Logic and Computability SS24, Assignment 1

Due: 20. 03. 2024, 23:59

The DPLL-Algorithm

For the following exercises use the DPLL algorithm (including Boolean Constraint Propagation (BCP), pure literals, and conflict-driven clause learning) to check on paper, if the following CNF formulas are satisfiable.
If the formula is satisfiable, give a satisfying model, else show a complete resolution proof for the formula's unsatisfiability.

- Write down all the steps of the DPLL algorithm,
- draw the conflict graphs,
- and state the resolution proofs for all learned clauses.

Rules:

- When resolving a conflict, only undo the last decision.
- Choose variables for decisions, BCP and pure literals in alphabetical order, starting with the *negative* phase ($\neg a > a > \neg b > b...$).
- Always try to perform BCP first, before checking for pure literals, before making a decision.

1. [3 Points] Use the DPLL algorithm with the rules as described above to check whether the following formula in CNF is satisfiable

   Clause 1: $\{a, \neg b, c\}$

   Clause 2: $\{b, \neg c, d\}$

   Clause 3: $\{a, \neg b\}$

   Clause 4: $\{a, c\}$

   Clause 5: $\{\neg c, \neg d\}$

   Clause 6: $\{\neg a, c\}$

2. [2 Points] Use the DPLL algorithm with the rules as described above to check whether the following formula in CNF is satisfiable

   Clause 1: $\{a, b, c\}$

   Clause 2: $\{\neg b, \neg c, e\}$

   Clause 3: $\{b, e\}$

   Clause 4: $\{b, \neg d\}$

   Clause 5: $\{\neg c, d\}$

   Clause 6: $\{\neg c, e\}$

   Clause 7: $\{\neg a, \neg b, \neg c\}$

   Clause 8: $\{a, c, \neg e\}$

3. [3 Points] Use the DPLL algorithm with the rules as described above to check whether the following formula in CNF is satisfiable

   Clause 1: $\{a, b\}$

   Clause 2: $\{\neg b, c\}$

   Clause 3: $\{\neg a, \neg c\}$

   Clause 4: $\{b, c\}$

   Clause 5: $\{a, \neg b\}$

   Clause 6: $\{\neg b, \neg c\}$

4. [3 Points] Construct a reduced ordered binary decision diagram (ROBDD) for the formula

$$f = (p \vee q) \wedge \neg(p \wedge q) \wedge r$$

using *variable order* $q < p < r$. Use complemented edges and a node for `true` as the only constant node. To simplify drawing, you may assume that *dangling edges* point to the constant node. Write down all cofactors that you compute to obtain the final result and mark them in the graph.

5. [3 Points] Construct a reduced ordered binary decision diagram (ROBDD) for the formula

$$f = (a \vee \neg b) \wedge \neg(c \vee d) \vee (a \wedge b)$$

using *variable order* $a < b < c < d$. Use complemented edges and a node for `true` as the only constant node. To simplify drawing, you may assume that *dangling edges* point to the constant node. Write down all cofactors that you compute to obtain the final result and mark them in the graph.