Contents

1 PBType Theory

Built: 09 July 2017
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

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

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

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

stateRole = PlatoonLeader

1.2 Theorems

[slCommand_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

[slOutput_distinct_clauses]

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

[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

2 ssmPB Theory

Built: 09 July 2017
Parent Theories: PBType, ssm11, OMNIType

2.1 Definitions

[secContext_def]

⊢ ∀ *cmd*.
secContext *cmd* =
[Name PlatoonLeader controls prop (SOME (SLc *cmd*))]

[ssmPBStateInterp_def]

⊢ ∀ *state*. ssmPBStateInterp *state* = TT

2.2 Theorems

[authenticationTest_cmd_reject_lemma]

⊢ ∀ *cmd*. ¬authenticationTest (prop (SOME *cmd*))

[authenticationTest_def]

⊢ (authenticationTest (Name PlatoonLeader says prop *cmd*) ⟺
T) ∧ (authenticationTest TT ⟺ F) ∧
(authenticationTest FF ⟺ F) ∧
(authenticationTest (prop *v*) ⟺ F) ∧
(authenticationTest (notf v_1) ⟺ F) ∧
(authenticationTest (v_2 andf v_3) ⟺ F) ∧
(authenticationTest (v_4 orf v_5) ⟺ F) ∧
(authenticationTest (v_6 impf v_7) ⟺ F) ∧
(authenticationTest (v_8 eqf v_9) ⟺ F) ∧
(authenticationTest (v_{10} says TT) ⟺ F) ∧
(authenticationTest (v_{10} says FF) ⟺ F) ∧
(authenticationTest (*v133* meet *v134* says prop v_{66}) ⟺ F) ∧
(authenticationTest (*v135* quoting *v136* says prop v_{66}) ⟺ F) ∧
(authenticationTest (v_{10} says notf v_{67}) ⟺ F) ∧
(authenticationTest (v_{10} says (v_{68} andf v_{69})) ⟺ F) ∧
(authenticationTest (v_{10} says (v_{70} orf v_{71})) ⟺ F) ∧
(authenticationTest (v_{10} says (v_{72} impf v_{73})) ⟺ F) ∧
(authenticationTest (v_{10} says (v_{74} eqf v_{75})) ⟺ F) ∧
(authenticationTest (v_{10} says v_{76} says v_{77}) ⟺ F) ∧
(authenticationTest (v_{10} says v_{78} speaks_for v_{79}) ⟺ F) ∧

(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})) $\wedge$

$(\forall\, v_{21}\ v_{22}.\ P\ (v_{21}$ `eqi` $v_{22})) \wedge$
$(\forall\, v_{23}\ v_{24}.\ P\ (v_{23}$ `doms` $v_{24})) \wedge$
$(\forall\, v_{25}\ v_{26}.\ P\ (v_{25}$ `eqs` $v_{26})) \wedge (\forall\, v_{27}\ v_{28}.\ P\ (v_{27}$ `eqn` $v_{28})) \wedge$
$(\forall\, v_{29}\ v_{30}.\ P\ (v_{29}$ `lte` $v_{30})) \wedge (\forall\, v_{31}\ v_{32}.\ P\ (v_{31}$ `lt` $v_{32})) \Rightarrow$
$\forall\, v.\ P\ v$

[PBNS_def]

```
⊢ (PBNS PLAN_PB (exec (SLc crossLD)) = MOVE_TO_ORP) ∧
  (PBNS PLAN_PB (exec (SLc incomplete)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (exec (SLc conductORP)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_ORP (exec (SLc incomplete)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (exec (SLc moveToPB)) = MOVE_TO_PB) ∧
  (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 PLAN_PB (trap (SLc crossLD)) = PLAN_PB) ∧
  (PBNS PLAN_PB (trap (SLc incomplete)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (trap (SLc conductORP)) = MOVE_TO_ORP) ∧
  (PBNS MOVE_TO_ORP (trap (SLc incomplete)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (trap (SLc moveToPB)) = CONDUCT_ORP) ∧
  (PBNS CONDUCT_ORP (trap (SLc incomplete)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_PB (trap (SLc conductPB)) = MOVE_TO_PB) ∧
  (PBNS MOVE_TO_PB (trap (SLc incomplete)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_PB (trap (SLc completePB)) = CONDUCT_PB) ∧
  (PBNS CONDUCT_PB (trap (SLc incomplete)) = CONDUCT_PB) ∧
  (PBNS PLAN_PB (discard (SLc crossLD)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (discard (SLc conductORP)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (discard (SLc moveToPB)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_PB (discard (SLc conductPB)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_PB (discard (SLc completePB)) = CONDUCT_PB)
```

[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)) ∧
    P PLAN_PB (trap (SLc crossLD)) ∧
    P PLAN_PB (trap (SLc incomplete)) ∧
    P MOVE_TO_ORP (trap (SLc conductORP)) ∧
    P MOVE_TO_ORP (trap (SLc incomplete)) ∧
```

P CONDUCT_ORP (trap (SLc moveToPB)) $\wedge$
P CONDUCT_ORP (trap (SLc incomplete)) $\wedge$
P MOVE_TO_PB (trap (SLc conductPB)) $\wedge$
P MOVE_TO_PB (trap (SLc incomplete)) $\wedge$
P CONDUCT_PB (trap (SLc completePB)) $\wedge$
P CONDUCT_PB (trap (SLc incomplete)) $\wedge$
P PLAN_PB (discard (SLc crossLD)) $\wedge$
P MOVE_TO_ORP (discard (SLc conductORP)) $\wedge$
P CONDUCT_ORP (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (discard (SLc conductPB)) $\wedge$
P CONDUCT_PB (discard (SLc completePB)) $\wedge$
$(\forall\, v_8\ v_6.\ P\ v_8$ (discard (ESCc v_6))) $\wedge$
P MOVE_TO_ORP (discard (SLc crossLD)) $\wedge$
P CONDUCT_ORP (discard (SLc crossLD)) $\wedge$
P MOVE_TO_PB (discard (SLc crossLD)) $\wedge$
P CONDUCT_PB (discard (SLc crossLD)) $\wedge$
P COMPLETE_PB (discard (SLc crossLD)) $\wedge$
P PLAN_PB (discard (SLc conductORP)) $\wedge$
P CONDUCT_ORP (discard (SLc conductORP)) $\wedge$
P MOVE_TO_PB (discard (SLc conductORP)) $\wedge$
P CONDUCT_PB (discard (SLc conductORP)) $\wedge$
P COMPLETE_PB (discard (SLc conductORP)) $\wedge$
P PLAN_PB (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_ORP (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (discard (SLc moveToPB)) $\wedge$
P CONDUCT_PB (discard (SLc moveToPB)) $\wedge$
P COMPLETE_PB (discard (SLc moveToPB)) $\wedge$
P PLAN_PB (discard (SLc conductPB)) $\wedge$
P MOVE_TO_ORP (discard (SLc conductPB)) $\wedge$
P CONDUCT_ORP (discard (SLc conductPB)) $\wedge$
P CONDUCT_PB (discard (SLc conductPB)) $\wedge$
P COMPLETE_PB (discard (SLc conductPB)) $\wedge$
P PLAN_PB (discard (SLc completePB)) $\wedge$
P MOVE_TO_ORP (discard (SLc completePB)) $\wedge$
P CONDUCT_ORP (discard (SLc completePB)) $\wedge$
P MOVE_TO_PB (discard (SLc completePB)) $\wedge$
P COMPLETE_PB (discard (SLc completePB)) $\wedge$
$(\forall\, v_9.\ P\ v_9$ (discard (SLc incomplete))) $\wedge$
$(\forall\, v_{13}\ v_{11}.\ P\ v_{13}$ (trap (ESCc v_{11}))) $\wedge$
P MOVE_TO_ORP (trap (SLc crossLD)) $\wedge$
P CONDUCT_ORP (trap (SLc crossLD)) $\wedge$
P MOVE_TO_PB (trap (SLc crossLD)) $\wedge$
P CONDUCT_PB (trap (SLc crossLD)) $\wedge$
P COMPLETE_PB (trap (SLc crossLD)) $\wedge$
P PLAN_PB (trap (SLc conductORP)) $\wedge$
P CONDUCT_ORP (trap (SLc conductORP)) $\wedge$
P MOVE_TO_PB (trap (SLc conductORP)) $\wedge$
P CONDUCT_PB (trap (SLc conductORP)) $\wedge$
P COMPLETE_PB (trap (SLc conductORP)) $\wedge$

P PLAN_PB (trap (SLc moveToPB)) ∧
P MOVE_TO_ORP (trap (SLc moveToPB)) ∧
P MOVE_TO_PB (trap (SLc moveToPB)) ∧
P CONDUCT_PB (trap (SLc moveToPB)) ∧
P COMPLETE_PB (trap (SLc moveToPB)) ∧
P PLAN_PB (trap (SLc conductPB)) ∧
P MOVE_TO_ORP (trap (SLc conductPB)) ∧
P CONDUCT_ORP (trap (SLc conductPB)) ∧
P CONDUCT_PB (trap (SLc conductPB)) ∧
P COMPLETE_PB (trap (SLc conductPB)) ∧
P PLAN_PB (trap (SLc completePB)) ∧
P MOVE_TO_ORP (trap (SLc completePB)) ∧
P CONDUCT_ORP (trap (SLc completePB)) ∧
P MOVE_TO_PB (trap (SLc completePB)) ∧
P COMPLETE_PB (trap (SLc completePB)) ∧
P COMPLETE_PB (trap (SLc incomplete)) ∧
(∀ v_{17} v_{15}. P v_{17} (exec (ESCc v_{15}))) ∧
P MOVE_TO_ORP (exec (SLc crossLD)) ∧
P CONDUCT_ORP (exec (SLc crossLD)) ∧
P MOVE_TO_PB (exec (SLc crossLD)) ∧
P CONDUCT_PB (exec (SLc crossLD)) ∧
P COMPLETE_PB (exec (SLc crossLD)) ∧
P PLAN_PB (exec (SLc conductORP)) ∧
P CONDUCT_ORP (exec (SLc conductORP)) ∧
P MOVE_TO_PB (exec (SLc conductORP)) ∧
P CONDUCT_PB (exec (SLc conductORP)) ∧
P COMPLETE_PB (exec (SLc conductORP)) ∧
P PLAN_PB (exec (SLc moveToPB)) ∧
P MOVE_TO_ORP (exec (SLc moveToPB)) ∧
P MOVE_TO_PB (exec (SLc moveToPB)) ∧
P CONDUCT_PB (exec (SLc moveToPB)) ∧
P COMPLETE_PB (exec (SLc moveToPB)) ∧
P PLAN_PB (exec (SLc conductPB)) ∧
P MOVE_TO_ORP (exec (SLc conductPB)) ∧
P CONDUCT_ORP (exec (SLc conductPB)) ∧
P CONDUCT_PB (exec (SLc conductPB)) ∧
P COMPLETE_PB (exec (SLc conductPB)) ∧
P PLAN_PB (exec (SLc completePB)) ∧
P MOVE_TO_ORP (exec (SLc completePB)) ∧
P CONDUCT_ORP (exec (SLc completePB)) ∧
P MOVE_TO_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc incomplete)) ⇒
∀ v v_1. P v v_1

[PBOut_def]

⊢ (PBOut PLAN_PB (exec (SLc crossLD)) = MoveToORP) ∧
(PBOut PLAN_PB (exec (SLc incomplete)) = PlanPB) ∧
(PBOut MOVE_TO_ORP (exec (SLc conductORP)) = ConductORP) ∧

```
(PBOut MOVE_TO_ORP (exec (SLc incomplete)) = MoveToORP) ∧
(PBOut CONDUCT_ORP (exec (SLc moveToPB)) = MoveToPB) ∧
(PBOut CONDUCT_ORP (exec (SLc incomplete)) = ConductORP) ∧
(PBOut MOVE_TO_PB (exec (SLc conductPB)) = ConductPB) ∧
(PBOut MOVE_TO_PB (exec (SLc incomplete)) = MoveToPB) ∧
(PBOut CONDUCT_PB (exec (SLc completePB)) = CompletePB) ∧
(PBOut CONDUCT_PB (exec (SLc incomplete)) = ConductPB) ∧
(PBOut PLAN_PB (trap (SLc crossLD)) = PlanPB) ∧
(PBOut PLAN_PB (trap (SLc incomplete)) = PlanPB) ∧
(PBOut MOVE_TO_ORP (trap (SLc conductORP)) = MoveToORP) ∧
(PBOut MOVE_TO_ORP (trap (SLc incomplete)) = MoveToORP) ∧
(PBOut CONDUCT_ORP (trap (SLc moveToPB)) = ConductORP) ∧
(PBOut CONDUCT_ORP (trap (SLc incomplete)) = ConductORP) ∧
(PBOut MOVE_TO_PB (trap (SLc conductPB)) = MoveToPB) ∧
(PBOut MOVE_TO_PB (trap (SLc incomplete)) = MoveToPB) ∧
(PBOut CONDUCT_PB (trap (SLc completePB)) = ConductPB) ∧
(PBOut CONDUCT_PB (trap (SLc incomplete)) = ConductPB) ∧
(PBOut PLAN_PB (discard (SLc crossLD)) = unAuthenticated) ∧
(PBOut MOVE_TO_ORP (discard (SLc conductORP)) =
 unAuthenticated) ∧
(PBOut CONDUCT_ORP (discard (SLc moveToPB)) =
 unAuthenticated) ∧
(PBOut MOVE_TO_PB (discard (SLc conductPB)) =
 unAuthenticated) ∧
(PBOut CONDUCT_PB (discard (SLc completePB)) =
 unAuthenticated)
```

[PBOut_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)) ∧
    P PLAN_PB (trap (SLc crossLD)) ∧
    P PLAN_PB (trap (SLc incomplete)) ∧
    P MOVE_TO_ORP (trap (SLc conductORP)) ∧
    P MOVE_TO_ORP (trap (SLc incomplete)) ∧
    P CONDUCT_ORP (trap (SLc moveToPB)) ∧
    P CONDUCT_ORP (trap (SLc incomplete)) ∧
    P MOVE_TO_PB (trap (SLc conductPB)) ∧
    P MOVE_TO_PB (trap (SLc incomplete)) ∧
    P CONDUCT_PB (trap (SLc completePB)) ∧
    P CONDUCT_PB (trap (SLc incomplete)) ∧
```

P PLAN_PB (discard (SLc crossLD)) $\wedge$
P MOVE_TO_ORP (discard (SLc conductORP)) $\wedge$
P CONDUCT_ORP (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (discard (SLc conductPB)) $\wedge$
P CONDUCT_PB (discard (SLc completePB)) $\wedge$
$(\forall v_8\ v_6.\ P\ v_8$ (discard (ESCc v_6))) $\wedge$
P MOVE_TO_ORP (discard (SLc crossLD)) $\wedge$
P CONDUCT_ORP (discard (SLc crossLD)) $\wedge$
P MOVE_TO_PB (discard (SLc crossLD)) $\wedge$
P CONDUCT_PB (discard (SLc crossLD)) $\wedge$
P COMPLETE_PB (discard (SLc crossLD)) $\wedge$
P PLAN_PB (discard (SLc conductORP)) $\wedge$
P CONDUCT_ORP (discard (SLc conductORP)) $\wedge$
P MOVE_TO_PB (discard (SLc conductORP)) $\wedge$
P CONDUCT_PB (discard (SLc conductORP)) $\wedge$
P COMPLETE_PB (discard (SLc conductORP)) $\wedge$
P PLAN_PB (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_ORP (discard (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (discard (SLc moveToPB)) $\wedge$
P CONDUCT_PB (discard (SLc moveToPB)) $\wedge$
P COMPLETE_PB (discard (SLc moveToPB)) $\wedge$
P PLAN_PB (discard (SLc conductPB)) $\wedge$
P MOVE_TO_ORP (discard (SLc conductPB)) $\wedge$
P CONDUCT_ORP (discard (SLc conductPB)) $\wedge$
P CONDUCT_PB (discard (SLc conductPB)) $\wedge$
P COMPLETE_PB (discard (SLc conductPB)) $\wedge$
P PLAN_PB (discard (SLc completePB)) $\wedge$
P MOVE_TO_ORP (discard (SLc completePB)) $\wedge$
P CONDUCT_ORP (discard (SLc completePB)) $\wedge$
P MOVE_TO_PB (discard (SLc completePB)) $\wedge$
P COMPLETE_PB (discard (SLc completePB)) $\wedge$
$(\forall v_9.\ P\ v_9$ (discard (SLc incomplete))) $\wedge$
$(\forall v_{13}\ v_{11}.\ P\ v_{13}$ (trap (ESCc v_{11}))) $\wedge$
P MOVE_TO_ORP (trap (SLc crossLD)) $\wedge$
P CONDUCT_ORP (trap (SLc crossLD)) $\wedge$
P MOVE_TO_PB (trap (SLc crossLD)) $\wedge$
P CONDUCT_PB (trap (SLc crossLD)) $\wedge$
P COMPLETE_PB (trap (SLc crossLD)) $\wedge$
P PLAN_PB (trap (SLc conductORP)) $\wedge$
P CONDUCT_ORP (trap (SLc conductORP)) $\wedge$
P MOVE_TO_PB (trap (SLc conductORP)) $\wedge$
P CONDUCT_PB (trap (SLc conductORP)) $\wedge$
P COMPLETE_PB (trap (SLc conductORP)) $\wedge$
P PLAN_PB (trap (SLc moveToPB)) $\wedge$
P MOVE_TO_ORP (trap (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (trap (SLc moveToPB)) $\wedge$
P CONDUCT_PB (trap (SLc moveToPB)) $\wedge$
P COMPLETE_PB (trap (SLc moveToPB)) $\wedge$
P PLAN_PB (trap (SLc conductPB)) $\wedge$

P MOVE_TO_ORP (trap (SLc conductPB)) ∧
P CONDUCT_ORP (trap (SLc conductPB)) ∧
P CONDUCT_PB (trap (SLc conductPB)) ∧
P COMPLETE_PB (trap (SLc conductPB)) ∧
P PLAN_PB (trap (SLc completePB)) ∧
P MOVE_TO_ORP (trap (SLc completePB)) ∧
P CONDUCT_ORP (trap (SLc completePB)) ∧
P MOVE_TO_PB (trap (SLc completePB)) ∧
P COMPLETE_PB (trap (SLc completePB)) ∧
P COMPLETE_PB (trap (SLc incomplete)) ∧
(∀ v_{17} v_{15}. P v_{17} (exec (ESCc v_{15}))) ∧
P MOVE_TO_ORP (exec (SLc crossLD)) ∧
P CONDUCT_ORP (exec (SLc crossLD)) ∧
P MOVE_TO_PB (exec (SLc crossLD)) ∧
P CONDUCT_PB (exec (SLc crossLD)) ∧
P COMPLETE_PB (exec (SLc crossLD)) ∧
P PLAN_PB (exec (SLc conductORP)) ∧
P CONDUCT_ORP (exec (SLc conductORP)) ∧
P MOVE_TO_PB (exec (SLc conductORP)) ∧
P CONDUCT_PB (exec (SLc conductORP)) ∧
P COMPLETE_PB (exec (SLc conductORP)) ∧
P PLAN_PB (exec (SLc moveToPB)) ∧
P MOVE_TO_ORP (exec (SLc moveToPB)) ∧
P MOVE_TO_PB (exec (SLc moveToPB)) ∧
P CONDUCT_PB (exec (SLc moveToPB)) ∧
P COMPLETE_PB (exec (SLc moveToPB)) ∧
P PLAN_PB (exec (SLc conductPB)) ∧
P MOVE_TO_ORP (exec (SLc conductPB)) ∧
P CONDUCT_ORP (exec (SLc conductPB)) ∧
P CONDUCT_PB (exec (SLc conductPB)) ∧
P COMPLETE_PB (exec (SLc conductPB)) ∧
P PLAN_PB (exec (SLc completePB)) ∧
P MOVE_TO_ORP (exec (SLc completePB)) ∧
P CONDUCT_ORP (exec (SLc completePB)) ∧
P MOVE_TO_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc incomplete)) ⇒
∀ v v_1. P v v_1

[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_justified_slCommand_exec_thm]

```
⊢ ∀NS Out M Oi Os cmd slCommand ins s 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) ⇒
    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))
```

[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))
```

[PlatoonLeader_slCommand_verified_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)) ⇒
    (M,Oi,Os) sat prop (SOME (SLc slCommand))
```

Index