Project 1 – deadline 1st December 2020

The project will consist of two parts:

1. implementation of the algorithms below (preferably in PROLOG; any other language is allowed but with 2 points penalization).
2. a written document, where you treat the two subjects below (between 2-10 pages without the code; the code is added at the end of the document).

You must work individually for the project. **Attention:** your projects (in pdf format) will be checked for similarities with Turnitin.

In the last 2 weeks before the winter holiday, I will meet individually with each of you on MS Teams to present your projects (30 minutes for each student, there will be a planning for that). **Attention:** your project is not graded unless you present it.

I will ask you questions about the code and about what you write in the document (you should not include any comments in your code; **Attention:** a program with any comments included will not be considered at all).

All the resources (on internet, books) that you consult for the project must be cited.

If you have questions, my email is cidota@fmi.unibuc.ro

You are required to solve two subjects:

1. Resolution:
   Create your own KB and a Question (entailed from KB), expressed in natural language (as the example in the last slide of C3). You can use other examples for inspiration (and indicate the source!), but you are not allowed to copy them exactly. Your KB represented in FOL must contain variables (not like the "Toddler" example in C3 page 18).
   a) Represent your KB in FOL, using a vocabulary that you will define.
   b) Prove "manually" (as in C3 page 19) that the Question is logically entailed from KB, by applying Resolution (only this part can be handwritten, scanned and inserted into the project).
   c) Use your implementation to prove that the Question is logically entailed from KB (explain how clauses are chosen for Resolution; different optimization techniques that you use)
   d) Use your implementation for the following finite sets of propositional clauses, written in CNF:
      i. [[¬a,b],[c,d],[¬d,b],[¬c,b],[¬b]]
      ii. [[¬b,a],[¬a,b,e],[e],[a, ¬e],[¬a]]
      iii. [[¬a,b],[c,f],[¬f,b],[¬c,b],[¬c]]
      iv. [[a,b],[¬a, ¬b]]

2. SAT solver – The Davis Putnam procedure
   Implement the Davis-Putnam SAT procedure. For S, a set of clauses in written in CNF, the procedure will display YES, respectively NOT, as S is satisfiable or not. In the case of YES, the procedure will also display the truth values assigned to the literals (e.g. {w/true; s/false; p/false ...}). Choose two strategies of selection of the atom to perform the • operation and discuss/compare the results.
   Use your implementation (with both strategies) for the following finite sets of propositional clauses, written in CNF:

   i. [[Toddler],[¬Toddler,Child],[¬Child,¬Male,Boy],[¬Infant,Child], [¬Child,¬Female,Girl], [Female], [Girl]]
   ii. [[Toddler],[¬Toddler,Child],[¬Child,¬Male,Boy],[¬Infant,Child], [¬Child,¬Female,Girl], [Female], [¬Girl]]
   iii. [[¬a,b],[c,d],[¬d,b],[¬c,b],[¬b]]
   iv. [[¬b,a],[¬a,b,e],[e],[a, ¬e],[¬a]]
   v. [[¬a,¬e,b],[¬d,e,¬b],[¬e,f,¬b],[f,¬a,e],[e,f,¬b]]
   vi. [[a,b],[¬a,¬b] ,[¬a,b] ,[a,¬b]]

Note: both procedures will be implemented in the versions presented at the course (from Ronald Brachman, Hector Levesque. Knowledge representation and reasoning, Morgan Kaufmann 2004).

The clauses will be represented in whatever format you choose. The input data will be read from a file and the results will be displayed on the screen.

Each subject will receive a grade. The grade for this project will be 60%G1+40%G2.

Marina Cidota

04.11.2020