

Due: October 5th

This assignment involves writing key parts of a resolution-style theorem prover and testing it on a knowledge base involving knowledge about computers. Your system should accept formulas in a LISP-style FOPC syntax as follows:

$\langle \text{formula} \rangle ::= \langle \text{prop} \rangle \mid (\text{all } \langle \text{var} \rangle \langle \text{formula} \rangle) \mid (\text{exists } \langle \text{var} \rangle \langle \text{formula} \rangle) \mid (\text{and } \langle \text{formula} \rangle^*)$   
 $\mid (\text{or } \langle \text{formula} \rangle^*) \mid (\text{not } \langle \text{formula} \rangle) \mid (\text{implies } \langle \text{formula} \rangle \langle \text{formula} \rangle)$

$\langle \text{prop} \rangle ::= (\langle \text{pred-name} \rangle \langle \text{term} \rangle^*)$

$\langle \text{term} \rangle ::= \langle \text{constant} \rangle \mid \langle \text{var} \rangle \mid \langle \text{function} \rangle$

$\langle \text{function} \rangle ::= (\langle \text{fn-name} \rangle \langle \text{term} \rangle^*)$

where  $\langle \text{constant} \rangle$ ,  $\langle \text{var} \rangle$ ,  $\langle \text{fn-name} \rangle$  and  $\langle \text{pred-name} \rangle$  are any atoms consisting of alphanumeric letters (a to z, 0 to 9). Note you can tell what type of object they are by where they appear in the formulas.

AS an example, the knowledge that all computers are either laptops or desktops might be expressed as the following three formulas:

$(\text{all } x1 (\text{implies } (\text{computer } x1) (\text{or } (\text{desktop } x1) (\text{laptop } x1))))$  ; i.e., all computers are either desktops or laptops

$(\text{all } x2 (\text{implies } (\text{desktop } x2) (\text{computer } x2)))$  ; i.e., all desktops are computers

$(\text{all } x2 (\text{implies } (\text{laptop } x3) (\text{computer } x3)))$  ; i.e., all laptops are computers

### A. The program

1. Write a function, ADD, that takes a LISP-style FOPC formula, converts it to a set of clauses, and adds them to your KB.
2. Write a function RESOLVE-ALL, that takes a clause and creates the set of all possible clauses that can be produced by any possible unifications of this clause into the knowledge base.
3. (447 students, or bonus for 247) Using the RESOLVE-ALL function, develop a full resolution proof procedure, PROVE, which takes a FOPC formula and returns T or NIL depending on whether it is entailed by the KB or not. Outline what search strategies you employed in order to try to get efficient search, and explain why you think it would be effective.

Pay particular attention in your documentation to how variables are represented and how this representation supports an effective unification algorithm.

### B. Developing a Knowledge Base

Develop an ontology and express the following facts in the LISP-style FOPC. Discuss the ontological decisions you made when deciding how best to represent the facts. Look at all the facts and the types of queries we want to make before deciding how to represent things. In particular, note that we want to make queries that explore how different objects are related. Your ontology will need to be designed to make such queries possible.

All computers are either laptops or desktops.

Every computer has a CPU, a memory chip, a hard disk, and a monitor.

Every CPU has a clock speed.

A Clock speed consists of a unit, *gigahertz*, and a number.

Every memory chip has a memory size.

Every hard disk has a memory size.

A memory size consists of a unit, *megabyte*, and a number.

Every monitor has a screen size.

A screen size consists of a unit, *inches*, and a number.

Your ontology should be designed so it can handle the following types of queries. These are motivated by a problem in natural language understanding where noun-noun modifications occur without identifying the relationship between the two objects. For example, "A 2 GHz CPU" refers to a CPU processor that has a clock speed of 2 GHz. More complicated cases involve intermediate objects that are not mentioned: e.g., a "2 GHz computer" refers to a computer that has a CPU that has a clock speed of 2 GHz. These would be the answers to queries like:

Is there a relationship between a CPU and 2 GHz?

Is there a relationship between a computer and 2 GHz?

Make sure that ontology is well described and justified. Explicitly describe how your representation makes reasoning about "relatedness" possible.

### **C. Testing**

Using your ADD function, add all the knowledge developed in part B to your knowledge base. Then demonstrate your RESOLVE-ALL function by testing it with the clauses that would be derived when trying to prove the above two queries. CSC447 students should actually run the full proof of the queries.

Make sure you also test your programs on some additional data that demonstrate the range of capabilities (and limitations) of your program.

### **D. Submit Code**

Submit your code for testing on some unknown knowledge base and queries in the usual way. Check the course web site for instructions.