Let the given natural number be the initial value of natural variable \( n \), and report the answer as the final value of binary variable \( p \). Define
\[
R \equiv p' = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right)
\]
\[
Q \equiv p' = (p = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right))
\]
Then the refinements are
\[
R \iff p := \top. \ Q
\]
\[
Q \iff \text{if } n=0 \text{ then } \text{ok} \text{ else } p := p = \text{even} \ n. \ n := \text{div} \ n \ 2. \ Q \ 	ext{fi}
\]
The proof of the first refinement is one trivial use of the Substitution Law. The second refinement can be proven by cases. The first case is
\[
p' = (p = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right)) \iff n=0 \land \text{ok}
\]
\[
= p' = (p = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right)) \iff n=0 \land p' = p \land n' = n
\]
\[
\iff p = (p = \top) \iff n=0 \land p' = p \land n' = n
\]
\[
\iff p = p \iff n=0 \land p' = p \land n' = n
\]
\[
\iff \top \iff n=0 \land p' = p \land n' = n
\]
\[
\iff \top
\]
Just before doing the last case, here is a piece of arithmetic.
\[
\text{div} (\text{div} n 2^i) 2^i = \text{div} n 2^{i+j}
\]
because chopping off \( i \) bits from the right end of a binary number followed by chopping off \( j \) more bits is the same as chopping off \( i+j \) bits.
The last refinement, last case, is
\[
p' = (p = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right)) \iff n=0 \land p' = p \land n' = n
\]
\[
\iff p' = (p = \text{even} \left( \Sigma i: \text{nat} \mod (\text{div} n 2^i) 2 \right)) \iff n=0 \land p' = p \land n' = n \text{ simplify}
\]
\[
\iff p = (p = \top) \iff n=0 \land p' = p \land n' = n \text{ identity}
\]
\[
\iff p = p \iff n=0 \land p' = p \land n' = n \text{ is reflexive}
\]
\[
\iff \top \iff n=0 \land p' = p \land n' = n \text{ base}
\]
\[
\iff \top
\]
Now for the timing. Define
\[
T \equiv \text{if } n=0 \text{ then } t'=t \text{ else } t' \leq t + \log n \ \text{fi}
\]
Then the refinements are
\[
T \iff p := \top. \ T
\]
\[
T \iff \text{if } n=0 \text{ then } \text{ok} \text{ else } p := p = \text{even} \ n. \ n := \text{div} \ n \ 2. \ t := t+1. \ T \ 	ext{fi}
\]
The proof of the first refinement is one trivial use of the Substitution Law. The second refinement is proven by cases. The first case is:
\[
T \iff n=0 \land \text{ok}
\]
\[
\iff \text{if } n=0 \text{ then } t'=t \text{ else } t' \leq t + \log n \ \text{fi} \iff n=0 \land n'=n \land p' = p \land t'=t \text{ context}
\]
\(\equiv\) if \(0=0\) then \(t\) else \(t \leq t + \log 0\) fi \(\iff\) \(n=0 \land n'=n \land p'=p \land t'=t\)

simplify

\(\equiv\) \(\top\) \(\iff\) \(n=0 \land n'=n \land p'=p \land t'=t\)

base

The other case is

\(\equiv\) \(\top\)

\(\equiv\) \(n>0 \land (p := p = \text{even}\ n.\ n := \text{div}\ n\ 2.\ t := t+1.\ T)\) expand \(T\) and substitute

\(\equiv\) if \(n=0\) then \(t'=t\) else \(t' \leq t + \log n\) fi

\(\iff\) \(n>0 \land \) if \(\text{div}\ n\ 2 = 0\) then \(t'=t+1\) else \(t' \leq t+1 + \log (\text{div}\ n\ 2)\) fi

use \(n>0\) as context

\(\equiv\) \(t' \leq t + \log n\)

\(\iff\) \(n>0 \land \) if \(n=1\) then \(t'=t+1\) else \(t' \leq t+1 + \log (\text{div}\ n\ 2)\) fi

increase \(\text{div}\ n\ 2\) to \(n/2\)

\(\equiv\) \(t' \leq t + \log n\)

\(\iff\) \(n>0 \land \) if \(n=1\) then \(t'=t+1\) else \(t' \leq t+1 + \log (n/2)\) fi

use context \(n=1\) in \(\text{then}\) part, and \(\log\) law in \(\text{else}\) part

\(\equiv\) \(t' \leq t + \log n\)

\(\iff\) \(n>0 \land \) if \(n=1\) then \(t'=t + \log n\) else \(t' \leq t + \log n\) fi

case idempotent and specialize

\(\equiv\) \(\top\)