Letter to the referees of EWD418.

(I refer to the referee who started with "I think this..." as Referee A, to the one startiong with "OPINION This paper contains..." as Referee B.)

In response to Referee A's remark (2): it was not my intention to introduce a terminological innovation when using the word "calculus". On the contrary! I wanted to be in step with common usage --which I felt obliged to explain, because many people are confused about it-- . I shall change the last sentence of the introduction, as that seems to be the source of the erroneous impression. Referee B suggested that the word "calculus" should be removed from the title because "nowhere a calculus for the derivation of programs is defined". Instead of defining, I thought --and still think-- that I have "shown" it, at least in my sense of the word "calculus" --and in very much the same sense in which D.E.Knuth uses the term (between quotes!) on page 283 of his recent article in the December 1974 issue of the ACM Computing Surveys--. In order not to arouse false hopes I shall, however, remove the word "calculus" from the title. (Besides that, I prefer underselling above overselling.)

Referee A asks (3) "Why have the notes on page 2 as notes?" Simply because I do not propose to allow empty guarded command sets; they are conclusions, irrelevant for the rest of the story, that have been included as redundant material so that the reader can check his understanding of the two constructs.

Referee A's suggestion (4) will be followed: as the note on page 5 stands, it contains indeed too many negations. I shall also follow his suggestion (5) --to combine the hypotheses of Theorem 3-- and ::although with some reluctance-- suggestion (6) "either don't mention personal feeling [...] or elaborate the reasons". I know that his suggestion is in full accordance with the scientific tradition (and that tradition is not without wisdom, as it eliminates, for instance, unsubstantiated slander!) But we loose by it too, for even scientists have, besides a head, a heart as well and there is no point in denying the latter one its right of existence. (More than once I have given advice or formulated an expectation while, when asked "to elaborate the reasons", I found myself unable to do so: I just felt it in my bones.) In this case elaboration would fall outside the scope of the article and the mentioning of my feelings will be removed.

The note on page 6 --referee A's remark (10)-- is hard to understand: the relation between wp and wdec is, although formally simple, conceptually hard to understand. I have to leave it as it stands, because it says as well as I can manage what I want to say and from Referee A's remark I cannot deduce a suggestion for improvement. (I don't see how his "Q(X)" suddenly enters the picture.) May I ask him to try to read the note on page 6 another time?

Referee A's remark that the four properties of wp are derivable was a remark that I expected. The point, however, is this:

either one postulates that our wp's describe an input/output relation and then they follow from more fundamental axioms of relations

or one introduces the wp's as a formal game and proves -by structural induction-- that our wp's enjoy those properties (which are then used for proving formally assertions about composite wp's). That the wp's can be interpreted as describing input/output relations then follows.

My preference is for the second position, but --intentionally, because it is not a key issue in this paper-- I have used the non-committing term "properties", and have left the option open whether "by definition" refers to the definition of the notion "weakest pre-condition" --for those readers who prefer the first position-- or to the definition of the specific wp's I have introduced. I shall mention, however, the issue of non-determinacy versus determinacy in relation to the 4th Property.

(As a matter of fact, the 2nd property is a special case of a more general property --call it "continuity" if you prefer--

If for $k \geq 0$ we have $C_k \Rightarrow C_{k+1}$, then we have for any S

$$wp(S, (\underline{E}\ k:.\ k \geq 0: C_k)) \Rightarrow (\underline{E}\ r:\ r \geq 0: wp(S, C_r)) \quad .$$

Also this is proved by structural induction. I did not consider its inclusion, although it is fundamental in demonstrating that computations guaranteed to terminate only embody bounded non-determinacy.)

Referee B makes two comments: he states that "theorem 3 is not entirely obvious and there ought to be a quick proof of it from Fix Point Induction (which ought to be given)." I thought it fairly obvious, for a guarded list is only invoked under such initial condition that P remains invariant and t is decreased by at least 1 . But then it cannot do so indefinitely, because that same condition is stated to imply $t \geq 0$. It is easily proved by mathematical induction with k --from the definition of the semantics of the repetitive construct (3.3)-- as induction variable. I shall add a remark to that effect. (I shall not remark explicitly that one of the reasons why the formalism introduced in section 3.1, 3.2 and 3.3 attracts me is that all those properties and theorems are quite easily proved without using the Fix Point machinery. I don't like to crack eggs with sledgehammers, even if the sledgehammer is very good at it.) Referee B further remarks "It is the semantics of guarded statements that are ostensibly discussed, but in fact the major point is really the use of wp(S, R) to "tighten things up a bit" from Hoare's treatment." I am glad to hear that he regards this "tightening things up a bit" really as "a major point" --so do I myself in my more exalted moments-- but that part of, shall we call it, "theory of programming semantics" was not the subject of this paper, which was meant to be of pratical assistance for the programming process. So I just mentioned the necessary results, and proceeded to show how they can be exploited while programmin

Referee B's second remark "It is not pointed out that wp(S, R) need not be a decidable statement. That fact might help some readers to see the disadvantages of the concept." reveals the nigger in the woodpile that I had thoug to have hidden so carefully! I have taken the point of view that as long as we have not proved that an initial state satisfies wp(S, T) , the semantics are not defined and that ~~that~~ a machine, embarking upon the execution of S in such an initial state may do as it pleases in the sense that, whatever it does, we have no right to complain. As an immediate consequence I must reject the follow program for trying to refute Goldbach's Conjecture that every natural number ≥ 2 is the average of two primes:

```
begin integer n, x, y; boolean refuted;
   n:= 1; refuted:= false;
   do non refuted →
      n:= n + 1; x:= 2; y:= 2 * n;
      do x < y and x + y < 2 * n → x:= smallestprime larger than(x)
       ▯ x < y and x + y > 2 * n → y:= largest prime smaller than(y)
      od;
      refuted:= (x + y ≠ 2 * n)
   od;
   printbool(refuted)
end
```

Because I have not proved that Goldbach's Comjecture is false, I have not proved that wp(S, T) is initially true, therefore the machine may act as it pleases and I am, therefore, not allowed to conclude that Goldbach's Conjecture is wrong when it prints "true" and stops! I would be allowed to draw that surprising conclusion, however, if the third line had been changed into

"do non refuted and n < 1 000 000 000 →" ,

i.e. we must be willing to state beforehand "howlong" we are willing to wait for the answer. Now the nigger is out, I shall insert --between parentheses, so as not to alarm the reader who would like to skip the remark--

"We consider the semantics of S only defined for those initial states for which has been established a priori that they satisfy wp(S, T)." ,

leaving to the reader to decide for himself whether he regards this way of running away from the halting problem a wise restriction or a sneaky way out. Is referee B, after the above still convinced that it is "a disadvantage of the concept"? I guess, that he is still convinced of the "disadvantages". But to quote from a personal letter from one of our greatest colleagues "Some problems are better evaded than solved, such as the problem how many angels can dance on the pin of a needle (for instance by denying the existence of angels)."

I shall do justice to Referee B's remark about "less efficiency at run time (e.g. unnecessary tests).". It greatly depends on the question whether we view the fuards of a guarded command set to be evaluated concurrently, something that is entirely permitted. But even when we think of evaluating them sequentially, the expectation value of the number of tests to be performed may be less: compare the published version of Euclid's Algorithm with

```
x, y:= X, Y;
while x ≠ y do if x > y then x:= x - y else y:= y - x fi od  .
```

Referee B's QUESTION, whether we can confine the do ... od construction to guarded command sets of a single guarded command --that is what his question boils down to-- is answered by "In principle, yes", but it could be harmful to the efficiency (see the above example). It would, furthermore, be an arbitrary restriction that would destroy the symmetry between the repetitive and the alternative construct and from which I, therefore, do not expect a simplification.

I shall include a further example --I think computing X^Y for a binary and a decimal machine-- to put some meat into the assertion of the ability to "map otherwise (trivially) different programs on the same program text", in the hope of also meeting Referee A's general request about the practical use. (Although I am not sure that I can convince everybody of the "practical" significance: norms of practicality tend to be heavily dependent on personal circumstances.)

I regret that only one suggestion has been made to shorten the paper --needless to say: it will be followed-- and thank my referees also for minor remarks as typing errors: your sets of typing errors pointed out to me had an empty intersection! In order to reduce the probability of introducing new typing errors, I shall produce the final version of the manuscript with scissors and paste.

With feelings of gratitude,

yours ever

Edsger W. Dijkstra

22nd January 1975
Plataanstraat 5
NUENEN - 4565
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow

Post-scriptum d.d. 28th January 1975: EWD418 has been rebuilt into EWD472. Efforts to include other examples have been abandoned as I discovered that they would lengthen the article considerably. I have tried, instead, to formulate explicitly what would be concluded from those examples. The lower half of EWD472-4 makes one of the advantages of the guarded command set explicit by comparing my version of Euclid's Algorithm with conventional ways of writing it down. On that same example I could graft some remarks about efficience. The paragraph extending from EWD472 - 12 to EWD472 - 13 gives a further description of why they are useful. As far as contents is concerned I think that the additions are improvements; the English of the new sections seems to run less smoothly, I am sorry to say!

EWD.