262 (machine squaring) Given a natural number, write a program to find its square using only addition, subtraction, doubling, halving, test for even, and test for zero, but not multiplication or division.

After trying the question, scroll down to the solution.

§ The question says we can double, but not multiply, so I'll take that to mean that we can multiply by $2$ but not by anything else. The question says we can halve, but not divide, so I'll take that to mean that we can divide by $2$ but not by anything else. This makes sense for "machine squaring" because, in machine language, multiplying by $2$ is just shift left, and dividing by $2$ is just shift right, and test for even just looks at the rightmost bit.

For a solution with linear time we could use

$$a^2 = (a-1)^2 + 2\times a - 1$$

For a solution with logarithmic time, use

$$\textbf{if } even\ a \textbf{ then } a^2 = 4\times(a/2)^2 \textbf{ else } a^2 = 4\times((a-1)/2)^2 + 2\times a - 1 \textbf{ fi}$$

Let all variables be natural.

$$\begin{array}{ll} x{:=}\ a^2 \ \Leftarrow & \textbf{if } a{=}0 \textbf{ then } x{:=}\ 0 \\ & \textbf{else if } even\ a \textbf{ then } a{:=}\ a/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2.\ \ x{:=}\ x\times 2\times 2 \\ & \quad \textbf{else } a{:=}\ (a-1)/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2+1.\ \ x{:=}\ x\times 2\times 2 + a\times 2 - 1 \textbf{ fi fi} \end{array}$$

Note that in the solution, the occurrences of $x{:=}\ a^2$ are recursive calls. Note also that in the usual binary representation of natural numbers, $a\times 2$ is just shift left, and both $a/2$ (for even $a$ ) and $(a-1)/2$ (for odd $a$ ) are just shift right. The refinement can be proven in $3$ cases. First case:

$$\begin{array}{lll} & a{=}0 \wedge (x{:=}\ 0) & \text{expand assignment} \\ = & a{=}0 \wedge a'{=}a \wedge x'{=}0 & \text{context} \\ = & a{=}0 \wedge a'{=}a \wedge x'{=}a^2 & \text{specialization} \\ \Rightarrow & x{:=}\ a^2 & \end{array}$$

Middle case:

$$\begin{array}{lll} & a{>}0 \wedge even\ a \wedge (a{:=}\ a/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2.\ \ x{:=}x\times 2\times 2) & \text{expand final assignment} \\ = & a{>}0 \wedge even\ a \wedge (a{:=}\ a/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2.\ \ a'{=}a \wedge x'{=}x\times 2\times 2) & \text{substitution law} \\ = & a{>}0 \wedge even\ a \wedge (a{:=}\ a/2.\ \ x{:=}\ a^2.\ \ a'{=}a\times 2 \wedge x'{=}x\times 2\times 2) & \text{substitution law} \\ = & a{>}0 \wedge even\ a \wedge (a{:=}\ a/2.\ \ a'{=}a\times 2 \wedge x'{=}a^2\times 2\times 2) & \text{substitution law} \\ = & a{>}0 \wedge even\ a \wedge a'{=}a/2\times 2 \wedge x'{=}(a/2)^2\times 2\times 2 & \text{arithmetic} \\ = & a{>}0 \wedge even\ a \wedge a'{=}a \wedge x'{=}a^2 & \text{specialization} \\ \Rightarrow & x{:=}\ a^2 & \end{array}$$

Last case:

$$\begin{array}{lll} & odd\ a \wedge (a{:=}\ (a-1)/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2+1.\ \ x{:=}\ x\times 2\times 2 + a\times 2 - 1) & \\ & & \text{expand final assignment} \\ = & odd\ a \wedge (a{:=}\ (a-1)/2.\ \ x{:=}\ a^2.\ \ a{:=}\ a\times 2+1.\ \ a'{=}a \wedge x' = x\times 4 + a\times 2 - 1) & \\ & & \text{substitution law} \\ = & odd\ a \wedge (a{:=}\ (a-1)/2.\ \ x{:=}\ a^2.\ \ a' = a\times 2+1 \wedge x' = x\times 4 + (a\times 2+1)\times 2 - 1) & \\ & & \text{arithmetic} \\ = & odd\ a \wedge (a{:=}\ (a-1)/2.\ \ x{:=}\ a^2.\ \ a' = a\times 2+1 \wedge x' = x\times 4 + a\times 4 + 1) & \text{substitution law} \\ = & odd\ a \wedge (a{:=}\ (a-1)/2.\ \ a' = a\times 2+1 \wedge x' = (a^2)\times 4 + a\times 4 + 1) & \text{substitution law} \\ = & odd\ a \wedge a'{=}a \wedge x'{=}a^2 & \text{specialization} \\ \Rightarrow & x{:=}\ a^2 & \end{array}$$

For the timing, replace $x{:=}\ a^2$ by $\textbf{if } a{=}0 \textbf{ then } t'{=}t \textbf{ else } t' \le t + 1 + log\ a \textbf{ fi}$ , and put $t{:=}\ t{+}1$ in front of the recursive calls. The proof is by cases. First,

$$\begin{array}{ll} & \textbf{if } a{=}0 \textbf{ then } t'{=}t \textbf{ else } t' \le t + 1 + log\ a \textbf{ fi} \ \Leftarrow\ a{=}0 \wedge x'{=}x \wedge t'{=}t \\ = & \top \end{array}$$

The second case, right side, is

$$\begin{array}{l} a{\ne}0 \wedge even\ a \wedge (\ a{:=}\ a/2.\ \ t{:=}\ t{+}1. \\ \qquad\qquad \textbf{if } a{=}0 \textbf{ then } t'{=}t \textbf{ else } t' \le t + 1 + log\ a \textbf{ fi}. \\ \qquad\qquad a{:=}\ a\times 2.\ \ x{:=}\ x\times 2\times 2) \end{array}$$

$=$ $\quad a{\ne}0 \wedge \textit{even } a \wedge$ **if** $a/2{=}0$ **then** $t'{=}t{+}1$ **else** $t' \le t + 2 + \textit{log } (a/2)$ **fi**

$=$ $\quad a{\ne}0 \wedge \textit{even } a \wedge\ t' \le t + 2 + \textit{log } (a/2)$

$=$ $\quad a{\ne}0 \wedge \textit{even } a \wedge\ t' \le t + 1 + \textit{log } a$

$\Longrightarrow$ $\quad$ **if** $a{=}0$ **then** $t'{=}t$ **else** $t' \le t + 1 + \textit{log } a$ **fi**

which is the left side. The third case, right side, is

$a{\ne}0 \wedge \textit{odd } a \wedge (\ \ a{:=}\ (a{-}1)/2.\ \ t{:=}\ t{+}1.$

**if** $a{=}0$ **then** $t'{=}t$ **else** $t' \le t + 1 + \textit{log } a$ **fi**.

$a{:=}\ a{\times}2 + 1.\ \ x{:=}\ x{\times}2{\times}2 + a{\times}2 - 1)$

$=$ $\quad a{\ne}0 \wedge \textit{odd } a \wedge$ **if** $(a{-}1)/2{=}0$ **then** $t'{=}t{+}1$ **else** $t' \le t + 2 + \textit{log } ((a{-}1)/2)$ **fi**

$=$ $\quad a{\ne}0 \wedge \textit{odd } a \wedge$ **if** $a{=}1$ **then** $t'{=}t{+}1$ **else** $t' \le t + 1 + \textit{log } (a{-}1)$ **fi**

$\Longrightarrow$ $\quad$ **if** $a{=}0$ **then** $t'{=}t$ **else** $t' \le t + 1 + \textit{log } a$ **fi**

which is the left side.

Here's the best solution. Define

$P\ \ =\ \ y' = y + x{\times}n\ \ \wedge$ **if** $x{=}0$ **then** $t'{=}t$ **else** $t' \le t + \textit{log } x$ **fi**

Then the program is

$y'{=}x^2\ \wedge$ **if** $x{=}0$ **then** $t'{=}t$ **else** $t' \le t + \textit{log } x$ **fi** $\ \Longleftarrow\ \ y{:=}\ 0.\ \ n{:=}\ x.\ \ P$

$P\ \Longleftarrow$ **if** *even* $x$ **then** $\textit{even } x \Rightarrow P$ **else** $\textit{odd } x \Rightarrow P$ **fi**

$\textit{even } x \Rightarrow P\ \Longleftarrow$ **if** $x{=}0$ **then** *ok* **else** $\textit{even } x \wedge x{>}0 \Rightarrow P$ **fi**

$\textit{odd } x \Rightarrow P\ \Longleftarrow\ \ y{:=}\ y{+}n.\ \ x{:=}\ x{-}1.\ \ \textit{even } x \Rightarrow P$

$\textit{even } x \wedge x{>}0 \Rightarrow P\ \Longleftarrow\ \ n{:=}\ 2{\times}n.\ \ x{:=}\ x/2.\ \ t{:=}\ t{+}1.\ \ x{>}0 \Rightarrow P$

$x{>}0 \Rightarrow P\ \Longleftarrow$ **if** *even* $x$ **then** $\textit{even } x \wedge x{>}0 \Rightarrow P$ **else** $\textit{odd } x \Rightarrow P$ **fi**