Let $n$ and $r$ be natural variables in the refinement

$$P \iff \text{if } n=1 \text{ then } r:=0 \text{ else } n:= \text{div } n \times 2. \text{ P. } r:= r+1 \text{ fi}$$

Suppose the operations $\text{div}$ 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

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

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 $2 \times \text{floor}(\log n)$. So $P$ can be

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

But $\text{floor}$ 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

$$(n=0 \Rightarrow t'=\infty) \land (n>0 \Rightarrow t' \leq t + 2 \times \log n)$$

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

First part, first case:

$$(n=0 \Rightarrow t'=\infty) \iff n=1 \land (r:= 0)$$

$$= n=0 \land n=1 \land (r:= 0) \Rightarrow t'=\infty$$

$$= \perp \land (r:= 0) \Rightarrow t'=\infty$$

$$= \perp \Rightarrow t'=\infty$$

$$\Rightarrow \top$$

First part, last case:

$$(n=0 \Rightarrow t'=\infty) \iff n+1 \land (t:= t+1. \ n:= \text{div } n \times 2. \ n=0 \Rightarrow t'=\infty. \ t:= t+1. \ r:= r+1)$$

$$= n=0 \land n+1 \land (t:= t+1. \ n:= \text{div } n \times 2. \ n=0 \Rightarrow t'=\infty. \ t:= t+1. \ r'=r+1 \land n'=n \land t'=t)$$

$$\Rightarrow t'=\infty$$

$$= n=0 \land (\text{div } n \times 2 = 0 \Rightarrow t'=\infty. \ r'=r+1 \land n'=n \land t'=t+1)$$

$$\Rightarrow t'=\infty$$

$$= n=0 \land (\exists n', n'', t''. \ (\text{div } n \times 2 = 0 \Rightarrow t''=\infty) \land r'=r''+1 \land n'=n'' \land t'=t''+1)$$

$$\Rightarrow t'=\infty$$

$$= n=0 \land (\exists n', n'', t''. \ t''=\infty \land r'=r''+1 \land n'=n'' \land t'=t''+1) \Rightarrow t'=\infty$$

$$= n=0 \land (n=1 \land (n'=0 \land t'=t) \Rightarrow t' \leq t + 2 \times \log n)$$

$$\Rightarrow \top$$

Last part, first case:

$$(n>0 \Rightarrow t' \leq t + 2 \times \log n) \iff n=1 \land (r:= 0)$$

$$= n=1 \land r'=0 \land n'=n \land t'=t \Rightarrow t' \leq t + 2 \times \log n$$

$$\Rightarrow \top$$

Last part, last case:
\[(n \geq 0 \Rightarrow t' \leq t + 2 \times \log n)\]
\[
\iff n > 1 \land (t := t + 1 \land n := \text{div} \ n \ 2 \land n > 0 \Rightarrow t' \leq t + 2 \times \log n \land t := t + 1 \land r := r + 1) \]

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

\section{Case Study}

I prove
\[
t' = t + 2 \times r' \iff \text{if } n = 1 \text{ then } r := 0 \\
\quad \text{else } t := t + 1 \land n := \text{div} \ n \ 2 \land t' = t + 2 \times r' \land t := t + 1 \land r := r + 1 \fi
\]

by cases. First case:
\[
t' = t + 2 \times r' \iff n = 1 \land (r := 0) \]
\[
\iff n = 1 \land r' = 0 \land n' = n \land t' = t
\]
\[
\iff \top
\]
Last case:

\[
t' = t + 2 \times r' \iff n \neq 1 \land (t := t + 1. \ n := \text{div} n 2. \ t' = t + 2 \times r'. \ t := t + 1. \ r := r + 1)
\]

expand final assignment

\[
t' = t + 2 \times r' \iff n \neq 1 \land (t := t + 1. \ n := \text{div} n 2. \ t' = t + 2 \times r'. \ t := t + 1. \ r' := r + 1 \land n' = n \land t' = t)
\]

substitution law in two parts

\[
t' = t + 2 \times r' \iff n \neq 1 \land (t' = t + 1 + 2 \times r'. \ r' = r + 1 \land n' = n \land t' = t + 1)
\]

one-point

\[
t' = t + 2 \times r' \iff n \neq 1 \land t' = t + 2 + 2 \times (r' - 1)
\]

\[
\top
\]