Stepwise implication operators in temporal logic

Ioannis Filippidis

December 22, 2017
URL update on July 23, 2018

Abstract

A collection of TLA$^+$ modules about operators for defining open-systems. Modules describing the semantics of relevant temporal logics precede those modules that pertain to stepwise implication operators. The latter modules contain theorems that express the operators *WhilePlus*, *WhilePlusHalf*, and *Unzip* in raw TLA$^+$. A definition of realizability follows, and a module on the effect of hiding history-determined variables on realizability. This document accompanies the dissertation available at: `http://resolver.caltech.edu/CaltechTHESIS:07202018-115217471`.

Contents

MODULE *TLASemantics*

Some notions about TLA+ in the metatheory.

The TLA+ fragment of constant operators servers as the metatheory. So the metatheory is *ZF* + CHOOSE + functions, similarly to [2, Chapter 16].

References

[1] *L.* Lamport, The temporal logic of actions, *TOPLAS*, 1994 10.1145/177492.177726

[2] *L.* Lamport, Specifying systems, Addison-Wesley, 2002

[3] *M*. Abadi and *L*. Lamport, "An old-fashioned recipe for real time", *TOPLAS*, 1994, 10.1145/186025.186058

EXTENDS *Naturals*, *NaturalsInduction*

CONSTANT *VarNames* META Set of all variable names [2, p.311].

Axiomatic definition of states and behaviors.

$$IsAFunction(f) \triangleq f = [x \in \text{DOMAIN } f \mapsto f[x]]$$

$$IsAState(s) \triangleq \land IsAFunction(s) \quad \land \text{DOMAIN } s = VarNames$$

$$IsABehavior(b) \triangleq \land IsAFunction(b) \quad \land \text{DOMAIN } b = Nat \quad \land \forall n \in Nat : IsAState(b[n])$$

$$NatGeq(n) \triangleq \{r \in Nat : r \geq n\}$$

The finite behavior made of the first n states of behavior sigma.

$$Prefix(sigma, n) \triangleq [i \in 0 \,..\, (n-1) \mapsto sigma[i]]$$

The infinite behavior that starts after the first n states of sigma.

$$Suffix(sigma, n) \triangleq [i \in NatGeq(n) \mapsto sigma[i]]$$

More formally, in this metatheoretic statement H should be a string that is a TLA+ formula.

$$PrefixSat(sigma, n, H) \triangleq \exists tau : \land IsABehavior(tau) \quad \land \forall i \in 0 \,..\, (n-1) : tau[i] = sigma[i] \quad \land tau \models H$$

If a behavior prefix can be extended to satisfy a propert P, then the same prefix can be extended to satisfy any property Q weaker than P.

```
THEOREM PrefixSatImp ≜
    ASSUME
        NEW n,    The condition n ∈ Nat is unused, so not assumed.
        NEW sigma, IsABehavior(sigma),
        TEMPORAL P, TEMPORAL Q,
        P ⇒ Q
    PROVE
        PrefixSat(sigma, n, P)
            ⇒ PrefixSat(sigma, n, Q)
    PROOF
    ⟨1⟩1. SUFFICIES
            ASSUME PrefixSat(sigma, n, P)
            PROVE PrefixSat(sigma, n, Q)
        OBVIOUS
    ⟨1⟩2. PICK tau :  ∧ IsABehavior(tau)
                      ∧ ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
                      ∧ tau ⊨ P
        BY ⟨1⟩1  DEF PrefixSat
    ⟨1⟩3. tau ⊨ Q
        ⟨2⟩1. P ⇒ Q
            OBVIOUS    BY PrefixSatImp!assumption
        ⟨2⟩ QED
            BY ⟨1⟩2, ⟨2⟩1
    ⟨1⟩4. ∧ IsABehavior(tau)
          ∧ ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
          ∧ tau ⊨ Q
        BY ⟨1⟩2, ⟨1⟩3
    ⟨1⟩ QED
        BY ⟨1⟩4  DEF PrefixSat   goal from ⟨1⟩1


 The first n states of tau and sigma match.
THEOREM PrefixSatAsSamePrefix ≜
    ASSUME
        NEW sigma, IsABehavior(sigma),
        NEW n ∈ Nat,
        TEMPORAL H
    PROVE
        PrefixSat(sigma, n, H)
         ≡ ∃ tau :  ∧ IsABehavior(tau)
                    ∧ Prefix(tau, n) = Prefix(sigma, n)
                    ∧ tau ⊨ H
    PROOF
    ⟨1⟩1. SUFFICES
            ASSUME NEW tau, IsABehavior(tau)
```

PROVE $(Prefix(tau, n) = Prefix(sigma, n))$
$\equiv \forall i \in 0 \,..\, (n-1) : \; tau[i] = sigma[i]$

BY DEF $PrefixSat$

⟨1⟩ DEFINE

$SamePrefix \triangleq Prefix(tau, n) = Prefix(sigma, n)$

$TauPrefix \triangleq [i \in 0 \,..\, (n-1) \mapsto tau[i]]$

$SigmaPrefix \triangleq [i \in 0 \,..\, (n-1) \mapsto sigma[i]]$

⟨1⟩2. $SamePrefix \equiv (TauPrefix = SigmaPrefix)$

BY DEF $Prefix, SamePrefix, TauPrefix, SigmaPrefix$

⟨1⟩3. $SamePrefix \equiv \land$ DOMAIN $TauPrefix =$ DOMAIN $SigmaPrefix$
$\land \forall i \in$ DOMAIN $TauPrefix$:
$TauPrefix[i] = SigmaPrefix[i]$

BY ⟨1⟩2

⟨1⟩4. DOMAIN $TauPrefix =$ DOMAIN $SigmaPrefix$

BY DEF $TauPrefix, SigmaPrefix$

⟨1⟩5. $\forall i \in$ DOMAIN $TauPrefix$: $\land TauPrefix[i] = tau[i]$
$\land SigmaPrefix[i] = sigma[i]$

BY DEF $TauPrefix, SigmaPrefix$

⟨1⟩6. $SamePrefix$
$\equiv \forall i \in$ DOMAIN $TauPrefix$: $tau[i] = sigma[i]$

BY ⟨1⟩3, ⟨1⟩4, ⟨1⟩5

⟨1⟩7. $SamePrefix$
$\equiv \forall i \in 0 \,..\, (n-1) : \; tau[i] = sigma[i]$

BY ⟨1⟩6

⟨1⟩ QED

BY ⟨1⟩7 DEF $SamePrefix$

PROPOSITION $SamePrefixImpliesPrefixSatToo \triangleq$

ASSUME

TEMPORAL H,

NEW $n \in Nat$,

NEW $sigma$, $IsABehavior(sigma)$,

NEW eta, $IsABehavior(eta)$,

$\land Prefix(sigma, n) = Prefix(eta, n)$

$\land PrefixSat(sigma, n, H)$

PROVE

$PrefixSat(eta, n, H)$

⟨1⟩1. PICK tau : $\land IsABehavior(tau)$
$\land Prefix(tau, n) = Prefix(sigma, n)$
$\land tau \models H$

BY $PrefixSatAsSamePrefix$

⟨1⟩2. $Prefix(sigma, n) = Prefix(eta, n)$

OBVIOUS BY $SamePrefixImpliesPrefixSatToo$!assumption

⟨1⟩3. $\land IsABehavior(tau)$

$\land \textit{Prefix}(\textit{tau}, n) = \textit{Prefix}(\textit{eta}, n)$
$\land \textit{tau} \models H$
BY ⟨1⟩1, ⟨1⟩2
⟨1⟩ QED
BY ⟨1⟩3, *PrefixSatAsSamePrefix*

If the first n states of two behaviors are the same,
then *PrefixSat* for n, H has the same value for both behaviors.

THEOREM *EquivPrefixSatIfSamePrefix* $\triangleq$
ASSUME
TEMPORAL H,
NEW $n \in \textit{Nat}$,
NEW *sigma*, $\textit{IsABehavior}(\textit{sigma})$,
NEW *eta*, $\textit{IsABehavior}(\textit{eta})$,
$\textit{Prefix}(\textit{sigma}, n) = \textit{Prefix}(\textit{eta}, n)$
PROVE
$\textit{PrefixSat}(\textit{sigma}, n, H) \equiv \textit{PrefixSat}(\textit{eta}, n, H)$
⟨1⟩1. $\textit{PrefixSat}(\textit{sigma}, n, H) \Rightarrow \textit{PrefixSat}(\textit{eta}, n, H)$
BY *SamePrefixImpliesPrefixSatToo*
⟨1⟩2. $\textit{PrefixSat}(\textit{eta}, n, H) \Rightarrow \textit{PrefixSat}(\textit{sigma}, n, H)$
BY *SamePrefixImpliesPrefixSatToo*
⟨1⟩ QED
BY ⟨1⟩1, ⟨1⟩2

The "while" operator $-\!\!\triangleright$
[3, *Sec.* $A4$ on p. $A-2$]

$\textit{sigma} \models \textit{While}(A, G) \triangleq$
$\land \forall n \in \textit{Nat} : \textit{PrefixSat}(\textit{sigma}, n, A) \Rightarrow \textit{PrefixSat}(\textit{sigma}, n, G)$
$\land \textit{sigma} \models A \Rightarrow G$

The "while plus" operator $\overset{+}{-\!\!\triangleright}$

Semantic form of stepwise implication.

For the safety properties $A \triangleq \Box \textit{EnvNext}$ and $G \triangleq \Box \textit{SysNext}$, the semantic operator corresponds to the syntactic operator as follows

$\textit{sigma} \models \textit{StepwiseImpl}(\textit{EnvNext}, \textit{SysNext}) \equiv \textit{PrefixPlusOne}(\textit{sigma}, A, G)$

$\textit{PrefixPlusOne}(\textit{sigma}, A, G) \triangleq$
$\forall n \in \textit{Nat} : \textit{PrefixSat}(\textit{sigma}, n, A) \Rightarrow \textit{PrefixSat}(\textit{sigma}, n+1, G)$

The while-plus operator [2, p.316].

$\textit{sigma} \models A \overset{+}{-\!\!\triangleright} G \triangleq$
$\land \textit{PrefixPlusOne}(\textit{sigma}, A, G)$
$\land \textit{sigma} \models A \Rightarrow G$

This theorem expands the definition of $\overset{+}{\rhd}$.

THEOREM *WhilePlusProperties* $\triangleq$

  ASSUME

    NEW *sigma*, *IsABehavior*(*sigma*),

    TEMPORAL A, **TEMPORAL** G,

    $A \overset{+}{\rhd} G$

  PROVE

    $\land\ sigma \models A \Rightarrow G$

    $\land\ \forall n \in Nat :$

      $PrefixSat(sigma, n, A) \Rightarrow PrefixSat(sigma, n+1, G)$

  BY **DEF** $\overset{+}{\rhd}$, *PrefixPlusOne*

We can view $\overset{+}{\rhd}$ (other stepwise operators too) as an infinite conjunction:

$sigma \models A \overset{+}{\rhd} G \equiv$

  $\land\ PrefixSat(sigma, 0, A) \Rightarrow PrefixSat(sigma, 1, G)$

  $\land\ PrefixSat(sigma, 1, A) \Rightarrow PrefixSat(sigma, 2, G)$

  ...

  ...

  $\land\ A \Rightarrow G$

Metatheoretic definition that means

$\{var \in VarNames : \neg \models F \equiv (\boldsymbol{\forall}\, var : F)\}$

It seems that a semantic definition needs to mention all other variables, thus an infinity of strings. A syntactic definition can be given for any (finite length) formula by simply parsing it.

$VariablesOf(formula) \triangleq \{$

  $var \in VarNames : \exists\, sigma, tau :$

    $\land\ IsABehavior(sigma)$

    $\land\ IsABehavior(tau)$ tau is same as in $\boldsymbol{\exists}$ DEF

    $\land\ RefinesUpToVar(tau, sigma, var)$

    $\land\ \neg((sigma \models formula) \equiv \neg(tau \models \neg formula))\}$

Stutter at state forever.

$Stutter(state) \triangleq [n \in Nat \mapsto state]$

Keep states $0 \,..\, k$ and stutter state k indefinitely.

$StutterAfter(sigma, n) \triangleq [i \in Nat \mapsto$ **IF** $i < n$ **THEN** $sigma[i]$ **ELSE** $sigma[n]]$

THEOREM *StutterAfterIsABehavior* $\triangleq$

  ASSUME

    NEW $n \in Nat$,

    NEW *sigma*,

    IsABehavior(*sigma*)

  PROVE

    LET

```
        eta ≜ StutterAfter(sigma, n)
    IN
        IsABehavior(eta)
⟨1⟩ DEFINE eta ≜ StutterAfter(sigma, n)
⟨1⟩1. ∧ IsAFunction(eta)
      ∧ DOMAIN eta = Nat
    BY DEF eta, StutterAfter
⟨1⟩2. ASSUME NEW i ∈ Nat
      PROVE IsAState(eta[i])
    ⟨2⟩1. ASSUME NEW r ∈ Nat
          PROVE IsAState(sigma[r])
        ⟨3⟩1. IsABehavior(sigma)
            OBVIOUS    BY ASSUME
        ⟨3⟩ QED
            BY ⟨2⟩1 DEF IsABehavior
    ⟨2⟩2. PICK r ∈ Nat : sigma[r] = eta[i]
        ⟨3⟩1.CASE i < n
            ⟨4⟩1. sigma[i] = eta[i]
                BY ⟨3⟩1 DEF eta, StutterAfter
            ⟨4⟩ QED
                BY ⟨4⟩1, ⟨1⟩2    The witness is i.
        ⟨3⟩2.CASE i ≥ n
            ⟨4⟩1. sigma[n] = eta[i]
                BY ⟨3⟩2 DEF eta, StutterAfter
            ⟨4⟩ QED
                BY ⟨4⟩1    The witness is n.
        ⟨3⟩ QED
            BY ⟨3⟩1, ⟨3⟩2, ⟨1⟩2
    ⟨2⟩3. IsAState(sigma[r])
        BY ⟨2⟩1, ⟨2⟩2
    ⟨2⟩ QED
        BY ⟨2⟩2, ⟨2⟩3
⟨1⟩ QED
    BY ⟨1⟩1, ⟨1⟩2 DEF IsABehavior

THEOREM StutterAfterHasSamePrefix ≜
    ASSUME
        NEW n ∈ Nat,
        NEW k ∈ Nat,
        k < n,
        NEW sigma,
        IsABehavior(sigma)
    PROVE
        LET
```

```
            eta ≜ StutterAfter(sigma, n)
        IN
            eta[k] = sigma[k]
    BY DEF StutterAfter

THEOREM StutteringTail ≜
    ASSUME
        NEW n ∈ Nat,
        NEW k ∈ Nat,
        k ≥ n,
        NEW sigma,
        IsABehavior(sigma)
    PROVE
        LET
            eta ≜ StutterAfter(sigma, n)
        IN
            eta[k] = sigma[n]
    BY DEF StutterAfter

THEOREM StutterAfterInit ≜
    ASSUME
        NEW n ∈ Nat,
        NEW sigma,
        IsABehavior(sigma)
    PROVE
        LET
            eta ≜ StutterAfter(sigma, n)
        IN
            eta[0] = sigma[0]
    ⟨1⟩1. n ∈ Nat
        OBVIOUS    BY StutterAfterInit!assumption
    ⟨1⟩2.CASE 0 < n
        BY StutterAfterHasSamePrefix
    ⟨1⟩3.CASE 0 ≥ n
        BY StutteringTail
    ⟨1⟩ QED
        BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3
```

Metatheoretic statements asserting that P is a TLA+ expression of a certain level. Used for bookkeeping during hand-written proofs.

```
IsStateLevel(P) ≜ TRUE
IsTemporalLevel(P) ≜ TRUE
```

Metatheoretic statement asserting that P is a TLA+ formula.
Useful to distinguish from raw TLA+ formulas that aren't TLA+ formulas.

$IsATLAPlusFormula(P) \triangleq$ TRUE

The stutter-free form $\natural\sigma$ of σ, [2, p.316].

If the behavior sigma contains a finite number of nonstuttering steps, then *Natural*(*sigma*) is a finite sequence. A similar operator in [1] yields an infinite sequence with a stuttering tail. Either of these definitions can be used to define temporal quantification.

$$
\begin{array}{l}
Natural(sigma) \triangleq \\
\quad \text{LET} \\
\qquad f[n \in Nat] \triangleq \text{IF } n = 0 \text{ THEN } 0 \\
\qquad\qquad\qquad\qquad \text{ELSE IF } sigma[n] = sigma[n-1] \\
\qquad\qquad\qquad\qquad\qquad \text{THEN } f[n-1] \\
\qquad\qquad\qquad\qquad\qquad \text{ELSE } f[n-1] + 1 \\
\qquad S \triangleq \{f[n] : n \in Nat\} \\
\quad \text{IN} \\
\qquad [n \in S \mapsto \\
\qquad\; sigma[\text{CHOOSE } i \in Nat : f[i] = n]]
\end{array}
$$

The behaviors s and t differ only in the values of variable var. This is $s =_{var} t$ in [1].

$$
\begin{array}{l}
EqualUpToVar(s, t, var) \triangleq \\
\quad \forall n \in Nat : \forall v \in VarNames : (v \neq var) \Rightarrow (s[n][v] = t[n][v])
\end{array}
$$

$\rho \sim \sigma$ asserts that the sequences $\natural\rho$ and $\natural\sigma$ are equal [1].

$Sim(rho, sigma) \triangleq Natural(rho) = Natural(sigma)$

A useful definition, based on [1].
RefinesUpToVar is noncommutative (though both *EqualUpToVar* and *Sim* are commutative).

$$
\begin{array}{l}
RefinesUpToVar(tau, sigma, var) \triangleq \\
\quad \exists rho : \; \wedge IsABehavior(rho) \\
\qquad\qquad\;\; \wedge Sim(rho, sigma) \\
\qquad\qquad\;\; \wedge EqualUpToVar(rho, tau, var)
\end{array}
$$

The similarity of sutter-free forms after overwriting the variable x in
the behavior τ. [2, p.316].

$$
\begin{array}{l}
SimUpToVar(sigma, tau, var) \triangleq \\
\quad \text{LET} \\
\qquad s \triangleq Natural(sigma) \\
\qquad t \triangleq Natural(tau) \\
\quad \text{IN} \\
\qquad s = [n \in \text{DOMAIN } t \mapsto [t[n] \text{ EXCEPT } ![var] = s[n][var]]]
\end{array}
$$

Temporal existential quantification, based on [1].

AXIOM

$$
\begin{aligned}
&sigma \models \textbf{∃}\, x : \;\; F \equiv \\
&\quad \exists\, tau : \;\; \land\, IsABehavior(tau) \\
&\qquad\qquad\;\; \land\, RefinesUpToVar(tau,\, sigma,\, \text{“x”}) \\
&\qquad\qquad\;\; \land\, tau \models F
\end{aligned}
$$

Temporal existential quantification, based on [2] and [3, p. $A - 2$].

AXIOM

$$
\begin{aligned}
&sigma \models \textbf{∃}\, x : \;\; F \equiv \\
&\quad \exists\, tau : \;\; \land\, IsABehavior(tau) \\
&\qquad\qquad\;\; \land\, SimUpToVar(sigma,\, tau,\, \text{“x”}) \\
&\qquad\qquad\;\; \land\, tau \models F
\end{aligned}
$$

MODULE *TemporalLogic*

Some definitions about temporal properties.

Author: Ioannis *Filippidis*

References

[1] *L*. Lamport, "Miscellany", 21 *April* 1991

[2] *M*. Abadi and *S*. Merz, "On TLA as a logic", Deductive Program Design, 1996

[3] *B*. Alpern and *F.B*. Schneider "Defining liveness", *IPL*, 1985 10.1016/0020−0190(85)90056−0

[4] *B*. Jonsson and *Y*.-K. Tsay, "Assumption/guarantee specifications in linear-time temporal logic", *TCS*, 1996, 10.1016/0304 − 3975(96)00069 − 2

[5] *M*. Abadi and *L*. Lamport, "Conjoining specifications", *TOPLAS*, 1995 10.1145/203095.201069

[6] *L*. Lamport, "Proving possibility properties", *TCS*, 1998 10.1016/*S*0304 − 3975(98)00129 − 7

[7] *M*. Abadi and *L*. Lamport, "An old-fashioned recipe for real time", *TOPLAS*, 1994, 10.1145/186025.186058

[8] *U*. Klein and *N*. Piterman and A. Pnueli, "Effective synthesis of asynchronous systems from *GR*(1) specifications", *VMCAI*, 2012, 10.1007/978 − 3 − 642 − 27940-9_19 (Technical report 2011 − 944 of *Courant Inst*. of Math. Sciences)

[9] A. Pnueli and *R*. Rosner, "On the synthesis of an asynchronous reactive module", *ICALP*, 1989, 10.1007/*BFb*0035790

EXTENDS *TLASemantics*, *NaturalsInduction*

Safety and liveness.

$$MustUnstep(b) \triangleq \land\ b = \text{TRUE} \quad \land\ \Box[b' = \text{FALSE}]_b \quad \land\ \Diamond(b = \text{FALSE})$$

$$SamePrefix(b, u, x) \triangleq \Box(b \Rightarrow (u = x))$$

$$Front(P(\_, \_), x, b) \triangleq \boldsymbol{\exists}\, u : P(u) \land SamePrefix(b, u, x)$$

A syntactic definition of closure [1].
See also [2, *Sec*. 5.3] and [4, *Sec*. 2.1 on p. 52].

$$Cl(P(\_), x) \triangleq \boldsymbol{\forall}\, b : MustUnstep(b) \Rightarrow Front(P, x, b)$$

A semantic definition of closure [6, *Eq*. (1) on p. 342] and [7, p. *A* − 2].
The syntactic and semantic definitions of closure are equivalent.

$sigma \models Cl(P) \triangleq \forall n \in Nat: PrefixSat(sigma, n, P)$

Using closure we can define safety and liveness [6, p. 343].

These definitions are equivalent to those that mention violating behaviors [6, *Eq.*(2) on p.343].

$IsSafety(P(\_)) \triangleq \forall x : \; P(x) \equiv Cl(P, x)$
$IsLiveness(P(\_)) \triangleq \forall x : \; Cl(P, x)$

Each property is decomposable into safety and liveness [3].

$SafetyPart(P(\_), x) \triangleq Cl(P, x)$
$LivenessPart(P(\_), x) \triangleq SafetyPart(P, x) \Rightarrow P(x)$ [4, *Sec.* 2.3 on p.54]

Conjoining the safety and liveness parts yields the property P.

THEOREM
  ASSUME
    TEMPORAL $P(\_)$, VARIABLE x
  PROVE
    $P(x) \equiv \land SafetyPart(P, x)$
    $\qquad \land LivenessPart(P, x)$
  PROOF
  $\langle 1 \rangle 1.$ $LivenessPart(P, x) \equiv (SafetyPart(P, x) \Rightarrow P(x))$
    BY DEF $LivenessPart$
  $\langle 1 \rangle$ QED
    BY $\langle 1 \rangle 1$

For any temporal property P, the safety part is a safety property and the liveness part is a liveness property.

THEOREM
  ASSUME
    TEMPORAL $P(\_)$, VARIABLE x
  PROVE
    LET
      $S(u) \triangleq SafetyPart(P, u)$
      $L(u) \triangleq LivenessPart(P, u)$
    IN
      $\land IsSafety(S, x)$
      $\land IsLiveness(L, x)$
  PROOF OMITTED

$IsMachineClosed(S(\_), L(\_), x) \triangleq$
  LET
    $SL(u) \triangleq S(u) \land L(u)$
  IN
    $S(x) \equiv Cl(SL, x)$

$IsConstant(P) \triangleq \exists c : \; \Box(P = c)$
$Canonical(Init, Next, L, v) \triangleq Init \land \Box[Next]_v \land L$

The "state machine" form.

$SM(Init, Next, v) \triangleq Init \land \Box[Next]_v$

Any action comprises of a nonstuttering and a stuttering part.

$StutteringPart(A, v) \triangleq A \land (v = v')$

$NonStutteringPart(A, v) \triangleq \langle A \rangle_v$ alternative name: *ChangingPart*

THEOREM
  ASSUME STATE v, ACTION A
  PROVE $A \equiv \lor StutteringPart(A, v)$
      $\lor \langle A \rangle_v$
  OMITTED

THEOREM
  ASSUME STATE v, ACTION A
  PROVE $\langle A \rangle_v \Rightarrow [A]_v$
  OMITTED

trick for handling other arities:

LET $P(x) \triangleq L(x.p, x.q)$
IN $IsLiveness(P)$

Temporal quantification in raw TLA+ with past.

Teamporal quantification that preserves stutter-invariance [8, *Sec.* 2.1].
See also [9] (where behavior indices are not used though).

$$
\begin{aligned}
&sigma, i \models \mathbf{\exists}\, x : \ F \equiv \\
&\quad \exists\, tau, k : \\
&\qquad \land IsABehavior(tau) \\
&\qquad \land k \in Nat \\
&\qquad \land tau, k \models F \\
&\qquad \land \exists\, rho : \\
&\qquad\quad \text{LET} \\
&\qquad\qquad Start(r) \triangleq 0 \,..\, (r - 1) \\
&\qquad\qquad End(r) \triangleq Nat \setminus Start(r) \\
&\qquad\qquad RhoFront \triangleq [n \in 0 \,..\, Start(k) \mapsto rho[n]] \\
&\qquad\qquad TauFront \triangleq [n \in 0 \,..\, Start(i) \mapsto tau[n]] \\
&\qquad\qquad RhoTail \triangleq [n \in End(k) \mapsto rho[n]] \\
&\qquad\qquad TauTail \triangleq [n \in End(i) \mapsto tau[n]] \\
&\qquad\quad \text{IN} \\
&\qquad\qquad \land IsABehavior(rho) \\
&\qquad\qquad \land Sim(RhoFront, TauFront) \\
&\qquad\qquad \land Sim(RhoTail, TauTail) \\
&\qquad\qquad \land EqualUpToVar(rho, tau, \text{“x”})
\end{aligned}
$$

Temporal quantification that breaks stutter-invariance [8, *Sec.* 2.1].

$sigma, i \models EEEx : F \equiv$
$\quad \exists tau : \land IsABehavior(tau)$
$\qquad\qquad \land EqualUpToVar(sigma, tau, \text{“x”})$
$\qquad\qquad \land tau, i \models F$

Properties of closure

LEMMA *ClosureProperties* $\triangleq$
ASSUME
TEMPORAL P, NEW $sigma$, NEW $n \in Nat$,
$\land IsABehavior(sigma)$
$\land sigma \models Cl(P)$
PROVE
$\exists tau : \land IsABehavior(tau)$
$\qquad \land \forall i \in 0 \,..\, n : tau[i] = sigma[i]$
$\qquad \land tau \models P$
OMITTED

[6, Prop. 1/item 2] and [7, *Sec.* $B3$ on p. $A-4$]

LEMMA *ClosureIsMonotonic* $\triangleq$
ASSUME
VARIABLE x,
TEMPORAL $A(\_)$, TEMPORAL $B(\_)$,
$\boldsymbol{\forall} u : A(u) \Rightarrow B(u)$
PROVE
$Cl(A, x) \Rightarrow Cl(B, x)$
PROOF
$\langle 1 \rangle 1.\ Cl(A, x) \equiv$
$\quad \boldsymbol{\forall} b : \lor \neg MustUnstep(b)$
$\qquad \lor \boldsymbol{\exists} u :$
$\qquad\quad \land A(u)$
$\qquad\quad \land \Box(b \Rightarrow (u = x))$
BY DEF Cl
$\langle 1 \rangle$ DEFINE
$H \triangleq \boldsymbol{\exists} u : \land A(u)$
$\qquad\qquad \land \Box(b \Rightarrow (u = x))$

$G \triangleq \boldsymbol{\exists} u : \land B(u)$
$\qquad\qquad \land \Box(b \Rightarrow (u = x))$
$\langle 1 \rangle 2.\ H \equiv \boldsymbol{\exists} u : \land A(u)$
$\qquad \land \boldsymbol{\forall} x : A(x) \Rightarrow B(x)$
$\qquad \land \Box(b \Rightarrow (u = x))$
BY DEF H and *ClosureIsMonotonic*!assumption

$\langle 1 \rangle 3.\ H \equiv \exists\, u : \ \land A(u)$
$\qquad \land A(u) \Rightarrow B(u)$
$\qquad \land \Box(b \ \Rightarrow (u = x))$

BY $\langle 1 \rangle 2$

$\langle 1 \rangle 4.\ H \Rightarrow G$

BY $\langle 1 \rangle 3$ DEF G

$\langle 1 \rangle 5.\ Cl(A, x) \Rightarrow$
$\quad \forall\, b : \ \lor \neg MustUnstep(b)$
$\qquad \lor G$

BY $\langle 1 \rangle 1$, $\langle 1 \rangle 4$ DEF H, G

$\langle 1 \rangle 6.\ Cl(B, x) \equiv$
$\quad \forall\, b : \ \lor \neg MustUnstep(b)$
$\qquad \lor \exists\, u :$
$\qquad\quad \land B(u)$
$\qquad\quad \land \Box(b \Rightarrow (u = x))$

BY DEF Cl

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 5$, $\langle 1 \rangle 6$ DEF G

If the closure of property P is satisfiable, so is P.

LEMMA $SATClosureInit \triangleq$

ASSUME

TEMPORAL P,
STATE $Init$, STATE v, ACTION $Next$,
NEW $sigma$,
$\land \models Cl(P, x) \equiv (Init \land \Box[Next]_v)$
$\land IsABehavior(sigma)$
$\land sigma \models Init$

PROVE

$\exists\, tau : \ \land IsABehavior(tau)$
$\qquad \land tau[0] = sigma[0]$
$\qquad \land sigma \models P$

PROOF

$\langle 1 \rangle$ DEFINE $eta \triangleq [n \in Nat \mapsto sigma[0]]$

$\langle 1 \rangle 2.\ IsABehavior(eta)$

$\langle 1 \rangle 3.\ eta \models \Box[\text{FALSE}]_v$

BY DEF eta

$\langle 1 \rangle 4.\ eta \models Init$

$\langle 2 \rangle 1.\ sigma \models Init$

OBVIOUS

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 1$ DEF eta

$\langle 1 \rangle 5.\ eta \models Cl(P)$

$\langle 2 \rangle 1.\ eta \models Init \land \Box[Next]_v$

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

$\langle 2 \rangle 2$. $\models Cl(P) \equiv (Init \land \Box[Next]_v)$

OBVIOUS

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle 6$. PICK $beta$: $\land\ IsABehavior(beta)$
$\land\ beta[0] = eta[0]$
$\land\ beta \models P$

BY $\langle 1 \rangle 5$, *ClosureProperties*

$\langle 1 \rangle$ QED

$\langle 2 \rangle 1$. $eta[0] = sigma[0]$

BY DEF eta

$\langle 2 \rangle$ QED

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

LEMMA *ClosureOfSafety* $\triangleq$

ASSUME

TEMPORAL P,

$IsSafety(P)$

PROVE

$Cl(P) \equiv P$

OMITTED

PROPOSITION *ClosureAndLiveness* $\triangleq$

ASSUME

VARIABLE x, VARIABLE y,

STATE $Init$, ACTION $Next$,

TEMPORAL L,

LET

$v \triangleq \langle x, y \rangle$

$S \triangleq Init \land \Box[Next]_v$

IN

$IsMachineClosed(S, L)$

PROVE

LET

$v \triangleq \langle x, y \rangle$

$S \triangleq Init \land \Box[Next]_v$

$P \triangleq S \land L$

IN

$P \equiv (L \land Cl(P))$

PROOF

$\langle 1 \rangle$ DEFINE

$v \triangleq \langle x, y \rangle$

$S \triangleq Init \land \Box[Next]_v$

$P \triangleq S \land L$

$\langle 1 \rangle 1.\ S \equiv Cl(S \land L)$

BY DEF *IsMachineClosed*

$\langle 1 \rangle 2.\ (S \land L) \equiv (L \land Cl(S \land L))$

BY $\langle 1 \rangle 1$

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 2$

If a property P implies a safety property Q, then the closure of P implies Q.

LEMMA $ClosureIsTightestSafety \triangleq$

ASSUME

TEMPORAL P, TEMPORAL Q,

$\land\ IsSafety(Q)$

$\land\ P \Rightarrow Q$

PROVE

$Cl(P) \Rightarrow Q$

OMITTED

The closure of a property P is the tightest safety property that P implies [6, Prop.1/item 1]. Also, *Extensivity* among *Kuratowski*'s closure axioms.

LEMMA $ClosureImplied \triangleq$

ASSUME

symbols u and b are undeclared in the current context

TEMPORAL $P(\_)$,

VARIABLE x

PROVE

$P(x) \Rightarrow Cl(P, x)$

PROOF

$\langle 1 \rangle 1.\ P(x) \Rightarrow \exists\, u :\ P(x) \land \Box(u = x)$

OBVIOUS

The bound variable u is a history-determined variable. u is undeclared in the current context, so u does not occur in the expression $P(x)$.

$\langle 1 \rangle 2.\ (\exists\, u :\ P(x) \land \Box(u = x))$
$\Rightarrow \forall\, b :\ \lor \neg MustUnstep(b)$
$\lor \exists\, u :\ \land P(u)$
$\land \Box(b \Rightarrow (u = x))$

$\langle 2 \rangle 1.\ (\exists\, u :\ P(x) \land \Box(u = x))$
$\Rightarrow \exists\, u :\ P(x) \land \Box(u = x) \land P(u)$

OMITTED a proof of this step should argue about all possible temporal-level expressions $P(\_)$, thus in terms of all the production rules of the grammar, and the semantics.

$\langle 2 \rangle 2.\ (\exists\, u :\ P(x) \land \Box(u = x) \land P(u))$
$\Rightarrow \exists\, u :\ \Box(u = x) \land P(u)$

OBVIOUS

⟨2⟩3. $(\exists u : \Box(u = x) \land P(u))$
$\Rightarrow \forall b : \exists u : \Box(u = x) \land P(u)$

OBVIOUS

The identifier b is undeclared in the current context, so b does not occur in the expression $P(u)$.

⟨2⟩4. $(\forall b : \exists u : \Box(u = x) \land P(u))$
$\Rightarrow \forall b : \lor \neg MustUnstep(b)$
$\lor \exists u : P(u) \land \Box(b \Rightarrow (u = x))$

OBVIOUS

⟨2⟩ QED

BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3, ⟨2⟩4

⟨1⟩3. $Cl(P, x) \equiv \forall b : \lor \neg MustUnstep(b)$
$\lor \exists u : \land P(u)$
$\land \Box(b \Rightarrow (u = x))$

BY DEF *Cl*

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3

COROLLARY *ClosureIdempotent* $\triangleq$

ASSUME

TEMPORAL P

PROVE

LET $C \triangleq Cl(P)$

IN $Cl(C) \equiv C$

⟨1⟩1. $Cl(P) \Rightarrow Cl(Cl(P))$

⟨2⟩1. $P \Rightarrow Cl(P)$

BY *ClosureImplied*

⟨2⟩ QED

BY ⟨2⟩1, *ClosureIsMonotonic*

⟨1⟩2. $Cl(Cl(P)) \Rightarrow Cl(P)$

OMITTED Sketch: for any n-prefix of sigma, pick an extension tau, with $tau \models Cl(P)$. BY DEF of Cl, every prefix of tau is extensible to a behavior that satisfies P. For the n-prefix of tau pick such an extension eta, with $eta \models P$.

The behaviors sigma and tau have common n-prefix. Thus, eta is an extension of $sigma[0 \,..\, n]$ that satisfies P. BY DEF of Cl, $sigma \models Cl(P)$.

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

LEMMA *ClosureOfImpl* $\triangleq$

ASSUME

TEMPORAL E, TEMPORAL M

PROVE

$(\mathit{Cl}(E) \Rightarrow \mathit{Cl}(M)) \;\Rightarrow\; \mathit{Cl}(E \Rightarrow M)$

PROOF

$\langle 1\rangle 1.$ $\mathit{Cl}(M) \Rightarrow \mathit{Cl}(E \Rightarrow M)$

$\langle 2\rangle 1.$ $M \Rightarrow (E \Rightarrow M)$

OBVIOUS

$\langle 2\rangle$ QED

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

$\langle 1\rangle 2.$ $(\neg \mathit{Cl}(E)) \Rightarrow \mathit{Cl}(E \Rightarrow M)$

$\langle 2\rangle 1.$ $(\neg \mathit{Cl}(E)) \Rightarrow \neg E$

BY *ClosureImplied*

$\langle 2\rangle 2.$ $(\neg E) \Rightarrow (E \Rightarrow M)$

OBVIOUS

$\langle 2\rangle 3.$ $(E \Rightarrow M) \;\Rightarrow\; \mathit{Cl}(E \Rightarrow M)$

BY *ClosureImplied*

$\langle 2\rangle$ QED

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

$\langle 1\rangle$ QED

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

PROPOSITION *ConjClosureInsideClosure* $\triangleq$

ASSUME

TEMPORAL A, TEMPORAL B,

$A \Rightarrow \mathit{Cl}(B)$

PROVE

$\mathit{Cl}(A) \;\equiv\; \mathit{Cl}(A \wedge \mathit{Cl}(B))$

PROOF

$\langle 1\rangle$ DEFINE

$Q \triangleq A \wedge \mathit{Cl}(B)$

$\langle 1\rangle 1.$ $\mathit{Cl}(Q) \Rightarrow \mathit{Cl}(A)$

$\langle 2\rangle 1.$ $Q \Rightarrow A$

BY DEF Q

$\langle 2\rangle$ QED

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

$\langle 1\rangle 2.$ $\mathit{Cl}(A) \Rightarrow \mathit{Cl}(Q)$

$\langle 2\rangle 1.$ $A \Rightarrow \mathit{Cl}(B)$

OBVIOUS BY *ConjClosureInsideClosure*!assumption

$\langle 2\rangle 2.$ $A \Rightarrow (A \wedge \mathit{Cl}(B))$

BY $\langle 2\rangle 1$

$\langle 2\rangle 3.$ $A \Rightarrow Q$

BY $\langle 2\rangle 2$ DEF Q

$\langle 2\rangle$ QED

BY $\langle 2\rangle 2$, *ClosureIsMonotonic*

$\langle 1\rangle$ QED

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

PROPOSITION *ClosureSample* $\triangleq$

ASSUME

these operators may depend on variables declared in the context where this theorem is used. So the bound identifiers declared within the theorem and its proof are assumed to stand for identifiers that are selected to be different from all previously declared identifiers.

This is required by the rules of TLA+, which doesn't allow redeclaration of an identifier, even a bounded identifier.

VARIABLE x,
CONSTANT $R(\_, \_)$,
TEMPORAL $P(\_)$

PROVE

$$\begin{array}{l} \lor \neg \exists u : R(u, x) \land Cl(P, u) \\ \lor \exists u : R(u, x) \land P(u) \end{array}$$

PROOF

$\langle 1 \rangle 1.$ $(\exists u : R(u, x) \land Cl(P, u))$

$$\equiv \exists u : \begin{array}{l} \land R(u, x) \\ \land \forall b : \\ \quad \lor \neg MustUnstep(b) \\ \quad \lor \exists r : \begin{array}{l} \land P(r) \\ \land \Box(b \Rightarrow (r = u)) \end{array} \end{array}$$

BY DEF *Cl*

$\langle 1 \rangle 2.$ $\exists q : MustUnstep(q)$

BY DEF *MustUnstep*

$\langle 1 \rangle 3.$ $(\exists u : R(u, x) \land Cl(P, u))$

$$\equiv \exists u : \begin{array}{l} \land R(u, x) \\ \land \exists q : MustUnstep(q) \\ \land \forall b : \\ \quad \lor \neg MustUnstep(b) \\ \quad \lor \exists r : \begin{array}{l} \land P(r) \\ \land \Box(b \Rightarrow (r = u)) \end{array} \end{array}$$

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

$\langle 1 \rangle 4.$ $(\exists u : R(u, x) \land Cl(P, u))$

$$\begin{array}{l} \equiv \exists u, q : \\ \quad \land R(u, x) \land MustUnstep(q) \\ \quad \land \forall b : \\ \quad\quad \lor \neg MustUnstep(b) \\ \quad\quad \lor \exists r : \begin{array}{l} \land P(r) \\ \land \Box(b \Rightarrow (r = u)) \end{array} \end{array}$$

BY $\langle 1 \rangle 3$ pull $\exists q$ outside

$\langle 1 \rangle 5.$ $(\exists u : R(u, x) \land Cl(P, u))$

$$\begin{array}{l} \Rightarrow \exists u, q : \\ \quad \land R(u, x) \land MustUnstep(q) \\ \quad \land \lor \neg MustUnstep(q) \\ \quad\quad \lor \exists r : \land P(r) \end{array}$$

$\land \Box(q \Rightarrow (r = u))$

BY ⟨1⟩4 DEF $\forall$ substitute STATE q for b

⟨1⟩6. $(\exists u : R(u, x) \land Cl(P, u))$
$\Rightarrow \exists u, q :$
$\land R(u, x) \land MustUnstep(q)$
$\land \exists r : \land P(r)$
$\land \Box(q \Rightarrow (r = u))$

BY ⟨1⟩5

⟨1⟩7. ASSUME VARIABLE q, VARIABLE u
PROVE $\lor \neg \land MustUnstep(q)$
$\land \Box(q \Rightarrow (r = u))$
$\lor r = u$

⟨2⟩1. ASSUME VARIABLE q, VARIABLE u
PROVE $MustUnstep(q) \Rightarrow (q = \text{TRUE})$
BY DEF $MustUnstep$

⟨2⟩ QED
BY ⟨2⟩1

⟨1⟩8. $(\exists u : R(u, x) \land Cl(P, u))$
$\Rightarrow \exists u, q, r :$
$\land R(u, x) \land P(r)$
$\land (r = u)$

BY ⟨1⟩6, ⟨1⟩7

⟨1⟩9. $(\exists u : R(u, x) \land Cl(P, u))$
$\Rightarrow \exists u, q, r :$
$R(r, x) \land P(r)$

⟨1⟩10. $\lor \neg \exists u : R(u, x) \land Cl(P, u)$
$\lor \exists r : R(r, x) \land P(r)$

BY ⟨1⟩9

⟨1⟩ QED
BY ⟨1⟩10

A property is equisatisfiable with its closure.
See also *SATClosureInit*

LEMMA *ClosureEquiSAT* $\triangleq$
ASSUME
TEMPORAL $P(\_)$
PROVE
$(\exists u : P(u)) \equiv \exists u : Cl(P, u)$
PROOF

⟨1⟩1. $(\exists u : P(u)) \Rightarrow \exists u : Cl(P, u)$

⟨2⟩1. ASSUME VARIABLE u
PROVE $P(u) \Rightarrow Cl(P, u)$
BY *ClosureImplied*

⟨2⟩ QED

BY $\langle 2 \rangle 1$

$\langle 1 \rangle 2.$ $(\exists\, u : \ Cl(P,\, u)) \ \Rightarrow \ \exists\, u : \ P(u)$

can also use: BY *ClosureSample*

$\langle 2 \rangle 1.$ $(\exists\, u : \ Cl(P,\, u))$

$$\equiv \exists\, u : \forall\, b : \ \lor \neg MustUnstep(b) \ \lor \exists\, r : \ \land P(r) \ \land \Box(b \Rightarrow (r = u))$$

BY DEF *Cl*

$\langle 2 \rangle 2.$ $(\exists\, u \ : \ Cl(P,\, u))$

$$\Rightarrow \exists\, u : \forall\, b : \ \lor \neg MustUnstep(b) \ \lor \exists\, r : \ P(r)$$

BY $\langle 2 \rangle 1$

$\langle 2 \rangle 3.$ $\exists\, q : \ MustUnstep(q)$

BY DEF *MustUnstep*

$\langle 2 \rangle 4.$ $(\exists\, u \ : \ Cl(P,\, u))$

$$\Rightarrow \exists\, u : \ \land \exists\, q : \ MustUnstep(q) \ \land \forall\, b : \ \lor \neg MustUnstep(b) \ \lor \exists\, r : \ P(r)$$

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

$\langle 2 \rangle 5.$ $(\exists\, u : \ Cl(P,\, u))$

$$\Rightarrow \exists\, u,\, q : \ \land MustUnstep(q) \ \land \forall\, b : \ \lor \neg MustUnstep(b) \ \lor \exists\, r : \ P(r)$$

BY $\langle 2 \rangle 4$

$\langle 2 \rangle 6.$ $(\exists\, u : \ Cl(P,\, u))$

$$\Rightarrow \exists\, u,\, q : \ \land MustUnstep(q) \ \land \ \lor \neg MustUnstep(q) \ \lor \exists\, r : \ P(r)$$

BY $\langle 2 \rangle 5$ DEF $\forall$

$\langle 2 \rangle 7.$ $(\exists\, u : \ Cl(P,\, u))$

$$\Rightarrow \exists\, u,\, q : \ \exists\, r : \ P(r)$$

BY $\langle 2 \rangle 6$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 7$

$\langle 1 \rangle$ QED

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

Properties that relate *PrefixSat* and *PrefixPlusOne* to closure.

LEMMA *PrefixSatOfClosure* $\triangleq$

ASSUME

NEW $sigma$, $IsABehavior(sigma)$,
NEW $n \in Nat$,
TEMPORAL P

PROVE
$PrefixSat(sigma, n, P) \Rightarrow PrefixSat(sigma, n, Cl(P))$

PROOF
$\langle 1 \rangle 1.\ P \Rightarrow Cl(P)$
BY $ClosureImplied$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 1$, $PrefixSatImp$

LEMMA $PrefixPlusOneEquivWhilePlusOfClosures \triangleq$
ASSUME
TEMPORAL E,
TEMPORAL M
PROVE
$PrefixPlusOne(E, M) \quad \equiv \quad (Cl(E) \overset{+}{\rhd} Cl(M))$
OMITTED *TODO*

LEMMA $WhilePlusOfClosures \triangleq$ I think this proof holds also in $RTLA+$
ASSUME
TEMPORAL A,
TEMPORAL G
PROVE
$(Cl(E) \overset{+}{\rhd} Cl(M)) \quad \equiv \quad PrefixPlusOne(Cl(E), Cl(M))$
$\langle 1 \rangle$ DEFINE
$CE \triangleq Cl(E)$
$CM \triangleq Cl(M)$
$\langle 1 \rangle 1.\ (Cl(Ec) \overset{+}{\rhd} Cl(Mc)) \quad \equiv \quad PrefixPlusOne(Ec, Mc)$
BY $PrefixPlusOneEquivWhilePlusOfClosures$
$\langle 1 \rangle 2.\ \land Cl(Ec) \equiv Ec$
$\land Cl(Mc) \equiv Mc$
BY $ClosureIdempotent$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

Properties of stepwise operators.

For brevity, this section uses the semantic closure operator $Cl(P(\_))$, instead of the syntactic operator $Cl(P(\_), x)$. These two operators yield the same result whenever property P depends on no variables other than its argument. Similar adaptations apply to other operators in this section.

In order for these conclusions to hold for other similar operators (*e.g.*, *WhilePlusHalf*), those operators should have the same basic properties, in particular *WhilePlusOfClosuresIsSafety* and *WhilePlusSafetyLivenessDecomp*.

The stepwise implication of safety properties is a safety property.
Equivalently, the stepwise implication of closures is a safety property.

[5, Lemma 1 on p. $A - 3$]

PROPOSITION *WhilePlusOfClosuresIsSafety* $\triangleq$
    ASSUME
        TEMPORAL A, TEMPORAL G
    PROVE
        LET $C \triangleq Cl(A) \overset{+}{\triangleright} Cl(G)$
        IN $IsSafety(C)$
    PROOF
    ⟨1⟩ DEFINE
        $ClA \triangleq Cl(A)$
        $ClG \triangleq Cl(G)$
        $C \triangleq Cl(A) \overset{+}{\triangleright} Cl(G)$
    ⟨1⟩1. SUFFICES $Cl(C) \equiv C$
        BY DEF *IsSafety*, C
    ⟨1⟩2. SUFFICES $Cl(C) \Rightarrow C$
        ⟨2⟩1. $C \Rightarrow Cl(C)$
            BY *ClosureImplied*
        ⟨2⟩ QED
            BY ⟨2⟩1, ⟨1⟩2
    ⟨1⟩3. SUFFICES
            ASSUME
                NEW $sigma$, $IsABehavior(sigma)$,
                $sigma \models Cl(C)$
            PROVE $sigma \models C$
        OBVIOUS
    ⟨1⟩4. SUFFICES
            ASSUME $\neg(sigma \models C)$
            PROVE FALSE
        OBVIOUS
    ⟨1⟩5. $\neg \forall n \in Nat :$
            $PrefixSat(sigma, n, ClA) \Rightarrow PrefixSat(sigma, n+1, ClG)$
        BY ⟨1⟩4 DEF $\overset{+}{\triangleright}$
    ⟨1⟩6. PICK $n \in Nat :$
            $PrefixSat(sigma, n, ClA) \land \neg PrefixSat(sigma, n+1, ClG)$
        BY ⟨1⟩5
    ⟨1⟩7. Any extension of sigma's n-prefix satisfies ClA.
        ASSUME
            NEW eta, $IsABehavior(eta)$,
            $Prefix(sigma, n) = Prefix(eta, n)$
        PROVE
            $PrefixSat(sigma, n, ClA) \equiv PrefixSat(eta, n, ClA)$

BY *EquivPrefixSatIfSamePrefix*

⟨1⟩8. No extension of sigma's $(n+1)$-prefix can satisfy *ClG*.

ASSUME

NEW *eta*, *IsABehavior*(*eta*),

Prefix(*sigma*, $n+1$) = *Prefix*(*eta*, $n+1$)

PROVE

PrefixSat(*sigma*, $n+1$, *ClG*) ≡ *PrefixSat*(*eta*, $n+1$, *ClG*)

BY *EquivPrefixSatIfSamePrefix*

⟨1⟩9.

ASSUME

NEW *eta*, *IsABehavior*(*eta*),

Prefix(*sigma*, $n+1$) = *Prefix*(*eta*, $n+1$)

PROVE

∧ *PrefixSat*(*eta*, *n*, *ClA*)

∧ ¬*PrefixSat*(*eta*, $n+1$, *ClG*)

⟨2⟩1. *Prefix*(*sigma*, *n*) = *Prefix*(*eta*, *n*)

BY ⟨1⟩9 DEF *Prefix*

⟨2⟩2. *PrefixSat*(*sigma*, *n*, *ClA*) ≡ *PrefixSat*(*eta*, *n*, *ClA*)

BY ⟨1⟩9, ⟨2⟩1, ⟨1⟩7

⟨2⟩3. *PrefixSat*(*sigma*, $n+1$, *ClG*) ≡ *PrefixSat*(*eta*, $n+1$, *ClG*)

BY ⟨1⟩9, ⟨1⟩8

⟨2⟩4. *PrefixSat*(*eta*, *n*, *ClA*)

BY ⟨2⟩2, ⟨1⟩6

⟨2⟩5. ¬*PrefixSat*(*eta*, $n+1$, *ClG*)

BY ⟨2⟩3, ⟨1⟩6

⟨2⟩ QED

BY ⟨2⟩4, ⟨2⟩5

⟨1⟩10. ∀ *eta* :

∨ ¬ ∧ *IsABehavior*(*eta*)

∧ *Prefix*(*sigma*, $n+1$) = *Prefix*(*eta*, $n+1$)

∨ ∧ *PrefixSat*(*eta*, *n*, *ClA*)

∧ ¬*PrefixSat*(*eta*, $n+1$, *ClG*)

BY ⟨1⟩9

⟨1⟩11. ∀ *eta* :

∨ ¬ ∧ *IsABehavior*(*eta*)

∧ *Prefix*(*sigma*, $n+1$) = *Prefix*(*eta*, $n+1$)

∨ ¬∀ $k \in$ *Nat* :

PrefixSat(*eta*, *k*, *ClA*) ⇒ *PrefixSat*(*eta*, $k+1$, *ClG*)

BY ⟨1⟩10

⟨1⟩12. ∀ *eta* :

∨ ¬ ∧ *IsABehavior*(*eta*)

∧ *Prefix*(*sigma*, $n+1$) = *Prefix*(*eta*, $n+1$)

∨ ¬(*eta* ⊨ *C*)

BY ⟨1⟩11 DEF *C*

⟨1⟩13. ¬∃ *eta* : ∧ *IsABehavior*(*eta*)

$\wedge$ *Prefix*(*sigma*, $n + 1$) = *Prefix*(*eta*, $n + 1$)
$\wedge$ *eta* $\models C$

BY $\langle 1 \rangle 12$

$\langle 1 \rangle 14$. $\neg$*PrefixSat*(*sigma*, *n*, *C*)

BY $\langle 1 \rangle 13$, *PrefixSatAsSamePrefix*

$\langle 1 \rangle 15$. $\neg$(*sigma* $\models$ *Cl*(*C*))

BY $\langle 1 \rangle 14$ DEF *Cl* The semantic definition of closure.

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 3$, $\langle 1 \rangle 15$ goal from $\langle 1 \rangle 4$

The open-system propery $A \stackrel{+}{\rhd} G$ is the conjunction of a safety and a liveness part. The safety part involves only closures, which is useful. The liveness part relates stepwise to logical implication ($\stackrel{+}{\rhd}$ to $\Rightarrow$).

That $Cl(A) \stackrel{+}{\rhd} Cl(G)$ is the safety part and $A \Rightarrow G$ the liveness part does not follow from this theorem, but from *WhilePlusSafetyLivenessDecomp*.

[5, Lemma 2 on p. $A - 3$]

THEOREM *WhilePlusAsConj* $\triangleq$

ASSUME

TEMPORAL *A*, TEMPORAL *G*

PROVE

$$A \stackrel{+}{\rhd} G \equiv \begin{array}{l} \wedge\, Cl(A) \stackrel{+}{\rhd} Cl(G) \\ \wedge\, A \Rightarrow G \end{array}$$

OMITTED

[5, Lemma 3 on p. $A - 3$]

PROPOSITION *StepwiseAntecedent* $\triangleq$

ASSUME

TEMPORAL *A*, TEMPORAL *G*

PROVE

$$(A \wedge (A \stackrel{+}{\rhd} G)) \Rightarrow G$$

OMITTED

PROPOSITION *StepwiseConsequent* $\triangleq$

ASSUME

TEMPORAL *A*, TEMPORAL *G*

PROVE

$$G \Rightarrow (A \stackrel{+}{\rhd} G)$$

OMITTED

Closure distributes over stepwise implication.

THEOREM *WhilePlusMachineClosedRepr* $\triangleq$

ASSUME

TEMPORAL *A*, TEMPORAL *G*

PROVE
$\qquad Cl(A \overset{+}{\rhd} G) \;\equiv\; (Cl(A) \overset{+}{\rhd} Cl(G))$
PROOF
⟨1⟩ DEFINE
$\qquad P \;\triangleq\; A \overset{+}{\rhd} G$
$\qquad C \;\triangleq\; Cl(A) \overset{+}{\rhd} Cl(G)$
⟨1⟩1. $Cl(P) \Rightarrow C$
⟨2⟩1. $P \;\Rightarrow C$
⟨3⟩1. $A \overset{+}{\rhd} G \equiv \;\wedge\; Cl(A) \overset{+}{\rhd} Cl(G)$
$\qquad\qquad \wedge\; A \Rightarrow G$
BY *WhilePlusAsConj*
⟨3⟩2. $A \overset{+}{\rhd} G \Rightarrow (Cl(A) \overset{+}{\rhd} Cl(G))$
BY ⟨3⟩1
⟨3⟩ QED
BY ⟨3⟩2 DEF P, C
⟨2⟩2. $Cl(P) \Rightarrow Cl(C)$
BY ⟨2⟩1, *ClosureIsMonotonic*
⟨2⟩3. $Cl(C) \equiv C$
⟨3⟩1. $IsSafety(C)$
BY *WhilePlusOfClosuresIsSafety* DEF C
⟨3⟩ QED
BY ⟨3⟩1, *ClosureOfSafety*
⟨2⟩ QED
BY ⟨2⟩2, ⟨2⟩3
⟨1⟩2. $C \Rightarrow Cl(P)$
⟨2⟩1. SUFFICES
ASSUME
NEW $sigma$, $IsABehavior(sigma)$,
$sigma \models C$
PROVE
$sigma \models Cl(P)$
OBVIOUS
⟨2⟩2.CASE $sigma \models Cl(A)$
⟨3⟩1. $sigma \models Cl(G)$
BY ⟨2⟩1, ⟨2⟩2, *StepwiseAntecedent* DEF C
⟨3⟩2. SUFFICES
ASSUME NEW $n \in Nat$
PROVE $PrefixSat(sigma, n, P)$
BY DEF Cl goal from ⟨2⟩1
⟨3⟩3. PICK tau : $\wedge\; IsABehavior(tau)$
$\qquad \wedge\; Prefix(tau, n) = Prefix(sigma, n)$
$\qquad \wedge\; tau \models G$
BY ⟨3⟩1, *PrefixSatAsSamePrefix*
⟨3⟩4. $tau \models A \overset{+}{\rhd} G$
BY ⟨3⟩3, *StepwiseConsequent*

$\langle 3 \rangle 5.\ \exists\, tau :\ \land\, IsABehavior(tau)$
$\qquad\qquad\qquad \land\, Prefix(tau,\ n) = Prefix(sigma,\ n)$
$\qquad\qquad\qquad \land\, tau \models P$

BY $\langle 3 \rangle 3,\ \langle 3 \rangle 4$ DEF P

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 5,\ PrefixSatAsSamePrefix$ goal from $\langle 2 \rangle 1$

$\langle 2 \rangle 3.$CASE $\neg sigma \models Cl(A)$

$\langle 3 \rangle 1.\ sigma \models Cl(A) \overset{+}{\triangleright} Cl(G)$

BY $\langle 2 \rangle 1$ DEF C

$\langle 3 \rangle 2.\ sigma \models A \Rightarrow G$

$\langle 4 \rangle 1.\ sigma \models \neg Cl(A)$

BY $\langle 2 \rangle 3$

$\langle 4 \rangle 2.\ (\neg Cl(A)) \Rightarrow \neg A$

BY $ClosureImplied$

$\langle 4 \rangle 3.\ sigma \models \neg A$

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

$\langle 4 \rangle$ QED

BY $\langle 4 \rangle 3$

$\langle 3 \rangle 3.\ sigma \models A \overset{+}{\triangleright} G$

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

$\langle 3 \rangle 4.\ sigma \models P$

BY $\langle 3 \rangle 3$ DEF P

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 4,\ ClosureImplied$

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle$ QED

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

A representation theorem.

THEOREM $WhilePlusSafetyLivenessDecomp\ \triangleq$

ASSUME

TEMPORAL A, TEMPORAL G

PROVE

LET

$AG \triangleq A \overset{+}{\triangleright} G$

$C \triangleq Cl(A) \overset{+}{\triangleright} Cl(G)$

IN

$\land\, SafetyPart(AG) \equiv C$

$\land\, LivenessPart(AG) \equiv (C \Rightarrow AG)$

PROOF

$\langle 1 \rangle 1.\ SafetyPart(A \overset{+}{\triangleright} G) \equiv Cl(A) \overset{+}{\triangleright} Cl(G)$

BY $WhilePlusMachineClosedRepr$ DEF $SafetyPart$

$\langle 1 \rangle 2.\ LivenessPart(A \overset{+}{\triangleright} G)$

$\equiv (Cl(A) \stackrel{+}{\triangleright} Cl(G)) \;\Rightarrow\; (A \stackrel{+}{\triangleright} G)$

BY ⟨1⟩1 DEF *LivenessPart*

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

THEOREM

ASSUME

TEMPORAL E, TEMPORAL M

PROVE

$Cl(E \stackrel{+}{\triangleright} M) \;\Rightarrow\; Cl(E \Rightarrow M)$

PROOF

⟨1⟩1. $Cl(E \stackrel{+}{\triangleright} M) \;\Rightarrow\; (Cl(E) \Rightarrow Cl(M))$

BY *WhilePlusSafetyLivenessDecomp*

⟨1⟩2. $(Cl(E) \Rightarrow Cl(M)) \;\Rightarrow\; Cl(E \Rightarrow M)$

BY *ClosureOfImpl*

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

Feedback sustains M.

THEOREM *WhilePlusFeedback* $\triangleq$

ASSUME

TEMPORAL M,

$\neg \models \neg M$ — M is satisfiable

PROVE

LET $C \triangleq Cl(M)$

IN $(C \stackrel{+}{\triangleright} C) \;\equiv\; C$

PROOF

⟨1⟩ DEFINE $C \triangleq Cl(M)$

⟨1⟩1. $C \;\Rightarrow\; (C \stackrel{+}{\triangleright} C)$

BY *StepwiseConsequent*

⟨1⟩2. $(C \stackrel{+}{\triangleright} C) \;\Rightarrow\; C$

⟨2⟩1. SUFFICES

ASSUME

NEW *sigma*, *IsABehavior*(*sigma*),

$sigma \models C \stackrel{+}{\triangleright} C$

PROVE

$sigma \models C$

OBVIOUS

⟨2⟩2. $\forall\, n \in Nat$:

$PrefixSat(sigma,\, n,\, C)$

$\Rightarrow PrefixSat(sigma,\, n+1,\, C)$

BY ⟨2⟩1, *WhilePlusProperties*

⟨2⟩3. $PrefixSat(sigma,\, 0,\, C)$

⟨3⟩1. $\neg \models \neg M$

OBVIOUS   BY *WhilePlusFeedback*!assumption

⟨3⟩2. $\exists\, tau :\ \wedge\ IsABehavior(tau)$
$\wedge\ tau \models M$

BY ⟨3⟩1

⟨3⟩ QED

BY ⟨3⟩2 DEF *PrefixSat*

⟨2⟩4. $\forall\, n \in Nat :\ PrefixSat(sigma,\ n,\ C)$

BY ⟨2⟩2, ⟨2⟩3, *NatInduction*

⟨2⟩5. $sigma \models Cl(C)$

BY ⟨2⟩4 DEF *Cl*   semantic DEF of closure

⟨2⟩ QED

BY ⟨2⟩5, *ClosureIdempotent* DEF *C*

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2 DEF *C*

Feeding the same temporal property as both arguments of the while-plus operator cancels out the liveness part of that property.

LEMMA *ErasingLiveness* $\triangleq$

ASSUME

TEMPORAL *M*

PROVE

$(M \overset{+}{\rhd} M)\ \equiv\ (Cl(M) \overset{+}{\rhd} Cl(M))$

PROOF

⟨1⟩1. $M \overset{+}{\rhd} M\ \equiv\ \wedge\ Cl(M) \overset{+}{\rhd} Cl(M)$
$\wedge\ M \Rightarrow M$

BY *WhilePlusAsConj*

⟨1⟩2. $M \Rightarrow M$

OBVIOUS

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

If M is satisfiable, then the while-plus property $M \overset{+}{\rhd} M$ is the closure $Cl(M)$ of M.

COROLLARY *ClosureViaWhilePlus* $\triangleq$

ASSUME

TEMPORAL M,

$\neg \models \neg M$

PROVE

$(M \overset{+}{\rhd} M)\ \equiv\ Cl(M)$

PROOF

⟨1⟩1. $(M \overset{+}{\rhd} M)\ \equiv\ (Cl(M) \overset{+}{\rhd} Cl(M))$

BY *ErasingLiveness*

⟨1⟩2. $(Cl(M) \overset{+}{\rhd} Cl(M))\ \equiv\ Cl(M)$

BY *WhilePlusFeedback*

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

An instance of the schema from [5, Lemma 5].

LEMMA *ConjoiningSafety* $\triangleq$
  ASSUME
    TEMPORAL A, TEMPORAL B, TEMPORAL C, TEMPORAL D,
    $\mathit{IsSafety}(A)$, $\mathit{IsSafety}(B)$, $\mathit{IsSafety}(C)$, $\mathit{IsSafety}(D)$
  PROVE
    $\lor \neg \land A \overset{+}{\triangleright} B$
    $\quad\quad \land C \overset{+}{\triangleright} D$
    $\lor (A \land B) \overset{+}{\triangleright} (C \land D)$
  OMITTED

THEOREM
  ASSUME
    TEMPORAL P, TEMPORAL Q,
    $\mathit{IsSafety}(P)$, $\mathit{IsSafety}(Q)$,
    $\neg \models \neg(P \land Q)$
  PROVE
    $\lor \neg \land P \overset{+}{\triangleright} Q$
    $\quad\quad \land Q \overset{+}{\triangleright} P$
    $\lor P \land Q$
  PROOF
  $\langle 1 \rangle 1.$ $\lor \neg \land P \overset{+}{\triangleright} Q$
    $\quad\quad \land Q \overset{+}{\triangleright} P$
    $\lor (P \land Q) \overset{+}{\triangleright} (Q \land P)$
    BY *ConjoiningSafety*
  $\langle 1 \rangle 2.$ $((P \land Q) \overset{+}{\triangleright} (Q \land P))$
    $\equiv (P \land Q)$
    $\langle 2 \rangle 1.$ $\neg \models \neg(P \land Q)$
      OBVIOUS
    $\langle 2 \rangle 2.$ $\mathit{IsSafety}(P \land Q)$
      OMITTED   The conjunction of safety properties is safety.
    $\langle 2 \rangle$ QED
      BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$, *WhilePlusFeedback*
  $\langle 1 \rangle$ QED
    BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$

If we weaken the first argument and strengthen the second argument of $\overset{+}{\triangleright}$, then the resulting open-system refines the open-system we started with.

THEOREM *RefinementOfWhilePlus* $\triangleq$
  ASSUME
    TEMPORAL A, TEMPORAL G,

TEMPORAL P, TEMPORAL R,
$\land P \Rightarrow A$
$\land G \Rightarrow R$

if each of A, G, P, R contains recurrence, then these are $GR(1)$ problems (via Klein-Pnueli). If each has $GR(1)$ liveness, then these are $GR(2)$ problems.

PROVE
$(A \overset{+}{\rhd} G) \;\Rightarrow\; (P \overset{+}{\rhd} R)$

PROOF

⟨1⟩1. $\land A \overset{+}{\rhd} G \equiv \land \mathit{PrefixPlusOne}(A, G)$
$\qquad\qquad\qquad\quad \land A \Rightarrow G$

$\land P \overset{+}{\rhd} R \equiv \land \mathit{PrefixPlusOne}(P, R)$
$\qquad\qquad\qquad\quad \land P \Rightarrow R$

BY DEF $\overset{+}{\rhd}$

⟨1⟩2. $(A \Rightarrow G) \;\Rightarrow\; (P \Rightarrow R)$
⟨2⟩1. SUFFICES $(P \land (A \Rightarrow G)) \Rightarrow R$
OBVIOUS
⟨2⟩2. $(P \land (A \Rightarrow G)) \;\Rightarrow\; G$
⟨3⟩1. $P \Rightarrow A$
OBVIOUS BY *RefinementOfWhilePlus*!assumption
⟨3⟩ QED
BY ⟨3⟩1
⟨2⟩3. $G \Rightarrow R$
OBVIOUS BY *RefinementOfWhilePlus*! assumption
⟨2⟩ QED
BY ⟨2⟩2, ⟨2⟩3 goal from ⟨2⟩1

⟨1⟩3. $\mathit{PrefixPlusOne}(A, G) \;\Rightarrow\; \mathit{PrefixPlusOne}(P, R)$
⟨2⟩1. SUFFICES
ASSUME
NEW $n \in \mathit{Nat}$,
NEW sigma, $\mathit{IsABehavior}(\mathit{sigma})$
PROVE
$\lor \neg(\mathit{PrefixSat}(\mathit{sigma}, n, A) \Rightarrow \mathit{PrefixSat}(\mathit{sigma}, n+1, G)$
$\lor \mathit{PrefixSat}(\mathit{sigma}, n, P) \Rightarrow \mathit{PrefixSat}(\mathit{sigma}, n+1, R)$
BY DEF *PrefixPlusOne*
⟨2⟩2. SUFFICES
ASSUME
$\land \mathit{PrefixSat}(\mathit{sigma}, n, A) \Rightarrow \mathit{PrefixSat}(\mathit{sigma}, n+1, G)$
$\land \mathit{PrefixSat}(\mathit{sigma}, n, P)$
PROVE
$\mathit{PrefixSat}(\mathit{sigma}, n+1, R)$
OBVIOUS goal from ⟨2⟩1
⟨2⟩3. $\mathit{PrefixSat}(\mathit{sigma}, n, A)$
⟨3⟩1. $\mathit{PrefixSat}(\mathit{sigma}, n, P)$

BY ⟨2⟩2

⟨3⟩2. $P \Rightarrow A$

OBVIOUS   BY *RefinementOfWhilePlus*! assumption

⟨3⟩3. $PrefixSat(sigma, n, P) \Rightarrow PrefixSat(sigma, n, A)$

⟨4⟩1. $IsABehavior(sigma)$

BY ⟨2⟩1

⟨4⟩ QED

BY ⟨4⟩1, ⟨3⟩2, *PrefixSatImp*

⟨3⟩ QED

BY ⟨3⟩1, ⟨3⟩3

⟨2⟩4. $PrefixSat(sigma, n+1, G)$

BY ⟨2⟩3, ⟨2⟩2

⟨2⟩ QED

⟨3⟩1. $G \Rightarrow R$

OBVIOUS   BY *RefinementOfWhilePlus*! assumption

⟨3⟩2. $PrefixSat(sigma, n+1, G) \Rightarrow PrefixSat(sigma, n+1, R)$

⟨4⟩1. $IsABehavior(sigma)$

BY ⟨2⟩1

⟨4⟩ QED

BY ⟨4⟩1, ⟨3⟩1, *PrefixSatImp*

⟨3⟩ QED

BY ⟨2⟩4, ⟨3⟩2   goal from ⟨2⟩2

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3

Proof of rewriting $\stackrel{+}{\triangleright}$ with safety as first argument.

PROPOSITION *WeakeningLivenessPreservesMachineClosure* $\triangleq$

ASSUME

TEMPORAL S, TEMPORAL L, TEMPORAL R,

$Cl(S \land L) \equiv S$

PROVE

$Cl(S \land (L \lor R)) \equiv S$

PROOF

⟨1⟩1. DEFINE

$Z \triangleq Cl(S \land (L \lor R))$

⟨1⟩2. $S \Rightarrow Z$

⟨2⟩1. $(S \land L) \Rightarrow (S \land (L \lor R))$

OBVIOUS

⟨2⟩2. $Cl(S \land L) \Rightarrow Cl(S \land (L \lor R))$

BY ⟨2⟩1, *ClosureIsMonotonic*

⟨2⟩3. $S \Rightarrow Cl(S \land (L \lor R))$

BY ⟨2⟩2

and *WeakeningLivenessPreservesMachineClosure*!assumption

⟨2⟩ QED

    BY ⟨2⟩3 DEF Z

⟨1⟩3. $Z \Rightarrow S$

    ⟨2⟩1. $(S \land (L \lor R)) \Rightarrow S$

        OBVIOUS

    ⟨2⟩2. $Cl(S \land (L \lor R)) \Rightarrow Cl(S)$

        BY ⟨2⟩1, *ClosureIsMonotonic*

    ⟨2⟩3. $Z \Rightarrow Cl(S)$

        BY ⟨2⟩2 DEF Z

    ⟨2⟩4. $Cl(S) \equiv S$

        ⟨3⟩1. $Cl(L \land S) \equiv S$

            OBVIOUS

            BY *WeakeningLivenessPreservesMachineClosure*!assumption

        ⟨3⟩2. $Cl(Cl(L \land S)) \equiv Cl(S)$

            BY ⟨3⟩1

        ⟨3⟩3. $Cl(Cl(L \land S)) \equiv Cl(L \land S)$

            BY *ClosureIdempotent*

        ⟨3⟩4. $Cl(L \land S) \equiv Cl(S)$

            BY ⟨3⟩2, ⟨3⟩3

        ⟨3⟩ QED

            BY ⟨3⟩1, ⟨3⟩4

    ⟨2⟩ QED

        BY ⟨2⟩3, ⟨2⟩4

⟨1⟩ QED

    BY ⟨1⟩1, ⟨1⟩2 DEF Z

A claim on p. 528 in [5].

THEOREM *RewritingWhilePlusWithSafetyArg1* $\triangleq$

    ASSUME

        TEMPORAL E, TEMPORAL M

    PROVE

        LET

            $ENew \triangleq Cl(E)$

            $MNew \triangleq Cl(M) \land (E \Rightarrow M)$

        IN

            $\land (E \overset{+}{\rhd} M) \equiv (ENew \overset{+}{\rhd} MNew)$

            $\land IsSafety(ENew)$

    PROOF

    ⟨1⟩ DEFINE

        $EM \triangleq E \overset{+}{\rhd} M$

        $ENew \triangleq Cl(E)$

        $MNew \triangleq Cl(M) \land (E \Rightarrow M)$

    ⟨1⟩1. $EM \equiv \land Cl(E) \overset{+}{\rhd} Cl(M)$

                $\land E \Rightarrow M$

BY $\mathit{WhilePlusAsConj}$

$\langle 1\rangle 2.\ EM \equiv \land Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M)$
$\qquad \land Cl(E) \Rightarrow Cl(M)$
$\qquad \land E \Rightarrow M$

$\langle 2\rangle 1.\ (Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M))$
$\qquad \equiv \land Cl(Cl(E)) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(Cl(M))$
$\qquad\quad \land Cl(E) \Rightarrow Cl(M)$

BY $\mathit{WhilePlusAsConj}$

$\langle 2\rangle 2. \lor \neg(Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M))$
$\qquad \lor Cl(E) \Rightarrow Cl(M)$

BY $\langle 2\rangle 1$

$\langle 2\rangle$ QED

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

$\langle 1\rangle 3.\ EM \equiv \land Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M)$
$\qquad \land Cl(E) \Rightarrow Cl(M)$
$\qquad \land (Cl(E) \land E) \Rightarrow M$

$\langle 2\rangle 1.\ E \Rightarrow Cl(E)$

BY $\mathit{ClosureImplied}$

$\langle 2\rangle 2.\ E \equiv (E \land Cl(E))$

BY $\langle 2\rangle 1$

$\langle 2\rangle$ QED

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

$\langle 1\rangle 4.\ EM \equiv \land Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M)$
$\qquad \land Cl(E) \Rightarrow Cl(M)$
$\qquad \land Cl(E) \Rightarrow (E \Rightarrow M)$

BY $\langle 1\rangle 3$

$\langle 1\rangle 5.\ EM \equiv \land Cl(E) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(M)$
$\qquad \land Cl(E) \Rightarrow \land Cl(M)$
$\qquad\qquad\qquad \land E \Rightarrow M$

BY $\langle 1\rangle 4$

$\langle 1\rangle 6.\ EM \equiv \land Cl(Cl(E)) \stackrel{+}{\rightarrow\!\!\triangleright} Cl(Cl(M) \land (E \Rightarrow M))$
$\qquad \land Cl(E) \Rightarrow \land Cl(M)$
$\qquad\qquad\qquad \land E \Rightarrow M$

$\langle 2\rangle 1.\ Cl(E) \equiv Cl(Cl(E))$

BY $\mathit{ClosureIdempotent}$

$\langle 2\rangle 2.\ Cl(M) \equiv Cl(Cl(M) \land (E \Rightarrow M))$

In words: The pair M, $Cl(M)$ is machine-closed.

$\langle 3\rangle 1.\ M \Rightarrow Cl(M)$

BY $\mathit{ClosureImplied}$

$\langle 3\rangle 2.\ M \equiv (Cl(M) \land M)$

BY $\langle 3\rangle 1$

$\langle 3\rangle 3.\ Cl(M) \equiv Cl(Cl(M) \land M)$

BY $\langle 3\rangle 2$

$\langle 3\rangle 4.\ Cl(M) \equiv Cl(Cl(M) \land (M \lor \neg E))$

BY $\langle 3\rangle 3$, $\mathit{WeakeningLivenessPreservesMachineClosure}$

with $S \triangleq Cl(M)$, $L \triangleq M$, $R \triangleq \neg E$

⟨3⟩ QED

BY ⟨3⟩4

⟨2⟩ QED

BY ⟨1⟩5, ⟨2⟩1, ⟨2⟩2

⟨1⟩7. $EM \equiv \land Cl(ENew) \overset{+}{\rhd} Cl(MNew)$
$\quad\land ENew \Rightarrow MNew$

BY ⟨1⟩6 DEF $ENew$, $MNew$

⟨1⟩8. $IsSafety(ENew)$

⟨2⟩1. $Cl(ENew) \equiv Cl(Cl(E))$

BY DEF $Enew$

⟨2⟩2. $Cl(Cl(E)) \equiv Cl(E)$

BY $ClosureIdempotent$

⟨2⟩3. $Cl(E) \equiv ENew$

BY DEF $ENew$

⟨2⟩4. $Cl(ENew) \equiv ENew$

BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3

⟨2⟩ QED

BY ⟨2⟩4 DEF $IsSafety$

⟨1⟩ QED

BY ⟨1⟩7, ⟨1⟩8 DEF EM, $ENew$, $MNew$

The liveness part is shifted to the second argument of $\overset{+}{\rhd}$.
A form of "saturation".

PROPOSITION

ASSUME

TEMPORAL E, TEMPORAL M

PROVE

$$E \overset{+}{\rhd} M \equiv \land Cl(E) \overset{+}{\rhd} Cl(M) \\ \land Cl(E) \Rightarrow (LivenessPart(E) \Rightarrow M)$$

PROOF

⟨1⟩ DEFINE

$EM \triangleq E \overset{+}{\rhd} M$

⟨1⟩1. $EM \equiv \land Cl(E) \overset{+}{\rhd} Cl(M)$
$\quad\land E \Rightarrow M$

BY $WhilePlusAsConj$

⟨1⟩2. $(E \Rightarrow M)$
$\quad\equiv ((E \land Cl(E)) \Rightarrow M)$

BY $ClosureImplied$

⟨1⟩3. $((E \land Cl(E)) \Rightarrow M)$
$\quad\equiv (Cl(E) \Rightarrow (E \Rightarrow M))$

OBVIOUS

⟨1⟩4. $(Cl(E) \Rightarrow (E \Rightarrow M))$
$\quad\equiv (Cl(E) \Rightarrow ((E \lor \neg Cl(E)) \Rightarrow M))$

```
        OBVIOUS
    ⟨1⟩5. LivenessPart(E) ≡ (Cl(E) ⇒ E)
        BY DEF LivenessPart
    ⟨1⟩ QED
        BY ⟨1⟩1, ⟨1⟩2, ⟨1⟩3, ⟨1⟩4, ⟨1⟩5
```

The raw version of *RuleINV1* from the module *TLAPS*.

```
THEOREM RuleRawINV1 ≜
    ASSUME
        STATE I, ACTION N,
        (I ∧ N) ⇒ I′
    PROVE
        (I ∧ □N) ⇒ □I
    OMITTED
```

MODULE *TemporalQuantification*

Proof rules for temporal quantifiers $\exists$, $\forall$ in TLA+.

References

[1] *L.* Lamport, The temporal logic of actions, *TOPLAS*, 1994 10.1145/177492.177726

[2] *L.* Lamport, Specifying systems, Addison-Wesley, 2002

EXTENDS *Naturals*, *NaturalsInduction*, *TLASemantics*

Proof rule *E*1 from [1, Fig.9 on p.905].

```
THEOREM RuleE1 ≜
    ASSUME
        TEMPORAL F(_),
        STATE f
    PROVE
        F(f) ⇒ (∃ x : F(x))
```

Proof rule (schema) for instantiating universal temporal quantification.

```
THEOREM InstantiateAA ≜
    ASSUME
        TEMPORAL F(_),
        STATE f
    PROVE
        (∀ x : F(x)) ⇒ F(f)
    PROOF
    ⟨1⟩1. SUFFICES
            ASSUME
                ¬((∀ x : F(x)) ⇒ F(f))
            PROVE
                FALSE
        OBVIOUS
    ⟨1⟩2. ∧ ∀ x : F(x)
          ∧ ¬F(f)
        BY ⟨1⟩1
    ⟨1⟩3. ¬∃ x : ¬F(x)
        ⟨2⟩1. ∀ x : F(x)
            BY ⟨1⟩2
        ⟨2⟩ QED
            BY ⟨2⟩1 DEF ∀    ∀ x: P ≜ ¬(∃ x: ¬P) [2, p.315]
    ⟨1⟩4. ∃ x : ¬F(x)
```

$\langle 2\rangle 1.\ \neg F(f)$
    BY $\langle 1\rangle 2$
$\langle 2\rangle 2.\ (\neg F(f)) \;\Rightarrow\; \exists\, x :\ \neg F(x)$
    BY $\mathit{RuleE1}$
$\langle 2\rangle$ QED
    BY $\langle 2\rangle 1,\ \langle 2\rangle 2$
$\langle 1\rangle$ QED
    BY $\langle 1\rangle 3,\ \langle 1\rangle 4$

THEOREM $\mathit{UniversalClosure} \triangleq$
ASSUME
    TEMPORAL $G(\_)$,
    ASSUME VARIABLE x
    PROVE $G(x)$
PROVE
    $\forall\, u :\ G(u)$
PROOF
$\langle 1\rangle 1.$ SUFFICES $\neg \exists\, u :\ \neg G(u)$
    BY $\langle 1\rangle 1$ DEF $\forall$
$\langle 1\rangle 2.$ SUFFICES
        ASSUME $\exists\, u :\ \neg G(u)$
        PROVE FALSE
    OBVIOUS   goal from $\langle 1\rangle 1$
$\langle 1\rangle 3.$ ASSUME VARIABLE u
    PROVE $G(u)$
    OBVIOUS   BY $\mathit{UniversalClosure}$!assumption
$\langle 1\rangle 4.\ \exists\, u :\ G(u) \land \neg G(u)$
    BY $\langle 1\rangle 2,\ \langle 1\rangle 3$
$\langle 1\rangle 5.\ \exists\, u :$ FALSE
    BY $\langle 1\rangle 4$
$\langle 1\rangle$ QED
    BY $\langle 1\rangle 5$

MODULE *WhilePlusTheorems*

We prove the equivalence of the "while-plus" operator $\overset{+}{-\triangleright}$ to a formula in raw TLA+ with the past operator *Earlier*. In other words, we convert $\overset{+}{-\triangleright}$ from TLA+ to a stepwise formula in raw TLA+ with past (*PastRTLA+*) that is more suitable for using synthesis algorithms originally developed for *LTL* [5]. The result that we formally prove is analogous to [4, Lemma *B*.1 on *p*.70].

Due to the past operator, the satisfaction relation $\models$ of *PastRTLA+* resembles that of *LTL* (it includes an index of the behavior state). So for *PastRTLA+* formulas we will use the notation

sigma, $i \models$ phi

and for TLA+ formulas the notation

sigma $\models$ phi

If phi is a TLA+ formula, then we can apply the equivalence

$(sigma \models \text{phi}) \equiv (\text{sigma}, 0 \models \text{phi})$

For the closure of a behavior sigma:

$(sigma \models Cl(F)) \equiv (sigma \models \forall n \in Nat: PrefixSat(sigma, n, F))$

In *PastRTLA+* we will allow writing [A]_v as shorthand for $A \lor (v = v')$. On its own, this expression is ungrammatical in TLA+.

The directive BY *Semantics* refers to *PastRTLA+* and TLA+ semantics.

Author: Ioannis *Filippidis*

References

[1] *M*. Abadi and *L*. Lamport, "Conjoining specifications", *TOPLAS*, 1995 10.1145/203095.201069

[2] *L*. Lamport, "Miscellany", 21 *April* 1991

[3] *M*. Abadi and *S*. Merz, "On TLA as a logic", Deductive Program Design, 1996

[4] *B*. Jonsson and *Y*.-K. Tsay, "Assumption/guarantee specifications in linear-time temporal logic", *TCS*, 1996 10.1016/0304 − 3975(96)00069 − 2

[5] *Y*.-K. Tsay, "Compositional verification in linear-time temporal logic", *FOSSACS*, 2000 10.1007/3 − 540 − 46432-8_23

[6] *U*. Klein and A. Pnueli, "Revisiting synthesis of *GR*(1) specifications", *HVC*, 2010 10.1007/978 − 3 − 642 − 19583-9_16

[7] *L*. Lamport, "Specifying concurrent program modules", *TOPLAS*, 1983 10.1145/69624.357207

[8] *K.L.* *McMillan*, "Circular compositional reasoning about liveness", *CHARME*, 1999, 10.1007/3 − 540 − 48153-2_30

[9] *K.S*. Namjoshi and *R.J*. Trefler, "On the completeness of compositional reasoning methods", *TOCL*, 2010 10.1145/1740582.1740584

EXTENDS *TLASemantics*, *TemporalLogic*, *Integers*, *TLAPS*

Definitions of past operators. For A an action, *UpToNow* corresponds to Historically in *LTL* and *Earlier* to *WeakPrevious* Historically. A different definition is needed when A is a temporal formula (using the Suffix operator), but we apply these operators to actions only.

$$sigma, i \models UpToNow(A) \triangleq \forall k \in 0 \,..\, i : \langle sigma[k], sigma[k+1]\rangle[[A]]$$

$$sigma, i \models Earlier(A) \triangleq \forall k \in 0 \,..\, (i-1) : \langle sigma[k], sigma[k+1]\rangle[[A]]$$

The definitions that work for A an arbitrary temporal formula.

$$sigma, i \models UpToNowTemporal(A) \triangleq \forall k \in 0 \,..\, i : Suffix(sigma, k), 0 \models A$$

$$sigma, i \models EarlierTemporal(A) \triangleq \forall k \in 0 \,..\, (i-1) : Suffix(sigma, k), 0 \models A$$

The syntactic definition of closure requires keeping track of variables, which is cumbersome. In this module we use the following semantic definition.

$$sigma \models Cl(P) \triangleq \forall n \in Nat : PrefixSat(sigma, n, P)$$

Incremental implication spread over a behavior. The operator *Earlier* is of *PastRTLA* + .

$$StepwiseImpl(EnvNext, SysNext) \triangleq \Box(Earlier(EnvNext) \Rightarrow SysNext)$$

Causal but not strictly [6]

$$WeakStepwiseImpl(EnvNext, SysNext) \triangleq \Box(UpToNow(EnvNext) \Rightarrow SysNext)$$

The "trianglelefteq" operator defined in [7, *p*.220].

$$sigma \models AsLongAs(P, Q) \triangleq \forall n \in Nat : (\forall m \in 0 \,..\, n : Suffix(sigma, m) \models P) \Rightarrow (Suffix(sigma, n) \models Q)$$

The "vartriangleleft" operator defined in [7, *p*.220]. The operators *OneStepLonger* and *PrefixPlusOne* are inequivalent.

$$sigma \models OneStepLonger(P, Q) \triangleq \forall n \in Nat : (\forall m \in 0 \,..\, (n-1) : Suffix(sigma, m) \models P) \Rightarrow (Suffix(sigma, n) \models Q)$$

The operator *OneStepLonger* can be expressed using the operator *AsLongAs*.

THEOREM ASSUME TEMPORAL P, Q
PROVE $OneStepLonger(P, Q) \equiv AsLongAs(Q \Rightarrow P, Q)$
PROOF OMITTED

An operator defined in [9, *p*.16:3] and slightly differently in [8].

$NotUntil(EnvNext, SysNext) \triangleq \neg Until(EnvNext, \neg SysNext)$

Comparing the definitions of *Lamport* [7], *Klein* and *Pnueli* [6], *MacMillan* [8], *Namjoshi* and *Trefler* [9].

THEOREM ASSUME ACTION E, S
PROVE $StepwiseImpl(E, S) \equiv OneStepLonger(E, S)$
PROOF OMITTED

THEOREM ASSUME ACTION E, S
PROVE $WeakStepwiseImpl(E, S) \equiv AsLongAs(E, S)$

THEOREM ASSUME ACTION E, S
PROVE $NotUntil(E, S) \equiv OneStepLonger(E, S)$
PROOF OMITTED

The *RawWhilePlus* operator is essentially the same with that studied by *Klein* and *Pnueli* [6]. The differences are in the strict causality and the initial condition (akin to comparing $\mathrel{-\!\triangleright}$ and $\overset{+}{\mathrel{-\!\triangleright}}$).

If the component initial condition *Is* constrains the initial value of component variables y, then use appropriate DEF of realizability.

If *SysNext* constrains x' (next env *var* values), then *RawWhilePlus* is unrealizable (for the same reason $\overset{+}{\mathrel{-\!\triangleright}}$ is unrealizable in that case). *LTL* synthesis literature passes *SysNext* that leaves x' unconstrained, so unrealizability does not arise there, but other issues do.

If *SysNext* results by rewriting a property as the conjunction of a machine-closed pair, then x' can happen to be constrained. If so, then unrealizability arises.

Any closed-system property G in $A \overset{+}{\mathrel{-\!\triangleright}} G$ has this issue (because the rewriting is always possible, and then the claims we prove apply). Only if G leaves x entirely unconstrained is unrealizability avoided. However, in that case G allows wild behavior within $PrefixSat(sigma, n, G)$.

$$
\begin{aligned}
&RawWhilePlus(\\
&\quad IeP(\_, \_), Ie, Is,\\
&\quad EnvNext, SysNext,\\
&\quad Le, Ls) \triangleq\\
&\quad \lor \neg\exists p, q : IeP(p, q) \quad \text{unsatisfiable assumption ?}\\
&\quad \lor \land Is\\
&\quad\quad \land \lor \neg Ie\\
&\quad\quad\quad \lor \land StepwiseImpl(EnvNext, SysNext)\\
&\quad\quad\quad\quad \land (\Box EnvNext \land Le) \Rightarrow Ls
\end{aligned}
$$

The *RawWhilePlus* operator offers 5 degrees of freedom, emphasized by the following canonical forms. The forms differ by whether the main operator is conjunction or disjunction.

$$
\begin{aligned}
&RawWhilePlusConj(InitA, InitB, EnvNext, SysNext, Liveness) \triangleq\\
&\quad \land InitB\\
&\quad \land InitA \Rightarrow \land StepwiseImpl(EnvNext, SysNext)
\end{aligned}
$$

$\land \lor \Diamond \neg EnvNext$
$\lor Liveness$

$RawWhilePlusDisj(InitC, InitD, EnvNext, SysNext, Liveness) \triangleq$
$InitC \Rightarrow \land InitD$
$\land StepwiseImpl(EnvNext, SysNext)$
$\land \lor \Diamond \neg EnvNext$
$\lor Liveness$

The operators *RawWhilePlusConj* and *RawWhilePlusDisj* can express the same properties, as shown by the following two theorems.

```
THEOREM
    ASSUME
        CONSTANT IeP(_, _),
        STATE Ie, STATE Is,
        ACTION EnvNext, ACTION SysNext,
        TEMPORAL Le, TEMPORAL Ls
    PROVE
        LET
            InitB ≜ (∃ p, q : IeP(p, q) ⇒ Is
            InitA ≜ Ie
            Liveness ≜ Le ⇒ Ls
        IN
            RawWhilePlusConj(InitA, InitB, EnvNext, SysNext, Liveness)
            ≡ RawWhilePlus(IeP, Ie, Is, EnvNext, SysNext, Le, Ls)
    PROOF OBVIOUS

THEOREM
    ASSUME
        CONSTANT IeP(_, _),
        STATE Ie, STATE Is,
        ACTION EnvNext, ACTION SysNext,
        TEMPORAL Le, TEMPORAL Ls,
    PROVE
        LET
            InitC ≜ ∧ ∃ p, q : IeP(p, q)
                    ∧ Is ⇒ Ie
            InitD ≜ Is
            Liveness ≜ Le ⇒ Ls
        IN
            RawWhilePlusDisj(InitC, InitD, EnvNext, SysNext, Liveness)
            ≡ RawWhilePlus(IeP, Ie, Is, EnvNext, SysNext, Le, Ls)
    PROOF OBVIOUS

PROPOSITION AlwaysSysNextImpliesStepwiseImpl ≜
```

$\lor \neg \Box SysNext$
$\lor StepwiseImpl(EnvNext, SysNext)$

PROOF

$\langle 1 \rangle 1.\ (\Box SysNext) \Rightarrow \Box(Earlier(EnvNext) \Rightarrow SysNext)$

BY *PTL*

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 1$ DEF *StepwiseImpl*

PROPOSITION *AlwaysEnvNextAndStepwiseImpl* $\triangleq$

$\lor \neg \land \Box EnvNext$
$\quad \land StepwiseImpl(EnvNext, SysNext)$
$\lor \Box SysNext$

PROOF

$\langle 1 \rangle 1.\ (\Box EnvNext) \Rightarrow \Box Earlier(EnvNext)$

BY DEF *Earlier*

$\langle 1 \rangle 2.\ \lor \neg \land \Box Earlier(EnvNext)$
$\quad \land \Box(Earlier(EnvNext) \Rightarrow SysNext)$
$\lor \Box SysNext$

BY *PTL*

$\langle 1 \rangle$ QED

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

Converting between *PastRTLA* + and TLA+.

The raw logic allows for stutter-sensitive properties, though the motivation for using the raw logic is to translate to a stepwise form and connect with results on fixpoint algorithms.

The satisfaction relation ($\models$) can be defined in two ways: with or without an explicit index of a state in the behavior (*i.e.*, *sigma* $\models P$ versus sigma, index $\models P$). TLA does not use such an index. An index is necessary to define past operators, because an index stores information from previous states in a behavior. We use an index, in order to include past operators.

There are two flavors of temporal quantification: one that preserves stutter-invariance (**∃**), and one that does not. The definition of **∃** in TLA and raw past TLA differ, because we are using $\models$ with an index. See the module *TemporalLogic* for how $|EE$ is defined in raw TLA with past.

LEMMA *CommonModels* $\triangleq$

ASSUME TEMPORAL F,
$\quad IsATLAPlusFormula(F)$

PROVE $(sigma, 0 \models F) \equiv (sigma \models F)$

PROOF

BY *Semantics*

Relating *PrefixSat* to closure.

```
LEMMA PrefixSatForClosure ≜
  ASSUME
    TEMPORAL P,
    NEW n ∈ Nat,
    NEW sigma,
    IsABehavior(sigma)
  PROVE
    PrefixSat(sigma, n, P) ≡ PrefixSat(sigma, n, Cl(P))
  PROOF
  ⟨1⟩1. PrefixSat(sigma, n, P)
          ≡ ∃ tau :  ∧ IsABehavior(tau)
                     ∧ ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
                     ∧ tau ⊨ P
    BY  DEF PrefixSat
  ⟨1⟩2. PrefixSat(sigma, n, Cl(P))
          ≡ ∃ tau :  ∧ IsABehavior(tau)
                     ∧ ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
                     ∧ tau ⊨ Cl(P)
    BY  DEF PrefixSat
  ⟨1⟩3. PrefixSat(sigma, n, P) ⇒ PrefixSat(sigma, n, Cl(P))
    ⟨2⟩1. P ⇒ Cl(P)
      BY ClosureImplied
    ⟨2⟩ QED
      BY ⟨1⟩1, ⟨2⟩1, ⟨1⟩2
  ⟨1⟩4. PrefixSat(sigma, n, Cl(P)) ⇒ PrefixSat(sigma, n, P)
    ⟨2⟩1. SUFFICES ASSUME PrefixSat(sigma, n, Cl(P))
                   PROVE  PrefixSat(sigma, n, P)
      OBVIOUS
    ⟨2⟩2. PICK tau :
            ∧  IsABehavior(tau)
            ∧  ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
            ∧  tau ⊨ Cl(P)
      BY ⟨2⟩1, ⟨1⟩2
    ⟨2⟩3. ∀ r ∈ Nat :  PrefixSat(tau, r, P)
      ⟨3⟩1. tau ⊨ Cl(P)
        BY ⟨2⟩2
      ⟨3⟩ QED
        BY ⟨3⟩1  DEF Cl   the semantic definition of closure
    ⟨2⟩4. PrefixSat(tau, n, P)
      BY ⟨2⟩3
    ⟨2⟩5. PICK eta :  ∧ IsABehavior(eta)
                      ∧ ∀ i ∈ 0 .. (n − 1) :  eta[i] = tau[i]
                      ∧ eta ⊨ P
      BY ⟨2⟩4  DEF PrefixSat
    ⟨2⟩6. ∀ i ∈ 0 .. (n − 1) :  eta[i] = sigma[i]
```

$\langle 3 \rangle 1.\ \land \forall i \in 0 \,..\, (n-1) : \ eta[i] = tau[i]$
$\quad \land \forall i \in 0 \,..\, (n-1) : \ tau[i] = sigma[i]$
BY $\langle 2 \rangle 2$, $\langle 2 \rangle 5$
$\langle 3 \rangle$ QED
BY $\langle 3 \rangle 1$
$\langle 2 \rangle 7.\ \land IsABehavior(eta)$
$\quad \land \forall i \in 0 \,..\, (n-1) : \ eta[i] = sigma[i]$
$\quad \land eta \models P$
BY $\langle 2 \rangle 5$, $\langle 2 \rangle 6$
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 7$, $\langle 1 \rangle 1$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 3$, $\langle 1 \rangle 4$

One direction of *PhiEquivRawPhi*.

PROPOSITION $RawPhiImpliesPhiStep11 \triangleq$
ASSUME
VARIABLE x, VARIABLE y,
NEW $sigma$, META NEW
$IsABehavior(sigma)$,
CONSTANT $IeP(\_, \_)$,
CONSTANT $IsP(\_, \_)$,
CONSTANT $NeP(\_, \_, \_, \_)$,
CONSTANT $NsP(\_, \_, \_, \_)$,
TEMPORAL Le, TEMPORAL Ls,
$\land \forall u, v : \ IeP(u, v) \in$ BOOLEAN
$\land \forall u, v : \ IsP(u, v) \in$ BOOLEAN
$\land \forall a, b, c, d : \ NeP(a, b, c, d) \in$ BOOLEAN
$\land \forall a, b, c, d : \ NsP(a, b, c, d) \in$ BOOLEAN ,
LET
$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$EnvNext \triangleq [Ne]_v$
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($
$IeP, Ie, Is,$
$EnvNext, SysNext, Le, Ls)$
IN
$sigma, 0 \models RawPhi$
PROVE

```
        LET
            v  ≜ ⟨x, y⟩
            Is  ≜ IsP(x, y)
            Ie  ≜ IeP(x, y)
            Ne  ≜ NeP(x, y, x', y')
            Ns  ≜ NsP(x, y, x', y')
            A  ≜ Ie ∧ □[Ne]_v ∧ Le
            G  ≜ Is ∧ □[Ns]_v ∧ Ls
        IN
            sigma ⊨ A ⇒ G
PROOF
⟨1⟩ DEFINE
    v  ≜ ⟨x, y⟩
    Is  ≜ IsP(x, y)
    Ie  ≜ IeP(x, y)
    Ne  ≜ NeP(x, y, x', y')
    Ns  ≜ NsP(x, y, x', y')
    A  ≜ Ie ∧ □[Ne]_v ∧ Le
    G  ≜ Is ∧ □[Ns]_v ∧ Ls
    EnvNext  ≜ [Ne]_v
    SysNext  ≜ [Ns]_v
    RawPhi  ≜ RawWhilePlus(
        IeP, Ie, Is,
        EnvNext, SysNext, Le, Ls)
⟨1⟩1. (sigma, 0 ⊨ A)  ≡  (sigma ⊨ A)
    ⟨2⟩1. IsATLAPlusFormula(A)
        BY  DEF A, Ie, Ne
    ⟨2⟩ QED
        BY ⟨2⟩1, CommonModels DEF A
⟨1⟩2. (sigma, 0 ⊨ G)  ≡  (sigma ⊨ G)
    ⟨2⟩1. IsATLAPlusFormula(G)
        BY  DEF G, Is, Ns
    ⟨2⟩ QED
        BY ⟨2⟩1, CommonModels DEF G
⟨1⟩3. SUFFICES ASSUME sigma, 0 ⊨ A
                PROVE  sigma ⊨ G
    ⟨2⟩1. (sigma ⊨ A ⇒ G)
            ≡ ((sigma ⊨ A) ⇒ (sigma ⊨ G))
       BY Semantics
    ⟨2⟩2.CASE ¬(sigma, 0 ⊨ A)
        ⟨3⟩1. ¬(sigma ⊨ A)
            BY ⟨2⟩2, ⟨1⟩1
        ⟨3⟩2. (sigma ⊨ A) ⇒ (sigma ⊨ G)
            BY ⟨3⟩1
        ⟨3⟩ QED
```

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

$\langle 2\rangle 3.$ CASE $sigma, 0 \models A$

$\langle 3\rangle 1.$ $sigma \models G$

BY $\langle 1\rangle 3$

$\langle 3\rangle 2.$ $(sigma \models A) \Rightarrow (sigma \models G)$

BY $\langle 3\rangle 1$

$\langle 3\rangle$ QED

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

$\langle 2\rangle$ QED

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

$\langle 1\rangle 4.$ $\land\ sigma \models Ie \land \Box[Ne]_v \land Le$

$\land\ sigma, 0 \models Ie \land \Box[Ne]_v \land Le$

$\langle 2\rangle 1.$ $sigma \models A$

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

$\langle 2\rangle 2.$ $sigma \models Ie \land \Box[Ne]_v \land Le$

BY $\langle 2\rangle 1$ DEF A

$\langle 2\rangle 3.$ $sigma, 0 \models Ie \land \Box[Ne]_v \land Le$

BY $\langle 2\rangle 2$, $CommonModels$ DEF Ie, Ne

$\langle 2\rangle$ QED

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

$\langle 1\rangle 5.$ $\exists\, p, q : \ IeP(p, q)$ The assumption is satisfiable.

$\langle 2\rangle 1.$ $sigma \models Ie$

BY $\langle 1\rangle 4$

$\langle 2\rangle 2.$ $sigma \models IeP(x, y)$

BY $\langle 2\rangle 1$ DEF Ie

$\langle 2\rangle$ QED

BY $\langle 2\rangle 2$

$\langle 1\rangle 6.$ $sigma, 0 \models \ \land Is$

$\land\ StepwiseImpl(EnvNext, SysNext)$

$\land\ (Le \land \Box EnvNext) \Rightarrow Ls$

$\langle 2\rangle 1.$ $sigma, 0 \models \ \land Ie$

$\land\ \exists\, p, q : \ IeP(p, q)$

BY $\langle 1\rangle 4$, $\langle 1\rangle 5$

$\langle 2\rangle 2.$ $sigma, 0 \models$

$\lor\ \ \neg \exists\, p, q : \ IeP(p, q)$

$\lor\ \ \land Is$

$\land\ \lor \neg Ie$

$\lor\ \land StepwiseImpl(EnvNext, SysNext)$

$\land\ (\Box EnvNext \land Le) \Rightarrow Ls$

$\langle 3\rangle 1.$ $sigma, 0 \models RawPhi$

OBVIOUS BY $RawPhiImpliesPhiStep11$!assumption

$\langle 3\rangle$ QED

BY $\langle 3\rangle 1$ DEF $RawPhi$, $RawWhilePlus$

$\langle 2\rangle$ QED

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

$\langle 1 \rangle 7.\ sigma, 0 \models Ls$

$\langle 2 \rangle 1.\ sigma, 0 \models Le \land \Box EnvNext$

BY $\langle 1 \rangle 4$ DEF $EnvNext$

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle 8.\ sigma, 0 \models \Box[Ns]_v$

$\langle 2 \rangle 1.\ sigma, 0 \models \Box EnvNext$

BY $\langle 1 \rangle 4$ DEF $EnvNext$

$\langle 2 \rangle 2.\ sigma, 0 \models StepwiseImpl(EnvNext, SysNext)$

BY $\langle 1 \rangle 6$

$\langle 2 \rangle 3.\ sigma, 0 \models \Box SysNext$

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

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 3$ DEF $SysNext$

$\langle 1 \rangle 9.\ sigma, 0 \models G$

$\langle 2 \rangle 1.\ sigma, 0 \models Is \land \Box[Ns]_v \land Ls$

BY $\langle 1 \rangle 6$, $\langle 1 \rangle 7$, $\langle 1 \rangle 8$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 1$ DEF G

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 9$, $\langle 1 \rangle 2$ $\Rightarrow$ goal of $\langle 1 \rangle 3$

If the first $(n-1)$ steps of a behavior sigma satisfy the assumption $Ie \land \Box[Ne]\_v$, and (causal) stepwise implication holds of sigma, then the first n steps of sigma satisfy the guarantee $Is \land \Box[Ns]\_v$.

Note that such any TLA+ safety property (like $Ie \land \Box[Ne]\_v$) is stutter-extensible [4], so it suffices to talk about the first $(n-1)$ steps, as opposed of the first n states. The n-th state matters only for the last step. The property $Ie \land \Box[Ne]\_v$ can be satisfied by any n-th state, by stuttering forever.

LEMMA $TakeOneMoreStep \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

NEW $sigma$, META NEW

$IsABehavior(sigma)$,

NEW $n \in Nat$,

CONSTANT $NeP(\_, \_, \_, \_)$,

CONSTANT $NsP(\_, \_, \_, \_)$,

$\land \forall a, b, c, d : NeP(a, b, c, d)$

$\land \forall a, b, c, d : NsP(a, b, c, d)$,

LET

$v \triangleq \langle x, y \rangle$

$Ne \triangleq NeP(x, y, x', y')$

$Ns \triangleq NsP(x, y, x', y')$

$EnvNext \triangleq [Ne]_v$

$SysNext \triangleq [Ns]_v$

IN

$\land\ PrefixSat(sigma,\ n,\ \Box[Ne]_v)$

$\land\ sigma, 0 \models StepwiseImpl(EnvNext,\ SysNext)$

PROVE

LET

$v \triangleq \langle x,\ y\rangle$

$Ns \triangleq NsP(x,\ y,\ x',\ y')$

IN

$\forall\, r \in 0\,..\,(n-1):$

$\langle sigma[r],\ sigma[r+1]\rangle[[[Ns]_v]]$

PROOF

$\langle 1\rangle$ DEFINE

$v \triangleq \langle x,\ y\rangle$

$Is \triangleq IsP(x,\ y)$

$Ie \triangleq IeP(x,\ y)$

$Ne \triangleq NeP(x,\ y,\ x',\ y')$

$Ns \triangleq NsP(x,\ y,\ x',\ y')$

$EnvNext \triangleq [Ne]_v$

$SysNext \triangleq [Ns]_v$

$PlusOne \triangleq Earlier(EnvNext) \Rightarrow SysNext)$

Behavior sigma's first $(n-1)$ steps of sigma satisfy *EnvNext*.

$\langle 1\rangle 1.$ ASSUME NEW $k \in 0\,..\,(n-2)$

PROVE $\langle sigma[k],\ sigma[k+1]\rangle[[EnvNext]]$

$\langle 2\rangle 1.$ $PrefixSat(sigma,\ n,\ Ie \land \Box[Ne]_v)$

OBVIOUS BY *TakeOneMoreStep*!assumption

$\langle 2\rangle 2.$ PICK $tau :\ \land\ IsABehavior(tau)$

$\land\ \forall\, i \in 0\,..\,(n-1):\ tau[i] = sigma[i]$

$\land\ tau \models Ie \land \Box[Ne]_v$

BY $\langle 2\rangle 1$ DEF *PrefixSat*

$\langle 2\rangle 3.$ ASSUME NEW $i \in Nat$

PROVE $\langle tau[i],\ tau[i+1]\rangle[[[Ne]_v]]$

BY $\langle 2\rangle 2$, *Semantics*

$\langle 2\rangle 4.$ $\langle tau[k],\ tau[k+1]\rangle = \langle sigma[k],\ sigma[k+1]\rangle$

$\langle 3\rangle 1.$ $\land\ k \in 0\,..\,(n-1)$

$\land\ (k+1) \in 0\,..\,(n-1)$

BY $\langle 1\rangle 1$

$\langle 3\rangle$ QED

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

$\langle 2\rangle$ QED

$\langle 3\rangle 1.$ $\langle tau[k],\ tau[k+1]\rangle[[EnvNext]]$

BY $\langle 1\rangle 1$ DEF *EnvNext*

$\langle 3\rangle$ QED

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

Convert to a statement that uses *Earlier*.

$\langle 1\rangle 2.$ ASSUME NEW $r \in 0\,..\,(n-1)$

PROVE $sigma, r \models Earlier(EnvNext)$

⟨2⟩1. SUFFICES ASSUME NEW $k \in 0 \,..\, (r-1)$
PROVE $\langle sigma[k], sigma[k+1] \rangle [[EnvNext]]$
BY DEF $Earlier$

⟨2⟩2. $k \in 0 \,..\, (n-2)$

⟨3⟩1. $(k \in Nat) \land (r \in Nat)$
BY ⟨2⟩1, ⟨1⟩2

⟨3⟩2. $(k \leq r-1) \land (r \leq n-1)$
BY ⟨2⟩1, ⟨1⟩2

⟨3⟩3. $k \leq n-2$
BY ⟨3⟩1, ⟨3⟩2

⟨3⟩ QED
BY ⟨3⟩1, ⟨3⟩3

⟨2⟩ QED
BY ⟨2⟩2, ⟨1⟩1

Plus one step for *SysNext*.

⟨1⟩3. ASSUME NEW $r \in Nat$,
$sigma, r \models Earlier(EnvNext)$
PROVE $\langle sigma[r], sigma[r+1] \rangle [[SysNext]]$

⟨2⟩1. $sigma, 0 \models \Box PlusOne$
BY DEF $StepwiseImpl$, $PlusOne$
and *TakeOneMoreStep*!assumption

⟨2⟩2. $\forall i \in Nat : \; sigma, i \models PlusOne$
BY ⟨2⟩1, *Semantics*

⟨2⟩3. $sigma, r \models PlusOne$
BY ⟨2⟩2, ⟨1⟩3

⟨2⟩4. $\lor \neg sigma, r \models Earlier(EnvNext)$
$\lor\ sigma, r \models SysNext$
BY ⟨2⟩3, *Semantics* DEF *PlusOne*

⟨2⟩5. $sigma, r \models SysNext$
BY ⟨2⟩4, ⟨1⟩3

⟨2⟩ QED
BY ⟨2⟩5, *Semantics*

⟨1⟩4. ASSUME NEW $r \in 0 \,..\, (n-1)$
PROVE $\langle sigma[r], sigma[r+1] \rangle [[SysNext]]$

⟨2⟩1. $sigma, r \models Earlier(EnvNext)$
BY ⟨1⟩2, ⟨1⟩4

⟨2⟩2. $r \in Nat :$
BY ⟨1⟩4

⟨2⟩ QED
BY ⟨2⟩1, ⟨2⟩2

⟨1⟩ QED
BY ⟨1⟩4 DEF *SysNext*

```
PROPOSITION RawPhiImpliesPhiStep12 ≜
  ASSUME
    VARIABLE x, VARIABLE y,
    NEW sigma,   META NEW
    IsABehavior(sigma),
    CONSTANT IeP(_, _),
    CONSTANT IsP(_, _),
    CONSTANT NeP(_, _, _, _),
    CONSTANT NsP(_, _, _, _),
    TEMPORAL Le, TEMPORAL Ls,
    ∧ ∀ u, v :  IeP(u, v) ∈ BOOLEAN
    ∧ ∀ u, v :  IsP(u, v) ∈ BOOLEAN
    ∧ ∀ a, b, c, d :  NeP(a, b, c, d) ∈ BOOLEAN
    ∧ ∀ a, b, c, d :  NsP(a, b, c, d) ∈ BOOLEAN ,
    LET
      v ≜ ⟨x, y⟩
      Is  ≜ IsP(x, y)
      Ie  ≜ IeP(x, y)
      Ne ≜ NeP(x, y, x', y')
      Ns ≜ NsP(x, y, x', y')
      EnvNext ≜ [Ne]_v
      SysNext ≜ [Ns]_v
      RawPhi ≜ RawWhilePlus(
        IeP, Ie, Is,
        EnvNext, SysNext, Le, Ls)
    IN
      ∧ IsMachineClosed(Ie ∧ □[Ne]_v, Le)
      ∧ IsMachineClosed(Is ∧ □[Ns]_v, Ls)
      ∧ sigma, 0 ⊨ RawPhi
  PROVE
    LET
      v ≜ ⟨x, y⟩
      Is  ≜ IsP(x, y)
      Ie  ≜ IeP(x, y)
      Ne ≜ NeP(x, y, x', y')
      Ns ≜ NsP(x, y, x', y')
      A ≜ Ie ∧ □[Ne]_v ∧ Le
      G ≜ Is ∧ □[Ns]_v ∧ Ls
    IN
      ∀ n ∈ Nat :
        PrefixSat(sigma, n, A) ⇒ PrefixSat(sigma, n + 1, G)
PROOF
⟨1⟩ DEFINE
  v ≜ ⟨x, y⟩
  Is  ≜ IsP(x, y)
```

$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$A \triangleq Ie \land \Box[Ne]_v \land Le$
$G \triangleq Is \land \Box[Ns]_v \land Ls$
$ClA \triangleq Cl(A)$
$ClG \triangleq Cl(G)$
$EnvNext \triangleq [Ne]_v$
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($
$\quad IeP, Ie, Is,$
$\quad EnvNext, SysNext, Le, Ls)$

$\langle 1\rangle 4.\ \land ClA \equiv (Ie \land \Box[Ne]_v)$
$\quad \land ClG \equiv (Is \land \Box[Ns]_v)$
BY DEF $ClA, ClG, A, G, IsMachineClosed$
and $RawPhiImpliesPhiStep12$!assumption

$\langle 1\rangle 8.\ sigma, 0 \models$
$\quad \lor\ \neg\exists p, q : IeP(p, q)$ unsatisfiable assumption ?
$\quad \lor\ \land Is$
$\quad\quad \land \lor \neg Ie$
$\quad\quad\quad \lor \land StepwiseImpl(EnvNext, SysNext)$
$\quad\quad\quad\quad \land (\Box EnvNext \land Le) \Rightarrow Ls$

$\langle 2\rangle 1.\ sigma, 0 \models RawPhi$
OBVIOUS BY $RawPhiImpliesPhiStep12$!assumption
$\langle 2\rangle$ QED
BY $\langle 2\rangle 1$ DEF $RawPhi, RawWhilePlus$

$\langle 1\rangle 1.$ SUFFICES ASSUME NEW $n \in Nat$
PROVE $PrefixSat(sigma, n, A) \Rightarrow PrefixSat(sigma, n+1, G)$
OBVIOUS

$\langle 1\rangle 2.$ SUFFICES $PrefixSat(sigma, n, ClA) \Rightarrow PrefixSat(sigma, n+1, ClG)$
$\langle 2\rangle 1.\ IsABehavior(sigma)$
OBVIOUS BY $RawPhiImpliesPhiStep12$!assumption
$\langle 2\rangle 2.\ IsTemporalLevel(A)$ META
BY DEF $A, Ie, Ne, IsTemporalLevel$
$\langle 2\rangle 3.\ IsTemporalLevel(G)$ META
BY DEF $G, Is, Ns, IsTemporalLevel$
$\langle 2\rangle 4.\ PrefixSat(sigma, n, A) \equiv PrefixSat(sigma, n, ClA)$
BY $\langle 1\rangle 1, \langle 2\rangle 1, \langle 2\rangle 2, PrefixSatForClosure$
$\langle 2\rangle 5.\ PrefixSat(sigma, n+1, G) \equiv PrefixSat(sigma, n+1, ClG)$
BY $\langle 1\rangle 1, \langle 2\rangle 1, \langle 2\rangle 3, PrefixSatForClosure$
$\langle 2\rangle$ QED
BY $\langle 1\rangle 2, \langle 2\rangle 4, \langle 2\rangle 5$

$\langle 1\rangle 3.$ SUFFICES ASSUME $PrefixSat(sigma, n, ClA)$
PROVE $PrefixSat(sigma, n+1, ClG)$
OBVIOUS

$\langle 1 \rangle 5.$ $\mathit{PrefixSat}(sigma,\ n,\ Ie \land \Box[Ne]_v)$
  BY $\langle 1 \rangle 3$, $\langle 1 \rangle 4$
$\langle 1 \rangle 6.$ SUFFICES $\mathit{PrefixSat}(sigma,\ n+1,\ Is \land \Box[Ns]_v)$
  BY $\langle 1 \rangle 4$
First we handle the initial conditions.
$\langle 1 \rangle 7.$ $\exists\, p,\, q\ \ IeP(p,\, q)$ *IeP* is satisfiable, so A is satisfiable.
  $\langle 2 \rangle 1.$ PICK $tau : \ \land\ \mathit{IsABehavior}(tau)$
    $\land\ \forall\, i \in 0\,..\,(n-1):\ tau[i] = sigma[i]$
    $\land\ tau \models Ie \land \Box[Ne]_v$
    BY $\langle 1 \rangle 5$ DEF *PrefixSat*
  $\langle 2 \rangle 2.$ $tau \models Ie$
    BY $\langle 2 \rangle 1$
  $\langle 2 \rangle 3.$ $tau \models IeP(x,\, y)$
    BY $\langle 2 \rangle 2$ DEF *Ie*
  $\langle 2 \rangle$ QED
    BY $\langle 2 \rangle 3$, *Semantics*
$\langle 1 \rangle 12.$ $sigma,\ 0 \models$
  $\land\ Is$
  $\land\ \lor\ \neg Ie$
    $\lor\ \land\ \mathit{StepwiseImpl}(EnvNext,\ SysNext)$
      $\land\ (\Box EnvNext \land Le) \Rightarrow Ls$
  BY $\langle 1 \rangle 8$, $\langle 1 \rangle 7$
$\langle 1 \rangle 9.$ ASSUME $n = 0$
  PROVE $\mathit{PrefixSat}(sigma,\ n+1,\ Is \land \Box[Ns]_v)$
  In this case satisfiability of the assumption suffices to prove that the consequent holds.
  $\langle 2 \rangle 1.$ SUFFICES $\mathit{PrefixSat}(sigma,\ 1,\ Is \land \Box[Ns]_v)$
    BY $\langle 1 \rangle 9$
  $\langle 2 \rangle 2.$ SUFFICES $\exists\, tau : \ \land\ \mathit{IsABehavior}(tau)$
    $\land\ tau[0] = sigma[0]$
    $\land\ tau \models Is \land \Box[Ns]_v$
    BY DEF *PrefixSat*
  $\langle 2 \rangle 3.$ $sigma[0] \models Is$
    BY $\langle 1 \rangle 12$
  $\langle 2 \rangle$ DEFINE $tau \triangleq [i \in Nat \mapsto sigma[0]]$
  $\langle 2 \rangle 4.$ $\mathit{IsAState}(sigma[0])$
    $\langle 3 \rangle 1.$ $\mathit{IsABehavior}(sigma)$
      OBVIOUS BY *RawPhiImpliesPhiStep*12!assumption
    $\langle 3 \rangle$ QED
      BY $\langle 3 \rangle 1$ DEF *IsABehavior*
  $\langle 2 \rangle 5.$ $\mathit{IsABehavior}(tau)$
    BY $\langle 2 \rangle 4$ DEF *tau*, *IsABehavior*
  $\langle 2 \rangle 6.$ $tau[0] = sigma[0]$
    BY DEF *tau*
  $\langle 2 \rangle 7.$ $tau[0] \models Is$

BY ⟨2⟩3, ⟨2⟩6

⟨2⟩8. $tau \models \Box[Ns]_v$

⟨3⟩1. ASSUME $i \in Nat$ all tau steps are stuttering

PROVE $tau[i+1] = tau[i]$

BY DEF tau

⟨3⟩2. ASSUME $i \in Nat,\ tau[i+1] = tau[i]$

PROVE $\langle tau[i],\ tau[i+1]\rangle[[[Ns]_v]]$

BY $Semantics$ stuttering step

⟨3⟩ QED

BY ⟨3⟩1, ⟨3⟩2, $Semantics$

⟨2⟩9. $tau \models Is \land \Box[Ns]_v$

BY ⟨2⟩7, ⟨2⟩8, $Semantics$

⟨2⟩ QED

BY ⟨2⟩5, ⟨2⟩6, ⟨2⟩9 WITNESS tau for goal of ⟨2⟩2

⟨1⟩10. SUFFICES ASSUME $n > 0$

PROVE $PrefixSat(sigma,\ n+1,\ Is \land \Box[Ns]_v)$

current goal from ⟨1⟩6

⟨2⟩1. $(n = 0) \lor (n > 0)$

BY ⟨1⟩1

⟨2⟩2. CASE $n = 0$

BY ⟨1⟩9

⟨2⟩3. CASE $n > 0$

BY ⟨1⟩10

⟨2⟩ QED

BY ⟨2⟩1, ⟨2⟩2, ⟨2⟩3

⟨1⟩15. $(n \in Nat) \land (n > 0)$

BY ⟨1⟩1, ⟨1⟩10

⟨1⟩11. $sigma, 0 \models Ie$

⟨2⟩1. PICK $tau :\ \land IsABehavior(tau)$

$\land \forall i \in 0 \,..\, (n-1) :\ tau[i] = sigma[i]$

$\land\ tau \models Ie \land \Box[Ne]_v$

BY ⟨1⟩5 DEF $PrefixSat$

⟨2⟩2. $tau[0] \models Ie$

BY ⟨2⟩1

⟨2⟩3. $sigma[0] = tau[0]$

BY ⟨2⟩1, ⟨1⟩15

⟨2⟩4. $sigma[0] \models Ie$

BY ⟨2⟩2, ⟨2⟩3

⟨2⟩ QED

⟨3⟩1. $IsStateLevel(Ie)$

BY DEF Ie

⟨3⟩ QED

BY ⟨2⟩4, ⟨3⟩1, $Semantics$

⟨1⟩13. $sigma, 0 \models\ \land Is$

$\land\ StepwiseImpl(EnvNext,\ SysNext)$

We omit the liveness conjunct, because irrelevant to this part of the proof, which concerns stepwise implication, thus only safety.

BY ⟨1⟩12, ⟨1⟩11

Done with initial conditions. We address below the stepwise implication.

⟨1⟩14. SUFFICES $\exists\, w : \; \land\, \mathit{IsABehavior}(w)$
$\land\, \forall\, i \in 0 \,..\, n : \; w[i] = \mathit{sigma}[i]$
$\land\, w \models \mathit{Is} \land \Box[\mathit{Ns}]_v$

current goal from ⟨1⟩10

⟨2⟩1. $\exists\, w : \; \land\, \mathit{IsABehavior}(w)$
$\land\, \forall\, i \in 0 \,..\, ((n+1)-1) : \; w[i] = \mathit{sigma}[i]$
$\land\, w \models \mathit{Is} \land \Box[\mathit{Ns}]_v$

BY ⟨1⟩14

⟨2⟩ QED

BY ⟨2⟩1 DEF *PrefixSat*

⟨1⟩ DEFINE $\mathit{eta} \triangleq \mathit{StutterAfter}(\mathit{sigma},\, n)$ Infinitely stuttering tail.

prove that *eta* is the WITNESS *w*

⟨1⟩16 $\mathit{eta} \models \Box[\mathit{Ns}]_v$

The first $(n-1)$ steps of *eta* satisfy the action [*Ns*]_*v*.

⟨2⟩1. ASSUME NEW $k \in 0 \,..\, (n-1)$
PROVE $\langle \mathit{eta}[k],\, \mathit{eta}[k \quad + 1]\rangle[[\mathit{SysNext}]]$

⟨3⟩1. $\langle \mathit{sigma}[k],\, \mathit{sigma}[k+1]\rangle[[\mathit{SysNext}]]$

⟨4⟩1. $\mathit{PrefixSat}(\mathit{sigma},\, n,\, \Box[\mathit{Ne}]_v)$

BY ⟨1⟩5 DEF *PrefixSat*

⟨4⟩2. $\mathit{sigma},\, 0 \models \mathit{StepwiseImpl}(\mathit{EnvNext},\, \mathit{SysNext})$

BY ⟨1⟩13

⟨4⟩ QED

BY ⟨4⟩1, ⟨4⟩2, *TakeOneMoreStep*

⟨3⟩2. $\langle \mathit{eta}[k],\, \mathit{eta}[k+1]\rangle = \langle \mathit{sigma}[k],\, \mathit{sigma}[k+1]\rangle$

⟨4⟩1. $\land\, k \in 0 \,..\, n$
$\land\, (k+1) \in 0 \,..\, n$

BY ⟨2⟩1

⟨4⟩2. $\forall\, i \in 0 \,..\, n : \; \mathit{eta}[i] = \mathit{sigma}[i]$

BY DEF *eta*, *StutterAfter*

⟨4⟩3. $\land\, \mathit{eta}[k] = \mathit{sigma}[k]$
$\land\, \mathit{eta}[k+1] = \mathit{sigma}[k+1]$

BY ⟨4⟩1, ⟨4⟩2

⟨4⟩ QED

BY ⟨4⟩3

⟨3⟩ QED

BY ⟨3⟩1, ⟨3⟩2

Steps of *eta* from the *n*-th onwards satisfy the action [*Ns*]_*v*.
These are stuttering steps, so this step's proof has no dependencies.

⟨2⟩2. ASSUME NEW $k \in \mathit{Nat},\, k \geq n$
PROVE $\langle \mathit{eta}[k],\, \mathit{eta}[k+1]\rangle[[\mathit{SysNext}]]$

$\langle 3 \rangle 1.\ \langle eta[k],\ eta[k+1] \rangle = \langle sigma[n],\ sigma[n] \rangle$

BY $\langle 2 \rangle 2$, *StutteringTail* DEF *eta*

$\langle 3 \rangle 2.\ \langle sigma[n],\ sigma[n] \rangle [[SysNext]]$

$\langle 4 \rangle 1.\ \langle sigma[n],\ sigma[n] \rangle [[v' = v]]$

BY DEF v

$\langle 4 \rangle$ QED

BY $\langle 4 \rangle 1$, *Semantics* DEF *SysNext*

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle 3.$ ASSUME NEW $k \in Nat$

PROVE $\langle eta[k],\ eta[k+1] \rangle [[SysNext]]$

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

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 3$, *Semantics* DEF *SysNext*

$\langle 1 \rangle 17.\ eta \models Is$

$\langle 2 \rangle 1.\ sigma,\ 0 \models Is$

BY $\langle 1 \rangle 13$

$\langle 2 \rangle 2.\ sigma[0] \models Is$

BY $\langle 2 \rangle 1$, *Semantics* DEF *Is*

$\langle 2 \rangle 3.\ eta[0] = sigma[0]$

$\langle 3 \rangle 1.\ n > 0$

BY $\langle 1 \rangle 10$

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 1$ DEF *eta*, *StutterAfter*

$\langle 2 \rangle 4.\ eta[0] \models Is$

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

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 4$, *Semantics* DEF *Is*

$\langle 1 \rangle 18.\ IsABehavior(eta)$

BY *StutterAfterIsABehavior*

$\langle 1 \rangle 19.$ ASSUME NEW $k \in 0 \,..\, n$

PROVE $eta[k] \quad = sigma[k]$

$\langle 2 \rangle 1.$CASE $k < n$

BY *StutterAfterHasSamePrefix* DEF *eta*

$\langle 2 \rangle 2.$CASE $k = n$

$\langle 3 \rangle 1.\ eta[k] = sigma[n]$

BY *StutteringTail*

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle 3.\ (k < n) \lor (k = n)$

BY $\langle 1 \rangle 19$

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle$ QED

$\langle 2 \rangle 1.\ \land IsABehavior(eta)$

$\land \forall i \in 0 \,..\, n : \; eta[i] = sigma[i]$
$\land \; eta \models Is \land \Box[Ns]_v$
BY ⟨1⟩18, ⟨1⟩19, ⟨1⟩17, ⟨1⟩16

⟨2⟩ QED
BY ⟨2⟩1 goal from ⟨1⟩14

LEMMA $RawPhiImpliesPhi \triangleq$

ASSUME
VARIABLE x, VARIABLE y,
NEW $sigma$, META NEW
$IsABehavior(sigma)$
CONSTANT $IeP(\_, \_)$,
CONSTANT $IsP(\_, \_)$,
CONSTANT $NeP(\_, \_, \_, \_)$,
CONSTANT $NsP(\_, \_, \_, \_)$,
TEMPORAL Le, TEMPORAL Ls,
$\land \forall u, v : \; IeP(u, v) \in$ BOOLEAN
$\land \forall u, v : \; IsP(u, v) \in$ BOOLEAN
$\land \forall a, b, c, d : \; NeP(a, b, c, d) \in$ BOOLEAN
$\land \forall a, b, c, d : \; NsP(a, b, c, d) \in$ BOOLEAN ,
LET
$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$EnvNext \triangleq [Ne]_v$
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($
$IeP, Ie, Is,$
$EnvNext, SysNext, Le, Ls)$
IN
$\land IsMachineClosed(Ie \land \Box[Ne]_v, Le)$
$\land IsMachineClosed(Is \land \Box[Ns]_v, Ls)$
$\land sigma, 0 \models RawPhi$

PROVE
LET
$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$A \triangleq Ie \land \Box[Ne]_v \land Le$
$G \triangleq Is \land \Box[Ns]_v \land Ls$

$Phi \triangleq A \stackrel{+}{\rhd} G$

IN

$sigma \models Phi$

PROOF

⟨1⟩ DEFINE

$v \triangleq \langle x, y \rangle$

$Is \triangleq IsP(x, y)$

$Ie \triangleq IeP(x, y)$

$Ne \triangleq NeP(x, y, x', y')$

$Ns \triangleq NsP(x, y, x', y')$

$A \triangleq Ie \land \Box[Ne]_v \land Le$

$G \triangleq Is \land \Box[Ns]_v \land Ls$

$Phi \triangleq A \stackrel{+}{\rhd} G$

$EnvNext \triangleq [Ne]_v$

$SysNext \triangleq [Ns]_v$

$RawPhi \triangleq RawWhilePlus($
$IeP, Ie, Is,$
$EnvNext, SysNext, Le, Ls)$

⟨1⟩ SUFFICES

$\land\ sigma \models A \Rightarrow G$

$\land\ \forall n \in Nat : \ PrefixSat(sigma, n, A)$
$\Rightarrow PrefixSat(sigma, n + 1, G)$

BY DEF $\stackrel{+}{\rhd}$, $PrefixPlusOne, A, G, Is, Ie, Ns, Ne$

⟨1⟩1. $sigma \models A \Rightarrow G$ The liveness part.

BY $RawPhiImpliesPhiStep11$

⟨1⟩2. $\forall n \in Nat :$ The safety part.

$PrefixSat(sigma, n, A) \Rightarrow PrefixSat(sigma, n + 1, G)$

BY $RawPhiImpliesPhiStep12$

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

The other direction of *PhiEquivRawPhi*.

LEMMA $PhiImpliesRawPhi \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

NEW $sigma$, META NEW

$IsABehavior(sigma)$,

CONSTANT $IeP(\_, \_)$,

CONSTANT $IsP(\_, \_)$,

CONSTANT $NeP(\_, \_, \_, \_)$,

CONSTANT $NsP(\_, \_, \_, \_)$,

TEMPORAL Le, TEMPORAL Ls,

$\land\ \forall u, v : \ IeP(u, v) \in$ BOOLEAN

$\wedge \forall u, v : IsP(u, v) \in$ BOOLEAN
$\wedge \forall a, b, c, d : NeP(a, b, c, d) \in$ BOOLEAN
$\wedge \forall a, b, c, d : NsP(a, b, c, d) \in$ BOOLEAN ,

LET

$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$A \triangleq Ie \wedge \Box[Ne]_v \wedge Le$
$G \triangleq Is \wedge \Box[Ns]_v \wedge Ls$
$Phi \triangleq A \stackrel{+}{\triangleright} G$

IN

$\wedge IsMachineClosed(Ie \wedge \Box[Ne]_v, Le)$
$\wedge IsMachineClosed(Is \wedge \Box[Ns]_v, Ls)$
$\wedge sigma \models Phi$

PROVE

LET

$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$EnvNext \triangleq [Ne]_v$
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($
$IeP, Ie, Is,$
$EnvNext, SysNext, Le, Ls)$

IN

$sigma, 0 \models RawPhi$

PROOF

$\langle 1 \rangle$ DEFINE

$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$A \triangleq Ie \wedge \Box[Ne]_v \wedge Le$
$G \triangleq Is \wedge \Box[Ns]_v \wedge Ls$
$Phi \triangleq A \stackrel{+}{\triangleright} G$
$ClA \triangleq Cl(A)$
$ClG \triangleq Cl(G)$
$EnvNext \triangleq [Ne]_v$
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($

IeP, Ie, Is,
$EnvNext$, $SysNext$, Le, Ls)

$\langle 1 \rangle 9.$ $\land\ ClA \equiv (Ie \land \Box[Ne]_v)$
$\land\ ClG \equiv (Is \land \Box[Ns]_v)$

$\langle 2 \rangle 1.$ $Cl(Ie \land \Box[Ne]_v \land Le) \equiv (Ie \land \Box[Ne]_v)$
BY DEF $IsMachineClosed$, Ie, Ne, v
and $PhiImpliesRawPhi$!assumption

$\langle 2 \rangle 2.$ $Cl(A) \equiv (Ie \land \Box[Ne]_v)$
BY $\langle 2 \rangle 1$ DEF A

$\langle 2 \rangle 3.$ $Cl(Is \land \Box[Ns]_v \land Ls) \equiv (Is \land \Box[Ns]_v)$
BY DEF $IsMachineClosed$, Is, Ns, v
and $PhiImpliesRawPhi$!assumption

$\langle 2 \rangle 4.$ $Cl(G) \equiv (Is \land \Box[Ns]_v)$
BY $\langle 2 \rangle 3$ DEF G

$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 2$, $\langle 2 \rangle 4$ DEF ClA, ClG

$\langle 1 \rangle 1.$ SUFFICES ASSUME $sigma, 0 \models \exists\, p, q : \ IeP(p, q)$
PROVE $sigma, 0 \models \ \land Is$
$\land \lor \neg Ie$
$\lor \land StepwiseImpl(EnvNext, SysNext)$
$\land (\Box EnvNext \land Le) \Rightarrow Ls$

BY DEF $RawWhilePlus$

$\langle 1 \rangle 2.$ $sigma, 0 \models Is$

$\langle 2 \rangle 1.$ $A \stackrel{+}{\rhd} G$
OBVIOUS BY $PhiImpliesRawPhi$!assumption

$\langle 2 \rangle 2.$ $PrefixSat(sigma, 0, A) \Rightarrow PrefixSat(sigma, 1, G)$
BY $\langle 2 \rangle 1$ DEF $\stackrel{+}{\rhd}$, $PrefixPlusOne$

$\langle 2 \rangle 3.$ $PrefixSat(sigma, 0, A)$

$\langle 3 \rangle 1.$ SUFFICES $PrefixSat(sigma, 0, ClA)$

$\langle 4 \rangle 1.$ $IsTemporalLevel(A)$
BY DEF A, Ie, Ne

$\langle 4 \rangle 2.$ $0 \in Nat$
OBVIOUS

$\langle 4 \rangle 3.$ $IsABehavior(sigma)$
OBVIOUS BY $PhiImpliesRawPhi$!assumption

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

$\langle 3 \rangle 2.$ SUFFICES $PrefixSat(sigma, 0, Ie \land \Box[Ne]_v)$
BY DEF ClA, $IsMachineClosed$
and $PhiImpliesRawPhi$!assumption

$\langle 3 \rangle 3.$ SUFFICES $\exists\, tau : \ \land IsABehavior(tau)$
$\land\ tau \models Ie \land \Box[Ne]_v$
BY DEF $PrefixSat$

$\langle 3 \rangle 4.$ PICK $p, q : \ IeP(p, q)$
BY $\langle 1 \rangle 1$

⟨3⟩5. DEFINE

$state \triangleq [var \in VarNames \mapsto \text{IF } var = \text{"x" THEN } p \text{ ELSE } q]$

$eta \triangleq Stutter(state)$

⟨3⟩6. $eta \models Ie$

⟨4⟩1. $eta[0] = state$

BY DEF eta, $Stutter$

⟨4⟩2. $\land\ state.x = p$

$\land\ state.y = q$

BY DEF $state$

⟨4⟩3. $state[[Ie]]$

BY ⟨4⟩2, $Semantics$ DEF $state$, Ie

⟨4⟩ QED

BY ⟨4⟩3, ⟨4⟩1, $Semantics$ DEF eta

⟨3⟩7. $eta \models \Box[Ne]_v$

⟨4⟩1. SUFFICES ASSUME NEW $i \in Nat$

PROVE $eta[i] \quad = eta[i+1]$

BY ⟨4⟩1, $Semantics$

⟨4⟩ QED

BY DEF eta, $Stutter$

⟨3⟩8. $eta \models Ie \land \Box[Ne]_v$

BY ⟨3⟩6, ⟨3⟩7

⟨3⟩9. $IsABehavior(eta)$

⟨4⟩1. $IsAState(state)$

BY DEF $state$, $IsAState$

⟨4⟩ QED

BY DEF eta, $Stutter$, $IsABehavior$

⟨3⟩ QED

BY ⟨3⟩8, ⟨3⟩9 goal from ⟨3⟩3

⟨2⟩5. PICK tau : $\land\ IsABehavior(tau)$

$\land\ \forall i \in 0 \,..\, (1-1) :\ tau[i] = sigma[i]$

$\land\ tau \models ClG$

⟨3⟩1. $PrefixSat(sigma, 1, G)$

BY ⟨2⟩2, ⟨2⟩3

⟨3⟩2. $PrefixSat(sigma, 1, ClG)$

BY ⟨3⟩1, $PrefixSatForClosure$ DEF ClG

⟨3⟩ QED

BY ⟨3⟩2 DEF $PrefixSat$

⟨2⟩6. $tau[0] = sigma[0]$

BY ⟨2⟩5

⟨2⟩7. $tau \models Is \land \Box[Ns]_v$

BY ⟨2⟩5 DEF $IsMachineClosed$, ClG, G

and $PhiImpliesRawPhi$!assumption

⟨2⟩8. $tau[0] \models Is$

BY ⟨2⟩7

⟨2⟩9. $sigma[0] \models Is$

BY $\langle 2\rangle 8$, $\langle 2\rangle 7$
$\langle 2\rangle$ QED
BY $\langle 2\rangle 9$, *Semantics*
$\langle 1\rangle 6$. SUFFICES ASSUME $sigma, 0 \models Ie$
PROVE $sigma, 0 \models \wedge StepwiseImpl(EnvNext, SysNext)$
$\wedge (\Box EnvNext \wedge Le) \Rightarrow Ls$

Previous goal from $\langle 1\rangle 1$

BY $\langle 1\rangle 2$, $\langle 1\rangle 6$
$\langle 1\rangle 7$. $sigma, 0 \models StepwiseImpl(EnvNext, SysNext)$ safety part
$\langle 2\rangle 1$. SUFFICES ASSUME NEW $n \in Nat$
PROVE $sigma, n \models Earlier(EnvNext) \Rightarrow SysNext)$
BY *Semantics* DEF *StepwiseImpl*
$\langle 2\rangle 2$. SUFFICES ASSUME $sigma, n \models Earlier(EnvNext)$
PROVE $sigma, n \models SysNext$
OBVIOUS
$\langle 2\rangle$ DEFINE $eta \triangleq StutterAfter(sigma, n)$
$\langle 2\rangle 6$. $IsABehavior(eta)$
BY $\langle 2\rangle 1$, *StutterAfterIsABehavior* DEF *eta*
$\langle 2\rangle 3$. $eta \models \Box[Ne]_v$
$\langle 3\rangle 1$. SUFFICES ASSUME NEW $k \in Nat$
PROVE $\langle eta[k], eta[k+1]\rangle [[[Ne]_v]]$
$\langle 4\rangle 1$. $IsATLAPlusFormula(\Box[Ne]_v)$
BY DEF *Ne*, *v*, *IsATLAPlusFormula*
$\langle 4\rangle$ QED
BY $\langle 3\rangle 1$, $\langle 4\rangle 1$, *Semantics*
$\langle 3\rangle 2$. $(n \in Nat) \wedge (k \in Nat)$
BY $\langle 2\rangle 1$, $\langle 3\rangle 1$
$\langle 3\rangle 3$. $IsABehavior(sigma)$
OBVIOUS BY *PhiImpliesRawPhi*!assumption
$\langle 3\rangle 4$.CASE $k < n$
$\langle 4\rangle 1$. $eta[k] = sigma[k]$
BY $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$, *StutterAfterHasSamePrefix* DEF *eta*
$\langle 4\rangle 2$. $eta[k+1] = sigma[k+1]$
$\langle 5\rangle 1$. $((k+1) \in Nat) \wedge (n \in Nat)$
BY $\langle 3\rangle 2$
$\langle 5\rangle 2$.CASE $(k+1) < n$
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 3\rangle 3$, *StutterAfterHasSamePrefix* DEF *eta*
$\langle 5\rangle 3$.CASE $(k+1) \geq n$
$\langle 6\rangle 1$. $eta[k+1] = sigma[n]$
BY $\langle 5\rangle 1$, $\langle 5\rangle 3$, $\langle 3\rangle 3$, *StutteringTail* DEF *eta*
$\langle 6\rangle 2$. $(k+1) = n$
$\langle 7\rangle 1$. $(k < n) \wedge (k+1) \geq n$
BY $\langle 3\rangle 4$, $\langle 5\rangle 3$
$\langle 7\rangle 2$. $(k+1) \leq n$
BY $\langle 7\rangle 1$, $\langle 5\rangle 1$

$\langle 7\rangle 3.\ \wedge (k+1 \leq n) \wedge (k+1 \geq n)$
$\wedge ((k+1) \in Nat) \wedge (n \in Nat)$
BY $\langle 7\rangle 1$, $\langle 7\rangle 2$, $\langle 5\rangle 1$
$\langle 7\rangle$ QED
BY $\langle 7\rangle 3$
$\langle 6\rangle$ QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$
$\langle 5\rangle$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 5\rangle 3$
$\langle 4\rangle 3.\ \langle eta[k], eta[k+1]\rangle = \langle sigma[k], sigma[k+1]\rangle$
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$
$\langle 4\rangle 4.\ \langle sigma[k], sigma[k+1]\rangle[[[Ne]_v]]$
$\langle 5\rangle 1.\ sigma, n \models Earlier([Ne]_v)$
BY $\langle 2\rangle 2$ DEF *EnvNext*
$\langle 5\rangle 2.\ \forall i \in 0 \mathrel{..} (n-1) :$
$\langle sigma[i], sigma[i+1]\rangle[[[Ne]_v]]$
BY DEF *Earlier*
$\langle 5\rangle 3.\ k \in 0 \mathrel{..} (n-1)$
BY $\langle 3\rangle 1$, $\langle 3\rangle 4$
$\langle 5\rangle$ QED
BY $\langle 5\rangle 2$, $\langle 5\rangle 3$
$\langle 4\rangle$ QED
BY $\langle 4\rangle 3$, $\langle 4\rangle 4$ goal from $\langle 3\rangle 1$
$\langle 3\rangle 5.$CASE $k \geq n$
$\langle 4\rangle 1.\ eta[k] = sigma[n]$
BY $\langle 3\rangle 2$, $\langle 3\rangle 5$, $\langle 3\rangle 3$, *StutteringTail* DEF *eta*
$\langle 4\rangle 2.\ eta[k+1] = sigma[n]$
$\langle 5\rangle 1.\ (k+1) \geq n$
BY $\langle 3\rangle 2$, $\langle 3\rangle 5$
$\langle 5\rangle 2.\ ((k+1) \in Nat) \wedge (n \in Nat)$
BY $\langle 3\rangle 2$
$\langle 5\rangle$ QED
BY $\langle 5\rangle 2$, $\langle 5\rangle 1$, $\langle 3\rangle 3$, *StutteringTail* DEF *eta*
$\langle 4\rangle 3.\ eta[k] = eta[k+1]$ A stuttering step.
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$
$\langle 4\rangle 4.\ \langle eta[k], eta[k+1]\rangle[[v' = v]]$
BY $\langle 4\rangle 3$, *Semantics* DEF *v*
$\langle 4\rangle$ QED
BY $\langle 4\rangle 4$, *Semantics* goal from $\langle 3\rangle 1$
$\langle 3\rangle$ QED
BY $\langle 3\rangle 2$, $\langle 3\rangle 4$, $\langle 3\rangle 5$
$\langle 2\rangle 4.\ PrefixSat(eta, n+1, Ie \wedge \Box[Ne]_v)$
$\langle 3\rangle 1.\ eta \models Ie \wedge \Box[Ne]_v$
$\langle 4\rangle 1.\ IsATLAPlusFormula(Ie)$
BY DEF *Ie*, *IsATLAPlusFormula*

$\langle 4\rangle 2.$ $sigma[0] \models Ie$

BY $\langle 1\rangle 6$, *Semantics* DEF *Ie*

$\langle 4\rangle 3.$ $eta[0] = sigma[0]$

BY *StutterAfterInit* DEF *eta*

$\langle 4\rangle 4.$ $eta[0] \models Ie$

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

$\langle 4\rangle 5.$ $eta \models Ie$

BY $\langle 4\rangle 1$, $\langle 4\rangle 4$, $\langle 2\rangle 6$, *Semantics* DEF *Ie*

$\langle 4\rangle$ QED

BY $\langle 4\rangle 5$, $\langle 2\rangle 3$

$\langle 3\rangle 2.$ $\land\ IsABehavior(eta)$
$\land\ \forall i \in 0 \,..\, ((n+1)-1):\ \ eta[i] = eta[i]$
$\land\ eta \models Ie \land \Box[Ne]_v$

BY $\langle 2\rangle 6$, $\langle 3\rangle 1$

$\langle 3\rangle$ QED

BY $\langle 3\rangle 2$ DEF *PrefixSat*

$\langle 2\rangle 5.$ $PrefixSat(sigma,\ n+1,\ A)$

$\langle 3\rangle 1.$ $(n+1) \in Nat$

BY $\langle 2\rangle 1$

$\langle 3\rangle 2.$ $PrefixSat(eta,\ n+1,\ ClA)$

BY $\langle 2\rangle 4$, $\langle 1\rangle 9$

$\langle 3\rangle 3.$ $PrefixSat(eta,\ n+1,\ A)$

Note that if *Le* is non-trivial, then *eta* may violate A, so we could not have used *eta* directly as the witness for $PrefixSat(eta,\ n{+}1,\ A)$, only for $PrefixSat(eta,\ n{+}1,\ ClA)$.

BY $\langle 3\rangle 1$, $\langle 3\rangle 2$, *PrefixSatForClosure* DEF *ClA*

$\langle 3\rangle 4.$ $Prefix(eta,\ n+1) = Prefix(sigma,\ n+1)$

$\langle 3\rangle$ QED

$\langle 4\rangle 1.$ $IsABehavior(sigma) \land IsABehavior(eta)$

BY $\langle 2\rangle 6$ and *PhiImpliesRawPhi*!assumption

$\langle 4\rangle 2.$ $IsTemporalLevel(A)$

BY DEF *A*, *Ie*, *Ne*, *v*

$\langle 4\rangle$ QED

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

$\langle 2\rangle 7.$ $PrefixSat(sigma,\ n+2,\ \Box[Ns]_v)$

$\langle 3\rangle 1.$ ASSUME NEW $r \in Nat$,
$PrefixSat(sigma,\ r,\ A)$
PROVE $PrefixSat(sigma,\ r+1,\ G)$

$\langle 4\rangle 1.$ $sigma \models A \overset{+}{\rhd} G$

OBVIOUS BY *PhiImpliesRawPhi*!assumption

$\langle 4\rangle 2.$ $sigma \models \forall k \in Nat:$
$\lor \neg PrefixSat(sigma,\ k,\ A)$
$\lor\ PrefixSat(sigma,\ k+1,\ G)$

BY $\langle 4\rangle 1$, *WhilePlusProperties*

$\langle 4\rangle$ QED

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

$\langle 3 \rangle 2.$ $PrefixSat(sigma,\ n+2,\ G)$

  $\langle 4 \rangle 1.$ $(n+1) \in Nat$

    BY $\langle 2 \rangle 1$

  $\langle 4 \rangle 2.$ $PrefixSat(sigma,\ (n+1)+1,\ G)$

    BY $\langle 4 \rangle 1$, $\langle 2 \rangle 5$, $\langle 3 \rangle 1$

  $\langle 4 \rangle 3.$ $(n+1)+1 = n+2$

    BY $\langle 2 \rangle 1$

  $\langle 4 \rangle$ QED

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

$\langle 3 \rangle 3.$ $PrefixSat(sigma,\ n+2,\ ClG)$

  $\langle 4 \rangle 1.$ $IsTemporalLevel(G)$

    BY DEF G, Is, Ns, v

  $\langle 4 \rangle 2.$ $(n+2) \in Nat$

    BY $\langle 2 \rangle 1$

  $\langle 4 \rangle 3.$ $IsABehavior(sigma)$

    OBVIOUS BY $PhiImpliesRawPhi$!assumption

  $\langle 4 \rangle$ QED

    BY $\langle 4 \rangle 1$, $\langle 4 \rangle 2$, $\langle 4 \rangle 3$, $PrefixSatForClosure$ DEF ClG

$\langle 3 \rangle$ QED

  BY $\langle 3 \rangle 3$, $\langle 1 \rangle 9$

$\langle 2 \rangle 8.$ $\langle sigma[n],\ sigma[n+1] \rangle [[[Ns]_v]]$

$\langle 3 \rangle 1.$ PICK $tau :$ $\land\ IsABehavior(tau)$
$\land\ \forall\, i \in 0\,..\,((n+2)-1) :\ \ tau[i] = sigma[i]$
$\land\ tau \models \Box[Ns]_v$

  BY $\langle 2 \rangle 7$ DEF $PrefixSat$

$\langle 3 \rangle 2.$ $\land\ n \in Nat$
$\land\ (n+1) \in Nat$

  BY $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $\land\ tau[n] = sigma[n]$
$\land\ tau[n+1] = sigma[n+1]$

  $\langle 4 \rangle 1.$ $\forall\, i \in 0\,..\,(n+1) :\ \ tau[i] = sigma[i]$

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

  $\langle 4 \rangle$ QED

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

$\langle 3 \rangle 4.$ $\langle tau[n],\ tau[n+1] \rangle [[[Ns]_v]]$

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

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle$ QED

$\langle 3 \rangle 1.$ $IsABehavior(sigma)$

  OBVIOUS BY $PhiImpliesRawPhi$!assumption

$\langle 3 \rangle 2.$ $n \in Nat$

  BY $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $sigma,\ n \models [Ns]_v$

  BY $\langle 2 \rangle 8$, $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, $Semantics$ DEF Ns, v

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 3$ DEF *SysNext* goal from $\langle 2 \rangle 2$

$\langle 1 \rangle 8.$ $sigma, 0 \models (\Box EnvNext \land Le) \Rightarrow Ls$ liveness part

$\langle 2 \rangle 1.$ SUFFICES ASSUME $sigma, 0 \models Le \land \Box EnvNext$

PROVE $sigma, 0 \models Ls$

BY *Semantics*

$\langle 2 \rangle 2.$ $sigma, 0 \models A$

$\langle 3 \rangle 1.$ $sigma, 0 \models Ie \land Le \land \Box EnvNext$

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

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 1$ DEF A, *EnvNext*

$\langle 2 \rangle 3.$ $sigma, 0 \models A \Rightarrow G$

$\langle 3 \rangle 1.$ $sigma \models A \overset{+}{\rightarrowtriangle} G$

OBVIOUS BY *PhiImpliesRawPhi*!assumption

$\langle 3 \rangle 2.$ $sigma \models A \Rightarrow G$

BY $\langle 3 \rangle 1$ DEF $\overset{+}{\rightarrowtriangle}$

$\langle 3 \rangle 3.$ $IsATLAPlusFormula(A \Rightarrow G)$

BY DEF A, G, Ie, Is, Ne, Ns, v

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 2$, $\langle 3 \rangle 3$, *CommonModels*

$\langle 2 \rangle 4.$ $sigma, 0 \models G$

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

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 4$ DEF G

$\langle 1 \rangle$ QED

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

This theorem proves that the solvers synthesize open-system TLA+ specs, whenever the pairs happen to be machine-closed, and Ns does not mention x'. If Ns does mention x' then the property resulting from $\overset{+}{\rightarrowtriangle}$ can be unrealizable (unless $\forall u$: $Ns(x, y, u, y')$ is not FALSE).

THEOREM *PhiEquivRawPhi* $\triangleq$

ASSUME

VARIABLE x, VARIABLE y,

NEW *sigma*, META NEW

$IsABehavior(sigma)$,

CONSTANT $IeP(\_, \_)$, The suffix "P" stands for "parametric".

CONSTANT $IsP(\_, \_)$,

CONSTANT $NeP(\_, \_, \_, \_)$,

CONSTANT $NsP(\_, \_, \_, \_)$,

TEMPORAL Le, TEMPORAL Ls, thus TLA+ formulas

$\land \forall u, v : IeP(u, v) \in$ BOOLEAN

$\land \forall u, v : IsP(u, v) \in$ BOOLEAN

$\land \forall a, b, c, d : NeP(a, b, c, d) \in$ BOOLEAN

$\land \forall a, b, c, d : NsP(a, b, c, d) \in$ BOOLEAN ,
LET
$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
IN
$\land IsMachineClosed(Ie \land \Box[Ne]_v, Le)$
$\land IsMachineClosed(Is \land \Box[Ns]_v, Ls)$
PROVE
LET
$v \triangleq \langle x, y \rangle$
$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$A \triangleq Ie \land \Box[Ne]_v \land Le$
$G \triangleq Is \land \Box[Ns]_v \land Ls$
$Phi \triangleq A \overset{+}{\rhd} G$
$EnvNext \triangleq [Ne]_v$ *RTLA+* but not TLA+ expression
$SysNext \triangleq [Ns]_v$
$RawPhi \triangleq RawWhilePlus($
$IeP, Ie, Is,$
$EnvNext, SysNext, Le, Ls)$
IN
$(sigma, 0 \models RawPhi) \equiv (sigma \models Phi)$
PROOF
BY *RawPhiImpliesPhi*, *PhiImpliesRawPhi*

Machine-unclosed representations.

This theorem tells us how to convert $\overset{+}{\rhd}$ to the stepwise form. The only difference with *PhiEquivRawPhi* is that A, G are not defined by machine-closed representations (meaning a conjunction of a machine-closed pair of properties).

A, G may be defined by machine-unclosed representations. So this theorem tells us that in general we have to first compute a machine-closed representation, before converting from $\overset{+}{\rhd}$ to the stepwise form, which we do in order to decide realizability via fixpoint computations.

In other words, this theorem differs from *PhiEquivRawPhi* in that defined symbols have been replaced by declarations of symbols together with axioms about their properties. So those symbols were defined by machine-closed expressions, whereas here they are only declared, and could be defined by machine-unclosed expressions.

In implementation we need to compute the closure of properties, so the closure needs to be expressible directly, without using temporal quantification. For open-system specifications where only finitely many relevant states, this rewriting is always possible.

THEOREM $\mathit{MachineUnclosedWhilePlus} \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

NEW sigma, META NEW

$\mathit{IsABehavior}(\mathit{sigma})$,

CONSTANT $\mathit{IeP}(\_, \_)$,

CONSTANT $\mathit{IsP}(\_, \_)$,

CONSTANT $\mathit{NeP}(\_, \_, \_, \_)$,

CONSTANT $\mathit{NsP}(\_, \_, \_, \_)$,

TEMPORAL Le, TEMPORAL Ls,

TEMPORAL A, TEMPORAL G,

$\land \forall u, v : \mathit{IeP}(u, v) \in$ BOOLEAN

$\land \forall u, v : \mathit{IsP}(u, v) \in$ BOOLEAN

$\land \forall a, b, c, d : \mathit{NeP}(a, b, c, d) \in$ BOOLEAN

$\land \forall a, b, c, d : \mathit{NsP}(a, b, c, d) \in$ BOOLEAN ,

LET

$v \triangleq \langle x, y \rangle$

$\mathit{Is} \triangleq \mathit{IsP}(x, y)$

$\mathit{Ie} \triangleq \mathit{IeP}(x, y)$

$\mathit{Ne} \triangleq \mathit{NeP}(x, y, x', y')$

$\mathit{Ns} \triangleq \mathit{NsP}(x, y, x', y')$

IN

$\land A \equiv (\mathit{Ie} \land \Box[\mathit{Ne}]_v \land \mathit{Le})$

$\land \mathit{IsMachineClosed}(\mathit{Ie} \land \Box[\mathit{Ne}]_v, \mathit{Le})$

$\land G \equiv (\mathit{Is} \land \Box[\mathit{Ns}]_v \land \mathit{Ls})$

$\land \mathit{IsMachineClosed}(\mathit{Is} \land \Box[\mathit{Ns}]_v, \mathit{Ls})$

PROVE

LET

$v \triangleq \langle x, y \rangle$

$\mathit{Is} \triangleq \mathit{IsP}(x, y)$

$\mathit{Ie} \triangleq \mathit{IeP}(x, y)$

$\mathit{Ne} \triangleq \mathit{NeP}(x, y, x', y')$

$\mathit{Ns} \triangleq \mathit{NsP}(x, y, x', y')$

$A \triangleq \mathit{Ie} \land \Box[\mathit{Ne}]_v \land \mathit{Le}$

$G \triangleq \mathit{Is} \land \Box[\mathit{Ns}]_v \land \mathit{Ls}$

$\mathit{EnvNext} \triangleq [\mathit{Ne}]_v$

$\mathit{SysNext} \triangleq [\mathit{Ns}]_v$

$\mathit{Phi} \triangleq A \stackrel{+}{\triangleright} G$

$\mathit{RawPhi} \triangleq \mathit{RawWhilePlus}($

$\mathit{IeP}, \mathit{Ie}, \mathit{Is}, \mathit{EnvNext}, \mathit{SysNext}, \mathit{Le}, \mathit{Ls})$

IN

$(\mathit{sigma}, 0 \models \mathit{RawPhi}) \equiv (\mathit{sigma} \models \mathit{Phi})$

PROOF SKETCH

$\langle 1 \rangle$ DEFINE

$v \triangleq \langle x, y \rangle$

$Is \triangleq IsP(x, y)$

$Ie \triangleq IeP(x, y)$

$Ne \triangleq NeP(x, y, x', y')$

$Ns \triangleq NsP(x, y, x', y')$

$A \triangleq Ie \land \Box[Ne]_v \land Le$

$G \triangleq Is \land \Box[Ns]_v \land Ls$

$P \triangleq Ie \land \Box[Ne]_v \land Le$

$Q \triangleq IS \land \Box[Ns]_v \land Ls$

$\langle 1 \rangle 1.\ A \overset{+}{\rhd} G \ \equiv\ P \overset{+}{\rhd} Q$

$\langle 2 \rangle 1.\ A \equiv P$

BY DEF A, P

$\langle 2 \rangle 2.\ G \equiv Q$

BY DEF G, Q

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle$ QED

BY $\langle 1 \rangle 1$, *PhiEquivRawPhi*

Alternative proof structure

The proof can be structured differently by using the identity:

$$A \overset{+}{\rhd} G \equiv \land\ C(A) \overset{+}{\rhd} C(G)$$
$$\land\ A \Rightarrow G$$

The second conjunct is present in the definitions of both of the operators $\overset{+}{\rhd}$ and *RawWhilePlus*. Only the first conjunct needs a lengthier proof, which reduces to

$$PrefixPlusOne(Cl(A), Cl(G))$$
$$\equiv \lor \neg \exists\, p, q\colon IeP(p, q)$$
$$\lor \land Is$$
$$\land IeP(x, y) \Rightarrow StepwiseImpl([Ne]\_v, [Ns]\_v)$$

where the actions and state predicates are those of the machine-closed canonical forms, as in the proof above.

Jonsson and *Tsay* structure their proof in this way. The module *WhilePlusHalfTheorems* follows this approach for the operator *WhilePlusHalf*.

[4, Lemma B.1 on p.70] does not hold for the case that H_E is unsatisfiable. Below is the analysis of that case. That case is covered by the theorems above.

The below proposition shows that:

$$\neg \models\ (\ (\Box \text{FALSE})\ \overset{+}{\rhd}\ (\Box \text{FALSE})\ )\ \equiv \Box(Earlier(\text{FALSE}) \Rightarrow \text{FALSE})$$

PROPOSITION

$\land\ \models \text{TRUE} \equiv ((\Box\text{FALSE}) \stackrel{+}{\rhd} (\Box\text{FALSE}))$
$\land\ raw \models\ \text{FALSE} \equiv \Box(\mathit{Earlier}(\text{FALSE}) \Rightarrow \text{FALSE})$
"raw" stands for "raw TLA+ with past"

PROOF

$\langle 1 \rangle 1.\ \models \text{TRUE} \equiv ((\Box\text{FALSE}) \stackrel{+}{\rhd} (\Box\text{FALSE}))$
    $\langle 2 \rangle 1.\ \models \text{FALSE} \equiv \Box\text{FALSE}$
        OBVIOUS
    $\langle 2 \rangle 2.\ (\text{FALSE} \stackrel{+}{\rhd} \text{FALSE}) \equiv ((\Box\text{FALSE}) \stackrel{+}{\rhd} (\Box\text{FALSE}))$
        BY $\langle 2 \rangle 1$
    $\langle 2 \rangle 3.\ \text{TRUE} \equiv (\text{FALSE} \stackrel{+}{\rhd} \text{FALSE})$
        BY *PhiEquivRawPhi*
    $\langle 2 \rangle$ QED
        BY $\langle 2 \rangle 2$, $\langle 2 \rangle 3$

$\langle 1 \rangle 2.\ raw \models\ \text{FALSE} \equiv \Box(\mathit{Earlier}(\text{FALSE}) \Rightarrow \text{FALSE})$
    $\langle 2 \rangle 1.$ ASSUME NEW $sigma$, $\mathit{IsABehavior}(sigma)$
        PROVE $(sigma, 0 \models \Box(\mathit{Earlier}(\text{FALSE}) \Rightarrow \text{FALSE}))$
            $\equiv\ \ \forall\, n \in \mathit{Nat} :\ sigma, n \models \mathit{Earlier}(\text{FALSE}) \Rightarrow \text{FALSE}$
        BY DEF $\Box$
    $\langle 2 \rangle 2.$ ASSUME NEW $sigma$, $\mathit{IsABehavior}(sigma)$
        PROVE $(sigma, 0 \models \Box(\mathit{Earlier}(\text{FALSE}) \Rightarrow \text{FALSE}))$
            $\equiv\ \ \forall\, n \in \mathit{Nat} :\ sigma, n \models \neg\mathit{Earlier}(\text{FALSE})$
        BY $\langle 2 \rangle 1$
    $\langle 2 \rangle 3.$ SUFFICES
            ASSUME NEW $sigma$, $\mathit{IsABehavior}(sigma)$
            PROVE $\exists\, n \in \mathit{Nat} :\ sigma, n \models \mathit{Earlier}(\text{FALSE})$
        BY $\langle 2 \rangle 2$
    $\langle 2 \rangle 4.\ sigma, 0 \models \mathit{Earlier}(\text{FALSE})$
        BY DEF *Earlier*
    $\langle 2 \rangle$ QED   goal from $\langle 2 \rangle 3$
        BY $\langle 2 \rangle 4$

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

MODULE *WhilePlusHalfTheorems*

Properties of the operator *WhilePlusHalf*, a variant of $\overset{+}{-\triangleright}$.

Below is a proof of the stepwise form of *WhilePlusHalf* in raw TLA+ with past temporal operators. This module includes definitions that are relevant to the family of stepwise implication operators. These definitions include syntactic and semantic definitions of these operators.

Author: Ioannis *Filippidis*

```
EXTENDS
    TLASemantics, TemporalLogic, TemporalQuantification,
    NaturalsInduction, TLAPS


PROPOSITION ShorterPrefixSat ≜
    ASSUME
        NEW n ∈ Nat,
        NEW sigma, IsABehavior(sigma),
        TEMPORAL G
    PROVE
        PrefixSat(sigma, n + 1, G) ⇒ PrefixSat(sigma, n, G)
    PROOF
    ⟨1⟩1. SUFFICES ASSUME PrefixSat(sigma, n + 1, G)
                   PROVE  PrefixSat(sigma, n, G)
        OBVIOUS
    ⟨1⟩2. ∃ tau :
            ∧ IsABehavior(tau)
            ∧ ∀ i ∈ 0 .. ((n + 1) − 1) :  tau[i] = sigma[i]
            ∧ tau ⊨ G
        BY ⟨1⟩1 DEF PrefixSat
    ⟨1⟩3. ∃ tau :
            ∧ IsABehavior(tau)
            ∧ ∀ i ∈ 0 .. n :  tau[i] = sigma[i]
            ∧ tau ⊨ G
        BY ⟨1⟩2
    ⟨1⟩4. ASSUME
            NEW tau,
            ∀ i ∈ 0 .. n :  tau[i] = sigma[i]
          PROVE
            ∀ i ∈ 0 .. (n − 1) :  tau[i] = sigma[i]
        ⟨2⟩4. n ∈ Nat
            OBVIOUS   BY ShorterPrefixSat
        ⟨2⟩1. SUFFICES ASSUME NEW i ∈ 0 .. (n − 1)
                       PROVE  tau[i]   = sigma[i]
            OBVIOUS
```

$\langle 2\rangle 2.$ ASSUME $n = 0$
PROVE FALSE
$\langle 3\rangle 1.\ (n - 1) = -1$
BY $\langle 2\rangle 2$
$\langle 3\rangle 2.\ 0 \,..\, (n - 1) = \{\}$
BY $\langle 3\rangle 1$
$\langle 3\rangle 3.\ i \in \{\}$
BY $\langle 3\rangle 2$, $\langle 2\rangle 1$
$\langle 3\rangle$ QED
BY $\langle 3\rangle 3$
$\langle 2\rangle 3.$CASE $n > 0$
$\langle 3\rangle 1.\ (n - 1) \in 0 \,..\, n$
BY $\langle 2\rangle 4$, $\langle 2\rangle 3$
$\langle 3\rangle 2.\ 0 \,..\, (n - 1) \subseteq 0 \,..\, n$
BY $\langle 3\rangle 1$
$\langle 3\rangle 3.\ i \in 0 \,..\, n$
BY $\langle 2\rangle 1$, $\langle 3\rangle 2$
$\langle 3\rangle$ QED
BY $\langle 3\rangle 3$, $\langle 1\rangle 4$
$\langle 2\rangle$ QED
BY $\langle 2\rangle 2$, $\langle 2\rangle 3$, $\langle 2\rangle 4$
$\langle 1\rangle 5.\ \exists\, tau :$
$\land\, IsABehavior(tau)$
$\land\, \forall\, i \in 0 \,..\, (n - 1) :\ tau[i] = sigma[i]$
$\land\, tau \models G$
BY $\langle 1\rangle 3$, $\langle 1\rangle 4$
$\langle 1\rangle$ QED
BY $\langle 1\rangle 5$ DEF $PrefixSat$

PROPOSITION
ASSUME
NEW $n \in Nat$,
NEW $sigma$, $IsABehavior(sigma)$,
TEMPORAL A, TEMPORAL G
PROVE
$PrefixPlusOne(sigma, A, G) \equiv$
$\forall\, n \in Nat :$
$PrefixSat(sigma, n, A) \Rightarrow \land\, PrefixSat(sigma, n, G)$
$\qquad \land\, PrefixSat(sigma, n + 1, G)$
PROOF
$\langle 1\rangle 1.\ PrefixPlusOne(sigma, A, G) \equiv$
$\forall\, n \in Nat :$
$PrefixSat(sigma, n, A)$
$\Rightarrow PrefixSat(sigma, n + 1, G)$

BY DEF *PrefixPlusOne*

$\langle 1 \rangle 2.$ $PrefixSat(sigma, n+1, G) \Rightarrow PrefixSat(sigma, n, G)$

BY *ShorterPrefixSat*

$\langle 1 \rangle$ QED

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

Semantic definition of "while" operators.

The semantic and syntactic definitions of $\stackrel{+}{\triangleright}$ and *WPH* are equivalent, despite the semantic ones omitting stutter-equivalence. This ows to the fact that temporal quantification serves for only replacing the behavior's tail, not for step-refinement.

The *While* operator from the module *TLASemantics*.
Copied here for comparison.

$$sigma \models While(A, G) \triangleq$$
$$\land \forall n \in Nat: PrefixSat(sigma, n, A) \Rightarrow PrefixSat(sigma, n, G)$$
$$\land\ sigma \models A \Rightarrow G$$

The *WhilePlus* operator. Compied here for comparison.

$$PrefixPlusOne(sigma, A, G) \triangleq$$
$$\forall n \in Nat: PrefixSat(sigma, n, A) \Rightarrow PrefixSat(sigma, n+1, G)$$

$$sigma \models A \stackrel{+}{\triangleright} G \triangleq$$
$$\land\ PrefixPlusOne(sigma, A, G)$$
$$\land\ sigma \models A \Rightarrow G$$

Attention: the signature of the operator is in the object language (TLA+),
but the definition is in the metatheory. Thus, x and y need delicate handling.

$sigma \models WhilePlusHalf(A, G, x, y) \equiv$ notice this is $\equiv$, not $\triangleq$

LET

$$SamePrefixSatXY(tau, n, H) \triangleq$$
$$\land\ IsABehavior(tau)$$
$$\land\ tau \models H$$
$$\land \forall i \in 0 \,..\, (n-1):$$
$$\land\ tau[i].x = sigma[i].x$$
$$\land\ tau[i].y = sigma[i].y$$
$$PrefixSatVar(n, H) \triangleq$$
$$\exists\, tau: \ SamePrefixSatXY(tau, n, H)$$
$$PrefixSatVarPlusHalf(n, H) \triangleq$$
$$\exists\, tau: \ \land\ SamePrefixSatXY(tau, n, H)$$
$$\land\ tau[n].y = sigma[n].y$$

IN

$$\land\ sigma \models F \Rightarrow G$$
$$\land \forall n \in Nat:$$
$$PrefixSatVar(n, F) \Rightarrow PrefixSatVarPlusHalf(n, G)$$

The semantic definitions of WPH and $\stackrel{+}{\triangleright}$ both are meaningful in raw \tlaplus. The syntactic definitions are equivalent to the semantic definitions within TLA+. The syntactic definitions are equivalent to the semantic ones also within raw TLA+, even after replacing temporal quantification by its stutter-sensitive version.

The reason is the same as mentioned above: the definitions utilize temporal quantification for only hiding the behavior's tail; not for step-refinement.

Each operator could be defined in roughly three ways: within TLA+ (*e.g.*, *WhilePlus*), which is also sensible within raw TLA+, within raw TLA+ (*e.g.*, *RawWhilePlus*), and in the metatheory, with the below definition as a demonstration:

$MetaWhilePlusHalf(sigma, A, G, x, y) \triangleq$
  LET
    $SamePrefixSatXY$ ...

Contents of module *OpenSystems* (refactored).

These are syntactic definitions for the family of "while" operators.

Variable b starts in BOOLEAN and changes at most once to FALSE.

$$MayUnstep(b) \triangleq \land b \in \text{BOOLEAN} \quad \land \Box[b' = \text{FALSE}]_b$$

Variable b starts in BOOLEAN and becomes FALSE with at most one change.

$$Unstep(b) \triangleq \land MayUnstep(b) \quad \land \Diamond(b = \text{FALSE})$$

Variable b starts TRUE and changes once to FALSE.

$$MustUnstep(b) \triangleq \land b = \text{TRUE} \quad \land Unstep(b)$$

Redefined from module *TemporalLogic* to change arity.

$$SamePrefix(b, u, v, x, y) \triangleq \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$$

$$PlusHalf(b, v, y) \triangleq \land v = y \quad \land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y \rangle}$$

Redefined from module *TemporalLogic* to change arity.

$$Front(P(\_, \_), x, y, b) \triangleq \exists\!\!\exists\, u, v : \land P(u, v) \quad \land SamePrefix(b, u, v, x, y)$$

$$FrontPlusHalf(P(\_, \_), x, y, b) \triangleq \exists\!\!\exists\, u, v : \land P(u, v) \quad \land SamePrefix(b, u, v, x, y) \quad \land PlusHalf(b, v, y)$$

$FrontPlus(P(\_, \_), x, y, b) \triangleq \exists\!\!\exists\, u, v :$
  LET
    $vars \triangleq \langle b, x, y, u, v \rangle$
    $Init \triangleq \langle u, v \rangle = \langle x, y \rangle$
    $Next \triangleq b \Rightarrow (\langle u', v' \rangle = \langle x', y' \rangle)$

$\quad Plus \triangleq \Box[Next]_{vars}$

IN

$\quad \land P(u, v)$

$\quad \land Init \land Plus$

An additional definition (not in the module *OpenSystems*).
This is a syntactic definition of the *While* operator.

$While(A(\_, \_), G(\_, \_), x, y) \triangleq$
$\quad \forall b :$
$\quad\quad (MayUnstep(b) \land Front(A, x, y, b)) \Rightarrow Front(G, x, y, b)$

The TLA+ operator $\overset{+}{\rhd}$ expressed within the logic [1, *p*.337].
[1] *Leslie Lamport*, "Specifying systems", Addison-Wesley, 2002

$WhilePlus(A(\_, \_), G(\_, \_), x, y) \triangleq$
$\quad \forall b :$
$\quad\quad (MayUnstep(b) \land Front(A, x, y, b)) \Rightarrow FrontPlus(G, x, y, b)$

A variant of the *WhilePlus* operator.

$WhilePlusHalf(A(\_, \_), G(\_, \_), x, y) \triangleq$
$\quad \forall b :$
$\quad\quad (MayUnstep(b) \land Front(A, x, y, b)) \Rightarrow FrontPlusHalf(G, x, y, b)$

An operator that forms an open system from the closed system that the
temporal property $P(x, y)$ describes.

$Unzip(P(\_, \_), x, y) \triangleq$

LET

$\quad Q(u, v) \triangleq P(v, u)$ swap back to x, y

$\quad A(u, v) \triangleq WhilePlusHalf(Q, Q, v, u)$ swap to y, x

IN

$\quad WhilePlusHalf(A, P, x, y)$

PROPOSITION $SwapInSamePrefix \triangleq$

ASSUME

VARIABLE u, VARIABLE v, VARIABLE x, VARIABLE y

PROVE

$SamePrefix(b, u, v, y, x)$
$\equiv SamePrefix(b, v, u, x, y)$

PROOF

$\langle 1 \rangle 1.$ ASSUME VARIABLE u, VARIABLE v

PROVE

$SamePrefix(b, u, v, y, x)$
$\equiv \Box(b \Rightarrow (\langle u, v \rangle = \langle y, x \rangle))$

BY DEF $SamePrefix$

$\langle 1 \rangle 2.$ ASSUME VARIABLE u, VARIABLE v

PROVE

$(\langle u, v \rangle = \langle y, x \rangle)$
$\equiv (\langle v, u \rangle \triangleq \langle x, y \rangle)$
OBVIOUS

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

How quantification of initial conditions is handled distinguishes between a disjoint-state specification ($\exists \forall$) and a shared-state specification ($\exists \exists$ or $\forall \forall$).

Below we use a definition of closure that takes three arguments.

$$Cl(P(\_), x, y) \triangleq \forall b : \; MustUnstep(b) \Rightarrow Front(P, x, y, b)$$

$$WPH(A, G, x, y) \triangleq WhilePlusHalf(A, G, x, y)$$

analogous to *ClosureEquiSAT*

PROPOSITION *ClosureEquiSATHalf* $\triangleq$
ASSUME
VARIABLE y,
TEMPORAL $P(\_, \_)$
PROVE
$(\exists u, v : \; (v = y) \land P(u, v))$
$\equiv \; \exists u, v : \; (v = y) \land Cl(P, u, v)$
PROOF

$\langle 1 \rangle$ DEFINE
$ClP(u, v) \triangleq Cl(P, u, v)$

$\langle 1 \rangle 1.$ $\lor \neg \exists u, v : \; \land v = y$
$\land P(u, v)$
$\lor \exists u, v : \; \land v = y$
$\land Cl(P, u, v)$
BY *ClosureImplied*

$\langle 1 \rangle 2.$ $\lor \neg \exists u, v : \; \land v = y$
$\land Cl(P, u, v)$
$\lor \exists u, v : \; \land v = y$
$\land P(u, v)$

$\langle 2 \rangle$ DEFINE $R(v, y) \triangleq v = y$

$\langle 2 \rangle 1.$ SUFFICES
$\lor \neg \exists u, v : \; R(v, y) \land Cl(P, u, v))$
$\lor \exists u, v : \; R(v, y) \land P(u, v)$
BY DEF R

$\langle 2 \rangle$ QED
BY *ClosureSample* goal from $\langle 2 \rangle 1$

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

```
PROPOSITION ReplaceWithClosureWithinFront ≜
    ASSUME
        VARIABLE x, VARIABLE y, VARIABLE b,
        TEMPORAL P(_, _)
    PROVE
        LET
            Fr(P(_, _), b) ≜ Front(P, x, y, b)
            ClP(u, v) ≜ Cl(P, u, v)
        IN
            ∨ ¬MustUnstep(b)
            ∨ Fr(P, b) ≡ Fr(ClP, b)
    PROOF
    ⟨1⟩ DEFINE
        Fr(P(_, _), b) ≜ Front(P, x, y, b)
        ClP(u, v) ≜ Cl(P, u, v)
    ⟨1⟩1. Fr(P, b)
            ≡ ∃ u, v :  ∧ P(u, v)
                        ∧ SamePrefix(b, u, v, x, y)
        BY DEF Fr, Front
    ⟨1⟩2. Fr(ClP, b)
            ≡ ∃ u, v :  ∧ ClP(u, v)
                        ∧ SamePrefix(b, u, v, x, y)
        BY DEF Fr, Front
    ⟨1⟩3. Fr(P, b) ⇒ Fr(ClP, b)
        ⟨2⟩1. ASSUME VARIABLE u, VARIABLE v
              PROVE P(u, v) ⇒ ClP(u, v)
            BY ClosureImplied
        ⟨2⟩2. Fr(P, b)
                ≡ ∃ u, v :  ∧ P(u, v) ∧ ClP(u, v)
                            ∧ SamePrefix(b, u, v, x, y)
            BY ⟨1⟩1, ⟨2⟩1
        ⟨2⟩3. Fr(P, b)
                ⇒ ∃ u, v :  ∧ ClP(u, v)
                            ∧ SamePrefix(b, u, v, x, y)
            BY ⟨2⟩2
        ⟨2⟩ QED
            BY ⟨2⟩3, ⟨1⟩2
    ⟨1⟩4. ∨ ¬MustUnstep(b)
          ∨ Fr(ClP, b) ⇒ Fr(P, b)
        ⟨2⟩1. Fr(ClP, b)
                ≡ ∃ u, v :  ∧ ∀ r :  ∨ ¬MustUnstep(r)
                                     ∨ Front(P, u, v, r)
                            ∧ SamePrefix(b, u, v, x, y)
            BY ⟨1⟩2 DEF ClP, Cl
        ⟨2⟩2. Fr(ClP, b)
```

$\Rightarrow \exists\, u, v : \quad \land \lor \neg MustUnstep(b)$
$\qquad\qquad\qquad\quad \lor Front(P, u, v, b)$
$\qquad\qquad\quad \land SamePrefix(b, u, v, x, y)$

BY ⟨2⟩1, *InstantiateAA*

⟨2⟩3. $\lor \neg MustUnstep(b)$
$\lor \neg Fr(ClP, b)$
$\lor \exists\, u, v : \quad \land Front(P, u, v, b)$
$\qquad\qquad \land SamePrefix(b, u, v, x, y)$

BY ⟨2⟩2

⟨2⟩4. $\lor \neg MustUnstep(b)$
$\lor \neg Fr(ClP, b)$
$\lor \exists\, u, v : \quad \land \exists\, p, q : \quad \land P(p, q)$
$\qquad\qquad\qquad\qquad \land SamePrefix(b, p, q, u, v)$
$\qquad\qquad \land SamePrefix(b, u, v, x, y)$

BY ⟨2⟩3 DEF *SamePrefix*

⟨2⟩5. $\lor \neg MustUnstep(b)$
$\lor \neg Fr(ClP, b)$
$\lor \exists\, u, v, p, q :$
$\quad \land P(p, q)$
$\quad \land SamePrefix(b, p, q, u, v)$
$\quad \land SamePrefix(b, u, v, x, y)$

BY ⟨2⟩4

⟨2⟩6. ASSUME
VARIABLE u, VARIABLE v, VARIABLE p, VARIABLE q
PROVE
$\lor \neg \land SamePrefix(b, p, q, u, v)$
$\qquad \land SamePrefix(b, u, v, x, y)$
$\lor SamePrefix(b, p, q, x, y)$

⟨3⟩1. SUFFICES
$\lor \neg \land \Box(b \Rightarrow (\langle p, q\rangle = \langle u, v\rangle))$
$\qquad \land \Box(b \Rightarrow (\langle u, v\rangle = \langle x, y\rangle))$
$\lor \Box(b \Rightarrow (\langle p, q\rangle = \langle x, y\rangle))$

BY DEF *SamePrefix*

⟨3⟩2. $(\land \Box(b \Rightarrow (\langle p, q\rangle = \langle u, v\rangle))$
$\quad \land \Box(b \Rightarrow (\langle u, v\rangle = \langle x, y\rangle))$
$) \equiv ($
$\Box(b \Rightarrow \ \land \langle p, q\rangle = \langle u, v\rangle$
$\qquad\qquad \land \langle u, v\rangle = \langle x, y\rangle)$
$)$

BY *PTL*

⟨3⟩3. $\lor \neg\Box(b \Rightarrow \ \land \langle p, q\rangle = \langle u, v\rangle$
$\qquad\qquad\qquad \land \langle u, v\rangle = \langle x, y\rangle)$
$\lor \Box(b \Rightarrow (\langle p, q\rangle = \langle x, y\rangle))$

BY *PTL*

⟨3⟩ QED

BY ⟨3⟩2, ⟨3⟩3 goal from ⟨3⟩1

⟨2⟩7. $\lor \neg \mathit{MustUnstep}(b)$
$\lor \neg \mathit{Fr}(\mathit{ClP}, b)$
$\lor \exists\, u, v, p, q :$
$\land P(p, q)$
$\land \mathit{SamePrefix}(b, p, q, x, y)$

BY ⟨2⟩5, ⟨2⟩6

⟨2⟩8. $\lor \neg \mathit{MustUnstep}(b)$
$\lor \neg \mathit{Fr}(\mathit{ClP}, b)$
$\lor \exists\, p, q :$
$\land P(p, q)$
$\land \mathit{SamePrefix}(b, p, q, x, y)$

BY ⟨2⟩7

⟨2⟩9. $\lor \neg \mathit{MustUnstep}(b)$
$\lor \neg \mathit{Fr}(\mathit{ClP}, b)$
$\lor \exists\, u, v :$
$\land P(u, v)$
$\land \mathit{SamePrefix}(b, u, v, x, y)$

BY ⟨2⟩8 rename the bound variables p, q to u, v

⟨2⟩10. $\lor \neg \mathit{MustUnstep}(b)$
$\lor \neg \mathit{Fr}(\mathit{ClP}, b)$
$\lor \mathit{Fr}(P, b)$

BY ⟨2⟩9, ⟨1⟩1

⟨2⟩ QED

BY ⟨2⟩10

⟨1⟩ QED

BY ⟨1⟩3, ⟨1⟩4

PROPOSITION *ReplaceWithClosureWithinFrontPlusHalf* $\triangleq$

ASSUME

VARIABLE x, VARIABLE y, VARIABLE b,
TEMPORAL $P(\_, \_)$

PROVE

LET

$\mathit{FPH}(P(\_, \_), b) \triangleq \mathit{FrontPlusHalf}(P, x, y, b)$
$\mathit{ClP}(u, v) \triangleq \mathit{Cl}(P, u, v)$

IN

$\lor \neg \mathit{MustUnstep}(b)$
$\lor \mathit{FPH}(P, b) \equiv \mathit{FPH}(\mathit{ClP}, b)$

PROOF

⟨1⟩ DEFINE

$\mathit{FPH}(P(\_, \_), b) \triangleq \mathit{FrontPlusHalf}(P, x, y, b)$
$\mathit{ClP}(u, v) \triangleq \mathit{Cl}(P, u, v)$

⟨1⟩1. $\mathit{FPH}(P, b)$

$\equiv \exists\, u, v : \;\; \land P(u, v)$
$\qquad\qquad \land\, SamePrefix(b, u, v, x, y)$
$\qquad\qquad \land\, PlusHalf(b, v, y)$
BY DEF FPH, $FrontPlusHalf$

$\langle 1 \rangle 2.\ FPH(ClP, b)$
$\equiv \exists\, u, v : \;\; \land ClP(u, v)$
$\qquad\qquad \land\, SamePrefix(b, u, v, x, y)$
$\qquad\qquad \land\, PlusHalf(b, v, y)$
BY DEF FPH, $FrontPlusHalf$

$\langle 1 \rangle 3.\ FPH(P, b) \Rightarrow FPH(ClP, b)$
$\langle 2 \rangle 1.$ ASSUME VARIABLE u, VARIABLE v
PROVE $P(u, v) \Rightarrow ClP(u, v)$
BY $ClosureImplied$
$\langle 2 \rangle$ QED
BY $\langle 1 \rangle 1$, $\langle 2 \rangle 1$, $\langle 1 \rangle 2$

$\langle 1 \rangle 4.\ \lor \neg MustUnstep(b)$
$\qquad \lor \neg FPH(ClP, b)$
$\qquad \lor FPH(P, b)$
$\langle 2 \rangle 1.\ FPH(ClP, b)$
$\equiv \exists\, u, v : \;\; \land \forall\, r : \;\; \lor \neg MustUnstep(r)$
$\qquad\qquad\qquad\qquad \lor\, Front(P, u, v, r)$
$\qquad\qquad \land\, SamePrefix(b, u, v, x, y)$
$\qquad\qquad \land\, PlusHalf(b, v, y)$
BY $\langle 1 \rangle 2$ DEF ClP, Cl
$\langle 2 \rangle 2.\ \lor \neg MustUnstep(b)$
$\qquad \lor \exists\, z : \;\; \land MustUnstep(z)$
$\qquad\qquad \land \Box(b \Rightarrow z)$
$\qquad\qquad \land \Box[z' = b]_{\langle b, z \rangle}$
$\langle 3 \rangle 1.$ SUFFICES
ASSUME
NEW $sigma$, $IsABehavior(sigma)$,
$sigma \models MustUnstep(b)$
PROVE $sigma \models \exists\, z : \;\; \land MustUnstep(z)$
$\qquad\qquad \land \Box(b \Rightarrow z)$
$\qquad\qquad \land \Box[z' = b]_{\langle b, z \rangle}$
OBVIOUS
$\langle 3 \rangle 2.\ tau \triangleq [n \in Nat \mapsto$
$[sigma[n]$ EXCEPT $![$"z"$] =$
IF $n = 0$ THEN TRUE
ELSE $sigma[n-1][$"b"$]]$
a one-step delay
$\langle 3 \rangle 3.\ IsABehavior(tau)$
BY $\langle 3 \rangle 1$ DEF tau
$\langle 3 \rangle 4.\ tau \models MustUnstep(b)$
$\langle 4 \rangle 1.\ sigma \models MustUnstep(b)$

BY $\langle 3 \rangle 1$

$\langle 4 \rangle 2.$ $\forall\, n \in Nat :\ tau[n][\text{"b"}] = sigma[n][\text{"b"}]$

BY DEF tau

$\langle 4 \rangle$ QED

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

$\langle 3 \rangle 5.$ $EqualUpToVar(tau, sigma, \text{"z"})$

BY DEF $EqualUpToVar$, tau

$\langle 3 \rangle 6.$ $Sim(sigma, sigma)$

BY DEF Sim

$\langle 3 \rangle 7.$ CHOOSE $k \in Nat$:

$\land \forall\, n \in 0 \,..\, k :\ tau[n][\text{"b"}] = \text{TRUE}$

$\land \forall\, n \in Nat :\ (n > k) \ \Rightarrow\ (tau[n][\text{"b"}] = \text{FALSE})$

BY $\langle 3 \rangle 4$ DEF $MustUnstep$, $Unstep$, $MayUnstep$

$\langle 3 \rangle 8.$ LET $m \triangleq k + 1$

IN $\land \forall\, n \in 0 \,..\, m :\ tau[n][\text{"z"}] = \text{TRUE}$

$\land \forall\, n \in Nat :\ (n > m) \ \Rightarrow\ (tau[n][\text{"z"}] = \text{FALSE})$

BY $\langle 3 \rangle 7$ DEF tau

$\langle 3 \rangle 9.$ $tau \models MustUnstep(z)$

BY $\langle 3 \rangle 8$ DEF $MustUnstep$, $Unstep$, $MayUnstep$

$\langle 3 \rangle 10.$ $tau \models\ \land \Box(b \Rightarrow z)$

$\land \Box[z' = b]_{\langle b,\, z \rangle}$

$\langle 4 \rangle 1.$ $\land \forall\, n \in 0 \,..\, k :\ tau[n][\text{"b"}] = \text{TRUE}$

$\land \forall\, n \in 0 \,..\, k :\ tau[n][\text{"z"}] = \text{TRUE}$

$\land \forall\, n \in Nat :\ (n > k) \ \Rightarrow\ (tau[n][\text{"b"}] = \text{FALSE})$

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

$\langle 4 \rangle 2.$ $\forall\, n \in Nat :\ (tau[n][\text{"b"}] \Rightarrow tau[n][\text{"z"}])$

Writing $(tau[n][\text{"b"}] = \text{TRUE}) \ \Rightarrow\ (tau[n][\text{"z"}] = \text{TRUE})$
would not lead to the desired conclusion below, unless we invoked the type invariant.

BY $\langle 4 \rangle 1$

$\langle 4 \rangle 3.$ $\forall\, n \in Nat :\ tau[n] \models (b \Rightarrow z)$

BY $\langle 4 \rangle 2$

$\langle 4 \rangle 4.$ $\forall\, n \in Nat :\ \langle tau[n],\, tau[n+1] \rangle \models z' = b$

BY DEF tau

$\langle 4 \rangle 5.$ $\forall\, n \in Nat$:

$\langle tau[n],\, tau[n+1] \rangle \models [z' = b]_{\langle b,\, z \rangle}$

BY $\langle 4 \rangle 4$

$\langle 4 \rangle$ QED

BY $\langle 4 \rangle 3$, $\langle 4 \rangle 5$

$\langle 3 \rangle 11.$ $tau \models\ \land MustUnstep(z)$

$\land \Box(b \Rightarrow z)$

$\land \Box[z' = b]_{\langle b,\, z \rangle}$

BY $\langle 3 \rangle 9$, $\langle 3 \rangle 10$

$\langle 3 \rangle 12.$ $\land IsABehavior(tau)$

$\land\ \land IsABehavior(sigma)$ $RefinesUpToVar$

$\quad\quad\quad\quad \land\ Sim(sigma,\ sigma)$
$\quad\quad\quad\quad \land\ EqualUpToVar(sigma,\ tau,\ \text{“z”})$
$\quad\quad\quad \land\ tau \models MustUnstep(z) \land \Box(b \Rightarrow z)$
$\quad\quad \langle 4\rangle 1.\ IsABehavior(tau)$
$\quad\quad\quad \text{BY}\ \langle 3\rangle 3$
$\quad\quad \langle 4\rangle 2.\ IsABehavior(sigma)$
$\quad\quad\quad \text{BY}\ \langle 3\rangle 1$
$\quad\quad \langle 4\rangle 3.\ EqualUpToVar(tau,\ sigma,\ \text{“z”})$
$\quad\quad\quad \text{BY}\ \langle 3\rangle 5$
$\quad\quad \langle 4\rangle 4.\ Sim(sigma,\ sigma)$
$\quad\quad\quad \text{BY}\ \langle 3\rangle 6$
$\quad\quad \langle 4\rangle\ \text{QED}$
$\quad\quad\quad \text{BY}\ \langle 4\rangle 1,\ \langle 4\rangle 2,\ \langle 4\rangle 3,\ \langle 4\rangle 4,\ \langle 3\rangle 11$
$\quad \langle 3\rangle\ \text{QED}$
$\quad\quad \text{BY}\ \langle 3\rangle 12\ \ \text{DEF}\ RefinesUpToVar,\ \exists$
$\langle 2\rangle 3.\ \lor \neg MustUnstep(b)$
$\quad\quad \lor \neg FPH(ClP,\ b)$
$\quad\quad \lor \exists\, u,\ v :$
$\quad\quad\quad \land \exists\, z :\ \ \land MustUnstep(z)$
$\quad\quad\quad\quad\quad\quad \land \Box(b \Rightarrow z)$
$\quad\quad\quad\quad\quad\quad \land \Box[z' = b]_{\langle b,\ z\rangle}$
$\quad\quad\quad \land \forall\, r :\ \ \lor \neg MustUnstep(r)$
$\quad\quad\quad\quad\quad\quad \lor Front(P,\ u,\ v,\ r)$
$\quad\quad\quad \land SamePrefix(b,\ u,\ v,\ x,\ y)$
$\quad\quad\quad \land PlusHalf(b,\ v,\ y)$
$\quad \text{BY}\ \langle 2\rangle 1,\ \langle 2\rangle 2$
$\langle 2\rangle 4.\ \lor \neg MustUnstep(b)$
$\quad\quad \lor \neg FPH(ClP,\ b)$
$\quad\quad \lor \exists\, u,\ v,\ z :$
$\quad\quad\quad \land MustUnstep(z)$
$\quad\quad\quad \land \Box(b \Rightarrow z)$
$\quad\quad\quad \land \Box[z' = b]_{\langle b,\ z\rangle}$
$\quad\quad\quad \land \forall\, r :\ \ \lor \neg MustUnstep(r)$
$\quad\quad\quad\quad\quad\quad \lor Front(P,\ u,\ v,\ r)$
$\quad\quad\quad \land SamePrefix(b,\ u,\ v,\ x,\ y)$
$\quad\quad\quad \land PlusHalf(b,\ v,\ y)$
$\quad \text{BY}\ \langle 2\rangle 3$
$\langle 2\rangle 5.\ \lor \neg MusUnstep(b)$
$\quad\quad \lor \neg FPH(ClP,\ b)$
$\quad\quad \lor \exists\, u,\ v,\ z :$
$\quad\quad\quad \land MustUnstep(z)$
$\quad\quad\quad \land \Box(b \Rightarrow z)$
$\quad\quad\quad \land \Box[z' = b]_{\langle b,\ z\rangle}$
$\quad\quad\quad \land\ \lor \neg MustUnstep(z)$
$\quad\quad\quad\quad \lor Front(P,\ u,\ v,\ z)$

$\wedge\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$
$\wedge\ \mathit{PlusHalf}(b,\ v,\ y)$
BY $\langle 2\rangle 4$

$\langle 2\rangle 6.$ $\vee\ \neg \mathit{MusUnstep}(b)$
$\vee\ \neg \mathit{FPH}(\mathit{ClP},\ b)$
$\vee\ \exists\, u,\ v,\ z :$
$\wedge\ \mathit{MustUnstep}(z)$
$\wedge\ \Box(b \Rightarrow z)$
$\wedge\ \Box[z' = b]_{\langle b,\, z\rangle}$
$\wedge\ \mathit{Front}(P,\ u,\ v,\ z)$
$\wedge\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$
$\wedge\ \mathit{PlusHalf}(b,\ v,\ y)$
BY $\langle 2\rangle 5$, $\mathit{InstantiateAA}$

$\langle 2\rangle 7.$ $\vee\ \neg \mathit{MusUnstep}(b)$
$\vee\ \neg \mathit{FPH}(\mathit{ClP},\ b)$
$\vee\ \exists\, u,\ v,\ z :$
$\wedge\ \mathit{MustUnstep}(z)$
$\wedge\ \Box(b \Rightarrow z)$
$\wedge\ \Box[z' = b]_{\langle b,\, z\rangle}$
$\wedge\ \exists\, p,\ q :$
$\wedge\ P(p,\ q)$
$\wedge\ \mathit{SamePrefix}(z,\ p,\ q,\ u,\ v)$
$\wedge\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$
$\wedge\ \mathit{PlusHalf}(b,\ v,\ y)$
BY $\langle 2\rangle 6$ DEF Front

$\langle 2\rangle 8.$ $\vee\ \neg \mathit{MusUnstep}(b)$
$\vee\ \neg \mathit{FPH}(\mathit{ClP},\ b)$
$\vee\ \exists\, u,\ v,\ z,\ p,\ q :$
$\wedge\ \mathit{MustUnstep}(z)$
$\wedge\ \Box(b \Rightarrow z)$
$\wedge\ \Box[z' = b]_{\langle b,\, z\rangle}$
$\wedge\ P(p,\ q)$
$\wedge\ \mathit{SamePrefix}(z,\ p,\ q,\ u,\ v)$
$\wedge\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$
$\wedge\ \mathit{PlusHalf}(b,\ v,\ y)$
BY $\langle 2\rangle 7$

$\langle 2\rangle 9.$ ASSUME
VARIABLE z, VARIABLE p, VARIABLE q,
VARIABLE u, VARIABLE v
PROVE
$\vee\ \neg \wedge\ \mathit{SamePrefix}(z,\ p,\ q,\ u,\ v)$
$\wedge\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$
$\wedge\ \Box(b \Rightarrow z)$
$\vee\ \mathit{SamePrefix}(b,\ p,\ q,\ x,\ y)$
BY DEF $\mathit{SamePrefix}$

$\langle 2\rangle 10.$ ASSUME

VARIABLE z, VARIABLE p, VARIABLE q,
VARIABLE u, VARIABLE v

PROVE

$$\begin{array}{l} \lor \neg \land MustUnstep(z) \\ \quad\ \ \land SamePrefix(z, p, q, u, v) \\ \quad\ \ \land PlusHalf(b, v, y) \\ \lor q = y \end{array}$$

$\langle 3\rangle 1.$ $PlusHalf(b, v, y) \Rightarrow (v = y)$
BY DEF $PlusHalf$

$\langle 3\rangle 2.$ $\lor \neg SamePrefix(z, p, q, u, v)$
$\lor z \Rightarrow (q = v)$
BY DEF $SamePrefix$

$\langle 3\rangle 3.$ $MustUnstep(z) \Rightarrow (z = \text{TRUE})$
BY DEF $MustUnstep$

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

$\langle 2\rangle 11.$ ASSUME

VARIABLE z, VARIABLE p, VARIABLE q,
VARIABLE u, VARIABLE v

PROVE

$$\begin{array}{l} \lor \neg \land \Box(b \Rightarrow z) \\ \quad\ \ \land \Box[z' = b]_{\langle b, z\rangle} \\ \quad\ \ \land SamePrefix(z, p, q, u, v) \\ \quad\ \ \land PlusHalf(b, v, y) \\ \lor \Box[b \Rightarrow (y' = q')]_{\langle b, y, q\rangle} \end{array}$$

$\langle 3\rangle 1.$ $\lor \neg \land \Box(b \Rightarrow z)$
$\quad\ \ \land \Box[z' = b]_{\langle b, z\rangle}$
$\lor \Box[b \Rightarrow z']_{\langle b, v, y, q\rangle}$

$\langle 4\rangle 1.$ SUFFICES

ASSUME

$(b \Rightarrow z) \land [z' = b]_{\langle b, z\rangle}$

PROVE

$[b \Rightarrow z']_{\langle b, v, y, q\rangle}$

BY PTL

$\langle 4\rangle 2.$ CASE UNCHANGED $\langle b, z\rangle$

$\langle 5\rangle 1.$ $(b \Rightarrow z') \equiv (b \Rightarrow z)$
BY $\langle 4\rangle 2$

$\langle 5\rangle 2.$ $b \Rightarrow z'$
BY $\langle 4\rangle 1$, $\langle 5\rangle 1$

$\langle 5\rangle$ QED
BY $\langle 5\rangle 2$ goal from $\langle 4\rangle 1$

$\langle 4\rangle 3.$ CASE $\neg$UNCHANGED $\langle b, z\rangle$

$\langle 5\rangle 1.$ $z' = b$

BY ⟨4⟩1, ⟨4⟩3

⟨5⟩2. $b \Rightarrow z'$

BY ⟨5⟩1

⟨5⟩ QED

BY ⟨5⟩2

⟨4⟩ QED

BY ⟨4⟩2, ⟨4⟩3

⟨3⟩2. $\lor \neg SamePrefix(z, p, q, u, v)$
$\lor \Box[z' \Rightarrow (v' = q')]_{\langle b, v, y, q, z\rangle}$

⟨4⟩1. $\lor \neg SamePrefix(z, p, q, u, v)$
$\lor \Box(z \Rightarrow (v = q))$

BY DEF *SamePrefix*

⟨4⟩ QED

BY ⟨4⟩1, *PTL*

⟨3⟩3. $\lor \neg \land \Box[b \Rightarrow z']_{\langle b, v, y, q\rangle}$
$\land \Box[z' \Rightarrow (v' = q')]_{\langle b, v, y, q, z\rangle}$
$\lor \Box[b \Rightarrow (v' = q')]_{\langle b, v, y, q\rangle}$

⟨4⟩1. SUFFICES

ASSUME

$\land \neg$UNCHANGED $\langle b, v, y, q\rangle$
$\land [b \Rightarrow z']_{\langle b, v, y, q\rangle}$
$\land [z' \Rightarrow (v' = q')]_{\langle b, v, y, q, z\rangle}$

PROVE

$b \Rightarrow (v' = q')$

BY *PTL*

⟨4⟩2. $\land b \Rightarrow z'$
$\land z' \Rightarrow (v' = q')$

BY ⟨4⟩1

⟨4⟩ QED

BY ⟨4⟩2

⟨3⟩4. $\lor \neg \land \Box(b \Rightarrow z)$
$\land \Box[z' = b]_{\langle b, z\rangle}$
$\land SamePrefix(z, p, q, u, v)$
$\lor \land \Box[z' \Rightarrow (v' = q')]_{\langle b, v, y, q, z\rangle}$
$\land \Box[b \Rightarrow z']_{\langle b, v, y, q\rangle}$

BY ⟨3⟩1, ⟨3⟩2

⟨3⟩5. $\lor \neg \land \Box(b \Rightarrow z)$
$\land \Box[z' = b]_{\langle b, z\rangle}$
$\land SamePrefix(z, p, q, u, v)$
$\lor \Box[b \Rightarrow (v' = q')]_{\langle b, v, y, q\rangle}$

BY ⟨3⟩3, ⟨3⟩4

$\langle 3\rangle 6.$ $\lor \neg \land \Box(b \Rightarrow z)$
$\quad\quad \land \Box[z' = b]_{\langle b,\, z\rangle}$
$\quad\quad \land\, \mathit{SamePrefix}(z,\, p,\, q,\, u,\, v)$
$\quad\quad \land\, \mathit{PlusHalf}(b,\, v,\, y)$
$\quad \lor \land \Box[b \Rightarrow (v' = q')]_{\langle b,\, v,\, y,\, q\rangle}$
$\quad\quad \land \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y\rangle}$
BY $\langle 3\rangle 5$ DEF *PlusHalf*

$\langle 3\rangle 7.$ $\lor \neg \land \Box[b \Rightarrow (v' = q')]_{\langle b,\, v,\, y,\, q\rangle}$
$\quad\quad \land \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y\rangle}$
$\quad\quad \land\, \mathit{SamePrefix}(z,\, p,\, q,\, u,\, v)$
$\quad\quad \land \Box(b \Rightarrow z)$
$\quad \lor \Box[b \Rightarrow (y' = q')]_{\langle b,\, y,\, q\rangle}$

$\langle 4\rangle 1.$ SUFFICES
ASSUME
$\land [b \Rightarrow (v' = q')]_{\langle b,\, v,\, y,\, q\rangle}$
$\land [b \Rightarrow (v' = y')]_{\langle b,\, v,\, y\rangle}$
$\land z \Rightarrow (\langle p,\, q\rangle = \langle u,\, v\rangle)$
$\land b \Rightarrow z$
PROVE
$[b \Rightarrow (y' = q')]_{\langle b,\, y,\, q\rangle}$
BY *PTL*

$\langle 4\rangle 2.$ SUFFICES
ASSUME $b \land \neg$UNCHANGED $\langle b,\, y,\, q\rangle$
PROVE $y' = q'$
OBVIOUS goal from $\langle 4\rangle 1$

$\langle 4\rangle 3.$ CASE UNCHANGED q
$\langle 5\rangle 1.$ $\neg$UNCHANGED $\langle b,\, y\rangle$
BY $\langle 4\rangle 2$, $\langle 4\rangle 3$
$\langle 5\rangle 2.$ $\land b \Rightarrow (v' = q')$
$\quad \land b \Rightarrow (v' = y')$
BY $\langle 5\rangle 1$, $\langle 4\rangle 1$
$\langle 5\rangle$ QED
$\langle 6\rangle 1.$ b
BY $\langle 4\rangle 2$
$\langle 6\rangle$ QED
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 6\rangle 1$ goal from $\langle 4\rangle 2$

$\langle 4\rangle 4.$ CASE $\neg$UNCHANGED q
$\langle 5\rangle 1.$ $v' \neq v$
$\langle 6\rangle 1.$ $b \Rightarrow (v' = q')$
BY $\langle 4\rangle 1$, $\langle 4\rangle 4$
$\langle 6\rangle 2.$ $v' = q'$
BY $\langle 6\rangle 1$, $\langle 4\rangle 2$
$\langle 6\rangle 3.$ $v' \neq q$
BY $\langle 4\rangle 4$, $\langle 6\rangle 2$

$\langle 6\rangle 4.\ q = v$

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

$\langle 6\rangle$ QED

$\langle 6\rangle 3$, $\langle 6\rangle 4$

$\langle 5\rangle 2.\ \land\ b \Rightarrow (v' = q')$

$\land\ b \Rightarrow (v' = y')$

BY $\langle 5\rangle 1$, $\langle 4\rangle 1$

$\langle 5\rangle$ QED

$\langle 6\rangle 1.\ b$

BY $\langle 4\rangle 2$

$\langle 6\rangle$ QED

BY $\langle 5\rangle 1$, $\langle 5\rangle 2$, $\langle 6\rangle 1$ goal from $\langle 4\rangle 2$

$\langle 4\rangle$ QED

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

$\langle 3\rangle$ QED

BY $\langle 3\rangle 6$, $\langle 3\rangle 7$

$\langle 2\rangle 12.\ \lor\ \neg MusUnstep(b)$

$\lor\ \neg FPH(ClP,\ b)$

$\lor\ \exists\ u,\ v,\ z,\ p,\ q :$

$\land\ P(p,\ q)$

$\land\ SamePrefix(b,\ p,\ q,\ x,\ y)$

$\land\ q = y$

$\land\ \Box[b \Rightarrow (y' = q')]_{\langle b,\ y,\ q\rangle}$

BY $\langle 2\rangle 8$, $\langle 2\rangle 9$, $\langle 2\rangle 10$, $\langle 2\rangle 11$

$\langle 2\rangle 13.\ \lor\ \neg MusUnstep(b)$

$\lor\ \neg FPH(ClP,\ b)$

$\lor\ \exists\ p,\ q :$

$\land\ P(p,\ q)$

$\land\ SamePrefix(b,\ p,\ q,\ x,\ y)$

$\land\ q = y$

$\land\ \Box[b \Rightarrow (y' = q')]_{\langle b,\ y,\ q\rangle}$

BY $\langle 2\rangle 12$

$\langle 2\rangle 14.\ \lor\ \neg MusUnstep(b)$

$\lor\ \neg FPH(ClP,\ b)$

$\lor\ \exists\ p,\ q :$

$\land\ P(p,\ q)$

$\land\ SamePrefix(b,\ p,\ q,\ x,\ y)$

$\land\ PlusHalf(b,\ q,\ y)$

BY $\langle 2\rangle 13$ DEF $PlusHalf$

$\langle 2\rangle$ QED

BY $\langle 2\rangle 14$, $\langle 1\rangle 1$

$\langle 1\rangle$ QED

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

This decomposition is of the same form as that of $\stackrel{+}{\rhd}$.

This fact can be used to prove a safety-liveness decomposition analogous to the theorem *WhilePlusMachineClosedRepr*.

THEOREM $WhilePlusHalfAsConj \triangleq$
ASSUME
VARIABLE x, VARIABLE y,
TEMPORAL $A(\_, \_)$, TEMPORAL $G(\_, \_)$
PROVE
LET
$ClA(u, v) \triangleq Cl(A, u, v)$
$ClG(u, v) \triangleq Cl(G, u, v)$
IN
$$WPH(A, G, x, y) \equiv \land WPH(ClA, ClG, x, y) \land A(x, y) \Rightarrow G(x, y)$$
PROOF
$\langle 1 \rangle$ DEFINE
$ClA(u, v) \triangleq Cl(A, u, v)$
$ClG(u, v) \triangleq Cl(G, u, v)$
$Fr(P(\_, \_), b) \triangleq Front(P, x, y, b)$
$FPH(P(\_, \_), b) \triangleq FrontPlusHalf(P, x, y, b)$
$\langle 1 \rangle$ USE DEF WPH, $WhilePlusHalf$, Fr, FPH, $Front$, $FrontPlusHalf$, $SamePrefix$, $PlusHalf$, $MustUnstep$, $MustUnstep$

$\langle 1 \rangle 3.$ ASSUME
TEMPORAL $Q(\_, \_)$, TEMPORAL $R(\_, \_)$
PROVE
$$WPH(Q, R, x, y) \equiv \land \forall b : (Fr(Q, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(R, b) \land \forall b : (Fr(Q, b) \land MustUnstep(b)) \Rightarrow FPH(R, b) \land \forall b : (Fr(Q, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(R, b)$$
$\langle 2 \rangle 1.$ $MayUnstep(b) \equiv \lor \Box(b = \text{TRUE}) \lor MustUnstep(b) \lor \Box(b = \text{FALSE})$
BY DEF $MayUnstep$
$\langle 2 \rangle 2.$ $WPH(Q, R, x, y) \equiv$
$$\forall b : \land (Fr(Q, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(R, b) \land (Fr(Q, b) \land MustUnstep(b)) \Rightarrow FPH(R, b) \land (Fr(Q, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(R, b)$$
BY $\langle 2 \rangle 1$ DEF WPH
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 2$ DEF $\forall$

$\langle 1 \rangle 4.$ ASSUME VARIABLE b
PROVE The first conjunct of $\langle 1 \rangle 3$ is $A \Rightarrow G$

```
    ( ∨ ¬ ∧ Fr(A, b)
          ∧ □(b = TRUE)
      ∨ FPH(G, b))
     ≡ (A(x, y) ⇒ G(x, y))
⟨2⟩1. ASSUME
        TEMPORAL P(_, _)
      PROVE
        ∨ ¬□(b = TRUE)
        ∨ P(x, y) ≡ ∃ u, v :
              ∧ P(u, v)
              ∧ SamePrefix(b, u, v, x, y))
    ⟨3⟩1. ASSUME VARIABLE u, VARIABLE v
          PROVE  ∨ ¬□(b = TRUE)
                 ∨ SamePrefix(b, u, v, x, y)
                    ≡ □(⟨u, v⟩ = ⟨x, y⟩)
        BY DEF SamePrefix
    ⟨3⟩ QED
        BY ⟨3⟩1 DEF ∃
⟨2⟩2. ∨ ¬□(b = TRUE)
      ∨ Fr(A, b) ≡ A(x, y)
    BY ⟨2⟩1 DEF Fr
⟨2⟩3. ∨ ¬□(b = TRUE)
      ∨ FPH(G, b) ≡ G(x, y)
    ⟨3⟩1. FPH(G, b) ≡ ∃ u, v :
                ∧ G(u, v)
                ∧ SamePrefix(b, u, v, x, y)
                ∧ PlusHalf(b, v, y)
        BY DEF FPH
    ⟨3⟩2. ASSUME VARIABLE u, VARIABLE v
          PROVE
            ∨ ¬□(b = TRUE)
            ∨ SamePrefix(b, u, v, x, y) ⇒ PlusHalf(b, v, y)
        BY DEF SamePrefix, PlusHalf
    ⟨3⟩3. ∨ ¬□(b = TRUE)
          ∨ FPH(G, b) ≡ ∃ u, v :
                  ∧ G(u, v)
                  ∧ SamePrefix(b, u, v, x, y)
        BY ⟨3⟩1, ⟨3⟩2
    ⟨3⟩ QED
        BY ⟨2⟩1, ⟨3⟩3
⟨2⟩ QED
    ⟨3⟩1. (( ∨ ¬ ∧ Fr(A, b)
                 ∧ □(b = TRUE)
             ∨ FPH(G, b))
             ≡ (A(x, y) ⇒ G(x, y)))
```

$\equiv$

$\lor \neg\Box(b = \text{TRUE})$

$\lor (Fr(A, b) \Rightarrow FPH(G, b))$

$\quad \equiv (A(x, y) \Rightarrow G(x, y)))$

OBVIOUS

⟨3⟩2. $\lor \neg\Box(b = \text{TRUE})$

$\lor (Fr(A, b) \Rightarrow FPH(G, b))$

$\quad \equiv (A(x, y) \Rightarrow G(x, y)))$

BY ⟨2⟩2, ⟨2⟩3

⟨3⟩ QED

BY ⟨3⟩1, ⟨3⟩2

⟨1⟩5. ASSUME VARIABLE b

PROVE $(Fr(A, b) \land MustUnstep(b)) \Rightarrow FPH(G, b)$

$\equiv \quad (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b)$

⟨2⟩1. $\lor \neg MustUnstep(b)$

$\lor Fr(A, b) \equiv Fr(ClA, b)$

BY *ReplaceWithClosureWithinFront*

⟨2⟩2. $\lor \neg MustUnstep(b)$

$\lor FPH(G, b) \equiv FPH(ClG, b)$

BY *ReplaceWithClosureWithinFrontPlusHalf*

⟨2⟩ QED

BY ⟨2⟩1, ⟨2⟩2

⟨1⟩6. ASSUME VARIABLE b

PROVE $(Fr(A, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(G, b)$

$\equiv \quad (Fr(ClA, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG, b)$

⟨2⟩1. $\lor \neg\Box(b = \text{FALSE})$

$\lor Fr(A, b) \equiv Fr(ClA, b)$

⟨3⟩1. ASSUME TEMPORAL $P(\_, \_)$

PROVE

$\lor \neg\Box(b = \text{FALSE})$

$\lor Fr(P, b) \equiv \exists\, u, v : \; P(u, v)$

BY DEF *Fr*, *SamePrefix*

⟨3⟩2. $(\exists\, u, v : \; A(u, v)) \;\equiv\; \exists\, u, v : \; ClA(u, v)$

BY *ClosureEquiSAT*

⟨3⟩3. $\lor \neg\Box(b = \text{FALSE})$

$\lor Fr(A, b) \equiv \exists\, u, v : \; A(u, v)$

BY ⟨3⟩1

⟨3⟩4. $\lor \neg\Box(b = \text{FALSE})$

$\lor Fr(ClA, b) \equiv \exists\, u, v : \; ClA(u, v)$

BY ⟨3⟩1

⟨3⟩ QED

BY ⟨3⟩2, ⟨3⟩3, ⟨3⟩4

⟨2⟩2. $\lor \neg\Box(b = \text{FALSE})$

$\lor\ FPH(G, b) \equiv FPH(ClG, b)$

$\langle 3\rangle 1.$ ASSUME TEMPORAL $P(\_, \_)$

PROVE

$\lor\ \neg\Box(b = \text{FALSE})$

$\lor\ FPH(G, b) \equiv \exists\, u, v : \ (v = y) \land P(u, v)$

BY DEF FPH, $SamePrefix$, $PlusHalf$

$\langle 3\rangle 2.$ $(\exists\, u, v : \ (v = y) \land G(u, v))$

$\equiv \exists\, u, v : \ (v = y) \land ClG(u, v)$

BY $ClosureEquiSATHalf$

$\langle 3\rangle 3.$ $\lor\ \neg\Box(b = \text{FALSE})$

$\lor\ FPH(G, b) \equiv \exists\, u, v : \ (v = y) \land G(u, v)$

BY $\langle 3\rangle 1$

$\langle 3\rangle 4.$ $\lor\ \neg\Box(b = \text{FALSE})$

$\lor\ FPH(ClG, b) \equiv \exists\, u, v : \ (v = y) \land ClG(u, v)$

BY $\langle 3\rangle 1$

$\langle 3\rangle$ QED

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

$\langle 2\rangle$ QED

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

$\langle 1\rangle 7.$ $WPH(A, G, x, y) \equiv$

$\land\ A(x, y) \Rightarrow G(x, y)$

$\land\ \forall\, b : \ (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b)$

$\land\ \forall\, b : \ (Fr(ClA, b) \land \Box(b = \text{FALSE})) \ \Rightarrow FPH(ClG, b)$

$\langle 2\rangle 1.$ $(A(x, y) \Rightarrow G(x, y))$

$\equiv (\forall\, b : \ A(x, y) \Rightarrow G(x, y))$

OBVIOUS

$\langle 2\rangle$ QED

BY $\langle 1\rangle 3$, $\langle 2\rangle 1$, $\langle 1\rangle 5$, $\langle 1\rangle 6$

$\langle 1\rangle 8.$ $\lor\ \neg(A(x, y) \Rightarrow G(x, y))$

$\lor\ \forall\, b : \ (Fr(ClA, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(ClG, b)$

$\langle 2\rangle 1.$ $(A(x, y) \Rightarrow G(x, y)) \ \Rightarrow \ (Cl(A, x, y) \Rightarrow Cl(G, x, y))$

BY $ClosureIsMonotonic$

$\langle 2\rangle 2.$ $(Cl(A, x, y) \Rightarrow Cl(G, x, y))$

$\equiv \forall\, b : \ \lor\ \neg\Box(b = \text{TRUE})$

$\lor\ Fr(ClA, b) \Rightarrow FPH(ClG, b)$

proof similar to that of $\langle 1\rangle 4$

$\langle 2\rangle$ QED

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

$\langle 1\rangle 9.$ $WPH(A, G, x, y) \equiv$

$\land\ A(x, y) \Rightarrow G(x, y)$

$\land\ \forall\, b : \ (Fr(ClA, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(ClG, b)$

$\land\ \forall\, b : \ (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b)$

$\land\ \forall\, b : \ (Fr(ClA, b) \land \Box(b = \text{FALSE})) \ \Rightarrow FPH(ClG, b)$

BY ⟨1⟩7, ⟨1⟩8

⟨1⟩ QED

⟨2⟩1. $WPH(ClA, ClG, x, y) \equiv$

$\land \forall b : (Fr(ClA, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(ClG, b)$

$\land \forall b : (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b)$

$\land \forall b : (Fr(ClA, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG, b)$

BY ⟨1⟩3 DEF ClA, ClG

⟨2⟩ QED

BY ⟨1⟩9, ⟨2⟩1

THEOREM $WhilePlusHalfSafetyLivenessDecomposition \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

TEMPORAL A, TEMPORAL G

PROVE

LET

$W \triangleq WhilePlusHalf(A, G, x, y)$

$C \triangleq Cl(A, x, y) \overset{+}{\rhd} Cl(G, x, y)$

IN

$\land SafetyPart(W) \equiv C$

$\land LivenessPart(W) \equiv (C \Rightarrow W)$

PROOF OMITTED similar to *WhilePlusSafetyLivenessDecomp*.

Expressing *WhilePlusHalf* in raw TLA+ with past.

An operator used to describe *WhilePlusHalf* in raw TLA+. The arguments *InitA* and *InitB* are not environment and component initial conditions; they are just appropriately defined predicates.

$RawWhilePlusHalf($

$InitA, InitB,$

$EnvNext, Next, SysNext,$

$Le, Ls) \triangleq$

$InitA \Rightarrow \land InitB$

$\land \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next)$

$\land SysNext)$

$\land (ILe \land \Box EnvNext) \Rightarrow Ls$

The conjunctive form of the operator has the advantage of making reasoning about closure easier, for the particular form of *InitA* that arises by translating *WPH* to raw TLA+.

Expanded form after the intended substitutions (see below):

$RawWhilePlusHalfFull($

$IeP(\_, \_), JeP(\_, \_), IsP(\_, \_),$
$EnvNext, Next, SysNext, Le, Ls) \triangleq$

$$
\begin{array}{l}
\lor \neg \exists\, p, q: \; IeP(p, q) \Rightarrow JeP(p, q) \\
\lor \land \exists\, p: \; IsP(p, y) \\
\quad \land \lor \neg \lor \neg IeP(x, y) \\
\qquad\quad \lor JeP(x, y) \\
\qquad \lor \land IsP(x, y) \\
\qquad\quad \land IeP(x, y) \lor \Box(Next \land SysNext) \\
\qquad\quad \land \lor \neg IeP(x, y) \\
\qquad\qquad \lor \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext) \\
\qquad\quad \land \lor \neg \lor \neg IeP(x, y) \\
\qquad\qquad\quad \lor JeP(x, y) \land Le \land \Box EnvNext \\
\qquad\qquad \lor Ls
\end{array}
$$

This is the "shallow" case.

PROPOSITION
ASSUME
CONSTANT $JeP(\_, \_)$, CONSTANT $IsP(\_, \_)$,
ACTION $EnvNext$, ACTION $Next$, ACTION $SysNext$,
TEMPORAL Le, TEMPORAL Ls
PROVE

$$
\begin{array}{l}
RawWhilePlusHalfFull(\\
\quad \text{TRUE}, JeP, IsP, EnvNext, Next, SysNext, Le, Ls) \\
\equiv \\
\lor \neg \exists\, p, q: \; JeP(p, q) \\
\lor \land \exists\, p: \; IsP(p, y) \\
\quad \land \lor \neg \; JeP(x, y) \\
\qquad \lor \land IsP(x, y) \\
\qquad\quad \land \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \land SysNext) \\
\qquad\quad \land (Le \land \Box EnvNext) \Rightarrow Ls
\end{array}
$$

PROOF OBVIOUS

If $\models IeP(x, y)$, then the above becomes

$$
\begin{array}{l}
\lor \neg \exists\, p, q: JeP(p, q) \\
\lor \land \exists\, p: IsP(p, y) \\
\quad \land \lor \neg JeP(x, y) \\
\qquad \lor \land IsP(x, y) \\
\qquad\quad \land \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext) \\
\qquad\quad \land \; (LeP(x, y) \land \Box EnvNext) \Rightarrow LsP(x, y)
\end{array}
$$

$$
\begin{array}{l}
SIH(EnvNext, Next, SysNext) \triangleq \\
\quad \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\; \land SysNext)
\end{array}
$$

PROPOSITION

ASSUME ACTION $EnvNext$, ACTION $Next$, ACTION $SysNext$

PROVE $WeakStepwiseImpl(EnvNext, SysNext)$

$\equiv SIH(EnvNext, SysNext, \text{TRUE})$

PROOF OBVIOUS

PROPOSITION

ASSUME ACTION $EnvNext$, ACTION $Next$, ACTION $SysNext$

PROVE $StepwiseImpl(EnvNext, SysNext)$

$\equiv SIH(EnvNext, \text{TRUE}, SysNext)$

$\equiv SIH(EnvNext, SysNext, SysNext)$

PROOF OBVIOUS

Stepwise form of *WhilePlusHalf*.

THEOREM $WhilePlusHalfStepwiseForm \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

NEW $sigma$, $IsABehavior(sigma)$,

CONSTANT $IeP(\_, \_)$,

CONSTANT $JeP(\_, \_)$,

CONSTANT $IsP(\_, \_)$,

CONSTANT $NeP(\_, \_, \_, \_)$,

CONSTANT $NsP(\_, \_, \_, \_)$,

TEMPORAL LeP, TEMPORAL LsP,

The constants are predicates.

TEMPORAL level expressions can only be formulas, so *LeP* and *LsP* are certainly Boolean-valued.

$\land \forall u, v : IeP(u, v) \in$ BOOLEAN

$\land \forall u, v : JeP(u, v) \in$ BOOLEAN

$\land \forall u, v : IsP(u, v) \in$ BOOLEAN

$\land \forall a, b, c, d : NeP(a, b, c, d) \in$ BOOLEAN

$\land \forall a, b, c, d : NsP(a, b, c, d) \in$ BOOLEAN ,

LET

$xy \triangleq \langle x, y \rangle$

$Is \triangleq IsP(x, y)$

$Ie \triangleq IeP(x, y)$

$Je \triangleq JeP(x, y)$

$Ne \triangleq NeP(x, y, x', y')$

$Ns \triangleq NsP(x, y, x', y')$

$Le \triangleq LeP(x, y)$

$Ls \triangleq LsP(x, y)$

$A(u, v) \triangleq$

LET

$I \triangleq IeP(u, v)$
$J \triangleq JeP(u, v)$
$N \triangleq NeP(u, v, u', v')$
$vrs = \langle u, v \rangle$
$L \triangleq LeP(u, v)$

IN

$I \Rightarrow (J \land \Box[N]_{vrs} \land L)$

$Q(u, v) \triangleq$
$\lor \neg IeP(u, v)$
$\lor \land JeP(u, v)$
$\land \Box[NeP(u, v, u', v')]_{\langle u, v \rangle}$

$R(u, v) \triangleq \land IsP(u, v)$
$\land \Box[NsP(u, v, u', v')]_{\langle u, v \rangle}$

IN

for our intended usage, the term $IeP \Rightarrow LeP$ never arises;
either IeP is TRUE, or LeP is TRUE. So this assumption reduces to either:
$Cl(I \Rightarrow (J \land \Box[N]\_vrs)) \equiv (I \Rightarrow (J \land \Box[N]\_vrs))$
or $Cl(J \land \Box[N]\_vrs \land L) \equiv (J \land \Box[N]\_vrs)$

The first case is easy to prove, because it is a safety property (alternatively, we can invoke the safety-liveness decomposition of *WhilePlusHalf*.

The second case is a typical machine-closure condition.

$\land \forall u, v : \; Cl(A, u, v) \equiv Q(u, v)$
$\land \forall u, v : \; IsMachineClosed(R, LsP, u, v)$

PROVE

LET

$A(u, v) \triangleq$

LET

$I \triangleq IeP(u, v)$
$J \triangleq JeP(u, v)$
$N \triangleq NeP(u, v, u', v')$
$vrs = \langle u, v \rangle$
$L \triangleq LeP(u, v)$

IN

$I \Rightarrow (J \land \Box[N]_{vrs} \land L)$

$G(u, v) \triangleq$

LET

$I \triangleq IsP(u, v)$
$N \triangleq NsP(u, v, u', v')$
$vrs \triangleq \langle u, v \rangle$
$L \triangleq LsP(u, v)$

IN

$I \land \Box[N]_{vrs} \land L$

$Phi \triangleq WhilePlusHalf(A, G, x, y)$

$xy \triangleq \langle x, y \rangle$
$Ie \triangleq IeP(x, y)$
$Is \triangleq IsP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$Le \triangleq LeP(x, y)$
$Ls \triangleq LsP(x, y)$

$EnvNext \triangleq [Ne]_{\langle x, y \rangle}$
$Next \triangleq [Ns]_{\langle x, y \rangle}$
$SysNext \triangleq [\exists r : NsP(x, y, r, y')]_y$

$RawPhi \triangleq RawWhilePlusHalfFull($
$IeP, JeP, IsP, EnvNext, SysNext, Le, Ls)$

IN

$(sigma, 0 \models RawPhi) \equiv (sigma \models Phi)$

PROOF

⟨1⟩ DEFINE

$Is \triangleq IsP(x, y)$
$Ie \triangleq IeP(x, y)$
$Je \triangleq JeP(x, y)$
$Ne \triangleq NeP(x, y, x', y')$
$Ns \triangleq NsP(x, y, x', y')$
$Le \triangleq LeP(x, y)$
$Ls \triangleq LsP(x, y)$

$A(u, v) \triangleq$

LET

$I \triangleq IeP(u, v)$
$J \triangleq JeP(u, v)$
$N \triangleq NeP(u, v, u', v')$
$vrs = \langle u, v \rangle$
$L \triangleq LeP(u, v)$

IN

$I \Rightarrow (J \land \Box[N]_{vrs} \land L)$

$G(u, v) \triangleq$

LET

$I \triangleq IsP(u, v)$
$N \triangleq NsP(u, v, u', v')$
$vrs \triangleq \langle u, v \rangle$
$L \triangleq LsP(u, v)$

IN

$I \land \Box[N]_{vrs} \land L$

$ClA(u, v) \triangleq Cl(A, u, v)$
$ClG(u, v) \triangleq Cl(G, u, v)$

$Fr(P(\_, \_), b) \triangleq Front(P, x, y, b)$
$FPH(P(\_, \_), b) \triangleq FrontPlusHalf(P, x, y, b)$

$Q(u, v) \triangleq IeP(u, v) \Rightarrow \land JeP(u, v)$
$\qquad\qquad\qquad\qquad\qquad\quad \land \Box[NeP(u, v, u', v')]_{\langle u, v \rangle}$
$R(u, v) \triangleq IsP(u, v) \land \Box[NsP(u, v, u', v')]_{\langle u, v \rangle}$

$EnvNext \triangleq [Ne]_{\langle x, y \rangle}$
$Next \triangleq [Ns]_{\langle x, y \rangle}$
$SysNext \triangleq [\exists r : NsP(x, y, r, y')]_y$

Not using bound variables via $\forall u, v$ here would be a mistake.
To understand why, consider the operator $Foo(u) \triangleq x = u$ and what the assertion $Foo(x)$ tells us (nothing).

⟨1⟩5. $\forall u, v : ClA(u, v) \equiv Q(u, v)$
  ⟨2⟩1. $\forall u, v : ClA(u, v) \equiv Cl(A, u, v)$
    BY DEF ClA
  ⟨2⟩2. $\forall u, v : Cl(A, u, v) \equiv$
      $\lor \neg IeP(u, v)$
      $\lor \land JeP(u, v)$
        $\land \Box[NeP(u, v, u', v')]_{\langle u, v \rangle}$
    BY DEF A
      and *WhilePlusHalfStepwiseForm*!assumption
  ⟨2⟩3. $\forall u, v : Cl(A, u, v) \equiv Q(u, v)$
    BY ⟨2⟩2 DEF Q
  ⟨2⟩ QED
    BY ⟨2⟩1, ⟨2⟩3

⟨1⟩6. $\forall u, v : ClG(u, v) \equiv R(u, v)$
  ⟨2⟩1. $\forall u, v : ClG(u, v) \equiv Cl(G, u, v)$
    BY DEF ClG
  ⟨2⟩2. ASSUME VARIABLE u, VARIABLE v
    PROVE $R(u, v) \equiv Cl(G, u, v)$
    ⟨3⟩1.LET $F(u, v) \triangleq R(u, v) \land LsP(u, v)$
      IN $R(u, v) \equiv Cl(F, u, v)$
      BY DEF R, *IsMachineClosed*
        and *WhilePlusHalfStepwiseForm*!assumption
    ⟨3⟩2. $G(u, v) \equiv (R(u, v) \land LsP(u, v))$
      BY DEF G, R
    ⟨3⟩ QED
      BY ⟨3⟩1, ⟨3⟩2
  ⟨2⟩ QED
    BY ⟨2⟩1, ⟨2⟩2

⟨1⟩7. ASSUME VARIABLE b
  PROVE $Fr(ClA, b) \equiv Fr(Q, b)$

$\langle 2\rangle 1.\ Fr(ClA,\ b) \equiv \exists\, u,\ v :$

$\quad \land\ ClA(u,\ v)$

$\quad \land\ SamePrefix(b,\ u,\ v,\ x,\ y)$

BY DEF Fr, $Front$, ClA

$\langle 2\rangle 2.\ Fr(ClA,\ b) \equiv \exists\, u,\ v :$

$\quad \land\ Q(u,\ v)$

$\quad \land\ SamePrefix(b,\ u,\ v,\ x,\ y)$

BY $\langle 2\rangle 1$, $\langle 1\rangle 5$

$\langle 2\rangle$ QED

BY $\langle 2\rangle 2$ DEF $Front$, Fr

$\langle 1\rangle 8.$ ASSUME VARIABLE b

PROVE $FPH(ClG,\ b) \equiv FPH(R,\ b)$

$\langle 2\rangle 1.\ FPH(ClG,\ b) \equiv \exists\, u,\ v :$

$\quad \land\ ClG(u,\ v)$

$\quad \land\ SamePrefix(b,\ u,\ v,\ x,\ y)$

$\quad \land\ PlusHalf(b,\ v,\ y)$

BY DEF FPH, $FrontPlusHalf$

$\langle 2\rangle 2.\ FPH(ClG,\ b) \equiv \exists\, u,\ v :$

$\quad \land\ R(u,\ v)$

$\quad \land\ SamePrefix(b,\ u,\ v,\ x,\ y)$

$\quad \land\ PlusHalf(b,\ v,\ y)$

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

$\langle 2\rangle$ QED

BY $\langle 2\rangle 2$ DEF $FrontPlusHalf$, FPH

$\langle 1\rangle 1.\ WPH(A,\ G,\ x,\ y) \equiv$

liveness part

$\land\ A(x,\ y) \Rightarrow G(x,\ y)$

initial condition

$\land\ \forall\, b :\ (Fr(ClA,\ b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG,\ b)$

stepwise implication

$\land\ \forall\, b :\ (Fr(ClA,\ b) \land MustUnstep(b)) \Rightarrow FPH(ClG,\ b)$

This expansion combines the theorem *WhilePlusHalfAsConj* with reversal of some of its final steps.

$\langle 2\rangle 1.\ WPH(A,\ G,\ x,\ y) \equiv$

$\quad \land\ WPH(ClA,\ ClG,\ x,\ y)$

$\quad \land\ A(x,\ y) \Rightarrow G(x,\ y)$

BY $WhilePlusHalfAsConj$

$\langle 2\rangle 2.\ WPH(A,\ G,\ x,\ y) \equiv$

$\quad \land\ \forall\, b :\ (Fr(ClA,\ b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(ClG,\ b)$

$\quad \land\ \forall\, b :\ (Fr(ClA,\ b) \land MustUnstep(b)) \Rightarrow FPH(ClG,\ b)$

$\quad \land\ \forall\, b :\ (Fr(ClA,\ b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG,\ b)$

$\quad \land\ A(x,\ y) \Rightarrow G(x,\ y)$

OMITTED

$\langle 2\rangle 3.\ \lor \neg(A(x, y) \Rightarrow G(x, y))$
$\lor \boldsymbol{\forall}\, b : (Fr(ClA, b) \land \Box(b = \text{TRUE})) \Rightarrow FPH(ClG, b)$

OMITTED

$\langle 2\rangle$ QED

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

The liveness part.

$\langle 1\rangle 2.\ (A(x, y) \Rightarrow G(x, y))$
$\equiv \lor \neg \lor \neg IeP(x, y)$
$\lor \land JeP(x, y)$
$\land \Box[NeP(x, y, x', y')]_{\langle x, y\rangle}$
$\land LeP(x, y)$
$\lor \land IsP(x, y)$
$\land \Box[NsP(x, y, x', y')]_{\langle x, y\rangle}$
$\land LsP(x, y)$

This assertion is expressed in TLA+. Any TLA+ formula is also a formula of raw TLA+ with past, so we can transfer this equivalence to the raw logic. The same observation applies to the assertion of step $\langle 1\rangle 6$ below.

BY DEF A, G

The initial condition.

$\langle 1\rangle 3.\ (\boldsymbol{\forall}\, b : (Fr(ClA, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG, b))$
$\equiv \lor \neg(\exists\, p, q : IeP(p, q) \Rightarrow JeP(p, q))$
$\lor \exists\, p : Is(p, y)$

$\langle 2\rangle 1.$ ASSUME VARIABLE b
PROVE $(\ (Fr(ClA, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(ClG, b)\ )$
$\equiv (\ (Fr(Q, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(R, b)\ )$

BY $\langle 1\rangle 6$, $\langle 1\rangle 7$

$\langle 2\rangle 2.$ ASSUME VARIABLE b
PROVE $(\ (Fr(Q, b) \land \Box(b = \text{FALSE})) \Rightarrow FPH(R, b)\ )$
$\equiv \lor \neg(Fr(Q, b) \land \Box(b = \text{FALSE}))$
$\lor FPH(R, b) \land \Box(b = \text{FALSE})$

OBVIOUS

$\langle 2\rangle 3.$ ASSUME VARIABLE b
PROVE $\lor \neg\Box(b = \text{FALSE})$
$\lor Fr(Q, b) \equiv \exists\, p, q : IeP(p, q) \Rightarrow JeP(p, q)$

$\langle 3\rangle 1.\ \lor \neg\Box(b = \text{FALSE})$
$\lor Fr(Q, b) \equiv \boldsymbol{\exists}\, u, v : \land Q(u, v)$
$\land SamePrefix(b, u, v, x, y)$

BY DEF Fr, $Front$

$\langle 3\rangle 2.\ \lor \neg\Box(b = \text{FALSE})$
$\lor Fr(Q, b) \equiv \boldsymbol{\exists}\, u, v : Q(u, v)$

$\langle 4\rangle 1.\ \lor \neg\Box(b = \text{FALSE})$

$\lor \mathit{SamePrefix}(b, u, v, x, y)$

BY DEF *SamePrefix*

⟨4⟩ QED

BY ⟨3⟩1, ⟨4⟩1

⟨3⟩3. $(\exists u, v : Q(u, v)) \equiv \exists p, q : \mathit{IeP}(p, q) \Rightarrow \mathit{JeP}(p, q)$

⟨4⟩1. $(\exists u, v : Q(u, v)) \equiv \exists u, v : \lor \neg \mathit{IeP}(u, v) \lor \land \mathit{JeP}(u, v) \land \Box[\mathit{NeP}(u, v, u', v')]_{\langle u, v \rangle}$

BY DEF Q

⟨4⟩2. $(\exists u, v : Q(u, v)) \equiv \lor \exists u, v : \neg \mathit{IeP}(u, v) \lor \exists u, v : \land \mathit{JeP}(u, v) \land \Box[\mathit{NeP}(u, v, u', v')]_{\langle u, v \rangle}$

BY ⟨4⟩1

⟨4⟩3. $(\exists u, v : Q(u, v)) \equiv \lor \exists p, q : \neg \mathit{IeP}(p, q) \lor \exists p, q : \mathit{JeP}(p, q)$

BY ⟨4⟩2 just stutter forever the initial state

⟨4⟩ QED

BY ⟨4⟩3

⟨3⟩ QED

BY ⟨3⟩2, ⟨3⟩3

⟨2⟩4. ASSUME VARIABLE b

PROVE $\lor \neg \Box(b = \text{FALSE}) \lor \mathit{FPH}(R, b) \equiv \exists p : \mathit{IsP}(p, y)$

⟨3⟩1. $\lor \neg \Box(b = \text{FALSE}) \lor \mathit{FPH}(R, b) \equiv \exists u, v : \land R(u, v) \land \mathit{SamePrefix}(b, u, v, x, y) \land \mathit{PlusHalf}(b, v, y)$

BY DEF *FPH*, *FrontPlusHalf*

⟨3⟩2. $\lor \neg \Box(b = \text{FALSE}) \lor \mathit{FPH}(R, b) \equiv \exists u, v : R(u, v) \land (v = y)$

⟨4⟩1. $\lor \neg \Box(b = \text{FALSE}) \lor \mathit{SamePrefix}(b, u, v, x, y)$

BY DEF *SamePrefix*

⟨4⟩2. $\lor \neg \Box(b = \text{FALSE}) \lor \mathit{PlusHalf}(b, v, y) \equiv (v = y)$

BY DEF *PlusHalf*

⟨4⟩ QED

BY ⟨3⟩1, ⟨4⟩1, ⟨4⟩2

⟨3⟩3. $(\exists u, v : R(u, v) \land (v = y)) \equiv \exists p : \mathit{IsP}(p, y)$

$\langle 4 \rangle 1.\ (\exists\, u, v : R(u, v) \land (v = y))$
$\equiv \exists\, u, v : \land\, IsP(u, v) \land (v = y)$
$\land\, \Box[NsP(u, v, u', v')]_{\langle u, v \rangle}$

BY DEF R

$\langle 4 \rangle 2.\ (\exists\, u, v : R(u, v) \land (v = y))$
$\equiv \exists\, u, v : IsP(u, v) \land (v = y)$

BY $\langle 4 \rangle 1$ just stutter forever the initial state

$\langle 4 \rangle 3.\ (\exists\, u, v : IsP(u, v) \land (v = y))$
$\equiv \exists\, p : IsP(p, y)$

$\langle 5 \rangle 1.\ (\exists\, u, v : IsP(u, v) \land (v = y))$
$\equiv \exists\, u, v : IsP(u, y) \land (v = y)$

OBVIOUS

$\langle 5 \rangle 2.\ (\exists\, u, v : IsP(u, y) \land (v = y))$
$\equiv \exists\, u : IsP(u, y) \land \exists\, v : v = y$

OBVIOUS

$\langle 5 \rangle 3.\ (\exists\, u : IsP(u, y) \land \exists\, v : v = y)$
$\equiv \exists\, u : IsP(u, y)$

OBVIOUS

$\langle 5 \rangle 4.\ (\exists\, u : IsP(u, y)) \equiv \exists\, p : IsP(p, y)$

OBVIOUS

$\langle 5 \rangle$ QED
BY $\langle 5 \rangle 1$, $\langle 5 \rangle 2$, $\langle 5 \rangle 3$, $\langle 5 \rangle 4$

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

$\langle 3 \rangle$ QED
BY $\langle 3 \rangle 2$, $\langle 3 \rangle 3$

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

The stepwise implication (part of safety).

$\langle 1 \rangle 4.\ (sigma \models \forall\, b : (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b))$
$\equiv sigma, 0 \models$ at this point we have to use past raw TLA+ to accommodate for the operator *Earlier*.

$\lor \neg \lor \neg Ie$
$\lor\, Je$
$\lor \land Is$
$\land\, Ie \lor \Box(Next \land SysNext)$
$\land\, Ie \Rightarrow \Box(Earlier(EnvNext) \Rightarrow \land\, Earlier(Next)$
$\land\, SysNext)$

$\langle 2 \rangle 7.$ ASSUME VARIABLE u, VARIABLE v
PROVE
$\lor \neg \land\, b = \text{TRUE}$
$\land\, \Box[b' = \text{FALSE}]_b$
$\lor\, \Box[b' \Rightarrow b]_{\langle u, v, x, y \rangle}$

$\langle 3 \rangle 1.\ \lor \neg \land b \in \text{BOOLEAN}$
$\quad\quad\quad \land [b' = \text{FALSE}]_b$
$\quad \lor b' \Rightarrow b$

OBVIOUS

$\langle 3 \rangle 2.\ \lor \neg \land b = \text{TRUE}$
$\quad\quad\quad \land \Box[b' = \text{FALSE}]_b$
$\quad \lor \Box(b \in \text{BOOLEAN}\ )$

OBVIOUS

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle 1.\ (\forall b : (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b))$
$\quad \equiv \forall b : \ \lor \neg MustUnstep(b)$
$\quad\quad\quad\quad \lor Fr(ClA, b) \Rightarrow FPH(ClG, b)$

OBVIOUS

$\langle 2 \rangle 2.\ (\forall b : (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b))$
$\quad \equiv \forall b : \ \lor \neg MustUnstep(b)$
$\quad\quad\quad\quad \lor Fr(Q, b) \Rightarrow FPH(R, b)$

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

$\langle 2 \rangle 3.$ ASSUME VARIABLE b

PROVE $\lor \neg MustUnstep(b)$
$\quad \lor Fr(Q, b) \equiv$
$\quad\quad \lor \neg IeP(x, y)$
$\quad\quad \lor \land JeP(x, y)$
$\quad\quad\quad \land \Box[b' \Rightarrow NeP(x, y, x', y')]_{\langle x, y \rangle}$

$\langle 3 \rangle$ USE DEF Ie, Je, Ne

$\langle 3 \rangle 1.\ Fr(Q, b) \equiv \exists u, v :$
$\quad \land Q(u, v)$
$\quad \land SamePrefix(b, u, v, x, y)$

BY DEF Fr, $Front$

$\langle 3 \rangle 2.\ \lor \neg MustUnstep(b)$
$\quad \lor (\exists u, v : \ \land Q(u, v)$
$\quad\quad\quad\quad \land SamePrefix(b, u, v, x, y))$
$\quad \equiv$
$\quad\quad \land Ie \Rightarrow \ \land Je$
$\quad\quad\quad\quad \land \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

$\langle 4 \rangle 1.\ \lor \neg MustUnstep(b)$
$\quad \lor \neg \exists u, v : \ \land Q(u, v)$
$\quad\quad\quad\quad \land SamePrefix(b, u, v, x, y)$
$\quad \lor Ie \Rightarrow \ \land Je$
$\quad\quad\quad \land \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

$\langle 5 \rangle$ DEFINE

$F \triangleq \ \land MustUnstep(b)$
$\quad\quad \land \exists u, v : \ \land Q(u, v)$
$\quad\quad\quad\quad \land SamePrefix(b, u, v, x, y)$

$\langle 5\rangle 1.\ \lor \neg F$

$\quad \lor \exists\, u,\ v :$

$\quad\quad \land\ Q(u,\ v)$

$\quad\quad \land\ \mathit{SamePrefix}(b,\ u,\ v,\ x,\ y)$

$\quad\quad \land\ \mathit{MustUnstep}(b)$

BY DEF F

$\langle 5\rangle 2.\ \lor \neg F$

$\quad \lor \exists\, u,\ v :$

$\quad\quad \land\ \lor \neg \mathit{IeP}(u,\ v)$

$\quad\quad\quad \lor\ \land\ \mathit{JeP}(u,\ v)$

$\quad\quad\quad\quad \land\ \Box[\mathit{NeP}(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$

$\quad\quad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$

$\quad\quad \land\ b = \text{TRUE}$

$\quad\quad \land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 5\rangle 1$ DEF Q, $\mathit{SamePrefix}$, $\mathit{MustUnstep}$

$\langle 5\rangle 6.\ \lor \neg F$

$\quad \lor \exists\, u,\ v :$

$\quad\quad \land\ \lor \neg \mathit{IeP}(u,\ v)$

$\quad\quad\quad \lor\ \land\ \mathit{JeP}(u,\ v)$

$\quad\quad\quad\quad \land\ \Box[$

$\quad\quad\quad\quad\quad [\mathit{NeP}(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$

$\quad\quad\quad\quad\quad]_{\langle u,\ v,\ x,\ y\rangle}$

$\quad\quad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$

$\quad\quad \land\ b = \text{TRUE}$

$\quad\quad \land\ \Box[b' \Rightarrow b]_{\langle u,\ v,\ x,\ y\rangle}$

BY $\langle 5\rangle 2$, $\langle 2\rangle 7$

$\langle 5\rangle 3.\ \lor \neg F$

$\quad \lor \exists\, u,\ v :$

$\quad\quad \land\ \lor \neg \mathit{IeP}(u,\ v)$

$\quad\quad\quad \lor\ \land\ \mathit{JeP}(u,\ v)$

$\quad\quad\quad\quad \land\ \Box[$

$\quad\quad\quad\quad\quad \land\ [\mathit{NeP}(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$

$\quad\quad\quad\quad\quad \land\ b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle)$

$\quad\quad\quad\quad\quad \land\ b' \Rightarrow (\langle u',\ v'\rangle = \langle x',\ y'\rangle)$

$\quad\quad\quad\quad\quad]_{\langle u,\ v,\ x,\ y\rangle}$

$\quad\quad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$

$\quad\quad \land\ b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle)$

$\quad\quad \land\ b = \text{TRUE}$

$\quad\quad \land\ \Box[b' \Rightarrow b]_{\langle u,\ v,\ x,\ y\rangle}$

BY $\langle 5\rangle 6$, $\mathit{RuleINV2}$

$\langle 5\rangle 4.\ \lor \neg F$

$\quad \lor \exists\, u,\ v :$

$\quad\quad \land\ \lor \neg \mathit{IeP}(u,\ v)$

$\quad\quad\quad \lor\ \land\ \mathit{JeP}(u,\ v)$

$\quad\quad\quad\quad \land\ \Box[$

$\land \lor \neg(b' \land b)$
$\quad \lor [NeP(x,\ y,\ x',\ y')]_{\langle x,\ y\rangle}$
$\land\ b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle)$
$\land\ b' \Rightarrow (\langle u',\ v'\rangle = \langle x',\ y'\rangle)$
$\land\ b' \Rightarrow b$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\land \langle u,\ v\rangle = \langle x,\ y\rangle$
BY $\langle 5\rangle 3$

$\langle 5\rangle 7.\ \lor \neg F$
$\lor \exists\, u,\ v :$
$\land \lor \neg IeP(u,\ v)$
$\quad \lor \land JeP(u,\ v)$
$\quad\quad \land \Box[$
$\quad\quad b' \Rightarrow [NeP(x,\ y,\ x',\ y')]_{\langle x,\ y\rangle}$
$\quad\quad]_{\langle u,\ v,\ x,\ y\rangle}$
$\land \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\land \langle u,\ v\rangle = \langle x,\ y\rangle$
BY $\langle 5\rangle 4$

$\langle 5\rangle 5.\ \lor \neg F$
$\lor \exists\, u,\ v :$
$\lor \neg IeP(x,\ y)$
$\lor \land JeP(x,\ y)$
$\quad \land \Box[b' \Rightarrow NeP(x,\ y,\ x',\ y')]_{\langle x,\ y\rangle}$
BY $\langle 5\rangle 4$

$\langle 5\rangle$ QED
$\langle 6\rangle 1.\ (\exists\, u,\ v :$
$IeP(x,\ y)$
$\Rightarrow \land JeP(x,\ y)$
$\quad \land \Box[b' \Rightarrow NeP(x,\ y,\ x',\ y')]_{\langle x,\ y\rangle})$
$\equiv$
$Ie \Rightarrow \land Je$
$\quad \land \Box[b' \Rightarrow Ne]_{\langle x,\ y\rangle}$
BY DEF $Ie,\ Je,\ Ne$
$\langle 6\rangle$ QED
BY $\langle 5\rangle 5,\ \langle 6\rangle 1$ DEF F

$\langle 4\rangle 2.\ \lor \neg MustUnstep(b)$
$\lor \neg Ie \Rightarrow \land Je$
$\quad \land \Box[b' \Rightarrow Ne]_{\langle x,\ y\rangle}$
$\lor \exists\, u,\ v :\ \land Q(u,\ v)$
$\quad \land SamePrefix(b,\ u,\ v,\ x,\ y)$
$\langle 5\rangle$ DEFINE
$H \triangleq \land MustUnstep(b)$
$\land Ie \Rightarrow \land Je$
$\quad \land \Box[b' \Rightarrow Ne]_{\langle x,\ y\rangle}$

$\langle 5\rangle 1.$ $\exists\, u,\ v :$
$\quad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\quad \land\ \Box[b']_{\langle u,\, v\rangle}$
OBVIOUS stutter u, v after b falls

$\langle 5\rangle 2.$ ASSUME VARIABLE u, VARIABLE v,
$b' \in$ BOOLEAN $\land\ [b']_{\langle u,\, v\rangle}$
PROVE $[(\neg b') \Rightarrow NeP(u,\ v,\ u',\ v')]_{\langle u,\, v\rangle}$

$\langle 6\rangle 1.$ SUFFICES ASSUME $\neg b' \land \neg$UNCHANGED $\langle u,\ v\rangle$
PROVE FALSE
OBVIOUS

$\langle 6\rangle 2.$ b'
$\langle 7\rangle 1.$ $\neg$UNCHANGED $\langle u,\ v\rangle$
BY $\langle 6\rangle 1$
$\langle 7\rangle$ QED
BY $\langle 7\rangle 1$, $\langle 5\rangle 2$

$\langle 6\rangle$ QED
BY $\langle 6\rangle 1$, $\langle 6\rangle 2$ goal from $\langle 6\rangle 1$

$\langle 5\rangle 3.$ $\lor\ \neg MustUnstep(b)$
$\lor\ \exists\, u,\ v :$
$\quad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\quad \land\ \Box[b']_{\langle u,\, v\rangle}$
$\quad \land\ \Box[(\neg b') \Rightarrow NeP(u,\ v,\ u',\ v')]_{\langle u,\, v\rangle}$

$\langle 6\rangle 1.$ $MustUnstep(b) \Rightarrow \Box(b \in$ BOOLEAN $)$
BY DEF $MustUnstep$
$\langle 6\rangle$ QED
BY $\langle 6\rangle 1$, $\langle 5\rangle 1$, $\langle 5\rangle 2$

$\langle 5\rangle 4.$ $\lor\ \neg H$
$\lor\ \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$
$\quad \land\ Ie \Rightarrow\ \land\ Je$
$\qquad\qquad \land\ \Box[b' \Rightarrow Ne]_{\langle x,\, y\rangle}$
BY $\langle 2\rangle 7$ DEF H, $MustUnstep$

$\langle 5\rangle 5.$ $\lor\ \neg H$
$\lor\ \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$
$\quad \land\ Ie \Rightarrow\ \land\ Je$
$\qquad\qquad \land\ \Box[b' \Rightarrow Ne]_{\langle x,\, y\rangle}$
$\quad \land\ \exists\, u,\ v :$
$\qquad \land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\qquad \land\ \Box[b']_{\langle u,\, v\rangle}$
$\qquad \land\ \Box[(\neg b') \Rightarrow NeP(u,\ v,\ u',\ v')]_{\langle u,\, v\rangle}$
BY $\langle 5\rangle 4$ DEF H

$\langle 5\rangle 6.$ $\lor\ \neg H$
$\lor\ \land\ b =$ TRUE
$\quad \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$
$\quad \land\ Ie \Rightarrow\ \land\ Je$
$\qquad\qquad \land\ \Box[b' \Rightarrow Ne]_{\langle x,\, y\rangle}$

$\land \; \exists\, u, v :$

$\land \; b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle)$

$\land \; \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$

$\land \; \Box[b']_{\langle u, v \rangle}$

$\land \; \Box[(\neg b') \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

$\land \; \lor \neg IeP(x, y)$

$\quad \lor \; \land \; JeP(x, y)$

$\quad\quad \land \; \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

BY $\langle 5 \rangle 5$ DEF H, $MustUnstep$, Ie, Je, Ne

$\langle 5 \rangle 7.$ $\lor \neg H$

$\lor \; \land \; \Box[b' \Rightarrow b]_{\langle u, v, x, y \rangle}$

$\quad \land \; Ie \Rightarrow \; \land \; Je$

$\quad\quad\quad \land \; \Box[b' \Rightarrow Ne]_{\langle x, y \rangle})$

$\quad \land \; \exists\, u, v :$

$\quad\quad \land \; \langle u, v \rangle = \langle x, y \rangle$

$\quad\quad \land \; \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$

$\quad\quad \land \; \Box[b']_{\langle u, v \rangle}$

$\quad\quad \land \; \Box[(\neg b') \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

$\quad\quad \land \; \lor \neg IeP(x, y)$

$\quad\quad\quad \lor \; \land \; JeP(x, y)$

$\quad\quad\quad\quad \land \; \Box[$

$\quad\quad\quad\quad\quad [b' \Rightarrow NeP(x, y, x', y')]_{\langle x, y \rangle}$

$\quad\quad\quad\quad\quad]_{\langle x, y, u, v \rangle}$

BY $\langle 5 \rangle 6$

$\langle 5 \rangle 8.$ $\lor \neg H$

$\lor \; \exists\, u, v :$

$\quad\quad \land \; \langle u, v \rangle = \langle x, y \rangle$

$\quad\quad \land \; \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$

$\quad\quad \land \; \Box[b']_{\langle u, v \rangle}$

$\quad\quad \land \; \Box[(\neg b') \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

$\quad\quad \land \; \lor \neg IeP(u, v)$

$\quad\quad\quad \lor \; \land \; JeP(u, v)$

$\quad\quad\quad\quad \land \; \Box[$

$\quad\quad\quad\quad\quad \land \; [b' \Rightarrow NeP(x, y, x', y')]_{\langle x, y \rangle}$

$\quad\quad\quad\quad\quad \land \; b' \Rightarrow b$

$\quad\quad\quad\quad\quad \land \; b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle)$

$\quad\quad\quad\quad\quad \land \; b' \Rightarrow (\langle u', v' \rangle = \langle x', y' \rangle)$

$\quad\quad\quad\quad\quad]_{\langle x, y, u, v \rangle}$

BY $\langle 5 \rangle 7$

$\langle 5 \rangle 9.$ $\lor \neg H$

$\lor \; \exists\, u, v :$

$\quad\quad \land \; \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$

$\quad\quad \land \; \Box[(\neg b') \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

$\quad\quad \land \; \lor \neg IeP(u, v)$

$\quad\quad\quad \lor \; \land \; JeP(u, v)$

$\land \Box[$
$\quad \land [b' \Rightarrow NeP(x, y, x', y')]_{\langle x, y \rangle}$
$\quad \land b' \Rightarrow (\langle u, v \rangle = \langle x, y \rangle)$
$\quad \land b' \Rightarrow (\langle u', v' \rangle = \langle x', y' \rangle)$
$\quad]_{\langle x, y, u, v \rangle}$

BY $\langle 5 \rangle 8$

$\langle 5 \rangle 10.$ $\lor \neg H$
$\quad \lor \exists\, u, v :$
$\qquad \land \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$
$\qquad \land \Box[(\neg b') \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$
$\qquad \land \lor \neg IeP(u, v)$
$\qquad \quad \lor \land Je(u, v)$
$\qquad \qquad \land \Box[b' \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

BY $\langle 5 \rangle 9$

$\langle 5 \rangle 11.$ $\lor \neg H$
$\quad \lor \exists\, u, v :$
$\qquad \land \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$
$\qquad \land \lor \neg IeP(u, v)$
$\qquad \land \lor \land JeP(u, v)$
$\qquad \qquad \land \Box[(\neg b') \Rightarrow$
$\qquad \qquad \qquad NeP(u, v, u', v')]_{\langle u, v \rangle}$
$\qquad \qquad \land \Box[b' \Rightarrow NeP(u, v, u', v')]_{\langle u, v \rangle}$

BY $\langle 5 \rangle 10$

$\langle 5 \rangle 12.$ $\lor \neg H$
$\quad \lor \exists\, u, v :$
$\qquad \land \Box(b \Rightarrow (\langle u, v \rangle = \langle x, y \rangle))$
$\qquad \land IeP(u, v) \Rightarrow$
$\qquad \quad \land JeP(u, v)$
$\qquad \quad \land \Box[NeP(u, v, u', v')]_{\langle u, v \rangle}$

$\langle 6 \rangle 1.$ $H \Rightarrow \Box(b \in$ BOOLEAN $)$
BY DEF H, *MustUnstep*

$\langle 6 \rangle$ QED
BY $\langle 5 \rangle 11$, $\langle 6 \rangle 1$

$\langle 5 \rangle 13.$ $\lor \neg H$
$\quad \lor \exists\, u, v :$
$\qquad \land Q(u, v)$
$\qquad \land SamePrefix(b, u, v, x, y)$

BY $\langle 5 \rangle 12$ DEF Q, *SamePrefix*

$\langle 5 \rangle$ QED
BY $\langle 5 \rangle 13$ DEF H

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

$\langle 3 \rangle$ QED *TODO*: turn into a SUFFICES to reduce indentation
BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$ DEF *Ie*, *Je*, *Ne*

$\langle 2\rangle 4.$ ASSUME VARIABLE b

PROVE $\lor \neg MustUnstep(b)$
$\lor FPH(R, b) \equiv \land IsP(x, y)$
$\land \Box[b' \Rightarrow NsP(x, y, x', y')]_{\langle x, y\rangle}$
$\land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$

$\langle 3\rangle$ USE DEF Is, Ns

In this direction, we derive a quantified formula that is independent of the bound variables u and v. This allows us to eliminate the temporal quantifier $\boldsymbol{\exists}$.

$\langle 3\rangle 1.$ $FPH(R, b) \equiv \boldsymbol{\exists}\, u, v :$
$\land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y)$

BY DEF FPH, $FrontPlusHalf$

$\langle 3\rangle 2.$ SUFFICES
$\lor \neg MustUnstep(b)$
$\lor (\boldsymbol{\exists}\, u, v : \land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y))$
$\equiv \land Is$
$\land \Box[b' \Rightarrow Ns]_{\langle x, y\rangle}$
$\land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$

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

$\langle 3\rangle 3.$ $\lor \neg MustUnstep(b)$
$\lor \neg \boldsymbol{\exists}\, u, v : \land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y)$
$\lor \land Is$
$\land \Box[b' \Rightarrow Ns]_{\langle x, y\rangle}$
$\land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$

$\langle 4\rangle$ DEFINE

$F \triangleq \land MustUnstep(b)$
$\land \boldsymbol{\exists}\, u, v : \land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y)$

$\langle 4\rangle 1.$ $\lor \neg F$
$\lor \boldsymbol{\exists}\, u, v : \land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y)$

BY DEF F

$\langle 4\rangle 2.$ $\lor \neg F$
$\lor \boldsymbol{\exists}\, u, v :$
$\land IsP(u, v)$
$\land \Box[NsP(u, v, u', v')]_{\langle u, v\rangle}$
$\land \Box(b \Rightarrow (\langle u, v\rangle = \langle x, y\rangle))$

$\land\ v = y$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$

BY $\langle 4 \rangle 1$ DEF R, *SamePrefix*, *PlusHalf*

$\langle 4 \rangle 3.\ \lor\ \neg F$
$\lor\ \exists\, u,\, v :$
$\land\ IsP(u,\, v)$
$\land\ \Box[NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle))$
$\land\ v = y$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4 \rangle 2$ DEF F, *MustUnstep*

$\langle 4 \rangle 4.\ \lor\ \neg F$
$\lor\ \exists\, u,\, v :$
$\land\ IsP(u,\, v)$
$\land\ \Box[NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle))$
$\land\ b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle$
$\land\ v = y$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4 \rangle 3$

$\langle 4 \rangle 5.\ \lor\ \neg F$
$\lor\ \exists\, u,\, v :$
$\land\ IsP(u,\, v)$
$\land\ \Box[NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle))$
$\land\ \langle u,\, v \rangle = \langle x,\, y \rangle$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4 \rangle 4$

$\langle 4 \rangle 6.\ \lor\ \neg F$
$\lor\ \exists\, u,\, v :$
$\land\ IsP(x,\, y)$
$\land\ \Box[NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle))$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4 \rangle 5$

$\langle 4 \rangle 7.\ \lor\ \neg F$
$\lor\ \exists\, u,\, v :$

$\land\ IsP(x,\ y)$
$\land\ \Box[NsP(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\ v,\ y\rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4\rangle 6$

$\langle 4\rangle 8.\ \lor \neg F$
$\lor\ \exists\, u,\ v :$
$\land\ IsP(x,\ y)$
$\land\ \Box[$
$[NsP(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\land\ \Box[$
$[b \Rightarrow (v' = y')]_{\langle b,\ v,\ y\rangle}$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4\rangle 7$

$\langle 4\rangle 9.\ \lor \neg F$
$\lor\ \exists\, u,\ v :$
$\land\ IsP(x,\ y)$
$\land\ \Box[$
$\land\ b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle)$
$\land\ b' \Rightarrow (\langle u',\ v'\rangle = \langle x',\ y'\rangle)$
$\land\ [NsP(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land\ \Box[$
$\land\ b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle)$
$\land\ [b \Rightarrow (v' = y')]_{\langle b,\ v,\ y\rangle}$
$\land\ [NsP(u,\ v,\ u',\ v')]_{\langle u,\ v\rangle}$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land\ \Box(b \Rightarrow (\langle u,\ v\rangle = \langle x,\ y\rangle))$
$\land\ \Box[$
$[b \Rightarrow (v' = y')]_{\langle b,\ v,\ y\rangle}$
$]_{\langle u,\ v,\ x,\ y\rangle}$
$\land\ b = \text{TRUE}$
$\land\ \Box[b' = \text{FALSE}]_b$

BY $\langle 4\rangle 8$

$\langle 4\rangle 10.\ \lor \neg F$
$\lor\ \exists\, u,\ v :$
$\land\ IsP(x,\ y)$
$\land\ \Box[$
$(b \land b') \Rightarrow [NsP(x,\ y,\ x',\ y')]_{\langle x,\ y\rangle}$

$]_{\langle u,\, v,\, x,\, y \rangle}$
$\land \Box[$
$\quad \land b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle)$
$\quad \land [b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\quad \land [NsP(u,\, v,\, u',\, v')]_v$
$]_{\langle u,\, v,\, x,\, y \rangle}$
$\land b = \text{TRUE}$
$\land \Box[b' = \text{FALSE}]_b$

BY $\langle 4 \rangle 9$

$\langle 4 \rangle 11.$ $\lor \neg F$
$\lor \exists u,\, v :$
$\quad \land IsP(x,\, y)$
$\quad \land \Box[$
$\qquad \land b' \Rightarrow b$
$\qquad \land (b \land b') \Rightarrow [NsP(x,\, y,\, x',\, y')]_{\langle x,\, y \rangle}$
$\quad]_{\langle u,\, v,\, x,\, y \rangle}$
$\quad \land \Box[$
$\qquad \land b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle)$
$\qquad \land [b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\qquad \land b \Rightarrow [NsP(x,\, y,\, u',\, y')]_y$
$\quad]_{\langle u,\, v,\, x,\, y \rangle}$
$\quad \land b = \text{TRUE}$
$\quad \land \Box[b' = \text{FALSE}]_b$

$\langle 5 \rangle 1.$ ASSUME VARIABLE u, VARIABLE v
$\quad \land b \Rightarrow (\langle u,\, v \rangle = \langle x,\, y \rangle)$
$\quad \land [b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\quad \land [NsP(u,\, v,\, u',\, v')]_v$
PROVE $b \Rightarrow [NsP(x,\, y,\, u',\, y')]_y$

$\langle 6 \rangle 1.$ SUFFICES ASSUME $b \land \neg$UNCHANGED y
PROVE $NsP(x,\, y,\, u',\, y')$
OBVIOUS

$\langle 6 \rangle 2.$ $(u = x) \land (v = y)$
$\langle 7 \rangle 1.$ b
BY $\langle 6 \rangle 1$
$\langle 7 \rangle 2.$ $\langle u,\, v \rangle = \langle x,\, y \rangle$
BY $\langle 5 \rangle 1$, $\langle 7 \rangle 1$
$\langle 7 \rangle$ QED
BY $\langle 7 \rangle 2$

$\langle 6 \rangle 3.$ $v' = y'$
$\langle 7 \rangle 1.$ b
BY $\langle 6 \rangle 1$
$\langle 7 \rangle 2.$ $b \Rightarrow (v' = y')$
$\langle 8 \rangle 1.$ $y' \neq y$
BY $\langle 6 \rangle 1$
$\langle 8 \rangle$ QED

BY $\langle 5\rangle 1,\ \langle 8\rangle 1$

$\langle 7\rangle$ QED

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

$\langle 6\rangle 4.\ v' \neq v$

$\langle 7\rangle 1.\ y' \neq y$

BY $\langle 6\rangle 1$

$\langle 7\rangle 2.\ (v = y) \land (v' = y')$

BY $\langle 6\rangle 2,\ \langle 6\rangle 3$

$\langle 7\rangle$ QED

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

$\langle 6\rangle 5.\ NsP(u,\ v,\ u',\ v')$

BY $\langle 5\rangle 1,\ \langle 6\rangle 4$

$\langle 6\rangle$ QED

$\langle 7\rangle 1.\ (u = x) \land (v = y) \land (v' = y')$

BY $\langle 6\rangle 2,\ \langle 6\rangle 3,\ \langle 6\rangle 4$

$\langle 7\rangle$ QED

BY $\langle 6\rangle 5,\ \langle 7\rangle 1$ goal from $\langle 6\rangle 1$

$\langle 5\rangle$ QED

BY $\langle 4\rangle 10,\ \langle 2\rangle 7,\ \langle 5\rangle 1$

$\langle 4\rangle 12.\ \lor \neg F$

$\lor\ \exists\ u,\ v :$

$\land\ IsP(x,\ y)$

$\land\ \Box[$

$b' \Rightarrow [NsP(x,\ y,\ x',\ y')]_{\langle x,\, y\rangle}$

$]_{\langle u,\, v,\, x,\, y\rangle}$

$\land\ \Box[$

$b \Rightarrow [\exists\, r :\ \ NsP(x,\ y,\ r,\ y')]_y$

$]_{\langle u,\, v,\, x,\, y\rangle}$

BY $\langle 4\rangle 11$

$\langle 4\rangle 13.\ \lor \neg F$

$\lor\ \exists\ u,\ v :$

$\land\ IsP(x,\ y)$

$\land\ \Box[b' \Rightarrow NsP(x,\ y,\ x',\ y')]_{\langle x,\, y\rangle}$

$\land\ \Box[b \Rightarrow \exists\, r :\ \ NsP(x,\ y,\ r,\ y')]_y$

BY $\langle 4\rangle 12$

$\langle 4\rangle 14.\ \lor \neg F$

$\lor\ \land\ IsP(x,\ y)$

$\land\ \Box[b' \Rightarrow NsP(x,\ y,\ x',\ y')]_{\langle x,\, y\rangle}$

$\land\ \Box[b \Rightarrow \exists\, r :\ \ NsP(x,\ y,\ r,\ y')]_y$

BY $\langle 4\rangle 13$

$\langle 4\rangle$ QED

BY $\langle 4\rangle 14$ DEF $F,\ Is,\ Ns$

$\langle 3\rangle 4.\ \lor \neg \land\ MustUnstep(b)$

$\land\ Is$

$\land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle}$
$\land \Box[b \Rightarrow \exists\, r : \; NsP(x,\, y,\, r,\, y')]_y$
$\lor$ **∃** $u,\, v : \; \land R(u,\, v)$
$\land \mathit{SamePrefix}(b,\, u,\, v,\, x,\, y)$
$\land \mathit{PlusHalf}(b,\, v,\, y)$

⟨4⟩ DEFINE
$H \triangleq \land \mathit{MustUnstep}(b)$
$\land \mathit{Is}$
$\land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle}$
$\land \Box[b \Rightarrow \exists\, r : \; NsP(x,\, y,\, r,\, y')]_y$

⟨4⟩1. **∃** $u,\, v$:
$\land \Box[b]_{\langle u,\, v \rangle}$ stuttering tail
same prefix
$\land \Box(b \Rightarrow (\langle x,\, y \rangle = \langle u,\, v \rangle))$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land \Box[(b \land \neg b') \Rightarrow$ falling edge
$\land v' = y'$
$\land u' =$ IF $y' = y$ THEN u
ELSE CHOOSE $r : \; NsP(x,\, y,\, r,\, y')$
$]_{\langle b,\, v,\, y \rangle}$
OMITTED *TODO*

⟨4⟩2. ASSUME VARIABLE u, VARIABLE v,
$b' \in$ BOOLEAN $\land [b]_{\langle u,\, v \rangle}$
PROVE $[(\neg b) \Rightarrow NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
OBVIOUS

⟨4⟩3. $\lor \neg \mathit{MustUnstep}(b)$
$\lor$ **∃** $u,\, v$:
$\land \Box[b]_{\langle u,\, v \rangle}$
$\land \Box(b \Rightarrow (\langle x,\, y \rangle = \langle u,\, v \rangle))$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y \rangle}$
$\land \Box[(b \land \neg b') \Rightarrow$
$\land v' = y'$
$\land u' =$ IF $y' = y$ THEN u
ELSE CHOOSE $r : \; NsP(x,\, y,\, r,\, y')$
$]_{\langle b,\, v,\, y \rangle}$
$\land \Box[(\neg b) \Rightarrow NsP(u,\, v,\, u',\, v')]_{\langle u,\, v \rangle}$
BY ⟨4⟩1, ⟨4⟩2

⟨4⟩4. $\lor \neg H$
$\lor \land b =$ TRUE
$\land \Box(b \in$ BOOLEAN $)$
$\land \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y \rangle}$
$\land \mathit{Is}$
$\land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle}$
$\land \Box[b \Rightarrow \exists\, r : \; NsP(x,\, y,\, r,\, y')]_y$

BY $\langle 2\rangle 7$ DEF H, $\mathit{MustUnstep}$

$\langle 4\rangle 5.\ \lor \neg H$

$\quad \lor \ \land\ b = \text{TRUE}$

$\qquad \land\ \Box(b \in \text{BOOLEAN}\ )$

$\qquad \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$

$\qquad \land\ \mathit{IsP}(x,\ y)$

$\qquad \land\ \Box[b' \Rightarrow \mathit{NsP}(x,\ y,\ x',\ y')]_{\langle x,\, y\rangle}$

$\qquad \land\ \Box[b \Rightarrow \exists\, r :\ \mathit{NsP}(x,\ y,\ r,\ y')]_{y}$

$\qquad \land\ \boldsymbol{\exists}\, u,\ v :$

$\qquad\quad \land\ \Box[b]_{\langle u,\, v\rangle}$

$\qquad\quad \land\ \Box(b \Rightarrow (\langle x,\ y\rangle = \langle u,\ v\rangle))$

$\qquad\quad \land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y\rangle}$

$\qquad\quad \land\ \Box[(b \land \neg b') \Rightarrow$

$\qquad\qquad\quad \land\ v' = y'$

$\qquad\qquad\quad \land\ u' = \text{IF}\ \ y' = y\ \ \text{THEN}\ \ u$

$\qquad\qquad\qquad\quad \text{ELSE}\ \ \text{CHOOSE}\ r :\ \mathit{NsP}(x,\ y,\ r,\ y')$

$\qquad\qquad]_{\langle b,\, v,\, y\rangle}$

$\qquad\quad \land\ \Box[(\neg b) \Rightarrow \mathit{NsP}(u,\ v,\ u',\ v')]_{\langle u,\, v\rangle}$

BY $\langle 4\rangle 3$, $\langle 4\rangle 4$ DEF H, Is, NsP

$\langle 4\rangle 6.\ \lor \neg H$

$\quad \lor\ \boldsymbol{\exists}\, u,\ v :$

$\qquad\quad \land\ b = \text{TRUE}$

$\qquad\quad \land\ b \Rightarrow \langle u,\ v\rangle = \langle x,\ y\rangle$

$\qquad\quad \land\ \mathit{IsP}(x,\ y)$

$\qquad\quad \land\ \Box(b \in \text{BOOLEAN}\ )$

$\qquad\quad \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$

$\qquad\quad \land\ \Box[b' \Rightarrow \mathit{NsP}(x,\ y,\ x',\ y')]_{\langle x,\, y\rangle}$

$\qquad\quad \land\ \Box[b \Rightarrow \exists\, r :\ \mathit{NsP}(x,\ y,\ r,\ y')]_{y}$

$\qquad\quad \land\ \Box[b]_{\langle u,\, v\rangle}$

$\qquad\quad \land\ \Box(b \Rightarrow (\langle x,\ y\rangle = \langle u,\ v\rangle))$

$\qquad\quad \land\ \Box[b \Rightarrow (v' = y')]_{\langle b,\, v,\, y\rangle}$

$\qquad\quad \land\ \Box[(b \land \neg b') \Rightarrow$

$\qquad\qquad\quad \land\ v' = y'$

$\qquad\qquad\quad \land\ u' = \text{IF}\ \ y' = y\ \ \text{THEN}\ \ u$

$\qquad\qquad\qquad\quad \text{ELSE}\ \ \text{CHOOSE}\ r :\ \mathit{NsP}(x,\ y,\ r,\ y')$

$\qquad\qquad]_{\langle b,\, v,\, y\rangle}$

$\qquad\quad \land\ \Box[(\neg b) \Rightarrow \mathit{NsP}(u,\ v,\ u',\ v')]_{\langle u,\, v\rangle}$

BY $\langle 4\rangle 5$

$\langle 4\rangle 7.\ \lor \neg H$

$\quad \lor\ \boldsymbol{\exists}\, u,\ v :$

$\qquad\quad \land\ \langle u,\ v\rangle = \langle x,\ y\rangle$

$\qquad\quad \land\ \mathit{IsP}(u,\ v)$

$\qquad\quad \land\ \Box(b \in \text{BOOLEAN}\ )$

$\qquad\quad \land\ \Box[b' \Rightarrow b]_{\langle u,\, v,\, x,\, y\rangle}$

$\qquad\quad \land\ \Box[$

$[b' \Rightarrow NsP(x,\ y,\ x',\ y')]_{\langle x,\ y \rangle}$
$]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box[$
$[b \Rightarrow \exists\, r : \ NsP(x,\ y,\ r,\ y')]_y$
$]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box(b \Rightarrow (\langle x,\ y \rangle = \langle u,\ v \rangle))$
$\land v = y$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b,\ v,\ y \rangle}$
$\land \Box[$
$[(b \land \neg b') \Rightarrow$
$\land v' = y'$
$\land u' = \text{IF } y' = y \text{ THEN } u$
$\text{ELSE CHOOSE } r : \ NsP(x,\ y,\ r,\ y')$
$]_{\langle b,\ v,\ y \rangle}$
$]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box[(\neg b) \Rightarrow NsP(u,\ v,\ u',\ v')]_{\langle u,\ v \rangle}$

BY $\langle 4 \rangle 6$

$\langle 4 \rangle 8.\ \lor \neg H$
$\lor\ \exists\, u,\ v :$
$\land IsP(u,\ v)$
$\land \Box(b \in \text{BOOLEAN})$
$\land \Box[b' \Rightarrow b]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box[$
$\land b' \Rightarrow b$
$\land b \Rightarrow (\langle x,\ y \rangle = \langle u,\ v \rangle)$
$\land b' \Rightarrow (\langle x',\ y' \rangle = \langle u',\ v' \rangle)$
$\land b' \Rightarrow [NsP(x,\ y,\ x',\ y')]_{\langle x,\ y \rangle}$
$]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box(b \Rightarrow (\langle x,\ y \rangle = \langle u,\ v \rangle))$
$\land v = y$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b,\ v,\ y \rangle}$
$\land \Box[$
$\land b \Rightarrow (\langle x,\ y \rangle = \langle u,\ v \rangle)$
$\land [b \Rightarrow \exists\, r : \ NsP(x,\ y,\ r,\ y')]_y$
$\land\ \lor \neg(b \land \neg b')$
$\lor\ \land v' = y'$
$\land u' = \text{IF } y' = y \text{ THEN } u$
$\text{ELSE CHOOSE } r : \ NsP(x,\ y,\ r,\ y')$
$]_{\langle u,\ v,\ x,\ y \rangle}$
$\land \Box[(\neg b) \Rightarrow NsP(u,\ v,\ u',\ v')]_{\langle u,\ v \rangle}$

$\langle 5 \rangle 1.$ ASSUME
$\land b \in \text{BOOLEAN}$
$\land b' \in \text{BOOLEAN}$
$\land b \land \neg b'$
PROVE

```
            ¬UNCHANGED b
        OBVIOUS
    ⟨5⟩2. ASSUME VARIABLE u, VARIABLE v,
                 ∧ b ∈ BOOLEAN
                 ∧ b′ ∈ BOOLEAN
           PROVE
                 ∨ ¬ ∨ ¬(b ∧ ¬b′)
                     ∨ ∧ v′ = y′
                       ∧ u′ = IF y′ = y THEN u
                                 ELSE CHOOSE r : NsP(x, y, r, y′)
                     ∨ UNCHANGED ⟨b, v, y⟩
                 ∨ ∨ ¬(b ∧ ¬b′)
                   ∨ ∧ ¬UNCHANGED b
                     ∧ ∨ ∧ v′ = y′
                           ∧ u′ = IF y′ = y THEN u
                                     ELSE CHOOSE r : NsP(x, y, r, y′)
                       ∨ UNCHANGED ⟨b, v, y⟩
        BY ⟨5⟩1
    ⟨5⟩3. ASSUME VARIABLE u, VARIABLE v,
                 ∧ b ∈ BOOLEAN
                 ∧ b′ ∈ BOOLEAN
           PROVE
                 ∨ ¬ ∨ ¬(b ∧ ¬b′)
                     ∨ ∧ v′ = y′
                       ∧ u′ = IF y′ = y THEN u
                                 ELSE CHOOSE r : NsP(x, y, r, y′)
                     ∨ UNCHANGED ⟨b, v, y⟩
                 ∨ ∨ ¬(b ∧ ¬b′)
                   ∨ ∧ v′ = y′
                     ∧ u′ = IF y′ = y THEN u
                               ELSE CHOOSE r : NsP(x, y, r, y′)
        BY ⟨5⟩2
    ⟨5⟩ QED
        BY ⟨4⟩7, ⟨5⟩3
⟨4⟩9. ∨ ¬H
      ∨ ∃ u, v :
            ∧ IsP(u, v)
            ∧ □(b ∈ BOOLEAN )
            ∧ □(b ⇒ (⟨x, y⟩ = ⟨u, v⟩))
            ∧ v = y
            ∧ □[b ⇒ (v′ = y′)]_⟨b, v, y⟩
            ∧ □[
                ∧ (b′ ∧ b) ⇒ (⟨x, y⟩ = ⟨u, v⟩)
                ∧ (b′ ∧ b) ⇒ (⟨x′, y′⟩ = ⟨u′, v′⟩)
```

$\wedge (b' \wedge b) \Rightarrow [NsP(u, v, u', v')]_{\langle u, v \rangle}$
$]_{\langle u, v, x, y \rangle}$
$\wedge \Box[$
$\wedge b \Rightarrow (\langle x, y \rangle = \langle u, v \rangle)$
$\wedge [b \Rightarrow \exists r : NsP(x, y, r, y')]_y$
$\wedge \vee \neg(b \wedge \neg b')$
$\vee \wedge v' = y'$
$\wedge u' =$ IF $y' = y$ THEN u
ELSE CHOOSE $r : NsP(x, y, r, y')$
$]_{\langle u, v, x, y \rangle}$
$\wedge \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$

BY $\langle 4 \rangle 8$

$\langle 4 \rangle 10.$ $\vee \neg H$
$\vee \exists u, v :$
$\wedge IsP(u, v)$
$\wedge \Box(b \in$ BOOLEAN $)$
$\wedge \Box(b \Rightarrow (\langle x, y \rangle = \langle u, v \rangle))$
$\wedge v = y$
$\wedge \Box[b \Rightarrow (v' = y')]_{\langle b, v, y \rangle}$

$\wedge \Box[$
$[(b' \wedge b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$]_{\langle u, v, x, y \rangle}$
$\wedge \Box[$
$\vee \neg(b \wedge \neg b')$
$\vee \wedge [\exists r : NsP(u, v, r, y')]_y$
$\wedge \langle x, y \rangle = \langle u, v \rangle$
$\wedge v' = y'$
$\wedge u' =$ IF $v' = v$ THEN u
ELSE CHOOSE $r : NsP(u, v, r, v')$
$]_{\langle u, v, x, y \rangle}$
$\wedge \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$

BY $\langle 4 \rangle 9$

$\langle 4 \rangle 11.$ $\vee \neg H$
$\vee \exists u, v :$
$\wedge IsP(u, v)$
$\wedge \Box(b \in$ BOOLEAN $)$
$\wedge \Box(b \Rightarrow (\langle x, y \rangle = \langle u, v \rangle))$
$\wedge v = y$
$\wedge \Box[b \Rightarrow (v' = y')]_{\langle b, v, y \rangle}$

$\wedge \Box[(b \wedge b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$\wedge \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$\wedge \Box[$
$\vee \neg(b \wedge \neg b')$

$\lor \land [\exists r : NsP(u, v, r, v')]_v$
$\land \langle x, y \rangle = \langle u, v \rangle$
$\land v' = y'$
$\land u' = \text{IF } v' = v \text{ THEN } u$
$\text{ELSE CHOOSE } r : NsP(u, v, r, v')$
$]_{\langle u, v, x, y \rangle}$

BY ⟨4⟩10

⟨4⟩12. ASSUME VARIABLE u, VARIABLE v,
$\land b \in \text{BOOLEAN}$
$\land b' \in \text{BOOLEAN}$
$\land \neg \text{UNCHANGED } \langle u, v \rangle$
$\land [\exists r : NsP(u, v, r, v')]_v$
$\land u' = \text{IF } v' = v \text{ THEN } u$
$\text{ELSE CHOOSE } r : NsP(u, v, r, v')$

PROVE
$NsP(u, v, u', v')$

⟨5⟩1. ASSUME UNCHANGED v
PROVE FALSE

⟨6⟩1. $u' = \text{IF } v' = v \text{ THEN } u$
$\text{ELSE CHOOSE } r : NsP(u, v, r, v')$
BY ⟨4⟩12

⟨6⟩2. $u' = u$
BY ⟨5⟩1, ⟨6⟩1

⟨6⟩3. UNCHANGED $\langle u, v \rangle$
BY ⟨5⟩1, ⟨6⟩2

⟨6⟩4. $\neg$UNCHANGED $\langle u, v \rangle$
BY ⟨4⟩12

⟨6⟩ QED
BY ⟨6⟩3, ⟨6⟩4

⟨5⟩2.CASE $\neg$UNCHANGED v

⟨6⟩1. $\exists r : NsP(u, v, r, v')$
BY ⟨4⟩12, ⟨5⟩2

⟨6⟩2. $u' = \text{CHOOSE } r : NsP(u, v, r, v')$

⟨7⟩1. $u' = \text{IF } v' = v \text{ THEN } u$
$\text{ELSE CHOOSE } r : NsP(u, v, r, v')$
BY ⟨4⟩12

⟨7⟩ QED
BY ⟨7⟩1, ⟨5⟩2

⟨6⟩ QED
BY ⟨6⟩1, ⟨6⟩2

⟨4⟩13. ASSUME VARIABLE u, VARIABLE v,
$\land b \in \text{BOOLEAN}$
$\land b' \in \text{BOOLEAN}$
$\land [\exists r : NsP(u, v, r, v')]_v$
$\land u' = \text{IF } v' = v \text{ THEN } u$

$\quad\quad\quad\quad\quad\quad \text{ELSE CHOOSE } r : \; NsP(u, v, r, v')$

$\quad\quad \text{PROVE } [NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad \text{BY } \langle 4\rangle 12$

$\langle 4\rangle 14. \lor \neg H$

$\quad\quad \lor \exists u, v :$

$\quad\quad\quad \land IsP(u, v)$

$\quad\quad\quad \land \Box(b \in \text{BOOLEAN})$

$\quad\quad\quad \land \Box(b \Rightarrow (\langle x, y\rangle = \langle u, v\rangle))$

$\quad\quad\quad \land v = y$

$\quad\quad\quad \land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y\rangle}$

$\quad\quad\quad \land \Box[(b \land b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[$

$\quad\quad\quad\quad \lor \neg(b \land \neg b')$

$\quad\quad\quad\quad \lor \land [\exists r : \; NsP(u, v, r, v')]_v$

$\quad\quad\quad\quad\quad \land u' = \text{IF } v' = v \text{ THEN } u$

$\quad\quad\quad\quad\quad\quad\quad\quad \text{ELSE CHOOSE } r : \; NsP(u, v, r, v')$

$\quad\quad\quad\quad\quad \land [NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad\quad]_{\langle u, v, x, y\rangle}$

$\quad \text{BY } \langle 4\rangle 11, \langle 4\rangle 13$

$\langle 4\rangle 15. \lor \neg H$

$\quad\quad \lor \exists u, v :$

$\quad\quad\quad \land IsP(u, v)$

$\quad\quad\quad \land \Box(b \in \text{BOOLEAN})$

$\quad\quad\quad \land \Box(b \Rightarrow (\langle x, y\rangle = \langle u, v\rangle))$

$\quad\quad\quad \land v = y$

$\quad\quad\quad \land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y\rangle}$

$\quad\quad\quad \land \Box[(b \land b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[$

$\quad\quad\quad\quad (b \land \neg b') \Rightarrow [NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad\quad]_{\langle u, v, x, y\rangle}$

$\quad \text{BY } \langle 4\rangle 12$

$\langle 4\rangle 16. \lor \neg H$

$\quad\quad \lor \exists u, v :$

$\quad\quad\quad \land IsP(u, v)$

$\quad\quad\quad \land \Box(b \in \text{BOOLEAN})$

$\quad\quad\quad \land \Box(b \Rightarrow (\langle x, y\rangle = \langle u, v\rangle))$

$\quad\quad\quad \land v = y$

$\quad\quad\quad \land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y\rangle}$

$\quad\quad\quad \land \Box[(b \land b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v\rangle}$

$\quad\quad\quad \land \Box[$

$[(b \land \neg b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$]_{\langle u, v, x, y \rangle}$

BY $\langle 4 \rangle 15$

$\langle 4 \rangle 17.$ $\lor \neg H$
$\lor \exists u, v :$
$\land IsP(u, v)$
$\land \Box(b \in \text{BOOLEAN})$
$\land \Box(b \Rightarrow (\langle x, y \rangle = \langle u, v \rangle))$
$\land v = y$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y \rangle}$

$\land \Box[(\neg b) \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$\land \Box[(b \land b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$
$\land \Box[(b \land \neg b') \Rightarrow NsP(u, v, u', v')]_{\langle u, v \rangle}$

BY $\langle 4 \rangle 16$

$\langle 4 \rangle 18.$ $\lor \neg H$
$\lor \exists u, v :$
$\land IsP(u, v)$
$\land \Box[NsP(u, v, u', v')]_{\langle u, v \rangle}$
$\land \Box(b \in \text{BOOLEAN})$
$\land \Box(b \Rightarrow (\langle x, y \rangle = \langle u, v \rangle))$
$\land v = y$
$\land \Box[b \Rightarrow (v' = y')]_{\langle b, v, y \rangle}$

BY $\langle 4 \rangle 17$

$\langle 4 \rangle 19.$ $\lor \neg H$
$\lor \exists u, v :$
$\land R(u, v)$
$\land SamePrefix(b, u, v, x, y)$
$\land PlusHalf(b, v, y)$

BY $\langle 4 \rangle 18$ DEF R, $SamePrefix$, $PlusHalf$

$\langle 4 \rangle$ QED
BY $\langle 4 \rangle 19$ DEF H

$\langle 3 \rangle$ QED
BY $\langle 3 \rangle 3$, $\langle 3 \rangle 4$ goal from $\langle 3 \rangle 2$

$\langle 2 \rangle 5.$ $(\forall b : (Fr(ClA, b) \land MustUnstep(b)) \Rightarrow FPH(ClG, b))$
$\equiv \forall b :$
$\lor \neg MustUnstep(b)$
$\lor \lor \neg \lor \neg IeP(x, y)$
$\lor \land JeP(x, y)$
$\land \Box[b' \Rightarrow NeP(x, y, x', y')]_{\langle x, y \rangle}$
$\lor \land IsP(x, y)$
$\land \Box[b' \Rightarrow NsP(x, y, x', y')]_{\langle x, y \rangle}$
$\land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$

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

At this point we transition to raw TLA+ with past operators.

⟨2⟩6. ASSUME

$$
\begin{array}{l}
sigma \models \forall b : \\
\quad \lor \neg MustUnstep(b) \\
\quad \lor \neg \lor \neg Ie \\
\qquad\quad \lor \land Je \\
\qquad\qquad \land \Box[b' \Rightarrow Ne]_{\langle x,\, y \rangle} \\
\quad \lor \land Is \\
\qquad \land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle} \\
\qquad \land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y
\end{array}
$$

PROVE

$$
\begin{array}{l}
sigma, 0 \models \\
\quad \lor \neg \lor \neg Ie \\
\qquad\quad \lor Je \\
\quad \lor \land Is \\
\qquad \land Ie \lor \Box(Next \land SysNext) \\
\qquad \land Ie \Rightarrow \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \land SysNext)
\end{array}
$$

⟨3⟩ USE DEF Ie, Je, Is, Ns, Ne

⟨3⟩1. $sigma, 0 \models \forall b :$

$$
\begin{array}{l}
\quad \lor \neg MustUnstep(b) \\
\quad \lor \neg \lor \neg Ie \\
\qquad\quad \lor \land Je \\
\qquad\qquad \land \Box[b' \Rightarrow Ne]_{\langle x,\, y \rangle} \\
\quad \lor \land Is \\
\qquad \land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle} \\
\qquad \land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y
\end{array}
$$

BY ⟨2⟩6

⟨3⟩2. SUFFICES

ASSUME $sigma, 0 \models Ie \Rightarrow Je$

PROVE

$$
\begin{array}{l}
\land sigma, 0 \models Is \\
\land \lor sigma, 0 \models Ie \\
\quad \lor sigma, 0 \models \Box(Next \land SysNext) \\
\\
\land \lor \neg sigma, 0 \models Ie \\
\quad \lor \forall i \in Nat : \\
\qquad\quad sigma, i \models \\
\qquad\qquad Earlier(EnvNext) \Rightarrow \land Earlier(Next) \\
\qquad\qquad\qquad\qquad\qquad\qquad\quad \land SysNext
\end{array}
$$

OBVIOUS

⟨3⟩3. $\forall tau :$

$\lor \neg IsABehavior(tau)$

$\lor \neg RefinesUpToVar(tau, sigma, \text{"b"})$

$\lor tau, 0 \models$ with the declaration VARIABLE b

$\lor \neg MustUnstep(b)$
$\lor \neg \lor \neg Ie$
$\quad \lor \land Je$
$\quad\quad \land \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$
$\lor \land Is$
$\quad \land \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
$\quad \land \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$
BY $\langle 3 \rangle 2$ DEF $\forall$ in raw TLA+

The following steps ($\langle 3 \rangle 4$, $\langle 3 \rangle 5$, $\langle 3 \rangle 6$) use sampling sequences, defined as tau, in order to draw the target conclusions.

$\langle 3 \rangle 4.$ $sigma, 0 \models Is$
$\langle 4 \rangle$ DEFINE $tau \triangleq$
LET $state(n) \triangleq [sigma[n]$ EXCEPT $![\text{"b"}] = (n = 0)]$
IN $[n \in Nat \mapsto state(n)]$
$\langle 4 \rangle 1.$ $\land IsABehavior(tau)$
$\land RefinesUpToVar(tau, sigma, \text{"b"})$
BY DEF tau, $IsABehavior$, $RefinesUpToVar$, Sim, $Natural$, $EqualUpToVar$
$\langle 4 \rangle 2.$ $tau, 0 \models MustUnstep(b)$
BY DEF tau, $MustUnstep$, $Unstep$, $MayUnstep$
$\langle 4 \rangle 3.$ $tau, 0 \models \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
BY DEF tau
$\langle 4 \rangle 4.$ $tau, 0 \models (Ie \Rightarrow Je) \Rightarrow Is$
BY $\langle 3 \rangle 3$, $\langle 4 \rangle 1$, $\langle 4 \rangle 2$, $\langle 4 \rangle 3$
$\langle 4 \rangle 5.$ $sigma, 0 \models (Ie \Rightarrow Je) \Rightarrow Is$
BY $\langle 4 \rangle 4$ DEF tau IeP, JeP, IsP are CONSTANTS
$\langle 4 \rangle$ QED
BY $\langle 4 \rangle 5$, $\langle 3 \rangle 2$ DEF tau

$\langle 3 \rangle 6.$ $\lor sigma, 0 \models Ie$
$\lor sigma, 0 \models \Box(Next \land SysNext)$
$\langle 4 \rangle 1.$ SUFFICES
ASSUME NEW $i \in Nat$, $sigma, 0 \models \neg Ie$
PROVE $sigma, i \models Next \land SysNext$
OBVIOUS
$\langle 4 \rangle 2.$ $Next \Rightarrow SysNext$
BY DEF $Next$, $SysNext$
$\langle 4 \rangle 3.$ SUFFICES $sigma, i \models Next$
BY $\langle 4 \rangle 2$ goal from $\langle 4 \rangle 1$
$\langle 4 \rangle 4.$ DEFINE $tau \triangleq$
LET $state(n) \triangleq [$
$sigma[n]$ EXCEPT $![\text{"b"}] = (n \leq i + 2)]$
IN $[n \in Nat \mapsto state(n)]$
$\langle 4 \rangle 5.$ $\land IsABehavior(tau)$

$\wedge$ *RefinesUpToVar*(*tau*, *sigma*, "b")
BY DEF *tau*, *IsABehavior*, *RefinesUpToVar*,
Sim, *Natura*, *EqualUpToVar*

$\langle 4 \rangle 6.$ $tau, 0 \models MustUnstep(b)$
BY DEF *tau*, *MustUnstep*, *Unstep*, *MayUnstep*

$\langle 4 \rangle 7.$ $tau, 0 \models \neg Ie$
BY $\langle 4 \rangle 1$ DEF *tau*, *Ie* *IeP* is independent of b.

$\langle 4 \rangle 8.$ $tau, 0 \models \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
BY $\langle 4 \rangle 5$, $\langle 4 \rangle 6$, $\langle 4 \rangle 7$, $\langle 3 \rangle 3$

$\langle 4 \rangle 9.$ $tau, (i + 1) \models b$
BY DEF *tau*

$\langle 4 \rangle 10.$ $tau, i \models b'$
BY $\langle 4 \rangle 9$

$\langle 4 \rangle 11.$ $tau, i \models b' \wedge [b' \Rightarrow Ns]_{\langle x, y \rangle}$
BY $\langle 4 \rangle 10$, $\langle 4 \rangle 8$

$\langle 4 \rangle 12.$ $tau, i \models [Ns]_{\langle x, y \rangle}$
BY $\langle 4 \rangle 11$

$\langle 4 \rangle 13.$ $tau, i \models Next$
BY $\langle 4 \rangle 12$ DEF *Next*

$\langle 4 \rangle$ QED
BY $\langle 4 \rangle 13$ DEF *tau*, *Next* Next is independent of b.

$\langle 3 \rangle 5.$ ASSUME
NEW $i \in Nat$,
$sigma, 0 \models Ie$
PROVE
$sigma, i \models Earlier(EnvNext) \Rightarrow \wedge Earlier(Next)$
$\wedge SysNext$

$\langle 4 \rangle$ DEFINE $tau \triangleq$
LET $state(n) \triangleq [sigma[n]$ EXCEPT $!["b"] = (n \leq i)]$
IN $[n \in Nat \mapsto state(n)]$

$\langle 4 \rangle 1.$ $\wedge$ *IsABehavior*(*tau*)
$\wedge$ *RefinesUpToVar*(*tau*, *sigma*, "b")
BY DEF *tau*, *IsABehavior*, *RefinesUpToVar*,
Sim, *Natural*, *EqualUpToVar*

$\langle 4 \rangle 2.$ $tau, 0 \models MustUnstep(b)$
BY DEF *tau*, *MustUnstep*, *Unstep*, *MayUnstep*

$\langle 4 \rangle 3.$ $tau, 0 \models$
$\vee \neg \vee \neg Ie$
$\vee \wedge Je$
$\wedge \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$
$\vee \wedge \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
$\wedge \Box[b \Rightarrow \exists r : NsP(x, y, r, y')]_y$
BY $\langle 4 \rangle 1$, $\langle 4 \rangle 2$, $\langle 3 \rangle 3$

$\langle 4 \rangle 4.$ SUFFICES ASSUME $sigma, i \models Earlier(EnvNext)$

PROVE $sigma, i \models \land Earlier(Next)$
$\qquad\qquad\qquad\qquad \land SysNext$

OBVIOUS

$\langle 4 \rangle 5.\ \forall k \in 0 \mathrel{..} (i-1) :$
$\qquad \langle sigma[k], sigma[k+1] \rangle [[EnvNext]]$

BY $\langle 4 \rangle 4$ DEF $Earlier$

$\langle 4 \rangle 6.$ CASE $i = 0$

$\langle 5 \rangle 1.\ tau, 0 \models \land b = \text{TRUE}$
$\qquad\qquad\qquad \land \Box(b' = \text{FALSE})$

$\langle 6 \rangle 1.\ \land tau[0][\text{“b”}] = \text{TRUE}$
$\qquad \land \forall j \in Nat \setminus \{0\} :\ tau[j][\text{“b”}] = \text{FALSE}$

BY DEF tau, $\langle 4 \rangle 6$

$\langle 6 \rangle$ QED

BY $\langle 6 \rangle 1$

$\langle 5 \rangle 2.\ tau, 0 \models \land Je$
$\qquad\qquad\qquad \land \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

$\langle 6 \rangle 1.\ tau, 0 \models Je$

$\langle 7 \rangle 1.\ tau, 0 \models Ie$

BY $\langle 3 \rangle 5$ DEF tau

IeP does not depend on b.

$\langle 7 \rangle 2.\ tau, 0 \models Ie \Rightarrow Je$

BY $\langle 3 \rangle 2$ DEF tau

$\langle 7 \rangle$ QED

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

$\langle 6 \rangle 2.\ tau, 0 \models \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

BY $\langle 5 \rangle 1$

$\langle 6 \rangle$ QED

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

$\langle 5 \rangle 3.\ tau, 0 \models$
$\qquad \land \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
$\qquad \land \Box[b \Rightarrow \exists r :\ NsP(x, y, r, y')]_y$

BY $\langle 4 \rangle 3$, $\langle 5 \rangle 2$

$\langle 5 \rangle 4.\ tau, i \models SysNext \land Earlier(Next)$

$\langle 6 \rangle 1.\ tau, 0 \models [b \Rightarrow \exists r :\ NsP(x, y, r, y')]_y$

BY $\langle 5 \rangle 3$

$\langle 6 \rangle 2.\ tau, 0 \models [\exists r :\ NsP(x, y, r, y')]_y$

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

$\langle 6 \rangle 3.\ tau, 0 \models SysNext$

BY $\langle 6 \rangle 2$ DEF $SysNext$

$\langle 6 \rangle 4.\ tau, 0 \models Earlier(Next)$

BY DEF $Earlier$

$\langle 6 \rangle 5.\ tau, 0 \models SysNext \land Earlier(Next)$

BY $\langle 6 \rangle 3$, $\langle 6 \rangle 4$

$\langle 6 \rangle$ QED

BY $\langle 6 \rangle 5$, $\langle 4 \rangle 6$

$\langle 5 \rangle$ QED   goal from $\langle 4 \rangle 4$
BY $\langle 5 \rangle 4$ DEF *tau*, *SysNext*, *Earlier*, *Next*

because variable b does not occur in the formula
$SysNext \land Earlier(Next)$
b is declared as VARIABLE b in $\langle 3 \rangle 3$

$\langle 4 \rangle 7.$CASE $i > 0$
$\langle 5 \rangle 1. \land \forall j \in 0 \,..\, i : \; tau[j][\text{“b”}] = \text{TRUE}$
$\land \forall j \in Nat : \; (j > i) \Rightarrow (tau[j][\text{“b”}] = \text{FALSE})$
BY DEF *tau*
$\langle 5 \rangle 2. \; tau, i \models Earlier(EnvNext)$
BY $\langle 4 \rangle 4$ DEF *tau*, *Earlier*, *EnvNext*, *Ne*
$\langle 5 \rangle 3. \; \forall k \in 0 \,..\, (i-1) :$
$\langle tau[k], tau[k+1] \rangle [[EnvNext]]$
$\langle 6 \rangle 1. \land (i-1) \in Nat$
$\land (i-1) \geq 0$
$\land (i-1) < i$
BY $\langle 3 \rangle 5$, $\langle 4 \rangle 7$
$\langle 6 \rangle$ QED
BY $\langle 5 \rangle 2$, $\langle 6 \rangle 1$ DEF *Earlier*
$\langle 5 \rangle 4. \; tau, 0 \models \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$
$\langle 6 \rangle 1. \; \forall k \in 0 \,..\, (i-1) :$
$\langle tau[k], tau[k+1] \rangle [[$
$[b' \Rightarrow Ne]_{\langle x, y \rangle}]]$
BY $\langle 5 \rangle 3$ DEF *EnvNext*
$\langle 6 \rangle 2. \; \forall k \in Nat : \; (k \geq i) \Rightarrow$
$\langle tau[k], tau[k+1] \rangle [[$
$[b' \Rightarrow Ne]_{\langle x, y \rangle}]]$
$\langle 7 \rangle 1. \; \forall k \in Nat : \; (k > i) \Rightarrow$
$\langle tau[k], tau[k+1] \rangle [[\neg b]]$
BY $\langle 5 \rangle 1$
$\langle 7 \rangle 2. \; \forall k \in Nat : \; (k \geq i) \Rightarrow$
$\langle tau[k], tau[k+1] \rangle [[\neg b']]$
BY $\langle 7 \rangle 1$ $b' = \text{FALSE}$ at these steps.
$\langle 7 \rangle$ QED
BY $\langle 7 \rangle 2$
$\langle 7 \rangle 3. \; \forall k \in Nat :$
$\langle tau[k], tau[k+1] \rangle [[$
$[b' \Rightarrow Ne]_{\langle x, y \rangle}]]$
BY $\langle 6 \rangle 1$, $\langle 6 \rangle 2$
$\langle 7 \rangle$ QED
BY $\langle 7 \rangle 3$
$\langle 5 \rangle 5. \; tau, 0 \models \; \land \Box[b' \Rightarrow Ns]_{\langle x, y \rangle}$
$\land \Box[b \Rightarrow \exists r : \; NsP(x, y, r, y')]_y$
$\langle 6 \rangle 1. \; tau, 0 \models \; \land Je$
$\land \Box[b' \Rightarrow Ne]_{\langle x, y \rangle}$

$\langle 7\rangle 1.\ tau,\ 0 \models Je$

$\langle 8\rangle 1.\ tau,\ 0 \models Ie$

BY $\langle 3\rangle 5$ DEF tau

IeP does not depend on b.

$\langle 8\rangle 2.\ tau,\ 0 \models Ie \Rightarrow Je$

BY $\langle 3\rangle 2$ DEF tau

$\langle 8\rangle$ QED

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

$\langle 7\rangle 2.\ tau,\ 0 \models \Box[b' \Rightarrow Ne]_{\langle x,\, y\rangle}$

BY $\langle 5\rangle 4$

$\langle 7\rangle$ QED

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

$\langle 6\rangle$ QED

BY $\langle 4\rangle 3$, $\langle 6\rangle 1$ DEF tau

$\langle 5\rangle 6.\ tau,\ i \models [\exists\, r :\ NsP(x,\, y,\, r,\, y')]_y$

$\langle 6\rangle 1.\ tau,\ i \models b$

BY $\langle 5\rangle 1$

$\langle 6\rangle 2.\ tau,\ 0 \models \Box[b \Rightarrow \exists\, r :\ NsP(x,\, y,\, r,\, y')]_y$

BY $\langle 5\rangle 5$

$\langle 6\rangle 3.\ tau,\ i \models [b \Rightarrow \exists\, r :\ NsP(x,\, y,\, r,\, y')]_y$

BY $\langle 6\rangle 2$

$\langle 6\rangle$ QED

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

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

$\langle 5\rangle 7.\ tau,\ i \models Earlier([Ns]_{\langle x,\, y\rangle})$

$\langle 6\rangle 1.\ \forall\, k \in Nat :\ tau,\ k \models [b' \Rightarrow Ns]_{\langle x,\, y\rangle}$

$\langle 7\rangle 1.\ tau,\ 0 \models \Box[b' \Rightarrow Ns]_{\langle x,\, y\rangle}$

BY $\langle 5\rangle 5$

$\langle 7\rangle$ QED

BY $\langle 7\rangle 1$

$\langle 6\rangle 2.\ \forall\, k \in 0 \,..\, i :\ tau,\ k \models b$

BY $\langle 5\rangle 1$

$\langle 6\rangle 3.\ \forall\, k \in 0 \,..\, (i-1) :\ tau,\ k \models b'$

$\langle 7\rangle 1.\ \land\ (i-1) \in Nat$
$\quad \land\ (i-1) \geq 0$
$\quad \land\ (i-1) < i$

BY $\langle 3\rangle 5$, $\langle 4\rangle 7$

$\langle 7\rangle$ QED

BY $\langle 6\rangle 2$, $\langle 7\rangle 1$

$\langle 6\rangle 4.\ \forall\, k \in 0 \,..\, (i-1) :$
$\quad tau,\ k \models b' \land [b' \Rightarrow Ns]_{\langle x,\, y\rangle}$

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

$\langle 6\rangle 5.\ \forall\, k \in 0 \,..\, (i-1) :$
$\quad tau,\ k \models [Ns]_{\langle x,\, y\rangle}$

BY $\langle 6\rangle 4$

$\langle 6\rangle$ QED

BY $\langle 6\rangle 5$ DEF $\mathit{Earlier}$

BY $\langle 5\rangle 1$

$\langle 5\rangle 8.$ $\mathit{tau},\ i \models \mathit{SysNext} \land \mathit{Earlier}(\mathit{Next})$

BY $\langle 5\rangle 6$, $\langle 5\rangle 7$ DEF $\mathit{SysNext}$, Next

$\langle 5\rangle$ QED goal from $\langle 4\rangle 4$

BY $\langle 5\rangle 8$ DEF tau, $\mathit{SysNext}$, $\mathit{Earlier}$, Next

$\langle 4\rangle$ QED

$\langle 5\rangle 1.$ $i \in \mathit{Nat}$

BY $\langle 3\rangle 5$

$\langle 5\rangle$ QED goal from $\langle 4\rangle 4$

BY $\langle 4\rangle 6$, $\langle 4\rangle 7$

$\langle 3\rangle$ QED

BY $\langle 3\rangle 4$, $\langle 3\rangle 5$, $\langle 3\rangle 6$ goal from $\langle 3\rangle 2$

$\langle 2\rangle 8.$ ASSUME

$\mathit{sigma},\ 0 \models$

$\lor \neg \lor \neg \mathit{Ie}$

$\quad \lor \mathit{Je}$

$\lor \land \mathit{Is}$

$\quad \land \mathit{Ie} \lor \Box(\mathit{Next} \land \mathit{SysNext})$

$\quad \land \mathit{Ie} \Rightarrow \Box(\mathit{Earlier}(\mathit{EnvNext}) \Rightarrow \land \mathit{Earlier}(\mathit{Next})$

$\qquad\qquad\qquad\qquad\qquad\qquad \land \mathit{SysNext})$

PROVE

$\mathit{sigma} \models \forall\, b :$

$\lor \neg \mathit{MustUnstep}(b)$

$\lor \neg \lor \neg \mathit{Ie}$

$\quad \lor \land \mathit{Je}$

$\qquad \land \Box[b' \Rightarrow \mathit{Ne}]_{\langle x,\, y\rangle}$

$\lor \land \mathit{Is}$

$\quad \land \Box[b' \Rightarrow \mathit{Ns}]_{\langle x,\, y\rangle}$

$\quad \land \Box[b \Rightarrow \exists\, r : \ \mathit{NsP}(x,\ y,\ r,\ y')]_{y}$

$\langle 3\rangle$ USE DEF Ie, Je, Is, Ns, Ne

$\langle 3\rangle 1.$ SUFFICES

ASSUME

NEW tau,

$\land \mathit{IsABehavior}(\mathit{tau})$

$\land \mathit{RefinesUpToVar}(\mathit{tau},\ \mathit{sigma},\ \text{“b”})$,

VARIABLE b

PROVE

$\mathit{tau},\ 0 \models$

$\lor \neg \mathit{MustUnstep}(b)$

$\lor \neg \lor \neg \mathit{Ie}$

$\quad \lor \land \mathit{Je}$

$\qquad \land \Box[b' \Rightarrow \mathit{Ne}]_{\langle x,\, y\rangle}$

$\lor \land Is$
$\land \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle}$
$\land \Box[b \Rightarrow \exists\, r : \; NsP(x,\, y,\, r,\, y')]_y$

BY DEF ∀

$\langle 3 \rangle 2.$ SUFFICES

ASSUME

$tau,\, 0 \models$
$\land\, MustUnstep(b)$
$\land\, Ie \Rightarrow \land\, Je$
$\qquad \land\, \Box[b' \Rightarrow Ne]_{\langle x,\, y \rangle}$

PROVE

$tau,\, 0 \models$
$\land\, Is$
$\land\, \Box[b' \Rightarrow Ns]_{\langle x,\, y \rangle}$
$\land\, \Box[b \Rightarrow \exists\, r : \; NsP(x,\, y,\, r,\, y')]_y$

OBVIOUS goal from $\langle 3 \rangle 1$

$\langle 3 \rangle$ DEFINE

$F \triangleq \lor \neg \lor \neg Ie$
$\qquad \lor Je$
$\quad \lor \land Is$
$\quad\; \land Ie \lor \Box(Next \land SysNext)$
$\quad\; \land Ie \Rightarrow \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next)$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext)$

$\langle 3 \rangle 3.$ $tau,\, 0 \models F$

$\langle 4 \rangle 1.$ $\forall\, rho :$
$\lor \neg IsABehavior(rho)$
$\lor \neg Sim(rho,\, sigma)$
$\lor\, rho,\, 0 \models F$

BY $\langle 2 \rangle 8$ DEF F Even though F is not a TLA+ formula, it is stutter-invariant.

$\langle 4 \rangle 2.$ $\forall\, rho,\, eta :$
$\lor \neg IsABehavior(eta)$
$\lor \neg IsABehavior(rho)$
$\lor \neg EqualUpToVar(rho,\, eta,\, \text{“b”})$
$\lor (eta,\, 0 \models F) \;\equiv\; (rho,\, 0 \models F)$

BY DEF F The variable b does not occur in F.

$\langle 4 \rangle 3.$ $\land\, IsABehavior(tau)$
$\land\, \exists\, rho : \; \land\, IsABehavior(rho)$
$\qquad\qquad \land\, Sim(rho,\, sigma)$
$\qquad\qquad \land\, EqualUpToVar(rho,\, tau,\, \text{“b”})$

BY $\langle 3 \rangle 1$ DEF $RefinesUpToVar$

$\langle 4 \rangle$ QED

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

$\langle 3 \rangle 4.$ CASE $tau,\, 0 \models \neg Ie$

⟨4⟩1. $tau, 0 \models$
$\land\ Is$
$\land\ \Box(Next \land SysNext)$
BY ⟨3⟩3, ⟨3⟩4 DEF F
⟨4⟩2. $tau, 0 \models$
$\land\ Is$
$\land\ \Box[Ns]_{\langle x,\, y\rangle}$
$\land\ \Box[\exists\, r : \ NsP(x, y, r, y')]_y$
BY ⟨4⟩1 DEF $Next$, $SysNext$
⟨4⟩ QED goal from ⟨3⟩2
BY ⟨4⟩2

⟨3⟩5. CASE $tau, 0 \models Ie$
⟨4⟩1. $tau, 0 \models\ \land\ Je$
$\land\ \Box[b' \Rightarrow Ne]_{\langle x,\, y\rangle}$
BY ⟨3⟩2, ⟨3⟩5
⟨4⟩2. $tau, 0 \models$
$\land\ Is$
$\land\ \Box(Earlier(EnvNext) \Rightarrow\ \land\ Earlier(Next)$
$\land\ SysNext)$
⟨5⟩1. $tau, 0 \models Ie \Rightarrow Je$
BY ⟨3⟩5, ⟨4⟩1
⟨5⟩ QED
BY ⟨3⟩3, ⟨5⟩1, ⟨3⟩5 DEF F
⟨4⟩3. SUFFICES $tau, 0 \models$
$\land\ \ \Box[b' \Rightarrow Ns]_{\langle x,\, y\rangle}$
$\land\ \ \Box[b \Rightarrow \exists\, r : \ NsP(x, y, r, y')]_y$
BY ⟨4⟩2 goal from ⟨3⟩2
⟨4⟩4. PICK $i \in Nat \setminus \{0\}$:
$\land\ \ \forall\, n \in 0 \,..\, i : \ tau, n \models b = \text{TRUE}$
$\land\ \ \forall\, n \in Nat : \ (n > i) \Rightarrow (tau, n \models b = \text{FALSE})$
⟨5⟩1. $tau, 0 \models MustUnstep(b)$
BY ⟨3⟩2
⟨5⟩ QED
BY ⟨5⟩1 DEF $MustUnstep$, $Unstep$, $MayUnstep$, tau

⟨4⟩5. $tau, i \models SysNext \land Earlier(Next)$
⟨5⟩1. $\forall\, n \in Nat : \ tau, n \models [b' \Rightarrow Ne]_{\langle x,\, y\rangle}$
BY ⟨4⟩1
⟨5⟩2. $\forall\, n \in 0 \,..\, (i-1) : \ tau, n \models \ b'$
BY ⟨4⟩4
⟨5⟩3. $\forall\, n \in 0 \,..\, (i-1)$:
$tau, n \models b' \land [b' \Rightarrow Ne]_{\langle x,\, y\rangle}$
BY ⟨5⟩2, ⟨5⟩1
⟨5⟩4. $\forall\, n \in 0 \,..\, (i-1)$:

$tau,\ n \models [Ne]_{\langle x,\, y\rangle}$
BY $\langle 5\rangle 3$
$\langle 5\rangle 5.$ $tau,\ i \models Earlier([Ne]_{\langle x,\, y\rangle})$
BY $\langle 5\rangle 4$ DEF *Earlier*
$\langle 5\rangle$ QED
BY $\langle 5\rangle 5$, $\langle 4\rangle 2$

$\langle 4\rangle 6.$ $tau,\ 0 \models \Box[b' \Rightarrow Ns]_{\langle x,\, y\rangle}$
$\langle 5\rangle 1.$ $\forall\, n \in 0 \,..\, (i-1) :$
$tau,\ n \models [b' \Rightarrow Ns]_{\langle x,\, y\rangle}$
$\langle 6\rangle 1.$ $\forall\, n \in 0 \,..\, (i-1) :$
$tau,\ n \models [Ns]_{\langle x,\, y\rangle}$
BY $\langle 4\rangle 5$ DEF *Next*
$\langle 6\rangle$ QED
BY $\langle 6\rangle 1$
$\langle 5\rangle 2.$ $\forall\, n \in Nat :\ (n \geq i)$
$\Rightarrow (tau,\ n \models [b' \Rightarrow Ns]_{\langle x,\, y\rangle})$
$\langle 6\rangle 1.$ $\forall\, n \ \in Nat :$
$(n > i) \Rightarrow (tau,\ n \models b = \text{FALSE})$
BY $\langle 4\rangle 4$
$\langle 6\rangle 2.$ $\forall\, n \ \in Nat :$
$(n \geq i) \Rightarrow (tau,\ n \models b' = \text{FALSE})$
BY $\langle 6\rangle 1$
BY $\langle 6\rangle 2$
$\langle 5\rangle 3.$ $\forall\, n \in Nat :\ tau,\ n \models [b' \Rightarrow Ns]_{\langle x,\, y\rangle}$
BY $\langle 5\rangle 1$, $\langle 5\rangle 2$
$\langle 5\rangle$ QED
BY $\langle 5\rangle 3$ DEF $\Box$
$\langle 4\rangle 7.$ $tau,\ 0 \models \Box[b \Rightarrow \exists\, r :\ Ns(x,\ y,\ r,\ y')]_y$
$\langle 5\rangle 1.$ $\forall\, n \in 0 \,..\, i :$
$tau,\ n \models [b \Rightarrow \exists\, r :\ NsP(x,\ y,\ r,\ y')]_y$
$\langle 6\rangle 1.$ $\forall\, n \in 0 \,..\, (i-1) :$
$tau,\ n \models \exists\, r :\ \lor NsP(x,\ y,\ r,\ y')$
$\lor (x = r)\ \land\ (y = y')$
BY $\langle 4\rangle 13$
$\langle 6\rangle 2.$ $\forall\, n \in 0 \,..\, (i-1) :$
$tau,\ n \models [\exists\, r :\ NsP(x,\ y,\ r,\ y')]_y$
BY $\langle 6\rangle 1$
$\langle 6\rangle 3.$ $\forall\, n \in 0 \,..\, i :$
$tau,\ n \models [\exists\, r :\ NsP(x,\ y,\ r,\ y')]_y$
$\langle 7\rangle 1.$ $tau,\ i \models [\exists\, r :\ NsP(x,\ y,\ r,\ y')]_y$
BY $\langle 4\rangle 5$ DEF *SysNext*
$\langle 7\rangle$ QED
BY $\langle 6\rangle 2$, $\langle 7\rangle 1$
$\langle 6\rangle$ QED

BY $\langle 6 \rangle 3$

$\langle 5 \rangle 2.$ $\forall\, n \in Nat : \; (n > i) \Rightarrow$
$tau,\, n \models [b \Rightarrow \exists\, r : \; Ns(x,\, y,\, r,\, y')]_y$

$\langle 6 \rangle 1.$ $\forall\, n \; \in Nat :$
$(n > i) \Rightarrow (tau,\, n \models b = \text{FALSE})$

BY $\langle 4 \rangle 4$

$\langle 6 \rangle$ QED

BY $\langle 6 \rangle 1$

$\langle 5 \rangle 3.$ $\forall\, n \in Nat :$
$tau,\, n \models [b \Rightarrow \exists\, r : \; Ns(x,\, y,\, r,\, y')]_y$

BY $\langle 5 \rangle 1$, $\langle 5 \rangle 2$

$\langle 5 \rangle$ QED

BY $\langle 5 \rangle 3$ DEF $\Box$

$\langle 4 \rangle$ QED goal from $\langle 4 \rangle 3$

BY $\langle 4 \rangle 6$, $\langle 4 \rangle 7$

$\langle 3 \rangle$ QED goal from $\langle 3 \rangle 2$

BY $\langle 3 \rangle 4$, $\langle 3 \rangle 5$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 5$, $\langle 2 \rangle 6$, $\langle 2 \rangle 8$

$\langle 1 \rangle$ QED

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

The above theorem assumes that actions are defined using constants.
Essentially the same proof can be used when the environment action *Ne* is defined using constant operators and *Earlier* (in other words, when this is an action that results from converting *WPH* to raw TLA+ with past).

In that case the proof should be modified to address two points:

1. The operator A is declared (not defined) in TLA+ and assumed to be equivalent to a raw TLA+ formula. This assumption is:

   $(\; sigma \models C(A,\, x,\, y) \;) \; \equiv \; (\text{sigma},\, 0 \models$
   $Ie \Rightarrow \; \land\, Je$
   $\land\, \Box[Earlier(Ne) \Rightarrow (Earlier(N) \land Ns))$
   $)$

   For writing the operator *Earlier* we have to be within raw TLA+, which is why on the left-hand-side we have $sigma \models Cl(A,\, x,\, y)$, whereas on the right-hand-side we have sigma, $0 \models \ldots$.

2. The proof should be carried out mostly within raw TLA+ with past. In other words, we should "move" to raw TLA+ before the step that replaces the closure $Cl(A,\, x,\, y)$ with a specific formula.

   Again, the reason is that the closure is expressed using past temporal operators, so we cannot write it in this form within TLA+.

3. Combining the two previous points, closure and past operators need to coexist within the same logic. This requires expressing temporal quantification $\boldsymbol{\exists}$ in raw TLA+ with past (since past operators need an indexed satisfaction relation ($\models$), so they are not expressible in TLA+).

   This definition is given in the module *TemporalLogic*.

4. When we reach the step of substituting u, v with x, y in the environment action (and vice versa in the reverse direction of proof), we have to do this replacement also within *Earlier*. This replacement is justified by observing that if b is true at some state in a behavior, then it must have been true in all previous states. Thus, $\langle x, y\rangle = \langle u, v\rangle$ in all those previous states (similar argument to how *SamePrefix* is handled).

We could write the existential quantifier outside the box $[\ldots]_{-}y$, though that would be ungrammatical as an action after $\Box$.

PROPOSITION
ASSUME
VARIABLE x, VARIABLE y,
CONSTANT $Next(\_, \_, \_, \_)$
PROVE
LET

$\exists x' : [Next(x, y, x', y')]_{-}\langle x, y\rangle$

Applying rigid quantification to a primed variable is ungrammatical in TLA+.

$$A \triangleq \exists u : \lor Next(x, y, u, y') \lor \langle u, y'\rangle = \langle x, y\rangle$$
$$B \triangleq \exists u : [Next(x, y, u, y')]_y$$
$$C \triangleq [\exists u : Next(x, y, u, y')]_y$$

IN
$\land A \equiv B$
$\land B \equiv C$

$\langle 1\rangle$ DEFINE
$$A \triangleq \exists u : \lor Next(x, y, u, y') \lor \langle u, y'\rangle = \langle x, y\rangle$$
$$B \triangleq \exists u : [Next(x, y, u, y')]_y$$
$$C \triangleq [\exists u : Next(x, y, u, y')]_y$$

$\langle 1\rangle 1.$ $A \equiv C$

$\langle 2\rangle 1.$ $(\exists u : \lor Next(x, y, u, y') \lor \langle u, y'\rangle = \langle x, y\rangle)$
$\equiv$
$\lor \exists u : Next(x, y, u, y')$
$\lor \exists u : \langle u, y'\rangle = \langle x, y\rangle$

OBVIOUS

$\langle 2\rangle 2.$ $(\exists u : \langle u, y'\rangle = \langle x, y\rangle)$
$\equiv \land \exists u : u = x$
$\land y' = y$

OBVIOUS

⟨2⟩3. $(\exists u : \lor Next(x, y, u, y') \lor \langle u, y' \rangle = \langle x, y \rangle) \equiv \lor \exists u : Next(x, y, u, y') \lor y' = y$

BY ⟨2⟩1, ⟨2⟩2

⟨2⟩ QED

BY ⟨2⟩3 DEF A, C

⟨1⟩2. $B \equiv C$

⟨2⟩1. $(\exists u : [Next(x, y, u, y')]_y) \equiv \exists u : \lor Next(x, y, u, y') \lor y' = y$

OBVIOUS

⟨2⟩2. $B \equiv \lor \exists u : Next(x, y, u, y') \lor y' = y$

BY ⟨2⟩1 DEF B

⟨2⟩ QED

BY ⟨2⟩2 DEF C

⟨1⟩ QED

BY ⟨1⟩1, ⟨1⟩2

Expressing *Unzip* in raw TLA+ with past.

We apply the raw form of *WhilePlusHalf* twice, recursively. The form was proved for constant actions, but as noted above the proof can be modified for the case of an environment action that contains past temporal operators.

THEOREM

ASSUME

VARIABLE x, VARIABLE y,

CONSTANT $I(\_, \_)$,

CONSTANT $N(\_, \_)$,

TEMPORAL $L(\_, \_)$

PROVE

LET

$P(u, v) \triangleq \land I(u, v) \land L(u, v) \land \Box[N(u, v, u', v')]_{\langle u, v \rangle}$

$EnvNext \triangleq [\exists r : N(x, y, x', r)]_x$

$SysNext \triangleq [\exists r : N(x, y, r, y')]_y$

$Next \triangleq [N(x, y, x', y')]_{\langle x, y \rangle}$

$Raw \triangleq$

$\land \exists p : I(p, y)$

$\land \lor \neg\exists q : I(x, q)$

$\lor \land I(x, y)$

$\land \Box \lor \neg \mathit{Earlier}(\mathit{EnvNext})$
$\quad \lor \mathit{SysNext} \land \mathit{Earlier}(\mathit{Next})$
$\land (\Box \mathit{EnvNext}) \;\Rightarrow\; L(x, y)$

IN

$\mathit{Unzip}(P, x, y) \equiv \mathit{Raw}$

PROOF

$\langle 1 \rangle$ DEFINE

$P(u, v) \;\triangleq\; \land I(u, v) \land L(u, v)$
$\qquad\qquad \land \Box[N(u, v, u', v')]_{\langle u, v \rangle}$
$Q(u, v) \;\triangleq\; P(v, u)$

$\langle 1 \rangle 1.$ $\mathit{Unzip}(P, x, y) \equiv$

LET

$A(u, v) \;\triangleq\; \mathit{WPH}(Q, Q, v, u)$

IN

$\mathit{WPH}(A, P, x, y)$

BY DEF Unzip

$\langle 1 \rangle 2.$ ASSUME VARIABLE u, VARIABLE v

PROVE $\mathit{WPH}(Q, Q, v, u) \equiv$

LET

$F \;\triangleq\; \exists p, q : \; I(p, q)$
$G \;\triangleq\; \exists q : \; I(u, q)$
$\mathit{Ie} \;\triangleq\; F \land (G \Rightarrow I(u, v))$
$\mathit{Je} \;\triangleq\; G$
$\mathit{Next} \;\triangleq\; [N(u, v, u', v')]_{\langle u, v \rangle}$
$\mathit{EnvNext} \;\triangleq\; [\exists r : \; N(u, v, u', r)]_u$

IN

$\lor \neg \mathit{Ie}$
$\lor \mathit{Je} \land \Box(\mathit{Earlier}(\mathit{Next}) \Rightarrow \mathit{EnvNext})$

$\langle 2 \rangle$ DEFINE

$F \;\triangleq\; \exists p, q : \; I(p, q)$
$G \;\triangleq\; \exists q : \; I(u, q)$
$\mathit{Ie} \;\triangleq\; F \land (G \Rightarrow I(u, v))$
$\mathit{Je} \;\triangleq\; G$
$\mathit{Next} \;\triangleq\; [N(u, v, u', v')]_{\langle u, v \rangle}$
$\mathit{EnvNext} \;\triangleq\; [\exists r : \; N(u, v, u', r)]_u$

$\langle 2 \rangle 1.$ $\mathit{WPH}(Q, Q, v, u) \equiv$

$\lor \neg \exists p, q : \; \text{TRUE} \Rightarrow I(p, q)$
$\lor \; \land \exists q : \; I(u, q)$
$\quad \land \; \lor \neg \lor \neg \text{TRUE}$
$\qquad\qquad \lor I(u, v)$
$\qquad \lor \; \land I(u, v)$
$\qquad\quad \land I(u, v) \lor \Box(\mathit{Next} \land \mathit{EnvNext})$
$\qquad\quad \land \; \lor \neg I(u, v)$
$\qquad\qquad \lor \Box(\mathit{Earlier}(\mathit{Next}) \Rightarrow \; \land \mathit{Earlier}(\mathit{Next})$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land \mathit{EnvNext})$

$\wedge\ \vee \neg \vee \neg$TRUE

$\quad\quad \vee\ I(u, v) \wedge L(u, v) \wedge \Box Next$

$\quad \vee\ L(u, v)$

BY *RawWhilePlusHalfFull* DEF Q

⟨2⟩2. $WPH(Q, Q, v, u) \equiv$

$\vee\ \neg\exists\, p, q : I(p, q)$

$\vee\ \wedge\ \exists\, q : I(u, q)$

$\quad \wedge\ \vee\ \neg I(u, v)$

$\quad\quad \vee\ \Box(Earlier(Next) \Rightarrow EnvNext)$

$\quad \wedge\ \vee\ \neg L(u, v)$

$\quad\quad \vee\ \neg\Box Next$

$\quad\quad \vee\ L(u, v)$

BY ⟨2⟩1

⟨2⟩3. $WPH(Q, Q, v, u) \equiv$

$\vee\ \neg\exists\, p, q : I(p, q)$

$\vee\ \wedge\ \exists\, q : I(u, q)$

$\quad \wedge\ \vee\ \neg I(u, v)$

$\quad\quad \vee\ \Box(Earlier(Next) \Rightarrow EnvNext)$

BY ⟨2⟩2

⟨2⟩4. $WPH(Q, Q, v, u) \equiv$

$\vee\ \neg F$

$\vee\ \wedge\ G$

$\quad \wedge\ \vee\ \neg I(u, v)$

$\quad\quad \vee\ \Box(Earlier(Next) \Rightarrow EnvNext)$

BY ⟨2⟩3 DEF F, G

⟨2⟩5. $WPH(Q, Q, v, u) \equiv$

$\vee\ \neg F$

$\vee\ G \wedge \neg I(u, v)$

$\vee\ G \wedge \Box(Earlier(Next) \Rightarrow EnvNext)$

BY ⟨2⟩4

⟨2⟩6. $WPH(Q, Q, v, u) \equiv$

$\vee\ \neg \wedge F$

$\quad\quad \wedge\ G \Rightarrow I(u, v)$

$\vee\ G \wedge \Box(Earlier(Next) \Rightarrow EnvNext)$

BY ⟨2⟩5

⟨2⟩ QED

BY ⟨2⟩6 DEF Ie, Je, F, G

⟨1⟩ DEFINE

$F \triangleq \exists\, p, q : I(p, q)$

$G \triangleq \exists\, q : I(x, q)$

$Ie \triangleq F \wedge (G \Rightarrow I(x, y))$

$Je \triangleq G$

These definitions differ from those in ⟨1⟩2 because they are in terms of x, y instead of u, v.

$\mathit{Next} \triangleq [N(x, y, x', y')]_{\langle x, y \rangle}$
$\mathit{EnvNext} \triangleq [\exists r : N(x, y, x', r)]_x$
$\mathit{SysNext} \triangleq [\exists r : N(x, y, r, y')]_y$

$\langle 1 \rangle 3.\ \mathit{Unzip}(P, x, y) \equiv$
$\quad \lor\ \neg \exists u, v :$
$\quad\quad \lor \neg \land \exists p, q : I(p, q)$
$\quad\quad\quad \land \lor \neg \exists q : I(u, q)$
$\quad\quad\quad\quad \lor I(u, v)$
$\quad\quad \lor \exists q : I(u, q)$
$\quad \lor\ \land \exists p : I(p, y)$
$\quad\quad \land \lor \neg \lor \neg \mathit{Ie}$
$\quad\quad\quad\quad \lor \mathit{Je}$
$\quad\quad\quad \lor \land I(x, y)$
$\quad\quad\quad\quad \land \mathit{Ie} \lor \Box(\mathit{Next})$
$\quad\quad\quad\quad \land \mathit{Ie} \Rightarrow \Box \lor \neg \mathit{Earlier}(\mathit{Earlier}(\mathit{Next}) \Rightarrow \mathit{EnvNext})$
$\quad\quad\quad\quad\quad\quad \lor \mathit{SysNext} \land \mathit{Earlier}(\mathit{Next})$
$\quad\quad\quad\quad \land \lor \neg \lor \neg \mathit{Ie}$
$\quad\quad\quad\quad\quad\quad \lor \mathit{Je} \land \Box(\mathit{Earlier}(\mathit{Next}) \Rightarrow \mathit{EnvNext})$
$\quad\quad\quad\quad\quad \lor L(x, y)$

BY DEF $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\mathit{WhilePlusHalfStepwiseForm}$

with the caveat about *WhilePlusHalfStepwiseForm* and past operators within the environment action that was noted ealier

$\langle 1 \rangle 11.\ \lor \neg I(x, y)$
$\quad \lor \mathit{Ie}$
$\langle 2 \rangle 1.\ I(x, y) \Rightarrow F$
$\langle 3 \rangle 1.\ I(x, y) \Rightarrow \exists p, q : I(p, q)$
OBVIOUS
$\langle 3 \rangle$ QED
BY $\langle 3 \rangle 1$ DEF F
$\langle 2 \rangle 2.\ I(x, y) \Rightarrow (G \Rightarrow I(x, y))$
OBVIOUS
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 1$, $\langle 2 \rangle 2$ DEF Ie

$\langle 1 \rangle 4.\ \exists u, v :$
$\quad \lor \neg \land \exists p, q : I(p, q)$
$\quad\quad \land \lor \neg \exists q : I(u, q)$
$\quad\quad\quad \lor I(u, v)$
$\quad \lor \exists q : I(u, q)$
$\langle 2 \rangle 1.\ (\exists u, v :$
$\quad\quad \lor \neg \land \exists p, q : I(p, q)$
$\quad\quad\quad \land \lor \neg \exists q : I(u, q)$
$\quad\quad\quad\quad \lor I(u, v)$
$\quad\quad \lor \exists q : I(u, q)$
$\quad) \equiv ($
$\quad \lor \exists u, v :$

$\neg \land \exists\, p,\, q : \; I(p,\, q)$
$\quad \land \; \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad \lor\, I(u,\, v)$
$\lor\, \exists\, u,\, v : \; \exists\, q : \; I(u,\, q)$
$)$

OBVIOUS

$\langle 2 \rangle 2.\ (\exists\, u,\, v : \; \exists\, q : \; I(u,\, q))$
$\quad \equiv \exists\, p,\, q : \; I(p,\, q)$

OBVIOUS

$\langle 2 \rangle 3.\ (\neg \land \exists\, p,\, q : \; I(p,\, q)$
$\quad \land \; \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad \lor\, I(u,\, v)$
$\quad) \equiv ($
$\qquad \lor \neg\exists\, p,\, q : \; I(p,\, q)$
$\qquad \lor \neg \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad\quad \lor\, I(u,\, v))$

OBVIOUS

$\langle 2 \rangle 4.\ (\exists\, u,\, v :$
$\quad \neg \land \exists\, p,\, q : \; I(p,\, q)$
$\qquad \land \; \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad\quad \lor\, I(u,\, v))$
$\quad \equiv ($
$\qquad \lor\, \exists\, u,\, v : \; \neg\exists\, p,\, q : \; I(p,\, q)$
$\qquad \lor\, \exists\, u,\, v : \; \land\, \exists\, q : \; I(u,\, q)$
$\qquad\qquad\qquad \land \neg I(u,\, v))$

BY $\langle 2 \rangle 3$

$\langle 2 \rangle 5.\ (\exists\, u,\, v :$
$\qquad \lor \neg \land \exists\, p,\, q : \; I(p,\, q)$
$\qquad\quad \land \; \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad\qquad \lor\, I(u,\, v)$
$\qquad \lor\, \exists\, q : \; I(u,\, q)$
$\quad) \equiv ($
$\quad \lor\, \exists\, u,\, v : \; \neg\exists\, p,\, q : \; I(p,\, q)$
$\quad \lor\, \exists\, u,\, v : \; \land\, \exists\, q : \; I(u,\, q)$
$\qquad\qquad\qquad \land \neg I(u,\, v)$
$\quad \lor\, \exists\, p,\, q : \; I(p,\, q)$
$\quad)$

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

$\langle 2 \rangle 6.\ (\exists\, u,\, v :$
$\qquad \lor \neg \land \exists\, p,\, q : \; I(p,\, q)$
$\qquad\quad \land \; \lor \neg\exists\, q : \; I(u,\, q)$
$\qquad\qquad \lor\, I(u,\, v)$
$\qquad \lor\, \exists\, q : \; I(u,\, q)$
$\quad) \equiv ($
$\quad \lor \neg\exists\, p,\, q : \; I(p,\, q)$

```
                ∨ ∃ p, q :  I(p, q)
                ∨ ∃ u, v :  ∧ ∃ q :  I(u, q)
                            ∧ ¬I(u, v)
                )
            BY ⟨2⟩5
        ⟨2⟩ QED
            BY ⟨2⟩6
⟨1⟩5. ∨ ¬F
      ∨ (Ie ⇒ Je)  ≡ G
    ⟨2⟩1. (Ie ⇒ Je)
             ≡ ((F ∧ (G ⇒ I(x, y))) ⇒  G)
        BY  DEF Ie, Je
    ⟨2⟩2. ∨ ¬F
          ∨ (Ie ⇒ Je)
               ≡ ((G ⇒ I(x, y)) ⇒ G)
        BY ⟨2⟩1
    ⟨2⟩3. G  ≡  ((G ⇒ I(x, y)) ⇒ G)
        ⟨3⟩1. ((G ⇒ I(x, y)) ⇒ G)
                ≡  ∨ ¬(G ⇒ I(x, y))
                   ∨ G
            OBVIOUS
        ⟨3⟩2. ((G ⇒ I(x, y)) ⇒ G)
                ≡  ∨ G ∧ ¬I(x, y)
                   ∨ G
            BY ⟨3⟩1
        ⟨3⟩ QED
            BY ⟨3⟩2
    ⟨2⟩ QED
        BY ⟨2⟩2, ⟨2⟩3
⟨1⟩6. Unzip(P, x, y) ≡
        ∧  ∃ p :  I(p, y)
        ∧   ∨ ¬G
            ∨ ∧ I(x, y)
              ∧ □ ∨ ¬Earlier(Earlier(Next) ⇒ EnvNext)
                  ∨ SysNext ∧ Earlier(Next)
              ∧ ∨ ¬(Je ∧ □(Earlier(Next) ⇒ EnvNext)
                ∨ L(x, y)
    BY ⟨1⟩3, ⟨1⟩4, ⟨1⟩5, ⟨1⟩11
⟨1⟩7. ASSUME NEW sigma, IsABehavior(sigma)
       PROVE
        (sigma, 0 ⊨
            □ ∨ ¬Earlier(Earlier(Next) ⇒ EnvNext)
              ∨ SysNext ∧ Earlier(Next))
         ≡
        sigma, 0 ⊨
```

$\Box \lor \neg Earlier(EnvNext)$
$\quad \lor SysNext \land Earlier(Next)$

⟨2⟩ DEFINE

$A \triangleq \lor \neg Earlier(\lor \neg Earlier(Next)$
$\qquad\qquad\qquad\qquad \lor EnvNext)$
$\qquad \lor SysNext \land Earlier(Next)$

$B \triangleq Earlier(EnvNext) \Rightarrow \land Earlier(Next)$
$\qquad\qquad\qquad\qquad\qquad \land SysNext$

⟨2⟩1. ASSUME $\forall n \in Nat : \; sigma, n \models A$
PROVE $\forall n \in Nat : \; sigma, n \models B$

⟨3⟩1. SUFFICES
ASSUME NEW $n \in Nat$,
PROVE $sigma, n \models B$
OBVIOUS

⟨3⟩2. SUFFICES
ASSUME $sigma, n \models Earlier(EnvNext)$
PROVE $sigma, n \models SysNext \land Earlier(Next)$
BY DEF B goal from ⟨3⟩1

⟨3⟩3. SUFFICES
$sigma, n \models Earlier(Earlier(Next) \Rightarrow EnvNext)$

⟨4⟩1. $sigma, n \models A$

⟨5⟩1. $\forall k \in Nat : \; sigma, k \models A$
BY ⟨2⟩1

⟨5⟩2. $n \in Nat$
BY ⟨3⟩1

⟨5⟩ QED
BY ⟨5⟩1, ⟨5⟩2

⟨4⟩2. $sigma, n \models \lor \neg Earlier(Earlier(Next) \Rightarrow EnvNext)$
$\qquad\qquad \lor SysNext \land Earlier(Next)$
BY ⟨4⟩1 DEF B

⟨4⟩ QED
BY ⟨3⟩3, ⟨4⟩2 goal from ⟨3⟩2

⟨3⟩4. $(sigma, n \models Earlier(EnvNext))$
$\Rightarrow sigma, n \models Earlier(EnvNext \lor \neg Earlier(Next))$

⟨4⟩1. $EnvNext \Rightarrow (EnvNext \lor \neg Earlier(Next))$
OBVIOUS

⟨4⟩2. $n \in Nat$
BY ⟨3⟩1

⟨4⟩3. $IsABehavior(sigma)$
BY ⟨1⟩99

⟨4⟩ QED
BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3 DEF $Earlier$

⟨3⟩ QED
BY ⟨3⟩2, ⟨3⟩4 goal from ⟨3⟩3

⟨2⟩2. ASSUME $\forall n \in Nat : \; sigma, n \models B$

PROVE $\forall n \in Nat : \; sigma, n \models A$

⟨3⟩1. SUFFICIES

ASSUME NEW $n \in Nat$

PROVE $sigma, n \models A$

OBVIOUS

⟨3⟩2. SUFFICES

ASSUME $sigma, n \models Earlier(Earlier(Next) \Rightarrow EnvNext)$

PROVE $sigma, n \models SysNext \land Earlier(Next)$

BY DEF A goal from ⟨3⟩1

⟨3⟩3. SUFFICES

$sigma, n \models Earlier(EnvNext)$

⟨4⟩1. $sigma, n \models B$

⟨5⟩1. $\forall k \in Nat : \; sigma, n \models B$

BY ⟨2⟩2

⟨5⟩2. $n \in Nat$

BY ⟨3⟩1

⟨5⟩ QED

BY ⟨5⟩1, ⟨5⟩2

⟨4⟩2. $sigma, n \models Earlier(EnvNext) \Rightarrow \; \land Earlier(Next)$
$\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext$

BY ⟨4⟩1 DEF B

⟨4⟩ QED

BY ⟨3⟩3, ⟨4⟩2 goal from ⟨3⟩2

⟨3⟩4. SUFFICES

ASSUME $sigma, n \models \neg Earlier(EnvNext)$

PROVE FALSE

OBVIOUS goal from ⟨3⟩3

⟨3⟩5. $sigma, 0 \models \neg Earlier(EnvNext)$

⟨3⟩ DEFINE

$P(m) \triangleq sigma, m \models \neg Earlier(EnvNext)$

prepare for downward induction

⟨4⟩1. $\forall k \in 1 \, .. \, n : \; P(k) \Rightarrow P(k-1)$

⟨5⟩1.CASE $n = 0$

OBVIOUS

⟨5⟩2. SUFFICES ASSUME $n > 0$

PROVE $\forall k \in 1 \, .. \, n : \; P(k) \Rightarrow P(k-1)$

⟨6⟩1. $n \in Nat$

BY ⟨3⟩1

⟨6⟩ QED

BY ⟨6⟩1, ⟨5⟩1, ⟨5⟩2

⟨5⟩3. SUFFICES

ASSUME NEW $k \in 1 \, .. \, n, P(k)$

PROVE $P(k-1)$

OBVIOUS goal from ⟨5⟩2

⟨5⟩4. $sigma, k \models \neg Earlier(EnvNext)$

$\text{BY}\ \langle 5\rangle 3\ \ \text{DEF}\ P$

$\langle 5\rangle 5.\ \text{PICK}\ j \in 0 \,..\, (k-1) :\ \ sigma, j \models \neg EnvNext$

$\langle 7\rangle 1.\ (k\ > 0) \land (k \in Nat)$

$\langle 8\rangle 1.\ k \in 1 \,..\, n$

$\text{BY}\ \langle 5\rangle 3$

$\langle 8\rangle 2.\ 1 \in 1 \,..\, n$ thus $1 \,..\, n \neq \{\}$

$\text{BY}\ \langle 3\rangle 1, \langle 5\rangle 2$

$\langle 8\rangle\ \text{QED}$

$\text{BY}\ \langle 8\rangle 1, \langle 8\rangle 2$

$\langle 7\rangle 2.\ \neg\forall\, r \in 0 \,..\, (k-1) :\ \ sigma, r \models EnvNext$

$\text{BY}\ \langle 5\rangle 4\ \ \text{DEF}\ Earlier$ the general DEF for past operators

$\langle 7\rangle 3.\ \exists\, r \in 0 \,..\, (k-1) :\ \ sigma, r \models \neg EnvNext$

$\text{BY}\ \langle 7\rangle 2$

$\langle 7\rangle 4.\ 0 \in 0 \,..\, (k-1)$ thus $0 \,..\, (k-1) \neq \{\}$

$\text{BY}\ \langle 7\rangle 1$

$\langle 7\rangle\ \text{QED}$

$\text{BY}\ \langle 7\rangle 3, \langle 7\rangle 4$

$\langle 5\rangle 6.\ j \in 0 \,..\, (n-1)$

$\langle 6\rangle 1.\ k \in 1 \,..\, n$

$\text{BY}\ \langle 5\rangle 3$

$\langle 6\rangle 2.\ j \in 0 \,..\, (k-1)$

$\text{BY}\ \langle 5\rangle 5$

$\langle 6\rangle\ \text{QED}$

$\text{BY}\ \langle 6\rangle 1, \langle 6\rangle 2$

$\langle 5\rangle 7.\ sigma, j \models Earlier(Next) \Rightarrow EnvNext$

$\langle 6\rangle 1.\ sigma, n \models Earlier(Earlier(Next) \Rightarrow EnvNext)$

$\text{BY}\ \langle 3\rangle 2$

$\langle 6\rangle 2.\ \forall\, r \in 0 \,..\, (n-1) :$

$sigma, r \models Earlier(Next) \Rightarrow EnvNext$

$\text{BY}\ \langle 6\rangle 1, \langle 3\rangle 1\ \ \text{DEF}\ Earlier$

$\langle 6\rangle\ \text{QED}$

$\text{BY}\ \langle 6\rangle 2, \langle 5\rangle 6$

$\langle 5\rangle 8.\ sigma, j \models \neg Earlier(Next)$

$\text{BY}\ \langle 5\rangle 5, \langle 5\rangle 7$

$\langle 5\rangle 9.\ sigma, (k-1) \models \neg Earlier(Next)$

$\text{BY}\ \langle 5\rangle 8, \langle 5\rangle 5, \langle 3\rangle 1\ \ \text{DEF}\ Earlier$

$\langle 5\rangle 10.\ sigma, (k-1) \models$

$\lor\ \neg Earlier(EnvNext)$

$\lor\ Earlier(Next) \land SysNext$

$\langle 6\rangle 1.\ (k-1) \in 0 \,..\, (n-1)$

$\text{BY}\ \langle 5\rangle 3$

$\langle 6\rangle 2.\ (k-1) \in Nat$

$\text{BY}\ \langle 6\rangle 1$

$\langle 6\rangle$ QED

BY $\langle 2\rangle 2$, $\langle 6\rangle 1$ DEF B $n \leftarrow (k-1)$

$\langle 5\rangle 11.$ $sigma, (k-1) \models \neg Earlier(EnvNext)$

BY $\langle 5\rangle 9$, $\langle 5\rangle 10$

$\langle 5\rangle$ QED

BY $\langle 5\rangle 11$ DEF P goal from $\langle 5\rangle 3$

$\langle 4\rangle 2.$ $P(0)$

BY $\langle 3\rangle 5$, *DownwardNatInduction*

see *NaturalsInduction*

$\langle 4\rangle$ QED

BY $\langle 4\rangle 2$ DEF P

$\langle 3\rangle 6.$ $sigma, 0 \models Earlier(EnvNext)$

BY DEF *Earlier*

$\langle 3\rangle$ QED

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

$\langle 2\rangle$ QED

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

$\langle 1\rangle 8.$ $Unzip(P, x, y) \equiv$

$\quad \land\ \exists p : I(p, y)$

$\quad \land\ \lor \neg G$

$\qquad \lor \land I(x, y)$

$\qquad\quad \land \Box \lor \neg Earlier(EnvNext)$

$\qquad\qquad \lor SysNext \land Earlier(Next)$

$\qquad\quad \land \lor \neg(Je \land \Box(Earlier(Next) \Rightarrow EnvNext)$

$\qquad\qquad \lor L(x, y)$

BY $\langle 1\rangle 6$, $\langle 1\rangle 7$

$\langle 1\rangle 9.$ $Unzip(P, x, y) \equiv$

$\quad \land\ \exists p : I(p, y)$

$\quad \land\ \lor \neg \exists q : I(x, q)$

$\qquad \lor \land I(x, y)$

$\qquad\quad \land \Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next)$

$\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext)$

$\qquad\quad \land \lor \neg\Box(Earlier(Next) \Rightarrow EnvNext)$

$\qquad\qquad \lor L(x, y)$

BY $\langle 1\rangle 8$ DEF G

$\langle 1\rangle 10.$ $\lor \neg\Box(Earlier(EnvNext) \Rightarrow \land Earlier(Next)$

$\qquad\qquad\qquad\qquad\qquad\qquad \land SysNext)$

$\quad \lor (\Box(Earlier(Next) \Rightarrow EnvNext))$

$\qquad \equiv \Box EnvNext$

OMITTED

$\langle 1\rangle$ QED

BY $\langle 1\rangle 9$, $\langle 1\rangle 10$

MODULE *UnzipTheorems*

Theorems related to the operator *Unzip*.

Obvious proofs below are so for *TLAPS* v1.4.3.

Author: Ioannis *Filippidis*

EXTENDS *WhilePlusHalfTheorems*

$ExistsUnique(P(\_)) \triangleq$
$\quad \land \exists u : P(u)$
$\quad \land \forall u, v : (P(u) \land P(v)) \Rightarrow (u = v)$

THEOREM $InessentialNoninterleaving \triangleq$
ASSUME
VARIABLE x, VARIABLE y, CONSTANT $Inv(\_, \_)$,
CONSTANT $SysTurn(\_, \_)$, CONSTANT $Next(\_, \_, \_, \_)$,
LET $SysNext(p, q, v) \triangleq \exists u : Next(p, q, u, v)$
$\quad EnvNext(p, q, u) \triangleq \exists v : Next(p, q, u, v)$
IN
$\land \lor \neg(SysTurn(x, y) \land Inv(x, y))$
$\quad \lor ExistsUnique($LAMBDA $r : EnvNext(x, y, r))$
$\land SysNext(x, y, y') \land EnvNext(x, y, x')$
PROVE
$\lor \neg \land SysTurn(x, y)$
$\quad \land Inv(x, y)$
$\lor Next(x, y, x', y')$
PROOF
⟨1⟩ DEFINE
$SysNext(p, q, v) \triangleq \exists u : Next(p, q, u, v)$
$EnvNext(p, q, u) \triangleq \exists v : Next(p, q, u, v)$
⟨1⟩1. SUFFICES
ASSUME $SysTurn(x, y) \land Inv(x, y)$
PROVE $Next(x, y, x', y')$
OBVIOUS
⟨1⟩3. SUFFICES
ASSUME $\neg Next(x, y, x', y')$
PROVE FALSE
OBVIOUS goal from ⟨1⟩1
⟨1⟩2. $\land \exists u : Next(x, y, u, y')$
$\quad \land \exists v : Next(x, y, x', v)$
⟨2⟩1. $\land SysNext(x, y, y')$
$\quad \land EnvNext(x, y, x')$

```
        OBVIOUS
    ⟨2⟩ QED
        BY ⟨2⟩1 DEF SysNext, EnvNext
⟨1⟩4. PICK u : Next(x, y, u, y')
    BY ⟨1⟩2
⟨1⟩10. PICK v : Next(x, y, x', v)
    BY ⟨1⟩2
⟨1⟩5. u ≠ x'
    ⟨2⟩1. SUFFICES ASSUME u = x'
                   PROVE FALSE
    ⟨2⟩2. Next(x, y, u, y')
        BY ⟨1⟩4
    ⟨2⟩3. Next(x, y, x', y')
        BY ⟨2⟩1, ⟨2⟩2
    ⟨2⟩4. ¬Next(x, y, x', y')
        BY ⟨1⟩3
    ⟨2⟩ QED
        BY ⟨2⟩3, ⟨2⟩4    goal from ⟨2⟩1
⟨1⟩6. SUFFICES u = x'
    BY ⟨1⟩5    goal from ⟨1⟩3
⟨1⟩7. ∧ ∃ a : Next(x, y, u, a)
      ∧ ∃ b : Next(x, y, x', b)
    BY ⟨1⟩4, ⟨1⟩10
⟨1⟩8. ∧ EnvNext(x, y, u)
      ∧ EnvNext(x, y, x')
    BY ⟨1⟩7 DEF EnvNext
⟨1⟩ QED
    ⟨2⟩1. ExistsUnique(LAMBDA r : EnvNext(x, y, r))
        ⟨3⟩1. SysTurn(x, y) ∧ Inv(x, y)
            BY ⟨1⟩1
        ⟨3⟩2. ∨ ¬(SysTurn(x, y) ∧ Inv(x, y))
              ∨ ExistsUnique(LAMBDA r : EnvNext(x, y, r))
            OBVIOUS
        ⟨3⟩ QED
            BY ⟨3⟩1, ⟨3⟩2
    ⟨2⟩ QED
        BY ⟨1⟩8, ⟨2⟩1 DEF ExistsUnique

THEOREM CPreSimplerByConjunctivity ≜
    ASSUME
        NEW Next, NEW SysNext, NEW EnvNext, NEW Target,
        Next ≡ (SysNext ∧ EnvNext)    Conjunctivity
    PROVE
        ( ∧ SysNext
```

$\land \mathit{EnvNext} \Rightarrow \mathit{Target})$

$\equiv$

$(\land \mathit{SysNext}$

$\land \mathit{EnvNext} \Rightarrow \land \mathit{Next}$

$\land \mathit{Target})$

PROOF OBVIOUS

$\land \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \mathit{Target}$
$\equiv$
$\land \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \mathit{Target}$
$\equiv$
$\land \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \land \mathit{SysNext}$
$\land \mathit{Target}$
$\equiv$
$\land \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \land \mathit{SysNext} \land \mathit{EnvNext}$
$\land \mathit{Target}$
$\equiv$
$\land \mathit{SysNext}$
$\land \mathit{EnvNext} \Rightarrow \land \mathit{Next}$
$\land \mathit{Target}$

THEOREM $\mathit{EquienablednessImpliesCartesianity} \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

CONSTANT $\mathit{EnvNext}(\_, \_, \_)$,

CONSTANT $\mathit{SysNext}(\_, \_, \_)$,

$(\exists u : \mathit{EnvNext}(x, y, u)) \equiv \exists v : \mathit{SysNext}(x, y, v)$

PROVE

The proof goal says that *NewNext* is *Cartesian*.

LET

$\mathit{NewNext}(p, q, u, v) \triangleq \land \mathit{EnvNext}(x, y, u)$
$\land \mathit{SysNext}(x, y, v)$

IN

$\land \mathit{SysNext}(x, y, y') \equiv \exists u : \mathit{NewNext}(x, y, u, y')$
$\land \mathit{EnvNext}(x, y, x') \equiv \exists v : \mathit{NewNext}(x, y, x', v)$

PROOF OBVIOUS

Actions \$*EnvNext*, *SysNext*\$ that result from \$*Unzip*\$ are enabled at the same states.

PROPOSITION $\mathit{EquiEnablednessFromUnzip} \triangleq$

ASSUME

VARIABLE x, VARIABLE y,

CONSTANT $\mathit{Next}(\_, \_, \_, \_)$,

CONSTANT $SysNext(\_, \_, \_)$,
CONSTANT $EnvNext(\_, \_, \_)$,
$\land \forall v : \ SysNext(x, y, v) \equiv \exists u : \ Next(x, y, u, v)$
$\land \forall u : \ EnvNext(x, y, u) \equiv \exists v : \ Next(x, y, u, v)$

PROVE

$(\exists u : \ EnvNext(x, y, u)) \equiv \exists v : \ SysNext(x, y, v)$

PROOF OBVIOUS

$\langle 1 \rangle 1.$ (ENABLED $EnvNext(x, y, x'))\ \equiv \exists u\colon EnvNext(x, y, u)$
$\langle 1 \rangle 2.$ $(\exists u\colon EnvNext(x, y, u))\ \equiv \exists u\colon \exists v\colon Next(x, y, u, v)$
$\langle 1 \rangle 3.$ $(\exists u\colon \exists v\colon Next(x, y, u, v))\ \equiv \exists v\colon \exists u\colon Next(x, y, u, v)$
$\langle 1 \rangle 4.$ $(\exists v\colon \exists u\colon Next(x, y, u, v))\ \equiv \exists v\colon SysNext(x, y, v)$
$\langle 1 \rangle 5.$ $(\exists v\colon SysNext(x, y, v))\ \equiv$ ENABLED $SysNext(x, y, y')$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 1$, $\langle 1 \rangle 2$, $\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$

COROLLARY

ASSUME

VARIABLE x, VARIABLE y,
CONSTANT $Next(\_, \_, \_, \_)$,
CONSTANT $SysNext(\_, \_, \_)$,
CONSTANT $EnvNext(\_, \_, \_)$,
$\land \forall v : \ SysNext(x, y, v) \equiv \exists u : \ Next(x, y, u, v)$
$\land \forall u : \ EnvNext(x, y, u) \equiv \exists v : \ Next(x, y, u, v)$

PROVE

LET

$NewNext(p, q, u, v) \triangleq \ \land EnvNext(x, y, u)$
$\qquad\qquad\qquad\qquad\quad \land SysNext(x, y, v)$

IN

$\land SysNext(x, y, y') \equiv \exists u : \ NewNext(x, y, u, y')$
$\land EnvNext(x, y, x') \equiv \exists v : \ NewNext(x, y, x', v)$

PROOF OBVIOUS

$\langle 1 \rangle 1.$ $(\exists u\colon EnvNext(x, y, u))\ \equiv \exists v\colon SysNext(x, y, v)$
BY *EquiEnablednessFromUnzip*
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 1$, *EquienablednessImpliesCartesianity*

COROLLARY

ASSUME

VARIABLE x, VARIABLE y,
CONSTANT $Next(\_, \_, \_, \_)$

PROVE

LET

The operators *SysNext* and *EnvNext* are already "balanced", but may not imply Next when conjoined. This is why we have to do the factorization as the next theorem below.

$SysNext(p, q, v) \ \triangleq \ \exists u : \ Next(p, q, u, v)$
$EnvNext(p, q, u) \ \triangleq \ \exists v : \ Next(p, q, u, v)$
$NewNext(p, q, u, v) \ \triangleq$

$\wedge \quad SysNext(x, y, v)$
$\wedge \quad EnvNext(x, y, u)$

NewNext is conjunctive and *Cartesian*,
so the controllable step operator is simpler when we apply Unzip to a property defined using *NewNext*.

IN

$\wedge\ SysNext(x, y, y') \ = \exists\, u :\ NewNext(x, y, u, y')$
$\wedge\ EnvNext(x, y, x') = \exists\, v :\ NewNext(x, y, x', v)$

PROOF OBVIOUS

PROPOSITION $PoofTheAntecedent \triangleq$

ASSUME

CONSTANT A, CONSTANT B,
CONSTANT C, CONSTANT D,
$A \Rightarrow D$

PROVE

$(\wedge A$
$\ \wedge (B \Rightarrow C))$
$\equiv$
$(\wedge A$
$\ \wedge (D \wedge B) \Rightarrow C)$

PROOF OBVIOUS

Even though *TLAPS* proves the above, below is a proof by hand.

PROPOSITION

ASSUME

CONSTANT A, CONSTANT B,
CONSTANT C, CONSTANT D,
$A \Rightarrow D$

PROVE

$(\wedge A$
$\ \wedge (B \Rightarrow C))$
$\equiv$
$(\wedge A$
$\ \wedge (D \wedge B) \Rightarrow C)$

PROOF

$\langle 1 \rangle 1.$ ASSUME $\wedge A$
$\wedge B \Rightarrow C$
PROVE $\wedge A$
$\wedge (D \wedge B) \Rightarrow C$

$\langle 2 \rangle 1.$ $\wedge A$
$\wedge C \vee \neg B$
BY $\langle 1 \rangle 1$

$\langle 2\rangle 2.\ (C \lor \neg B) \Rightarrow (C \lor \neg B \lor \neg D)$
OBVIOUS
$\langle 2\rangle 3.\ \land A$
$\land\ C \lor \neg B \lor \neg D$
BY $\langle 2\rangle 1$, $\langle 2\rangle 2$
$\langle 2\rangle$ QED
BY $\langle 2\rangle 3$
$\langle 1\rangle 2.$ ASSUME $\land A$
$\land\ (D \land B) \Rightarrow C$
PROVE $\land A$
$\land\ B \Rightarrow C$
$\langle 2\rangle 1.\ \land A$
$\land\ \lor \neg(D \land B)$
$\lor\ C$
BY $\langle 1\rangle 2$
$\langle 2\rangle 2.\ \land A$
$\land\ \lor \neg D \lor \neg B$
$\lor\ C$
BY $\langle 2\rangle 1$
$\langle 2\rangle 3.\ \lor\ \land A$
$\land\ \lor \neg B$
$\lor\ C$
$\lor\ \land A$
$\land\ \neg D$
BY $\langle 2\rangle 2$
$\langle 2\rangle 4.\ \neg(A \land \neg D)$
$\langle 3\rangle 1.\ A \Rightarrow D$
OBVIOUS
$\langle 3\rangle$ QED
BY $\langle 3\rangle 1$
$\langle 2\rangle 5.\ \lor\ \land A$
$\land\ B \Rightarrow C$
$\lor$ FALSE
BY $\langle 2\rangle 3$, $\langle 2\rangle 4$
$\langle 2\rangle$ QED
BY $\langle 2\rangle 5$
$\langle 1\rangle$ QED
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$

THEOREM *SeparatingTheRealizablePart* $\triangleq$
ASSUME
VARIABLE x, VARIABLE y,
CONSTANT $Next(\_, \_, \_, \_)$,

$\text{CONSTANT } \mathit{Target}(\_, \_),$
$\text{CONSTANT } \mathit{EnvNext}(\_, \_, \_),$
$\text{CONSTANT } \mathit{SysNext}(\_, \_, \_),$
$(\text{ENABLED } \mathit{SysNext}(x, y, y')) \Rightarrow \text{ENABLED } \mathit{EnvNext}(x, y, x')$

PROVE

LET

$\mathit{NewNext}(u, v) \triangleq$
$\quad \land \quad \mathit{SysNext}(x, y, v) \land \mathit{EnvNext}(x, y, u)$
$\quad \land \quad \forall w : \mathit{EnvNext}(x, y, w) \Rightarrow \mathit{Next}(x, y, w, v)$

The second conjunct shrinks the first in order to ensure receptivity at those states.

$\mathit{NewSysNext}(v) \triangleq \exists u : \mathit{NewNext}(u, v)$
$\mathit{NewEnvNext}(u) \triangleq \exists v : \mathit{NewNext}(u, v)$

$A \triangleq \exists v :$
$\quad \land \mathit{SysNext}(x, y, v)$
$\quad \land \forall u : \mathit{EnvNext}(x, y, u) \Rightarrow \land \mathit{Next}(x, y, u, v) \; \land \mathit{Target}(u, v)$

$B \triangleq \exists v :$
$\quad \land \mathit{NewSysNext}(v)$
$\quad \land \forall u : \mathit{NewEnvNext}(u) \Rightarrow \land \mathit{NewNext}(u, v) \; \land \mathit{Target}(x', v)$

$C \triangleq \exists v :$
$\quad \land \mathit{NewSysNext}(v)$
$\quad \land \forall u : \mathit{NewEnvNext}(u) \Rightarrow \mathit{Target}(u, v)$

IN

$\land \mathit{NewNext}(x', y') \Rightarrow \mathit{Next}(x, y, x', y')$
$\land A \equiv B$
$\land A \equiv C$
$\land \mathit{NewNext}(x', y') \equiv (\mathit{NewSysNext}(y') \land \mathit{NewEnvNext}(x'))$

PROOF

$\langle 1 \rangle$ DEFINE

$A \triangleq \exists v :$
$\quad \land \mathit{SysNext}(x, y, v)$
$\quad \land \forall u : \mathit{EnvNext}(x, y, u) \Rightarrow \land \mathit{Next}(x, y, u, v) \; \land \mathit{Target}(u, v)$

$\langle 1 \rangle 1.\ A \equiv$
$\exists v : \land \mathit{SysNext}(x, y, v)$
$\quad \land \forall u : \mathit{EnvNext}(x, y, u) \Rightarrow \mathit{Next}(x, y, u, v)$
$\quad \land \forall u : \mathit{EnvNext}(x, y, u) \Rightarrow \mathit{Target}(u, v)$

$\langle 1 \rangle 2.$ DEFINE $\mathit{NewSysNext}(p, q, v) \triangleq$
$\quad \land \mathit{SysNext}(p, q, v)$
$\quad \land \forall r : \mathit{EnvNext}(p, q, r) \Rightarrow \mathit{Next}(p, q, r, v)$

This definition of *NewSysNext* differs from that in the theorem statement. Nevertheless, we show their equivalence below.

$\langle 1 \rangle 3.\ A \equiv$

$\exists\, v : \ \land\, NewSysNext(x, y, v)$
$\qquad \land\, \forall\, u : \ EnvNext(x, y, u) \Rightarrow Target(u, v)$

BY $\langle 1 \rangle 1$ DEF $NewSysNext$

$\langle 1 \rangle 4.\ A \equiv$

$\exists\, v : \ \forall\, u :$
$\quad \land\ NewSysNext(x, y, v)$
$\quad \land\ \lor \neg \land\, EnvNext(x, y, u)$
$\qquad\qquad \land$ ENABLED $NewSysNext(x, y, y')$
$\qquad \lor\ Target(u, v)$

$\langle 2 \rangle 1.$ ASSUME NEW v
PROVE $NewSysNext(x, y, v) \equiv \forall\, u : \ NewSysNext(x, y, v)$
BY DEF $NewSysNext$

$\langle 2 \rangle 2.$ ASSUME NEW v
PROVE $NewSysNext(x, y, v) \Rightarrow$ ENABLED $NewSysNext(x, y, y')$
BY DEF $NewSysNext$

$\langle 2 \rangle 3.\ (\exists\, v : \ \land\, NewSysNext(x, y, v)$
$\qquad\qquad \land\, \forall\, u : \ EnvNext(x, y, u) \Rightarrow Target(u, v))$
$\quad \equiv$
$\quad (\exists\, v :$
$\qquad \land\, \forall\, u : \ NewSysNext(x, y, v)$
$\qquad \land\, \forall\, u : \ EnvNext(x, y, u) \Rightarrow Target(u, v))$
BY $\langle 2 \rangle 1$

$\langle 2 \rangle 4.\ A \equiv$
$\quad \exists\, v : \ \forall\, u :$
$\qquad \land\ NewSysNext(x, y, v)$
$\qquad \land\ EnvNext(x, y, u) \Rightarrow Target(u, v)$
BY $\langle 1 \rangle 3$, $\langle 2 \rangle 3$

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

$\langle 1 \rangle 5.$ DEFINE $NewNext(p, q, u, v) \ \triangleq$
$\quad \land\, SysNext(p, q, v) \land EnvNext(p, q, u)$
$\quad \land\, \forall\, r : \ EnvNext(p, q, r) \Rightarrow Next(p, q, r, v)$

$\langle 1 \rangle 6.$ ASSUME NEW p, NEW q, NEW u, NEW v
PROVE $NewNext(p, q, u, v) \ \equiv \ \land\, NewSysNext(p, q, v)$
$\qquad\qquad\qquad\qquad\qquad \land\, EnvNext(p, q, u)$
BY DEF $NewNext$, $NewSysNext$

$\langle 1 \rangle 7.$ ASSUME NEW p, NEW q, NEW v
PROVE $NewSysNext(p, q, v) \ \equiv \ \exists\, u : \ NewNext(p, q, u, v)$

$\langle 2 \rangle 1.\ (\exists\, u : \ NewNext(p, q, u, v))$
$\quad \equiv \exists\, u : \ \land\, SysNext(p, q, v) \land EnvNext(p, q, u)$
$\qquad\qquad \land\, \forall\, r : \ EnvNext(p, q, r) \Rightarrow Next(p, q, r, v)$
BY DEF $NewNext$

$\langle 2 \rangle 2.\ (\exists\, u : \ NewNext(p, q, u, v))$
$\quad \equiv \land\, SysNext(p, q, v)$

$\land \exists u : \ EnvNext(p, q, u)$
$\land \forall r : \ EnvNext(p, q, r) \Rightarrow Next(p, q, r, v)$
BY $\langle 2 \rangle 1$
$\langle 2 \rangle 3.$ $SysNext(p, q, v) \Rightarrow \exists u : \ EnvNext(p, q, u)$
$\langle 3 \rangle 1.$ $SysNext(p, q, v) \Rightarrow \exists s : \ SysNext(p, q, s)$
OBVIOUS
$\langle 3 \rangle 2.$ $(\exists s : \ SysNext(p, q, s)) \ \Rightarrow \ \exists u : \ EnvNext(p, q, u)$
$\langle 4 \rangle 1.$ (ENABLED $SysNext(x, y, y')$) $\Rightarrow$ ENABLED $EnvNext(x, y, x')$
OBVIOUS BY *SeparatingTheRealizablePart*!assumption
$\langle 4 \rangle$ QED
BY $\langle 4 \rangle 1$
$\langle 3 \rangle$ QED
BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$
$\langle 2 \rangle 4.$ $(\exists u : \ NewNext(p, q, u, v))$
$\equiv \land SysNext(p, q, v)$
$\land \forall r : \ EnvNext(p, q, r) \Rightarrow Next(p, q, r, v)$
BY $\langle 2 \rangle 2$, $\langle 2 \rangle 3$
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 4$ DEF *NewSysNext*
$\langle 1 \rangle 8.$ DEFINE $NewEnvNext(p, q, u) \triangleq \exists v : \ NewNext(p, q, u, v)$
$\langle 1 \rangle 9.$ ASSUME NEW p, NEW q, NEW u
PROVE $NewEnvNext(p, q, u) \equiv \land EnvNext(p, q, u)$
$\land$ ENABLED $NewSysNext(p, q, y')$
$\langle 2 \rangle$ DEFINE $F \triangleq NewEnvNext(p, q, u)$
$\langle 2 \rangle 1.$ F
$\equiv \exists v : \ \land SysNext(p, q, v) \land EnvNext(p, q, u)$
$\land \forall r : \ EnvNext(p, q, r) \Rightarrow Next(p, q, r, v)$
$\langle 2 \rangle 2.$ F
$\equiv \exists v : \ \land NewSysNext(p, q, v)$
$\land EnvNext(p, q, u)$
$\langle 2 \rangle 3.$ $F \ \equiv \ EnvNext(p, q, u) \land \exists v : \ NewSysNext(p, q, v)$
$\langle 2 \rangle 4.$ $F \ \equiv \ EnvNext(p, q, u) \land$ ENABLED $NewSysNext(p, q, y')$
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 4$ DEF F
$\langle 1 \rangle 10.$ $A \equiv$
$\exists v : \ \forall u :$
$\land NewSysNext(x, y, v)$
$\land NewEnvNext(x, y, u) \Rightarrow Target(u, v)$
BY $\langle 1 \rangle 4$, $\langle 1 \rangle 9$
$\langle 1 \rangle 11.$ ASSUME NEW p, NEW q, NEW u, NEW v
PROVE $NewNext(p, q, u, v) \equiv \land NewSysNext(p, q, v)$
$\land NewEnvNext(p, q, u)$
$\langle 2 \rangle 1.$ $NewNext(p, q, u, v) \ \equiv \ \land NewSysNext(p, q, v)$
$\land EnvNext(p, q, u)$
BY $\langle 1 \rangle 6$

$\langle 2 \rangle 2.$ $\mathit{NewSysNext}(p, q, v) \Rightarrow$ ENABLED $\mathit{NewSysNext}(p, q, y')$
OBVIOUS
$\langle 2 \rangle 3.$ $\mathit{NewNext}(p, q, u, v) \equiv \land \mathit{NewSysNext}(p, q, v)$
$\land \mathit{EnvNext}(p, q, u)$
$\land$ ENABLED $\mathit{NewSysNext}(p, q, y')$
BY $\langle 2 \rangle 1, \langle 2 \rangle 2$
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 3, \langle 1 \rangle 9$
$\langle 1 \rangle 12.$ $A \equiv$
$\exists\, v : \forall\, u :$
$\land \mathit{NewSysNext}(x, y, v)$
$\land \mathit{NewEnvNext}(x, y, u) \Rightarrow \land \mathit{NewNext}(x, y, u, v)$
$\land \mathit{Target}(u, v)$
BY $\langle 1 \rangle 10, \langle 1 \rangle 11,$ $\mathit{CPreSimplerByConjunctivity}$
$\langle 1 \rangle 13.$ ASSUME NEW p, NEW q, NEW u, NEW v
PROVE $\mathit{NewNext}(p, q, u, v) \Rightarrow \mathit{Next}(p, q, u, v)$
$\langle 2 \rangle 1.$ SUFFICES ASSUME $\mathit{NewNext}(p, q, u, v)$
PROVE $\mathit{Next}(p, q, u, v)$
OBVIOUS
$\langle 2 \rangle 2.$ $\land \mathit{SysNext}(p, q, v) \land \mathit{EnvNext}(p, q, u)$
$\land \forall\, r : \mathit{EnvNext}(p, q, r) \Rightarrow \mathit{Next}(p, q, r, v)$
BY $\langle 2 \rangle 1$ DEF $\mathit{NewNext}$
$\langle 2 \rangle 3.$ $\land \mathit{EnvNext}(p, q, u)$
$\land \forall\, r : \mathit{EnvNext}(p, q, r) \Rightarrow \mathit{Next}(p, q, r, v)$
BY $\langle 2 \rangle 2$
$\langle 2 \rangle$ QED
BY $\langle 2 \rangle 3$ goal from $\langle 2 \rangle 1$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 7, \langle 1 \rangle 10, \langle 1 \rangle 11, \langle 1 \rangle 12, \langle 1 \rangle 13$ DEF $\mathit{NewNext}, \mathit{NewEnvNext}$

COROLLARY
ASSUME
VARIABLE p, VARIABLE q,
CONSTANT $\mathit{Next}(\_, \_, \_, \_)$,
CONSTANT $\mathit{Target}(\_, \_)$
PROVE
LET
$\mathit{SysNext}(x, y, v) \triangleq \exists\, u : \mathit{Next}(x, y, u, v)$
$\mathit{EnvNext}(x, y, u) \triangleq \exists\, v : \mathit{Next}(x, y, u, v)$
$\mathit{NewNext}(x, y, u, v) \triangleq$
$\land \mathit{SysNext}(x, y, v) \land \mathit{EnvNext}(x, y, u)$
$\land \forall\, w : \mathit{EnvNext}(x, y, w) \Rightarrow \mathit{Next}(x, y, w, v)$
$\mathit{NewSysNext}(x, y, v) \triangleq \exists\, u : \mathit{NewNext}(x, y, u, v)$
$\mathit{NewEnvNext}(x, y, u) \triangleq \exists\, v : \mathit{NewNext}(x, y, u, v)$

$A(x, y) \triangleq \exists v : \forall u :$
$\land SysNext(x, y, v)$
$\land EnvNext(x, y, u) \Rightarrow \land Next(x, y, u, v)$
$\land Target(u, v)$

IN

$\land NewNext(p, q, p', q') \Rightarrow Next(p, q, p', q')$

Conjunctivity and *Cartesianity*

$\land NewNext(p, q, p', q')$
$\equiv \land NewSysNext(p, q, q')$
$\land NewEnvNext(p, q, p')$

$\land A(p, q) \equiv \exists v : \forall u :$
$\land NewSysNext(p, q, v)$
$\land NewEnvNext(p, q, u) \Rightarrow \land NewNext(p, q, u, v)$
$\land Target(u, v)$

$\land A(p, q) \equiv \exists v : \forall u :$
$\land NewSysNext(p, q, v)$
$\land NewEnvNext(p, q, u) \Rightarrow Target(u, v)$

PROOF

BY *EquiEnablednessFromUnzip*, *SeparatingTheRealizablePart*

Unzip has desirable properties:

1. the assumption is by construction safety, and
2. the assumption is well-separated.

Recall that:

$Unzip(P) \equiv WPH(WPH(P, P), P)$

The assumption in the *WhilePlusHalf* that defines *Unzip* is a safety property. That this property, namely $WPH(C, C, y, x)$, is safety follows similarly to the proof of *WhilePlusMachineClosedRepr*.

PROPOSITION

ASSUME

TEMPORAL $P(\_, \_)$,

VARIABLE x, VARIABLE y

PROVE

LET $C \triangleq Cl(P, x, y)$

IN $WPH(P, P, y, x) \equiv WPH(C, C, y, x)$

PROOF

$\langle 1 \rangle$ DEFINE

$C \triangleq Cl(P, x, y)$

$\langle 1 \rangle 1.$ $WPH(P, P, y, x) \equiv \land WPH(C, C, y, x)$
$\land P(y, x) \Rightarrow P(y, x)$

BY *WhilePlusHalfAsConj*

$\langle 1 \rangle 2.$ $P(y, x) \Rightarrow P(y, x)$

OBVIOUS

⟨1⟩ QED
BY ⟨1⟩1, ⟨1⟩2

This proposition ensures well-separation of the first and second argument given to *WhilePlusHalf* for defining *Unzip*.

PROPOSITION
ASSUME
TEMPORAL $P(\_, \_)$,
VARIABLE x, VARIABLE y
PROVE
LET
$Q(u, v) \triangleq P(v, u)$
$E(u, v) \triangleq WPH(Q, Q, v, u)$
IN
$\land\ Cl(P, x, y) \Rightarrow Cl(E, x, y)$
$\land\ P(x, y) \Rightarrow Cl(E, x, y)$
PROOF
⟨1⟩ DEFINE
$Q(u, v) \triangleq P(v, u)$
$E(u, v) \triangleq WPH(Q, Q, v, u)$
⟨1⟩1. $P(x, y) \Rightarrow WPH(Q, Q, y, x)$
⟨2⟩1. $WPH(Q, Q, y, x) \equiv$
$\forall\, b : \lor \neg \land\ MayUnstep(b)$
$\land\ Front(Q, y, x, b)$
$\lor\ FrontPlusHalf(Q, y, x, b)$
BY DEF *WPH*, *WhilePlusHalf*
⟨2⟩2. ASSUME VARIABLE b
PROVE $P(x, y) \Rightarrow FrontPlusHalf(Q, y, x, b)$
⟨3⟩1. $FrontPlusHalf(Q, y, x, b)$
$\equiv \exists\, u, v :$
$\land\ Q(u, v)$
$\land\ SamePrefix(b, u, v, y, x)$
$\land\ PlusHalf(b, v, x)$
BY DEF *FrontPlusHalf*
⟨3⟩2. ASSUME VARIABLE u, VARIABLE v
PROVE
$SamePrefix(b, u, v, y, x)$
$\equiv SamePrefix(b, v, u, x, y)$
BY *SwapInSamePrefix*
⟨3⟩3. $FrontPlusHalf(Q, y, x, b)$
$\equiv \exists\, v, u :$
$\land\ P(v, u)$
$\land\ SamePrefix(b, v, u, x, y)$
$\land\ PlusHalf(b, v, x)$

BY ⟨3⟩1, ⟨3⟩2

⟨3⟩4. $FrontPlusHalf(Q, y, x, b)$
$\equiv \exists\, u, v :$
$\land\ P(u, v)$
$\land\ SamePrefix(b, u, v, x, y)$
$\land\ PlusHalf(b, u, x)$

BY ⟨3⟩3

⟨3⟩5. $P(x, y) \Rightarrow \exists\, u, v :$
$\land\ \Box(\langle u, v\rangle = \langle x, y\rangle)$
$\land\ P(x, y)$

OBVIOUS

⟨3⟩6. $P(x, y) \Rightarrow \exists\, u, v :$
$\land\ \Box(\langle u, v\rangle = \langle x, y\rangle)$
$\land\ u = x$
$\land\ P(u, v)$

BY ⟨3⟩5

⟨3⟩7. $P(x, y) \Rightarrow \exists\, u, v :$
$\land\ \Box(\langle u, v\rangle = \langle x, y\rangle)$
$\land\ u = x$
$\land\ \Box[u' = x']_{\langle b, u, x\rangle}$
$\land\ P(u, v)$

BY ⟨3⟩6 TLA rule

⟨3⟩8. $P(x, y) \Rightarrow \exists\, u, v :$
$\land\ \Box(b \Rightarrow (\langle u, v\rangle = \langle x, y\rangle))$
$\land\ u = x$
$\land\ \Box[b \Rightarrow (u' = x')]_{\langle b, u, x\rangle}$
$\land\ P(u, v)$

BY ⟨3⟩7

⟨3⟩ QED

BY ⟨3⟩4, ⟨3⟩8

⟨2⟩ QED

BY ⟨2⟩1, ⟨2⟩2

⟨1⟩2. $P(x, y) \Rightarrow E(x, y)$

BY ⟨1⟩1 DEF E

⟨1⟩3. $P(x, y) \Rightarrow Cl(E, x, y)$

⟨2⟩1. $E(x, y) \Rightarrow Cl(E, x, y)$

BY *ClosureImplied*

⟨2⟩ QED

BY ⟨1⟩2, ⟨2⟩1

⟨1⟩4. $Cl(P, x, y) \Rightarrow Cl(E, x, y)$

⟨2⟩1. $Cl(P, x, y) \Rightarrow Cl(Cl(E, x, y), x, y)$

BY ⟨1⟩3, *ClosureIsMonotonic*

⟨2⟩2. $Cl(E, x, y) \equiv Cl(Cl(E, x, y), x, y)$

BY *ClosureIdempotent*

⟨2⟩ QED

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

$\langle 1\rangle$ QED

BY $\langle 1\rangle 3$, $\langle 1\rangle 4$ DEF E

Expand an expression that occurs in the first argument of *WhilePlusHalf* within *Unzip*.

PROPOSITION
ASSUME
TEMPORAL $P(\_, \_)$,
VARIABLE x, VARIABLE y, VARIABLE b
PROVE
LET
$Q(u, v) \triangleq P(v, u)$
IN
$Front(Q, y, x, b) \equiv Front(P, x, y, b)$

PROOF

$\langle 1\rangle$ DEFINE
$Q(u, v) \triangleq P(v, u)$

$\langle 1\rangle 1.$ $Front(Q, y, x, b)$
$\equiv \exists u, v : \land Q(u, v)$
$\land SamePrefix(b, u, v, y, x)$
BY DEF $Front$

$\langle 1\rangle 2.$ ASSUME VARIABLE u, VARIABLE v
PROVE
$SamePrefix(b, u, v, y, x)$
$\equiv SamePrefix(b, v, u, x, y)$
BY $SwapInSamePrefix$

$\langle 1\rangle 3.$ ASSUME VARIABLE u, VARIABLE v
PROVE $Q(u, v) \equiv P(v, u)$
BY DEF Q

$\langle 1\rangle 4.$ $Front(Q, y, x, b)$
$\equiv \exists u, v : \land P(v, u)$
$\land SamePrefix(b, v, u, x, y)$
BY $\langle 1\rangle 1$, $\langle 1\rangle 2$, $\langle 1\rangle 3$

$\langle 1\rangle 5.$ $Front(Q, y, x, b)$
$\equiv \exists v, u : \land P(v, u)$
$\land SamePrefix(b, v, u, x, y)$
BY $\langle 1\rangle 4$

$\langle 1\rangle 6.$ $Front(P, x, y, b)$
$\equiv \exists v, u : \land P(v, u)$
$\land SamePrefix(b, v, u, x, y)$
BY DEF $Front$

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

THEOREM $NotExtensible \triangleq$

ASSUME

$$\exists\, tau : \begin{array}{l} \land\ IsABehavior(tau) \\ \land\ tau \models B \\ \land\ tau[0].a = 1 \\ \land\ tau[1].a = 20 \\ \land\ tau[0].b = 2 \end{array}$$

PROVE FALSE

PROOF

$\langle 1 \rangle 3.$ PICK tau :

$\land$ $IsABehavior(tau)$

$\land$ $tau \models B$

$\land$ LET

$s0 \triangleq tau[0]$

$s1 \triangleq tau[1]$

IN

$\land\ s0.a = 1$

$\land\ s1.a = 20$

$\land\ s0.b = 2$

$\langle 1 \rangle$ DEFINE

$s0 \triangleq tau[0]$

$s1 \triangleq tau[1]$

$IsNonstuttering(step) \triangleq step[1] \neq step[2]$

$\langle 1 \rangle 4.$ $tau \models \Box\Diamond(b = 2)$

BY $\langle 1 \rangle 3$ DEF B

$\langle 1 \rangle 1.$ $IsNonstuttering(\langle s0, s1 \rangle)$

$\langle 2 \rangle 1.$ $s0.a \neq s1.a$

BY DEF $s0$, $s1$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 1$ DEF $IsNonstuttering$

$\langle 1 \rangle 2.$ $s1.b = 1$

BY $\langle 1 \rangle 1$ DEF B, $s0$

$\langle 1 \rangle 5.$ $\exists\, i \in Nat : tau[i].b \neq tau[i+1].b$

A step that changes b eventually occurs.

BY $\langle 1 \rangle 2$, $\langle 1 \rangle 4$ DEF $s1$

$\langle 1 \rangle 6.$ $\forall\, n \in Nat$:

$\lor\ tau[n] = tau[n+1]$

$\lor\ tau[n] \neq s1$

$\lor\ \langle tau[n], tau[n+1] \rangle[[b' \neq b]]$

Any nonstuttering step from s_1 must change b.

BY $\langle 1 \rangle 2$, $\langle 1 \rangle 3$ DEF $s1$, B

$\langle 1 \rangle 7.$ $\exists\, j \in Nat$:

$\land\ \forall\, k \in 1 \,..\, j : tau[k] = s1$

$\land\ \langle tau[j], tau[j+1] \rangle[[b' \neq b]]$

The earliest nonstutering step after $tau[1]$ does change b.

BY $\langle 1 \rangle 5$, $\langle 1 \rangle 6$, *LeastNumberPrinciple*

$\langle 1 \rangle 8.$ $tau[j+1].b = 20$

$\langle 2 \rangle 1.$ $tau[j].a = 20$

$\langle 3 \rangle 1.$ $tau[j] = s1$

BY $\langle 1 \rangle 7$

$\langle 3 \rangle 2.$ $s1.a = 20$

BY $\langle 1 \rangle 3$ DEF $s1$

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle 2.$ $\langle tau[j], tau[j+1] \rangle[[b' = a]]$

$\langle 3 \rangle 1.$ $\langle tau[j], tau[j+1] \rangle[[b' \neq b]]$

BY $\langle 1 \rangle 7$

$\langle 3 \rangle 2.$ $tau \models B$

BY $\langle 1 \rangle 3$

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 1$, $\langle 3 \rangle 2$ DEF B

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle 9.$ $tau[j+1].b \in 1 \,..\, 2$

$\langle 2 \rangle 1.$ $tau \models B$

BY $\langle 1 \rangle 3$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 1$ DEF B

$\langle 1 \rangle$ QED

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

MODULE *Realizability*

A definition of what it means for a function to realize a property.

References

Ioannis *Filippidis*, *Richard M.* Murray "Formalizing synthesis in TLA+" Technical Report, *California* Institute of Technology, 2016 http://*resolver.caltech.edu*/*CaltechCDSTR*:2016.004

Leslie *Lamport* "Miscellany" 21 *April* 1991, note sent to TLA mailing list http://*lamport.org*/tla/notes/91-04-21.*txt*

EXTENDS *FiniteSets*

$$IsAFunction(f) \triangleq f = [u \in \text{DOMAIN } f \mapsto f[u]]$$

$$IsAFiniteFcn(f) \triangleq \begin{array}{l} \land IsAFunction(f) \\ \land IsFiniteSet(\text{DOMAIN } f) \end{array}$$

MODULE *Inner*

VARIABLES x, y

CONSTANTS $f, g, mem0$

$$Realization(mem, e(\_, \_)) \triangleq$$

$$\begin{array}{l} \text{LET} \\ \quad v \triangleq \langle mem, x, y\rangle \\ \quad A \triangleq \begin{array}{l} \land x' = f[v] \\ \land mem' = g[v] \end{array} \\ \text{IN} \\ \quad \land mem = mem0 \\ \quad \land \Box[e(v, v') \Rightarrow A]_v \\ \quad \land \text{WF}_{\langle mem, x\rangle}(e(v, v') \land A) \end{array}$$

$$Realize(Phi(\_, \_), e(\_, \_)) \triangleq$$

$$\begin{array}{l} \land \ IsAFiniteFcn(f) \land IsAFiniteFcn(g) \\ \land \ (\exists\!\!\exists\, mem : \ Realization(mem, e)) \Rightarrow Phi(x, y) \end{array}$$

$$Inner(f, g, mem0, x, y) \triangleq \text{INSTANCE } Inner$$

$$IsARealization(f, g, mem0, Phi(\_, \_), e(\_, \_)) \triangleq$$

$$\forall\!\!\forall\, x, y : \quad Inner(f, g, mem0, x, y)!Realize(Phi, e)$$

$$IsRealizable(Phi(\_, \_), e(\_, \_)) \triangleq$$

$$\exists f, g, mem0 : \quad IsARealization(f, g, mem0, Phi, e)$$

MODULE *HistoryIsRealizable*

For a specification that includes history-determined variables, we prove that
it suffices to synthesize an implementation with the history variables unhidden. More precisely

LET
  $Spec(x, h) \triangleq Prop(x) \land History(x, h)$
  $SpecH(x) \triangleq \exists h: Spec(x, h)$
IN
  $IsRealizable(SpecH) \equiv IsRealizable(Spec)$

This result is useful for using temporal synthesis algorithms that do not reason about $\exists$ (for example $GR(k)$ synthesis), and then hiding the history variables, in order to obtain an implementation for properties that contain temporal quantification of only history variables.

Author: *Ioannis Filippidis*

References

[1] *M*. Abadi and *L*. Lamport "The existence of refinement mappings", *TCS*, 1991, 10.1016/0304-3975(91)90224-P

[2] *M*. Abadi and *L*. Lamport "An old-fashioned recipe for real time" *TOPLAS*, 1994, 10.1145/186025.186058

[3] *N*. Piterman and A. Pnueli and *Y*. Sa'ar "Synthesis of *reactive*(1) designs", *VMCAI*, 2006, 10.1007/11609773_24

[4] *L*. Lamport and *S*. Merz "Auxiliary variables in TLA+", *ArXiv*, 2017, https://*arxiv.org*/*pdf*/1703.05121.*pdf*

EXTENDS *TemporalLogic*, *TLAPS*

MODULE *HistoryDeterminedVar*

VARIABLE v,
CONSTANT $Init(\_, \_)$ corresponds to f in [2, *Eq*.(4)]
CONSTANT $Next(\_, \_, \_)$ corresponds to g in [2, *Eq*.(4)]

$Hist(h, v) \triangleq$
  LET
    $N \triangleq \langle h' = Next(h, v, v') \rangle_v$
  IN
    $\land\ h = Init(v)$
    $\land\ \Box[N]_{\langle h, v \rangle}$

THEOREM $HistoryExists \triangleq$
  $\forall v : \exists h : Hist(h, v)$
  PROOF OMITTED

ATTENTION: Below this point, $\boldsymbol{\exists}$ stands for stutter-sensitive quantification. For simplicity, a stutter-sensitive definition of realizability is used, in raw TLA+. An analogous result can be shown in TLA+.

We also use a simplified definition of *IsRealizable*, with mu omitted. Realizations that contain an initial condition are used.

MODULE *RawHistoryDeterminedVar*

```
VARIABLE v,
CONSTANT Init(_, _)
CONSTANT Next(_, _, _)

Hist(h, v) ≜
    LET
        N ≜ h' = Next(h, v, v')
    IN
        ∧ h = Init(v)
        ∧ □N

THEOREM HistoryExists ≜
    ∀ v : ∃ h : Hist(h, v)
    PROOF OMITTED
```

```
PROPOSITION ImplEE ≜
    ASSUME
        TEMPORAL A(_), TEMPORAL B(_),
        ∀ q : A(q) ⇒ B(q)
    PROVE
        (∃ q : A(q)) ⇒ (∃ q : B(q))
```

```
PROPOSITION HidingHistoryPreservesRealizability ≜
    ASSUME
        CONSTANT I(_, _, _),
        TEMPORAL Phi(_, _, _),
        IsRealizable(I, Phi)
    PROVE
        LET
            Init(x, y)  ≜ ∃ q : I(x, y, q)
            PhiH(x, y)  ≜ ∃ q : Phi(x, y, q)
        IN
            IsRealizable(Init, PhiH)
PROOF
⟨1⟩ DEFINE
    g(x, y) ≜ CHOOSE q : I(x, y, q)
```

$Init(x, y) \triangleq \exists q : I(x, y, q)$

We cannot use CHOOSE to define fx, ... because CHOOSE cannot be applied to a temporal-level expression. PICK can, but can occur only in proofs.

$\langle 1 \rangle 1.$ PICK $fx1, fq, fm1, r$:

$$
\begin{aligned}
&\land\ IsAFunction(fx1)\\
&\land\ IsAFunction(fq)\\
&\land\ IsAFunction(fm1)\\
&\land\ \forall x, y, q :\\
&\qquad \lor \neg \exists m :\\
&\qquad\qquad \text{LET}\\
&\qquad\qquad\qquad args \triangleq \langle x, y, q, m \rangle\\
&\qquad\qquad \text{IN}\\
&\qquad\qquad\qquad \land I(x, y, q)\\
&\qquad\qquad\qquad \land m = r\\
&\qquad\qquad\qquad \land \Box \land x' = fx1[args]\\
&\qquad\qquad\qquad\qquad \land q' = fq[args]\\
&\qquad\qquad\qquad\qquad \land m' = fm1[args]\\
&\qquad \lor Phi(x, y, q)
\end{aligned}
$$

BY DEF $IsRealizable$

and $HidingHistoryPreservesRealizability$! assumption

$\langle 1 \rangle 2.$ $\forall x, y$:

$$
\begin{aligned}
&\lor \neg \exists q, m :\\
&\qquad \text{LET}\\
&\qquad\qquad args \triangleq \langle x, y, q, m \rangle\\
&\qquad \text{IN}\\
&\qquad\qquad \land I(x, y, q)\\
&\qquad\qquad \land m = r\\
&\qquad\qquad \land \Box \land x' = fx1[args]\\
&\qquad\qquad\qquad \land q' = fq[args]\\
&\qquad\qquad\qquad \land m' = fm1[args]\\
&\lor \exists q : Phi(x, y, q)
\end{aligned}
$$

BY $\langle 1 \rangle 1$, $ImplEE$

$\langle 1 \rangle$ DEFINE

$$
\begin{aligned}
&repack(t) \triangleq\\
&\qquad \text{LET}\\
&\qquad\qquad x \triangleq t[1]\\
&\qquad\qquad y \triangleq t[2]\\
&\qquad\qquad q \triangleq t[3]\\
&\qquad\qquad m1 \triangleq t[4]\\
&\qquad \text{IN}\\
&\qquad\qquad \langle x, y, \langle m1, q \rangle \rangle\\
&Value(t, F(\_)) \triangleq\\
&\qquad \text{LET}\\
&\qquad\qquad x \triangleq t[1]\\
&\qquad\qquad y \triangleq t[2]
\end{aligned}
$$

```
            m2 ≜ t[3]
             initial arguments
            argsi ≜ ⟨x, y, g(x, y), r⟩
            init ≜ F(argsi)
             arguments when changes occur
            args ≜ ⟨x, y, m2[2], m2[1]⟩
            later ≜ F(args)
        IN
            IF m2 = ⟨r⟩
                THEN init
                ELSE later
    fx2 ≜
        LET
            OldDom ≜ DOMAIN fx1
            R ≜ {repack(t) : t ∈ OldDom}
            S ≜ R ∪ {⟨x, y, ⟨r⟩⟩}
            F(args) ≜ fx1[args]
        IN
            [t ∈ S ↦ Value(z, F)]
    fm2 ≜
        LET
            OldDoms ≜ (DOMAIN fm1) ∪ DOMAIN fq
                 here ∪ is necessary
            R ≜ {repack(t) : t ∈ OldDoms}
            S ≜ R ∪ {⟨x, y, ⟨r⟩⟩}
            F(args) ≜ ⟨fm1[args], fq[args]⟩
        IN
            [t ∈ S ↦ Value(z, F)]
⟨1⟩3. ∀ x, y :
        ∨ ¬∃ m2 :
            LET
                args ≜ ⟨x, y, m2⟩
            IN
                ∧ ∃ q : I(x, y, q)
                ∧ m2 = ⟨r⟩
                ∧ □ ∧ x′ = fx2[args]
                    ∧ m2′ = fm2[args]
        ∨ ∃ q : Phi(x, y, q)
    ⟨2⟩1. ASSUME VARIABLE x, VARIABLE y
          PROVE
            ∨ ¬∃ m2 :
                LET
                    args ≜ ⟨x, y, m2⟩
                IN
                    ∧ ∃ q : I(x, y, q)
```

$\land\ m2 = \langle r \rangle$
$\land\ \Box \land x' = fx2[args]$
$\qquad \land m2' = fm2[args]$

$\lor\ \exists\, q,\ m :$

LET

$\quad args \triangleq \langle x,\ y,\ q,\ m \rangle$

IN

$\quad \land\ I(x,\ y,\ q)$
$\quad \land\ m = r$
$\quad \land\ \Box \land x' = fx1[args]$
$\qquad \land q' = fq[args]$
$\qquad \land m' = fm1[args]$

$\langle 3 \rangle$ DEFINE $A \triangleq$

$\exists\, m2 :$

LET

$\quad args \triangleq \langle x,\ y,\ m2 \rangle$

IN

$\quad \land\ \exists\, q :\ I(x,\ y,\ q)$
$\quad \land\ m2 = \langle r \rangle$
$\quad \land\ \Box \land x' = fx2[args]$
$\qquad \land m2' = fm2[args]$

$\langle 3 \rangle 1.\ \lor \neg A$
$\lor\ \exists\, m2 :$

q is determined by history

$\land\ \exists\, q :\ \land q = g(x,\ y)$
$\qquad \land \Box(q' = m2[2]')$

m is determined by history

$\land\ \exists\, m :\ \land m = r$
$\qquad \land \Box(m' = m2[1]')$

from A

$\land$ LET

$\quad args \triangleq \langle x,\ y,\ m2 \rangle$

IN

$\quad \land\ \exists\, q :\ I(x,\ y,\ q)$
$\quad \land\ m2 = \langle r \rangle$
$\quad \land\ \Box \land x' = fx2[args]$
$\qquad \land m2' = fm2[args]$

BY $RawHistoryDeterminedVar!HistoryExists$

$\langle 3 \rangle 2.\ \lor \neg A$
$\lor\ \exists\, m2,\ q,\ m :$

LET

$\quad args \triangleq \langle x,\ y,\ m2 \rangle$

IN

$\quad \land\ \exists\, z :\ I(x,\ y,\ z)$ avoid synonymy with q
$\quad \land\ q = g(x,\ y)$

$$
\begin{array}{l}
\quad\quad\quad\wedge\ m = r \\
\quad\quad\quad\wedge\ \Box \wedge q' \ = m2[2]' \\
\quad\quad\quad\quad\quad \wedge\ m' = m2[1]' \\
\quad\quad\quad\wedge\ m2 = \langle r \rangle \\
\quad\quad\quad\wedge\ \Box \wedge x' = fx2[args] \\
\quad\quad\quad\quad\quad \wedge\ m2' = fm2[args]
\end{array}
$$

BY $\langle 3 \rangle 1$

$\langle 3 \rangle 3.$ $\vee \neg A$

$$
\begin{array}{l}
\vee\ \exists\, m2,\ q,\ m : \\
\quad \text{LET} \\
\quad\quad args i \triangleq \langle x,\ y,\ g(x,\ y),\ r \rangle \\
\quad\quad args \triangleq \langle x,\ y,\ m2[2],\ m2[1] \rangle \\
\quad \text{IN} \\
\quad\quad \wedge\ I(x,\ y,\ q) \\
\quad\quad \wedge\ q \ = g(x,\ y) \\
\quad\quad \wedge\ m = r \\
\quad\quad \wedge\ \Box \wedge q' \ = m2[2]' \\
\quad\quad\quad\quad \wedge\ m' = m2[1]' \\
\quad\quad \wedge\ m2 = \langle r \rangle \\
\quad\quad \wedge\ \Box \wedge x' = \text{IF } m2 = \langle r \rangle \\
\quad\quad\quad\quad\quad\quad \text{THEN } fx1[argsi] \\
\quad\quad\quad\quad\quad\quad \text{ELSE } fx1[args] \\
\quad\quad\quad\quad \wedge\ m2' = \text{IF } m2 = \langle r \rangle \\
\quad\quad\quad\quad\quad\quad\quad \text{THEN } \langle fm1[argsi],\ fq[argsi] \rangle \\
\quad\quad\quad\quad\quad\quad\quad \text{ELSE } \langle fm1[args],\ fq[args] \rangle
\end{array}
$$

BY $\langle 3 \rangle 2$ DEF g, $fx2$, $fm2$

$\langle 3 \rangle 4.$ $\vee \neg \wedge m2 = \langle r \rangle$

$$
\begin{array}{l}
\quad\quad \wedge\ \Box \exists\, a,\ b :\ \ m2' = \langle a,\ b \rangle \\
\vee\ \Box (m2' \neq \langle r \rangle)
\end{array}
$$

OBVIOUS

$\langle 3 \rangle 5.$ $\vee \neg A$

$$
\begin{array}{l}
\vee\ \exists\, m2,\ q,\ m : \\
\quad \text{LET} \\
\quad\quad argsi \triangleq \langle x,\ y,\ q,\ m \rangle \\
\quad\quad args \triangleq \langle x,\ y,\ q,\ m \rangle \\
\quad \text{IN} \\
\quad\quad \wedge\ I(x,\ y,\ q) \\
\quad\quad \wedge\ q \ = g(x,\ y) \\
\quad\quad \wedge\ m = r \\
\quad\quad \wedge\ \Box \wedge q' \ = m2[2]' \\
\quad\quad\quad\quad \wedge\ m' = m2[1]' \\
\quad\quad \wedge\ m2 = \langle r \rangle \\
\quad\quad \wedge\ \Box \wedge x' = \text{IF } m2 = \langle r \rangle \\
\quad\quad\quad\quad\quad\quad \text{THEN } fx1[argsi]
\end{array}
$$

$$
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{ELSE } \ fx1[args] \\
\qquad\qquad\qquad\qquad\qquad\qquad \wedge\ m2' = \text{IF } \ m2 = \langle r \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{THEN } \ \langle fm1[argsi],\ fq[argsi] \rangle \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{ELSE } \ \langle fm1[args],\ fq[args] \rangle \\
\qquad\quad \text{BY } \langle 3 \rangle 3,\ \langle 3 \rangle 4 \\
\quad \langle 3 \rangle 6.\ \vee \neg A \\
\qquad\quad \vee\ \exists\, m2,\ q,\ m : \\
\qquad\qquad \text{LET} \\
\qquad\qquad\qquad args \ \triangleq\ \langle x,\ y,\ q,\ m \rangle \\
\qquad\qquad \text{IN} \\
\qquad\qquad\qquad \wedge\ I(x,\ y,\ q) \\
\qquad\qquad\qquad \wedge\ m = r \\
\qquad\qquad\qquad \wedge\ \Box \wedge q' \ = m2[2]' \\
\qquad\qquad\qquad\qquad \wedge\ m' = m2[1]' \\
\qquad\qquad\qquad \wedge\ \Box \wedge x' \ = fx1[args] \\
\qquad\qquad\qquad\qquad \wedge\ m2' = \langle fm1[args],\ fq[args] \rangle \\
\qquad\quad \text{BY } \langle 3 \rangle 5 \\
\quad \langle 3 \rangle 7.\ \vee \neg A \\
\qquad\quad \vee\ \exists\, q,\ m : \\
\qquad\qquad \text{LET} \\
\qquad\qquad\qquad args \ \triangleq\ \langle x,\ y,\ q,\ m \rangle \\
\qquad\qquad \text{IN} \\
\qquad\qquad\qquad \wedge\ I(x,\ y,\ q) \\
\qquad\qquad\qquad \wedge\ m = r \\
\qquad\qquad\qquad \wedge\ \Box \wedge x' \ = fx1[args] \\
\qquad\qquad\qquad\qquad \wedge\ q' \ = fq[args] \\
\qquad\qquad\qquad\qquad \wedge\ m' = fm1[args] \\
\qquad\quad \text{BY } \langle 3 \rangle 6 \\
\quad \langle 3 \rangle \ \text{QED} \\
\qquad\quad \text{BY } \langle 3 \rangle 7 \ \ \text{DEF } \ A \\
\langle 2 \rangle \ \text{QED} \\
\qquad \text{BY } \langle 1 \rangle 2,\ \langle 2 \rangle 1 \\
\langle 1 \rangle 4.\ \exists\, fx,\ fm,\ m0 : \\
\qquad \wedge\ IsAFunction(fx) \\
\qquad \wedge\ IsAFunction(fm) \\
\qquad \wedge\ \forall\, x,\ y : \\
\qquad\qquad \vee\ \neg \exists\, m : \\
\qquad\qquad\quad \text{LET} \\
\qquad\qquad\qquad args \ \triangleq\ \langle x,\ y,\ m \rangle \\
\qquad\qquad\quad \text{IN} \\
\qquad\qquad\qquad \wedge\ Init(x,\ y) \\
\qquad\qquad\qquad \wedge\ m = m0 \\
\qquad\qquad\qquad \wedge\ \Box \wedge x' \ = fx[args] \\
\qquad\qquad\qquad\qquad \wedge\ m' = fm[args] \\
\qquad\qquad \vee\ \exists\, q : \ \ Phi(x,\ y,\ q)
\end{array}
$$

```
    ⟨2⟩1. ∧ IsAFunction(fx2)
          ∧ IsAFunction(fm2)
        BY DEF fx2, fm2
    ⟨2⟩ QED
        BY ⟨2⟩1, ⟨1⟩3 DEF Init
⟨1⟩ QED
    BY ⟨1⟩4 DEF IsRealizable
```

Revealing history-determined variables leaves realizability unchanged.

Caution:

1. The next value y; of the environment variable y should not occur in *fnext* if we want a *Moore* implementation.
2. h should be history-determined by functions. Functions instead of operators are necessary for a straightforward proof that a function that controls the value of h does exist.

   Otherwise we would have to argue in terms of what values are relevant to realizability, which is complicated, and likely requires reasoning outside the object language (an independence-like proof).

```
PROPOSITION UnhidingHistoryFuncPreservesRealizability ≜
    ASSUME
        CONSTANT finit, CONSTANT fnext,
        CONSTANT Init(_, _),
        TEMPORAL Phi(_, _, _),
        ∧ LET
                PhiH(x, y) ≜ ∃ h : Phi(x, y, h)
          IN
                IsRealizable(Init, PhiH)
        ∧ LET
                History(h, x, y) ≜
                    ∧ h = finit[x, y]
                    ∧ □(h' = fnext[h, x, y, x'])
          IN
                ∀ x, y, h : Phi(x, y, h) ⇒ History(h, x, y)
    PROVE
        LET I(x, y, h) ≜ Init(x, y) ∧ (h = finit[x, y])
        IN  IsRealizable(I, Phi)
PROOF
⟨1⟩ DEFINE
    I(x, y, h) ≜ Init(x, y) ∧ (h = finit[x, y])
    History(h, x, y) ≜ ∧ h = finit[x, y]
                       ∧ □(h' = fnext[h, x, y, x'])
⟨1⟩1. PICK fx, fm, m0 :
        ∧ IsAFunction(fx)
        ∧ IsAFunction(fm)
```

```
∧  ∀ x, y :
      ∨ ¬∃ m :
          LET
               args ≜ ⟨x, y, m⟩
          IN
               ∧ Init(x, y)
               ∧ m = m0
               ∧ □ ∧ x′  = fx[args]
                   ∧ m′ = fm[args]
      ∨ ∃ h :  Phi(x, y, h)
BY  DEF IsRealizable
        and UnhidingHistoryFuncPreservesRealizability!assumption
⟨1⟩2. ∀ x, y :
        ∨ ¬∃ m :  Realization(Init, m0, fx, fm)
        ∨ ∃ q :  Phi(x, y, q)
  BY ⟨1⟩1
⟨1⟩3. ∀ x, y, h :
        ∨ ¬ ∧ ∃ m :  Realization(Init, m0, fx, fm)
            ∧ History(h, x, y)
        ∨  ∧ ∃ q :  Phi(x, y, q)
           ∧ History(h, x, y)
  BY ⟨1⟩2
⟨1⟩4. ∀ x, y, h :
        ∨ ¬∃ m :  ∧ Realization(Init, m0, fx, fm)
                  ∧ History(h, x, y)
        ∨ Phi(x, y, h)
  ⟨2⟩1. ∀ x, y, q :  Phi(x, y, q)  ⇒  History(q, x, y)
      OBVIOUS   BY UnhidingHistoryFuncPreservesRealizability!assumption
  ⟨2⟩2. ∀ x, y, h :
          ∨ ¬∃ m :  ∧ Realization(Init, m0, fx, fm)
                    ∧ History(h, x, y)
          ∨ ∃ q :  ∧ Phi(x, y, q)
                   ∧ History(q, x, y)
                   ∧ History(h, x, y)
    BY ⟨1⟩3, ⟨2⟩1
  ⟨2⟩3. ∀ x, y, q, h :
          ∨ ¬ ∧ History(q, x, y)
              ∧ History(h, x, y)
          ∨ □(q = h)
    ⟨3⟩1. ASSUME VARIABLE x, VARIABLE y, VARIABLE h, VARIABLE q
          PROVE
                ∧ History(q, x, y) ≡
                      ∧ q = finit[x, y]
                      ∧ □(q′ = fnext[q, x, y, x′]
                ∧ History(h, x, y) ≡
```

$\wedge\ h = \mathit{finit}[x, y]$
$\wedge\ \Box(h' = \mathit{fnext}[h, x, y, x'])$
BY DEF *History*

⟨3⟩ DEFINE
$H \triangleq \mathit{History}(q, x, y) \wedge \mathit{History}(h, x, y)$
$\mathit{Inv} \triangleq q = h$
$\mathit{Next}(u) \triangleq u' = \mathit{fnext}[u, x, y, x']$

⟨3⟩2. $H \Rightarrow \mathit{Inv}$
⟨4⟩1. $H \Rightarrow (q = \mathit{finit}[x, y])$
BY ⟨3⟩1 DEF H
⟨4⟩2. $H \Rightarrow (h = \mathit{finit}[x, y])$
BY ⟨3⟩1 DEF H
⟨4⟩ QED
BY ⟨4⟩1, ⟨4⟩2 DEF *Inv*

⟨3⟩3. $(\mathit{Inv} \wedge \mathit{Next}(q) \wedge \mathit{Next}(h)) \Rightarrow \mathit{Inv}'$
⟨4⟩1. $\mathit{Inv} \Rightarrow (\langle q, x, y, x' \rangle = \langle h, x, y, x' \rangle)$
BY DEF *Inv*
⟨4⟩2. $\vee\ \neg \wedge \langle q, x, y, x' \rangle = \langle h, x, y, x' \rangle$
$\qquad \wedge\ \mathit{Next}(q) \wedge \mathit{Next}(h)$
$\vee\ q' = h'$
BY DEF *Next*
⟨4⟩3. $(q' = h') \equiv \mathit{Inv}'$
BY DEF *Inv*
⟨4⟩ QED
BY ⟨4⟩1, ⟨4⟩2, ⟨4⟩3

⟨3⟩ QED
BY ⟨3⟩1, ⟨3⟩2, ⟨3⟩3, *RuleRawINV1*

⟨2⟩4. $\forall x, y, h :$
$\vee\ \neg \exists m : \wedge\ \mathit{Realization}(\mathit{Init}, m0, \mathit{fx}, \mathit{fm})$
$\qquad \wedge\ \mathit{History}(h, x, y)$
$\vee\ \exists q :$
$\wedge\ \mathit{Phi}(x, y, q)$
$\wedge\ \Box(q = h)$
$\wedge\ \mathit{History}(q, x, y)$
$\wedge\ \mathit{History}(h, x, y)$
BY ⟨2⟩2, ⟨2⟩3

in effect flexible substitution

⟨2⟩5. $\forall x, y, h :$
$\vee\ \neg \exists m : \wedge\ \mathit{Realization}(\mathit{Init}, m0, \mathit{fx}, \mathit{fm})$
$\qquad \wedge\ \mathit{History}(h, x, y)$
$\vee\ \exists q :$
$\wedge\ \mathit{Phi}(x, y, h)$
$\wedge\ \Box(q = h)$
BY ⟨2⟩4

⟨2⟩ QED

BY $\langle 2\rangle 5$

$\langle 1\rangle 5.\ \forall x, y, h :$

$(\exists m : \land Realization(Init, m0, fx, fm)$
$\qquad\qquad \land History(h, x, y))$

$\equiv \exists m :$

LET

$args \triangleq \langle x, y, m\rangle$

IN

$\land Init(x, y)$
$\land h = finit[x, y]$
$\land m = m0$
$\land \Box \land x' = fx[args]$
$\qquad \land h' = fnext[h, x, y, x']$
$\qquad \land m' = fm[args]$

BY $\langle 1\rangle 1$ DEF $Realization, History$

$\langle 1\rangle$ DEFINE

$Dom \triangleq (\text{DOMAIN } fx) \cup (\text{DOMAIN } fm)$
$Proj(T, i) \triangleq \{t[i] : t \in T\}$
$DomX \triangleq Proj(Dom, 1) \cup Proj(\text{DOMAIN } fnext, 2)$
$DomY \triangleq Proj(Dom, 2) \cup Proj(\text{DOMAIN } fnext, 3)$
$DomM \triangleq Proj(Dom, 3)$
$DomH \triangleq Proj(\text{DOMAIN } fnext, 1)$

repacking

$S \triangleq (DomX \times DomH) \times DomY \times DomM$
$F(f, t) \triangleq \text{LET } x \triangleq t[1][1]\ y \triangleq t[2]\ m \triangleq t[3]$
$\qquad \text{IN } f[x, y, m]$
$G(f, t) \triangleq \text{LET } x \triangleq t[1][1]\ h \triangleq t[1][2]\ y \triangleq t[2]\ m \triangleq t[3]$
$\qquad \text{IN } f[h, x, y, fx[x, y, m]]$
$fx2 \triangleq [t \in S \mapsto F(fx, t)]$
$fm2 \triangleq [t \in S \mapsto F(fm, t)]$
$fh2 \triangleq [t \in S \mapsto G(fnext, t)]$

$\langle 1\rangle 6.\ \forall x, y, h :$

$(\exists m : \land Realization(Init, m0, fx, fm)$
$\qquad\qquad \land History(h, x, y))$

$\equiv \exists m :$

LET

$args \triangleq \langle\langle h, x\rangle, y, m\rangle$

IN

$\land I(x, y, h)$
$\land m = m0$
$\land \Box \land x' = fx2[args]$
$\qquad \land h' = fh2[args]$
$\qquad \land m' = fm2[args]$

BY $\langle 1\rangle 5$ DEF $I, fx2, fm2, fh2$

$\langle 1\rangle$ QED

BY $\langle 1 \rangle 4$, $\langle 1 \rangle 6$ DEF *IsRealizable* DEF I

Combining the two previous directions into one theorem.

THEOREM *RealizingHistory* $\triangleq$

ASSUME

CONSTANT *finit*, CONSTANT *fnext*,

CONSTANT $Init(\_, \_)$,

TEMPORAL $Phi(\_, \_, \_)$,

LET

$$History(h, x, y) \triangleq \land\ h = finit[x, y] \quad \land\ \Box(h' = fnext[h, x, y, x'])$$

IN

$$\forall\, x, y, h : Phi(x, y, h) \Rightarrow History(h, x, y)$$

PROVE

LET

$$I(x, y, h) \triangleq Init(x, y) \land (h = finit[x, y])$$

$$PhiH(x, y) \triangleq \exists\, h : Phi(x, y, h)$$

IN

$$IsRealizable(I, Phi) \equiv IsRealizable(Init, PhiH)$$

PROOF

BY *HidingHistoryPreservesRealizability*, *UnhidingHistoryFuncPreservesRealizability*

MODULE *Representation*

A safety formula $\Box Next$ in $RTLA+$ can be unsatisfiable even when $Next$ is.
This cannot happen with the TLA+ formula $\Box[Next]\_v$, because deadends cannot form. Deadends return when conjoining a liveness formula to $\Box[Next]\_v$.

In other words, there is no such thing as an unsatisfiable TLA+ formula of the form $\Box[Next]\_v$ (or $Init \land \Box[Next]\_v$ whenever $Init$ is satisfiable).

Conjoining an initial condition $Init$ to $\Box[Next]\_v$ preserves information present in $Init$ and $Next$ (at least that information which is essential when taking steps forward, which is what matters for $RawWhilePlus$).

Conjoining a livevess formula to the safety formula $Init \land \Box[Next]\_v$ destroys information, in the sense that the resulting property is representable by multiple canonical formulas. Among these canonical formulas are some whose subformulas $Init$, $Next$, Liveness lead to different $RawWhilePlus$ properties.

Author: Ioannis *Filippidis*

References

---

[1] *L.* Lamport, "Proving possibility properties", *TCS*, 1998 10.1016/*S*0304-3975(98)00129-7

EXTENDS *TemporalLogic*, *TLASemantics*

Any safety property is machine-closed with respect to TRUE [1, Prop.3].

PROPOSITION
  ASSUME
    STATE $Init$,
    ACTION $Next$
  PROVE
    $Cl(Init \land \Box[Next]_v \land \Box\Diamond \text{TRUE})$
    $\equiv Init \land \Box[Next]_v$
  PROOF
  ⟨1⟩ DEFINE
    $A \triangleq Init \land \Box[Next]_v \land \Box\Diamond \text{TRUE}$
    $B \triangleq Init \land \Box[Next]_v$
  ⟨1⟩1. $\text{TRUE} \equiv \Box\Diamond \text{TRUE}$
    BY PTL
  ⟨1⟩2. $A \equiv B$
    BY ⟨1⟩1
  ⟨1⟩3. $Cl(A) \equiv Cl(B)$
    BY ⟨1⟩2
  ⟨1⟩4. $Cl(B) \equiv B$
    B is a safety property.
  ⟨1⟩ QED

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

RTLA+ results.

In *RTLA*+ a weaker action yields weaker safety.

PROPOSITION $WeakerActionRTLA \triangleq$
ASSUME
ACTION $A1$, ACTION $A2$,
$A1 \Rightarrow A2$
PROVE
$(\Box A1) \Rightarrow (\Box A2)$
PROOF
$\langle 1 \rangle 1.\ \forall\, s1, s2 : (IsAState(s1) \land IsAState(s2)) \Rightarrow$
$\langle s1, s2 \rangle[[A1]] \Rightarrow \langle s1, s2 \rangle[[A2]]$
$\langle 1 \rangle 2.$ SUFFICES
ASSUME
NEW $sigma$, $IsABehavior(sigma)$,
$sigma \models \Box A1$
PROVE
$sigma \models \Box A2$
$\langle 1 \rangle 3.$ ASSUME NEW $n \in Nat$
PROVE $\langle sigma[n], sigma[n+1] \rangle[[A1]]$
BY $\langle 1 \rangle 2$
$\langle 1 \rangle 4.$ ASSUME NEW $n \in Nat$
PROVE $\langle sigma[n], sigma[n+1] \rangle[[A2]]$
$\langle 2 \rangle 1.\ \land IsAState(sigma[n])$
$\land IsAState(sigma[n+1])$
BY $\langle 1 \rangle 2$, $\langle 1 \rangle 4$ DEF $IsABehavior$
$\langle 2 \rangle$ QED
BY $\langle 1 \rangle 1$, $\langle 1 \rangle 3$, $\langle 2 \rangle 1$
$\langle 1 \rangle$ QED
BY $\langle 1 \rangle 2$, $\langle 1 \rangle 4$

In *RTLA*+ equal actions yield same safety.

COROLLARY $EquivActionsRTLA \triangleq$
ASSUME
ACTION $A1$, ACTION $A2$,
$A1 \equiv A2$
PROVE
$(\Box A1) \equiv (\Box A2)$
PROOF
$\langle 1 \rangle 1.\ (\Box A1) \Rightarrow (\Box A2)$
BY $WeakerActionRTLA$

$\langle 1 \rangle 2.\ (\Box A2) \;\Rightarrow\; (\Box A1)$

  BY *WeakerActionRTLA*

$\langle 1 \rangle$ QED

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

The converse of the previous proposition does not hold, due to deadends.
This *RTLA+* fact corresponds in TLA+ to the multiplicity of representations of a property as a conjunction of safety and liveness.

PROPOSITION *RawTails* $\triangleq$

  PROVE

    LET

      $A1 \triangleq$ FALSE

      $A2 \triangleq (x = 1) \land (x' = 2)$

    IN

      $\land (\Box A1) \;\equiv\; (\Box A2)$

      $\land \neg \models A1 \equiv A2$

  PROOF

  $\langle 1 \rangle$ DEFINE

    $A1 \triangleq$ FALSE

    $A2 \triangleq (x = 1) \land (x' = 2)$

  $\langle 1 \rangle 1.\ \neg \models A1 \equiv A2$

    $\langle 2 \rangle 1.$ SUFFICES

      $\exists\, s,\, t : \;\land IsAState(s)$

      $\qquad\quad\;\; \land IsAState(t)$

      $\qquad\quad\;\; \land \langle s,\, t \rangle [[A2 \land \neg A1]]$

    $\langle 2 \rangle 2.$ PICK $s : \; IsAState(s) \land s[[x]] = 1$

    $\langle 2 \rangle 3.$ PICK $t : \; IsAState(t) \land t[[t]] = 2$

    $\langle 2 \rangle$ QED

      BY $\langle 2 \rangle 2$, $\langle 2 \rangle 3$ DEF $A1$, $A2$ goal from $\langle 2 \rangle 1$

  $\langle 1 \rangle 2.\ (\Box A1) \;\equiv\; (\Box A2)$

    $\langle 2 \rangle 1.\ (\Box A1) \;\Rightarrow\; (\Box A2)$

      $\langle 3 \rangle 1.\ \neg(\Box A1)$

        BY DEF $A1$

      $\langle 3 \rangle$ QED

        BY $\langle 3 \rangle 1$, *PTL*

    $\langle 2 \rangle 2.\ (\Box A2) \;\Rightarrow\; (\Box A1)$

      $\langle 3 \rangle 1.$ SUFFICES $\neg \Box A2$ *A2* leads to a deadend.

      $\langle 3 \rangle 2.$ SUFFICES

        ASSUME $\exists\, sigma : \;\land IsABehavior(sigma)$

        $\qquad\qquad\qquad\quad \land \Box A2$

        PROVE FALSE

      $\langle 3 \rangle 3.$ PICK $sigma : \;\land IsABehavior(sigma)$

      $\qquad\qquad\qquad\;\; \land sigma \models \Box A2$

        BY $\langle 3 \rangle 2$

$\langle 3 \rangle 4.\ \land\ sigma[0][[x]] = 1$
$\quad\ \land\ sigma[1][[x]] = 2$

$\langle 4 \rangle 1.\ \langle sigma[0],\ sigma[1] \rangle [[(x = 1) \land (x' = 2)]]$

BY $\langle 3 \rangle 3$ DEF $A2$

$\langle 4 \rangle$ QED

BY $\langle 4 \rangle 1$

$\langle 3 \rangle 5.\ \neg \langle sigma[1],\ sigma[2] \rangle [[A2]]$

BY $\langle 3 \rangle 4$ DEF $A2$

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 3,\ \langle 3 \rangle 5$ goal from $\langle 3 \rangle 2$

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle$ QED

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

In $RTLA+$ equivalent initial conditions and actions yield the same safety property.

COROLLARY

ASSUME

STATE $I1$, STATE $I2$,

ACTION $A1$, ACTION $A2$,

$I1 \equiv I2$,

$A1 \equiv A2$

PROVE

$(I1 \land \Box A1) \equiv (I2 \land \Box A2)$

PROOF

$\langle 1 \rangle 1.\ (\Box A1) \equiv (\Box A2)$

BY $EquivActionsRTLA$

$\langle 1 \rangle 2.\ (I1 \land \Box A1) \equiv (I1 \land \Box A2)$

BY $\langle 1 \rangle 1$

$\langle 1 \rangle 3.\ I1 \equiv I2$

OBVIOUS

$\langle 1 \rangle$ QED

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

TLA+ results.

Similar to the previous corollary, but in TLA+.

PROPOSITION

ASSUME

STATE $I1$, STATE $I2$, STATE v,

ACTION $A1$, ACTION $A2$,

$I1 \equiv I2$,

$A1 \equiv A2$

PROVE

$(I1 \land \Box[A1]_v) \;\equiv\; (I2 \land \Box[A2]_v)$

OBVIOUS

Two equivalent tails are defined by actions with equivalent nonstuttering parts.

PROPOSITION $InvertingTails \triangleq$

ASSUME

STATE v,

ACTION $A1$, ACTION $A2$,

$(\Box[A1]_v) \;\equiv\; (\Box[A2]_v)$

PROVE

$\langle A1 \rangle_v \;\equiv\; \langle A2 \rangle_v$

PROOF

$\langle 1 \rangle 1.$ SUFFICES

ASSUME

NEW $s1$, NEW $s2$, $IsAState(s1)$, $IsAState(s2)$,

$\land \langle s1,\, s2 \rangle [[\langle A1 \rangle_v]]$

$\land \neg \langle s1,\, s2 \rangle [[\langle A2 \rangle_v]]$

PROVE FALSE

BY $Semantics$

$\langle 1 \rangle$ DEFINE $sigma \triangleq [n \in Nat \mapsto$ IF $n = 0$ THEN $s1$ ELSE $s2]$

$\langle 1 \rangle 2.$ $IsABehavior(sigma)$

BY DEF $sigma$, $IsABehavior$

$\langle 1 \rangle 3.$ $sigma \models \Box[A1]_v$

$\langle 2 \rangle 1.$ $\langle A \rangle_v \;\Rightarrow\; [A]_v$

OBVIOUS

$\langle 2 \rangle$ QED

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

$\langle 1 \rangle 4.$ $\neg(sigma \models \Box[A2]_v)$

$\langle 2 \rangle 1.$ $sigma[0] \neq sigma[1]$

$\langle 3 \rangle 1.$ $\langle sigma[0],\, sigma[1] \rangle [[\langle A1 \rangle_v]]$

BY $\langle 1 \rangle 1$ DEF $sigma$

$\langle 3 \rangle$ QED

BY $\langle 3 \rangle 1$

$\langle 2 \rangle 2.$ $\neg \langle sigma[0],\, sigma[1] \rangle [[v' = v]]$

BY $\langle 1 \rangle 3$

$\langle 2 \rangle 3.$ $\neg \langle sigma[0],\, sigma[1] \rangle [[\langle A1 \rangle_v \lor (v' = v)]]$

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

$\langle 2 \rangle 4.$ $\neg \langle sigma[0],\, sigma[1] \rangle [[[A1]_v]]$

$\langle 3 \rangle 1.$ $((v' = v) \land A1) \;\Rightarrow\; (v' = v)$

$\langle 3 \rangle 2.$ $A1 \equiv \lor A1 \land (v' = v)$

$\qquad\qquad \lor \langle A1 \rangle_v$

$\langle 3 \rangle 3.\ [A1]_v \equiv\ \lor \langle A1 \rangle_v$
$\lor\ v' = v$

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

$\langle 3 \rangle$ QED

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

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 4$

$\langle 1 \rangle 5.\ (sigma \models \Box[A1]_v)\ \equiv\ (sigma \models \Box[A2]_v)$

BY $\langle 1 \rangle 2$

$\langle 1 \rangle$ QED

$\langle 1 \rangle 3$, $\langle 1 \rangle 4$, $\langle 1 \rangle 5$ goal from $\langle 1 \rangle 1$

LEMMA *BoxActionEnabled* $\triangleq$

ASSUME

STATE v, ACTION A

PROVE

ENABLED $[A]_v$

$\langle 1 \rangle 1.\ [A]_v \equiv (A \lor (v' = v))$

OBVIOUS

$\langle 1 \rangle 2.$ ENABLED $(v' = v)$

$\langle 2 \rangle 1.$ SUFFICES

ASSUME NEW $s1$, $IsAState(s1)$

PROVE $\exists\, s2 : \ \land IsAState(s2)$
$\land \langle s1, s2 \rangle [[v' = v]]$

OBVIOUS

$\langle 2 \rangle$ DEFINE $s2 \triangleq s1$

$\langle 2 \rangle 3.\ IsAState(s2)$

BY $\langle 2 \rangle 1$ DEF $s2$

$\langle 2 \rangle 4.\ s2[[v]] = s1[[v]]$

BY $\langle 2 \rangle 1$, $\langle 2 \rangle 3$ DEF $s2$

$\langle 2 \rangle 5.\ \langle s1, s2 \rangle [[v' = v]]$

BY $\langle 2 \rangle 4$

$\langle 2 \rangle$ QED

BY $\langle 2 \rangle 3$, $\langle 2 \rangle 5$

$\langle 1 \rangle 3.$ ASSUME ACTION P, ACTION Q

PROVE (ENABLED P) $\Rightarrow$ ENABLED $(P \lor Q)$

OBVIOUS

$\langle 1 \rangle 4.$ ENABLED $(A \lor (v' = v))$

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

$\langle 1 \rangle$ QED

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

In the presence of an initial condition, the actions of two state machines are equivalent only at reachable states, but may differ elsewhere.

PROPOSITION *InvertingStateMachines* $\triangleq$

    ASSUME
        STATE $I1$, STATE $I2$, STATE v,
        ACTION $A1$, ACTION $A2$,
        LET
            $SM1 \triangleq I1 \land \Box[A1]_v$
            $SM2 \triangleq I2 \land \Box[A2]_v$
        IN
            $SM1 \equiv SM2$
    PROVE
        LET
            $SM1 \triangleq I1 \land \Box[A1]_v$
        IN
            $\land\ I1 \equiv I2$
            $\land\ SM1 \Rightarrow \Box[A1 \land A2]_v$
    ⟨1⟩ DEFINE
        $SM1 \triangleq I1 \land \Box[A1]_v$
        $SM2 \triangleq I2 \land \Box[A2]_v$
    ⟨1⟩1. $SM1 \equiv SM2$
        OBVIOUS BY *InvertingStateMachines*
    ⟨1⟩2. $I1 \equiv I2$
        ⟨2⟩1. $I1 \Rightarrow I2$
            ⟨3⟩1. SUFFICES
                    ASSUME NEW s, $IsAState(s)$, $s[[I1]]$
                    PROVE $s[[I2]]$
                BY STATE $I1$, STATE $I2$
            ⟨3⟩ DEFINE $sigma \triangleq Stutter(s)$
            ⟨3⟩2. $IsABehavior(sigma)$
                BY ⟨3⟩1 DEF $signa$, $Stutter$, $IsABehavior$
            ⟨3⟩3. $sigma \models SM1$
                ⟨4⟩1. $sigma \models I1$
                    ⟨5⟩1. $sigma[0] = s$
                        BY DEF $sigma$, $Stutter$
                    ⟨5⟩2. $s[[I1]]$
                        BY ⟨3⟩1
                    ⟨5⟩ QED
                        BY ⟨5⟩1, ⟨5⟩2
                ⟨4⟩2. $sigma \models \Box[A1]_v$
                    ⟨5⟩1. $sigma \models \Box[\text{FALSE}]_v$
                        BY DEF $sigma$, $Stutter$
                    ⟨5⟩ QED
                        BY ⟨4⟩3
                ⟨4⟩ QED
                    BY ⟨4⟩1, ⟨4⟩2 DEF $SM1$
            ⟨3⟩4. $sigma \models SM2$

BY ⟨3⟩3, ⟨1⟩1

⟨3⟩5. $sigma \models I2$

BY ⟨3⟩4 DEF $SM2$

⟨3⟩ QED

⟨4⟩1. $sigma[0][[I2]]$

BY ⟨3⟩5, ⟨3⟩2 and STATE $I2$

⟨4⟩2. $sigma[0] = s$

BY DEF $sigma$, $Stutter$

⟨4⟩ QED

BY ⟨4⟩1, ⟨4⟩2

⟨2⟩2. $I2 \Rightarrow I1$

PROOF similar to that of ⟨2⟩1.

⟨2⟩ QED

BY ⟨2⟩1, ⟨2⟩2

⟨1⟩3. $SM1 \;\Rightarrow\; (SM1 \land SM2)$

BY ⟨1⟩1

⟨1⟩4. $SM1 \;\Rightarrow\; (\Box[A1]_v \land \Box[A2]_v)$

BY ⟨1⟩3 DEF $SM1$, $SM2$

⟨1⟩5. $SM1 \;\Rightarrow\; \Box[A1 \land A2]_v$

BY ⟨1⟩4

⟨1⟩ QED

BY ⟨1⟩2, ⟨1⟩5

MODULE *StepComparison*

Comparison of the strictly causal and causal controllable step operators.

Author: Ioannis *Filippidis*

```
CONSTANT SysNext(_, _, _), EnvNext(_, _, _), Target(_, _)

  SysNext by syntax is independent of u, so of x'
Step(x, y) ≜
    ∃ v : ∀ u :
        ∧ SysNext(x, y, v)
        ∧ EnvNext(x, y, u) ⇒ Target(u, v)

StepU(x, y) ≜
    ∃ v : ∀ u :
        EnvNext(x, y, u) ⇒ ∧ SysNext(x, y, v)
                           ∧ Target(u, v)

THEOREM
    ASSUME
        VARIABLE x, VARIABLE y
    PROVE
        Step(x, y) ≡ ∧ ∃ v : SysNext(x, y, v)
                     ∧ StepU(x, y)
    BY DEF Step, StepU

  Detailed proof because it is instructive.
THEOREM SameThmWithDetailedProof ≜
    ASSUME
        VARIABLE x, VARIABLE y
    PROVE
        Step(x, y) ≡ ∧ ∃ v : SysNext(x, y, v)
                     ∧ StepU(x, y)
PROOF
    ⟨1⟩ DEFINE
        A(u, v) ≜
            ∧ SysNext(x, y, v)
            ∧ EnvNext(x, y, u) ⇒ Target(u, v)
        B(u, v) ≜
            ∧ SysNext(x, y, v)
            ∧ EnvNext(x, y, u) ⇒ ∧ SysNext(x, y, v)
                                 ∧ Target(u, v)
```

$F \triangleq \exists v : \forall u : B(u, v)$
$EnabledEnv \triangleq \exists u : EnvNext(x, y, u)$
$EnabledSys \triangleq \exists v : SysNext(x, y, v)$

$\langle 1 \rangle 1.$ $\land F \equiv \exists v : \forall u : B(u, v)$
$\quad \land Step(x, y) \equiv \exists v : \forall u : A(u, v)$
  BY DEF $A, B, F, Step$

$\langle 1 \rangle 2.$ $Step(x, y) \equiv F$
  $\langle 2 \rangle 1.$ SUFFICES ASSUME NEW u, NEW v
    PROVE $A(u, v) \equiv B(u, v)$
    BY $\langle 2 \rangle 1, \langle 1 \rangle 1$
  $\langle 2 \rangle$ QED
    BY DEF A, B

$\langle 1 \rangle 3.$ $Step(x, y) \equiv$
$\quad \exists v : \land SysNext(x, y, v)$
$\quad\quad \land \forall u : EnvNext(x, y, u) \Rightarrow \land SysNext(x, y, v)$
$\quad\quad\quad \land Target(u, v)$
  BY $\langle 1 \rangle 2$ DEF F

$\langle 1 \rangle 4.$ $Step(x, y) \Rightarrow \land EnabledSys$
$\quad \land StepU(x, y)$
  $\langle 2 \rangle 1.$ $Step(x, y) \Rightarrow \exists v : SysNext(x, y, v)$
    BY $\langle 1 \rangle 3$
  $\langle 2 \rangle 2.$ $Step(x, y) \Rightarrow StepU(x, y)$
    BY $\langle 1 \rangle 3$ DEF $StepU$
  $\langle 2 \rangle$ QED
    BY $\langle 2 \rangle 1, \langle 2 \rangle 2$ DEF $EnabledSys$

$\langle 1 \rangle 5.$ $(EnabledSys \land StepU(x, y)) \Rightarrow Step(x, y)$
  $\langle 2 \rangle 1.$CASE $\neg EnabledEnv$
    $\langle 3 \rangle 1.$ $EnabledSys \Rightarrow \exists v : \land SysNext(x, y, v)$
      BY DEF $EnabledSys$
    $\langle 3 \rangle 2.$ $\forall v : \forall u : EnvNext(x, y, u) \Rightarrow \land SysNext(x, y, v)$
$\quad\quad \land Target(u, v)$
      BY $\langle 2 \rangle 1$
    $\langle 3 \rangle 3.$ $EnabledSys \Rightarrow$
$\quad \exists v : \land SysNext(x, y, v)$
$\quad\quad \land \forall u : EnvNext(x, y, u) \Rightarrow \land SysNext(x, y, v)$
$\quad\quad\quad \land Target(u, v)$
      BY $\langle 3 \rangle 1, \langle 3 \rangle 2$
    $\langle 3 \rangle$ QED
      BY $\langle 3 \rangle 3, \langle 1 \rangle 3$
  $\langle 2 \rangle 2.$CASE $EnabledEnv$
    $\langle 3 \rangle 1.$ SUFFICES ASSUME $EnabledSys \land StepU(x, y)$
      PROVE $Step(x, y)$
      OBVIOUS
    $\langle 3 \rangle 2.$ PICK $v : \forall u : EnvNext(x, y, u) \Rightarrow \land SysNext(x, y, v)$
$\quad\quad \land Target(u, v)$

BY $\langle 3\rangle 1$ DEF *StepU*
$\langle 3\rangle 3.$ $SysNext(x,\ y,\ v)$
$\langle 4\rangle 1.$ PICK $r:\ EnvNext(x,\ y,\ r)$
BY $\langle 2\rangle 2$ DEF *EnabledEnv*
$\langle 4\rangle 2.$ $EnvNext(x,\ y,\ r) \Rightarrow SysNext(x,\ y,\ v)$
BY $\langle 3\rangle 2$
$\langle 4\rangle$ QED
BY $\langle 4\rangle 1$, $\langle 4\rangle 2$
$\langle 3\rangle$ QED
BY $\langle 3\rangle 2$, $\langle 3\rangle 3$ DEF *Step*
$\langle 2\rangle$ QED
BY $\langle 2\rangle 1$, $\langle 2\rangle 2$
$\langle 1\rangle$ QED
BY $\langle 1\rangle 4$, $\langle 1\rangle 5$

EnvNext here depends on *y*'

THEOREM
ASSUME
CONSTANT $EnvNextR(\_,\ \_,\ \_,\ \_)$,
VARIABLE x, VARIABLE y
PROVE
LET

$$StepR(x,\ y) \triangleq \exists v:\ \forall u: \land SysNext(x,\ y,\ v) \land EnvNextR(x,\ y,\ u,\ v) \Rightarrow Target(u,\ v)$$

$$StepUR(x,\ y) \triangleq \exists v:\ \forall u: EnvNextR(x,\ y,\ u,\ v) \Rightarrow \land SysNext(x,\ y,\ v) \land Target(u,\ v)$$

IN

$$StepR(x,\ y) \equiv \land \exists v:\ SysNext(x,\ y,\ v) \land StepUR(x,\ y)$$