DEEC
DEPARTAMENTO DE ENGENHARIA
ELETROTÉCNICA E DE COMPUTADORES
TÉCNICO LISBOA

Instituto Superior Técnico

Artificial Intelligence and Decision Systems (IASD)
Lab classes, 2014/2015

Assignment #2

Version 1.0 - October 22, 2014

A propositional logic reasoner based on the Resolution principle consists of the following key elements:

- a program to convert logical sentences in propositional logic into the clausal normal form (CNF).
- a resolution-based theorem prover for propositional logic, assuming a CNF knowledge base.

The goal of this lab assignment is to create, in Python, the first element (CNF converter) of the resolution-based propositional logic reasoner.

1 CNF converter

The main function of the converter takes as argument a sentence, and returns a list of disjunctions (which are lists themselves).

The input sentence is represented as a syntactic tree, using tuples, according to the following rules:

atom	$\longrightarrow$	*string*
negation	$\longrightarrow$	`('not',` *sentence* `)`
conjunction	$\longrightarrow$	`('and',` *sentence* `,` *sentence* `)`
disjunction	$\longrightarrow$	`('or',` *sentence* `,` *sentence* `)`
implication	$\longrightarrow$	`('=>',` *sentence* `,` *sentence* `)`
equivalence	$\longrightarrow$	`('<=>',` *sentence* `,` *sentence* `)`

The output should be a list of disjunctions, where each one of them is a list of literals; each literal can be a symbol (*i.e.,* a string) or a negation of a symbol (as above).

Suggestions:

- Write a set of small auxiliary functions to test if a given sentence is an atom, a negation, a conjunction, etc.

- Work recursively, and test for the aplicability of each transformation rule for each subsentence in the recursion.
- Consider not only the transformation rules, but also simplification rules.

Program desirable features

Besides solving the main problem, *i.e.*, convert a set of sentences to CNF, the program should:

- read all sentences to convert from an input file;
- have an option to list all sentences before conversion;
- have an option to convert one of those sentences explaining the CNF conversion process step by step;
- have an option to list all sentences after conversion;
- write everything in an output file.

An example of an input file is:

```
('=>', 'Mythical', 'Immortal')
('=>', ('not', 'Mythical'), ('and', ('not', 'Immortal'), 'Mammal'))
('=>', ('or', 'Immortal', 'Mammal'), 'Horned')
('=>', 'Horned', 'Magical')
```

The deliverable of this Lab Assignment consists of two components:

- the Python source code
- the short report with no more than 2 pages.

Deadline: 24h00 of 21-Nov-2014, by email to lmmc@isr.ist.utl.pt