

# AI Project #4: Natural Numbers

Steven Wirsz<sup>1</sup>

<sup>1</sup>California State University, Northridge

Due Date:

**February xx, 2018 23:59 hours**

Use numerals to represent in Prolog the natural numbers. The constant `z` (for zero) is a numeral, and if `X` is a numeral then `s(X)` is also a numeral (where the function `s` represents `++`). The numerals corresponding to 0, 1, 2, 3, etc. are `z`, `s(z)`, `s(s(z))`, and `s(s(s(z)))`, etc. Add code comments to show your understanding and demonstrate with test cases.

Define a number of predicates to interact with these numerals. For example: `plus(z,Y,Y)`.

`plus(s(X),Y,V) :-`

`plus(X,Y,U), % remove a function from X till reaching zero`

`s(U) = V. % add a function to V each time`

REQUIREMENTS:

1. Create an implementation of natural numbers
2. Implement `plus`
3. Implement `equal`
4. Implement `less than`
5. Implement `greater than`
6. Implement `minus`
7. Implement `multiplication`
8. Implement `mod/remainder`
9. Implement `factorial`
10. Extra credit: implement `number` so that `number(X, N)` is true if `X` is a numeral corresponding to the decimal integer `N`. For example, `number(s(z),1)` is true and `number(s(s(s(z))),2)` is false.
11. Extra credit: implement `e_number`, so that `e_number(X, Y)` is true if `Y` is a phrase in English for positive natural numbers between zero and 100, representing `X`. For example:

`e_number(s(s(s(z))), three)` returns true

`e_number(X, twenty five)` returns `X = s(s(s(...(z)...))`

**Submit a working .pl file that demonstrates the cases above with a significant number of code comments to explain the operation of each predicate and also provide test cases.**

Last Revised: January 21, 2018