476 Let b: *bin* be the user's variable, and let n: *nat* be the implementer's variable, and let the operations be

$step \;=\; \textbf{if } n>0 \textbf{ then } n:= n-1 \textbf{ else } ok \textbf{ fi}$
$done \;=\; b:= n=0$

Show that there is no transformer to get rid of n so that

step	is transformed to	*ok*
done	is transformed to	$b:= \bot$

even though the user cannot detect the difference.

After trying the question, scroll down to the solution.

§ Suppose a user is imagining that there is a variable n with some natural value, and *step* decreases it until it is 0, and *done* says whether n is 0. But actually there is no variable n, and *step* does nothing, and *done* always says " n is not yet 0 ". The user will never know that there is no variable n. That's why the question says "the user cannot detect the difference".

The transformer would have to be a binary expression D in variables b and n such that

(0) $\exists n\cdot D$

(1) $\forall n\cdot D \Rightarrow \exists n'\cdot D' \wedge \textbf{if } n{>}0 \textbf{ then } n:=n-1 \textbf{ else } ok \textbf{ fi} \;\Longleftarrow\; ok$

(2) $\forall n\cdot D \Rightarrow \exists n'\cdot D' \wedge (b:=n{=}0) \;\Longleftarrow\; b:=\bot$

Instead of a binary expression, let D be a relation between a binary and a natural. Then

(0) $\exists n\cdot D\,b\,n$

(1) $\forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge \textbf{if } n{>}0 \textbf{ then } n:=n-1 \textbf{ else } ok \textbf{ fi} \;\Longleftarrow\; ok$

(2) $\forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge (b:=n{=}0) \;\Longleftarrow\; b:=\bot$

Now calculate, starting with the left side of 1.

$\forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge \textbf{if } n{>}0 \textbf{ then } n:=n-1 \textbf{ else } ok \textbf{ fi}$ — replace **if** and $:=$ and ok

$= \forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge (n{>}0 \wedge n'{=}n-1 \wedge b'{=}b) \vee (n{=}0 \wedge n'{=}n \wedge b'{=}b)$ — distribute $D\,b'n'$

$= \forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot (D\,b'n' \wedge n{>}0 \wedge n'{=}n-1 \wedge b'{=}b) \vee (D\,b'n' \wedge n{=}0 \wedge n'{=}n \wedge b'{=}b)$ — splitting

$= \forall n\cdot D\,b\,n \Rightarrow (\exists n'\cdot D\,b'n' \wedge n{>}0 \wedge n'{=}n-1 \wedge b'{=}b) \vee (\exists n'\cdot D\,b'n' \wedge n{=}0 \wedge n'{=}n \wedge b'{=}b)$ — one-point

$= \forall n\cdot D\,b\,n \Rightarrow (D\,b'(n-1) \wedge n{>}0 \wedge b'{=}b) \vee (D\,b'n \wedge n{=}0 \wedge b'{=}b)$ — context

$= \forall n\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0 \wedge b'{=}b) \vee (D\,b\,n \wedge n{=}0 \wedge b'{=}b)$ — factor out $b'{=}b$

$= b'{=}b \wedge \forall n\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0) \vee (D\,b\,n \wedge n{=}0)$ — context

$= b'{=}b \wedge \forall n\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0) \vee (\top \wedge n{=}0)$ — identity

$= b'{=}b \wedge \forall n\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0) \vee n{=}0$ — basic $\forall$ law

$= b'{=}b \wedge (\forall n\!: 0\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0) \vee n{=}0)$ — The 0 case is $\top$.
$\quad \wedge (\forall n\!: nat{+}1\cdot D\,b\,n \Rightarrow (D\,b\,(n-1) \wedge n{>}0) \vee n{=}0)$

$= b'{=}b \wedge \forall n\!: nat{+}1\cdot D\,b\,n \Rightarrow D\,b\,(n-1)$ — change of variable

$= b'{=}b \wedge \forall n\!: nat\cdot D\,b\,(n+1) \Rightarrow D\,b\,n$ — mirror

$= b'{=}b \wedge \forall n\cdot D\,b\,n \Leftarrow D\,b\,(n+1)$

According to (1), this is refined by ok (in a context without n). So

$b'{=}b \wedge (\forall n\cdot D\,b\,n \Leftarrow D\,b\,(n+1)) \;\Leftarrow\; b'{=}b$ — context

$= \top \wedge (\forall n\cdot D\,b\,n \Leftarrow D\,b\,(n+1)) \;\Leftarrow\; b'{=}b$ — identity and mirror

$= b'{=}b \;\Rightarrow\; \forall n\cdot D\,b\,n \Leftarrow D\,b\,(n+1)$ — one-point b'

(3) $= \forall n\cdot D\,b\,n \Leftarrow D\,b\,(n+1)$

$=$ $D\,b\,n$ is antimonotonic in n; in other words, an initial (possibly empty, possibly infinite) segment of $D\,b\,0;\, D\,b\,1;\, ...$ is $\top$, and after that it's $\bot$.

Now calculate, starting with the left side of 2.

$= \forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge (b:=n{=}0)$ — replace $:=$

$= \forall n\cdot D\,b\,n \Rightarrow \exists n'\cdot D\,b'n' \wedge b'{=}(n{=}0) \wedge n'{=}n$ — factor out $b'{=}(n{=}0)$

$= \forall n\cdot D\,b\,n \Rightarrow b'{=}(n{=}0) \wedge \exists n'\cdot D\,b'n' \wedge n'{=}n$ — one-point n'

$= \forall n\cdot D\,b\,n \Rightarrow b'{=}(n{=}0) \wedge D\,b'n$ — splitting

$= (\forall n\cdot D\,b\,n \Rightarrow b'{=}(n{=}0)) \wedge (\forall n\cdot D\,b\,n \Rightarrow D\,b'n)$

According to (2), this is refined by $b:=\bot$ (in a context without n). So, explicitly quantifying over b and b',

$\forall b, b'\cdot (\forall n\cdot D\,b\,n \Rightarrow b'{=}(n{=}0)) \wedge (\forall n\cdot D\,b\,n \Rightarrow D\,b'n) \;\Leftarrow\; (b:=\bot)$ — expand $:=$

$= \forall b, b'\cdot (\forall n\cdot D\,b\,n \Rightarrow b'{=}(n{=}0)) \wedge (\forall n\cdot D\,b\,n \Rightarrow D\,b'n) \;\Leftarrow\; b'{=}\bot$ — one-point b'

$= \forall b\cdot (\forall n\cdot D\,b\,n \Rightarrow n{>}0) \wedge (\forall n\cdot D\,b\,n \Rightarrow D\,\bot\,n)$ — contrapositive

$=\quad \forall b\cdot (\forall n\cdot n=0 \Rightarrow \neg D\ b\ n) \wedge (\forall n\cdot D\ b\ n \Rightarrow D \perp n)$ one-point n

(4) $=\quad \forall b\cdot\ \neg D\ b\ 0 \wedge (\forall n\cdot D\ b\ n \Rightarrow D \perp n)$

From (4) we have $\neg D\ b\ 0$. So from (3) we have $\forall n\cdot \neg D\ b\ n$. And this contradicts (0). Hence there is no data transformer to do the job.

If the user is willing to test *done* to see if $n=0$ before each use of *step* , then we can weaken *step* to $n>0 \Rightarrow (n:= n-1)$, and perhaps find a transformer to transform *step* to *ok* and transform *done* to $b:= \perp$.