MODULE *GFX*

EXTENDS *Integers*, *FiniteSets*, *TLAPS*

Module *Integers* defines the standard operators on integers like + and the sets *Int* of integers and *Nat* of natural numbers. (The *TLAPS* prover has built-in knowledge of integers and does not use the definitions in the module.) Module *TLAPS* asserts some theorems and defines some "tactics" for use in proofs.

CONSTANT *Proc*

ASSUME $ProcFinite \triangleq IsFiniteSet(Proc)$

We assume that *Proc* is a finite set, where the operator *IsFiniteSet* is defined in the *FiniteSets* module.

$NUnion(A) \triangleq \text{UNION } \{A[i] : i \in Nat\}$

*****

```
--algorithm GFX {
   variables A1 = [i ∈ Nat ↦ {}], result = [i ∈ Proc ↦ {}] ;
   process ( Pr ∈ Proc )
     variables known = {self}, notKnown = {} ;
     { a: known := known ∪ NUnion(A1) ;
          notKnown := {i ∈ 0 .. (Cardinality(known)) : known ≠ A1[i]} ;
          if ( notKnown ≠ {} )
            { b: with ( i ∈ notKnown ) { A1[i] := known } ;
                 goto a
            }
          else { result[self] := known }
     }
 }
```

****

BEGIN TRANSLATION

VARIABLES *A1*, *result*, *pc*, *known*, *notKnown*

$vars \triangleq \langle A1, result, pc, known, notKnown \rangle$

$ProcSet \triangleq (Proc)$

$Init \triangleq$ Global variables

$\land A1 = [i \in Nat \mapsto \{\}]$

$\land result = [i \in Proc \mapsto \{\}]$

Process *Pr*

$\land known = [self \in Proc \mapsto \{self\}]$

$\land notKnown = [self \in Proc \mapsto \{\}]$

$\land pc = [self \in ProcSet \mapsto \text{"a"}]$

$a(self) \triangleq \land pc[self] = \text{"a"}$

$\land known' = [known \text{ EXCEPT } ![self] = known[self] \cup NUnion(A1)]$

$\land notKnown' = [notKnown \text{ EXCEPT } ![self] = \{i \in 0 \,..\, (Cardinality(known'[self])) : known'[self] \neq$

$$
\begin{array}{l}
\land \text{IF } \mathit{notKnown}'[\mathit{self}] \neq \{\} \\
\quad \text{THEN } \land \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“b”}] \\
\quad \phantom{\text{THEN }} \land \text{UNCHANGED } \mathit{result} \\
\quad \text{ELSE } \land \mathit{result}' = [\mathit{result} \text{ EXCEPT } ![\mathit{self}] = \mathit{known}'[\mathit{self}]] \\
\quad \phantom{\text{ELSE }} \land \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“Done”}] \\
\land A1' = A1
\end{array}
$$

$$
\begin{array}{lcl}
b(\mathit{self}) & \triangleq & \land \mathit{pc}[\mathit{self}] = \text{“b”} \\
& & \land \exists\, i \in \mathit{notKnown}[\mathit{self}] : \\
& & \quad A1' = [A1 \text{ EXCEPT } ![i] = \mathit{known}[\mathit{self}]] \\
& & \land \mathit{pc}' = [\mathit{pc} \text{ EXCEPT } ![\mathit{self}] = \text{“a”}] \\
& & \land \text{UNCHANGED } \langle \mathit{result}, \mathit{known}, \mathit{notKnown} \rangle
\end{array}
$$

$$\mathit{Pr}(\mathit{self}) \triangleq a(\mathit{self}) \lor b(\mathit{self})$$

$$
\begin{array}{lcl}
\mathit{Next} & \triangleq & (\exists\, \mathit{self} \in \mathit{Proc} : \mathit{Pr}(\mathit{self})) \\
& & \lor \quad \text{Disjunct to prevent deadlock on termination} \\
& & \quad ((\forall\, \mathit{self} \in \mathit{ProcSet} : \mathit{pc}[\mathit{self}] = \text{“Done”}) \land \text{UNCHANGED } \mathit{vars})
\end{array}
$$

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

$$\mathit{Termination} \triangleq \Diamond(\forall\, \mathit{self} \in \mathit{ProcSet} : \mathit{pc}[\mathit{self}] = \text{“Done”})$$

END TRANSLATION

$$\mathit{snapshot} \triangleq \text{UNION } \{A1[i] : i \in \mathit{Nat}\}$$

This definition was used in an earlier version of the algorithm, and since the proof of this version was adapted from the proof of the earlier one, references to it are still present in the proof.

The type-correctness invariant.

$$
\begin{array}{lcl}
\mathit{TypeOK} & \triangleq & \land A1 \in [\mathit{Nat} \to \text{SUBSET } \mathit{Proc}] \\
& & \land \mathit{result} \in [\mathit{Proc} \to \text{SUBSET } \mathit{Proc}] \\
& & \land \mathit{pc} \in [\mathit{Proc} \to \{\text{“a”}, \text{“b”}, \text{“Done”}\}] \\
& & \land \mathit{known} \in [\mathit{Proc} \to \text{SUBSET } \mathit{Proc}] \\
& & \land \mathit{notKnown} \in [\mathit{Proc} \to \text{SUBSET } \mathit{Nat}] \\
& & \land \forall\, p \in \mathit{Proc} : (\mathit{pc}[p] = \text{“b”}) \Rightarrow (\mathit{notKnown}[p] \neq \{\})
\end{array}
$$

$$\mathit{Done}(i) \triangleq \mathit{result}[i] \neq \{\}$$

The invariance of the following predicate captures the correctnes of the algorithm and is the key to proving that it the algorithm implements/refines its spec.

$$
\begin{array}{l}
\mathit{GFXCorrect} \triangleq \forall\, i, j \in \mathit{Proc} : \\
\qquad \land \mathit{Done}(i) \land \mathit{Done}(j) \\
\qquad \land \mathit{Cardinality}(\mathit{result}[i]) = \mathit{Cardinality}(\mathit{result}[j]) \\
\qquad \Rightarrow (\mathit{result}[i] = \mathit{result}[j])
\end{array}
$$

We now define $PA1$ to be the set of all values that $A1$ could assume if some subset of processes that are ready to write wrote.

$$NotAProc \triangleq \text{CHOOSE } n : n \notin Proc$$

An arbitrary value that is not a process.

$$ReadyToWrite(i, p) \triangleq \begin{array}{l} \land\ pc[p] = \text{"b"} \\ \land\ i \in notKnown[p] \end{array}$$

True iff process p could write $known[p]$ to $A1[i]$ in its next step.

$$\begin{array}{l} WriterAssignment \triangleq \{f \in [Nat \rightarrow Proc \cup \{NotAProc\}] : \\ \quad \forall\, i \in Nat : \\ \quad\quad (f[i] \in Proc) \Rightarrow \land\ ReadyToWrite(i, f[i]) \\ \quad\quad\quad\quad\quad\quad\quad\quad\ \land\ \forall\, j \in Nat \setminus \{i\} : \\ \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad f[j] \neq f[i]\} \end{array}$$

The set of functions f that assigns to each Nat i either a unique process that is ready to write i, or $NotAProc$.

$$PV(wa) \triangleq [i \in Nat \mapsto \text{IF } wa[i] = NotAProc \text{ THEN } A1[i] \text{ ELSE } known[wa[i]]]$$

$$PA1 \triangleq \{PV(wa) : wa \in WriterAssignment\}$$

We now complete the definition of the inductive invariant Inv.

$InvB$ is an uninteresting part of the invariant that asserts properties which are easily seen to be true by examining the code.

$$\begin{array}{l} InvB \triangleq \land\ \forall\, i \in Nat : (A1[i] \neq \{\}) \Rightarrow (Cardinality(A1[i]) \geq i) \\ \quad\quad\quad\ \land\ \forall\, p \in Proc : \\ \quad\quad\quad\quad\quad \land\ (pc[p] = \text{"b"}) \Rightarrow \forall\, i \in notKnown[p] : i \leq Cardinality(known[p]) \\ \quad\quad\quad\quad\quad \land\ p \in known[p] \\ \quad\quad\quad\quad\quad \land\ (result[p] \neq \{\}) \equiv (pc[p] = \text{"Done"}) \\ \quad\quad\quad\quad\quad \land\ (result[p] \neq \{\}) \Rightarrow (result[p] = known[p]) \end{array}$$

$InvC$ is the interesting part of the inductive invariant that captures the essence of the algorithm.

$$\begin{array}{l} InvC \triangleq \forall\, p \in Proc : \\ \quad\quad \text{LET } S \triangleq result[p] \\ \quad\quad\quad\quad\ k \triangleq Cardinality(S) \\ \quad\quad \text{IN}\quad k > 0 \Rightarrow \forall\, P \in PA1 : \\ \quad\quad\quad\quad\quad\quad \lor\ Cardinality(\text{UNION } \{P[i] : i \in Nat\}) > k \\ \quad\quad\quad\quad\quad\quad \lor\ S \subseteq \text{UNION } \{P[i] : i \in Nat\} \end{array}$$

Inv is the complete inductive invariant. Its invariance trivially implies the invariance of $GFXCorrect$.

$$Inv \triangleq TypeOK \land InvB \land InvC \land GFXCorrect$$

When we have a library of useful theorems about finite sets, we should be able to use it to prove the following simple results that are needed for the proof. Now, we just assume them. The results are all obvious, and they have been checked by *TLC* to make sure there are no silly errors in these TLA+ versions.

THEOREM $EmptySetCardinality \triangleq Cardinality(\{\}) = 0$
PROOF OMITTED

THEOREM $PositiveCardinalityImpliesNonEmpty \triangleq$
$\forall S : Cardinality(S) \in Nat \land Cardinality(S) > 0 \Rightarrow S \neq \{\}$
$\langle 1 \rangle$ SUFFICES ASSUME NEW S, $Cardinality(S) \in Nat$, $Cardinality(S) > 0$, $S = \{\}$
PROVE FALSE
OBVIOUS
$\langle 1 \rangle$ QED
BY $EmptySetCardinality$

THEOREM $NonEmptySetCardinality \triangleq$
$\forall S : IsFiniteSet(S) \land S \neq \{\} \Rightarrow (Cardinality(S) > 0)$
PROOF OMITTED

THEOREM $SingletonCardinalty \triangleq \forall x : Cardinality(\{x\}) = 1$
PROOF OMITTED

THEOREM $SubsetFinite \triangleq$
$\forall S : IsFiniteSet(S) \Rightarrow \forall T \in \text{SUBSET } S : IsFiniteSet(T)$
PROOF OMITTED

THEOREM $CardType \triangleq \forall S : IsFiniteSet(S) \Rightarrow Cardinality(S) \in Nat$
PROOF OMITTED

THEOREM $SubsetCardinality \triangleq$
$\forall T : IsFiniteSet(T) \Rightarrow \forall S \in \text{SUBSET } T :$
$(S \neq T) \Rightarrow (Cardinality(S) < Cardinality(T))$
PROOF OMITTED

THEOREM $SubsetCardinality2 \triangleq$
$\forall T : IsFiniteSet(T) \Rightarrow$
$\forall S \in \text{SUBSET } T : (Cardinality(S) \leq Cardinality(T))$
PROOF OMITTED

THEOREM $IntervalFinite \triangleq \forall i, j \in Int : IsFiniteSet(i \,..\, j)$
PROOF OMITTED

THEOREM $IntervalCardinality \triangleq$
$\forall i, j \in Int : i \leq j \Rightarrow Cardinality(i \,..\, j) = j - i + 1$

THEOREM $PigeonHolePrinciple \triangleq$
$\forall S, T :$
$\land IsFiniteSet(S) \land IsFiniteSet(T)$

$\wedge\ \mathit{Cardinality}(T) < \mathit{Cardinality}(S)$
$\Rightarrow \forall f \in [S \rightarrow T] :$
$\qquad \exists x, y \in S : (x \neq y) \wedge (f[x] = f[y])$

PROOF OMITTED

The following is a simple corollary of Theorem *PigeonHolePrinciple*,

COROLLARY $\mathit{InjectionCardinality} \triangleq$
$\forall S, T, f :$
$\quad \wedge\ \mathit{IsFiniteSet}(S) \wedge \mathit{IsFiniteSet}(T)$
$\quad \wedge\ f \in [S \rightarrow T]$
$\quad \wedge\ \forall x, y \in S : x \neq y \Rightarrow f[x] \neq f[y]$
$\quad \Rightarrow \mathit{Cardinality}(S) \leq \mathit{Cardinality}(T)$

BY $\mathit{PigeonHolePrinciple}, \mathit{CardType}, \mathit{SMT}$

The theorems above were checked for silly mistakes by having *TLC* check that with this definition, *Test* equals ⟨TRUE, . . . , TRUE⟩.

$\mathit{Test} \triangleq$
$\langle$
$\quad \mathit{EmptySetCardinality},$
$\quad \forall S \in \text{SUBSET } (0 \,..\, 3) : \mathit{NonEmptySetCardinality}!(S),$
$\quad \mathit{SingletonCardinalty}!(\text{“abc”}),$
$\quad \forall S \in \text{SUBSET } (0 \,..\, 3) : \mathit{SubsetFinite}!(S),$
$\quad \forall S \in \text{SUBSET } (0 \,..\, 3) : \mathit{CardType}!(S),$
$\quad \forall T \in \text{SUBSET } (0 \,..\, 3) : \mathit{SubsetCardinality}!(T),$
$\quad \forall T \in \text{SUBSET } (0 \,..\, 3) : \mathit{SubsetCardinality2}!(T),$
$\quad \forall i, j \in ((-4) \,..\, 4) : \mathit{IntervalFinite}!(i, j),$
$\quad \forall i, j \in ((-4) \,..\, 4) : \mathit{IntervalCardinality}!(i, j),$
$\quad \forall S, T \in \text{SUBSET } (0 \,..\, 3) : \mathit{PigeonHolePrinciple}!(S, T)$
$\rangle$

LEMMA $\mathit{NotAProcProp} \triangleq \mathit{NotAProc} \notin \mathit{Proc}$
BY $\mathit{NoSetContainsEverything}$ DEF $\mathit{NotAProc}$

USE DEF NUnion

The following theorem asserts the invariance of *Inv*. This obviously implies the invariance of *GFXCorrect*.

THEOREM $\mathit{Invariance} \triangleq \mathit{Spec} \Rightarrow \Box \mathit{Inv}$
$\langle 1 \rangle 1.\ \mathit{Init} \Rightarrow \mathit{Inv}$
$\quad \langle 2 \rangle$ USE DEF $\mathit{Init}, \mathit{Inv}, \mathit{TypeOK}, \mathit{ProcSet}, \mathit{ReadyToWrite}, \mathit{WriterAssignment}, \mathit{PV}, \mathit{PA1}$
$\quad \langle 2 \rangle 1.\ \mathit{Init} \Rightarrow \mathit{TypeOK}$
$\qquad$ BY SMT
$\quad \langle 2 \rangle 2.\ \mathit{Init} \Rightarrow \mathit{InvB}$
$\qquad \langle 3 \rangle 1.$ ASSUME Init
$\qquad\qquad$ PROVE $\mathit{InvB}!1$
$\qquad\quad$ BY $\langle 3 \rangle 1, \mathit{EmptySetCardinality}$, *SingletonCardinalty*, SMT DEF InvB

$\langle 3\rangle 2.$ ASSUME *Init*
PROVE $InvB!2$
$\langle 4\rangle 1.$ $\forall\, p \in Proc : Cardinality(known[p]) = 1$
BY $\langle 3\rangle 2$, *SingletonCardinalty* *SMT* fails on this
$\langle 4\rangle 2.$ QED
BY $\langle 3\rangle 2$, $\langle 4\rangle 1$, *SMT* DEF *InvB*
$\langle 3\rangle 3.$ QED
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$ DEF *InvB*
$\langle 2\rangle 3.$ $Init \Rightarrow InvC$
$\langle 3\rangle$ SUFFICES ASSUME *Init*, NEW $p \in Proc$
PROVE $\neg(Cardinality(result[p]) > 0)$
BY DEF *InvC*
$\langle 3\rangle$ QED
BY *EmptySetCardinality*, *SMT*
$\langle 2\rangle 4.$ $Init \Rightarrow GFXCorrect$
BY *SMT* DEF *GFXCorrect*
$\langle 2\rangle 5.$ QED
BY $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$

$\langle 1\rangle 2.$ $Inv \land [Next]_{vars} \Rightarrow Inv'$
$\langle 2\rangle 1.$ $Inv \land$ UNCHANGED $vars \Rightarrow Inv'$
$\langle 3\rangle$ SUFFICES ASSUME *Inv*, $vars' = vars$
PROVE Inv'
OBVIOUS
$\langle 3\rangle 1.$ $TypeOK'$
BY *SMT* DEF *Inv*, *TypeOK*, *ProcSet*, *ReadyToWrite*,
WriterAssignment, *PA1*, *PV*, *vars*
$\langle 3\rangle 2.$ $InvB'$
BY *SMT* DEF *Inv*, *TypeOK*, *InvB*, *PA1*, *PV*, *vars*
$\langle 3\rangle 3.$ $InvC'$
$\langle 4\rangle$ $WriterAssignment' = WriterAssignment$
BY DEF *WriterAssignment*, *ReadyToWrite*, *vars*
$\langle 4\rangle$ QED
BY DEF *Inv*, *InvC*, *PA1*, *PV*, *vars* *SMT* Failed
$\langle 3\rangle 4.$ $GFXCorrect'$
BY DEF *Inv*, *GFXCorrect*, *Done*, *vars*
$\langle 3\rangle 5.$ QED
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$ DEF *Inv*
$\langle 2\rangle 2.$ $Inv \land Next \Rightarrow Inv'$
$\langle 3\rangle$ SUFFICES ASSUME *Inv*, *Next*
PROVE Inv'
OBVIOUS
$\langle 3\rangle 1.$ $IsFiniteSet(snapshot) \land (snapshot \subseteq Proc)$
BY ONLY *TypeOK*, *ProcFinite*, *SubsetFinite*, *SMT* DEF *Inv*, *TypeOK*, *snapshot*
$\langle 3\rangle 2.$ $\forall\, p \in Proc : Cardinality(known[p]) \in Nat$

BY $ProcFinite$, $SubsetFinite$, $CardType$, SMT DEF Inv, $TypeOK$

$\langle 3\rangle 3.$ $TypeOK'$

$\langle 4\rangle$ USE DEF Inv, $TypeOK$

$\langle 4\rangle 1.$ ASSUME NEW $p \in Proc$, $a(p)$
PROVE $TypeOK'$

BY $\langle 4\rangle 1$, $\langle 3\rangle 1$, $CardType$, $ProcFinite$, $SubsetFinite$, $SMTT(120)$ DEF a

$\langle 4\rangle 2.$ ASSUME NEW $p \in Proc$, $b(p)$
PROVE $TypeOK'$

BY $\langle 4\rangle 2$, SMT DEF b

$\langle 4\rangle 3$ QED

BY $\langle 2\rangle 1$, $\langle 4\rangle 1$, $\langle 4\rangle 2$ DEF $Next$, Pr, $ProcSet$

$\langle 3\rangle$ USE DEF Inv, $ProcSet$, $ReadyToWrite$, $PotentialValues$, $PA1$

$\langle 3\rangle 4.$ $IsFiniteSet(snapshot') \wedge (snapshot' \subseteq Proc)$

$\langle 4\rangle 1.$ $snapshot' \subseteq Proc$

BY $\langle 3\rangle 3$, $ProcFinite$, $SubsetFinite$, SMT DEF Inv, $TypeOK$, $snapshot$

$\langle 4\rangle 2.$ $IsFiniteSet(snapshot')$

BY $\langle 4\rangle 1$, $ProcFinite$, $SubsetFinite$ DEF $snapshot$

$\langle 4\rangle 3.$ QED

BY $\langle 4\rangle 1$, $\langle 4\rangle 2$

$\langle 3\rangle 5.$ $\forall\, p \in Proc : Cardinality(known'[p]) \in Nat$

BY $\langle 3\rangle 3$, $ProcFinite$, $SubsetFinite$, $CardType$, SMT DEF Inv, $TypeOK$

$\langle 3\rangle$ DEFINE $InvCI(q, P) \triangleq InvC!(q)!1!2!(P)$
$Snapshot(P) \triangleq$ UNION $\{P[i] : i \in Nat\}$

The following step is used only in the proof of $InvC'$ for a $b(p)$ action in the proof of $\langle 3\rangle 8$. It could probably also be used to simplify the proof of one or more cases in the proof of $InvC'$ for an $a(p)$ action.

$\langle 3\rangle 6.$ ASSUME NEW $p \in Proc$, NEW $q \in Proc \setminus \{p\}$,
$Cardinality(result'[q]) > 0$,
$$\forall\, qq \in Proc \setminus \{p\} : \begin{array}{l} \wedge\ known'[qq] = known[qq] \\ \wedge\ notKnown'[qq] = notKnown[qq] \\ \wedge\ pc'[qq] = pc[qq] \\ \wedge\ result'[qq] = result[qq], \end{array}$$
$pc'[p] \neq$ "b",
NEW $P \in PA1$
PROVE $InvCI(q, P)'$

$\langle 4\rangle$ DEFINE $S \triangleq result[q]$
$k \triangleq Cardinality(result[q])$

$\langle 4\rangle 1.$ $\wedge\ IsFiniteSet(S')$
$\wedge\ k' \in Nat$

BY $\langle 3\rangle 3$, $ProcFinite$, $SubsetFinite$, $CardType$, SMT DEF $TypeOK$

$\langle 4\rangle 2.$ $S = S' \wedge k = k'$

BY $\langle 3\rangle 6$, SMT DEF $TypeOK$

$\langle 4\rangle 3.$ $\forall\, i \in Nat : \neg ReadyToWrite(i, p)'$

BY $\langle 3\rangle 6$ DEF $ReadyToWrite$

$\langle 4\rangle 4.$ $\forall\, i \in Nat : \{r \in Proc : ReadyToWrite(i, r)'\} \subseteq$

$\{r \in Proc : ReadyToWrite(i, r)\}$

BY ⟨3⟩6, ⟨4⟩3, *SMT* DEF *ReadyToWrite*, *TypeOK*

⟨4⟩5. $result[q] = result'[q]$

BY ⟨3⟩6 DEF *TypeOK*

⟨4⟩6. $InvCI(q, P)$

BY ⟨3⟩6 DEF *InvC*

⟨4⟩7.CASE $S \subseteq Snapshot(P)$

BY ⟨4⟩7, ⟨4⟩2, ⟨4⟩1, ⟨4⟩4, ⟨4⟩3, ⟨3⟩6, *SMT* DEF *TypeOK*

⟨4⟩8.CASE $Cardinality(\text{UNION } \{P[i] : i \in Nat\}) > Cardinality(S)$

BY ⟨4⟩2, ⟨4⟩8

⟨4⟩9. QED

BY ⟨4⟩6, ⟨4⟩7, ⟨4⟩8

⟨3⟩7. ASSUME NEW $p \in Proc$, $a(p)$
PROVE Inv'

⟨4⟩ USE DEF *Inv*

⟨4⟩1. $InvB'$

BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩5, ⟨3⟩7, *SMT* DEF *a*, *TypeOK*, *InvB*

⟨4⟩2. $InvC'$

⟨5⟩ SUFFICES ASSUME NEW $q \in Proc$, $Cardinality(result'[q]) > 0$,
NEW $P \in PA1'$
PROVE $InvCI(q, P)'$

The goal is the body of the $\forall\, p \in Proc$: quantifier with q substituted for p.

BY DEF *InvC*

⟨5⟩ $(pc[p] = \text{“a”}) \land (A1' = A1)$

BY ⟨3⟩7 DEF *a*

⟨5⟩ DEFINE $S \triangleq result[q]$
$k \triangleq Cardinality(result[q])$
$InvA1q(Q) \triangleq \lor\ S \subseteq Snapshot(P)$
$\lor\ Cardinality(\text{UNION } \{Q[i] : i \in Nat\}) > Cardinality(S)$

⟨5⟩1. $\land\ IsFiniteSet(S')$
$\land\ k' \in Nat$

BY ⟨3⟩3 *TypeOK′*, *ProcFinite*, *SubsetFinite*, *CardType*, *SMT* DEF *TypeOK*

⟨5⟩2. $\forall\, r \in Proc$:
$\land\ IsFiniteSet(result[r])$
$\land\ IsFiniteSet(result'[r])$
$\land\ IsFiniteSet(known[r])$
$\land\ IsFiniteSet(known'[r])$
$\land\ Cardinality(result[r]) \in Nat$
$\land\ Cardinality(result'[r]) \in Nat$
$\land\ Cardinality(known[r]) \in Nat$
$\land\ Cardinality(known'[r]) \in Nat$

BY ⟨3⟩3 *TypeOK′*, *ProcFinite*, *SubsetFinite*, *CardType*, *SMT* DEF *TypeOK*

⟨5⟩3.CASE $\land\ known' = [known \text{ EXCEPT } ![p]$
$= known[p] \cup \text{UNION } \{A1[i] : i \in Nat\}]$

$\land\ notKnown' =$
$\quad [notKnown\ \text{EXCEPT}\ ![p] =$
$\qquad \{i \in 0\,..\,(Cardinality(known'[p])) :$
$\qquad\quad known'[p] \neq A1[i]\}]$
$\land\ notKnown'[p] \neq \{\}$
$\land\ pc' = [pc\ \text{EXCEPT}\ ![p] = \text{“b”}]$
$\land\ \text{UNCHANGED}\ result$

$\langle 6\rangle 1.$ ASSUME NEW $Q \in PA1$
PROVE $\land\ IsFiniteSet(Snapshot(Q))$
$\land\ Cardinality(Snapshot(Q)) \in Nat$
$\langle 7\rangle$ PICK $wa \in WriterAssignment : Q = PV(wa)$
BY DEF $PA1$
$\langle 7\rangle 1.\ \forall i \in Nat : wa[i] \neq NotAProc \Rightarrow wa[i] \in Proc$
BY DEF $WriterAssignment$
$\langle 7\rangle 2.\ \forall i \in Nat : PV(wa)[i] \in \text{SUBSET}\ Proc$
BY $\langle 7\rangle 1$, $\langle 3\rangle 3$ $TypeOK'$, SMT DEF PV, $TypeOK$
$\langle 7\rangle 3.$ UNION $\{Q[j] : j \in Nat\} \subseteq Proc$
BY $\langle 7\rangle 2$, SMT
$\langle 7\rangle 4.\ IsFiniteSet(\text{UNION}\ \{Q[j] : j \in Nat\})$
BY $\langle 7\rangle 3$, $ProcFinite$, $SubsetFinite$ SMT failed on this.
$\langle 7\rangle 5.$ QED
BY $\langle 7\rangle 4$, $CardType$, SMT * sm: SMT fails here

$\langle 6\rangle 2.\ A1 \in PA1$
$\langle 7\rangle$ DEFINE $wa \triangleq [i \in Nat \mapsto NotAProc]$
$\langle 7\rangle 1.\ wa \in WriterAssignment$
BY SMT, $NotAProcProp$ DEF $WriterAssignment$
$\langle 7\rangle 2.\ PV(wa) = A1$
BY DEF $TypeOK$, PV
$\langle 7\rangle 3.$ QED
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$ DEF $PA1$

$\langle 6\rangle 3.\ \land\ p \neq q$
$\land\ S = S'$
$\langle 7\rangle 1.\ result'[q] \neq \{\}$
BY $(k' > 0)$, $\langle 5\rangle 1$, $PositiveCardinalityImpliesNonEmpty$
$\langle 7\rangle 2.\ result'[p] = \{\}$
BY $\langle 5\rangle 3$, $\langle 8\rangle 1$,$\langle 4\rangle 1$ $InvB'$, SMT DEF $TypeOK$, $InvB$ sm: triviality check doesn't get $InvB'$
$\langle 7\rangle 3.$ QED
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$, $\langle 5\rangle 3$, SMT DEF $TypeOK$

$\langle 6\rangle 4.\ \land\ IsFiniteSet(\text{UNION}\ \{P[i] : i \in Nat\})$
$\land\ Cardinality(\text{UNION}\ \{P[i] : i \in Nat\}) \in Nat$
$\langle 7\rangle$ PICK $wa \in WriterAssignment' : P = PV(wa)'$
BY DEF $PA1$
$\langle 7\rangle 1.\ \forall i \in Nat : wa[i] \neq NotAProc \Rightarrow wa[i] \in Proc$
BY DEF $WriterAssignment$
$\langle 7\rangle 2.\ \forall i \in Nat : PV(wa)'[i] \in \text{SUBSET}\ Proc$

BY $\langle 7\rangle 1$, $\langle 3\rangle 3$ $TypeOK'$, SMT DEF PV, $TypeOK$

$\langle 7\rangle 3.$ UNION $\{P[j] : j \in Nat\} \subseteq Proc$

BY $\langle 7\rangle 2$, SMT

$\langle 7\rangle 4.$ $IsFiniteSet(\text{UNION } \{P[j] : j \in Nat\})$

BY $\langle 7\rangle 3$, $ProcFinite$, $SubsetFinite$ SMT failed on this.

$\langle 7\rangle 5.$ QED

BY $\langle 7\rangle 4$, $CardType$ SMT fails here

$\langle 6\rangle 5.$CASE $P \in PA1$

$\langle 7\rangle$ $InvCI(q, P)$

BY $\langle 6\rangle 3$, $\langle 6\rangle 5$, $\langle 5\rangle 3$, SMT DEF $InvC$

$\langle 7\rangle$ QED

BY $\langle 5\rangle 3$, $\langle 6\rangle 3$

$\langle 6\rangle 6.$CASE $P \notin PA1$

$\langle 7\rangle 1.$ PICK $j \in Nat : P[j] = known'[p]$

$\langle 8\rangle$ SUFFICES ASSUME $\forall\, i \in Nat : P[i] \neq known'[p]$

PROVE $P \in PA1$

BY $\langle 6\rangle 6$

$\langle 8\rangle$ PICK $wa \in WriterAssignment' : P = PV(wa)'$

BY DEF $PA1$

$\langle 8\rangle 1.$ $\forall\, i \in Nat : wa[i] \neq p$

BY $NotAProcProp$, SMT DEF PV

$\langle 8\rangle 2.$ $\forall\, i \in Nat : PV(wa)' = PV(wa)$

BY $\langle 8\rangle 1$, $\langle 5\rangle 3$, SMT DEF $TypeOK$, PV DEF PV added 31 May 2013

$\langle 8\rangle 3.$ $wa \in WriterAssignment$

$\langle 9\rangle 1.$ ASSUME NEW $i \in Nat$, $wa[i] \in Proc$

PROVE $ReadyToWrite(i, wa[i])$

$\langle 10\rangle 1.$ $ReadyToWrite(i, wa[i])' \Rightarrow ReadyToWrite(i, wa[i])$

BY $\langle 8\rangle 1$, $\langle 5\rangle 3$, SMT DEF $TypeOK$, $ReadyToWrite$

$\langle 10\rangle 2.$ QED

BY $\langle 9\rangle 1$, $\langle 10\rangle 1$, SMT DEF $WriterAssignment$

$\langle 9\rangle 2.$ QED

BY $\langle 9\rangle 1$, SMT DEF $WriterAssignment$

$\langle 8\rangle 4.$ QED

BY $\langle 8\rangle 2$, $\langle 8\rangle 3$, SMT DEF $PA1$

$\langle 7\rangle 2.$ $Snapshot(A1) \subseteq known'[p]$

BY $\langle 5\rangle 3$, SMT DEF $snapshot$, $TypeOK$

$\langle 7\rangle 3.$ $\lor$ $Cardinality(Snapshot(A1)) > k$

$\lor$ $S \subseteq Snapshot(A1)$

BY $\langle 6\rangle 2$, $\langle 6\rangle 3$ DEF $InvC$, $TypeOK$ SMT fails here

$\langle 7\rangle 4.$ $\lor$ $Cardinality(P[j]) > k$

$\lor$ $S \subseteq P[j]$

$\langle 8\rangle 1.$CASE $Cardinality(Snapshot(A1)) > k$

$\langle 9\rangle 1.$ $IsFiniteSet(known[p])$

BY $ProcFinite$, $SubsetFinite$, SMT DEF $TypeOK$

$\langle 9\rangle 2.$ $Cardinality(Snapshot(A1)) \leq Cardinality(known'[p])$

BY $\langle 9\rangle 1$, $\langle 7\rangle 2$, $\langle 5\rangle 2$, *SubsetCardinality2*

$\langle 9\rangle$ $Cardinality(known[p]) \in Nat$

BY $\langle 9\rangle 1$, *CardType*

$\langle 9\rangle$ $Cardinality(Snapshot(A1)) \in Nat$

BY $\langle 6\rangle 1$, $\langle 6\rangle 2$, *CardType*

$\langle 9\rangle$ $k \in Nat$

BY $\langle 6\rangle 3$, $\langle 5\rangle 1$

$\langle 9\rangle 3$. QED

BY $\langle 5\rangle 2$, $\langle 7\rangle 1$, $\langle 9\rangle 2$, $\langle 8\rangle 1$, *SMT*

$\langle 8\rangle 2$.CASE $S \subseteq Snapshot(A1)$

BY $\langle 8\rangle 2$, $\langle 6\rangle 1$, $\langle 6\rangle 2$, $\langle 7\rangle 1$, $\langle 7\rangle 2$, $\langle 7\rangle 3$, *CardType*, *ProcFinite*, *SubsetFinite*, *SubsetCardinality2*, *SMT* DEF *TypeOK*

$\langle 8\rangle 3$. QED

BY $\langle 8\rangle 1$, $\langle 8\rangle 2$, $\langle 7\rangle 3$

$\langle 7\rangle 5$. $P[j] \subseteq Snapshot(P)$

BY $\langle 5\rangle 1$, $\langle 6\rangle 3$ DEF *TypeOK*

$\langle 7\rangle 6$. QED

$\langle 8\rangle 1$.CASE $Cardinality(P[j]) > k$

$\langle 9\rangle$ $\land IsFiniteSet(Snapshot(P))$

$\land Cardinality(Snapshot(P)) \in Nat$

BY $\langle 6\rangle 4$

$\langle 9\rangle$ $\land Cardinality(P[j]) \in Nat$

BY $\langle 7\rangle 1$, $\langle 3\rangle 3$ $TypeOK'$, *ProcFinite*, *SubsetFinite*, *CardType* DEF *TypeOK*

$\langle 9\rangle$ $\land k \in Nat$

BY *ProcFinite*, *SubsetFinite*, *CardType* DEF *TypeOK*

$\langle 9\rangle$ $Cardinality(P[j]) \leq Cardinality(Snapshot(P))$

BY $\langle 7\rangle 5$, $\langle 6\rangle 1$, *SubsetCardinality2*, *SMT*

$\langle 9\rangle$ QED

BY $\langle 8\rangle 1$, $\langle 7\rangle 1$, $\langle 5\rangle 3$, *SMT* DEF *TypeOK*

$\langle 8\rangle 2$.CASE $S \subseteq P[j]$

BY $\langle 8\rangle 2$, $\langle 7\rangle 5$, $\langle 5\rangle 3$, *SMT* DEF *TypeOK*

$\langle 8\rangle 3$. QED

BY $\langle 8\rangle 1$, $\langle 8\rangle 2$, $\langle 7\rangle 4$

$\langle 6\rangle 7$. QED

BY $\langle 6\rangle 5$, $\langle 6\rangle 6$

$\langle 5\rangle 4$.CASE
$$\begin{aligned}
&\land known' = [known \text{ EXCEPT } ![p] \\
&\qquad\qquad = known[p] \cup \text{UNION } \{A1[i] : i \in Nat\}] \\
&\land notKnown' = \\
&\qquad [notKnown \text{ EXCEPT } ![p] = \\
&\qquad\quad \{i \in 0 \,..\, (Cardinality(known'[p])) : \\
&\qquad\qquad known'[p] \neq A1[i]\}] \\
&\land notKnown'[p] = \{\} \\
&\land result' = [result \text{ EXCEPT } ![p] = known'[p]] \\
&\land pc' = [pc \text{ EXCEPT } ![p] = \text{“Done”}]
\end{aligned}$$

$\langle 6\rangle 2$. $PA1' = PA1$

$\langle 7 \rangle 1.$ ASSUME NEW $i \in Nat$, NEW $r \in Proc$
PROVE $ReadyToWrite(i, r)' = ReadyToWrite(i, r)$
BY $\langle 5 \rangle 4$, SMT DEF $ReadyToWrite$, $TypeOK$
$\langle 7 \rangle 2.$ $WriterAssignment' = WriterAssignment$
BY $\langle 7 \rangle 1$, SMT DEF $WriterAssignment$
$\langle 7 \rangle 3.$ ASSUME NEW $wa \in WriterAssignment$, NEW $i \in Nat$,
$wa[i] \neq NotAProc$
PROVE $known'[wa[i]] = known[wa[i]]$
$\langle 8 \rangle$ USE $\langle 7 \rangle 3$
$\langle 8 \rangle 1.$ $ReadyToWrite(i, wa[i])$
BY $NotAProcProp$, SMT DEF $WriterAssignment$
$\langle 8 \rangle 2.$ $wa[i] \neq p$
BY $\langle 5 \rangle 4$, $\langle 8 \rangle 1$, SMT DEF $ReadyToWrite$
$\langle 8 \rangle 3.$ $wa[i] \in Proc$
BY SMT DEF $WriterAssignment$
$\langle 8 \rangle 4.$ QED
BY $\langle 8 \rangle 2$, $\langle 8 \rangle 3$, $\langle 5 \rangle 4$, SMT DEF $TypeOK$
$\langle 7 \rangle 4.$ $A1' = A1$
BY $\langle 5 \rangle 4$
$\langle 7 \rangle 5.$ QED
$\langle 8 \rangle$ SUFFICES ASSUME NEW $wa \in WriterAssignment$,
NEW $i \in Nat$
PROVE $PV(wa)[i] = PV(wa)[i]'$
$\langle 9 \rangle$ ASSUME NEW $wa \in WriterAssignment$
PROVE $\wedge\, PV(wa) = [i \in Nat \mapsto PV(wa)[i]]$
$\wedge\, PV(wa)' = [i \in Nat \mapsto PV(wa)[i]]$
BY DEF PV
$\langle 9 \rangle$ QED
BY $\langle 7 \rangle 2$ DEF $PA1$
$\langle 8 \rangle 1.$CASE $wa[i] = NotAProc$
BY $\langle 8 \rangle 1$, $\langle 7 \rangle 4$ DEF $PA1$, PV
$\langle 8 \rangle 2.$CASE $wa[i] \neq NotAProc$
BY $\langle 8 \rangle 2$, $\langle 7 \rangle 3$ DEF $PA1$, PV
$\langle 8 \rangle 3.$ QED
BY $\langle 8 \rangle 1$, $\langle 8 \rangle 2$
$\langle 6 \rangle 3.$ SUFFICES ASSUME $p = q$
PROVE $InvCI(q, P)'$
$\langle 7 \rangle$ SUFFICES ASSUME $p \neq q$
PROVE $InvCI(q, P)'$
OBVIOUS
$\langle 7 \rangle$ SUFFICES ASSUME $result'[q] \neq \{\}$
PROVE $InvCI(q, P)'$
OBVIOUS
$\langle 7 \rangle$ $Cardinality(result'[q]) > 0$
BY $\langle 5 \rangle 2$, $NonEmptySetCardinality$, SMT

$\langle 7 \rangle$ $result'[q] = result[q]$
BY $\langle 5 \rangle 4$, SMT DEF $TypeOK$
$\langle 7 \rangle$ $InvCI(q, P)$
BY $\langle 6 \rangle 2$, SMT DEF $InvC$
$\langle 7 \rangle$ QED
BY $\langle 6 \rangle 2$, SMT
$\langle 6 \rangle 4.$ $\land \forall i \in 0 \,..\, Cardinality(known'[p]) : known'[p] = A1[i]$
$\land known'[p] = NUnion(A1)$
$\land Cardinality(known'[p]) \geq 0$
$\langle 7 \rangle 1.$ $\forall i \in 0 \,..\, Cardinality(known'[p]) : known'[p] = A1[i]$
$\langle 8 \rangle$ $\land notKnown'[p] = \{i \in 0 \,..\, Cardinality(known'[p]) : known'[p] \neq A1[i]\}$
$\land notKnown'[p] = \{\}$
BY $\langle 5 \rangle 4$, SMT DEF $TypeOK$
$\langle 8 \rangle$ QED
OBVIOUS
$\langle 7 \rangle 2.$ $Cardinality(known'[p]) \geq 0$
BY $\langle 5 \rangle 2$, $\langle 4 \rangle 1$ $InvB'$, $NonEmptySetCardinality$, SMT DEF $InvB$ * sm: triviality check
$\langle 7 \rangle 3.$ $known'[p] = A1[0]$
BY $\langle 5 \rangle 2$, $\langle 7 \rangle 1$, $\langle 7 \rangle 2$, SMT DEF $TypeOK$
$\langle 7 \rangle 4.$ $NUnion(A1) \subseteq known'[p]$
BY $\langle 5 \rangle 4$ DEF $NUnion$, $TypeOK$
$\langle 7 \rangle 5.$ $NUnion(A1) = known'[p]$
BY $\langle 7 \rangle 3$, $\langle 7 \rangle 4$ DEF $NUnion$
$\langle 7 \rangle 6.$ QED
BY $\langle 7 \rangle 1$, $\langle 7 \rangle 2$, $\langle 7 \rangle 5$
$\langle 6 \rangle 5.$CASE $\exists i \in 0 \,..\, Cardinality(known'[p]) : P[i] = A1[i]$
$\langle 7 \rangle 1.$ PICK $i \in 0 \,..\, Cardinality(known'[p]) : P[i] = A1[i]$
BY $\langle 6 \rangle 5$ sm: original proof, certainly a typo – BY $\langle 6 \rangle 4$
$\langle 7 \rangle 2.$ $A1[i] \subseteq NUnion(P)$
BY $\langle 7 \rangle 1$, SMT DEF $NUnion$, $TypeOK$
$\langle 7 \rangle 3.$ $known'[p] \subseteq NUnion(P)$
BY $\langle 7 \rangle 2$, $\langle 6 \rangle 4$
$\langle 7 \rangle 4.$ $result'[p] = known'[p]$
BY $\langle 5 \rangle 4$, SMT DEF $TypeOK$
$\langle 7 \rangle 5.$ QED
BY $\langle 7 \rangle 3$, $\langle 7 \rangle 4$, $\langle 6 \rangle 3$
$\langle 6 \rangle 6.$CASE $\forall i \in 0 \,..\, Cardinality(known'[p]) : P[i] \neq A1[i]$
$\langle 7 \rangle$ PICK $wa \in WriterAssignment : P = PV(wa)$
BY $\langle 6 \rangle 2$ DEF $PA1$
$\langle 7 \rangle 1.$ $\forall i \in 0 \,..\, Cardinality(known'[p]) : \land wa[i] \neq NotAProc$
$\land P[i] = known[wa[i]]$
BY $\langle 6 \rangle 6$, SMT DEF PV
$\langle 7 \rangle 2.$ $\forall i \in 0 \,..\, Cardinality(known'[p]) : \land wa[i] \in Proc$
$\land ReadyToWrite(i, wa[i])$

BY $\langle 7\rangle 1$, *NotAProcProp*, *SMT* DEF *WriterAssignment*

$\langle 7\rangle 3.\ \forall\, i, j \in 0\,..\,Cardinality(known'[p]) : i \neq j \Rightarrow wa[i] \neq wa[j]$

BY $\langle 5\rangle 2$, $\langle 7\rangle 2$, *SMT* DEF *WriterAssignment*

$\langle 7\rangle 4.\ \forall\, i \in 0\,..\,Cardinality(known'[p]) : wa[i] \in P[i]$

BY $\langle 7\rangle 1$, $\langle 7\rangle 2$, *SMT* DEF *InvB*

$\langle 7\rangle 6.\ \land\ IsFiniteSet(\text{UNION}\ \{P[i] : i \in Nat\})$

$\quad\ \land\ Cardinality(\text{UNION}\ \{P[i] : i \in Nat\}) \quad \in Nat$

$\langle 8\rangle 1.\ \forall\, i \in Nat : wa[i] \neq NotAProc \Rightarrow wa[i] \in Proc$

BY DEF *WriterAssignment*

$\langle 8\rangle 2.\ \forall\, i \in Nat : PV(wa)'[i] \in \text{SUBSET}\ Proc$

BY $\langle 8\rangle 1$, $\langle 3\rangle 3$ *TypeOK'*, *SMT* DEF *PV*, *TypeOK*

$\langle 8\rangle 3.\ \text{UNION}\ \{P[j] : j \in Nat\} \subseteq Proc$

$\langle 9\rangle$ SUFFICES ASSUME NEW $j \in Nat$PROVE $P[j] \subseteq Proc$

OBVIOUS

$\langle 9\rangle 1.\ wa[j] = NotAProc \lor wa[j] \in Proc$

BY *SMT* DEF *WriterAssignment*

$\langle 9\rangle 2.$ QED

BY $\langle 9\rangle 1$, *SMT* DEF *TypeOK*, *PV*

$\langle 8\rangle 4.\ IsFiniteSet(\text{UNION}\ \{P[j] : j \in Nat\})$

BY $\langle 8\rangle 3$, *ProcFinite*, *SubsetFinite* *SMT* failed on this.

$\langle 8\rangle 5.$ QED

BY $\langle 8\rangle 4$, *CardType* *SMT* fails here

$\langle 7\rangle 5.\ Cardinality(\text{UNION}\ \{P[i] : i \in Nat\}) \geq Cardinality(known'[p]) + 1$

$\langle 8\rangle$ DEFINE $C \triangleq Cardinality(known'[p])$

$\qquad SS \triangleq 0\,..\,C$

$\qquad TT \triangleq \text{UNION}\ \{P[i] : i \in SS\}$

$\qquad UU \triangleq \text{UNION}\ \{P[i] : i \in Nat\}$

$\qquad f \triangleq [i \in 0\,..\,C \mapsto wa[i]]$

$\langle 8\rangle$ SUFFICES $Cardinality(UU) \geq C + 1$

OBVIOUS

$\langle 8\rangle 1.\ \land\ C \in Nat$

$\quad\ \land\ SS \subseteq Nat$

$\quad\ \land\ IsFiniteSet(SS)$

$\quad\ \land\ Cardinality(SS) = C + 1$

BY $\langle 5\rangle 2$, *IntervalCardinality*, *IntervalFinite*, *SMT*

$\langle 8\rangle 2.\ \land\ TT \subseteq UU$

$\quad\ \land\ IsFiniteSet(UU)$

$\quad\ \land\ IsFiniteSet(TT)$

BY $\langle 7\rangle 6$, $\langle 8\rangle 1$, *SubsetFinite*, *Z3* *SMT* used to work but now fails

$\langle 8\rangle 3.\ f \in [SS \to TT]$

BY $\langle 7\rangle 4$, *SMT*

$\langle 8\rangle 4.\ \forall\, x, y \in SS : (x \neq y) \Rightarrow (f[x] \neq f[y])$

BY $\langle 7\rangle 3$, *SMT*

$\langle 8\rangle$ HIDE DEF *SS*, *TT*, *UU*, *C*, *f*

$\langle 8\rangle 5.\ Cardinality(SS) \leq Cardinality(TT)$

BY ⟨8⟩1, ⟨8⟩2, ⟨8⟩3, ⟨8⟩4, *InjectionCardinality* *SMT* fails here

⟨8⟩6. $\land$ $Cardinality(TT) \leq Cardinality(UU)$
$\land$ $Cardinality(UU) \in Nat$
$\land$ $Cardinality(TT) \in Nat$

BY ⟨8⟩2, *CardType*, *SubsetCardinality*2, *SMT*

⟨8⟩7. QED

BY ⟨8⟩1, ⟨8⟩5, ⟨8⟩6, *SMT*

⟨7⟩7. QED

⟨8⟩ $result'[p] = known'[p]$

BY ⟨5⟩4, *SMT* DEF *TypeOK*

⟨8⟩ QED

BY ⟨5⟩2, ⟨6⟩3, ⟨7⟩5, ⟨7⟩6, *SMT*

⟨6⟩7. QED

BY ⟨6⟩5, ⟨6⟩6

⟨5⟩5. QED

BY ⟨3⟩7, ⟨5⟩3, ⟨5⟩4 DEF a

⟨4⟩3. $GFXCorrect'$

⟨5⟩1.CASE UNCHANGED *result*

This handles the IF / THEN case.

BY ⟨5⟩1, *SMT* DEF *TypeOK*, *GFXCorrect*, *Done*

⟨5⟩2.CASE $\land known' = [known$ EXCEPT $![p] = known[p] \cup$ UNION $\{A1[i] : i \in Nat\}]$
$\land notKnown' =$
$[notKnown$ EXCEPT $![p] =$
$\{i \in 0 \,..\, (Cardinality(known'[p])) :$
$known'[p] \neq A1[i]\}]$
$\land notKnown'[p] = \{\}$
$\land result' = [result$ EXCEPT $![p] = known'[p]]$
$\land pc' = [pc$ EXCEPT $![p] =$ "Done"$]$
$\land A1' = A1$

This is the IF / ELSE case (simplified).

⟨6⟩ SUFFICES ASSUME NEW $q \in Proc$, NEW $r \in Proc$,
$q \neq r$,
$Done(q)' \land Done(r)'$,
$Cardinality(result'[q]) = Cardinality(result'[r])$
PROVE $result'[q] = result'[r]$

BY DEF *GFXCorrect*

⟨6⟩1.CASE $p \notin \{q, r\}$

⟨7⟩1. $\land result'[q] = result[q]$
$\land result'[r] = result[r]$
$\land Done(q)' = Done(q)$
$\land Done(r)' = Done(r)$

BY ⟨5⟩2, ⟨6⟩1 DEF *TypeOK*, *Done*

⟨7⟩2. QED

BY $\langle 7 \rangle 1$, *SMT* DEF *GFXCorrect*, *TypeOK*

$\langle 6 \rangle 2$. CASE $p \in \{q, r\}$

$\langle 7 \rangle$ SUFFICES ASSUME NEW $s \in Proc$,
$p \neq s$,
$Done(p)' \land Done(s)'$,
$Cardinality(result'[p]) = Cardinality(result'[s])$
PROVE $result'[p] = result'[s]$

Zenon or *Isabelle* used to prove this in one step, but on 31 May 2013 it no longer did and the proof had to be decomposed.

$\langle 8 \rangle 1$. CASE $p = q$
BY $\langle 8 \rangle 1$

$\langle 8 \rangle 2$. CASE $p = q$
BY $\langle 8 \rangle 2$

$\langle 8 \rangle 3$. QED
BY $\langle 8 \rangle 1$, $\langle 8 \rangle 2$, $\langle 6 \rangle 2$

$\langle 7 \rangle 1$. $\land\ result'[s] = result[s]$
$\land\ Done(s)$
BY $\langle 5 \rangle 2$ DEF *TypeOK*, *Done*

$\langle 7 \rangle$ DEFINE $S \triangleq result[s]$
$k \triangleq Cardinality(result[s])$

$\langle 7 \rangle 2$. $\land\ k \in Nat$
$\land\ k > 0$
BY $\langle 7 \rangle 1$, *ProcFinite*, *SubsetFinite*, *CardType*, *NonEmptySetCardinality*, *SMT* DEF *TypeOK*, *Do*

$\langle 7 \rangle 3$. $\lor\ S \subseteq Snapshot(A1)$
$\lor\ Cardinality(\text{UNION } \{A1[i] : i \in Nat\}) > Cardinality(S)$

$\langle 8 \rangle 1$. $InvC!(s)!1!2$
BY $\langle 7 \rangle 2$, *InvC* DEF *InvC*

$\langle 8 \rangle 2$. $A1 \in PA1$

$\langle 9 \rangle$ DEFINE $wa \triangleq [i \in Nat \mapsto NotAProc]$

$\langle 9 \rangle 1$. $wa \in WriterAssignment$
BY *SMT*, *NotAProcProp* DEF *WriterAssignment*

$\langle 9 \rangle 2$. $PV(wa) = A1$
BY DEF *TypeOK*, *PV*

$\langle 9 \rangle 3$. QED
BY $\langle 9 \rangle 1$, $\langle 9 \rangle 2$ DEF *PA1*

$\langle 8 \rangle 3$. QED
BY $\langle 8 \rangle 1$, $\langle 8 \rangle 2$

$\langle 7 \rangle 4$. $result'[p] = A1[k]$

$\langle 8 \rangle 1$. $result'[p] = known'[p]$
BY $\langle 5 \rangle 2$, *SMT* DEF *TypeOK*

$\langle 8 \rangle 2$. $k \in 0 \,..\, Cardinality(known'[p])$
BY $\langle 8 \rangle 1$, $\langle 7 \rangle 1$, $\langle 7 \rangle 2$, *SMT*

$\langle 8 \rangle 3$. $\forall\, i \in 0 \,..\, Cardinality(known'[p]) : known'[p] = A1[i]$
BY $\langle 5 \rangle 2$, *SMT* DEF *TypeOK*

$\langle 8\rangle 4.$ $known'[p] = A1[k]$
BY $\langle 8\rangle 2$, $\langle 8\rangle 3$, SMT DEF $TypeOK$
$\langle 8\rangle 5.$ QED
BY $\langle 8\rangle 1$, $\langle 8\rangle 4$
$\langle 7\rangle 5.$ $result'[p] = $ UNION $\{A1[i] : i \in Nat\}$
$\langle 8\rangle 1.$ UNION $\{A1[i] : i \in Nat\} \subseteq result'[p]$
BY $\langle 5\rangle 2$, $\langle 7\rangle 2$, SMT DEF $TypeOK$
$\langle 8\rangle 2.$ $result'[p] \subseteq$ UNION $\{A1[i] : i \in Nat\}$
BY $\langle 7\rangle 2$, $\langle 7\rangle 4$, SMT
$\langle 8\rangle 3.$ QED
BY $\langle 8\rangle 1$, $\langle 8\rangle 2$
$\langle 7\rangle 6.$ $Cardinality($UNION $\{A1[i] : i \in Nat\}) = k$
BY $\langle 7\rangle 1$, $\langle 7\rangle 5$
$\langle 7\rangle 7.$ $S \subseteq result'[p]$
BY $\langle 7\rangle 2$, $\langle 7\rangle 3$, $\langle 7\rangle 5$, $\langle 7\rangle 6$, SMT
$\langle 7\rangle 8.$ $IsFiniteSet(result'[p])$
BY $\langle 3\rangle 3$ $TypeOK'$, $ProcFinite$, $SubsetFinite$, SMT DEF $TypeOK$
$\langle 7\rangle 9.$ $S = result'[p]$
$\langle 8\rangle$ $(Cardinality(S) = k) \land (Cardinality(result'[p]) = k)$
BY $\langle 7\rangle 5$, $\langle 7\rangle 6$ SMT fails here
$\langle 8\rangle$ $\neg(Cardinality(S) < Cardinality(result'[p]))$
BY $\langle 7\rangle 2$, $\langle 7\rangle 5$, $\langle 7\rangle 6$, $\langle 7\rangle 7$, $\langle 7\rangle 8$, $SubsetCardinality$, SMT
$\langle 8\rangle$ QED
BY $\langle 7\rangle 2$, $\langle 7\rangle 5$, $\langle 7\rangle 6$, $\langle 7\rangle 7$, $\langle 7\rangle 8$, $SubsetCardinality$, SMT
$\langle 7\rangle 10.$ QED
BY $\langle 7\rangle 1$, $\langle 7\rangle 9$
$\langle 6\rangle 3.$ QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$
$\langle 5\rangle 3.$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 3\rangle 7$ DEF a
$\langle 4\rangle 4.$ QED
BY $\langle 3\rangle 3$, $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$ DEF Inv
$\langle 3\rangle 8.$ ASSUME NEW $p \in Proc$, $b(p)$
PROVE Inv'
$\langle 4\rangle$ USE $b(p)$ DEF Inv
$\langle 4\rangle 1.$ $InvB'$
$\langle 5\rangle 1.$ $InvB!1'$
BY DEF $TypeOK$, $InvB$, b
$\langle 5\rangle 2.$ $InvB!2'$
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 5$, $\langle 3\rangle 8$,SMT DEF b, $TypeOK$, $InvB$ sm: SMT fails here when given unnecessary facts
$\langle 5\rangle 3.$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$ DEF $InvB$
$\langle 4\rangle 2.$ $InvC'$

Since $b(p)$ just removes p from $ReadyToWrite(nextwr[p])$ and sets $A1[i]$ to $known[p]$, $PA1'$ is a subset of $PA1$. Since no other relevant variables are changed, $InvC!(q)!1!2(P)$ is left unchanged for any P in $PA1$.

$\langle 5\rangle$ SUFFICES ASSUME NEW $q \in Proc$, $Cardinality(result'[q]) > 0$,
NEW $P \in PA1'$
PROVE $InvCI(q, P)'$

The goal is the body of the $\forall p \in Proc :$ quantifier with q substituted for p.

BY DEF $InvC$

$\langle 5\rangle 1.$ $P \in PA1$

The following proof was copied with slight modification from the proof for action d in $RV3$.

$\langle 6\rangle$ PICK $wa \in WriterAssignment' : P = PV(wa)'$
BY DEF $PA1$
$\langle 6\rangle 1.$ ASSUME NEW $i \in Nat$, NEW $qa \in Proc$,
$ReadyToWrite(i, qa)'$
PROVE $ReadyToWrite(i, qa)$
$\langle 7\rangle$ $\wedge\ pc'[qa] = \text{"b"} \Rightarrow pc[qa] = \text{"b"}$
$\wedge\ notKnown' = notKnown$
BY SMT DEF b, $TypeOK$
$\langle 7\rangle$ QED
BY $\langle 6\rangle 1$, SMT DEF $ReadyToWrite$
$\langle 6\rangle 2.$ $wa \in WriterAssignment$
BY $\langle 6\rangle 1$, SMT DEF $WriterAssignment$
$\langle 6\rangle 3.$ PICK $j \in notKnown[p] : A1' = [A1$ EXCEPT $![j] = known[p]]$
BY DEF b
$\langle 6\rangle$ $j \in Nat$
BY DEF $TypeOK$
$\langle 6\rangle 4.$CASE $wa[j] \neq NotAProc$
$\langle 7\rangle 1.$ $PV(wa)' = PV(wa)$
$\langle 8\rangle 1.$ SUFFICES ASSUME NEW $i \in Nat$
PROVE $PV(wa)'[i] = PV(wa)[i]$
BY DEF PV
$\langle 8\rangle 2.$CASE $wa[i] \neq NotAProc$
$\langle 9\rangle$ $known'[wa[i]] = known[wa[i]]$
BY DEF b
$\langle 9\rangle$ QED
BY $\langle 8\rangle 2$ DEF PV
$\langle 8\rangle 3.$CASE $wa[i] = NotAProc$
$\langle 9\rangle$ $i \neq j$
BY $\langle 6\rangle 4$, $\langle 8\rangle 3$
$\langle 9\rangle$ $A1'[i] = A1[i]$
BY $\langle 6\rangle 3$, SMT DEF $TypeOK$
$\langle 9\rangle$ QED
BY $\langle 8\rangle 3$ DEF PV
$\langle 8\rangle 4.$ QED

$\quad\quad\quad$ BY $\langle 8\rangle 2,\ \langle 8\rangle 3$

$\quad$ $\langle 7\rangle 2.$ QED

$\quad\quad$ BY $\langle 6\rangle 2,\ \langle 7\rangle 1$ DEF $PA1$

$\langle 6\rangle 5.$CASE $wa[j] = NotAProc$

$\quad$ $\langle 7\rangle$ DEFINE $za \triangleq [wa \text{ EXCEPT } ![j] = p]$

$\quad$ $\langle 7\rangle 1.$ $za \in WriterAssignment$

$\quad\quad$ $\langle 8\rangle 1.$ $\forall\, i \in Nat : wa[i] \neq p$

$\quad\quad\quad$ $\langle 9\rangle$ $\forall\, i \in Nat : \neg ReadyToWrite(i,\ p)'$

$\quad\quad\quad\quad$ BY SMT DEF $b,\ TypeOK,\ ReadyToWrite$

$\quad\quad\quad$ $\langle 9\rangle$ QED

$\quad\quad\quad\quad$ BY SMT DEF $WriterAssignment$

$\quad\quad$ $\langle 8\rangle 2.$ $ReadyToWrite(j,\ p)$

$\quad\quad\quad$ BY DEF $b,\ ReadyToWrite$

$\quad\quad$ $\langle 8\rangle 3.$ ASSUME NEW $i \in Nat$, NEW $k \in Nat \setminus \{i\},\ wa[i] \in Proc$

$\quad\quad\quad\quad$ PROVE $za[i] \neq za[k]$

$\quad\quad\quad$ $\langle 9\rangle$ $wa \in [Nat \rightarrow Proc \cup \{NotAProc\}]$

$\quad\quad\quad\quad$ BY DEF $WriterAssignment$

$\quad\quad\quad$ $\langle 9\rangle$CASE $j \notin \{i,\ k\}$

$\quad\quad\quad\quad$ $\langle 10\rangle$ $wa[i] \neq wa[k]$

$\quad\quad\quad\quad\quad$ BY $\langle 8\rangle 3,\ SMT$ DEF $WriterAssignment$

$\quad\quad\quad\quad$ $\langle 10\rangle$ $za[i] = wa[i] \land za[k] = wa[k]$

$\quad\quad\quad\quad\quad$ OBVIOUS

$\quad\quad\quad\quad$ $\langle 10\rangle$ QED

$\quad\quad\quad\quad\quad$ BY $\langle 8\rangle 1,\ SMT$

$\quad\quad\quad$ $\langle 9\rangle$CASE $j \in \{i,\ k\}$

$\quad\quad\quad\quad$ BY $\langle 8\rangle 1,\ SMT$

$\quad\quad\quad$ $\langle 9\rangle$ QED

$\quad\quad\quad\quad$ OBVIOUS

$\quad\quad$ $\langle 8\rangle 4.$ QED

$\quad\quad\quad$ BY $\langle 6\rangle 2,\ \langle 8\rangle 2,\ \langle 8\rangle 3,\ SMT$ DEF $WriterAssignment$

$\quad$ $\langle 7\rangle 2.$ $PV(wa)' = PV(za)$

$\quad\quad$ $\langle 8\rangle 1.$ $wa = [k \in Nat \mapsto wa[k]]$

$\quad\quad\quad\quad$ BY DEF $WriterAssignment$

$\quad\quad$ $\langle 8\rangle 2.$ SUFFICES ASSUME NEW $i \in Nat$

$\quad\quad\quad\quad\quad\quad\quad$ PROVE $PV(wa)'[i] = PV(za)[i]$

$\quad\quad\quad$ BY DEF PV

$\quad\quad$ $\langle 8\rangle 3.$CASE $wa[i] \neq NotAProc$

$\quad\quad\quad$ $\langle 9\rangle 1.$ $i \neq j$

$\quad\quad\quad\quad$ BY $\langle 6\rangle 5,\ \langle 8\rangle 3$

$\quad\quad\quad$ $\langle 9\rangle 2.$ $known'[wa[i]] = known[wa[i]]$

$\quad\quad\quad\quad$ BY $\langle 9\rangle 1$ DEF b

$\quad\quad\quad$ $\langle 9\rangle 3.$ $PV(wa)'[i] = known'[wa[i]]$

$\quad\quad\quad\quad$ BY $\langle 8\rangle 3,\ SMT$ DEF PV

$\quad\quad\quad$ $\langle 9\rangle 4.$ $za[i] = wa[i]$

$\quad\quad\quad\quad\quad$ BY $\langle 8\rangle 1,\ \langle 9\rangle 1$

$\langle 9\rangle 5.\ PV(za)[i] = known[wa[i]]$
BY $\langle 9\rangle 4$, $\langle 8\rangle 3$ DEF PV
$\langle 9\rangle 6.$ QED
BY $\langle 9\rangle 2$, $\langle 9\rangle 3$, $\langle 9\rangle 5$
$\langle 8\rangle 4.$CASE $wa[i] = NotAProc$
$\langle 9\rangle 1.$CASE $i \neq j$
$\langle 10\rangle\ A1'[i] = A1[i]$
BY $\langle 9\rangle 1$, $\langle 6\rangle 3$, SMT DEF $TypeOK$
$\langle 10\rangle\ wa[i] = za[i]$
BY $\langle 8\rangle 1$, $\langle 9\rangle 1$
$\langle 10\rangle$ QED
BY $\langle 8\rangle 4$, $\langle 9\rangle 1$, $\langle 6\rangle 1$, SMT DEF PV
$\langle 9\rangle 2.$CASE $i = j$
$\langle 10\rangle 1.\ PV(wa)'[j] = A1[j]'$
BY $\langle 9\rangle 2$, $\langle 8\rangle 4$ DEF PV
$\langle 10\rangle 2.\ za[j] = p$
BY $\langle 8\rangle 1$, $\langle 9\rangle 2$
$\langle 10\rangle 3.\ PV(za)[j] = known[p]$
BY $\langle 10\rangle 2$, $NotAProcProp$, SMT DEF PV
$\langle 10\rangle 4.\ A1'[j] = known[p]$
BY $\langle 6\rangle 3$, SMT DEF $TypeOK$
$\langle 10\rangle$ HIDE DEF za
$\langle 10\rangle 5.$ QED
BY $\langle 9\rangle 2$, $\langle 10\rangle 1$, $\langle 10\rangle 3$, $\langle 10\rangle 4$
$\langle 9\rangle 3.$ QED
BY $\langle 9\rangle 1$, $\langle 9\rangle 2$
$\langle 8\rangle 5.$ QED
BY $\langle 8\rangle 3$, $\langle 8\rangle 4$
$\langle 7\rangle 3.$ QED
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$ DEF $PA1$
$\langle 6\rangle 6.$ QED
BY $\langle 6\rangle 4$, $\langle 6\rangle 5$
$\langle 5\rangle 2.\quad \forall\, qq \in Proc \setminus \{p\} : \begin{array}{l} \land\ known'[qq] = known[qq] \\ \land\ notKnown'[qq] = notKnown[qq] \\ \land\ pc'[qq] = pc[qq] \\ \land\ result'[qq] = result[qq] \end{array}$
BY $\langle 3\rangle 8$, SMT DEF b, $TypeOK$
$\langle 5\rangle 3.\ pc'[p] = \text{“a”}$
BY $\langle 3\rangle 8$, SMT DEF $TypeOK$, b
$\langle 5\rangle 4.\ q \neq p$
$\langle 6\rangle 1.\ Cardinality(result'[q]) \in Nat$
BY $\langle 3\rangle 3$ $TypeOK'$, $ProcFinite$, $SubsetFinite$, $CardType$, SMT DEF $TypeOK$ sm: failure of triviality c
$\langle 6\rangle 2.\ result'[q] \neq \{\}$
BY $\langle 6\rangle 1$, $PositiveCardinalityImpliesNonEmpty$
$\langle 6\rangle 3.\ pc'[q] = \text{“Done”}$

BY $\langle 6\rangle 2$, $\langle 4\rangle 1$ $InvB'$, SMT DEF $TypeOK$, $InvB$ sm: failure of triviality check

$\langle 6\rangle 4.$ QED

BY $\langle 6\rangle 3$, $\langle 5\rangle 3$

$\langle 5\rangle$ HIDE DEF $PA1$, $InvCI$

$\langle 5\rangle 5.$ $InvCI(q, P)'$

BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 5\rangle 3$, $\langle 5\rangle 4$, $\langle 3\rangle 6$, SMT DEF $TypeOK$

$\langle 5\rangle 6.$ QED

BY $\langle 5\rangle 5$ DEF $InvCI$

$\langle 4\rangle 3.$ $GFXCorrect'$

BY $\langle 3\rangle 8$, SMT DEF b, $TypeOK$, $GFXCorrect$, $Done$

$\langle 4\rangle 4.$ QED

BY $\langle 3\rangle 3$, $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$

$\langle 3\rangle$ HIDE DEF Inv

$\langle 3\rangle 9.$ QED

BY $\langle 2\rangle 1$, $\langle 3\rangle 7$, $\langle 3\rangle 8$ DEF $Next$, Pr, $ProcSet$

$\langle 2\rangle 3.$ QED

BY $\langle 2\rangle 1$, $\langle 2\rangle 2$

$\langle 1\rangle 3.$ QED

*** PROOF

This follows from a $\langle 1\rangle 1$, $\langle 1\rangle 2$, and a simple TLA proof rule.

*** OMITTED

The following theorem combined with theorem *Invariance* shows that the algorithm has the desired space complexity.

THEOREM $Inv \Rightarrow \forall\, i \in Nat : \ (i > Cardinality(Proc)) \Rightarrow \ (A1[i] = \{\})$

$\langle 1\rangle$ SUFFICES ASSUME Inv, NEW $i \in Nat$, $i > Cardinality(Proc)$

PROVE $A1[i] = \{\}$

OBVIOUS

$\langle 1\rangle$ SUFFICES $\neg(Cardinality(A1[i]) \geq i)$

BY DEF Inv, $InvB$

$\langle 1\rangle 1.$ $A1[i] \subseteq Proc$

BY DEF Inv, $TypeOK$

$\langle 1\rangle 2.$ $\land\ Cardinality(Proc) \in Nat$

$\land\ Cardinality(A1[i]) \in Nat$

$\land\ Cardinality(A1[i]) \leq Cardinality(Proc)$

BY $ProcFinite$, $SubsetFinite$, $CardType$, $SubsetCardinality2$, SMT DEF Inv, $TypeOK$

$\langle 1\rangle 3.$ QED

BY $\langle 1\rangle 2$, SMT

The Refinement Proof

$pcBar \triangleq [p \in Proc \mapsto$ IF $pc[p] =$ "Done" THEN "Done" ELSE "A"$]$

For every symbol F defined in module *GFXSpec*, this defines $PS!F$ to have the same definition as F except with every defined constant and variable of *GFXSpec* replaced by the expression specified in the WITH clause. Constants and variables not explicitly substituted for in the WITH clause are replaced by the symbols of the same name in module *GFX*.

$PS \triangleq$ INSTANCE $GFXSpec$ WITH $pc \leftarrow pcBar$

The following lemmas are the heart of the proof that algorithm *GFX* implements/refines algorithm *GFXSpec*.

LEMMA $InitImplication \triangleq Init \Rightarrow PS!Init$
BY DEF $Init$, $PS!Init$, $ProcSet$, $PS!ProcSet$, $pcBar$

LEMMA $StepSimulation \triangleq Inv \land Inv' \land [Next]_{vars} \Rightarrow [PS!Next]_{PS!vars}$

$\langle 1 \rangle$ SUFFICES ASSUME Inv, Inv', $[Next]_{vars}$
PROVE $[PS!Next]_{PS!vars}$
OBVIOUS

$\langle 1 \rangle 1.$CASE UNCHANGED $vars$
BY $\langle 1 \rangle 1$, SMT DEF $vars$, $PS!vars$, $pcBar$

$\langle 1 \rangle 2.$ ASSUME NEW $p \in Proc$, $a(p)$
PROVE $[PS!Next]_{PS!vars}$

$\langle 2 \rangle$ $pc[p] = \text{"a"}$
BY $\langle 1 \rangle 2$ DEF a

$\langle 2 \rangle 1.$CASE $\land\ known' = [known$ EXCEPT $![p] = known[p] \cup$ UNION $\{A1[i] : i \in Nat\}]$
$\land\ notKnown' = [notKnown$ EXCEPT $![p] = \{i \in 0 \,..\, (Cardinality(known'[p])) : known'[p] \neq A1[i]\}]$
$\land\ notKnown'[p] \neq \{\}$
$\land\ pc' = [pc$ EXCEPT $![p] = \text{"b"}]$
$\land$ UNCHANGED $result$
$\land\ A1' = A1$

$\langle 3 \rangle$ $\land\ result' = result$
$\land\ pc' = [pc$ EXCEPT $![p] = \text{"b"}]$
$\land\ pc \in [Proc \rightarrow \{\text{"a"}, \text{"b"}, \text{"Done"}\}]$
$\land\ pc[p] = \text{"a"}$
BY $\langle 2 \rangle 1$ DEF Inv, $TypeOK$

$\langle 3 \rangle$ $\land\ pcBar \in [Proc \rightarrow \{\text{"A"}, \text{"Done"}\}]$
$\land\ pcBar' \in [Proc \rightarrow \{\text{"A"}, \text{"Done"}\}]$
BY DEF $pcBar$

$\langle 3 \rangle$ SUFFICES ASSUME NEW $q \in Proc$
PROVE $pcBar[q]' = pcBar[q]$
BY DEF $PS!vars$

$\langle 3 \rangle$ QED
BY DEF $PS!vars$, $pcBar$ *SMT* used to prove this but doesn't now

$\langle 2 \rangle 2.$CASE $\land\ known' = [known$ EXCEPT $![p] = known[p] \cup$ UNION $\{A1[i] : i \in Nat\}]$
$\land\ notKnown' =$

$[notKnown \text{ EXCEPT } ![p] =$
$\{i \in 0 \,..\, (Cardinality(known'[p])) : known'[p] \neq A1[i]\}]$
$\land\; notKnown'[p] = \{\}$
$\land\; result' = [result \text{ EXCEPT } ![p] = known'[p]]$
$\land\; pc' = [pc \text{ EXCEPT } ![p] = \text{“Done”}]$
$\land\; A1' = A1$

$\langle 3\rangle 1.\ \forall x : Cardinality(x) = PS!Cardinality(x)$
BY DEF $Cardinality$, $PS!Cardinality$

$\langle 3\rangle 2.\ pcBar[p] = \text{“A”}$
BY $\langle 2\rangle 2$, SMT DEF $TypeOK$, $pcBar$

$\langle 3\rangle 3.\ \exists P \in \{Q \in \text{SUBSET } Proc :$
$\land\; p \in Q$
$\land\; \forall q \in Proc \setminus \{p\} :$
$\lor\; Cardinality(result[q]) \neq Cardinality(Q)$
$\lor\; Q = result[q]$
$\} :$
$result' = [result \text{ EXCEPT } ![p] = P]$

$\langle 4\rangle$ DEFINE $P \triangleq known'[p]$
$\langle 4\rangle$ SUFFICES $\land\; P \in \text{SUBSET } Proc$
$\land\; p \in P$
$\land\; \forall q \in Proc \setminus \{p\} :$
$\lor\; Cardinality(result[q]) \neq Cardinality(P)$
$\lor\; P = result[q]$
BY $\langle 2\rangle 2$

$\langle 4\rangle 1.\ P \in \text{SUBSET } Proc$
BY DEF $TypeOK$, Inv

$\langle 4\rangle 2.\ p \in P$
BY DEF $InvB$, Inv

$\langle 4\rangle 3.$ ASSUME NEW $q \in Proc \setminus \{p\}$
PROVE $\lor\; Cardinality(result[q]) \neq Cardinality(P)$
$\lor\; P = result[q]$

$\langle 5\rangle 1.\ \land\; Cardinality(P) \in Nat$
$\land\; Cardinality(result[q]) \in Nat$
$\land\; IsFiniteSet(P)$
$\land\; IsFiniteSet(result[q])$
BY $ProcFinite$, $SubsetFinite$, $CardType$, SMT DEF Inv, $TypeOK$

$\langle 5\rangle 2.\ \land\; Cardinality(P) \neq 0$
$\land\; P \neq \{\}$
BY $\langle 4\rangle 2$, $\langle 5\rangle 1$, $NonEmptySetCardinality$, SMT DEF $Done$

$\langle 5\rangle 3.$ CASE $result[q] = \{\}$
BY $\langle 5\rangle 2$, $\langle 5\rangle 3$, $EmptySetCardinality$, SMT

$\langle 5\rangle 4.$ CASE $result[q] \neq \{\}$

$\langle 6\rangle 1.\ \land\; result'[q] = result[q]$
$\land\; result'[p] = known'[p]$
BY $\langle 2\rangle 2$, SMT DEF Inv, $TypeOK$

$\langle 6\rangle 2.$ QED
BY $\langle 6\rangle 1$, $\langle 5\rangle 2$, $\langle 5\rangle 4$, SMT DEF Inv, $GFXCorrect$, $Done$
$\langle 5\rangle 5.$ QED
BY $\langle 5\rangle 3$, $\langle 5\rangle 4$
$\langle 4\rangle 4.$ QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$, $\langle 4\rangle 3$
$\langle 3\rangle 4.$ $pcBar' = [pcBar$ EXCEPT $![p] =$ "Done"$]$
$\langle 4\rangle$ $\wedge\ pcBar \in [Proc \to \{$"A", "Done"$\}]$
$\wedge\ pcBar' \in [Proc \to \{$"A", "Done"$\}]$
BY DEF $pcBar$
$\langle 4\rangle$ SUFFICES ASSUME NEW $q \in Proc$
PROVE $pcBar[q]' =$ IF $q = p$ THEN "Done"
ELSE $pcBar[q]$
OBVIOUS
$\langle 4\rangle$ $\wedge\ pc' = [pc$ EXCEPT $![p] =$ "Done"$]$
$\wedge\ pc \in [Proc \to \{$"a", "b", "Done"$\}]$
BY $\langle 2\rangle 2$ DEF Inv, $TypeOK$
$\langle 4\rangle$ QED
BY SMT DEF $pcBar$
$\langle 3\rangle 5.$ QED
BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$, SMT DEF $PS!A$, $PS!Next$, $PS!Pr$
$\langle 2\rangle 4.$ QED
BY $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 1\rangle 2$ DEF a

$\langle 1\rangle 3.$ ASSUME NEW $p \in Proc$, $b(p)$
PROVE UNCHANGED $PS!vars$
$\langle 2\rangle$ SUFFICES ASSUME NEW $q \in Proc$
PROVE $pcBar[q]' = pcBar[q]$
$\langle 3\rangle$ $result' = result$
BY $\langle 1\rangle 3$ DEF b
$\langle 3\rangle$ $pcBar' = pcBar$
BY DEF $pcBar$
$\langle 3\rangle$ QED
BY DEF $PS!vars$
$\langle 2\rangle 1.$CASE $q = p$
BY $\langle 1\rangle 3$, SMT DEF $PS!vars$, $pcBar$, b, Inv, $TypeOK$
$\langle 2\rangle 2.$CASE $q \neq p$
BY $\langle 1\rangle 3$, SMT DEF $PS!vars$, $pcBar$, b, Inv, $TypeOK$
$\langle 2\rangle 3.$ QED
BY $\langle 2\rangle 1$, $\langle 2\rangle 2$
$\langle 1\rangle 4.$ QED
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$ DEF $Next$, Pr

THEOREM $Spec \Rightarrow PS!Spec$
** PROOF

This theorem follows easily by simple TLA reasoning from Theorem Invariance and Lemmas *InitImplication* and *StepSimulation*. However, since *TLAPS* does not yet do temporal reasoning, it can't check the proof, so there's no point writing it out.
** OMITTED

* Modification History
* Last modified *Fri* May 31 05:07:33 *PDT* 2013 by *lamport*
* Last modified *Thu Feb* 14 18:15:21 *CET* 2013 by *caroledelporte*
* Last modified *Thu Feb* 14 14:52:13 *CET* 2013 by *caroledelporte*
* Last modified *Wed Feb* 13 13:37:38 *PST* 2013 by *lamport*
* Last modified *Thu Jan* 03 11:57:19 *CET* 2013 by *merz*
* Last modified *Thu Jan* 03 09:18:28 *CET* 2013 by *merz*
* Created *Wed Jun* 20 01:57:10 *PDT* 2012 by *lamport*