MODULE *ExponentiationOperator*

EXTENDS *Integers*, *Sequences*, *TLC*

CONSTANT *valuesForM*, *valuesForN*
ASSUME $\land$ *valuesForM* $\subseteq$ *Int*
$\land$ *valuesForN* $\subseteq$ *Int*
$\land \forall\, n \in$ *valuesForN* $: n \geq 0$

```
--algorithm exponential{

 variables
     m ∈ valuesForM,
     n  ∈ valuesForN,
     result = 1,
     i = 1 ;

 {
     while ( i ≤ n ) {
         result := m * result ;
         i := i + 1 ;
      } ;

     assert result = m^n ;
  }

}
```

BEGIN TRANSLATION (*chksum*(*pcal*) = "*d*01051*e*0" $\land$ *chksum*(*tla*) = "2*d*8*cab*5*c*")
VARIABLES *m*, *n*, *result*, *i*, *pc*

$vars \triangleq \langle m,\ n,\ result,\ i,\ pc \rangle$

$Init \triangleq$ Global variables
$\land\ m \in valuesForM$
$\land\ n\ \in valuesForN$
$\land\ result = 1$
$\land\ i = 1$
$\land\ pc =$ "Lbl_1"

$Lbl\_1 \triangleq$ $\land\ pc =$ "Lbl_1"
$\land$ IF $i \leq n$
THEN $\land\ result' = m * result$
$\land\ i' = i + 1$
$\land\ pc' =$ "Lbl_1"
ELSE $\land\ Assert(result = m^n,$
"Failure of assertion at line 24, column 9.")
$\land\ pc' =$ "Done"
$\land$ UNCHANGED $\langle result,\ i \rangle$

$\land$ UNCHANGED $\langle m, n \rangle$

Allow infinite stuttering to prevent deadlock on termination.

$Terminating \triangleq pc = \text{“Done”} \land \text{UNCHANGED } vars$

$Next \triangleq Lbl\_1$
$\quad \lor Terminating$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION

\ * Modification History
\ * Last modified *Tue Mar* 12 09:06:39 *CET* 2024 by *jeujeus*
\ * Created *Tue Mar* 12 09:02:27 *CET* 2024 by *jeujeus*