

Homework 5 (prolog) – MATH 2510 Spring 2018

Consider the prolog code *arithmetic.pl* at

<https://github.com/tomcuchta/math2510spring2018/blob/master/arithmetic.pl>

Copy this code to SWISH.

What appears is an implementation of a fragment of “first order arithmetic”. Lines 5-7 define what a “numeral” is — a numeral is 0 or a combination of “successors” of 0 in the following sense:

Number	Numeral
0	0
1	<code>succ(0)</code>
2	<code>succ(succ(0))</code>
3	<code>succ(succ(succ(0)))</code>
4	<code>succ(succ(succ(succ(0))))</code>
⋮	⋮

Line 11 is a 3-term relation encoding $0 + Y = Y$ (first two arguments are the things you add and the third is the sum). Lines 15-16 define general addition in the sense that `add(X,Y,Z)` means $X + Y = Z$. It defines it “recursively”, in the sense that it says “ $(X + 1) + Y = Z + 1$ means $X + Y = Z$ ”. This is enough to define addition in general, as you will see. (We will write formal proofs concerning this later on!)

Also defined is multiplication. Line 20 encodes the idea that $X \cdot 0 = 0$ (the underscore in front of X is there for technical reasons – prolog complains if a variable only appears once but that makes it not complain). Then lines 24 and 25 encode the idea that $X(Y + 1) = Z$ means $X \cdot Y + X = Z$. The code `mult(X,Y,P), add(P,X,Z)` is saying “ $X \cdot Y = P$ and $P + X = Z$ ”.

1. Run the query `add(succ(0),succ(succ(0)),X)`. What does Prolog tell you? What addition problem did it compute?
2. What query do you run to ask prolog to solve the equation $3 + X = 4$?
3. Run the query `mult(succ(succ(0)),succ(succ(0)),X)`. What is the result? What arithmetic problem is being computed?
4. What query do you run to solve $1 + 3X = 4$? What is the result when you run this query?
5. Exponentiation can also be defined recursively. This problem has you implementing exponentiation in prolog code like addition and multiplication were above. Implement exponentiation as `expo` using the base fact that $X^0 = 1$ and the fact that $X^{n+1} = Z$ means that $X^n \cdot X = Z$.
6. What query do you run to calculate 2^3 ? What is the result when it is run?
7. Recall that the factorial $n!$ is defined recursively. Use $0! = 1$ and $(n + 1)! = (n + 1) \cdot n!$ to define it as `fac` in prolog.
8. Calculate $3!$ in prolog. What is the result?