The (Abacus-)Uncomputability of the Abacus Machine Halting Function

Let AM be the set of all Abacus machines.

A. Show that AM is enumerable

Given that AM is enumerable, let us consider some enumeration E = AM_1, AM_2, … of Abacus machines, and define, relative to such an enumeration E, the Abacus Machine Halting Function AH_E: N x N $\rightarrow$ N as follows:

> AH_E(m,n) = 1 if Abacus Machine AM_m, when started with n stones in register 1, and all other registers being empty, halts.
>
> AH_E(m,n) = 0 if Abacus Machine AM_m, when started with n stones in register 1, and all other registers being empty, does not halt.

Let us say that some function f:$N^k \rightarrow$N is Abacus-computable* if and only if there exists some Abacus-machine A such that for all <n_1, …, n_k>: if f(n_1, …, n_k) = n, then A, when started with n_1 stones in register 1, n_2 stones in register 2, …, n_k stones in register k, and all other registers being empty, will halt with n_1 stones in register 1, n_2 stones in register 2, …, n_k stones in register k, n stones in register k+1, and all other registers being empty.

B. Prove that given any enumeration E, the Abacus Machine Halting Function AH_E is not Abacus-computable*

Since AH_E is not Abacus-computable*, it is very likely that AH_E is not Abacus-computable through any 'effective' way of using Abacus machines to compute functions from natural numbers to natural numbers.

C. Relative to some enumeration E = TM_1, TM_2, … of Turing-machines, define the Turing Machine Halting Function TH_E: N x N $\rightarrow$ N as follows:

> TH_E(m,n) = 1 if Turing Machine TM_m, when started on the leftmost 1 of a block of n consecutive 1's on an otherwise blank tape, halts.
>
> TH_E(m,n) = 0 if Turing Machine TM_m, when started with n stones in register 1, and all other registers being empty, does not halt.

Without making any appeal to Turing's Thesis, explain why TH_E is not Abacus-computable*