Copyright Notice

A theorem about odd powers of odd integers.

Theorem. For any odd $p \geq 1$, integer $K \geq 1$, and odd r such that that $1 \leq r < 2^K$, a value x exists such that

R: $1 \leq x < 2^K$ and $2^K | (x^p - r)$ and $odd(x)$.

Note. For "a|b" read: "a divides b". (End of note.)

Proof. The existence of x is proved by designing a program computing x satisfying R .

Trying to establish R by means of a repetitive construct, we must choose an invariant relation. This time we apply the well-known technique of replacing a constant by a variable, and replace the constant K by the variable k . Introducing $d = 2^k$ for the sake of brevity, we then get

P: $d = 2^k$ and $1 \leq x < d$ and $d | (x^p - r)$ and $odd(x)$.

This choice of invariant relation P is suggested by the observation that R is trivial to satisfy for $K = 1$; hence P is trivial to establish initially. The simplest structure to try for our program is therefore:

```
x, k, d := 1, 1, 2 {P};
do k ≠ K → "increase k by 1 under invariance of P" od {R} .
```

Increasing k by 1 (together with doubling d) can only violate the term $d | (x^p - r)$. The weakest precondition that $d := 2*d$ does not do so is --according to the axiom of assignment-- $(2*d) | (x^p - r)$. Hence an acceptable component for "increase k by 1 under invariance of P" is

$$(2*d) | (x^p - r) \rightarrow k, d := k+1, 2*d \quad .$$

In the case non $(2*d) | (x^p - r)$ we conclude from $d | (x^p - r)$ that $x^p - r$ is an odd multiple of d . Because d is even, and p and x are odd, the binomial expansion tells us that $(x+d)^p - x^p$ is an odd multiple of d , and that hence $(x+d)^p - r$ is a multiple of $2*d$. Because also d is doubled, $x < d$ remains true under $x := x+d$, because d is even $odd(x)$ obviously remains true, and our program becomes:

```
x, k, d := 1, 1, 2 {P};
do k ≠ K → if (2*d)|(x^P-r) → k, d := k+1, 2*d {P}
              ▯ non (2*d)|(x^P-r) → x, k, d := x+d, k+1, 2*d {P}
           fi {P}
od {R}
```

Because this program obviously terminates, its existence proves the theorem. (End of proof.)

* * *

With the argument as given, the above program was found in five minutes. I only mention this in reply to Zohar Manna and Richard Waldinger, who wrote in "Synthesis: Dreams => Programs" (SRI Technical Note 156, November 1977)

> "Our instructors at the Structured Programming School have urged us to find the appropriate invariant assertion before introducing a loop. But how are we to select the successful invariant when there are so many promising candidates around? [...] Recursion seems to be the ideal vehicle for systematic program construction [...]. In choosing to emphasize iteration instead, the proponents of structured programming have had to resort to more dubious (sic!) means."

Although I haven't used the term Structured Programming any more for at least five years, and although I have a vested interest in recursion, yet I felt addressed by the two gentlemen. So it seemed only appropriate to record that the "more dubious means" have --again!-- been pretty effective. (I have evidence that, despite the existence of this very simple solution, the problem is not trivial: many computing scientists could not solve the programming problem within an hour. Try it on your colleagues, if you don't believe me.)

Plataanstraat 5
5671 AL Nuenen
The Netherlands

prof.dr.Edsger W.Dijkstra
Burroughs Research Fellow