CITS3005 Knowledge Representation Homework Practise: 2022

August 15, 2023

This is a set of exercises for discussion in the workshop, and are the set homework exercises from 2022. They required students to complete three tasks demonstrating: formal representations of knowledge and reasoning; logic programming; and knowledge representation.

1. Show how the following logic puzzle can be formalised in first order logic, and resolution can be applied to determine the preferred editor of each programmer.

   There are four programmers, Karen, Kamal, Kevin *and* Kiri *who work at the start up company,* KnowWay. *They have employee numbers:* 1, 2, 3 *or* 4, *reflecting the order they started working in the company, and they each have a preferred programming environment:* Vim, Emacs, VSCode, *or* Notepad. *You are told the following facts:*

   - *Kevin is employee number 2.*
   - *The programmer who prefers VSCode was employed before Kevin.*
   - *Employee number 4, Kamal and the programmer who prefers Notepad are all different people.*
   - *The programmer who prefers Notepad was employed directly before Karen.*
   - *Employee number 3 is either the programmer who prefers Emacs or Kevin.*

   (a) Formalise this scenario in first order logic, clearly identifying the predicates, functions and constants used, and expressing each fact as a proposition in first order logic. **10 marks**

   (b) Taking your answer to the first part as a knowledge base, convert the knowledge base to conjunctive normal form and apply resolution to find out who prefers Notepad. **10 marks**

   To use logical symbols you can choose to hand write your answers and upload a scan, use the Latex typesetting package, use symbols in an word processing application, or use the following ASCII representations:

Symbol	$\neg a$	$a \wedge b$	$a \vee b$	$a \to b$	$\exists X a$	$\forall X a$
ASCII	`~a`	`a /\ b`	`a \/ b`	`a -> b`	`Exists X. a`	`Forall X. a`

2. Write a logic program to solve the satisfiability problem for propositional logic. Specifically:

   - Your program should have *constants* `t` (true) and `f` (false).
   - Your program should have *functions* `atom`, `or`, `and`, `not`, `implies`.
   - your program should have a *predicate* `sat`, that is true if the proposition provided as an argument is satisfiable.

   For example, if given the query:
   `?- sat(and(or(X,Y),and(not(X),Y))).`
   the program should respond with: `X = atom(f)`
   `Y = atom(t)`

   If you have inifnite loops, try the alternative formulation:
   `?- sat(and(or(atom(X),atom(Y)),and(not(atom(X)),atom(Y)))).`
   the program should respond with: `X = f`
   `Y = t`

   20 marks

3. Provide a natural language description of the resolution rule, giving a clear description of the following terms and the relations between them:

   term. predicate. literal. conjunction. disjunction. clause. conjunctive normal form. substitution. unifier. resolution rule.

   10 marks