

Homework 8 — Various topics — assigned Wednesday 16 April, due Friday 2 May

Total number of points available on this homework is 200. Full credit is equivalent to 100 points.

Reading assignment

Read Chapters 4 and 5 of *Clause and Effect*.

Optional reading: Chapter 4 of *Programming in Prolog*.

8.1 [Prolog] Worksheets (20pts)

Complete Worksheets 20, 22, 23, and 24 from *Clause and Effect* (5pts each).

8.2 [Prolog] Essay (20pts)

Summarize the main points of the paper by Jacques Cohen, “Describing Prolog By Its Interpretation And Compilation”, Communications of the ACM, 28(12), 1311–1324, 1985. Use one page or less. Submit your paper summary as a separate PostScript file, `summary.ps`.

8.3 [Prolog] Relations (20pts)

8.3.1 Queries (16pts: 16 questions, 1pt each)

Given the following Prolog database:

```
a(b,c).
a(c,d).
a(d,e).
a(e,f).
h(X,Y) :- a(X,Z), a(Z,Y).
g(X,Y) :- a(X,Y).
g(X,Y) :- a(X,Z), g(Z,Y).
k(X,X).
k(X,Y) :- g(X,Y).
j(X,Y) :- a(X,Y).
j(X,Y) :- j(X,Z), j(Z,Y).
```

what are Prolog's answers to the following queries (explain in detail):

1. `?- a(b,c).`
2. `?- a(b,d).`
3. `?- a(X,Y).`
4. `?- h(b,e).`
5. `?- h(c,Y).`

6. `?- h(X,Y).`
7. `?- g(b,c).`
8. `?- g(e,Y).`
9. `?- g(X,d).`
10. `?- g(X,b).`
11. `?- k(b,c).`
12. `?- k(e,Y).`
13. `?- k(X,d).`
14. `?- k(X,b).`
15. `?- j(b,c).`
16. `?- j(e,Y).`

Hint: It may be helpful to draw a diagram (a graph) showing the atoms and the relations that hold between them; it may be helpful to answer the question below first.

8.3.2 Meaning (4pts)

Describe, using standard mathematical terms, *both symbolically and in English prose*, the relations *h*, *g*, and *k* in terms of the relation *a*, where the relations are as defined by the Prolog code above.

8.4 [Prolog] Lists (20pts)

Write a predicate `filter/3` such that `filter(L1,L2,L3)`, where *L1*, *L2*, and *L3* are lists, means that *L3* consists of those elements of *L1* that are not elements of *L2*, in the same order as they are in *L1*.

For example:

```
?- filter([1,2,3,4,5,6,7,8,9],[3,6],X).
X = [1, 2, 4, 5, 7, 8, 9] ;
No
```

Hint: Use the predefined predicate `member/2`. Here is the help entry:

```
member(Elem, List)
    Succeeds when Elem can be unified with one of the members of List.
    The predicate can be used with any instantiation pattern.
```

It is OK for `filter` to work only one way, from *L1* and *L2* to *L3*.

8.5 [Prolog] Context-sensitive grammar (30pts)

Consider a language L over the alphabet $\{a, b, c\}$, $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. It is generated by the following grammar:

$$G = (\{S, X, Y\}, \{a, b, c\}, S, F),$$

where the productions are:

$$F = \{S \rightarrow abc, S \rightarrow aXbc, Xb \rightarrow bX, Xc \rightarrow Ybcc, bY \rightarrow Yb, aY \rightarrow aaX, aY \rightarrow aa\}$$

This grammar is not context-free, because on the left-hand sides of some productions there are strings longer than single non-terminals. (In fact, there is no context-free grammar for this language.) However, the productions in this grammar satisfy a *length-increasing* property: the number of symbols on the right is never less than the number on the left. Such grammars are called context-sensitive.¹

Write a Prolog program that determines whether a string belongs to the language L . Strings over the alphabet $\{a, b, c\}$ will be represented as Prolog lists of Prolog atoms `a`, `b`, and `c`. You should split the program into a set of predicates that are general-purpose (for any context-sensitive grammar), and a set of predicates that describe the grammar G .

The program should result in queries like these:

```
?- accepts([ ]).
No
?- accepts([a]).
No
?- accepts([a,b]).
No
?- accepts([a,b,b]).
No
?- accepts([a,b,c]).
Yes
?- accepts([a,a,b,b,c,c]).
Yes
?- accepts([a,a,a,a,a,a,b,b,b,b,b,b,c,c,c,c,c,c]).
Yes
?- accepts([a,a,a,a,a,a,b,b,b,b,b,b,c,c,c,c,c,c,a]).
No
```

You may find useful the predefined predicate `length/2`, which works as follows: given a list `X`, the goal `length(X, N)` succeeds and sets `N` to the integer equal to the length of the list `X`.

You should test your program on the queries above and others you find useful to cover the various cases of strings in the language and outside the language. Make sure that your program does not go into an infinite loop on any inputs. Do not submit your program unless it has passed all these tests.

8.6 ★ [ML] Ramanujan (20pts)

Use (and extend) your code from Exercise 5.4 to evaluate $e^{\pi\sqrt{163}}$ to at least 100 significant digits.

¹Note that there are grammars, and their languages, that are not even context-sensitive.

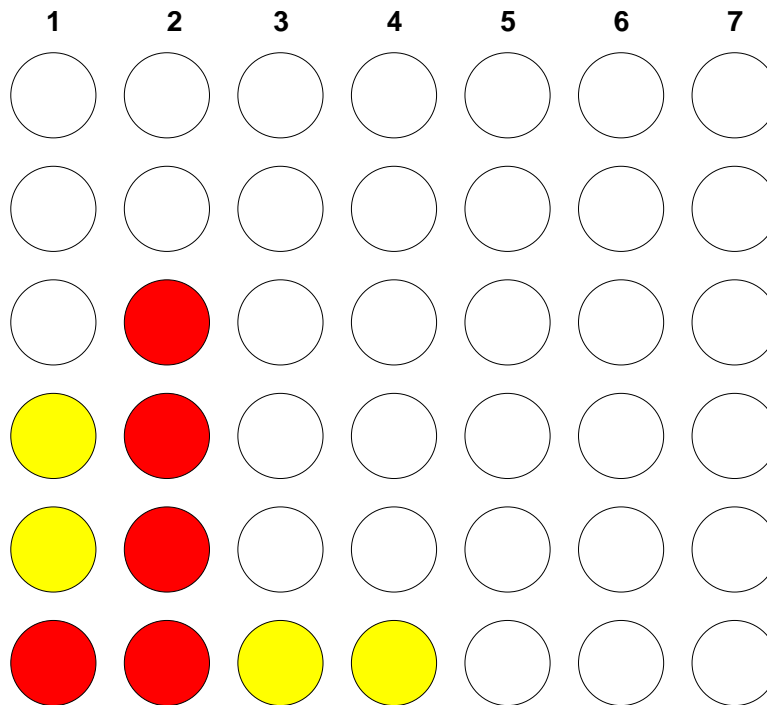
8.7 [Puzzle] Tic-tac-toe (10pts)

Revisit Exercise 3.4 using the long integer arithmetic developed in Exercise 4.6 (or 5.4.2) to compute the numbers of strategies exactly.

8.8 ★★ [Puzzle] Connect Four (60pts)

The game of *Connect Four* is played as follows (see Figure 1)): two players, red and yellow, alternate moves. Red goes first. A move consists of dropping a colored token into one of seven slots. Each slot feeds a vertical column that has room for six tokens. The seven columns are arranged side by side to form a 7×6 grid. The first player who achieves a line of four tokens, horizontally, vertically, or diagonally, wins, and the game stops immediately.

Redo Exercises 2.5, 3.4, and 4.7 for the game of Connect Four.



moves: R1, Y1, R2, Y3, R2, Y4, R2, Y1, R2; R wins

Figure 1: A game of *Connect Four*.

How to turn in

Turn in your code by running

```
~roshan/handin your-file
```

on a regular UNM CS machine.

You should use whatever filename is appropriate in place of your-file. You can put multiple files on the command line, or even directories. Directories will have their entire contents handed in, so please be sure to clean out any cruft.