University of Pittsburgh
CS 2740 Knowledge Representation
Professor Milos Hauskrecht

Handout 2
September 10, 2008

Problem assignment 1

Due: Wednesday, September 17, 2008

Lisp

The goal of this assignment is to practice your lisp programming skills on two simple problems.

Problem 1. Your background

Please write a paragraph about yourself, the department/program you are in, your educational background, research objectives.

Problem 2. Priority queue

Priority queue is a structure often used in AI to support search routines. Let us assume we have a set of objects and each object is assigned a priority weight. The object priority-weight pair can be represented by a list (object priority-weight).

Write the following functions to support the priority queue operations:

- add-queue (queue element): that takes two arguments: a priority queue, and the new object-weight pair (element) to be inserted into the queue, and returns the updated queue.
- top-queue (queue): that takes a queue and returns the element with the highest priority weight.
- remove-top-queue (queue): that takes a queue, removes the element with the highest weight from the queue and returns the new queue.

Hint: the queue can be implemented as a list of object-weight pairs.

After thoroughly testing the function, please include it in problem1-2.lisp file and submit.

Problem 3. Simple logic evaluator

In this assignment, we implement a function that lets us evaluate the sentences in the propositional logic and determines its truth value.

Sentences in propositional logic correspond to either elementary (atomic) or composite sentences. Composite sentences are built from atomic propositions using logical connectives. Let us assume the following symbols are used to represent logical connectives in Lisp:

- negation (not A)
- logical-and, (and A B C)
- logical-or, (or A B D)
- implication, (impl A B)
- biconditional, (bic A B)

where symbols A,B,C, etc correspond to either atomic propositions (with True or False values) or another sentence. For example, the expression:

$$(A \vee B \vee \neg C) \wedge (\neg A \vee D) \wedge \neg B$$

will be represented as:
(and (or A B (not C)) (or (not A) D) (or (not B))).

Write a lisp function eval-logic that takes (1) the logical expression (sentence) written using the above representation, (2) the assignment of truth values to all atomic propositions (represented as a list of assignments), and returns true if the logical expression is satisfied. For example, the following Lisp code:

(setq atomicprop '((A T) (B NIL) (C T) (D NIL)))
(setq sentence '(and (or A B (not C)) (or (not A) D) (or (not B))))
(eval-logic sentence atomicprop)

should return NIL (or False).

After thoroughly testing the function/s, please include it in problem1-3.lisp file and submit.

Problem 4. Propositional logic

Translate the following sentences in English into propositional logic. For each sentence you should try to identify propositions that allow you to express as much structure in the sentence as possible.

- It is below freezing and snowing.
- It is below freezing but not snowing.
- If it is below freezing, it is also snowing.
- It is either below freezing, or it is snowing, but it is not snowing if it is below freezing.
- If it rains in the afternoon we will go shooping.
- We will go shopping only if it rains in the afternoon.
- We will go shopping if and only if it rains in the afternoon.