NAME: SANKET SANJAYPANT HIRE
ENTRY NO: 2016CS50402

Theorem: *Compiler + Stack machine calculates a bigint whose value is the same as what eval computes.*
Proof: *By Induction on the height of exptree.*
Base Case: *The base case is, ht(exptree t) = 0,*
i.e. Expression e = N(m),
where N(m) is a numeral of bigint m.
Suppose (eval e), (compile e) and (stackmc (bigint list = []) (opcode list))
where opcode list is (compile e).
Then,

(A) *eval e ⇒ eval (N(m)) ⇒ m* ***(By definition of eval)***

Now,

The post order opcode list is formed by compile.

(B) *compile e ⇒ compile (N(m)) ⇒ [CONST(m)] (By definition of compile)*

Now,

The opcode list has one element. Since its CONST operation on bigint m, the stack(bigint list) is pushed by bigint i.

(C) *stackmc [] (compile e) ⇒ stackmc [] ([CONST(m)])*

After completion of this operation, stack machine returns to the top of stack i.e. m.

From (A) and (C), eval e = stackmc [] (compile e).
Induction Hypothesis: *Suppose we have shown that for all t': exptree such that ht(t') ≤ k:*

eval t' = stackmc [] (compile t')

Induction Case: *Let ht(t) = k+1, i.e*
eval t will perform eval on each node of exptree from leaf to root exptree considering both LHS(t1) and RHS(t2) of the node of exptree as an operation. i.e eval (exptree of eval t1 and eval t2), the final result will be result of recursion stack of eval.

eval (t) = eval (operation eval(t1) eval(t2))

Now, Considering the opcode list as a post-order tree,

CONST of bigint will be the leaf nodes.
ABS and UNARYMINUS will be the nodes attached to the leaf nodes.
And remaining opcodes (PLUS, TIMES, MINUS, DIV and REM) will be postorder-tree(t) with postorder-tree t1 and postorder-tree t2 as its children. Each opcode node is calculating values according to its opcodes and prepending to the empty bigint list. Since this operation will perform accumulation of the results from leaf to root of the postorder tree, the final result of opcode list calculation will be at the first entry of bigint list(or top of the stack)

postorder (t) = postorder (operation postorder(t1) postorder(t2))

Since evaltree and postorder-tree are following the same order of operations Hence, Compiler + Stack machine calculates a bigint whose value is the same as what eval computes.