Talal Jawaid

4/25/2019

CSC 135

Professor Lu

* 1. Logic Programming systems are also called \_\_\_\_\_\_**deductive**\_\_\_\_\_\_ databases
  2. The process of pattern matching to make statements identical is called \_\_\_**unification**\_\_\_.

**2**. **(12%)** G**ive a concise answer to each questions below:**

(a) What are the differences between procedural programming and logic programming?

Four main differences between procedural and logic programming

- architecture

- syntax

- computation

- control

Procedural programming

Von Neumann Machine – multiple steps

Syntax: sequence of statements(a,s,I)

Computation: Sequential statements execution

Control : Logic and Control Mixed together

Logic Programming

Abstract model ( dealing with objects and their relationships)

Logic formulas (Horn Clauses)

Deduction of the clauses

Logic and control can be separated

(b) What are the deficiencies of Prolog?

Resolution order control

* - Ordering of pattern matching during resolution
* - Cut operator

Closed World Assumption

It has only the knowledge of the database

True or fail system rather than true or false

Negation problem

Prolog not operator is **not** equivalent to logical NOT operator

(c) What are the motivations for Logic programming?

Logic is used to represent program

Deductions are used as computation

A higher-level language does more automatically

* Can concentrate more on what is to be done and less on how to do it

Ideal: Algorithm = logic (what) + Control (how) – only specify logic and let system take care of control

**3.Use the set notation to describe resolution as a refutation system.**

Given a set of clauses S & and goal G,

negate G

\*{S} U {¬G}

\* existence of contradiction => derivation of empty clause

Based on {S} U {¬G} is only inconsistent if {S} U {G} is also consistent.

**4.** **(25%) Give deduction trees of resolution** (a) using (1) and (5); (b) using (2) and (5)

for the following set of clauses. Show each level of unification with instantiation (for example {m|Y}).

(1) anc (X, Y) ˅ ~par (X, Y)

(2) anc(X, Y) ˅ ~par (X, Z) ˅ ~anc (Z, Y)

(3) par (d, b)

(4) par (b, m)

(5) ~anc (X, m)

**Deduction trees images posted to website**

<http://athena.ecs.csus.edu/~jawaidt/prob4a.jpg>

<http://athena.ecs.csus.edu/~jawaidt/prob4b.jpg>

**Images won’t display in .pdf for some reason.**

**5. (20%) Conjunctions and Backtracking**.  Using the example of "Who teaches what" (see LogicProglecture page 19 in Canvas),

(a) try to trace through search process for Query 2;

(b) try to trace through Query 1, but with sub-goals reversed.

**Answer posted to website**

<http://athena.ecs.csus.edu/~jawaidt/prob5a.jpg>

<http://athena.ecs.csus.edu/~jawaidt/prob5b.jpg>

**6.**

**Answer posted to website as requested**

[http://athena.ecs.csus.edu/~jawaidt/](http://athena.ecs.csus.edu/~jawaidt/prob4a.jpg)

<http://athena.ecs.csus.edu/~jawaidt/prob6code.txt>

**Scroll to bottom**