Programming Languages

Spring 2019

Dr. Gurka Prolog Programs

Programs (2 + 1)

1. Dietary. Create a database that has facts about foods and meals. At a minimum, the database should include vegetables, fruits, desserts, snacks, proteins, starches, and drinks. Rules should specify what makes a valid meal (do rice and potatoes make a meal? is steak and beans and rice a meal? what meals can you make with corn?) and a valid snack (if it’s salty?). You may add more to the database for more interesting runs. Begin with ideas discussed in class.
2. Dating. Create a database that can be used to match people for dates. The facts will be details abut each individual person (likesanimals(john), middleaged(Sarah)), and the rules will specify what conditions make a match (match if they are in the same age group and like animals, or match if they work in the same profession, etc., you decide). Queries will do the matching (who should Bob date? what are all the database matches? should Jose date Sheila?), and could also answer questions like “who likes pets?” Begin with ideas discussed in class.
3. (extra / challenge) A scheduler for CS courses, including what professors teach what courses and what times classes meet. There will be more information to be added: see me for a short design session (required for this problem).

Project Specifications

1. Write two Prolog programs and run queries on them. The queries should be a thorough exercise of the database’s information and capabilities, including queries that produce no/null answers.
2. You may work entirely alone, or brainstorm together in class and then create you own solutions.
3. Hand in: cover letter, source code, and a short sample of some output from your programs (no more than one page).
4. Cover letter. The usual contents plus short answers to these questions:
   1. Who did you work with (if anyone) and how much?
   2. What development and execution environment(s) did you use?
   3. What was easiest about learning and using Prolog?
   4. What did you find interesting, powerful, quirky, etc., about the language and programming in it?
   5. If you did the extra credit, explain the overall logic of the program, any problems, and any fancy bits. Do not hand in extra credit that is not functioning correctly.
   6. Any other comments on the language (as a new user and/or student in CS 3210)?
5. Complete submissions must include both the Moodle post and the paper copy in class. You may submit a partial (non-extra-credit) project (Moodle and paper) for partial credit if that part is running correctly, and if you list the deficiencies in your cover letter.
6. Wednesday, May 8 (last day of class). Projects must have both a complete Moodle submission and a complete paper submission, and they must match. No late projects.

Prolog Project Design Worksheet – Dietary

(not collected)

facts

rules

queries

Prolog Project Design Worksheet – Dating

(not collected)

facts

rules

queries