**1** Consider the formula

$$(\overline{1} \vee \overline{2}) \wedge (\overline{1} \vee 2 \vee 3) \wedge (4 \vee 5) \wedge (\overline{6} \vee \overline{5} \vee 7) \wedge (\overline{3} \vee \overline{7} \vee \overline{8}) \wedge (\overline{3} \vee 8 \vee \overline{9}) \wedge (9 \vee \overline{10}) \wedge (\overline{3} \vee 8 \vee 10)$$

and suppose a DPLL inference sequence reached the state $1^d\, \overline{2}\, 3\, \overline{4}^d\, 5\, 6^d\, 7\, \overline{8}\, \overline{9}\, \overline{10}$.

[3] (a) Construct an implication graph and give three different cuts together with the induced implied (backjump) clauses. Which nodes are UIPs?

[3] (b) Give an implied clause that has as few literals as possible (among the implied clauses that can be obtained from the graph). Derive this clause by resolution from the conflict.

**2** Consider the following formula $\varphi$:

$$\mathsf{a} \approx \mathsf{b} \wedge (\mathsf{f}(\mathsf{a}) \not\approx \mathsf{f}(\mathsf{b}) \vee \mathsf{f}(\mathsf{a}) \not\approx \mathsf{c}) \wedge (\mathsf{a} \not\approx \mathsf{b} \vee \mathsf{f}(\mathsf{c}) \not\approx \mathsf{c}) \wedge \mathsf{f}(\mathsf{b}) \approx \mathsf{f}(\mathsf{f}(\mathsf{a}))$$

[1] (a) Construct a propositional skeleton of $\varphi$.

[5] (b) Determine satisfiability of $\varphi$ using DPLL($T$). Explain which DPLL($T$) inference rules are used, which EUF problems appear and how they are solved, and why the formula is eventually determined to be (un)satisfiable.

**3** Consider the following set of equations $E$:

$$\mathsf{f}(\mathsf{a}) \approx \mathsf{a} \qquad \mathsf{f}(\mathsf{f}(\mathsf{a})) \approx \mathsf{f}(\mathsf{b}) \qquad \mathsf{g}(\mathsf{a},\mathsf{b}) \approx \mathsf{b} \qquad \mathsf{f}(\mathsf{c}) \approx \mathsf{g}(\mathsf{a},\mathsf{c}) \qquad \mathsf{c} \approx \mathsf{g}(\mathsf{c},\mathsf{a})$$

and use congruence closure to determine whether the following hold:

[3] (a) $E \models \mathsf{g}(\mathsf{f}(\mathsf{f}(\mathsf{a})),\mathsf{b}) \approx \mathsf{b}$

[2] (b) $E \models \mathsf{f}(\mathsf{c}) \approx \mathsf{f}(\mathsf{g}(\mathsf{a},\mathsf{c}))$

**4** Suppose a run of the Simplex algorithm with variable order $x > y > s_1 > s_2 > s_3 > s_4$ reached the following intermediate state:

$$\begin{array}{c} \\ s_1 \\ x \\ y \\ s_4 \end{array} \begin{array}{c} \begin{array}{cc} s_2 & s_3 \end{array} \\ \begin{pmatrix} 2 & 1 \\ -1 & -1 \\ 1 & 0 \\ -4 & -3 \end{pmatrix} \end{array} \quad \begin{array}{l} \\ s_1 \leq 1 \\ s_2 \leq 4 \\ s_3 \leq -6 \\ s_4 \leq 7 \end{array}$$

with assignment $x = 2$, $y = 4$, $s_1 = 2$, $s_2 = 4$, $s_3 = -6$, $s_4 = 2$.

[2] (a) Which variable pairs are suitable for a pivot step, and why?

[5] (b) Perform one possible pivot step, and compute the new assignment. Does it satisfy the constraints?

5 Consider bitvector variables $\mathbf{a}_3$, $\mathbf{b}_3$, and $\mathbf{c}_3$ of three bits each. Suppose that $\mathbf{a}_3 = a_2a_1a_0$, with $a_2$ being the most significant bit (so $\mathbf{a} = 2^0 \cdot a_0 + 2^1 \cdot a_1 + 2^2 \cdot a_2$), and similarly for $\mathbf{b}_3$ and $\mathbf{c}_3$.

[4] (a) Give a propositional formula which encodes the signed comparison $\mathbf{a}_3 >_u (\mathbf{b}_3 \mathbin{\&} \mathbf{c}_3)$. Here & denotes bitwise AND, and $>_u$ denotes unsigned greater-than comparison.
For example, $\mathbf{2}_3 >_u (\mathbf{7}_3 \mathbin{\&} \mathbf{1}_3)$ is true, while $\mathbf{2}_3 >_u (\mathbf{7}_3 \mathbin{\&} \mathbf{4}_3)$ is false.

[2] (b) Give a propositional formula which encodes that the multiplication $\mathbf{a}_3 \times \mathbf{b}_3$ overflows, i.e., the result is no longer representable in 3 bits.

⋆ 6 Consider the following formula $\psi$ combining uninterpreted functions and linear real arithmetic:

$$(z = 0) \wedge (x + z \geqslant y) \wedge (y \leqslant x) \wedge (\mathsf{f}(z) > z) \wedge (\mathsf{f}(x) = \mathsf{f}(z) + \mathsf{f}(y))$$

[2] (a) Purify $\psi$.

[2] (b) Use the Nelson-Oppen procedure to determine satisfiability of $\psi$. You can use either the deterministic or the nondeterministic version.

Exercises marked with a ⋆ are optional and give extra points.