Satisfiability: Davis-Putnam (DP) and Davis-Logemann-Loveland (DLL) Algorithms

1. This project must be done and submitted individually.
2. Submit one ZIP file, called `<student number>.zip` to the folder "Project 3 Submissions" in Luminus by **Monday 11 March midnight**.
3. The ZIP file contains the file `<student number>.pl`, a commented prolog program with your answers.
4. After the deadline and until Monday 18 March at midnight, you can submit to the folder "Project 3 Late Submissions" in Luminus (penalties apply).

Question 1 [15 marks]

The propositional Davis-Putnam (DP) and the Davis-Logemann-Loveland (DLL) algorithms are checking the satisfiability (SAT) of a set of propositional clauses.

A set of propositional clauses (or, equivalently, a propositional formula in conjunctive normal form) is represented as a list of lists. The outer list represents the set (or, equivalently, the conjunction). The inner lists represent the clauses. Propositions are Prolog atoms. Lliterals are propositions or negations of propositions. The negation of a proposition is indicated with the unary functor `not/1` with the proposition as argument.

For example, formula in conjunctive normal form:

$$((x_1 \vee x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_1) \wedge (x_1 \vee \neg x_2 \vee \neg x_3))$$

is represented by the term:

```
[[x1, x2, not(x3)], [x2, not(x1)], [x1, not(x2), not(x3)]]
```

(a) Write a Prolog procedure, `dp/1`, that implements DP as described in the lecture adding the necessary and convenient refinements (e.g. duplicate literal elimination in clause, elimination of tautoligical clauses ($l \in C$ and $\neg l \in C$), elimination of duplicate clauses, eliminiation of subsumed clauses (C_1 subsumes C_2 if $C_1 \subset C_2$), etc.). `dp(+CNF)` succeeds if the set of clauses represented by the ground term `CNF` is satisfiable and fails otherwise. The code and the comments in the code are taken into account for the evaluation. (5)

(b) Write a Prolog procedure, `dll/1`, that implements DLL as described in the lecture. Do not implement the pure literal simplification. `dll(+CNF)` succeeds if the set of clauses represented by the ground term `CNF` is satisfiable and fails otherwise. The code and the comments in the code are taken into account for the evaluation. (5)

(c) The remaining marks are competitively attributed to the fastest sound and complete implementations (for a non-disclosed set of test examples). (5)

– END OF PAPER –