Contents

1 PBTypeIntegrated Theory

Built: 16 May 2018
Parent Theories: OMNIType

1.1 Datatypes

omniCommand = ssmPlanPBComplete | ssmMoveToORPComplete
| ssmConductORPComplete | ssmMoveToPBComplete
| ssmConductPBComplete | invalidOmniCommand

plCommand = crossLD | conductORP | moveToPB | conductPB
| completePB | incomplete

slCommand = PL plCommand | OMNI omniCommand

slOutput = PlanPB | MoveToORP | ConductORP | MoveToPB
| ConductPB | CompletePB | unAuthenticated
| unAuthorized

slState = PLAN_PB | MOVE_TO_ORP | CONDUCT_ORP | MOVE_TO_PB
| CONDUCT_PB | COMPLETE_PB

stateRole = PlatoonLeader | Omni

1.2 Theorems

[omniCommand_distinct_clauses]

⊢ ssmPlanPBComplete ≠ ssmMoveToORPComplete ∧
ssmPlanPBComplete ≠ ssmConductORPComplete ∧
ssmPlanPBComplete ≠ ssmMoveToPBComplete ∧
ssmPlanPBComplete ≠ ssmConductPBComplete ∧
ssmPlanPBComplete ≠ invalidOmniCommand ∧
ssmMoveToORPComplete ≠ ssmConductORPComplete ∧
ssmMoveToORPComplete ≠ ssmMoveToPBComplete ∧
ssmMoveToORPComplete ≠ ssmConductPBComplete ∧
ssmMoveToORPComplete ≠ invalidOmniCommand ∧
ssmConductORPComplete ≠ ssmMoveToPBComplete ∧
ssmConductORPComplete ≠ ssmConductPBComplete ∧
ssmConductORPComplete ≠ invalidOmniCommand ∧
ssmMoveToPBComplete ≠ ssmConductPBComplete ∧
ssmMoveToPBComplete ≠ invalidOmniCommand ∧
ssmConductPBComplete ≠ invalidOmniCommand

[plCommand_distinct_clauses]

⊢ crossLD ≠ conductORP ∧ crossLD ≠ moveToPB ∧
crossLD ≠ conductPB ∧ crossLD ≠ completePB ∧
crossLD ≠ incomplete ∧ conductORP ≠ moveToPB ∧

conductORP ≠ conductPB ∧ conductORP ≠ completePB ∧
conductORP ≠ incomplete ∧ moveToPB ≠ conductPB ∧
moveToPB ≠ completePB ∧ moveToPB ≠ incomplete ∧
conductPB ≠ completePB ∧ conductPB ≠ incomplete ∧
completePB ≠ incomplete

[slCommand_distinct_clauses]

⊢ ∀ a′ a. PL a ≠ OMNI a′

[slCommand_one_one]

⊢ (∀ a a′. (PL a = PL a′) ⟺ (a = a′)) ∧
∀ a a′. (OMNI a = OMNI a′) ⟺ (a = a′)

[slOutput_distinct_clauses]

⊢ PlanPB ≠ MoveToORP ∧ PlanPB ≠ ConductORP ∧
PlanPB ≠ MoveToPB ∧ PlanPB ≠ ConductPB ∧
PlanPB ≠ CompletePB ∧ PlanPB ≠ unAuthenticated ∧
PlanPB ≠ unAuthorized ∧ MoveToORP ≠ ConductORP ∧
MoveToORP ≠ MoveToPB ∧ MoveToORP ≠ ConductPB ∧
MoveToORP ≠ CompletePB ∧ MoveToORP ≠ unAuthenticated ∧
MoveToORP ≠ unAuthorized ∧ ConductORP ≠ MoveToPB ∧
ConductORP ≠ ConductPB ∧ ConductORP ≠ CompletePB ∧
ConductORP ≠ unAuthenticated ∧ ConductORP ≠ unAuthorized ∧
MoveToPB ≠ ConductPB ∧ MoveToPB ≠ CompletePB ∧
MoveToPB ≠ unAuthenticated ∧ MoveToPB ≠ unAuthorized ∧
ConductPB ≠ CompletePB ∧ ConductPB ≠ unAuthenticated ∧
ConductPB ≠ unAuthorized ∧ CompletePB ≠ unAuthenticated ∧
CompletePB ≠ unAuthorized ∧ unAuthenticated ≠ unAuthorized

[slState_distinct_clauses]

⊢ PLAN_PB ≠ MOVE_TO_ORP ∧ PLAN_PB ≠ CONDUCT_ORP ∧
PLAN_PB ≠ MOVE_TO_PB ∧ PLAN_PB ≠ CONDUCT_PB ∧
PLAN_PB ≠ COMPLETE_PB ∧ MOVE_TO_ORP ≠ CONDUCT_ORP ∧
MOVE_TO_ORP ≠ MOVE_TO_PB ∧ MOVE_TO_ORP ≠ CONDUCT_PB ∧
MOVE_TO_ORP ≠ COMPLETE_PB ∧ CONDUCT_ORP ≠ MOVE_TO_PB ∧
CONDUCT_ORP ≠ CONDUCT_PB ∧ CONDUCT_ORP ≠ COMPLETE_PB ∧
MOVE_TO_PB ≠ CONDUCT_PB ∧ MOVE_TO_PB ≠ COMPLETE_PB ∧
CONDUCT_PB ≠ COMPLETE_PB

[stateRole_distinct_clauses]

⊢ PlatoonLeader ≠ Omni

2 ssmPB Theory

Built: 16 May 2018

Parent Theories: PBType, ssm11, OMNIType

2.1 Definitions

[secContext_def]

$\vdash \forall\, cmd\,.$
secContext cmd =
[Name PlatoonLeader controls prop (SOME (SLc cmd))]

[ssmPBStateInterp_def]

$\vdash \forall\, state\,.$ ssmPBStateInterp $state$ = TT

2.2 Theorems

[authenticationTest_cmd_reject_lemma]

$\vdash \forall\, cmd\,.$ ¬authenticationTest (prop (SOME cmd))

[authenticationTest_def]

$\vdash$ (authenticationTest (Name PlatoonLeader says prop cmd) $\iff$
T) $\wedge$ (authenticationTest TT $\iff$ F) $\wedge$
(authenticationTest FF $\iff$ F) $\wedge$
(authenticationTest (prop v) $\iff$ F) $\wedge$
(authenticationTest (notf v_1) $\iff$ F) $\wedge$
(authenticationTest (v_2 andf v_3) $\iff$ F) $\wedge$
(authenticationTest (v_4 orf v_5) $\iff$ F) $\wedge$
(authenticationTest (v_6 impf v_7) $\iff$ F) $\wedge$
(authenticationTest (v_8 eqf v_9) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says TT) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says FF) $\iff$ F) $\wedge$
(authenticationTest (*v133* meet *v134* says prop v_{66}) $\iff$ F) $\wedge$
(authenticationTest (*v135* quoting *v136* says prop v_{66}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says notf v_{67}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{68} andf v_{69})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{70} orf v_{71})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{72} impf v_{73})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{74} eqf v_{75})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{76} says v_{77}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{78} speaks_for v_{79}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{80} controls v_{81}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says reps v_{82} v_{83} v_{84}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{85} domi v_{86}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{87} eqi v_{88}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{89} doms v_{90}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{91} eqs v_{92}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{93} eqn v_{94}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{95} lte v_{96}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{97} lt v_{98}) $\iff$ F) $\wedge$
(authenticationTest (v_{12} speaks_for v_{13}) $\iff$ F) $\wedge$
(authenticationTest (v_{14} controls v_{15}) $\iff$ F) $\wedge$
(authenticationTest (reps v_{16} v_{17} v_{18}) $\iff$ F) $\wedge$

(authenticationTest (v_{19} domi v_{20}) $\iff$ F) $\wedge$
(authenticationTest (v_{21} eqi v_{22}) $\iff$ F) $\wedge$
(authenticationTest (v_{23} doms v_{24}) $\iff$ F) $\wedge$
(authenticationTest (v_{25} eqs v_{26}) $\iff$ F) $\wedge$
(authenticationTest (v_{27} eqn v_{28}) $\iff$ F) $\wedge$
(authenticationTest (v_{29} lte v_{30}) $\iff$ F) $\wedge$
(authenticationTest (v_{31} lt v_{32}) $\iff$ F)

[authenticationTest_ind]

$\vdash \forall P.$
(∀ *cmd*. P (Name PlatoonLeader says prop *cmd*)) $\wedge$ P TT $\wedge$
P FF $\wedge$ (∀ v. P (prop v)) $\wedge$ (∀ v_1. P (notf v_1)) $\wedge$
(∀ v_2 v_3. P (v_2 andf v_3)) $\wedge$ (∀ v_4 v_5. P (v_4 orf v_5)) $\wedge$
(∀ v_6 v_7. P (v_6 impf v_7)) $\wedge$ (∀ v_8 v_9. P (v_8 eqf v_9)) $\wedge$
(∀ v_{10}. P (v_{10} says TT)) $\wedge$ (∀ v_{10}. P (v_{10} says FF)) $\wedge$
(∀ *v133* *v134* v_{66}. P (*v133* meet *v134* says prop v_{66})) $\wedge$
(∀ *v135* *v136* v_{66}. P (*v135* quoting *v136* says prop v_{66})) $\wedge$
(∀ v_{10} v_{67}. P (v_{10} says notf v_{67})) $\wedge$
(∀ v_{10} v_{68} v_{69}. P (v_{10} says (v_{68} andf v_{69}))) $\wedge$
(∀ v_{10} v_{70} v_{71}. P (v_{10} says (v_{70} orf v_{71}))) $\wedge$
(∀ v_{10} v_{72} v_{73}. P (v_{10} says (v_{72} impf v_{73}))) $\wedge$
(∀ v_{10} v_{74} v_{75}. P (v_{10} says (v_{74} eqf v_{75}))) $\wedge$
(∀ v_{10} v_{76} v_{77}. P (v_{10} says v_{76} says v_{77})) $\wedge$
(∀ v_{10} v_{78} v_{79}. P (v_{10} says v_{78} speaks_for v_{79})) $\wedge$
(∀ v_{10} v_{80} v_{81}. P (v_{10} says v_{80} controls v_{81})) $\wedge$
(∀ v_{10} v_{82} v_{83} v_{84}. P (v_{10} says reps v_{82} v_{83} v_{84})) $\wedge$
(∀ v_{10} v_{85} v_{86}. P (v_{10} says v_{85} domi v_{86})) $\wedge$
(∀ v_{10} v_{87} v_{88}. P (v_{10} says v_{87} eqi v_{88})) $\wedge$
(∀ v_{10} v_{89} v_{90}. P (v_{10} says v_{89} doms v_{90})) $\wedge$
(∀ v_{10} v_{91} v_{92}. P (v_{10} says v_{91} eqs v_{92})) $\wedge$
(∀ v_{10} v_{93} v_{94}. P (v_{10} says v_{93} eqn v_{94})) $\wedge$
(∀ v_{10} v_{95} v_{96}. P (v_{10} says v_{95} lte v_{96})) $\wedge$
(∀ v_{10} v_{97} v_{98}. P (v_{10} says v_{97} lt v_{98})) $\wedge$
(∀ v_{12} v_{13}. P (v_{12} speaks_for v_{13})) $\wedge$
(∀ v_{14} v_{15}. P (v_{14} controls v_{15})) $\wedge$
(∀ v_{16} v_{17} v_{18}. P (reps v_{16} v_{17} v_{18})) $\wedge$
(∀ v_{19} v_{20}. P (v_{19} domi v_{20})) $\wedge$
(∀ v_{21} v_{22}. P (v_{21} eqi v_{22})) $\wedge$
(∀ v_{23} v_{24}. P (v_{23} doms v_{24})) $\wedge$
(∀ v_{25} v_{26}. P (v_{25} eqs v_{26})) $\wedge$ (∀ v_{27} v_{28}. P (v_{27} eqn v_{28})) $\wedge$
(∀ v_{29} v_{30}. P (v_{29} lte v_{30})) $\wedge$ (∀ v_{31} v_{32}. P (v_{31} lt v_{32})) $\Rightarrow$
∀ v. P v

[PBNS_def]

$\vdash$ (PBNS PLAN_PB (exec (SLc crossLD)) = MOVE_TO_ORP) $\wedge$
(PBNS PLAN_PB (exec (SLc incomplete)) = PLAN_PB) $\wedge$
(PBNS MOVE_TO_ORP (exec (SLc conductORP)) = CONDUCT_ORP) $\wedge$
(PBNS MOVE_TO_ORP (exec (SLc incomplete)) = MOVE_TO_ORP) $\wedge$
(PBNS CONDUCT_ORP (exec (SLc moveToPB)) = MOVE_TO_PB) $\wedge$

(PBNS CONDUCT_ORP (exec (SLc incomplete)) = CONDUCT_ORP) ∧
(PBNS MOVE_TO_PB (exec (SLc conductPB)) = CONDUCT_PB) ∧
(PBNS MOVE_TO_PB (exec (SLc incomplete)) = MOVE_TO_PB) ∧
(PBNS CONDUCT_PB (exec (SLc completePB)) = COMPLETE_PB) ∧
(PBNS CONDUCT_PB (exec (SLc incomplete)) = CONDUCT_PB) ∧
(PBNS s (trap (SLc cmd)) = s) ∧
(PBNS s (discard (SLc cmd)) = s)

[PBNS_ind]

⊢ ∀ P.
P PLAN_PB (exec (SLc crossLD)) ∧
P PLAN_PB (exec (SLc incomplete)) ∧
P MOVE_TO_ORP (exec (SLc conductORP)) ∧
P MOVE_TO_ORP (exec (SLc incomplete)) ∧
P CONDUCT_ORP (exec (SLc moveToPB)) ∧
P CONDUCT_ORP (exec (SLc incomplete)) ∧
P MOVE_TO_PB (exec (SLc conductPB)) ∧
P MOVE_TO_PB (exec (SLc incomplete)) ∧
P CONDUCT_PB (exec (SLc completePB)) ∧
P CONDUCT_PB (exec (SLc incomplete)) ∧
(∀ s cmd. P s (trap (SLc cmd))) ∧
(∀ s cmd. P s (discard (SLc cmd))) ∧
(∀ s v_6. P s (discard (ESCc v_6))) ∧
(∀ s v_9. P s (trap (ESCc v_9))) ∧
(∀ v_{12}. P PLAN_PB (exec (ESCc v_{12}))) ∧
P PLAN_PB (exec (SLc conductORP)) ∧
P PLAN_PB (exec (SLc moveToPB)) ∧
P PLAN_PB (exec (SLc conductPB)) ∧
P PLAN_PB (exec (SLc completePB)) ∧
(∀ v_{15}. P MOVE_TO_ORP (exec (ESCc v_{15}))) ∧
P MOVE_TO_ORP (exec (SLc crossLD)) ∧
P MOVE_TO_ORP (exec (SLc moveToPB)) ∧
P MOVE_TO_ORP (exec (SLc conductPB)) ∧
P MOVE_TO_ORP (exec (SLc completePB)) ∧
(∀ v_{18}. P CONDUCT_ORP (exec (ESCc v_{18}))) ∧
P CONDUCT_ORP (exec (SLc crossLD)) ∧
P CONDUCT_ORP (exec (SLc conductORP)) ∧
P CONDUCT_ORP (exec (SLc conductPB)) ∧
P CONDUCT_ORP (exec (SLc completePB)) ∧
(∀ v_{21}. P MOVE_TO_PB (exec (ESCc v_{21}))) ∧
P MOVE_TO_PB (exec (SLc crossLD)) ∧
P MOVE_TO_PB (exec (SLc conductORP)) ∧
P MOVE_TO_PB (exec (SLc moveToPB)) ∧
P MOVE_TO_PB (exec (SLc completePB)) ∧
(∀ v_{24}. P CONDUCT_PB (exec (ESCc v_{24}))) ∧
P CONDUCT_PB (exec (SLc crossLD)) ∧
P CONDUCT_PB (exec (SLc conductORP)) ∧
P CONDUCT_PB (exec (SLc moveToPB)) ∧
P CONDUCT_PB (exec (SLc conductPB)) ∧

$(\forall\, v_{26}.\ P$ COMPLETE_PB (exec $v_{26})) \Rightarrow$
$\forall\, v\ v_1.\ P\ v\ v_1$

[PBOut_def]

$\vdash$ (PBOut PLAN_PB (exec (SLc crossLD)) = MoveToORP) $\wedge$
(PBOut PLAN_PB (exec (SLc incomplete)) = PlanPB) $\wedge$
(PBOut MOVE_TO_ORP (exec (SLc conductORP)) = ConductORP) $\wedge$
(PBOut MOVE_TO_ORP (exec (SLc incomplete)) = MoveToORP) $\wedge$
(PBOut CONDUCT_ORP (exec (SLc moveToPB)) = MoveToPB) $\wedge$
(PBOut CONDUCT_ORP (exec (SLc incomplete)) = ConductORP) $\wedge$
(PBOut MOVE_TO_PB (exec (SLc conductPB)) = ConductPB) $\wedge$
(PBOut MOVE_TO_PB (exec (SLc incomplete)) = MoveToPB) $\wedge$
(PBOut CONDUCT_PB (exec (SLc completePB)) = CompletePB) $\wedge$
(PBOut CONDUCT_PB (exec (SLc incomplete)) = ConductPB) $\wedge$
(PBOut s (trap (SLc cmd)) = unAuthorized) $\wedge$
(PBOut s (discard (SLc cmd)) = unAuthenticated)

[PBOut_ind]

$\vdash \forall P.$
P PLAN_PB (exec (SLc crossLD)) $\wedge$
P PLAN_PB (exec (SLc incomplete)) $\wedge$
P MOVE_TO_ORP (exec (SLc conductORP)) $\wedge$
P MOVE_TO_ORP (exec (SLc incomplete)) $\wedge$
P CONDUCT_ORP (exec (SLc moveToPB)) $\wedge$
P CONDUCT_ORP (exec (SLc incomplete)) $\wedge$
P MOVE_TO_PB (exec (SLc conductPB)) $\wedge$
P MOVE_TO_PB (exec (SLc incomplete)) $\wedge$
P CONDUCT_PB (exec (SLc completePB)) $\wedge$
P CONDUCT_PB (exec (SLc incomplete)) $\wedge$
$(\forall s\ cmd.\ P\ s$ (trap (SLc cmd))) $\wedge$
$(\forall s\ cmd.\ P\ s$ (discard (SLc cmd))) $\wedge$
$(\forall s\ v_6.\ P\ s$ (discard (ESCc v_6))) $\wedge$
$(\forall s\ v_9.\ P\ s$ (trap (ESCc v_9))) $\wedge$
$(\forall v_{12}.\ P$ PLAN_PB (exec (ESCc v_{12}))) $\wedge$
P PLAN_PB (exec (SLc conductORP)) $\wedge$
P PLAN_PB (exec (SLc moveToPB)) $\wedge$
P PLAN_PB (exec (SLc conductPB)) $\wedge$
P PLAN_PB (exec (SLc completePB)) $\wedge$
$(\forall v_{15}.\ P$ MOVE_TO_ORP (exec (ESCc v_{15}))) $\wedge$
P MOVE_TO_ORP (exec (SLc crossLD)) $\wedge$
P MOVE_TO_ORP (exec (SLc moveToPB)) $\wedge$
P MOVE_TO_ORP (exec (SLc conductPB)) $\wedge$
P MOVE_TO_ORP (exec (SLc completePB)) $\wedge$
$(\forall v_{18}.\ P$ CONDUCT_ORP (exec (ESCc v_{18}))) $\wedge$
P CONDUCT_ORP (exec (SLc crossLD)) $\wedge$
P CONDUCT_ORP (exec (SLc conductORP)) $\wedge$
P CONDUCT_ORP (exec (SLc conductPB)) $\wedge$
P CONDUCT_ORP (exec (SLc completePB)) $\wedge$
$(\forall v_{21}.\ P$ MOVE_TO_PB (exec (ESCc v_{21}))) $\wedge$

```
     P MOVE_TO_PB (exec (SLc crossLD)) ∧
     P MOVE_TO_PB (exec (SLc conductORP)) ∧
     P MOVE_TO_PB (exec (SLc moveToPB)) ∧
     P MOVE_TO_PB (exec (SLc completePB)) ∧
     (∀ v24. P CONDUCT_PB (exec (ESCc v24))) ∧
     P CONDUCT_PB (exec (SLc crossLD)) ∧
     P CONDUCT_PB (exec (SLc conductORP)) ∧
     P CONDUCT_PB (exec (SLc moveToPB)) ∧
     P CONDUCT_PB (exec (SLc conductPB)) ∧
     (∀ v26. P COMPLETE_PB (exec v26)) ⇒
     ∀ v v1. P v v1
```

[PlatoonLeader_exec_slCommand_justified_thm]

```
⊢ ∀ NS Out M Oi Os.
    TR (M,Oi,Os) (exec (SLc slCommand))
      (CFG authenticationTest ssmPBStateInterp
         (secContext slCommand)
         (Name PlatoonLeader says prop (SOME (SLc slCommand))::
              ins) s outs)
      (CFG authenticationTest ssmPBStateInterp
         (secContext slCommand) ins
         (NS s (exec (SLc slCommand)))
         (Out s (exec (SLc slCommand))::outs)) ⟺
    authenticationTest
      (Name PlatoonLeader says prop (SOME (SLc slCommand))) ∧
    CFGInterpret (M,Oi,Os)
      (CFG authenticationTest ssmPBStateInterp
         (secContext slCommand)
         (Name PlatoonLeader says prop (SOME (SLc slCommand))::
              ins) s outs) ∧
    (M,Oi,Os) sat prop (SOME (SLc slCommand))
```

[PlatoonLeader_slCommand_lemma]

```
⊢ CFGInterpret (M,Oi,Os)
    (CFG authenticationTest ssmPBStateInterp
       (secContext slCommand)
       (Name PlatoonLeader says prop (SOME (SLc slCommand))::
            ins) s outs) ⇒
  (M,Oi,Os) sat prop (SOME (SLc slCommand))
```

3 PBIntegratedDef Theory

Built: 16 May 2018

Parent Theories: PBTypeIntegrated, aclfoundation

3.1 Definitions

[secAuthorization_def]

```
⊢ ∀xs. secAuthorization xs = secHelper (getOmniCommand xs)
```

[secHelper_def]

```
⊢ ∀cmd.
    secHelper cmd =
    [Name Omni controls prop (SOME (SLc (OMNI cmd)))]
```

3.2 Theorems

[getOmniCommand_def]

```
⊢ (getOmniCommand [] = invalidOmniCommand) ∧
  (∀xs cmd.
     getOmniCommand
       (Name Omni controls prop (SOME (SLc (OMNI cmd)))::xs) =
     cmd) ∧
  (∀xs. getOmniCommand (TT::xs) = getOmniCommand xs) ∧
  (∀xs. getOmniCommand (FF::xs) = getOmniCommand xs) ∧
  (∀xs v2. getOmniCommand (prop v2::xs) = getOmniCommand xs) ∧
  (∀xs v3. getOmniCommand (notf v3::xs) = getOmniCommand xs) ∧
  (∀xs v5 v4.
     getOmniCommand (v4 andf v5::xs) = getOmniCommand xs) ∧
  (∀xs v7 v6.
     getOmniCommand (v6 orf v7::xs) = getOmniCommand xs) ∧
  (∀xs v9 v8.
     getOmniCommand (v8 impf v9::xs) = getOmniCommand xs) ∧
  (∀xs v11 v10.
     getOmniCommand (v10 eqf v11::xs) = getOmniCommand xs) ∧
  (∀xs v13 v12.
     getOmniCommand (v12 says v13::xs) = getOmniCommand xs) ∧
  (∀xs v15 v14.
     getOmniCommand (v14 speaks_for v15::xs) =
     getOmniCommand xs) ∧
  (∀xs v16.
     getOmniCommand (v16 controls TT::xs) =
     getOmniCommand xs) ∧
  (∀xs v16.
     getOmniCommand (v16 controls FF::xs) =
     getOmniCommand xs) ∧
  (∀xs v134.
     getOmniCommand (Name v134 controls prop NONE::xs) =
     getOmniCommand xs) ∧
  (∀xs v144.
     getOmniCommand
       (Name PlatoonLeader controls prop (SOME v144)::xs) =
     getOmniCommand xs) ∧
  (∀xs v146.
     getOmniCommand
```

```
     (Name Omni controls prop (SOME (ESCc v146))::xs) =
   getOmniCommand xs) ∧
(∀ xs v150 .
   getOmniCommand
     (Name Omni controls prop (SOME (SLc (PL v150)))::xs) =
   getOmniCommand xs) ∧
(∀ xs v68 v136 v135 .
   getOmniCommand (v135 meet v136 controls prop v68::xs) =
   getOmniCommand xs) ∧
(∀ xs v68 v138 v137 .
   getOmniCommand (v137 quoting v138 controls prop v68::xs) =
   getOmniCommand xs) ∧
(∀ xs v69 v16 .
   getOmniCommand (v16 controls notf v69::xs) =
   getOmniCommand xs) ∧
(∀ xs v71 v70 v16 .
   getOmniCommand (v16 controls (v70 andf v71)::xs) =
   getOmniCommand xs) ∧
(∀ xs v73 v72 v16 .
   getOmniCommand (v16 controls (v72 orf v73)::xs) =
   getOmniCommand xs) ∧
(∀ xs v75 v74 v16 .
   getOmniCommand (v16 controls (v74 impf v75)::xs) =
   getOmniCommand xs) ∧
(∀ xs v77 v76 v16 .
   getOmniCommand (v16 controls (v76 eqf v77)::xs) =
   getOmniCommand xs) ∧
(∀ xs v79 v78 v16 .
   getOmniCommand (v16 controls v78 says v79::xs) =
   getOmniCommand xs) ∧
(∀ xs v81 v80 v16 .
   getOmniCommand (v16 controls v80 speaks_for v81::xs) =
   getOmniCommand xs) ∧
(∀ xs v83 v82 v16 .
   getOmniCommand (v16 controls v82 controls v83::xs) =
   getOmniCommand xs) ∧
(∀ xs v86 v85 v84 v16 .
   getOmniCommand (v16 controls reps v84 v85 v86::xs) =
   getOmniCommand xs) ∧
(∀ xs v88 v87 v16 .
   getOmniCommand (v16 controls v87 domi v88::xs) =
   getOmniCommand xs) ∧
(∀ xs v90 v89 v16 .
   getOmniCommand (v16 controls v89 eqi v90::xs) =
   getOmniCommand xs) ∧
(∀ xs v92 v91 v16 .
   getOmniCommand (v16 controls v91 doms v92::xs) =
   getOmniCommand xs) ∧
(∀ xs v94 v93 v16 .
```

```
      getOmniCommand (v16 controls v93 eqs v94::xs) =
      getOmniCommand xs) ∧
   (∀ xs v96 v95 v16.
      getOmniCommand (v16 controls v95 eqn v96::xs) =
      getOmniCommand xs) ∧
   (∀ xs v98 v97 v16.
      getOmniCommand (v16 controls v97 lte v98::xs) =
      getOmniCommand xs) ∧
   (∀ xs v99 v16 v100.
      getOmniCommand (v16 controls v99 lt v100::xs) =
      getOmniCommand xs) ∧
   (∀ xs v20 v19 v18.
      getOmniCommand (reps v18 v19 v20::xs) =
      getOmniCommand xs) ∧
   (∀ xs v22 v21.
      getOmniCommand (v21 domi v22::xs) = getOmniCommand xs) ∧
   (∀ xs v24 v23.
      getOmniCommand (v23 eqi v24::xs) = getOmniCommand xs) ∧
   (∀ xs v26 v25.
      getOmniCommand (v25 doms v26::xs) = getOmniCommand xs) ∧
   (∀ xs v28 v27.
      getOmniCommand (v27 eqs v28::xs) = getOmniCommand xs) ∧
   (∀ xs v30 v29.
      getOmniCommand (v29 eqn v30::xs) = getOmniCommand xs) ∧
   (∀ xs v32 v31.
      getOmniCommand (v31 lte v32::xs) = getOmniCommand xs) ∧
   ∀ xs v34 v33.
     getOmniCommand (v33 lt v34::xs) = getOmniCommand xs
```

[getOmniCommand_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ cmd xs.
       P
         (Name Omni controls prop (SOME (SLc (OMNI cmd)))::
              xs)) ∧ (∀ xs. P xs ⇒ P (TT::xs)) ∧
    (∀ xs. P xs ⇒ P (FF::xs)) ∧
    (∀ v2 xs. P xs ⇒ P (prop v2::xs)) ∧
    (∀ v3 xs. P xs ⇒ P (notf v3::xs)) ∧
    (∀ v4 v5 xs. P xs ⇒ P (v4 andf v5::xs)) ∧
    (∀ v6 v7 xs. P xs ⇒ P (v6 orf v7::xs)) ∧
    (∀ v8 v9 xs. P xs ⇒ P (v8 impf v9::xs)) ∧
    (∀ v10 v11 xs. P xs ⇒ P (v10 eqf v11::xs)) ∧
    (∀ v12 v13 xs. P xs ⇒ P (v12 says v13::xs)) ∧
    (∀ v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
    (∀ v16 xs. P xs ⇒ P (v16 controls TT::xs)) ∧
    (∀ v16 xs. P xs ⇒ P (v16 controls FF::xs)) ∧
    (∀ v134 xs. P xs ⇒ P (Name v134 controls prop NONE::xs)) ∧
    (∀ v144 xs.
```

$P\ xs \Rightarrow$
P (Name PlatoonLeader controls prop (SOME $v144$)::xs)) $\wedge$
$(\forall\, v146\ xs.$
$P\ xs \Rightarrow$
P (Name Omni controls prop (SOME (ESCc $v146$))::xs)) $\wedge$
$(\forall\, v150\ xs.$
$P\ xs \Rightarrow$
P
(Name Omni controls prop (SOME (SLc (PL $v150$)))::
xs)) $\wedge$
$(\forall\, v135\ v136\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v135$ meet $v136$ controls prop v_{68}::xs)) $\wedge$
$(\forall\, v137\ v138\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v137$ quoting $v138$ controls prop v_{68}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{69}\ xs.\ P\ xs \Rightarrow P$ (v_{16} controls notf v_{69}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{70}\ v_{71}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls (v_{70} andf v_{71})::xs)) $\wedge$
$(\forall\, v_{16}\ v_{72}\ v_{73}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls (v_{72} orf v_{73})::xs)) $\wedge$
$(\forall\, v_{16}\ v_{74}\ v_{75}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls (v_{74} impf v_{75})::xs)) $\wedge$
$(\forall\, v_{16}\ v_{76}\ v_{77}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls (v_{76} eqf v_{77})::xs)) $\wedge$
$(\forall\, v_{16}\ v_{78}\ v_{79}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{78} says v_{79}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{80}\ v_{81}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{80} speaks_for v_{81}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{82}\ v_{83}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{82} controls v_{83}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls reps v_{84} v_{85} v_{86}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{87}\ v_{88}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{87} domi v_{88}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{89}\ v_{90}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{89} eqi v_{90}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{91}\ v_{92}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{91} doms v_{92}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{93}\ v_{94}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{93} eqs v_{94}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{95}\ v_{96}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{95} eqn v_{96}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{97}\ v_{98}\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{97} lte v_{98}::xs)) $\wedge$
$(\forall\, v_{16}\ v_{99}\ v100\ xs.$
$P\ xs \Rightarrow P$ (v_{16} controls v_{99} lt $v100$::xs)) $\wedge$
$(\forall\, v_{18}\ v_{19}\ v_{20}\ xs.\ P\ xs \Rightarrow P$ (reps v_{18} v_{19} v_{20}::xs)) $\wedge$
$(\forall\, v_{21}\ v_{22}\ xs.\ P\ xs \Rightarrow P$ (v_{21} domi v_{22}::xs)) $\wedge$
$(\forall\, v_{23}\ v_{24}\ xs.\ P\ xs \Rightarrow P$ (v_{23} eqi v_{24}::xs)) $\wedge$
$(\forall\, v_{25}\ v_{26}\ xs.\ P\ xs \Rightarrow P$ (v_{25} doms v_{26}::xs)) $\wedge$

```
     (∀ v27 v28 xs. P xs ⇒ P (v27 eqs v28::xs)) ∧
     (∀ v29 v30 xs. P xs ⇒ P (v29 eqn v30::xs)) ∧
     (∀ v31 v32 xs. P xs ⇒ P (v31 lte v32::xs)) ∧
     (∀ v33 v34 xs. P xs ⇒ P (v33 lt v34::xs)) ⇒
     ∀ v. P v
```

[secContext_def]

```
⊢ (secContext PLAN_PB (x::xs) =
    [prop (SOME (SLc (OMNI ssmPlanPBComplete))) impf
     Name PlatoonLeader controls
     prop (SOME (SLc (PL crossLD)))]) ∧
  (secContext MOVE_TO_ORP (x::xs) =
   [prop (SOME (SLc (OMNI ssmMoveToORPComplete))) impf
    Name PlatoonLeader controls
    prop (SOME (SLc (PL conductORP)))]) ∧
  (secContext CONDUCT_ORP (x::xs) =
   [prop (SOME (SLc (OMNI ssmConductORPComplete))) impf
    Name PlatoonLeader controls
    prop (SOME (SLc (PL moveToPB)))]) ∧
  (secContext MOVE_TO_PB (x::xs) =
   [prop (SOME (SLc (OMNI ssmMoveToPBComplete))) impf
    Name PlatoonLeader controls
    prop (SOME (SLc (PL conductPB)))]) ∧
  (secContext CONDUCT_PB (x::xs) =
   [prop (SOME (SLc (OMNI ssmConductPBComplete))) impf
    Name PlatoonLeader controls
    prop (SOME (SLc (PL completePB)))])
```

[secContext_ind]

```
⊢ ∀ P.
     (∀ x xs. P PLAN_PB (x::xs)) ∧
     (∀ x xs. P MOVE_TO_ORP (x::xs)) ∧
     (∀ x xs. P CONDUCT_ORP (x::xs)) ∧
     (∀ x xs. P MOVE_TO_PB (x::xs)) ∧
     (∀ x xs. P CONDUCT_PB (x::xs)) ∧ (∀ v4. P v4 []) ∧
     (∀ v5 v6. P COMPLETE_PB (v5::v6)) ⇒
     ∀ v v1. P v v1
```

Index