159 Let n and r be natural variables in the refinement

$P \Leftarrow$ **if** $n{=}1$ **then** $r{:=}\ 0$ **else** $n{:=}\ div\ n\ 2.\ P.\ r{:=}\ r{+}1$ **fi**

Suppose the operations div and $+$ each take time 1 and all else is free (even the call is free). Insert appropriate time increments, and find an appropriate P to express the execution time in terms of

(a) the initial values of the memory variables. Prove the refinement for your choice of P.

§ With time increments added, I must prove

$P \Leftarrow$ **if** $n{=}1$ **then** $r{:=}\ 0$ **else** $t{:=}\ t{+}1.\ n{:=}\ div\ n\ 2.\ P.\ t{:=}\ t{+}1.\ r{:=}\ r{+}1$ **fi**

How should we choose P? Execution of P proceeds as follows. If n is initially 0, then n is divided by 2, making it again 0, and we are in an infinite loop. If n is initially positive, then it is repeatedly divided by 2 (rounding down) until it becomes 1, then r is assigned 0, then r is incremented as many times as n was divided by 2. The number of times n is divided by 2 until it becomes 1 is the logarithm (base 2) of n. This may not be obvious, so I can easily code this procedure in any implemented programming language I like, and run it for a variety of initial values for n and r and for initial time 0, and see that the final value of t is $2 \times floor\ (log\ n)$. So P can be

$(n{=}0 \Rightarrow t'{=}\infty) \wedge (n{>}0 \Rightarrow t' = t + 2 \times floor\ (log\ n))$

But $floor$ is an awkward function to work with, so I'll get rid of it by replacing the exact time with an upper bound. My choice of P is

$(n{=}0 \Rightarrow t'{=}\infty) \wedge (n{>}0 \Rightarrow t' \le t + 2 \times log\ n)$

I prove it in parts (each conjunct separately), and I prove each part by cases.
First part, first case:

$(n{=}0 \Rightarrow t'{=}\infty) \Leftarrow n{=}1 \wedge (r{:=}\ 0)$ portation

$= n{=}0 \wedge n{=}1 \wedge (r{:=}\ 0) \Rightarrow t'{=}\infty$

$= \bot \wedge (r{:=}\ 0) \Rightarrow t'{=}\infty$

$= \bot \Rightarrow t'{=}\infty$

$= \top$

First part, last case:

$(n{=}0 \Rightarrow t'{=}\infty) \Leftarrow n{\neq}1 \wedge (t{:=}\ t{+}1.\ n{:=}\ div\ n\ 2.\ n{=}0 \Rightarrow t'{=}\infty.\ t{:=}\ t{+}1.\ r{:=}\ r{+}1)$
portation and expand final assignment

$= n{=}0 \wedge n{\neq}1 \wedge (t{:=}\ t{+}1.\ n{:=}\ div\ n\ 2.\ n{=}0 \Rightarrow t'{=}\infty.\ t{:=}\ t{+}1.\ r'{=}r{+}1 \wedge n'{=}n \wedge t'{=}t)$
$\Rightarrow t'{=}\infty$ simplify, and substitution law in two parts

$= n{=}0 \wedge (div\ n\ 2 = 0 \Rightarrow t'{=}\infty.\ r'{=}r{+}1 \wedge n'{=}n \wedge t'{=}t{+}1)$
$\Rightarrow t'{=}\infty$ eliminate sequential composition

$= n{=}0 \wedge (\exists r'', n'', t''\cdot\ (div\ n\ 2 = 0 \Rightarrow t''{=}\infty) \wedge r'{=}r''{+}1 \wedge n'{=}n'' \wedge t'{=}t''{+}1)$
$\Rightarrow t'{=}\infty$ context: $n{=}0$

$= n{=}0 \wedge (\exists r'', n'', t''\cdot\ t''{=}\infty \wedge r'{=}r''{+}1 \wedge n'{=}n'' \wedge t'{=}t''{+}1) \Rightarrow t'{=}\infty$ one-point

$= n{=}0 \wedge t'{=}\infty{+}1 \Rightarrow t'{=}\infty$ absorption and specialization

$= \top$

Last part, first case:

$(n{>}0 \Rightarrow t' \le t + 2 \times log\ n) \Leftarrow n{=}1 \wedge (r{:=}\ 0)$ portation and expand assignment

$= n{=}1 \wedge r'{=}0 \wedge n'{=}n \wedge t'{=}t \Rightarrow t' \le t + 2 \times log\ n$ context, and $log\ 1 = 0$

$= \top$

Last part, last case:

$(n{>}0 \Rightarrow t' \le t + 2 \times log\ n)$

$\Leftarrow$ $n{\neq}1 \wedge (t{:=}\ t{+}1.\ \ n{:=}\ div\ n\ 2.\ \ n{>}0 \Rightarrow t' \le t + 2 \times log\ n.\ \ t{:=}\ t{+}1.\ \ r{:=}\ r{+}1)$

portation and expand final assignment

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (t{:=}t{+}1.\ n{:=}\ div\ n\ 2.\ n{>}0 \Rightarrow t' \le t + 2{\times}log\ n.\ t{:=}\ t{+}1.\ r'{=}r{+}1 \wedge n'{=}n{\wedge}t'{=}t)$

substitution law in two parts

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (div\ n\ 2{>}0 \Rightarrow t' \le t{+}1{+}\ 2 \times log(div\ n\ 2).\ \ r'{=}r{+}1 \wedge n'{=}n \wedge t'{=}t{+}1)$

eliminate sequential composition

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (\exists r'', n'', t''\cdot \quad (div\ n\ 2{>}0 \Rightarrow t'' \le t{+}1{+}\ 2 \times log(div\ n\ 2))$
$\wedge\ \ r'{=}r''{+}1 \wedge n'{=}n'' \wedge t'{=}t''{+}1)$ one-point for n'' and t''

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (\exists r''\cdot\ (div\ n\ 2{>}0 \Rightarrow t' \le t{+}2{+}\ 2 \times log(div\ n\ 2)) \wedge\ r'{=}r''{+}1)$ distributive

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (\exists r'{:}\ nat\cdot\ r'{=}r''{+}1) \wedge (div\ n\ 2{>}0 \Rightarrow t' \le t{+}2{+}\ 2 \times log(div\ n\ 2))$

in preparation for one-point, rewrite $r'{=}r''{+}1$ and make $r''{:}\ nat$ an explicit conjunct

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (\exists r'{:}\ nat\cdot\ r''{=}r'{-}1 \wedge (r''{:}nat)) \wedge (div\ n\ 2{>}0 \Rightarrow t' \le t{+}2{+}\ 2 \times log(div\ n\ 2))$

now use one-point

$=$ $t' \le t + 2 \times log\ n$

$\Leftarrow$ $n{>}1 \wedge (r'{-}1{:}\ nat) \wedge (div\ n\ 2{>}0 \Rightarrow t' \le t{+}2{+}\ 2 \times log(div\ n\ 2))$

$=$ $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge (r'{-}1{:}\ nat) \wedge (div\ n\ 2{>}0 \Rightarrow t' \le t + 2 + 2 \times log(div\ n\ 2))$

simplify $div\ n\ 2{>}0$

$=$ $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(div\ n\ 2))$

increase $div\ n\ 2$ to $n/2$

this will increase $t + 2 + 2 \times log(div\ n\ 2)$

this will weaken $t' \le t + 2 + 2 \times log(div\ n\ 2)$

this will weaken $n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(div\ n\ 2)$

this will weaken $n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(div\ n\ 2))$

this will strengthen $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(div\ n\ 2))$

so we need to put $\Longleftarrow$ in the left margin

$\Longleftarrow$ $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(n/2))$ drop $r'{\ge}1$

this will weaken $n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(n/2))$

this will strengthen $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge r'{\ge}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(n/2))$

so again we need to put $\Longleftarrow$ in the left margin

$\Longleftarrow$ $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge (n{>}1 \Rightarrow t' \le t + 2 + 2 \times log(n/2))$ discharge and simplify

$=$ $t' \le t + 2 \times log\ n \ \Leftarrow\ n{>}1 \wedge t' \le t + 2 \times log\ n$ specialization

$=$ $\top$

(b) the final values of the memory variables. Prove the refinement for your choice of P.

§ I prove

$t' = t{+}2{\times}r' \ \Longleftarrow$ **if** $n{=}1$ **then** $r{:=}\ 0$
else $t{:=}\ t{+}1.\ \ n{:=}\ div\ n\ 2.\ \ t' = t{+}2{\times}r'.\ \ t{:=}\ t{+}1.\ \ r{:=}\ r{+}1$ **fi**

by cases. First case:

$t' = t{+}2{\times}r' \ \Leftarrow\ n{=}1 \wedge (r{:=}\ 0)$ expand assignment

$=$ $t' = t{+}2{\times}r' \ \Leftarrow\ n{=}1 \wedge r'{=}0 \wedge n'{=}n \wedge t'{=}t$

$=$ $\top$

Last case:

$t' = t+2\times r' \;\Leftarrow\; n\neq 1 \wedge (t := t+1.\;\; n := \mathit{div}\; n\; 2.\;\; t' = t+2\times r'.\;\; t := t+1.\;\; r := r+1)$

expand final assignment

$= \quad t' = t+2\times r' \Leftarrow n\neq 1 \wedge (t:=t+1.\; n:=\mathit{div}\; n\; 2.\; t' = t+2\times r'.\; t:= t+1.\; r'=r+1 \wedge n'=n \wedge t'=t)$

substitution law in two parts

$= \quad t' = t+2\times r' \;\Leftarrow\; n\neq 1 \wedge (t' = t+1+2\times r'.\; r'=r+1 \wedge n'=n \wedge t'=t+1)$ one-point

$= \quad t' = t+2\times r' \;\Leftarrow\; n\neq 1 \wedge t' = t+2+2\times(r'-1)$

$= \quad \top$