Jacob Alspaw
jaa134
4/24/15
EECS 302

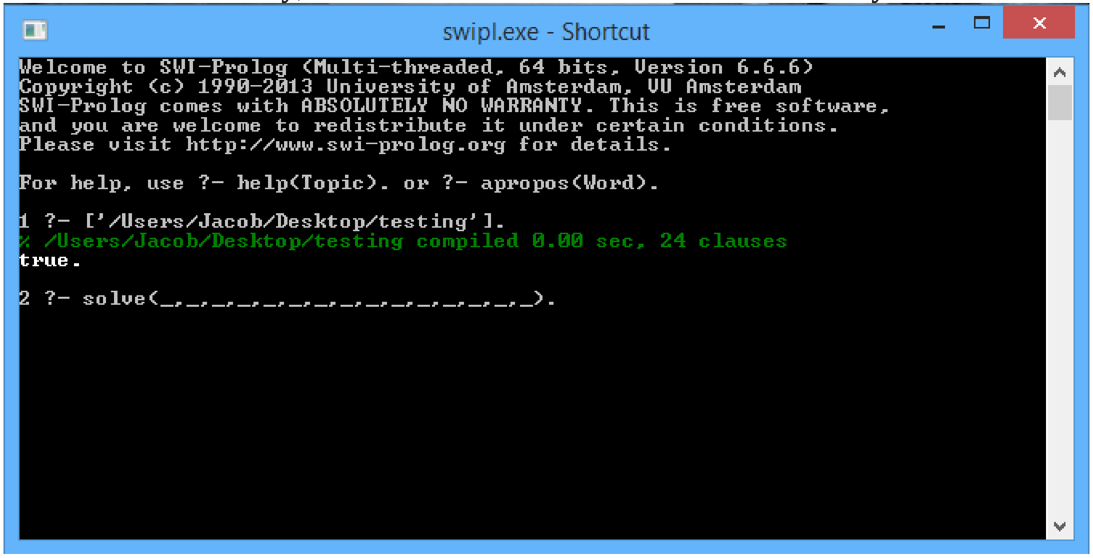
Hello Professor,
After one very long night, my project works. It was quite simple to implement after I got rolling. Prolog is a very simple language and VERY useful. Thanks for pushing me to learn it!

My project is split into sections. The first denotes objects: desserts, entrées, and soups. The second section is rules dictated in your email. The third was the mixture of these rules to see what combinations of objects would work for an individual day. The final section is the rule that solves the system. It first assures all inputs are different (i.e. no reusing variables in the same function call). Then it sees if these variables will work for each day. If they do, then the function returns true.

Prolog is very remarkable. The solve method uses place-holders or don't-cares to return a solution. In the scenario you assigned me, the "solve" method has a combination total of 15^15 possibilities. This number equates to about 4x10^17 combinations.
And Prolog returns a Boolean within a mere second. Absolutely brilliant.

To test my program:
1: Compile the program: ['<location path>'].
2: Call the method: solve(_,_,_,_,_,_,_,_,_,_,_,_,_,_,_).
3: Observe the results.

Notes
It is important to finish each statement with a period. The underscores in the solve method are needed. There should be exactly fifteen, each separated by commas. The method, when instantiated correctly, will return the correct results for each day.



```
Welcome to SWI-Prolog (Multi-threaded, 64 bits, Version 6.6.6)
Copyright (c) 1990-2013 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?- ['/Users/Jacob/Desktop/testing'].
% /Users/Jacob/Desktop/testing compiled 0.00 sec, 24 clauses
true.

2 ?- solve(_,_,_,_,_,_,_,_,_,_,_,_,_,_,_).
```