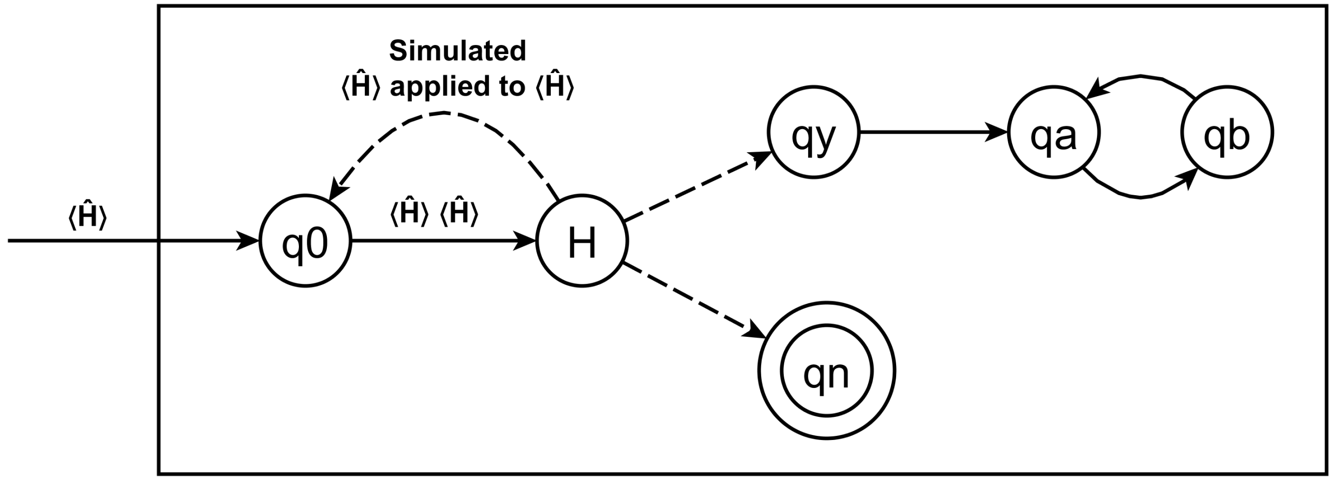
(q0) is prepended to H to copy the ⟨M⟩ input of Ĥ. The transition from (qa) to (qb) is the conventional infinite loop appended to the (qy) accept state of embedded_H. **⊢*** indicates an arbitrary number of moves.

Ĥ.q0 ⟨M⟩ ⊢* embedded_H ⟨M⟩ ⟨M⟩ ⊢* Ĥ.qy ∞ // see diagram below for (qa) and (qb)
Ĥ.q0 ⟨M⟩ ⊢* embedded_H ⟨M⟩ ⟨M⟩ ⊢* Ĥ.qn

Analysis of Linz Halting Problem Proof



When Ĥ is applied to ⟨Ĥ⟩
Ĥ.q0 ⟨Ĥ⟩ ⊢* embedded_H ⟨Ĥ⟩ ⟨Ĥ⟩ ⊢* Ĥ.qy ∞
Ĥ.q0 ⟨Ĥ⟩ ⊢* embedded_H ⟨Ĥ⟩ ⟨Ĥ⟩ ⊢* Ĥ.qn

computation that halts… “the Turing machine will halt whenever it enters a final state” (Linz:1990:234)

Non-halting behavior patterns can be matched in N steps
⟨Ĥ⟩ Halting is reaching its simulated final state of ⟨Ĥ.qn⟩ in a finite number of steps

N steps of ⟨Ĥ⟩ correctly simulated by embedded_H are the actual behavior of this input:
(a) Ĥ.q0 The input ⟨Ĥ⟩ is copied then transitions to embedded_H
(b) embedded_H is applied to ⟨Ĥ⟩ ⟨Ĥ⟩ (input and copy) which simulates ⟨Ĥ⟩ applied to ⟨Ĥ⟩
(c) **which begins at its own simulated ⟨Ĥ.q0⟩ to repeat the process**

Simulation invariant: ⟨Ĥ⟩ correctly simulated by embedded_H never reaches its own simulated final state of ⟨Ĥ.qn⟩.

Therefore when embedded_H aborts the simulation of its input and transitions to its own final state of Ĥ.qn it is merely reporting this verified fact.

Linz, Peter 1990. An Introduction to Formal Languages and Automata. Lexington/Toronto: D. C. Heath and Company. (317-320)