Problems

1. As discussed in class Prolog implements a remainder operation for integer division:

\[ +\text{Expr1} \text{ rem } +\text{Expr2} \]

That is, \( 3 \text{ rem } 2 = 1 \) and \( 4 \text{ rem } 2 = 0 \), etc. Use this operation to implement the two operators,

\[ B \rightarrow \text{even}(A) \mid \text{odd}(A) \]

in the semantic specification given in class where \( A \) is the nonterminal for the arithmetic expressions and \( B \) the nonterminal for the boolean expressions. Show that your specification works by computing the semantic value of some example expressions.

2. Show that the expressions \( \text{odd}(a) \) and \( \text{even}(\text{add}(a, 1)) \) are semantically equivalent for all \( a \in A_{\text{exp}} \) using Prolog. Write a proof score and show that your proof score works.

3. Construct an inductive proof in Prolog that demonstrates that all computations on boolean expressions produce a value of either true or false.

4. Implement the semantics of the do - while loop from Assignment #1 in the Prolog specification using the syntax,

\[ C \rightarrow \text{dowhile}(C, B) \]

Here \( C \) are commands and \( B \) are boolean expressions. The informal specification of this command is that \( c \) is executed as long as the boolean expression remains true. Extend the specification given in class and demonstrate that your semantics work with some examples.

Hand in hardcopies of your language specification and all relevant proof scores and runs.