

Re: Penrose's Computing Pi Description?

Source: <http://sci.tech-archive.net/Archive/sci.logic/2005-06/msg00189.html>

- *From:* "H. J. Sander Bruggink" <sanderbruggink@xxxxxxxxxxx>
 - *Date:* Fri, 10 Jun 2005 14:06:51 +0200
-

Stephen Harris wrote:

"H. J. Sander Bruggink" wrote

In the standard definition of a TM, it can only output things after it has finished. The "output" of a non-terminating TM is usually taken to be undefined.

Is that because it is assumed that the input is uncomputable?

What do you mean by an input being uncomputable? I have heard about functions being uncomputable, but not about an input being uncomputable.

The way a TM outputs things, is by leaving the output on the tape. However, the tape is also where the TM stores intermediate results. Therefore, in principle, while the TM is running there is no way (in general) to distinguish the output from the intermediate results.

Of course, for some particular TM, you could prove that some symbol on the tape will never change again. But in general, that is not true.

How about for Pi? How about for the infinite many of irrationals produced by taking the square root of a prime?

What do you mean with "what about Pi"? We are talking about TMs and not about numbers, right?

Re: Penrose's Computing Pi Description?

> Is the algorithmic method which Penrose

states can compute Pi (one digit at a time and then appending)
is the algorithm the computation of one particular digit or is
the algorithm considered to include the execution of the
continuing series of computations.

I think that what Penrose means, is that an algorithm **must**
stop in finite time. So an algorithm to compute Pi does not
exist, because one needs infinitely many steps to compute it.
However, an algorithm can be constructed which computes Pi to
an arbitrary number of digits, and mathematicians often deal
with infinity that way.

However, an algorithm exists which can compute the nth decimal
of Pi for each value of n. So it is one algorithm which is
used time after time.

Has the algorithm halted after finding one digit of Pi,
or is the algorithm considered the entire process of
finitely, step by step, churning out an endless sequence
of digits?

Yes, the algorithm has halted after finding one digit of Pi.

Are there two algorithms? Or one algorithm
which includes writing finite values along the way. Or
just one algorithm which writes a finite value and this
algorithm is repeated over and over. Must an identical
algorithm compute the same place in the decimal
expansion of Pi.

What is an identical algorithm? When are two algorithms the
same? Is it when they have the same representation in some
fixed (programming) language, is it when they compute the
same function, or something in between?

It seems to me that appending the
last digit computed to the other digits already computed
must be part of the algorithm and this part of the process
doesn't halt so that the "entire" algorithm doesn't halt?

Re: Penrose's Computing Pi Description?

I hold the broader picture of what the algorithm includes,
but I could be wrong.

Usually, an algorithm is required to halt in order to have
output. This does correspond to my intuition about what an
algorithm should be.

groente
-- Sander
.