

Computer can't do everything

The claim is that the following Pascal/C procedure cannot exist:

Procedure H(LIST,ARG)

Specification of what H(LIST,ARG) is to do:

If LIST is the listing of a procedure P, then the procedure H does the following:
 If procedure P is run with input ARG and stops
 then H outputs YES and stops
 otherwise it outputs NO and stops.

Proof by contradiction: Assume a procedure H fulfilling the above specification exists. Then it is easy to construct the following new procedure with the specification given:

Procedure NH(LIST)

Specification:

If the execution of H(LIST,LIST) outputs YES
 then NH(LIST) runs into an infinite loop.
 If the execution of H(LIST,LIST) outputs NO
 then NH(LIST) stops.

How is NH(LIST) constructed? Simply use the code for H except that

- each occurrence of ARG in the code of H is replaced by LIST,
- each statement in the code that outputs YES is replaced by a statement that causes an infinite loop,
- and each statement that outputs NO is simply deleted causing the machine to stop without outputting anything.

Clearly if a procedure H exists following its specification then NH follows its specification.

Now consider the execution H(<NH>,<NH>), where <NH> is a listing of the procedure NH. From the specification of H we know that the execution H(<NH>,<NH>) either outputs YES or NO.

Case H(<NH>,<NH>) outputs YES: Then by the specification/construction of NH, the execution NH(<NH>) loops. Thus by the specification of H, the execution H(<NH>,<NH>) outputs NO. Contradiction!

Case H(<NH>,<NH>) outputs NO: Then by the specification/construction of NH, the execution NH(<NH>) stops. Thus by the specification of H, the execution H(<NH>,<NH>) outputs YES. Contradiction again!

Since both cases lead to a contradiction we are done and the procedure H with the above specification cannot exist.