CSc 144–002 — Discrete Mathematics for Computer Science I — Spring 2024 (McCann)
https://cs.arizona.edu/classes/cs144/spring24-002/

Homework #5

(50 points)

Due Date: March 15th, 2024, at the beginning of class

Directions

1. **This is an INDIVIDUAL assignment; do your own work! Submitting answers created by computers or by other people is NOT doing your own work.**
2. Start early! Getting help is much easier n days before the due date/time than it will be n hours before. Help is available from the class staff via piazza.com and our office hours.
3. Write complete answers to each of the following questions, in accordance with the given directions. Create your solutions as a PDF document such that each answer is clearly separated from neighboring answers, to help the TAs easily read them. Show your work, when appropriate, for possible partial credit.
4. When your PDF is ready to be turned in, do so on gradescope.com. Be sure to assign pages to problems after you upload your PDF. Need help? See "Submitting an Assignment" on https://help.gradescope.com/.
5. **Solutions submitted more than five minutes late will cost you a late day. Submissions more than 24 hours late are worth no points.**

Topic: Direct Proofs and Disproofs

1. (10 points) A perfect square is an integer that is the square of an integer. For example, 0, 1, 4, and 9 are the smallest perfect squares ($0^2 = 0, 1^2 = (-1)^2 = 1$, etc.).
   (a) For each of those four perfect squares, is the value that is two larger than it a perfect square?
   (b) Based on these examples, does it seem likely that all the rest of the values two greater than a perfect square are also not perfect squares?
   (c) Prove, using a direct proof, that if s is a perfect square greater than or equal to 4, then $s+2$ is not a perfect square. *HINT:* Argue that, for a perfect square s, the next perfect square larger than s (that is, $(x+1)^2$, assuming $s = x^2$) must be larger than $s + 2$.
2. (10 points) Using a direct proof with cases, prove this conjecture: If c is a perfect cube, then c is one less than a multiple of nine, one more than a multiple of nine, or exactly a multiple of nine. *HINT:* How close is every integer to a multiple of three?
3. (2 points) Disprove this conjecture, using a counter–example: If $x \mid yz$, then $x \mid y$ or $x \mid z$, where $x, y, z \in \mathbb{Z}$.

Topic: Sets

4. (4 points) Use set builder notation to describe each of these sets.
   (a) $\{\alpha, \beta, \gamma, \delta, \epsilon\}$
   (b) $\{0, 4, 8, 12, \ldots\}$
5. (8 points) Evaluate each of the following set expressions.
   (a) $\{a, \{b\}, \{c, d\}\} \subseteq \{\{a, b\}, \{b, c\}, \{c, d\}\}$
   (b) $\{a, i, e, u, o\} \subset \{u, i, a, o, e\}$
   (c) $\{x \mid x = 2y + 1, y \in \{0, 1, 2, 3, 4\}\} = \{1, 3, 5, 7\}$
   (d) $\mid \mathcal{P}(\{a, b, c, d, e, f, g, h, i, j\}) \mid$
   (e) $\mathcal{P}(\{0, 2, 4, 8\})$

(Continued on the back . . .)

Topic: Prolog

Background: On D2L is a video (in the Content – Assignments – Homework #5 area) that explains how to access our main instructional machine (`lectura.cs.arizona.edu`), and demonstrates how to run and use a version of Prolog (`gprolog`), including how to access lectura, create a text file containing a Prolog database, load a Prolog database into `gprolog`, and ask queries. We've also distributed a five–page tutorial document titled "Quick–'n'–Dirty Prolog Tutorial" that you should read.

The first thing you need to do is find the email you received when you started taking CS classes. It was sent from `it-admin@cs.arizona.edu` with the subject line of "Welcome of the Department of Computer Science." It contains information about your CS account. If you can't find that email, or changed your CS password at that time and forgot it, you can reset it by visiting `https://helpdesk.cs.arizona.edu` and clicking on the red link that reads "Click here to self manage your CS account/password." When you have your account ready to go, have read the tutorial, and have watched that video, you should be ready for the following questions. Start early so that you can get help if you need it!

6. (10 points) Our solar system has a star at its center (the Sun), some small rocky planets that orbit relatively close to it (Mercury, Venus, Earth, and Mars), and three moons (the Earth's Moon, and Mars' Deimos and Phobos).

   Create a Prolog database in a file named `planets.pl` on lectura that contains a set of facts, three rules, and a comment at the top that gives your name and NetID. The facts are all of the form `orbits(X,Y).` , where each fact states that body X orbits body Y. You'll need four facts to say that the four planets orbit the Sun, and three more to describe the orbits of the three moons. Remember, constants like planet names must be in lower case in Prolog. The three rules are: (1) `planet(P)` — A body P is a planet if it orbits the sun. (2) `moon(M)` — A body M is a moon if it orbits a planet. (3) `superorbits(X,Y)` — A body X superorbits a body Y when X orbits a body that is orbiting Y.

   When your database is created on lectura, load it into Prolog and use it to answer these four queries:

   (a) `orbits(earth,sun).` (In English: Does the Earth orbit the Sun?)

   (b) `planet(mars).` (In English: Is Mars a planet?)

   (c) `moon(mercury).` (In English: Is Mercury a moon?)

   (d) `superorbits(B,sun).` (In English: Which bodies are superorbiting the Sun?)

   (For this last query, remember to press the semicolon key ($\boxed{;}$) when you see an answer followed by a question mark, to allow Prolog to find and display all of the answers.)

   As your answer to this question in your PDF, include screenshot(s) that show: (a) The content of your database, and (b) the results gprolog produced when you ran the four queries.

7. (6 points) Consider the following predicates: $\forall x(T(x) \rightarrow A(x))$ and $\forall x(A(x) \rightarrow U(x))$, where $T(x) : x$ lives in Tucson, $A(x) : x$ lives in Arizona, and $U(x) : x$ lives in the US, with $x \in$ People. You are given that Phillipa lives in Tucson. Does Phillipa live in the US? We could prove it ourselves, but let's ask our logical servant Prolog do it for us!

   You'll need one fact (named `livesintucson`, stating that Phillipa lives in Tucson), and two rules (conversions of $\forall x(T(x) \rightarrow A(x))$ and $\forall x(A(x) \rightarrow U(x))$ into Prolog syntax). When this very small database is ready, use it to answer this question: Does Phillipa live in the US?

   As your answer to this question in your PDF, include screenshot(s) that show: (a) The fact and the rules in your database, and (b) the result gprolog produced when you ran the query to answer the question.