Contents

1 PlanPBType Theory

Built: 16 May 2018
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

plCommand = receiveMission | warno | tentativePlan | recon
| report1 | completePlan | opoid | supervise | report2
| complete | plIncomplete | invalidPlCommand

psgCommand = initiateMovement | psgIncomplete
| invalidPsgCommand

slCommand = PL plCommand | PSG psgCommand

slOutput = PlanPB | ReceiveMission | Warno | TentativePlan
| InitiateMovement | Recon | Report1 | CompletePlan
| Opoid | Supervise | Report2 | Complete
| unAuthenticated | unAuthorized

slState = PLAN_PB | RECEIVE_MISSION | WARNO | TENTATIVE_PLAN
| INITIATE_MOVEMENT | RECON | REPORT1 | COMPLETE_PLAN
| OPOID | SUPERVISE | REPORT2 | COMPLETE

stateRole = PlatoonLeader | PlatoonSergeant

1.2 Theorems

[plCommand_distinct_clauses]

⊢ receiveMission ≠ warno ∧ receiveMission ≠ tentativePlan ∧
receiveMission ≠ recon ∧ receiveMission ≠ report1 ∧
receiveMission ≠ completePlan ∧ receiveMission ≠ opoid ∧
receiveMission ≠ supervise ∧ receiveMission ≠ report2 ∧
receiveMission ≠ complete ∧ receiveMission ≠ plIncomplete ∧
receiveMission ≠ invalidPlCommand ∧ warno ≠ tentativePlan ∧
warno ≠ recon ∧ warno ≠ report1 ∧ warno ≠ completePlan ∧
warno ≠ opoid ∧ warno ≠ supervise ∧ warno ≠ report2 ∧
warno ≠ complete ∧ warno ≠ plIncomplete ∧
warno ≠ invalidPlCommand ∧ tentativePlan ≠ recon ∧
tentativePlan ≠ report1 ∧ tentativePlan ≠ completePlan ∧
tentativePlan ≠ opoid ∧ tentativePlan ≠ supervise ∧
tentativePlan ≠ report2 ∧ tentativePlan ≠ complete ∧
tentativePlan ≠ plIncomplete ∧
tentativePlan ≠ invalidPlCommand ∧ recon ≠ report1 ∧
recon ≠ completePlan ∧ recon ≠ opoid ∧ recon ≠ supervise ∧
recon ≠ report2 ∧ recon ≠ complete ∧ recon ≠ plIncomplete ∧
recon ≠ invalidPlCommand ∧ report1 ≠ completePlan ∧

report1 ≠ opoid ∧ report1 ≠ supervise ∧ report1 ≠ report2 ∧
report1 ≠ complete ∧ report1 ≠ plIncomplete ∧
report1 ≠ invalidPlCommand ∧ completePlan ≠ opoid ∧
completePlan ≠ supervise ∧ completePlan ≠ report2 ∧
completePlan ≠ complete ∧ completePlan ≠ plIncomplete ∧
completePlan ≠ invalidPlCommand ∧ opoid ≠ supervise ∧
opoid ≠ report2 ∧ opoid ≠ complete ∧ opoid ≠ plIncomplete ∧
opoid ≠ invalidPlCommand ∧ supervise ≠ report2 ∧
supervise ≠ complete ∧ supervise ≠ plIncomplete ∧
supervise ≠ invalidPlCommand ∧ report2 ≠ complete ∧
report2 ≠ plIncomplete ∧ report2 ≠ invalidPlCommand ∧
complete ≠ plIncomplete ∧ complete ≠ invalidPlCommand ∧
plIncomplete ≠ invalidPlCommand

[psgCommand_distinct_clauses]

⊢ initiateMovement ≠ psgIncomplete ∧
initiateMovement ≠ invalidPsgCommand ∧
psgIncomplete ≠ invalidPsgCommand

[slCommand_distinct_clauses]

⊢ ∀ a' a. PL a ≠ PSG a'

[slCommand_one_one]

⊢ (∀ a a'. (PL a = PL a') ⟺ (a = a')) ∧
∀ a a'. (PSG a = PSG a') ⟺ (a = a')

[slOutput_distinct_clauses]

⊢ PlanPB ≠ ReceiveMission ∧ PlanPB ≠ Warno ∧
PlanPB ≠ TentativePlan ∧ PlanPB ≠ InitiateMovement ∧
PlanPB ≠ Recon ∧ PlanPB ≠ Report1 ∧ PlanPB ≠ CompletePlan ∧
PlanPB ≠ Opoid ∧ PlanPB ≠ Supervise ∧ PlanPB ≠ Report2 ∧
PlanPB ≠ Complete ∧ PlanPB ≠ unAuthenticated ∧
PlanPB ≠ unAuthorized ∧ ReceiveMission ≠ Warno ∧
ReceiveMission ≠ TentativePlan ∧
ReceiveMission ≠ InitiateMovement ∧ ReceiveMission ≠ Recon ∧
ReceiveMission ≠ Report1 ∧ ReceiveMission ≠ CompletePlan ∧
ReceiveMission ≠ Opoid ∧ ReceiveMission ≠ Supervise ∧
ReceiveMission ≠ Report2 ∧ ReceiveMission ≠ Complete ∧
ReceiveMission ≠ unAuthenticated ∧
ReceiveMission ≠ unAuthorized ∧ Warno ≠ TentativePlan ∧
Warno ≠ InitiateMovement ∧ Warno ≠ Recon ∧ Warno ≠ Report1 ∧
Warno ≠ CompletePlan ∧ Warno ≠ Opoid ∧ Warno ≠ Supervise ∧
Warno ≠ Report2 ∧ Warno ≠ Complete ∧
Warno ≠ unAuthenticated ∧ Warno ≠ unAuthorized ∧
TentativePlan ≠ InitiateMovement ∧ TentativePlan ≠ Recon ∧
TentativePlan ≠ Report1 ∧ TentativePlan ≠ CompletePlan ∧
TentativePlan ≠ Opoid ∧ TentativePlan ≠ Supervise ∧
TentativePlan ≠ Report2 ∧ TentativePlan ≠ Complete ∧

TentativePlan ≠ unAuthenticated ∧
TentativePlan ≠ unAuthorized ∧ InitiateMovement ≠ Recon ∧
InitiateMovement ≠ Report1 ∧
InitiateMovement ≠ CompletePlan ∧ InitiateMovement ≠ Opoid ∧
InitiateMovement ≠ Supervise ∧ InitiateMovement ≠ Report2 ∧
InitiateMovement ≠ Complete ∧
InitiateMovement ≠ unAuthenticated ∧
InitiateMovement ≠ unAuthorized ∧ Recon ≠ Report1 ∧
Recon ≠ CompletePlan ∧ Recon ≠ Opoid ∧ Recon ≠ Supervise ∧
Recon ≠ Report2 ∧ Recon ≠ Complete ∧
Recon ≠ unAuthenticated ∧ Recon ≠ unAuthorized ∧
Report1 ≠ CompletePlan ∧ Report1 ≠ Opoid ∧
Report1 ≠ Supervise ∧ Report1 ≠ Report2 ∧
Report1 ≠ Complete ∧ Report1 ≠ unAuthenticated ∧
Report1 ≠ unAuthorized ∧ CompletePlan ≠ Opoid ∧
CompletePlan ≠ Supervise ∧ CompletePlan ≠ Report2 ∧
CompletePlan ≠ Complete ∧ CompletePlan ≠ unAuthenticated ∧
CompletePlan ≠ unAuthorized ∧ Opoid ≠ Supervise ∧
Opoid ≠ Report2 ∧ Opoid ≠ Complete ∧
Opoid ≠ unAuthenticated ∧ Opoid ≠ unAuthorized ∧
Supervise ≠ Report2 ∧ Supervise ≠ Complete ∧
Supervise ≠ unAuthenticated ∧ Supervise ≠ unAuthorized ∧
Report2 ≠ Complete ∧ Report2 ≠ unAuthenticated ∧
Report2 ≠ unAuthorized ∧ Complete ≠ unAuthenticated ∧
Complete ≠ unAuthorized ∧ unAuthenticated ≠ unAuthorized

[slRole_distinct_clauses]

⊢ PlatoonLeader ≠ PlatoonSergeant

[slState_distinct_clauses]

⊢ PLAN_PB ≠ RECEIVE_MISSION ∧ PLAN_PB ≠ WARNO ∧
PLAN_PB ≠ TENTATIVE_PLAN ∧ PLAN_PB ≠ INITIATE_MOVEMENT ∧
PLAN_PB ≠ RECON ∧ PLAN_PB ≠ REPORT1 ∧
PLAN_PB ≠ COMPLETE_PLAN ∧ PLAN_PB ≠ OPOID ∧
PLAN_PB ≠ SUPERVISE ∧ PLAN_PB ≠ REPORT2 ∧
PLAN_PB ≠ COMPLETE ∧ RECEIVE_MISSION ≠ WARNO ∧
RECEIVE_MISSION ≠ TENTATIVE_PLAN ∧
RECEIVE_MISSION ≠ INITIATE_MOVEMENT ∧
RECEIVE_MISSION ≠ RECON ∧ RECEIVE_MISSION ≠ REPORT1 ∧
RECEIVE_MISSION ≠ COMPLETE_PLAN ∧ RECEIVE_MISSION ≠ OPOID ∧
RECEIVE_MISSION ≠ SUPERVISE ∧ RECEIVE_MISSION ≠ REPORT2 ∧
RECEIVE_MISSION ≠ COMPLETE ∧ WARNO ≠ TENTATIVE_PLAN ∧
WARNO ≠ INITIATE_MOVEMENT ∧ WARNO ≠ RECON ∧ WARNO ≠ REPORT1 ∧
WARNO ≠ COMPLETE_PLAN ∧ WARNO ≠ OPOID ∧ WARNO ≠ SUPERVISE ∧
WARNO ≠ REPORT2 ∧ WARNO ≠ COMPLETE ∧
TENTATIVE_PLAN ≠ INITIATE_MOVEMENT ∧ TENTATIVE_PLAN ≠ RECON ∧
TENTATIVE_PLAN ≠ REPORT1 ∧ TENTATIVE_PLAN ≠ COMPLETE_PLAN ∧
TENTATIVE_PLAN ≠ OPOID ∧ TENTATIVE_PLAN ≠ SUPERVISE ∧
TENTATIVE_PLAN ≠ REPORT2 ∧ TENTATIVE_PLAN ≠ COMPLETE ∧

INITIATE_MOVEMENT ≠ RECON ∧ INITIATE_MOVEMENT ≠ REPORT1 ∧
INITIATE_MOVEMENT ≠ COMPLETE_PLAN ∧
INITIATE_MOVEMENT ≠ OPOID ∧ INITIATE_MOVEMENT ≠ SUPERVISE ∧
INITIATE_MOVEMENT ≠ REPORT2 ∧ INITIATE_MOVEMENT ≠ COMPLETE ∧
RECON ≠ REPORT1 ∧ RECON ≠ COMPLETE_PLAN ∧ RECON ≠ OPOID ∧
RECON ≠ SUPERVISE ∧ RECON ≠ REPORT2 ∧ RECON ≠ COMPLETE ∧
REPORT1 ≠ COMPLETE_PLAN ∧ REPORT1 ≠ OPOID ∧
REPORT1 ≠ SUPERVISE ∧ REPORT1 ≠ REPORT2 ∧
REPORT1 ≠ COMPLETE ∧ COMPLETE_PLAN ≠ OPOID ∧
COMPLETE_PLAN ≠ SUPERVISE ∧ COMPLETE_PLAN ≠ REPORT2 ∧
COMPLETE_PLAN ≠ COMPLETE ∧ OPOID ≠ SUPERVISE ∧
OPOID ≠ REPORT2 ∧ OPOID ≠ COMPLETE ∧ SUPERVISE ≠ REPORT2 ∧
SUPERVISE ≠ COMPLETE ∧ REPORT2 ≠ COMPLETE

2 ssmPlanPB Theory

Built: 16 May 2018
Parent Theories: PlanPBDef, ssm

2.1 Theorems

[inputOK_def]

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

(inputOK (v_{12} speaks_for v_{13}) ⟺ F) ∧
(inputOK (v_{14} controls v_{15}) ⟺ F) ∧
(inputOK (reps v_{16} v_{17} v_{18}) ⟺ F) ∧
(inputOK (v_{19} domi v_{20}) ⟺ F) ∧
(inputOK (v_{21} eqi v_{22}) ⟺ F) ∧
(inputOK (v_{23} doms v_{24}) ⟺ F) ∧
(inputOK (v_{25} eqs v_{26}) ⟺ F) ∧ (inputOK (v_{27} eqn v_{28}) ⟺ F) ∧
(inputOK (v_{29} lte v_{30}) ⟺ F) ∧ (inputOK (v_{31} lt v_{32}) ⟺ F)

[inputOK_ind]

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

[planPBNS_def]

⊢ (planPBNS WARNO (exec x) =
if
(getRecon x = [SOME (SLc (PL recon))]) ∧

```
      (getTenativePlan x = [SOME (SLc (PL tentativePlan))]) ∧
      (getReport x = [SOME (SLc (PL report1))]) ∧
      (getInitMove x = [SOME (SLc (PSG initiateMovement))])
    then
      REPORT1
    else WARNO) ∧
   (planPBNS PLAN_PB (exec x) =
    if getPlCom x = receiveMission then RECEIVE_MISSION
    else PLAN_PB) ∧
   (planPBNS RECEIVE_MISSION (exec x) =
    if getPlCom x = warno then WARNO else RECEIVE_MISSION) ∧
   (planPBNS REPORT1 (exec x) =
    if getPlCom x = completePlan then COMPLETE_PLAN
    else REPORT1) ∧
   (planPBNS COMPLETE_PLAN (exec x) =
    if getPlCom x = opoid then OPOID else COMPLETE_PLAN) ∧
   (planPBNS OPOID (exec x) =
    if getPlCom x = supervise then SUPERVISE else OPOID) ∧
   (planPBNS SUPERVISE (exec x) =
    if getPlCom x = report2 then REPORT2 else SUPERVISE) ∧
   (planPBNS REPORT2 (exec x) =
    if getPlCom x = complete then COMPLETE else REPORT2) ∧
   (planPBNS s (trap v0) = s) ∧ (planPBNS s (discard v1) = s)
```

[planPBNS_ind]

```
⊢ ∀P.
    (∀x. P WARNO (exec x)) ∧ (∀x. P PLAN_PB (exec x)) ∧
    (∀x. P RECEIVE_MISSION (exec x)) ∧
    (∀x. P REPORT1 (exec x)) ∧ (∀x. P COMPLETE_PLAN (exec x)) ∧
    (∀x. P OPOID (exec x)) ∧ (∀x. P SUPERVISE (exec x)) ∧
    (∀x. P REPORT2 (exec x)) ∧ (∀s v0. P s (trap v0)) ∧
    (∀s v1. P s (discard v1)) ∧
    (∀v6. P TENTATIVE_PLAN (exec v6)) ∧
    (∀v7. P INITIATE_MOVEMENT (exec v7)) ∧
    (∀v8. P RECON (exec v8)) ∧ (∀v9. P COMPLETE (exec v9)) ⇒
    ∀v v1. P v v1
```

[planPBOut_def]

```
⊢ (planPBOut WARNO (exec x) =
    if
      (getRecon x = [SOME (SLc (PL recon))]) ∧
      (getTenativePlan x = [SOME (SLc (PL tentativePlan))]) ∧
      (getReport x = [SOME (SLc (PL report1))]) ∧
      (getInitMove x = [SOME (SLc (PSG initiateMovement))])
    then
      Report1
    else unAuthorized) ∧
   (planPBOut PLAN_PB (exec x) =
    if getPlCom x = receiveMission then ReceiveMission
```

```
     else unAuthorized) ∧
    (planPBOut RECEIVE_MISSION (exec x) =
     if getPlCom x = warno then Warno else unAuthorized) ∧
    (planPBOut REPORT1 (exec x) =
     if getPlCom x = completePlan then CompletePlan
     else unAuthorized) ∧
    (planPBOut COMPLETE_PLAN (exec x) =
     if getPlCom x = opoid then Opoid else unAuthorized) ∧
    (planPBOut OPOID (exec x) =
     if getPlCom x = supervise then Supervise
     else unAuthorized) ∧
    (planPBOut SUPERVISE (exec x) =
     if getPlCom x = report2 then Report2 else unAuthorized) ∧
    (planPBOut REPORT2 (exec x) =
     if getPlCom x = complete then Complete else unAuthorized) ∧
    (planPBOut s (trap v0) = unAuthorized) ∧
    (planPBOut s (discard v1) = unAuthenticated)
```

[planPBOut_ind]

```
⊢ ∀P.
    (∀x. P WARNO (exec x)) ∧ (∀x. P PLAN_PB (exec x)) ∧
    (∀x. P RECEIVE_MISSION (exec x)) ∧
    (∀x. P REPORT1 (exec x)) ∧ (∀x. P COMPLETE_PLAN (exec x)) ∧
    (∀x. P OPOID (exec x)) ∧ (∀x. P SUPERVISE (exec x)) ∧
    (∀x. P REPORT2 (exec x)) ∧ (∀s v0. P s (trap v0)) ∧
    (∀s v1. P s (discard v1)) ∧
    (∀v6. P TENTATIVE_PLAN (exec v6)) ∧
    (∀v7. P INITIATE_MOVEMENT (exec v7)) ∧
    (∀v8. P RECON (exec v8)) ∧ (∀v9. P COMPLETE (exec v9)) ⇒
    ∀v v1. P v v1
```

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_justified_lemma]

```
⊢ s ≠ WARNO ⇒
  plCommand ≠ invalidPlCommand ⇒
  plCommand ≠ report1 ⇒
  ∀NS Out M Oi Os.
    TR (M,Oi,Os)
      (exec
         (inputList
            [Name PlatoonLeader says
             prop (SOME (SLc (PL plCommand)))]))
      (CFG inputOK secContext secContextNull
         ([Name PlatoonLeader says
           prop (SOME (SLc (PL plCommand)))]::ins) s outs)
      (CFG inputOK secContext secContextNull ins
         (NS s
            (exec
               (inputList
                  [Name PlatoonLeader says
```

```
                    prop (SOME (SLc (PL plCommand)))])))
          (Out s
             (exec
                (inputList
                   [Name PlatoonLeader says
                    prop (SOME (SLc (PL plCommand)))])):::
              outs)) ⟺
     authenticationTest inputOK
       [Name PlatoonLeader says
        prop (SOME (SLc (PL plCommand)))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secContextNull
          ([Name PlatoonLeader says
            prop (SOME (SLc (PL plCommand)))]::ins) s outs) ∧
     (M,Oi,Os) satList
     propCommandList
       [Name PlatoonLeader says
        prop (SOME (SLc (PL plCommand)))]
```

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_justified_thm]

```
⊢ s ≠ WARNO ⇒
  plCommand ≠ invalidPlCommand ⇒
  plCommand ≠ report1 ⇒
  ∀ NS Out M Oi Os.
    TR (M,Oi,Os) (exec [SOME (SLc (PL plCommand))])
      (CFG inputOK secContext secContextNull
         ([Name PlatoonLeader says
           prop (SOME (SLc (PL plCommand)))]::ins) s outs)
      (CFG inputOK secContext secContextNull ins
         (NS s (exec [SOME (SLc (PL plCommand))]))
         (Out s (exec [SOME (SLc (PL plCommand))])::outs)) ⟺
    authenticationTest inputOK
      [Name PlatoonLeader says
       prop (SOME (SLc (PL plCommand)))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG inputOK secContext secContextNull
         ([Name PlatoonLeader says
           prop (SOME (SLc (PL plCommand)))]::ins) s outs) ∧
    (M,Oi,Os) satList [prop (SOME (SLc (PL plCommand)))]
```

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_lemma]

```
⊢ s ≠ WARNO ⇒
  plCommand ≠ invalidPlCommand ⇒
  plCommand ≠ report1 ⇒
  ∀ M Oi Os.
    CFGInterpret (M,Oi,Os)
      (CFG inputOK secContext secContextNull
         ([Name PlatoonLeader says
           prop (SOME (SLc (PL plCommand)))]::ins) s outs) ⇒
```

```
      (M,Oi,Os) satList
      propCommandList
        [Name PlatoonLeader says
         prop (SOME (SLc (PL plCommand)))]
```

[PlatoonLeader_psgCommand_notDiscard_thm]

```
 ⊢ ∀NS Out M Oi Os.
      ¬TR (M,Oi,Os)
         (discard
            (inputList
               [Name PlatoonLeader says
                prop (SOME (SLc (PSG psgCommand)))]))
         (CFG inputOK secContext secContextNull
            ([Name PlatoonLeader says
              prop (SOME (SLc (PSG psgCommand)))]::ins) s outs)
         (CFG inputOK secContext secContextNull ins
            (NS s
               (discard
                  (inputList
                     [Name PlatoonLeader says
                      prop (SOME (SLc (PSG psgCommand)))])))
            (Out s
               (discard
                  (inputList
                     [Name PlatoonLeader says
                      prop (SOME (SLc (PSG psgCommand)))]))::
                outs))
```

[PlatoonLeader_trap_psgCommand_justified_lemma]

```
 ⊢ ∀NS Out M Oi Os.
      TR (M,Oi,Os)
        (trap
           (inputList
              [Name PlatoonLeader says
               prop (SOME (SLc (PSG psgCommand)))]))
        (CFG inputOK secContext secContextNull
           ([Name PlatoonLeader says
             prop (SOME (SLc (PSG psgCommand)))]::ins) s outs)
        (CFG inputOK secContext secContextNull ins
           (NS s
              (trap
                 (inputList
                    [Name PlatoonLeader says
                     prop (SOME (SLc (PSG psgCommand)))])))
           (Out s
              (trap
                 (inputList
                    [Name PlatoonLeader says
                     prop (SOME (SLc (PSG psgCommand)))]))::
```

```
            outs)) ⟺
     authenticationTest inputOK
       [Name PlatoonLeader says
        prop (SOME (SLc (PSG psgCommand)))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secContextNull
          ([Name PlatoonLeader says
            prop (SOME (SLc (PSG psgCommand)))]::ins) s outs) ∧
     (M,Oi,Os) sat prop NONE
```

[PlatoonLeader_trap_psgCommand_lemma]

```
 ⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secContextNull
          ([Name PlatoonLeader says
            prop (SOME (SLc (PSG psgCommand)))]::ins) s outs) ⇒
     (M,Oi,Os) sat prop NONE
```

[PlatoonLeader_WARNO_exec_report1_justified_lemma]

```
 ⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os)
       (exec
          (inputList
             [Name PlatoonLeader says
              prop (SOME (SLc (PL recon)));
              Name PlatoonLeader says
              prop (SOME (SLc (PL tentativePlan)));
              Name PlatoonSergeant says
              prop (SOME (SLc (PSG initiateMovement)));
              Name PlatoonLeader says
              prop (SOME (SLc (PL report1)))]))
       (CFG inputOK secContext secContextNull
          ([Name PlatoonLeader says
            prop (SOME (SLc (PL recon)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL tentativePlan)));
            Name PlatoonSergeant says
            prop (SOME (SLc (PSG initiateMovement)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL report1)))]::ins) WARNO outs)
       (CFG inputOK secContext secContextNull ins
          (NS WARNO
             (exec
                (inputList
                   [Name PlatoonLeader says
                    prop (SOME (SLc (PL recon)));
                    Name PlatoonLeader says
                    prop (SOME (SLc (PL tentativePlan)));
                    Name PlatoonSergeant says
```

```
                  prop (SOME (SLc (PSG initiateMovement)));
                  Name PlatoonLeader says
                  prop (SOME (SLc (PL report1)))])))
        (Out WARNO
           (exec
              (inputList
                 [Name PlatoonLeader says
                  prop (SOME (SLc (PL recon)));
                  Name PlatoonLeader says
                  prop (SOME (SLc (PL tentativePlan)));
                  Name PlatoonSergeant says
                  prop (SOME (SLc (PSG initiateMovement)));
                  Name PlatoonLeader says
                  prop (SOME (SLc (PL report1)))])::outs)) ⟺
    authenticationTest inputOK
      [Name PlatoonLeader says prop (SOME (SLc (PL recon)));
       Name PlatoonLeader says
       prop (SOME (SLc (PL tentativePlan)));
       Name PlatoonSergeant says
       prop (SOME (SLc (PSG initiateMovement)));
       Name PlatoonLeader says
       prop (SOME (SLc (PL report1)))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG inputOK secContext secContextNull
         ([Name PlatoonLeader says
           prop (SOME (SLc (PL recon)));
           Name PlatoonLeader says
           prop (SOME (SLc (PL tentativePlan)));
           Name PlatoonSergeant says
           prop (SOME (SLc (PSG initiateMovement)));
           Name PlatoonLeader says
           prop (SOME (SLc (PL report1)))]::ins) WARNO outs) ∧
    (M,Oi,Os) satList
    propCommandList
      [Name PlatoonLeader says prop (SOME (SLc (PL recon)));
       Name PlatoonLeader says
       prop (SOME (SLc (PL tentativePlan)));
       Name PlatoonSergeant says
       prop (SOME (SLc (PSG initiateMovement)));
       Name PlatoonLeader says prop (SOME (SLc (PL report1)))]
```

[PlatoonLeader_WARNO_exec_report1_justified_thm]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os)
      (exec
         [SOME (SLc (PL recon)); SOME (SLc (PL tentativePlan));
          SOME (SLc (PSG initiateMovement));
          SOME (SLc (PL report1))])
      (CFG inputOK secContext secContextNull
```

```
          ([Name PlatoonLeader says
            prop (SOME (SLc (PL recon)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL tentativePlan)));
            Name PlatoonSergeant says
            prop (SOME (SLc (PSG initiateMovement)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL report1)))]::ins) WARNO outs)
        (CFG inputOK secContext secContextNull ins
          (NS WARNO
             (exec
                [SOME (SLc (PL recon));
                 SOME (SLc (PL tentativePlan));
                 SOME (SLc (PSG initiateMovement));
                 SOME (SLc (PL report1))]))
          (Out WARNO
             (exec
                [SOME (SLc (PL recon));
                 SOME (SLc (PL tentativePlan));
                 SOME (SLc (PSG initiateMovement));
                 SOME (SLc (PL report1))])::outs)) ⟺
      authenticationTest inputOK
        [Name PlatoonLeader says prop (SOME (SLc (PL recon)));
         Name PlatoonLeader says
         prop (SOME (SLc (PL tentativePlan)));
         Name PlatoonSergeant says
         prop (SOME (SLc (PSG initiateMovement)));
         Name PlatoonLeader says
         prop (SOME (SLc (PL report1)))] ∧
      CFGInterpret (M,Oi,Os)
        (CFG inputOK secContext secContextNull
          ([Name PlatoonLeader says
            prop (SOME (SLc (PL recon)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL tentativePlan)));
            Name PlatoonSergeant says
            prop (SOME (SLc (PSG initiateMovement)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL report1)))]::ins) WARNO outs) ∧
      (M,Oi,Os) satList
      [prop (SOME (SLc (PL recon)));
       prop (SOME (SLc (PL tentativePlan)));
       prop (SOME (SLc (PSG initiateMovement)));
       prop (SOME (SLc (PL report1)))]
```

[PlatoonLeader_WARNO_exec_report1_lemma]

```
 ⊢ ∀M Oi Os.
      CFGInterpret (M,Oi,Os)
        (CFG inputOK secContext secContextNull
```

```
          ([Name PlatoonLeader says
            prop (SOME (SLc (PL recon)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL tentativePlan)));
            Name PlatoonSergeant says
            prop (SOME (SLc (PSG initiateMovement)));
            Name PlatoonLeader says
            prop (SOME (SLc (PL report1)))]::ins) WARNO outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name PlatoonLeader says prop (SOME (SLc (PL recon)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL tentativePlan)));
        Name PlatoonSergeant says
        prop (SOME (SLc (PSG initiateMovement)));
        Name PlatoonLeader says prop (SOME (SLc (PL report1)))]
```

[PlatoonSergeant_trap_plCommand_justified_lemma]

```
⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os)
       (trap
          (inputList
             [Name PlatoonSergeant says
              prop (SOME (SLc (PL plCommand)))]))
       (CFG inputOK secContext secContextNull
          ([Name PlatoonSergeant says
            prop (SOME (SLc (PL plCommand)))]::ins) s outs)
       (CFG inputOK secContext secContextNull ins
          (NS s
             (trap
                (inputList
                   [Name PlatoonSergeant says
                    prop (SOME (SLc (PL plCommand)))]))))
          (Out s
             (trap
                (inputList
                   [Name PlatoonSergeant says
                    prop (SOME (SLc (PL plCommand)))]))::
               outs)) ⟺
     authenticationTest inputOK
       [Name PlatoonSergeant says
        prop (SOME (SLc (PL plCommand)))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputOK secContext secContextNull
          ([Name PlatoonSergeant says
            prop (SOME (SLc (PL plCommand)))]::ins) s outs) ∧
     (M,Oi,Os) sat prop NONE
```

[PlatoonSergeant_trap_plCommand_justified_thm]

$\vdash \forall NS\ Out\ M\ Oi\ Os.$
TR (M, Oi, Os) (trap [SOME (SLc (PL $plCommand$))])
(CFG inputOK secContext secContextNull
([Name PlatoonSergeant says
prop (SOME (SLc (PL $plCommand$)))]::ins) s $outs$)
(CFG inputOK secContext secContextNull ins
(NS s (trap [SOME (SLc (PL $plCommand$))]))
(Out s (trap [SOME (SLc (PL $plCommand$))])::$outs$)) $\iff$
authenticationTest inputOK
[Name PlatoonSergeant says
prop (SOME (SLc (PL $plCommand$)))] $\wedge$
CFGInterpret (M, Oi, Os)
(CFG inputOK secContext secContextNull
([Name PlatoonSergeant says
prop (SOME (SLc (PL $plCommand$)))]::ins) s $outs$) $\wedge$
(M, Oi, Os) sat prop NONE

[PlatoonSergeant_trap_plCommand_lemma]

$\vdash \forall M\ Oi\ Os.$
CFGInterpret (M, Oi, Os)
(CFG inputOK secContext secContextNull
([Name PlatoonSergeant says
prop (SOME (SLc (PL $plCommand$)))]::ins) s $outs$) $\Rightarrow$
(M, Oi, Os) sat prop NONE

Index