theory *EX-RecMult-deep* = *SALOverflowFWInst-deep*:

1 SAL Example: Multiplication

This example is about a program that multplies variable C1 with C2 and writes the result to C3.

1.1 Program Variables

The variables C1, C2, C3 are just abbreviations for the addresses 1, 2 and 3.

constdefs
C1 :: *nat* — cell1
C1 ≡ *1*
C2 :: *nat* — cell2
C2 ≡ *2*
C3 :: *nat* — cell3
C3 ≡ *3*

At some point in the program we need to check whether a variable contains number one or zero. Since conditional jumps in SAL expect their arguments in form of variables, we introduce variables Zero and One. The progam uses them to the constants NAT 0 and NAT 1.

constdefs
Zero::*nat* — stores NAT 0
Zero ≡ *4*
One:: *nat* — stores NAT 1
One ≡ *5*

Our program will do multiplication by recursively calling a procedure that decrements C1 and adds C2 to C3. Whenever this procedure is called, the return address, which is the current address incremented by one, is dumped in a variable r. The address of r, which is 6, is kept in the variable Ra.

constdefs
r :: *nat* — return address buffer
r ≡ *6*
Ra :: *nat* — stores the address of R
Ra ≡ *7*

The program maintains a stack of return addresses. This stack starts at address Soff and grows towards higher addresses. The variable S contains the stack pointer. This is the address, where the next element of the stack is going to be stored.

constdefs
S:: *nat* — stack pointer
S ≡ *8*
Soff:: *nat* — stack base
Soff ≡ *9*

Since SAL programs are executed on a default initial state, we have to simulate the input phase of the program by SET operations. In this example we analyse the program for inputs arg1 and arg2.

constdefs
arg1::*nat*
arg1≡*2*
arg2::*nat*
arg2≡*3*

1.2 Program Code

Next, we present the program without annotations.

prog = [(*0*, [Procedure 0 (Startup)
SET Zero 0, Initialise Zero with NAT 0.
SET One 1, Initialise One with NAT 1.
SET POS r, Initialise POS with NAT r (r = 6).
SET S Soff, Initialise S with NAT Soff (Soff = 9).
SET C1 arg1, Initialise Argument C1 with NAT arg1.
SET C2 arg2, Initialise Argument C2 with NAT arg2.
SET C3 0, Initialise Result Variable C3 with NAT 0.
CALL r 1, Start multplication procedure.
HALT]), Stop execution

(*1*,[Procedure 1 (Multiplication)
MOV Ra S, Push the return address onto the Stack.
JMPEQ C1 Zero, If C1 is NAT 0 we are done, otherwise . . .
SUB C1 One,
. . . we decrement C1
ADD C3 C2, and add C2 to the result variable C3.
INC S, We increment the stack pointer, before .
CALL r 1, we do the recursive call.
SUB S One, After return we restore the old stack pointer
MOV S Ra, and copy the return address from the stack to r.
RET r])] We finish this call by returning to the caller.

1.3 Program with Annotations

constdefs
prog :: *SALprogram*
prog ≡ [
(*0*, [(*SET Zero 0*,*None*),
(*SET One 1*,*None*),
(*SET Ra r*,*None*),
(*SET S Soff*,*None*),
(*SET C1 arg1*,*None*),
(*SET C2 arg2*,*None*),
(*SET C3 0*,*None*),

$$
\begin{array}{l}
(CALL\ r\ 1, Some\ (\Lambda\ [(V\ C1 \doteq C\ (NAT\ arg1)),\\
\qquad (V\ C2 \doteq C\ (NAT\ arg2)),\\
\qquad (V\ C3 \doteq C\ (NAT\ 0)),\\
\qquad (V\ Zero \doteq C\ (NAT\ 0)),\\
\qquad (V\ One \doteq C\ (NAT\ 1)),\\
\qquad (V\ Ra \doteq C\ (NAT\ r)),\\
\qquad (V\ S \doteq C\ (NAT\ Soff)),\\
\qquad (\prod\ 0.\ (\Lambda\ [\ Neg\ (Lv\ 0 \doteq (C\ (NAT\ C1))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ C2))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ C3))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ Zero))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ One))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ Ra))),\\
\qquad\qquad Neg\ (Lv\ 0 \doteq (C\ (NAT\ S)))]) \supset ((Deref\ (Lv\ 0)) \doteq\\
(Old\ (Deref\ (Lv\ 0)))))]))),
\end{array}
$$

$$
\begin{array}{l}
(JMPB\ 0,\ Some\ \ (V\ C3 \doteq (C\ (NAT\ arg1)) \odot (C\ (NAT\ arg2))))]),\\
(1,[\ (MOV\ Ra\ S, Some\ (\Lambda\ [\ Ty\ (V\ C1)\ Nat,\ Ty\ (V\ C2)\ Nat,\ Ty\ (V\ C3)\ Nat,\\
\qquad ((V\ C3) \oplus ((V\ C1) \odot (V\ C2))) \preceq (C\ (NAT\ MAX)),\\
\qquad (V\ Zero) \doteq (C\ (NAT\ 0)),\\
\qquad (V\ One) \doteq (C\ (NAT\ 1)),\\
\qquad (V\ Ra) \doteq (C\ (NAT\ r)),\\
\qquad Ty\ (V\ S)\ Nat,\ (C\ (NAT\ Soff)) \preceq (V\ S),\\
\qquad ((V\ S) \oplus (V\ C1)) \preceq C\ (NAT\ MAX),\\
\qquad (V\ r) \doteq\ Rp,\\
\qquad (\prod\ 0.\ (Neg\ (Lv\ 0 \doteq (C\ (NAT\ r)))) \supset ((Deref\ (Lv\ 0)) \doteq (Old\\
(Deref\ (Lv\ 0)))))]))),
\end{array}
$$

$$
\begin{array}{l}
(JMPEQ\ C1\ Zero\ 7, None),\\
(SUB\ C1\ One, None),\\
(ADD\ C3\ C2, None),\\
(INC\ S, None),\\
(CALL\ r\ 1, Some\ (\Lambda\ [\ Ty\ (V\ C1)\ Nat,\ Ty\ (V\ C2)\ Nat,\ Ty\ (V\ C3)\ Nat,\\
\qquad Ty\ (Old\ (V\ C1))\ Nat,\ Ty\ (Old\ (V\ C2))\ Nat,\ \ Ty\ (Old\ (V\\
C3))\ Nat,\\
\qquad (C\ (NAT\ 0)) \prec (Old\ (V\ C1)),\\
\qquad (V\ C1) \doteq Old\ ((V\ C1) \ominus (C\ (NAT\ 1))),\\
\qquad (V\ C2) \doteq Old\ (V\ C2),\\
\qquad (V\ C3) \doteq Old\ ((V\ C3) \oplus (V\ C2)),\\
\qquad ((V\ C3) \oplus ((V\ C1) \odot (V\ C2))) \preceq (C\ (NAT\ MAX)),\\
\qquad (V\ Zero) \doteq (C\ (NAT\ 0)),\\
\qquad (V\ One) \doteq (C\ (NAT\ 1)),\\
\qquad (V\ Ra) \doteq (C\ (NAT\ r)),\\
\qquad Ty\ (V\ S)\ Nat,\\
\qquad (C\ (NAT\ Soff)) \preceq (V\ S),\\
\qquad Ty\ (Old\ (V\ S))\ Nat,\\
\qquad (V\ S) \doteq Old\ ((V\ S) \oplus (C\ (NAT\ 1))),\\
\qquad Deref\ (Old\ (V\ S)) \doteq Rp,\\
\qquad ((V\ S) \oplus (V\ C1)) \preceq (C\ (NAT\ MAX)),
\end{array}
$$

$$\prod\ 0.\ ((\Lambda\ [\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ r)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ S)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C1)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C3)),\ (((Lv\ 0)) \oplus (C\ (NAT\ 1))) \prec (V\ S)]) \supset ((Deref\ (Lv\ 0)) \doteq (Old\ (Deref\ (Lv\ 0)))))]))$$

$$(SUB\ S\ One, Some\ (\Lambda\ [\ Ty\ (V\ C3)\ Nat,\ Ty\ (Old\ (V\ C3))\ Nat,\ Ty\ (Old\ (V\ C1))\ Nat,\ Ty\ (Old\ (V\ C2))\ Nat,\ (V\ C3) \doteq Old\ ((V\ C3) \oplus ((V\ C1) \odot (V\ C2))),\ (V\ Zero) \doteq (C\ (NAT\ 0)),\ (V\ One) \doteq (C\ (NAT\ 1)),\ (V\ Ra) \doteq (C\ (NAT\ r)),\ Ty\ (V\ S)\ Nat,\ (C\ (NAT\ Soff)) \preceq (V\ S),\ Ty\ (Old\ (V\ S))\ Nat,\ (V\ S) \doteq Old\ ((V\ S) \oplus (C\ (NAT\ 1))),\ Deref\ (Old\ (V\ S)) \doteq Rp,\ ((V\ S) \oplus (V\ C1)) \preceq (C\ (NAT\ MAX)),\ \prod\ 0.\ ((\Lambda\ [\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ r)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ S)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C1)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C3)),\ (((Lv\ 0)) \oplus (C\ (NAT\ 1))) \prec (V\ S)]) \supset ((Deref\ (Lv\ 0)) \doteq (Old\ (Deref\ (Lv\ 0)))))]))),$$

$$(MOV\ S\ Ra, None),$$

$$(RET\ r, Some\ (\Lambda\ [\ Ty\ (V\ C3)\ Nat,\ Ty\ (Old\ (V\ C3))\ Nat,\ Ty\ (Old\ (V\ C1))\ Nat,\ Ty\ (Old\ (V\ C2))\ Nat,\ (V\ C3) \doteq Old\ ((V\ C3) \oplus ((V\ C1) \odot (V\ C2))),\ (V\ Zero) \doteq (C\ (NAT\ 0)),\ (V\ One) \doteq (C\ (NAT\ 1)),\ (V\ Ra) \doteq C\ (NAT\ r),\ (V\ r) \doteq Rp,\ Ty\ (V\ S)\ Nat,\ (C\ (NAT\ Soff)) \preceq (V\ S),\ ((V\ S) \oplus (V\ C1)) \preceq (C\ (NAT\ MAX)),\ (V\ S) \doteq Old\ (V\ S),\ \prod\ 0.\ ((\Lambda\ [\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ r)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ S)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C1)),\ Neg\ ((Lv\ 0) \doteq C\ (NAT\ C3)),\ (Lv\ 0) \prec (V\ S)]) \supset ((Deref\ (Lv\ 0)) \doteq (Old\ (Deref\ (Lv\ 0)))))]))$$

])

]

1.4 The Verification Condition

generate-code [*term-of*]
vcg = *vcgSALDeep*
prg = *prog*

ML {* *set show-brackets*; *}
ML {* *val vc* = *vcg prog*; *}
ML {* *File.write* (*Path.unpack pvc.txt*) (*Pretty.str-of* (*Sign.pretty-term* (*sign-of* (*the-context* ())) (*term-of-form vc*))); *}
ML {* *val pvc* = (*Pretty.str-of* (*Sign.pretty-term* (*sign-of* (*the-context* ())) (*term-of-form vc*))); *}
ML {* *reset show-brackets* *}

constdefs
vc::*SALform*
vc ≡ (Λ [((Λ [(*Pc* ≐ (*C* (*POS* (*0*, *0*)))), (∏ *0*. ((*Deref* (*Lv 0*)) ≐ (*C ILLEGAL*))), (∏ *0*. ((*Old* (*Deref* (*Lv 0*))) ≐ (*C ILLEGAL*))), (*Rp* ≐ (*C* (*POS* (*0*, *0*)))), (*Tm* ≐ (*C* (*NAT 0*)))]) ⊃ (Λ [((*C* (*NAT 0*)) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), ((*Pc* ≐ (*C* (*POS* (*0*, *0*)))) ⊃ (Λ [((*C* (*NAT* (*Suc 0*))) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), (((*C* (*POS* (*0*, (*Suc 0*)))) ≐ (*C* (*POS* (*0*, (*Suc 0*))))) ⊃ (Λ [((*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), (((*C* (*POS* (*0*, (*Suc* (*Suc 0*))))) ≐ (*C* (*POS* (*0*, (*Suc* (*Suc 0*)))))) ⊃ (Λ [((*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), (((*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc 0*)))))) ≐ (*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc 0*))))))) ⊃ (Λ [((*C* (*NAT* (*Suc* (*Suc 0*)))) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))),
(((*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc 0*))))))) ≐ (*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))) ⊃ (Λ [((*C* (*NAT* (*Suc* (*Suc* (*Suc 0*))))) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), (((*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))) ≐ (*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*))))))))) ⊃ (Λ [((*C* (*NAT 0*)) ⪯ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))))))))), (((*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*))))))))) ≐ (*C* (*POS* (*0*, (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))) ⊃ (Λ [*T*, (Λ [((*C* (*NAT* (*Suc* (*Suc 0*)))) ≐ (*C* (*NAT* (*Suc* (*Suc 0*))))), ((*C* (*NAT* (*Suc* (*Suc* (*Suc 0*))))) ≐ (*C* (*NAT* (*Suc* (*Suc* (*Suc 0*)))))), ((*C* (*NAT 0*)) ≐ (*C* (*NAT 0*))), ((*C* (*NAT 0*)) ≐ (*C* (*NAT 0*))), ((*C* (*NAT* (*Suc 0*))) ≐ (*C* (*NAT* (*Suc 0*)))), ((*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))) ≐ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))), ((*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))))) ≐ (*C* (*NAT* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*))))))))))))), (∏ *0*. ((Λ [(*Neg* ((*Lv 0*) ≐ (*C* (*NAT* (*Suc 0*))))), (*Neg* ((*Lv 0*) ≐ (*C* (*NAT* (*Suc* (*Suc 0*)))))), (*Neg* ((*Lv 0*) ≐ (*C* (*NAT* (*Suc* (*Suc*

$(Suc\ 0)))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))$,
$(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))$]) $\supset$ ((IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))$ THEN $(C\ (NAT\ 0))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))$ THEN $(C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ 0)))$ THEN $(C\ (NAT\ (Suc\ (Suc\ 0))))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$ THEN $(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))$ THEN $(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))$ THEN $(C\ (NAT\ (Suc\ 0)))$ ELSE (IF $(Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))$ THEN $(C\ (NAT\ 0))$ ELSE $(Deref\ (Lv\ 0)))))))))$ $\doteq$ $(Old\ (Deref\ (Lv\ 0)))))))])])])])])])])])])])$, $(\Lambda\ [((\Lambda\ [(\Lambda\ [T, (\Lambda\ [((V\ (Suc\ 0)) \doteq (C\ (NAT\ (Suc\ (Suc\ 0)))))$, $((V\ (Suc\ (Suc\ 0))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0))))))$, $((V\ (Suc\ (Suc\ (Suc\ 0)))) \doteq (C\ (NAT\ 0)))$, $((V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))) \doteq (C\ (NAT\ 0)))$, $((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (C\ (NAT\ (Suc\ 0))))$,
$((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$, $((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))$, $(\prod 0.\ ((\Lambda\ [(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ 0)))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))$, $(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))$]) $\supset$ $((Deref\ (Lv\ 0)) \doteq (Old\ (Deref\ (Lv\ 0))))))])])$, $(Pc \doteq (C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))])$ $\supset$ $(\Lambda\ [(\Lambda\ [(Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))\ vtype.Nat)$, $(Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))\ vtype.Nat)])$, $(\Lambda\ [(Ty\ (V\ (Suc\ 0))\ vtype.Nat)$, $(Ty\ (V\ (Suc\ (Suc\ 0)))\ vtype.Nat)$, $(Ty\ (V\ (Suc\ (Suc\ (Suc\ 0))))\ vtype.Nat)$, $(((V\ (Suc\ (Suc\ (Suc\ 0)))) \oplus ((V\ (Suc\ 0)) \odot (V\ (Suc\ (Suc\ 0))))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))))$,
$((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (C\ (NAT\ 0)))$, $((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ 0))))$,
$((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$, $(Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))\ vtype.Nat)$, $((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))) \preceq (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$, $(((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \oplus (V\ (Suc\ 0))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))))$,
$((C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))) \doteq (C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))$, $(\prod 0.\ ((Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))) \supset ((IF\ (Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))\ THEN\ (C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))\ ELSE\ (Deref\ (Lv\ 0))) \doteq (Deref\ (Lv\ 0)))))])])])$, $(\Lambda\ [((\Lambda\ [(\Lambda\ [T, ((V\ (Suc\ (Suc\ (Suc\ 0)))) \doteq ((C\ (NAT\ (Suc\ (Suc\ 0)))) \odot (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))])$, $(Pc \doteq (C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))])$ $\supset$ $(\Lambda\ [T, ((V\ (Suc\ (Suc\ (Suc\ 0)))) \doteq ((C\ (NAT$

$(Suc\ (Suc\ 0)))) \odot (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))))]))]), (\Lambda\ [((\Lambda\ [(\Lambda\ [(Ty\ (V$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))\ vtype.Nat), (Ty\ (V\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ vtype.Nat)]),$
$(\Lambda\ [(Ty\ (V\ (Suc\ 0))\ vtype.Nat), (Ty\ (V\ (Suc\ (Suc\ 0)))\ vtype.Nat), (Ty\ (V\ (Suc$
$(Suc\ (Suc\ 0))))\ vtype.Nat), (((V\ (Suc\ (Suc\ (Suc\ 0)))) \oplus ((V\ (Suc\ 0)) \odot (V\ (Suc$
$(Suc\ 0))))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))) \doteq$
$(C\ (NAT\ 0))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (C\ (NAT\ (Suc\ 0)))), ((V$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ 0)))))))), (Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))$
$vtype.Nat), ((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$
$\preceq (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))), (((V\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \oplus (V\ (Suc\ 0))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))))),$
$((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq Rp), (\prod 0.\ ((Neg\ ((Lv\ 0) \doteq (C$
$(NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \supset ((Deref\ (Lv\ 0)) \doteq (Old\ (Deref$
$(Lv\ 0))))))])]),$
$(Pc \doteq (C\ (POS\ ((Suc\ 0),\ 0))))]) \supset (\Lambda\ [(\Lambda\ [(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0)))\ vtype.Nat), (Ty\ (IF\ (V$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc$
$(Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))$
$ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ vtype.Nat)]), ((\Lambda\ [(Ty\ (IF\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0)))\ vtype.Nat),$
$(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ vtype.Nat), ((IF\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0))) \doteq (IF\ (V\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc$
$0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V$
$(Suc\ (Suc\ (Suc\ (Suc\ 0)))))), ((C\ (POS\ ((Suc\ 0), (Suc\ 0)))) \doteq (C\ (POS\ ((Suc$
$0), (Suc\ 0)))))]) \supset (\Lambda\ [(\Lambda\ [((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$0)))))))))) \doteq$
$(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$0)))))))) \doteq Rp), (Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))$
$\doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ THEN\ (Deref\ (V\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$0)))))))) \doteq Rp)]), (\Lambda\ [(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty$
$(Old\ (V\ (Suc\ (Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ 0)))\ vtype.Nat), (Ty$
$(Old\ (V\ (Suc\ (Suc\ 0))))\ vtype.Nat), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ (Deref\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (Old\ ((V$
$(Suc\ (Suc\ (Suc\ 0)))) \oplus ((V\ (Suc\ 0)) \odot (V\ (Suc\ (Suc\ 0)))))), ((IF\ (V\ (Suc\ (Suc$
$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))$

$THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\doteq(C\ (NAT\ 0))),$
$((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))),((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq Rp),(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))),vtype.Nat),((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\preceq(IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))),$
$(((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\oplus(IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0))))\preceq(C\ (NAT\ (Suc\ 0)))))))))))))))))))))),((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\doteq(Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))),(\prod 0.\ ((\Lambda\ [(Neg\ ((Lv\ 0)\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))),(Neg\ ((Lv\ 0)\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))),(Neg\ ((Lv\ 0)\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))),(Neg\ ((Lv\ 0)\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))),((Lv\ 0)\prec(IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))])\supset((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(Lv\ 0)\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (Deref\ (Lv\ 0)))\doteq(Old\ (Deref\ (Lv\ 0))))]])]),$
$((\Lambda\ [(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0)))\ vtype.Nat),(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ vtype.Nat),$
$(Neg\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0)))\doteq(IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\doteq(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))),((C\ (POS\ ((Suc\ 0),$

$(Suc\ 0)))) \doteq (C\ (POS\ ((Suc\ 0), (Suc\ 0))))]) \supset (\Lambda\ [(\Lambda\ [(Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0)))\ vtype.Nat), (Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ vtype.Nat)]), (((C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ 0))))) \doteq (C\ (POS\ ((Suc\ 0),$
$(Suc\ (Suc\ 0)))))) \supset (\Lambda\ [(((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0)))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))))))))), (((C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ 0))))))$
$\doteq (C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ 0)))))) \supset (\Lambda\ [((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \prec (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))))))))), (((C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \doteq (C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \supset (\Lambda\ [T, (\Lambda\ [(Ty\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0))) \ominus (IF\ (V\ (Suc\ (Suc\ (\ Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))\ vtype.Nat), (Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq$
$(C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))))\ vtype.Nat), (Ty\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0)))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ 0)))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ (\ Suc\ 0))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ (Suc\ (Suc\ 0)))))\ vtype.Nat), ((C\ (NAT\ 0)) \prec (Old\ (V\ (Suc\ 0)))), (((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ 0))) \ominus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (Old\ ((V\ (Suc\ 0)) \ominus (C\ (NAT\ (Suc\ 0))))), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (\ Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0)))) \doteq (Old\ (V\ (Suc\ (Suc\ 0))))), (((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq$
$(C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0)))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))))) \doteq (Old\ ((V$

$(Suc\ (Suc\ (Suc\ 0)))) \oplus (V\ (Suc\ (Suc\ 0))))))$, $((((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ 0))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))\ ))) \oplus ((((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ 0))) \ominus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (\ Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \odot (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))))) \preceq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))))))$,
$((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))\ \doteq\ (C\ (NAT\ 0))), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ 0)))), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))), (Ty\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ 0))))\ vtype.Nat), ((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))) \preceq ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \oplus (C\ (NAT\ (Suc\ 0))))),$
$(Ty\ (Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ vtype.Nat),$
$(((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \oplus (C\ (NAT\ (Suc\ 0)))) \doteq (Old\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \oplus (C\ (NAT\ (Suc\ 0))))), ((IF\ (Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \oplus (C\ (NAT\ (Suc\ 0))))\ ELSE\ (IF\ (Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))\ THEN\ ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ 0))))\ THEN\ (Deref\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))\ ELSE\ (V\ (Suc\ (Suc\ 0))))\ ELSE\ (IF$

(Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ≐ (C (NAT (Suc 0))) THEN ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc 0))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc 0))) ⊖ (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (Deref (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))) ≐ Rp), ((((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊕ (C (NAT (Suc 0)))) ⊕ ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc 0))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc 0))) ⊖ (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc 0))))))))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))),

(∏ 0. ((Λ [(Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))), (Neg ((Lv 0) ≐ (C (NAT (Suc 0))))), (Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc 0))))))), (((Lv 0) ⊕ (C (NAT (Suc 0)))) ≺ ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊕ (C (NAT (Suc 0)))))]) ⊃ ((IF (Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊕ (C (NAT (Suc 0)))) ELSE (IF (Lv 0) ≐ (C (NAT (Suc (Suc (Suc 0))))) THEN ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc 0))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc 0))))) ⊕ (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc 0)))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc 0))))) ELSE (IF (Lv 0) ≐ (C (NAT (Suc 0))) THEN ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc 0))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc 0))) ⊖ (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (V (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (Lv 0) THEN (Deref (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (Deref (Lv 0)))))) ≐ (Old (Deref (Lv 0))))))])]))]))]))]))])]),

(Λ [((Λ [(Λ [T, (Λ [(Ty (V (Suc 0)) vtype.Nat), (Ty (V (Suc (Suc 0))) vtype.Nat), (Ty (V (Suc (Suc (Suc 0)))) vtype.Nat), (Ty (Old (V (Suc 0))) vtype.Nat), (Ty (Old (V (Suc (Suc 0)))) vtype.Nat), (Ty (Old (V (Suc (Suc (Suc 0))))) vtype.Nat), ((C (NAT 0)) ≺ (Old (V (Suc 0)))), ((V (Suc 0)) ≐ (Old ((V (Suc

0)) ⊖ (C (NAT (Suc 0)))))), ((V (Suc (Suc 0))) ≐ (Old (V (Suc (Suc 0))))), ((V (Suc (Suc (Suc 0)))) ≐ (Old ((V (Suc (Suc (Suc 0)))) ⊕ (V (Suc (Suc 0)))))), (((V (Suc (Suc (Suc 0)))) ⊕ ((V (Suc 0)) ⊙ (V (Suc (Suc 0))))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))))))))), ((V (Suc (Suc (Suc (Suc 0))))) ≐ (C (NAT 0))), ((V (Suc (Suc (Suc (Suc (Suc 0)))))) ≐ (C (NAT (Suc 0)))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) vtype.Nat), ((C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) ≼ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (Ty (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) vtype.Nat), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (Old ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ⊕ (C (NAT (Suc 0)))))),

(Suc (Suc 0)))))))) vtype.Nat), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) vtype.Nat)]), (Λ [(Ty (V (Suc 0)) vtype.Nat), (Ty (V (Suc (Suc 0))) vtype.Nat), (Ty (V (Suc (Suc (Suc 0)))) vtype.Nat), (((V (Suc (Suc (Suc 0)))) ⊕ ((V (Suc 0)) ⊙ (V (Suc (Suc 0))))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))))))))), ((V (Suc (Suc (Suc (Suc 0))))) ≐ (C (NAT 0))), ((V (Suc (Suc (Suc (Suc (Suc 0)))))) ≐ (C (NAT (Suc 0)))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) vtype.Nat), ((C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) ≼ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ⊕ (V (Suc 0))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))))))))), ((C (POS ((Suc 0), (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ≐ (C (POS ((Suc 0), (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (∏ 0. ((Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊃ ((IF (Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) THEN (C (POS ((Suc 0), (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ELSE (Deref (Lv 0))) ≐ (Deref (Lv 0)))))])))]), (Λ [((Λ [(Λ [(Λ [(Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) vtype.Nat), (Ty (V (Suc (Suc (Suc (Suc (Suc 0)))))) vtype.Nat)]), (Λ [(Ty (V (Suc (Suc (Suc 0)))) vtype.Nat), (Ty (Old (V (Suc (Suc (Suc 0))))) vtype.Nat), (Ty (Old (V (Suc 0))) vtype.Nat), (Ty (Old (V (Suc (Suc 0)))) vtype.Nat), ((V (Suc (Suc (Suc 0)))) ≐ (Old ((V (Suc (Suc (Suc 0)))) ⊕ ((V (Suc 0)) ⊙ (V (Suc (Suc 0))))))), ((V (Suc (Suc (Suc (Suc 0))))) ≐ (C (NAT 0))), ((V (Suc (Suc (Suc (Suc (Suc 0)))))) ≐ (C (NAT (Suc 0)))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) vtype.Nat), ((C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) ≼ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (Ty (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) vtype.Nat),
((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ≐ (Old ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ⊕ (C (NAT (Suc 0)))))), ((Deref (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) ≐ Rp), (((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) ⊕ (V (Suc 0))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc

$0))))))))))))))))))$), ($\prod 0$. ((Λ [(Neg ((Lv 0) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc 0))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc (Suc (Suc 0)))))))), (((Lv 0) $\oplus$ (C (NAT (Suc 0)))) $\prec$ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))]) $\supset$ (($Deref$ (Lv 0)) $\doteq$ (Old ($Deref$ (Lv 0))))))]]), (Pc $\doteq$ (C (POS ((Suc 0), (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))]) $\supset$ (Λ [(Λ [(Ty ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $vtype.Nat$), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $vtype.Nat$)]), (((C (POS ((Suc 0), (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $\doteq$ (C (POS ((Suc 0),
(Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $\supset$ (Λ [(Λ [((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))
$THEN$ (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))
$THEN$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ ($Deref$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$
(V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ Rp), (Ty (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))
$THEN$ (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))
$THEN$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ ($Deref$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$
(V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) Pos)]),
(Λ [(Ty (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) $THEN$ (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))
$THEN$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ ($Deref$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$
(V (Suc (Suc (Suc (Suc (Suc 0)))))) $vtype.Nat$), (Ty (Old (V (Suc (Suc (Suc 0))))) $vtype.Nat$), (Ty (Old (V (Suc 0))) $vtype.Nat$), (Ty (Old (V (Suc (Suc 0)))) $vtype.Nat$), ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT (Suc (Suc (Suc 0))))) $THEN$ (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))
$THEN$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ ($Deref$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ (V (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (Old ((V (Suc (Suc (Suc 0)))) $\oplus$ ((V (Suc 0)) $\odot$ (V (Suc (Suc 0))))))), ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) $THEN$ (IF
((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))
$THEN$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ ($Deref$ ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\ominus$ (V (Suc (Suc (Suc (Suc (Suc 0))))))) $ELSE$ (V (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT 0))), ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc 0))))))) $THEN$ (IF ((V (Suc

$(Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ 0)))), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus$
$(V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))), ((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq Rp), (Ty\ (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN$
$(IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) vtype.Nat), ((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \preceq (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))),$
$(((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))),$
$(((IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ (Deref\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))\ ELSE\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \oplus (IF\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ 0)))\ THEN\ (IF\ ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \ominus (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))\ THEN$

((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))) ELSE (Deref ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (V (Suc 0)))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))))),
((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc 0))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc 0))))))) ELSE (Deref ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ELSE ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc 0))))))) ≐ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (∏ 0. ((Λ [(Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))), (Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))), (Neg ((Lv 0) ≐ (C (NAT (Suc 0))))), (Neg ((Lv 0) ≐ (C (NAT (Suc (Suc (Suc 0)))))))), ((Lv 0) ≺ (IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc 0)))))))
≐
(C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (Deref ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ELSE ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))]) ⊃ ((IF (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (Lv 0) THEN (IF ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (Deref ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ELSE (IF (Lv 0) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) THEN ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ⊖ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ELSE (Deref (Lv 0)))) ≐ (Old (Deref (Lv 0))))))])])])]), (Λ [((Λ [(Λ [(Λ [((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ Rp),
(Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) Pos)]), (Λ [(Ty (V (Suc (Suc (Suc 0)))) vtype.Nat), (Ty (Old (V (Suc (Suc (Suc 0))))) vtype.Nat), (Ty (Old (V (Suc 0))) vtype.Nat), (Ty (Old (V (Suc (Suc 0)))) vtype.Nat), ((V (Suc (Suc (Suc 0)))) ≐ (Old ((V (Suc (Suc (Suc 0)))) ⊕ ((V (Suc 0)) ⊙ (V (Suc (Suc (Suc 0)))))))), ((V (Suc (Suc (Suc (Suc 0))))) ≐ (C (NAT 0))), ((V (Suc (Suc (Suc (Suc (Suc 0)))))) ≐ (C (NAT (Suc 0)))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ≐ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))) ≐ Rp), (Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) vtype.Nat), ((C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) ≼ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))),
(((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) ⊕ (V (Suc 0))) ≼ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) ≐ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))), (∏ 0. ((Λ [(Neg

$((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))),$
$(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))),$
$(Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ 0))))), (Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))))), ((Lv\ 0) \prec (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))])$
$\supset ((Deref\ (Lv\ 0)) \doteq (Old\ (Deref\ (Lv\ 0))))))])]), (\Lambda\ [((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (C\ (POS\ (0, (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))), (Pc \doteq (C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))), (\Lambda\ [((Old\ (V\ (Suc\ 0))) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ 0)))))), ((Old\ (V\ (Suc\ (Suc\ 0)))) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0))))))), ((Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))) \doteq (Old\ (C\ (NAT\ 0)))), ((Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \doteq (Old\ (C\ (NAT\ 0)))), ((Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \doteq (Old\ (C\ (NAT\ Suc\ 0))))), ((Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))), ((Old\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0)))))))))))), (\prod 0.\ ((\Lambda\ [(Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ 0)))))), (Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ 0))))))), (Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0)))))))),$
$(Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))), (Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))), (Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))), (Neg\ ((Old\ (Lv\ 0)) \doteq (Old\ (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))))))])$
$\supset ((Old\ (Deref\ (Lv\ 0))) \doteq (Old\ (Old\ (Deref\ (Lv\ 0))))))])]) \supset (\Lambda\ [T, ((V\ (Suc\ (Suc\ (Suc\ 0)))) \doteq ((C\ (NAT\ (Suc\ (Suc\ 0)))) \odot (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))])), ((\Lambda\ [(\Lambda\ [(\Lambda\ [((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \doteq Rp), (Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))\ Pos)]), (\Lambda\ [(Ty\ (V\ (Suc\ (Suc\ (Suc\ 0))))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ (Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ 0)))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc\ (Suc\ 0))))\ vtype.Nat), ((V\ (Suc\ (Suc\ (Suc\ 0)))) \doteq (Old\ ((V\ (Suc\ (Suc\ (Suc\ 0)))) \oplus ((V\ (Suc\ 0)) \odot (V\ (Suc\ (Suc\ 0))))))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))) \doteq (C\ (NAT\ 0))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))$
$\doteq (C\ (NAT\ (Suc\ 0)))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \doteq (C$
$(NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ 0)))))))) \doteq Rp), (Ty\ (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$
$vtype.Nat), ((C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$
$\preceq (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))),$
$(((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))) \oplus (V\ (Suc\ 0))) \preceq (C$
$(NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc$
$(Suc\ 0))))))))))))))))), ((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))$
$\doteq (Old\ (V\ (Suc\ (\ Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))), (\prod 0.\ ((\Lambda\ [(Neg$
$((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))), (Neg\ ((Lv\ 0) \doteq$
$(C\ (NAT\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))), (Neg\ ((Lv\ 0) \doteq$
$(C\ (NAT\ (Suc\ 0))))), (Neg\ ((Lv\ 0) \doteq (C\ (NAT\ (Suc\ (Suc\ (Suc\ 0))))))), ((Lv\ 0)$
$\prec (V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))))]) \supset ((Deref\ (Lv\ 0))$
$\doteq (Old\ (Deref\ (Lv\ 0))))))])]), (\Lambda\ [((V\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))) \doteq$
$(C\ (POS\ ((Suc\ 0), (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))), (Pc \doteq (C\ (POS$
$((Suc\ 0), (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ (Suc\ 0))))))))), (\Lambda\ [(Ty\ (Old\ (V$
$(Suc\ 0)))\ vtype.Nat), (Ty\ (\ Old\ (V\ (Suc\ (Suc\ 0))))\ vtype.Nat), (Ty\ (Old\ (V\ (Suc$
$(Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty\ (Old\ (Old\ (V\ (Suc\ 0))))\ vtype.Nat), (Ty\ (Old$
$(Old\ (V\ (Suc\ (Suc\ 0)))))\ vtype.Nat), (Ty\ (Old\ (Old\ (V\ (Suc\ (Suc\ (Suc\ 0))))))$
$vtype.Nat), ((Old\ (C\ (NAT\ 0))) \prec (Old\ (Old\ (V\ (Suc\ 0))))), ((Old\ (V\ (Suc\ 0)))$

$\doteq$ (Old (Old ((V (Suc 0)) $\ominus$ (C (NAT (Suc 0))))))), ((Old (V (Suc (Suc 0)))) $\doteq$ (Old (Old (V (Suc (Suc 0)))))),
((Old (V (Suc (Suc (Suc 0))))) $\doteq$ (Old (Old ((V (Suc (Suc (Suc 0)))) $\oplus$ (V (Suc (Suc 0))))))), ((Old ((V (Suc (Suc (Suc 0)))) $\oplus$ ((V (Suc 0)) $\odot$ (V (Suc (Suc 0)))))) $\preceq$ (Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))))), ((Old (V (Suc (Suc (Suc (Suc 0)))))) $\doteq$ (Old (C (NAT 0)))), ((Old (V (Suc (Suc (Suc (Suc (Suc 0))))))) $\doteq$ (Old (C (NAT (Suc 0))))), ((Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))) $\doteq$ (Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))), (Ty (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $vtype.Nat$), ((Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))) $\preceq$ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))), (Ty (Old (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $vtype.Nat$), ((Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) $\doteq$ (Old (Old ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) $\oplus$ (C (NAT (Suc 0))))))), ((Old ($Deref$ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $\doteq$ (Old Rp)), ((Old ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))) $\oplus$ (V (Suc 0)))) $\preceq$ (Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))))),

($\prod$ 0. ((Λ [(Neg ((Old (Lv 0)) $\doteq$ (Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))), (Neg ((Old (Lv 0)) $\doteq$ (Old (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))))), (Neg ((Old (Lv 0)) $\doteq$ (Old (C (NAT (Suc 0)))))), (Neg ((Old (Lv 0)) $\doteq$ (Old (C (NAT (Suc (Suc (Suc 0))))))), ((Old ((Lv 0) $\oplus$ (C (NAT (Suc 0))))) $\prec$ (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))])
$\supset$ ((Old ($Deref$ (Lv 0))) $\doteq$ (Old (Old ($Deref$ (Lv 0)))))])]) $\supset$ (Λ [(Λ [(Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $vtype.Nat$), (Ty (V (Suc (Suc (Suc (Suc (Suc 0)))))) $vtype.Nat$)], (Λ [(Ty (V (Suc (Suc (Suc 0)))) $vtype.Nat$), (Ty (Old (Old (V (Suc (Suc (Suc 0)))))) $vtype.Nat$), (Ty (Old (Old (V (Suc 0)))) $vtype.Nat$), (Ty (Old (Old (V (Suc (Suc 0))))) $vtype.Nat$), ((V (Suc (Suc (Suc 0)))) $\doteq$ (Old (Old ((V (Suc (Suc (Suc 0)))) $\oplus$ ((V (Suc 0)) $\odot$ (V (Suc (Suc 0))))))), ((V (Suc (Suc (Suc (Suc 0))))) $\doteq$ (C (NAT 0))), ((V (Suc (Suc (Suc (Suc (Suc 0)))))) $\doteq$ (C (NAT (Suc 0)))), ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))),
(Ty (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $vtype.Nat$), ((C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $\preceq$ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))), (Ty (Old (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $vtype.Nat$), ((V (Suc
(Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $\doteq$ (Old (Old ((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $\oplus$ (C (NAT (Suc 0))))))), (($Deref$ (Old (Old (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))) $\doteq$ (Old Rp)), (((V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))) $\oplus$ (V (Suc 0))) $\preceq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0)))))))))))))))))))), ($\prod$ 0. ((Λ [(Neg ((Lv 0) $\doteq$ (C (NAT

(Suc (Suc (Suc (Suc (Suc (Suc 0))))))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc 0))))), (Neg ((Lv 0) $\doteq$ (C (NAT (Suc (Suc (Suc 0)))))))), (((Lv 0) $\oplus$ (C (NAT (Suc 0)))) $\prec$ (V (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc (Suc 0))))))))))])
$\supset$ (($Deref$ (Lv 0)) $\doteq$ (Old (Old ($Deref$ (Lv 0)))))])])])])

1.5 Verifying the program

First we ensure that the program is wellformed.

lemma *wf-prog*:
wf prog
apply (*simp add*: *wf-def domC-prog checkPos.simps prog-def Let-def split-def fst-conv snd-conv cmd.simps ret-succs.simps callpoints-def isCall-def anF.simps*)
done

Establish System Invariants

lemma *vc-proof-startup*:
$\bigwedge$ *s*. ⟦ *wf prg*; *s* ∈ *isafeP prg* ⟧ ⟹ *sysinv* (*s*,*prg*) ∧ *sysinv2* (*s*,*prg*) ∧ (∃ *c css*. *cs* (*snd* (*snd s*)) = *c*#*css*)**done**

Then, we prove the verification condition

lemma *nat-mult-mono*:
0 < (*n*::*nat*) ⟹ *x* ≤ *n* ∗ *x*
apply (*induct n*)
apply *simp*
apply *simp*
done

lemma *vc-prog-holds*:
provable prog vc
apply (*simp only*: *provable-def valid-def split-paired-all* | *rule HOLprf* | *rule ballI* | *rule allI* | *rule impI*)+
apply (*rename-tac pn i m e I*)
apply (*cut-tac wf-prog*, *drule vc-proof-startup*,*assumption*, (*erule conjE* | *erule exE*)+, (*simp only*: *fst-conv snd-conv*))

— Now, the prelude is finished. The main proof starts . . .
apply (*simp only*: *vc-def*)

apply (*case-tac prog*,((*pn*,*i*),*m*,*e*) ⊨ (*initF prog*))
apply (*simp add*: *initF-def valid-def form.cases validF-validFs.simps nv-def Let-def split-def fst-conv snd-conv id-lookup-def FiniteMap.del-def*)

— not initF prog

— startzustand
apply (*subst validF-Imp*)
apply (*rule impI*)
apply (*subst validF-And-Cons*)
apply (*rule conjI*)
apply (*simp add*: *valid-def validF-validFs.simps initF-def Let-def split-def fst-conv*

snd-conv nv-def FiniteMap.del-def update-def)
apply (*rule impI*)
apply (*erule conjE* | *erule exE*)+
apply (*case-tac css*)
apply *simp*

apply (*rule allI*)
apply *simp*

apply (*subst validF-And-Cons*)
apply (*rule conjI*)
— (0,7) - - ¿ (1,0)
apply (*simp add*: *validF-validFs.simps form.cases Let-def split-def fst-conv snd-conv lift-def nv-def FiniteMap.del-def*)

apply (*subst validF-And-Cons*)
apply (*rule conjI*)
— (0,8) - - ¿ (0,8)
apply (*simp add*: *validF-validFs.simps form.cases Let-def split-def fst-conv snd-conv lift-def nv-def*)

apply (*subst validF-And-Cons*)
apply (*rule conjI*)
— (1,0) - - ¿ (1,5)
apply (*simp only*: *validF-And-Cons validF-Imp*)
apply (*rule conjI*)
apply (*rule impI*)
apply (*simp add*: *validF-validFs.simps form.cases Let-def split-def fst-conv snd-conv lift-def nv-def FiniteMap.del-def*)
apply (*erule conjE*)+
apply (*case-tac css*)
apply *simp*

— css = a list
apply (*subgoal-tac* (∃ *n8*. *m 8* = *NAT n8*) ∧ (∃ *n3*. *m 3* = *NAT n3*) ∧ (∃ *n2*. *m 2* = *NAT n2*) ∧ (∃ *n1*. *m 1* = *NAT n1*))
prefer *2*
apply (*rule conjI*)
apply (*case-tac m 8*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*rule conjI*)
apply (*case-tac m 3*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*rule conjI*)
apply (*case-tac m 2*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*case-tac m 1*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*subgoal-tac fst* (*snd* (*callstate e*)) *1* = *m 1*)
prefer *2*
apply (*erule-tac x*=*NAT 1* **in** *allE*)
apply (*simp add*: *nat-number update-def Let-def split-def fst-conv snd-conv id-lookup-def*)
apply (*subgoal-tac fst* (*snd* (*callstate e*)) *2* = *m 2*)
prefer *2*
apply (*erule-tac x*=*NAT 2* **in** *allE*)
apply (*simp add*: *nat-number update-def Let-def split-def fst-conv snd-conv id-lookup-def*)
apply (*subgoal-tac fst* (*snd* (*callstate e*)) *3* = *m 3*)
prefer *2*
apply (*erule-tac x*=*NAT 3* **in** *allE*)
apply (*simp add*: *nat-number update-def Let-def split-def fst-conv snd-conv id-lookup-def*)
apply (*subgoal-tac fst* (*snd* (*callstate e*)) *8* = *m 8*)
prefer *2*
apply (*erule-tac x*=*NAT 8* **in** *allE*)
apply (*simp add*: *nat-number update-def Let-def split-def fst-conv snd-conv id-lookup-def*)
apply (*erule conjE* | *erule exE*)+
apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)
apply (*rule conjI*)
apply (*rule impI*)
apply (*rule allI*)
apply (*erule-tac x*=*tv* **in** *allE*)
apply (*rule impI*)
apply (*erule conjE*)+
apply (*case-tac tv*)
apply (*simp add*: *nat-number update-def Let-def*)

apply (*simp add*: *nat-number update-def Let-def*)

apply (*simp add*: *nat-number update-def Let-def*)

apply (*rule impI*)
apply (*rule conjI*)
apply (*subgoal-tac n2* $\leq$ *n1* $*$ *n2*)
prefer *2*
apply (*rule nat-mult-mono*)
apply *assumption*
apply *arith*

apply (*rule conjI*)
apply (*simp add*: *diff-mult-distrib*)

apply (*rule conjI*)
apply (*simp add*: *split-def fst-conv snd-conv Let-def*)

apply (*rule allI*)
apply (*erule-tac x=tv* **in** *allE*)
apply (*rule impI*)
apply (*erule conjE*)+
apply (*case-tac tv*)
apply (*simp add*: *nat-number update-def Let-def*)

apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)

apply (*simp add*: *nat-number update-def Let-def*)

apply (*simp add*: *validF-validFs.simps*)

apply (*subst validF-And-Cons*)
apply (*rule conjI*)
— (1,5) nach (1,0)
apply (*simp only*: *update-def nat-number validF-validFs.simps map.simps form.cases Let-def split-def fst-conv snd-conv lift-def nv-def id-lookup-def eval.simps simp-thms*)
apply (*rule impI*)
apply (*erule conjE*)+
apply (*case-tac css*)
apply *simp*

— css = a list
apply (*subgoal-tac* (∃ *n8*. *m 8* = *NAT n8*) ∧ (∃ *n3*. *m 3* = *NAT n3*) ∧ (∃ *n2*. *m 2* = *NAT n2*) ∧ (∃ *n1*. *m 1* = *NAT n1*))
prefer *2*
apply (*rule conjI*)
apply (*case-tac m 8*)
apply (*simp add*: *nat-number*)
apply (*rule-tac x=nat* **in** *exI*)
apply *assumption*
apply (*erule-tac V=m* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))) = *?P* **in** *thin-rl*)
apply (*simp add*: *nat-number tval.cases*)

apply (*rule conjI*)
apply (*case-tac m 3*)
apply (*simp add*: *nat-number*)
apply (*rule-tac x=nat* **in** *exI*)
apply *assumption*
apply (*erule-tac V=m* (*Suc* (*Suc* (*Suc 0*))) = *?P* **in** *thin-rl*)

apply (*simp add*: *nat-number*)

apply (*rule conjI*)
apply (*case-tac m 2*)
apply (*simp add*: *nat-number*)
apply (*rule-tac x=nat* **in** *exI*)
apply *assumption*
apply (*erule-tac V=m* (*Suc* (*Suc 0*)) = *?P* **in** *thin-rl*)
apply (*simp add*: *nat-number*)

apply (*case-tac m 1*)
apply (*simp add*: *nat-number*)
apply (*rule-tac x=nat* **in** *exI*)
apply *assumption*
apply (*erule-tac V=m* (*Suc 0*) = *?P* **in** *thin-rl*)
apply (*simp add*: *nat-number*)

apply (*erule conjE* | *erule exE*)+
apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)

apply (*subst validF-And-Cons*)
apply (*rule conjI*)
— (1,6) nach (1,8)
apply (*simp only*: *validF-validFs.simps map.simps form.cases Let-def split-def fst-conv snd-conv lift-def nv-def id-lookup-def eval.simps simp-thms tval.cases*)
apply (*rule impI*)
apply (*case-tac css*)
apply (*simp add*: *Let-def*)
— case css = a list
apply (*erule conjE* | *erule exE*)+
apply (*simp add*: *Let-def*)
apply (*rule context-conjI*)
— Wenn S=Soff=9, dann s'a=8, dann m 8 = NAT und m 8 = POS .. Typkonflikt
apply (*case-tac fst* (*snd* (*callstate e*)) (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*case-tac fst* (*snd* (*callstate e*)) (*Suc 0*))
apply (*simp add*: *MAX-def nat-number*)
prefer *2*
apply (*simp add*: *nat-number MAX-def*)
apply (*simp add*: *nat-number MAX-def*)

apply (*case-tac fst* (*snd* (*callstate e*)) (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*)))))))))
apply (*simp add*: *nat-number*)

prefer *2*
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*case-tac nata = Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc* (*Suc 0*))))))))
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)

apply (*case-tac m* (*Suc 0*))
apply (*simp add*: *MAX-def nat-number*)
prefer *2*
apply (*simp add*: *nat-number MAX-def*)
apply (*simp add*: *nat-number MAX-def*)

apply (*rule allI*)
apply (*case-tac tv*)
apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)

prefer *2*
apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)

apply (*simp add*: *nat-number update-def Let-def id-lookup-def*)
apply (*rule impI*)
apply (*erule conjE*)+
apply (*erule-tac x=NAT natc* **in** *allE*)
apply (*simp add*: *lift-def Let-def nv-def nat-number update-def*)

apply (*subst validF-And-Cons-Nil*)
— (1,8) nach (0,8)
apply (*simp only*: *nat-number update-def validF-validFs.simps map.simps form.cases Let-def split-def fst-conv snd-conv lift-def id-lookup-def eval.simps simp-thms*)
apply (*rule conjI*)
apply (*rule impI*)
apply (*erule conjE*)+
apply *simp*

apply (*rule impI*)
apply (*erule conjE*)+
apply (*simp add*: *nv-def*)
apply (*rule conjI*)
apply (*erule conjE*)+
apply (*case-tac fst* (*snd* (*callstate* (*snd* (*snd* (*callstate e*))))) *3*)
apply (*simp add*: *nat-number*)

prefer *2*
apply (*simp add*: *nat-number*)

apply (*simp add*: *nat-number*)
apply (*case-tac fst* (*snd* (*callstate* (*snd* (*snd* (*callstate e*))))) *2*)

apply (*simp add*: *nat-number*)

prefer *2*
apply (*simp add*: *nat-number*)

apply (*simp add*: *nat-number*)
apply (*case-tac fst* (*snd* (*callstate* (*snd* (*snd* (*callstate e*))))) (*Suc 0*))
apply (*simp add*: *nat-number*)

prefer *2*
apply (*simp add*: *nat-number*)

apply (*simp add*: *diff-mult-distrib nat-number*)

apply (*rule conjI*)
apply (*erule conjE*)+
apply (*case-tac fst* (*snd* (*callstate* (*snd* (*snd* (*callstate e*))))) *8*)
apply (*simp add*: *nat-number*)

prefer *2*
apply (*simp add*: *nat-number*)
apply (*simp add*: *nat-number*)
apply (*erule-tac x*=*NAT nat* **in** *allE*)
apply (*simp add*: *nat-number id-lookup-def Let-def*)
apply (*case-tac css*)
apply (*simp add*: *id-lookup-def Let-def*)
apply (*simp add*: *callstate-def Let-def split-def fst-conv snd-conv id-lookup-def*)
apply (*case-tac nat* = *8*)
apply (*simp add*: *nat-number*)

apply (*simp add*: *nat-number*)

apply (*rule allI*)
apply (*simp add*: *nat-number id-lookup-def Let-def lift-def*)
apply (*case-tac tv*)
apply *simp*
prefer *2*
apply *simp*
apply (*erule-tac x*=*tv* **in** *allE*)
apply (*erule-tac x*=*tv* **in** *allE*)
apply (*simp add*: *id-lookup-def Let-def*)
apply (*rule impI*)
apply (*erule conjE*)+
apply (*simp add*: *split-def Let-def*)
done

end