Topics/Examples

- \(+X=Y \) and \(+a=b\). (when you have uninstantiated variables the result is false -- major problem with HW3 (that could have been caught with testing) ).
- Most of you figured out that you can use parentheses.
- ; (semicolon) in a query or rule means logical or and since it has lower precedence than and you can use it to create predicates with only one rule where more than one was previously required -- this is unadvisable since it’s harder to read. Use separate rules for separate cases (base case, recursive case, etc.)
- \(- \) is a little more handy than \(+ \) when comparing two things, e.g. in \(X=-Y\) vs. \(+X=Y\), (but you need \(- \) to negate a predicate).

reverse list [LPN] (with accumulator… not so impressive speed-wise -- list operations and structures are highly optimized in Prolog. The main thing is that there are a fair number of Prolog patterns that we don’t have time to cover). source.

HW2… if you can do the rest you can do this ( a query is just the part after the predicate part of a rule).

HW3 ([COSI]) -- have other concise examples in link below
file (remember, this started with image of solar system and we built the KB).

HW4 -- I elected to use a different example from the HWs… map coloring is straightforward, cryptarithmetic has a straightforward pattern (the order is what matters -- generating and testing if the order is wrong can be very slow), and the logic problems you have to take one at a time. This one is from https://www.brainbashers.com/showpuzzles.asp?puzzle=ZYOT code(brainbash.pl)

HW5 -- Some basic list examples: lists.pl

Lists Example1: (source)

Plan Example: (depth-first search (really iterative deepening w/ bplan) ) (source)

[COSI] = https://cosidesk.wikispaces.com/sample+knowledge+bases (one mistake in that was that they didn’t say neptune orbits the sun)