

CS344: Programming Languages

Homework 11: Prolog

For both of the following problems, you are required to submit a sample prolog session that demonstrates each predicate working correctly, with comments (hand written or typed) added to describe what is being tested and what you are doing.

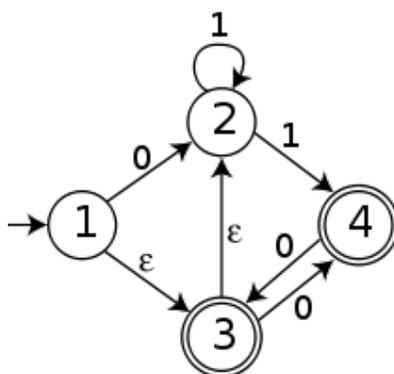
As a suggestion, you can use the Unix command script to store this easily as follows:

```
script sfilename           // a copy of your session will be saved
bash$ swill                // script may change the unix prompt
?-                          // Do your PROLOG thing.
?- ...
?- ^D                      // exit PROLOG
bash$ ^D                   // control-D exits from script
Script done. File is sfilename.
$!pr sfilename             // print your script and annotate it
                           // (or use pico and add comments that way)
```

Required Problems

- As mentioned in class, Prolog is ideal for simulating NFA's as well as DFA's, since the language will backtrack and explore alternate possible valid parsings if some parsing fails.

Code the specification for the following NFA in prolog as NFA.pl, and test your code by showing the derivations for several strings both in the language and not in the language. Show at least one string which has several possible parsings (although only one may end up in an accept state).



- We're going to return to the idea of storing a set in a list structure, but in Prolog instead of Haskell. Consider a set to be a list of integers (so that we have a comparison operator) where elements are stored in the list in sorted order, and no repetition is allowed. Given this representation, define the following predicates in Prolog:

(a) `member(X,L)`, which holds iff the element `X` occurs in `L`.

- (b) $\text{subset}(L,K)$, which holds iff L is a subset of K .
 - (c) $\text{disjoint}(L,K)$, which holds iff L and K are disjoint (i.e. they have no elements in common).
 - (d) $\text{union}(L,K,M)$, which holds iff M is the union of L and K .
 - (e) $\text{intersection}(L,K,M)$, which holds iff M is the intersection of L and K .
 - (f) $\text{difference}(L,K,M)$, which holds iff M is the difference of L and K .
3. In class, we did an implementation of map coloring (based on example 2.9 from http://www.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html). I noted that this wasn't particularly efficient, since it waits until after everything is assigned a color before looking for conflicts in the map. Modify that program so that it checks for conflicts while coloring a region (before moving on to color the next region).

For full credit, show a transcript where you implement this and get it to work on several maps, including getting at least one coloring for the map of the US which was provided. (Note: please don't crash Turing while you do so!)