



解決「停機問題」(Halting Problem)的演算法，若存在，演算法為何？若不存在，證明你的論述。

解決停機問題的演算法並不存在，我們可採「矛盾證法」(proof by contradiction)證明之。假設停機問題有解，存在演算法  $H(P, I)$  可以判斷任何程式  $P$  在任何輸入  $I$  的情況下可否停機(終將結束，或將無止境地運行下去)。若  $P$  在輸入  $I$  時可停機，則  $H$  輸出“yes”(停機)，否則輸出“no”(不會停)。程式本身也是一種資料，因此它可作為輸入，故  $H$  可判定當以  $P$  作為  $P$  自己的輸入時， $P$  是否停機。此時，我們建構另一個演算法  $U(P)$  如下：

•  $U(P)$  呼叫  $H(P, P)$ ：

✓ 如果  $H(P, P)$  輸出“no”，則  $U$  立即停止運算

✓ 如果  $H(P, P)$  輸出“yes”，則  $U$  進入無窮迴圈

換言之， $U(P)$  做出與  $H(P, P)$  的輸出相反的動作。

接著考慮  $U(U)$ ，亦即讓  $U$  演算法以自身  $U$  作為輸入，檢視執行結果。因  $U(U)$  係由  $H$  演算法建構而成，底下讓我們來檢視  $H(U, U)$  的輸出：

- 若  $H(U, U)$  輸出“yes”(停機)，則  $U$  進入無窮迴圈，然依  $H$  的定義， $H(U, U)$  輸出“yes”，代表  $H$  判定  $U$  會停機，可是實際上  $U$  卻不會停，形成矛盾，意味著  $H$  無法給出正確答案。
- 若  $H(U, U)$  輸出“no”，則  $U$  停止，仍然矛盾，因  $H(U, U)$  判定  $U$  不會停，可是實際上  $U$  卻會停機，再次意味著  $H$  無法給出正確答案。

綜上討論可知， $H$  雖被宣稱存在卻無法給出正確答案，故  $H$  存在的假設不成立，因此不存在解決停機問題的方法。

註：矛盾證法的邏輯為：若  $p$  則  $q$   $\equiv$  非  $q$  則非  $p$ ，二種主張完全等義

例：若下雨( $p$ )，則地濕( $q$ )  $\equiv$  若地未濕(非  $q$ )，則未下雨(非  $p$ )

證明停機問題的解方是否存在，我們操作矛盾證法的邏輯。為了證明解方不存在，我們故意逆向思考，假設停機問題有解，可由演算法  $H$  解決。基於如此假設，之後經過系列步驟的推導，最後得到結論： $H$  無法正確解答停機問題，顯然與先前的假設不符，矛盾的結果源於一開始的假設有誤，藉此推翻原假設，從而說明假設不成立。