MODULE *pluscal_power3*

EXTENDS *Integers*, *TLC*

CONSTANTS x x is the input

```
--algorithm squareroot{
variables cz = 0, cv = 0, cw = 1, ct = 3, k = 0, r ;
{
l1:  assert k ≥ 0 ∧ cz = k                    * k * k  ∧ cv + ct = 3 * (k + 1) * (k + 1)
     ∧ cz + cv + cw = (k + 1) * (k + 1) * (k + 1) ∧ cv = 3 * k * k  ∧ cw = 3 * k + 1 ;
in: while ( k < x )
        {
l2: assert    k ≥ 0 ∧    k <  x ∧ cz = k * k                      * k  ∧ cv + ct = 3 * (k + 1) * (k + 1)
    ∧ cz + cv + cw = (k + 1) * (k + 1) * (k + 1) ∧ cv = 3 * k * k  ∧ cw = 3 * k + 1 ;
           cz := cz + cv + cw ;
l3: assert    k ≥ 0 ∧    k <  x ∧ cz = (k + 1) * (k + 1) * (k + 1)  ∧ cv + ct = 3 * (k + 1) * (k + 1)
    ∧ cz = (k + 1) * (k + 1) * (k + 1) ∧ cv = 3 * k * k  ∧ cw = 3 * k + 1 ;
         cv := cv + ct ;
l4: assert    k ≥ 0 ∧    k <  x ∧ cz = (k + 1) * (k + 1) * (k + 1)  ∧ cv = 3 * (k + 1) * (k + 1)
    ∧ cz = (k + 1) * (k + 1) * (k + 1)   ∧ cw = 3 * k + 1 ;
         ct := ct + 6 ;
l5: assert    k ≥ 0 ∧    k <  x ∧ cz = (k + 1) * (k + 1) * (k + 1)  ∧ cv = 3 * (k + 1) * (k + 1)
    ∧ cz = (k + 1) * (k + 1) * (k + 1)   ∧ cw = 3 * k + 1 ∧ cv + ct = 3 * (k + 2) * (k + 2) ;
         cw := cw + 3 ;
l6: assert    k ≥ 0 ∧    k <  x ∧ cz = (k + 1)                    * (k + 1) * (k + 1)  ∧ cv = 3 * (k + 1) * (k + 1)
    ∧ cz + cv + cw = (k + 2) * (k + 2) * (k + 2)   ∧ cw = 3 * (k + 1) + 1 ∧ cv + ct = 3 * (k + 2) * (k + 2) ;
           k := k + 1 ;
l7: assert    k ≥ 0 ∧    k ≤  x ∧ cz = (k) * (k) * (k)  ∧ cv = 3 * (k) * (k)
    ∧ cz + cv + cw = (k + 1) * (k + 1) * (k + 1)   ∧ cw = 3 * (k) + 1 ∧ cv + ct = 3 * (k + 1) * (k + 1) ;
          } ;
l8: assert k = x ∧  cz = (k) * (k) * (k)  ∧ cv = 3 * (k) * (k) ∧ cz + cv + cw = (k + 1) * (k + 1) * (k + 1)
    ∧ cw = 3 * (k) + 1 ∧ cv + ct = 3 * (k + 1) * (k + 1) ;
  r := cz ;
l9: assert r = cz ∧ k = x ∧  cz = (k) * (k) * (k)  ∧ cv = 3 * (k) * (k)
    ∧ cz + cv + cw = (k + 1) * (k + 1) * (k + 1) ∧ cw = 3 * (k) + 1 ∧ cv + ct = 3 * (k + 1) * (k + 1) ;
l58: print ⟨x, r⟩ ;
    }
 }
```

BEGIN TRANSLATION

CONSTANT *defaultInitValue*

VARIABLES cz, cv, cw, ct, k, r, pc

$vars \triangleq \langle cz, cv, cw, ct, k, r, pc\rangle$

$Init \triangleq$ Global variables

$\land\ cz\ = 0$
$\land\ cv\ = 0$
$\land\ cw = 1$
$\land\ ct\ = 3$
$\land\ k = 0$
$\land\ r = defaultInitValue$
$\land\ pc =$ "l1"

$l1\ \triangleq\ \land\ pc =$ "l1"
$\land\ Assert($ $k \geq 0 \land cz = k * k * k\ \land cv + ct = 3 * (k+1) * (k+1)$
$\land\ cz + cv + cw = (k+1) * (k+1) * (k+1) \land cv = 3 * k * k\ \land cw = 3 * k + 1,$
"Failure of assertion at line 11, column 6.")
$\land\ pc' =$ "in"
$\land$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k,\ r\rangle$

$in\ \triangleq\ \land\ pc =$ "in"
$\land$ IF $k < x$
THEN $\land\ pc' =$ "l2"
ELSE $\land\ pc' =$ "l8"
$\land$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k,\ r\rangle$

$l2\ \triangleq\ \land\ pc =$ "l2"
$\land\ Assert($ $k \geq 0 \land\ k < x \land cz = k * k * k\ \land cv + ct = 3 * (k+1) * (k+1)$
$\land\ cz + cv + cw = (k+1) * (k+1) * (k+1) \land cv = 3 * k * k\ \land cw = 3 * k + 1,$
"Failure of assertion at line 15, column 5.")
$\land\ cz' = cz + cv + cw$
$\land\ pc' =$ "l3"
$\land$ UNCHANGED $\langle cv,\ cw,\ ct,\ k,\ r\rangle$

$l3\ \triangleq\ \land\ pc =$ "l3"
$\land\ Assert($ $k \geq 0 \land\ k < x \land cz = (k+1) * (k+1) * (k+1)\ \land cv + ct = 3 * (k+1) * (k+1)$
$\land\ cz = (k+1) * (k+1) * (k+1) \land cv = 3 * k * k\ \land cw = 3 * k + 1,$
"Failure of assertion at line 18, column 5.")
$\land\ cv' = cv + ct$
$\land\ pc' =$ "l4"
$\land$ UNCHANGED $\langle cz,\ cw,\ ct,\ k,\ r\rangle$

$l4\ \triangleq\ \land\ pc =$ "l4"
$\land\ Assert($ $k \geq 0 \land\ k < x \land cz = (k+1) * (k+1) * (k+1)\ \land cv = 3 * (k+1) * (k+1)$
$\land\ cz = (k+1) * (k+1) * (k+1)\ \land cw = 3 * k + 1,$
"Failure of assertion at line 21, column 5.")
$\land\ ct' = ct + 6$
$\land\ pc' =$ "l5"
$\land$ UNCHANGED $\langle cz,\ cv,\ cw,\ k,\ r\rangle$

$l5\ \triangleq\ \land\ pc =$ "l5"
$\land\ Assert($ $k \geq 0 \land\ k < x \land cz = (k+1) * (k+1) * (k+1)\ \land cv = 3 * (k+1) * (k+1)$

$\wedge\ cz = (k+1)*(k+1)*(k+1) \quad \wedge\ cw = 3*k+1 \wedge cv + ct = 3*(k+2)*(k+2),$
"Failure of assertion at line 24, column 5.")
$\wedge\ cw' = cw + 3$
$\wedge\ pc' =$ "l6"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ ct,\ k,\ r\rangle$

$l6 \triangleq \wedge\ pc =$ "l6"
$\wedge\ Assert($ $k \geq 0 \wedge \quad k < \ x \wedge cz = (k+1)$ $*(k+1)*(k+1) \ \wedge cv = 3*(k+1)$
$\wedge\ cz + cv + cw = (k+2)*(k+2)*(k+2) \quad \wedge cw = 3*(k+1)+1 \wedge cv + ct = 3*(k+2)*$
"Failure of assertion at line 27, column 5.")
$\wedge\ k' = k + 1$
$\wedge\ pc' =$ "l7"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ r\rangle$

$l7 \triangleq \wedge\ pc =$ "l7"
$\wedge\ Assert($ $k \geq 0 \wedge \quad k \leq \ x \wedge cz = (k)*(k)*(k) \ \wedge cv = 3*(k)*(k)$
$\wedge\ cz + cv + cw = (k+1)*(k+1)*(k+1) \quad \wedge cw = 3*(k)+1 \wedge cv + ct = 3*(k+1)*(k$
"Failure of assertion at line 30, column 5.")
$\wedge\ pc' =$ "in"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k,\ r\rangle$

$l8 \triangleq \wedge\ pc =$ "l8"
$\wedge\ Assert($ $k = x \wedge \ \ cz = (k)*(k)*(k) \ \wedge cv = 3*(k)*(k) \wedge cz + cv + cw = (k+1)*(k+1)*(k$
$\wedge\ cw = 3*(k)+1 \wedge cv + ct = 3*(k+1)*(k+1),$
"Failure of assertion at line 33, column 5.")
$\wedge\ r' = cz$
$\wedge\ pc' =$ "l9"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k\rangle$

$l9 \triangleq \wedge\ pc =$ "l9"
$\wedge\ Assert($ $r = cz \wedge k = x \wedge \ \ cz = (k)*(k)*(k) \ \wedge cv = 3*(k)*(k)$
$\wedge\ cz + cv + cw = (k+1)*(k+1)*(k+1) \wedge cw = 3*(k)+1 \wedge cv + ct = 3*(k+1)*(k+$
"Failure of assertion at line 36, column 5.")
$\wedge\ pc' =$ "l58"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k,\ r\rangle$

$l58 \triangleq \wedge\ pc =$ "l58"
$\wedge\ PrintT(\langle x,\ r\rangle)$
$\wedge\ pc' =$ "Done"
$\wedge$ UNCHANGED $\langle cz,\ cv,\ cw,\ ct,\ k,\ r\rangle$

Allow infinite stuttering to prevent deadlock on termination.
$Terminating \triangleq pc =$ "Done" $\wedge$ UNCHANGED $vars$

$Next \triangleq l1 \vee in \vee l2 \vee l3 \vee l4 \vee l5 \vee l6 \vee l7 \vee l8 \vee l9 \vee l58$
$\vee\ Terminating$

$Spec \triangleq Init \wedge \Box[Next]_{vars}$

$Termination \triangleq \Diamond(pc = \text{“Done”})$

END TRANSLATION

$MAX \triangleq 32768(*16 \text{ bits } *)$
$D \triangleq 0 \,..\, MAX$
$(* x \leq 32768 *)\ DD(X) \triangleq (X \# defaultInitValue \Rightarrow X \in D)\ init(X) \triangleq (X \# defaultInitValue)$

$Safety\_absence \triangleq DD(y1) \wedge DD(y2) \wedge DD(y3) \wedge DD(z)$

$Safety\_partialcorrectness \triangleq pc = \text{“Done”} \Rightarrow z * z \leq x \wedge x < (z+1) * (z+1)$

$Inv \triangleq$
$\wedge\ pc \in \{\text{“}l0\text{”}, \text{“l1”}, \text{“l2”}, \text{“l3”}, \text{“l4”}, \text{“Done”}, \text{“l5”}, \text{“}l33\text{”}\}$
$\wedge\ (init(y1) \wedge init(y2) \wedge init(y3) \wedge init(z)) \Rightarrow (y1 \in Nat \wedge y2 \in Nat \wedge y3 \in Nat \wedge z \in Nat)$
$\wedge\ pc = \text{“}l0\text{”} \Rightarrow (y1 = defaultInitValue \wedge y2 = defaultInitValue \wedge y3 = defaultInitValue \wedge z = defaultInitValue)$
$\wedge\ pc = \text{“l1”} \Rightarrow y2 = (y1+1) * (y1+1) \wedge y3 = 2 * y1 + 1 \wedge y1 * y1 \leq x$
$\wedge\ pc = \text{“l2”} \Rightarrow y2 = (y1+1) * (y1+1) \wedge y3 = 2 * y1 + 1 \wedge y2 \leq x$
$\wedge\ pc = \text{“l3”} \Rightarrow y2 = (y1+1) * (y1+1) \wedge y3 = 2 * y1 + 1 \wedge y1 * y1 \leq x$
$\wedge\ pc = \text{“l4”} \Rightarrow y2 = (y1+1) * (y1+1) \wedge y3 = 2 * y1 + 1 \wedge y1 * y1 \leq x \wedge x < y2$
$\wedge\ Safety\_partialcorrectness$

check $\triangleq$
$\wedge\ Safety\_absence$
$\wedge\ Safety\_partialcorrectness$
$\wedge\ Inv$