extend M to a total valuation by making all previously undefined atoms false, then those clauses in N that were already true in M remain true, and in all the other clauses one previously undefined negative literal becomes true, so we get a total model of N.

Assignment 2

Part (a) The following literals are maximal in clauses (1)–(5):
(1): literal 1;
(2): literal 1 and 2 (literal 3 is smaller than 1);
(3): literal 1 (literals 2 & 3 are smaller than 1);
(4): literal 1 (literal 2 is smaller than 1);
(5): literal 1 and 2.
We get the following three $\text{Res}^{\succ}_{sel}$ inferences:

Factorization (2) literals 1 and 2:
mgu $\{x \mapsto b,\, z \mapsto f(y)\}$,
conclusion $Q(g(b, f(y))) \vee \neg R(b, f(y))$ (6).

Resolution (3) literal 1, (4) literal 1
(after renaming x in (4) to x'):
mgu $\{x' \mapsto f(c),\, x \mapsto c\}$,
conclusion $\neg Q(f(c)) \vee \neg R(c, c) \vee Q(f(c))$ (7).

Resolution (3) literal 1, (5) literal 1
(after renaming x in (5) to x'):
mgu $\{x' \mapsto f(b),\, x \mapsto b\}$,
conclusion $R(c, y) \vee \neg Q(f(c)) \vee \neg R(b, c)$ (8).

Grading scheme:
3/3/2 points for three correct inferences,
-2 points further (incorrect) inferences,
-1 point for other errors, e.g., wrong unifiers.

Part (b) All conclusions of $\text{Res}^{\succ}_{sel}$ inferences between the clauses in N are redundant w.r.t. N: Clauses (6) and (8) are subsumed by (1); clause (7) contains complementary literals and is therefore a tautology. So N is saturated up to redundancy.

Part (c) If sel selects literal 3 in (2), literal 2 and/or 3 in (3), literal 1 in (5), and optionally literal 1 in (1) and/or literal 1 in (4), then there are no $\text{Res}^{\succ}_{sel}$ inferences between the clauses (1)–(5).

Assignment 3

Part (a) The shortest rewrite proof has the form $b \leftarrow_E f(f(f(t))) \rightarrow_E f(b)$, where the term t can be chosen arbitrarily. (There are also more complicated rewrite proofs that consist of more than two steps.)

Notes:

- The assignment asked how the rewrite proof of $b \leftrightarrow^*_E f(b)$ looks, it did not ask for a derivation of $E \vdash b \approx f(b)$ using the congruence axioms.
- $\leftrightarrow^*_E$ is the reflexive-transitive closure of $\leftrightarrow_E$ (that is, $\rightarrow_E \cup \leftarrow_E$). It is neither the union nor the intersection of $\rightarrow^*_E$ and $\leftarrow^*_E$. In fact, neither $b \rightarrow^*_E f(b)$ nor $b \leftarrow^*_E f(b)$ holds, since every rewrite proof of $b \leftrightarrow^*_E f(b)$ must contain at least one $\rightarrow_E$ step and one $\leftarrow_E$ step.

Part (b) The universe of $\mathrm{T}_\Sigma(\emptyset)/E$ consists of 5 congruence classes, namely $[c] = \{c\}$, $[d] = \{d\}$, $[f(c)] = \{f(c)\}$, $[f(d)] = \{f(d)\}$, and $[b]$. The latter contains all remaining ground terms, that is, b, $f(b)$, and all terms of the form $f(f(t))$ with $t \in \mathrm{T}_\Sigma(\emptyset)$.

Assignment 4

Since an LPO is a simplification ordering, we know that $h(t) \succ_{\text{lpo}} t$, so $s \succeq_{\text{lpo}} h(t)$ implies $s \succeq_{\text{lpo}} h(t) \succ_{\text{lpo}} t$ and thus $s \succ_{\text{lpo}} t$ by transitivity. This proves the "if" part.

The "only if" part is proved by induction over $|s| + |t|$. First assume that the top symbol f of s is different from h. Since h is the smallest element of the precedence, we have $f \succ h$, so $s \succ_{\text{lpo}} t$ implies $s \succ_{\text{lpo}} h(t)$ by Case (2b).

Otherwise $s = h(s')$.

If $h(s') \succ_{\text{lpo}} t$ by Case (1), then t is a variable that occurs in $h(s')$. So either $s' = t$, then $s = h(s') = h(t)$, or $s' \rhd t$, then $s' \succ_{\text{lpo}} t$, and by compatibility with contexts $s = h(s') \succ_{\text{lpo}} h(t)$.

If $h(s') \succ_{\text{lpo}} t$ by Case (2a), then $s' = t$ or $s' \succ_{\text{lpo}} t$. In the first case $s = h(s') = h(t)$, in the second case $s = h(s') \succ_{\text{lpo}} h(t)$ by compatibility with contexts.

We cannot have $h(s') \succ_{\text{lpo}} t$ by Case (2b) since h is the smallest element of the precedence. So it remains to consider the case that $h(s') \succ_{\text{lpo}} t$ by Case (2c). Then $t = h(t')$ and $s' \succ_{\text{lpo}} t'$. By induction, we get $s' \succeq_{\text{lpo}} h(t')$. So $s' = h(t')$ or $s' \succ_{\text{lpo}} h(t')$, therefore $s = h(s') = h(h(t')) = h(t)$ or $s = h(s') \succ_{\text{lpo}} h(h(t')) = h(t)$ by Case (2c).

Grading scheme:
4 points for the "if" part,
10 points for the "only if" part.

Assignment 5

To simplify the notation, we omit parentheses after unary operators.

We start with the given equations (1)–(2).

$$
\begin{array}{rr@{\qquad}rr}
f\,g\,x \approx h\,x & (1) & f\,g\,x \to h\,x & (3)\\
g\,f\,x \approx h\,x & (2) & g\,f\,x \to h\,x & (4)\\
h\,f\,x \approx f\,h\,x & (5) & f\,h\,x \to h\,f\,x & (6)\\
h\,g\,x \approx g\,h\,x & (7) & g\,h\,x \to h\,g\,x & (8)\\
h\,h\,x \approx g\,h\,f\,x & (9) & & \\
h\,h\,x \approx h\,g\,f\,x & (10) & & \\
h\,h\,x \approx h\,h\,x & (11) & & \\
h\,h\,x \approx f\,h\,g\,x & (12) & & \\
h\,h\,x \approx h\,f\,g\,x & (13) & & \\
h\,h\,x \approx h\,h\,x & (14) & &
\end{array}
$$

By applying "Orient" twice, we replace (1)–(2) by the corresponding rewrite rules (3)–(4).

Using the critical pair between rules (3) and (4), "Deduce" adds equation (5). Then "Orient" replaces equation (5) by rule (6).

Using the critical pair between rules (4) and (3), "Deduce" adds equation (7). Then "Orient" replaces equation (7) by rule (8).

Using the critical pair between rules (4) and (6), "Deduce" adds equation (9). "Simplify-Eq" uses rewrite rule (8) to replace equation (9) by equation (10); then "Simplify-Eq" uses rewrite rule (4) to replace equation (10) by equation (11). Equation (11) is trivial, so it can be eliminated using "Delete".

Using the critical pair between rules (3) and (8), "Deduce" adds equation (12). "Simplify-Eq" uses rewrite rule (6) to replace equation (12) by equation (13); then "Simplify-Eq" uses rewrite rule (3) to replace equation (13) by equation (14). Equation (14) is trivial, so it can be eliminated using "Delete".

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 $\{(3), (4), (6), (8)\}$.

Grading scheme:
3/3/2/1 points for four critical pairs,
2/1/1/1 points for four simplifications,
up to -2 points for other errors.

Assignment 6

Part (a)

$R = \{f(h(x)) \to g(x),\ g(h(x)) \to f(x)\}$;
$\text{DP}(R) =$
$\{f^\sharp(h(x)) \to g^\sharp(x),\ g^\sharp(h(x)) \to f^\sharp(x)\}$.

Part (b)

$R = \{f(h(x)) \to f(x),\ g(h(x)) \to g(x)\}$;
$\text{DP}(R) =$
$\{f^\sharp(h(x)) \to f^\sharp(x),\ g^\sharp(h(x)) \to g^\sharp(x)\}$.

Part (c)

$R = \{f(h(x)) \to f(x),\ f(g(x)) \to f(h(x))\}$;
$\text{DP}(R) =$
$\{f^\sharp(h(x)) \to f^\sharp(x),\ f^\sharp(g(x)) \to f^\sharp(h(x))\}$.

Grading scheme:
2 points if R is not terminating or if the condition $l \not\rhd u$ is violated; no points if there is more than one error or if the graph does not have the desired shape.