

Assignment 3

Problem: Resolution in First Order Logic

In this assignment you will implement First-Order Resolution to check the satisfiability of sentences in \mathcal{P}_1 . The datatype you will be using in this assignment is :

```
signature FOL = sig
  datatype Term = CONST of string
                | VAR of string
                | FUNC of string * Term list

  datatype Form = TOP1                                     | BOTTOM1
                | PRED of string * Term list              | NOT1 of Form
                | AND1 of Form * Form                     | OR1 of Form * Form
                | IMP1 of Form * Form                     | IFF1 of Form * Form
                | ITE1 of Form * Form * Form
                | FORALL of string * Form
                | EXISTS of string * Form

  val makePrenex : Form -> Form
  val makePCNF : Form -> Form
  val makeSCNF : Form -> Form
  val resolve : Form -> bool
end
```

0. Implement algorithms to convert formulae in prenex normal forms using the function `makePrenex`. Implement `makePCNF` to convert this into a PCNF by converting the quantifier-free part into a CNF.
1. Implement algorithms to skolemize by generating new temporary constant and function symbols and convert a PCNF into a SCNF. Implement the function `makeSCNF`.
2. Implement first-order resolution for closed formulae. To make it simple and deterministic it might be a good idea to restrict the algorithm to Horn clauses, i.e., each clause has only one positive literal. Implement the function `resolve`.

The function `resolve` should first call `makePrenex`, then `makePCNF`, then `makeSCNF` and finally apply resolution on the result.