Contents

1 PlanPBType Theory

Built: 10 June 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: 10 June 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]

```
⊢ ∀ 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)) ⟺
    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_lemma]

```
⊢ ∀ M Oi Os.
    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
```

3 PlanPBDef Theory

Built: 10 June 2018

Parent Theories: PlanPBType, aclfoundation, OMNIType

3.1 Definitions

[PL_notWARNO_Auth_def]

```
⊢ ∀ cmd.
    PL_notWARNO_Auth cmd =
    if cmd = report1 then prop NONE
    else
      Name PlatoonLeader says prop (SOME (SLc (PL cmd))) impf
      Name PlatoonLeader controls prop (SOME (SLc (PL cmd)))
```

[PL_WARNO_Auth_def]

```
⊢ PL_WARNO_Auth =
  prop (SOME (SLc (PL recon))) impf
  prop (SOME (SLc (PL tentativePlan))) impf
  prop (SOME (SLc (PSG initiateMovement))) impf
  Name PlatoonLeader controls prop (SOME (SLc (PL report1)))
```

[secContext_def]

```
⊢ ∀s x.
    secContext s x =
    if s = WARNO then
      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
        [PL_WARNO_Auth;
         Name PlatoonLeader controls
         prop (SOME (SLc (PL recon)));
         Name PlatoonLeader controls
         prop (SOME (SLc (PL tentativePlan)));
         Name PlatoonSergeant controls
         prop (SOME (SLc (PSG initiateMovement)))]
      else [prop NONE]
    else if getPlCom x = invalidPlCommand then [prop NONE]
    else [PL_notWARNO_Auth (getPlCom x)]
```

[secContextNull_def]

```
⊢ ∀x. secContextNull x = [TT]
```

3.2 Theorems

[getInitMove_def]

```
⊢ (getInitMove [] = [NONE]) ∧
  (∀xs.
     getInitMove
       (Name PlatoonSergeant says
        prop (SOME (SLc (PSG initiateMovement)))::xs) =
     [SOME (SLc (PSG initiateMovement))]) ∧
  (∀xs. getInitMove (TT::xs) = getInitMove xs) ∧
  (∀xs. getInitMove (FF::xs) = getInitMove xs) ∧
  (∀xs v2. getInitMove (prop v2::xs) = getInitMove xs) ∧
  (∀xs v3. getInitMove (notf v3::xs) = getInitMove xs) ∧
  (∀xs v5 v4. getInitMove (v4 andf v5::xs) = getInitMove xs) ∧
  (∀xs v7 v6. getInitMove (v6 orf v7::xs) = getInitMove xs) ∧
  (∀xs v9 v8. getInitMove (v8 impf v9::xs) = getInitMove xs) ∧
  (∀xs v11 v10.
     getInitMove (v10 eqf v11::xs) = getInitMove xs) ∧
  (∀xs v12. getInitMove (v12 says TT::xs) = getInitMove xs) ∧
  (∀xs v12. getInitMove (v12 says FF::xs) = getInitMove xs) ∧
  (∀xs v134.
     getInitMove (Name v134 says prop NONE::xs) =
     getInitMove xs) ∧
  (∀xs v144.
```

```
   getInitMove
     (Name PlatoonLeader says prop (SOME v144)::xs) =
   getInitMove xs) ∧
(∀xs v146.
   getInitMove
     (Name PlatoonSergeant says prop (SOME (ESCc v146))::
          xs) =
   getInitMove xs) ∧
(∀xs v150.
   getInitMove
     (Name PlatoonSergeant says prop (SOME (SLc (PL v150)))::
          xs) =
   getInitMove xs) ∧
(∀xs.
   getInitMove
     (Name PlatoonSergeant says
      prop (SOME (SLc (PSG psgIncomplete)))::xs) =
   getInitMove xs) ∧
(∀xs.
   getInitMove
     (Name PlatoonSergeant says
      prop (SOME (SLc (PSG invalidPsgCommand)))::xs) =
   getInitMove xs) ∧
(∀xs v68 v136 v135.
   getInitMove (v135 meet v136 says prop v68::xs) =
   getInitMove xs) ∧
(∀xs v68 v138 v137.
   getInitMove (v137 quoting v138 says prop v68::xs) =
   getInitMove xs) ∧
(∀xs v69 v12.
   getInitMove (v12 says notf v69::xs) = getInitMove xs) ∧
(∀xs v71 v70 v12.
   getInitMove (v12 says (v70 andf v71)::xs) =
   getInitMove xs) ∧
(∀xs v73 v72 v12.
   getInitMove (v12 says (v72 orf v73)::xs) =
   getInitMove xs) ∧
(∀xs v75 v74 v12.
   getInitMove (v12 says (v74 impf v75)::xs) =
   getInitMove xs) ∧
(∀xs v77 v76 v12.
   getInitMove (v12 says (v76 eqf v77)::xs) =
   getInitMove xs) ∧
(∀xs v79 v78 v12.
   getInitMove (v12 says v78 says v79::xs) =
   getInitMove xs) ∧
(∀xs v81 v80 v12.
   getInitMove (v12 says v80 speaks_for v81::xs) =
   getInitMove xs) ∧
```

```
(∀ xs v83 v82 v12.
   getInitMove (v12 says v82 controls v83::xs) =
   getInitMove xs) ∧
(∀ xs v86 v85 v84 v12.
   getInitMove (v12 says reps v84 v85 v86::xs) =
   getInitMove xs) ∧
(∀ xs v88 v87 v12.
   getInitMove (v12 says v87 domi v88::xs) =
   getInitMove xs) ∧
(∀ xs v90 v89 v12.
   getInitMove (v12 says v89 eqi v90::xs) = getInitMove xs) ∧
(∀ xs v92 v91 v12.
   getInitMove (v12 says v91 doms v92::xs) =
   getInitMove xs) ∧
(∀ xs v94 v93 v12.
   getInitMove (v12 says v93 eqs v94::xs) = getInitMove xs) ∧
(∀ xs v96 v95 v12.
   getInitMove (v12 says v95 eqn v96::xs) = getInitMove xs) ∧
(∀ xs v98 v97 v12.
   getInitMove (v12 says v97 lte v98::xs) = getInitMove xs) ∧
(∀ xs v99 v12 v100.
   getInitMove (v12 says v99 lt v100::xs) = getInitMove xs) ∧
(∀ xs v15 v14.
   getInitMove (v14 speaks_for v15::xs) = getInitMove xs) ∧
(∀ xs v17 v16.
   getInitMove (v16 controls v17::xs) = getInitMove xs) ∧
(∀ xs v20 v19 v18.
   getInitMove (reps v18 v19 v20::xs) = getInitMove xs) ∧
(∀ xs v22 v21.
   getInitMove (v21 domi v22::xs) = getInitMove xs) ∧
(∀ xs v24 v23.
   getInitMove (v23 eqi v24::xs) = getInitMove xs) ∧
(∀ xs v26 v25.
   getInitMove (v25 doms v26::xs) = getInitMove xs) ∧
(∀ xs v28 v27.
   getInitMove (v27 eqs v28::xs) = getInitMove xs) ∧
(∀ xs v30 v29.
   getInitMove (v29 eqn v30::xs) = getInitMove xs) ∧
(∀ xs v32 v31.
   getInitMove (v31 lte v32::xs) = getInitMove xs) ∧
∀ xs v34 v33. getInitMove (v33 lt v34::xs) = getInitMove xs
```

[getInitMove_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ xs.
       P
         (Name PlatoonSergeant says
          prop (SOME (SLc (PSG initiateMovement)))::xs)) ∧
```

$(\forall xs.\ P\ xs \Rightarrow P\ (\texttt{TT::}xs)) \wedge (\forall xs.\ P\ xs \Rightarrow P\ (\texttt{FF::}xs)) \wedge$
$(\forall v_2\ xs.\ P\ xs \Rightarrow P\ (\texttt{prop}\ v_2\texttt{::}xs)) \wedge$
$(\forall v_3\ xs.\ P\ xs \Rightarrow P\ (\texttt{notf}\ v_3\texttt{::}xs)) \wedge$
$(\forall v_4\ v_5\ xs.\ P\ xs \Rightarrow P\ (v_4\ \texttt{andf}\ v_5\texttt{::}xs)) \wedge$
$(\forall v_6\ v_7\ xs.\ P\ xs \Rightarrow P\ (v_6\ \texttt{orf}\ v_7\texttt{::}xs)) \wedge$
$(\forall v_8\ v_9\ xs.\ P\ xs \Rightarrow P\ (v_8\ \texttt{impf}\ v_9\texttt{::}xs)) \wedge$
$(\forall v_{10}\ v_{11}\ xs.\ P\ xs \Rightarrow P\ (v_{10}\ \texttt{eqf}\ v_{11}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says TT::}xs)) \wedge$
$(\forall v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says FF::}xs)) \wedge$
$(\forall v134\ xs.\ P\ xs \Rightarrow P\ (\texttt{Name}\ v134\ \texttt{says prop NONE::}xs)) \wedge$
$(\forall v144\ xs.$
$\quad P\ xs \Rightarrow$
$\quad P\ (\texttt{Name PlatoonLeader says prop (SOME}\ v144\texttt{)::}xs)) \wedge$
$(\forall v146\ xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad (\texttt{Name PlatoonSergeant says prop (SOME (ESCc}\ v146\texttt{))::}$
$\quad\quad\quad xs)) \wedge$
$(\forall v150\ xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad (\texttt{Name PlatoonSergeant says}$
$\quad\quad\ \texttt{prop (SOME (SLc (PL}\ v150\texttt{)))::}xs)) \wedge$
$(\forall xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad (\texttt{Name PlatoonSergeant says}$
$\quad\quad\ \texttt{prop (SOME (SLc (PSG psgIncomplete)))::}xs)) \wedge$
$(\forall xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad (\texttt{Name PlatoonSergeant says}$
$\quad\quad\ \texttt{prop (SOME (SLc (PSG invalidPsgCommand)))::}xs)) \wedge$
$(\forall v135\ v136\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P\ (v135\ \texttt{meet}\ v136\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge$
$(\forall v137\ v138\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P\ (v137\ \texttt{quoting}\ v138\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says notf}\ v_{69}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{70}\ \texttt{andf}\ v_{71}\texttt{)::}xs)) \wedge$
$(\forall v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{72}\ \texttt{orf}\ v_{73}\texttt{)::}xs)) \wedge$
$(\forall v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{74}\ \texttt{impf}\ v_{75}\texttt{)::}xs)) \wedge$
$(\forall v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{76}\ \texttt{eqf}\ v_{77}\texttt{)::}xs)) \wedge$
$(\forall v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{78}\ \texttt{says}\ v_{79}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ v_{80}\ v_{81}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{80}\ \texttt{speaks\_for}\ v_{81}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ v_{82}\ v_{83}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{82}\ \texttt{controls}\ v_{83}\texttt{::}xs)) \wedge$
$(\forall v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says reps}\ v_{84}\ v_{85}\ v_{86}\texttt{::}xs)) \wedge$

```
   (∀ v12 v87 v88 xs. P xs ⇒ P (v12 says v87 domi v88::xs)) ∧
   (∀ v12 v89 v90 xs. P xs ⇒ P (v12 says v89 eqi v90::xs)) ∧
   (∀ v12 v91 v92 xs. P xs ⇒ P (v12 says v91 doms v92::xs)) ∧
   (∀ v12 v93 v94 xs. P xs ⇒ P (v12 says v93 eqs v94::xs)) ∧
   (∀ v12 v95 v96 xs. P xs ⇒ P (v12 says v95 eqn v96::xs)) ∧
   (∀ v12 v97 v98 xs. P xs ⇒ P (v12 says v97 lte v98::xs)) ∧
   (∀ v12 v99 v100 xs. P xs ⇒ P (v12 says v99 lt v100::xs)) ∧
   (∀ v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
   (∀ v16 v17 xs. P xs ⇒ P (v16 controls v17::xs)) ∧
   (∀ v18 v19 v20 xs. P xs ⇒ P (reps v18 v19 v20::xs)) ∧
   (∀ v21 v22 xs. P xs ⇒ P (v21 domi v22::xs)) ∧
   (∀ v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
   (∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
   (∀ 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
```

[getPlCom_def]

```
⊢ (getPlCom [] = invalidPlCommand) ∧
  (∀ xs cmd.
     getPlCom
       (Name PlatoonLeader says prop (SOME (SLc (PL cmd)))::
            xs) =
     cmd) ∧ (∀ xs. getPlCom (TT::xs) = getPlCom xs) ∧
  (∀ xs. getPlCom (FF::xs) = getPlCom xs) ∧
  (∀ xs v2. getPlCom (prop v2::xs) = getPlCom xs) ∧
  (∀ xs v3. getPlCom (notf v3::xs) = getPlCom xs) ∧
  (∀ xs v5 v4. getPlCom (v4 andf v5::xs) = getPlCom xs) ∧
  (∀ xs v7 v6. getPlCom (v6 orf v7::xs) = getPlCom xs) ∧
  (∀ xs v9 v8. getPlCom (v8 impf v9::xs) = getPlCom xs) ∧
  (∀ xs v11 v10. getPlCom (v10 eqf v11::xs) = getPlCom xs) ∧
  (∀ xs v12. getPlCom (v12 says TT::xs) = getPlCom xs) ∧
  (∀ xs v12. getPlCom (v12 says FF::xs) = getPlCom xs) ∧
  (∀ xs v134.
     getPlCom (Name v134 says prop NONE::xs) = getPlCom xs) ∧
  (∀ xs v146.
     getPlCom
       (Name PlatoonLeader says prop (SOME (ESCc v146))::xs) =
     getPlCom xs) ∧
  (∀ xs v151.
     getPlCom
       (Name PlatoonLeader says prop (SOME (SLc (PSG v151)))::
            xs) =
     getPlCom xs) ∧
  (∀ xs v144.
     getPlCom
       (Name PlatoonSergeant says prop (SOME v144)::xs) =
```

```
   getPlCom xs) ∧
(∀ xs v68 v136 v135.
   getPlCom (v135 meet v136 says prop v68::xs) =
   getPlCom xs) ∧
(∀ xs v68 v138 v137.
   getPlCom (v137 quoting v138 says prop v68::xs) =
   getPlCom xs) ∧
(∀ xs v69 v12.
   getPlCom (v12 says notf v69::xs) = getPlCom xs) ∧
(∀ xs v71 v70 v12.
   getPlCom (v12 says (v70 andf v71)::xs) = getPlCom xs) ∧
(∀ xs v73 v72 v12.
   getPlCom (v12 says (v72 orf v73)::xs) = getPlCom xs) ∧
(∀ xs v75 v74 v12.
   getPlCom (v12 says (v74 impf v75)::xs) = getPlCom xs) ∧
(∀ xs v77 v76 v12.
   getPlCom (v12 says (v76 eqf v77)::xs) = getPlCom xs) ∧
(∀ xs v79 v78 v12.
   getPlCom (v12 says v78 says v79::xs) = getPlCom xs) ∧
(∀ xs v81 v80 v12.
   getPlCom (v12 says v80 speaks_for v81::xs) =
   getPlCom xs) ∧
(∀ xs v83 v82 v12.
   getPlCom (v12 says v82 controls v83::xs) = getPlCom xs) ∧
(∀ xs v86 v85 v84 v12.
   getPlCom (v12 says reps v84 v85 v86::xs) = getPlCom xs) ∧
(∀ xs v88 v87 v12.
   getPlCom (v12 says v87 domi v88::xs) = getPlCom xs) ∧
(∀ xs v90 v89 v12.
   getPlCom (v12 says v89 eqi v90::xs) = getPlCom xs) ∧
(∀ xs v92 v91 v12.
   getPlCom (v12 says v91 doms v92::xs) = getPlCom xs) ∧
(∀ xs v94 v93 v12.
   getPlCom (v12 says v93 eqs v94::xs) = getPlCom xs) ∧
(∀ xs v96 v95 v12.
   getPlCom (v12 says v95 eqn v96::xs) = getPlCom xs) ∧
(∀ xs v98 v97 v12.
   getPlCom (v12 says v97 lte v98::xs) = getPlCom xs) ∧
(∀ xs v99 v12 v100.
   getPlCom (v12 says v99 lt v100::xs) = getPlCom xs) ∧
(∀ xs v15 v14.
   getPlCom (v14 speaks_for v15::xs) = getPlCom xs) ∧
(∀ xs v17 v16.
   getPlCom (v16 controls v17::xs) = getPlCom xs) ∧
(∀ xs v20 v19 v18.
   getPlCom (reps v18 v19 v20::xs) = getPlCom xs) ∧
(∀ xs v22 v21. getPlCom (v21 domi v22::xs) = getPlCom xs) ∧
(∀ xs v24 v23. getPlCom (v23 eqi v24::xs) = getPlCom xs) ∧
(∀ xs v26 v25. getPlCom (v25 doms v26::xs) = getPlCom xs) ∧
```

$(\forall\, xs\ v_{28}\ v_{27}.\ \texttt{getPlCom}\ (v_{27}\ \texttt{eqs}\ v_{28}\texttt{::}xs) = \texttt{getPlCom}\ xs) \wedge$
$(\forall\, xs\ v_{30}\ v_{29}.\ \texttt{getPlCom}\ (v_{29}\ \texttt{eqn}\ v_{30}\texttt{::}xs) = \texttt{getPlCom}\ xs) \wedge$
$(\forall\, xs\ v_{32}\ v_{31}.\ \texttt{getPlCom}\ (v_{31}\ \texttt{lte}\ v_{32}\texttt{::}xs) = \texttt{getPlCom}\ xs) \wedge$
$\forall\, xs\ v_{34}\ v_{33}.\ \texttt{getPlCom}\ (v_{33}\ \texttt{lt}\ v_{34}\texttt{::}xs) = \texttt{getPlCom}\ xs$

[getPlCom_ind]

$\vdash \forall P.$
$P\ \texttt{[]} \wedge$
$(\forall\, cmd\ xs.$
P
$(\texttt{Name PlatoonLeader says prop (SOME (SLc (PL}\ cmd\texttt{)))::}$
$xs\texttt{))} \wedge (\forall\, xs.\ P\ xs \Rightarrow P\ \texttt{(TT::}xs\texttt{)}) \wedge$
$(\forall\, xs.\ P\ xs \Rightarrow P\ \texttt{(FF::}xs\texttt{)}) \wedge$
$(\forall\, v_2\ xs.\ P\ xs \Rightarrow P\ \texttt{(prop}\ v_2\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_3\ xs.\ P\ xs \Rightarrow P\ \texttt{(notf}\ v_3\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_4\ v_5\ xs.\ P\ xs \Rightarrow P\ (v_4\ \texttt{andf}\ v_5\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_6\ v_7\ xs.\ P\ xs \Rightarrow P\ (v_6\ \texttt{orf}\ v_7\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_8\ v_9\ xs.\ P\ xs \Rightarrow P\ (v_8\ \texttt{impf}\ v_9\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_{10}\ v_{11}\ xs.\ P\ xs \Rightarrow P\ (v_{10}\ \texttt{eqf}\ v_{11}\texttt{::}xs\texttt{)}) \wedge$
$(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says TT::}xs\texttt{)}) \wedge$
$(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says FF::}xs\texttt{)}) \wedge$
$(\forall\, v134\ xs.\ P\ xs \Rightarrow P\ \texttt{(Name}\ v134\ \texttt{says prop NONE::}xs\texttt{)}) \wedge$
$(\forall\, v146\ xs.$
$P\ xs \Rightarrow$
P
$\texttt{(Name PlatoonLeader says prop (SOME (ESCc}\ v146\texttt{))::}$
$xs\texttt{))} \wedge$
$(\forall\, v151\ xs.$
$P\ xs \Rightarrow$
P
$\texttt{(Name PlatoonLeader says}$
$\texttt{prop (SOME (SLc (PSG}\ v151\texttt{)))::}xs\texttt{))} \wedge$
$(\forall\, v144\ xs.$
$P\ xs \Rightarrow$
$P\ \texttt{(Name PlatoonSergeant says prop (SOME}\ v144\texttt{)::}xs\texttt{))} \wedge$
$(\forall\, v135\ v136\ v_{68}\ xs.$
$P\ xs \Rightarrow P\ (v135\ \texttt{meet}\ v136\ \texttt{says prop}\ v_{68}\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v137\ v138\ v_{68}\ xs.$
$P\ xs \Rightarrow P\ (v137\ \texttt{quoting}\ v138\ \texttt{says prop}\ v_{68}\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says notf}\ v_{69}\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ (v_{70}\ \texttt{andf}\ v_{71})\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ (v_{72}\ \texttt{orf}\ v_{73})\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ (v_{74}\ \texttt{impf}\ v_{75})\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ (v_{76}\ \texttt{eqf}\ v_{77})\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{78}\ \texttt{says}\ v_{79}\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{80}\ v_{81}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{80}\ \texttt{speaks\_for}\ v_{81}\texttt{::}xs\texttt{))} \wedge$
$(\forall\, v_{12}\ v_{82}\ v_{83}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{82}\ \texttt{controls}\ v_{83}\texttt{::}xs\texttt{))} \wedge$

```
(∀ v12 v84 v85 v86 xs.
   P xs ⇒ P (v12 says reps v84 v85 v86::xs)) ∧
(∀ v12 v87 v88 xs. P xs ⇒ P (v12 says v87 domi v88::xs)) ∧
(∀ v12 v89 v90 xs. P xs ⇒ P (v12 says v89 eqi v90::xs)) ∧
(∀ v12 v91 v92 xs. P xs ⇒ P (v12 says v91 doms v92::xs)) ∧
(∀ v12 v93 v94 xs. P xs ⇒ P (v12 says v93 eqs v94::xs)) ∧
(∀ v12 v95 v96 xs. P xs ⇒ P (v12 says v95 eqn v96::xs)) ∧
(∀ v12 v97 v98 xs. P xs ⇒ P (v12 says v97 lte v98::xs)) ∧
(∀ v12 v99 v100 xs. P xs ⇒ P (v12 says v99 lt v100::xs)) ∧
(∀ v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
(∀ v16 v17 xs. P xs ⇒ P (v16 controls v17::xs)) ∧
(∀ v18 v19 v20 xs. P xs ⇒ P (reps v18 v19 v20::xs)) ∧
(∀ v21 v22 xs. P xs ⇒ P (v21 domi v22::xs)) ∧
(∀ v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
(∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
(∀ 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
```

[getPsgCom_def]

```
⊢ (getPsgCom [] = invalidPsgCommand) ∧
  (∀ xs cmd.
     getPsgCom
       (Name PlatoonSergeant says prop (SOME (SLc (PSG cmd)))::
            xs) =
     cmd) ∧ (∀ xs. getPsgCom (TT::xs) = getPsgCom xs) ∧
  (∀ xs. getPsgCom (FF::xs) = getPsgCom xs) ∧
  (∀ xs v2. getPsgCom (prop v2::xs) = getPsgCom xs) ∧
  (∀ xs v3. getPsgCom (notf v3::xs) = getPsgCom xs) ∧
  (∀ xs v5 v4. getPsgCom (v4 andf v5::xs) = getPsgCom xs) ∧
  (∀ xs v7 v6. getPsgCom (v6 orf v7::xs) = getPsgCom xs) ∧
  (∀ xs v9 v8. getPsgCom (v8 impf v9::xs) = getPsgCom xs) ∧
  (∀ xs v11 v10. getPsgCom (v10 eqf v11::xs) = getPsgCom xs) ∧
  (∀ xs v12. getPsgCom (v12 says TT::xs) = getPsgCom xs) ∧
  (∀ xs v12. getPsgCom (v12 says FF::xs) = getPsgCom xs) ∧
  (∀ xs v134.
     getPsgCom (Name v134 says prop NONE::xs) = getPsgCom xs) ∧
  (∀ xs v144.
     getPsgCom (Name PlatoonLeader says prop (SOME v144)::xs) =
     getPsgCom xs) ∧
  (∀ xs v146.
     getPsgCom
       (Name PlatoonSergeant says prop (SOME (ESCc v146))::
            xs) =
     getPsgCom xs) ∧
  (∀ xs v150.
     getPsgCom
```

```
     (Name PlatoonSergeant says prop (SOME (SLc (PL v150))))::
          xs) =
   getPsgCom xs) ∧
(∀ xs v68 v136 v135.
   getPsgCom (v135 meet v136 says prop v68::xs) =
   getPsgCom xs) ∧
(∀ xs v68 v138 v137.
   getPsgCom (v137 quoting v138 says prop v68::xs) =
   getPsgCom xs) ∧
(∀ xs v69 v12.
   getPsgCom (v12 says notf v69::xs) = getPsgCom xs) ∧
(∀ xs v71 v70 v12.
   getPsgCom (v12 says (v70 andf v71)::xs) = getPsgCom xs) ∧
(∀ xs v73 v72 v12.
   getPsgCom (v12 says (v72 orf v73)::xs) = getPsgCom xs) ∧
(∀ xs v75 v74 v12.
   getPsgCom (v12 says (v74 impf v75)::xs) = getPsgCom xs) ∧
(∀ xs v77 v76 v12.
   getPsgCom (v12 says (v76 eqf v77)::xs) = getPsgCom xs) ∧
(∀ xs v79 v78 v12.
   getPsgCom (v12 says v78 says v79::xs) = getPsgCom xs) ∧
(∀ xs v81 v80 v12.
   getPsgCom (v12 says v80 speaks_for v81::xs) =
   getPsgCom xs) ∧
(∀ xs v83 v82 v12.
   getPsgCom (v12 says v82 controls v83::xs) =
   getPsgCom xs) ∧
(∀ xs v86 v85 v84 v12.
   getPsgCom (v12 says reps v84 v85 v86::xs) =
   getPsgCom xs) ∧
(∀ xs v88 v87 v12.
   getPsgCom (v12 says v87 domi v88::xs) = getPsgCom xs) ∧
(∀ xs v90 v89 v12.
   getPsgCom (v12 says v89 eqi v90::xs) = getPsgCom xs) ∧
(∀ xs v92 v91 v12.
   getPsgCom (v12 says v91 doms v92::xs) = getPsgCom xs) ∧
(∀ xs v94 v93 v12.
   getPsgCom (v12 says v93 eqs v94::xs) = getPsgCom xs) ∧
(∀ xs v96 v95 v12.
   getPsgCom (v12 says v95 eqn v96::xs) = getPsgCom xs) ∧
(∀ xs v98 v97 v12.
   getPsgCom (v12 says v97 lte v98::xs) = getPsgCom xs) ∧
(∀ xs v99 v12 v100.
   getPsgCom (v12 says v99 lt v100::xs) = getPsgCom xs) ∧
(∀ xs v15 v14.
   getPsgCom (v14 speaks_for v15::xs) = getPsgCom xs) ∧
(∀ xs v17 v16.
   getPsgCom (v16 controls v17::xs) = getPsgCom xs) ∧
(∀ xs v20 v19 v18.
```

```
    getPsgCom (reps v18 v19 v20::xs) = getPsgCom xs) ∧
  (∀ xs v22 v21. getPsgCom (v21 domi v22::xs) = getPsgCom xs) ∧
  (∀ xs v24 v23. getPsgCom (v23 eqi v24::xs) = getPsgCom xs) ∧
  (∀ xs v26 v25. getPsgCom (v25 doms v26::xs) = getPsgCom xs) ∧
  (∀ xs v28 v27. getPsgCom (v27 eqs v28::xs) = getPsgCom xs) ∧
  (∀ xs v30 v29. getPsgCom (v29 eqn v30::xs) = getPsgCom xs) ∧
  (∀ xs v32 v31. getPsgCom (v31 lte v32::xs) = getPsgCom xs) ∧
  ∀ xs v34 v33. getPsgCom (v33 lt v34::xs) = getPsgCom xs
```

[getPsgCom_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ cmd xs.
       P
         (Name PlatoonSergeant says
          prop (SOME (SLc (PSG 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 xs. P xs ⇒ P (v12 says TT::xs)) ∧
    (∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
    (∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
    (∀ v144 xs.
       P xs ⇒
       P (Name PlatoonLeader says prop (SOME v144)::xs)) ∧
    (∀ v146 xs.
       P xs ⇒
       P
         (Name PlatoonSergeant says prop (SOME (ESCc v146))::
            xs)) ∧
    (∀ v150 xs.
       P xs ⇒
       P
         (Name PlatoonSergeant says
          prop (SOME (SLc (PL v150)))::xs)) ∧
    (∀ v135 v136 v68 xs.
       P xs ⇒ P (v135 meet v136 says prop v68::xs)) ∧
    (∀ v137 v138 v68 xs.
       P xs ⇒ P (v137 quoting v138 says prop v68::xs)) ∧
    (∀ v12 v69 xs. P xs ⇒ P (v12 says notf v69::xs)) ∧
    (∀ v12 v70 v71 xs. P xs ⇒ P (v12 says (v70 andf v71)::xs)) ∧
    (∀ v12 v72 v73 xs. P xs ⇒ P (v12 says (v72 orf v73)::xs)) ∧
    (∀ v12 v74 v75 xs. P xs ⇒ P (v12 says (v74 impf v75)::xs)) ∧
    (∀ v12 v76 v77 xs. P xs ⇒ P (v12 says (v76 eqf v77)::xs)) ∧
    (∀ v12 v78 v79 xs. P xs ⇒ P (v12 says v78 says v79::xs)) ∧
```

```
   (∀ v12 v80 v81 xs.
      P xs ⇒ P (v12 says v80 speaks_for v81::xs)) ∧
   (∀ v12 v82 v83 xs.
      P xs ⇒ P (v12 says v82 controls v83::xs)) ∧
   (∀ v12 v84 v85 v86 xs.
      P xs ⇒ P (v12 says reps v84 v85 v86::xs)) ∧
   (∀ v12 v87 v88 xs. P xs ⇒ P (v12 says v87 domi v88::xs)) ∧
   (∀ v12 v89 v90 xs. P xs ⇒ P (v12 says v89 eqi v90::xs)) ∧
   (∀ v12 v91 v92 xs. P xs ⇒ P (v12 says v91 doms v92::xs)) ∧
   (∀ v12 v93 v94 xs. P xs ⇒ P (v12 says v93 eqs v94::xs)) ∧
   (∀ v12 v95 v96 xs. P xs ⇒ P (v12 says v95 eqn v96::xs)) ∧
   (∀ v12 v97 v98 xs. P xs ⇒ P (v12 says v97 lte v98::xs)) ∧
   (∀ v12 v99 v100 xs. P xs ⇒ P (v12 says v99 lt v100::xs)) ∧
   (∀ v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
   (∀ v16 v17 xs. P xs ⇒ P (v16 controls v17::xs)) ∧
   (∀ v18 v19 v20 xs. P xs ⇒ P (reps v18 v19 v20::xs)) ∧
   (∀ v21 v22 xs. P xs ⇒ P (v21 domi v22::xs)) ∧
   (∀ v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
   (∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
   (∀ 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
```

[getRecon_def]

```
⊢ (getRecon [] = [NONE]) ∧
  (∀ xs.
     getRecon
       (Name PlatoonLeader says prop (SOME (SLc (PL recon)))::
            xs) =
     [SOME (SLc (PL recon))]) ∧
  (∀ xs. getRecon (TT::xs) = getRecon xs) ∧
  (∀ xs. getRecon (FF::xs) = getRecon xs) ∧
  (∀ xs v2. getRecon (prop v2::xs) = getRecon xs) ∧
  (∀ xs v3. getRecon (notf v3::xs) = getRecon xs) ∧
  (∀ xs v5 v4. getRecon (v4 andf v5::xs) = getRecon xs) ∧
  (∀ xs v7 v6. getRecon (v6 orf v7::xs) = getRecon xs) ∧
  (∀ xs v9 v8. getRecon (v8 impf v9::xs) = getRecon xs) ∧
  (∀ xs v11 v10. getRecon (v10 eqf v11::xs) = getRecon xs) ∧
  (∀ xs v12. getRecon (v12 says TT::xs) = getRecon xs) ∧
  (∀ xs v12. getRecon (v12 says FF::xs) = getRecon xs) ∧
  (∀ xs v134.
     getRecon (Name v134 says prop NONE::xs) = getRecon xs) ∧
  (∀ xs v146.
     getRecon
       (Name PlatoonLeader says prop (SOME (ESCc v146))::xs) =
     getRecon xs) ∧
  (∀ xs.
```

```
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL receiveMission)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says prop (SOME (SLc (PL warno)))::
          xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL tentativePlan)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report1)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says prop (SOME (SLc (PL opoid)))::
          xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL complete)))::xs) =
   getRecon xs) ∧
(∀xs.
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL plIncomplete)))::xs) =
   getRecon xs) ∧
```

```
(∀ xs .
   getRecon
     (Name PlatoonLeader says
      prop (SOME (SLc (PL invalidPlCommand)))::xs) =
   getRecon xs) ∧
(∀ xs v151 .
   getRecon
     (Name PlatoonLeader says prop (SOME (SLc (PSG v151)))::
          xs) =
   getRecon xs) ∧
(∀ xs v144 .
   getRecon
     (Name PlatoonSergeant says prop (SOME v144)::xs) =
   getRecon xs) ∧
(∀ xs v68 v136 v135 .
   getRecon (v135 meet v136 says prop v68::xs) =
   getRecon xs) ∧
(∀ xs v68 v138 v137 .
   getRecon (v137 quoting v138 says prop v68::xs) =
   getRecon xs) ∧
(∀ xs v69 v12 .
   getRecon (v12 says notf v69::xs) = getRecon xs) ∧
(∀ xs v71 v70 v12 .
   getRecon (v12 says (v70 andf v71)::xs) = getRecon xs) ∧
(∀ xs v73 v72 v12 .
   getRecon (v12 says (v72 orf v73)::xs) = getRecon xs) ∧
(∀ xs v75 v74 v12 .
   getRecon (v12 says (v74 impf v75)::xs) = getRecon xs) ∧
(∀ xs v77 v76 v12 .
   getRecon (v12 says (v76 eqf v77)::xs) = getRecon xs) ∧
(∀ xs v79 v78 v12 .
   getRecon (v12 says v78 says v79::xs) = getRecon xs) ∧
(∀ xs v81 v80 v12 .
   getRecon (v12 says v80 speaks_for v81::xs) =
   getRecon xs) ∧
(∀ xs v83 v82 v12 .
   getRecon (v12 says v82 controls v83::xs) = getRecon xs) ∧
(∀ xs v86 v85 v84 v12 .
   getRecon (v12 says reps v84 v85 v86::xs) = getRecon xs) ∧
(∀ xs v88 v87 v12 .
   getRecon (v12 says v87 domi v88::xs) = getRecon xs) ∧
(∀ xs v90 v89 v12 .
   getRecon (v12 says v89 eqi v90::xs) = getRecon xs) ∧
(∀ xs v92 v91 v12 .
   getRecon (v12 says v91 doms v92::xs) = getRecon xs) ∧
(∀ xs v94 v93 v12 .
   getRecon (v12 says v93 eqs v94::xs) = getRecon xs) ∧
(∀ xs v96 v95 v12 .
   getRecon (v12 says v95 eqn v96::xs) = getRecon xs) ∧
```

```
(∀ xs v98 v97 v12.
   getRecon (v12 says v97 lte v98::xs) = getRecon xs) ∧
(∀ xs v99 v12 v100.
   getRecon (v12 says v99 lt v100::xs) = getRecon xs) ∧
(∀ xs v15 v14.
   getRecon (v14 speaks_for v15::xs) = getRecon xs) ∧
(∀ xs v17 v16.
   getRecon (v16 controls v17::xs) = getRecon xs) ∧
(∀ xs v20 v19 v18.
   getRecon (reps v18 v19 v20::xs) = getRecon xs) ∧
(∀ xs v22 v21. getRecon (v21 domi v22::xs) = getRecon xs) ∧
(∀ xs v24 v23. getRecon (v23 eqi v24::xs) = getRecon xs) ∧
(∀ xs v26 v25. getRecon (v25 doms v26::xs) = getRecon xs) ∧
(∀ xs v28 v27. getRecon (v27 eqs v28::xs) = getRecon xs) ∧
(∀ xs v30 v29. getRecon (v29 eqn v30::xs) = getRecon xs) ∧
(∀ xs v32 v31. getRecon (v31 lte v32::xs) = getRecon xs) ∧
∀ xs v34 v33. getRecon (v33 lt v34::xs) = getRecon xs
```

[getRecon_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ xs.
       P
         (Name PlatoonLeader says
          prop (SOME (SLc (PL recon)))::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 xs. P xs ⇒ P (v12 says TT::xs)) ∧
    (∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
    (∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
    (∀ v146 xs.
       P xs ⇒
       P
         (Name PlatoonLeader says prop (SOME (ESCc v146))::
              xs)) ∧
    (∀ xs.
       P xs ⇒
       P
         (Name PlatoonLeader says
          prop (SOME (SLc (PL receiveMission)))::xs)) ∧
    (∀ xs.
       P xs ⇒
       P
         (Name PlatoonLeader says
```

```
      prop (SOME (SLc (PL warno)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL tentativePlan)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report1)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL opoid)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL complete)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL plIncomplete)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL invalidPlCommand)))::xs)) ∧
(∀ v151 xs.
   P xs ⇒
   P
```

```
      (Name PlatoonLeader says
       prop (SOME (SLc (PSG v151)))::xs)) ∧
   (∀v144 xs.
      P xs ⇒
      P (Name PlatoonSergeant says prop (SOME v144)::xs)) ∧
   (∀v135 v136 v68 xs.
      P xs ⇒ P (v135 meet v136 says prop v68::xs)) ∧
   (∀v137 v138 v68 xs.
      P xs ⇒ P (v137 quoting v138 says prop v68::xs)) ∧
   (∀v12 v69 xs. P xs ⇒ P (v12 says notf v69::xs)) ∧
   (∀v12 v70 v71 xs. P xs ⇒ P (v12 says (v70 andf v71)::xs)) ∧
   (∀v12 v72 v73 xs. P xs ⇒ P (v12 says (v72 orf v73)::xs)) ∧
   (∀v12 v74 v75 xs. P xs ⇒ P (v12 says (v74 impf v75)::xs)) ∧
   (∀v12 v76 v77 xs. P xs ⇒ P (v12 says (v76 eqf v77)::xs)) ∧
   (∀v12 v78 v79 xs. P xs ⇒ P (v12 says v78 says v79::xs)) ∧
   (∀v12 v80 v81 xs.
      P xs ⇒ P (v12 says v80 speaks_for v81::xs)) ∧
   (∀v12 v82 v83 xs.
      P xs ⇒ P (v12 says v82 controls v83::xs)) ∧
   (∀v12 v84 v85 v86 xs.
      P xs ⇒ P (v12 says reps v84 v85 v86::xs)) ∧
   (∀v12 v87 v88 xs. P xs ⇒ P (v12 says v87 domi v88::xs)) ∧
   (∀v12 v89 v90 xs. P xs ⇒ P (v12 says v89 eqi v90::xs)) ∧
   (∀v12 v91 v92 xs. P xs ⇒ P (v12 says v91 doms v92::xs)) ∧
   (∀v12 v93 v94 xs. P xs ⇒ P (v12 says v93 eqs v94::xs)) ∧
   (∀v12 v95 v96 xs. P xs ⇒ P (v12 says v95 eqn v96::xs)) ∧
   (∀v12 v97 v98 xs. P xs ⇒ P (v12 says v97 lte v98::xs)) ∧
   (∀v12 v99 v100 xs. P xs ⇒ P (v12 says v99 lt v100::xs)) ∧
   (∀v14 v15 xs. P xs ⇒ P (v14 speaks_for v15::xs)) ∧
   (∀v16 v17 xs. P xs ⇒ P (v16 controls v17::xs)) ∧
   (∀v18 v19 v20 xs. P xs ⇒ P (reps v18 v19 v20::xs)) ∧
   (∀v21 v22 xs. P xs ⇒ P (v21 domi v22::xs)) ∧
   (∀v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
   (∀v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
   (∀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
```

[getReport_def]

```
⊢ (getReport [] = [NONE]) ∧
  (∀xs.
     getReport
       (Name PlatoonLeader says
        prop (SOME (SLc (PL report1)))::xs) =
     [SOME (SLc (PL report1))]) ∧
  (∀xs. getReport (TT::xs) = getReport xs) ∧
  (∀xs. getReport (FF::xs) = getReport xs) ∧
```

```
(∀ xs v2. getReport (prop v2::xs) = getReport xs) ∧
(∀ xs v3. getReport (notf v3::xs) = getReport xs) ∧
(∀ xs v5 v4. getReport (v4 andf v5::xs) = getReport xs) ∧
(∀ xs v7 v6. getReport (v6 orf v7::xs) = getReport xs) ∧
(∀ xs v9 v8. getReport (v8 impf v9::xs) = getReport xs) ∧
(∀ xs v11 v10. getReport (v10 eqf v11::xs) = getReport xs) ∧
(∀ xs v12. getReport (v12 says TT::xs) = getReport xs) ∧
(∀ xs v12. getReport (v12 says FF::xs) = getReport xs) ∧
(∀ xs v134 .
   getReport (Name v134 says prop NONE::xs) = getReport xs) ∧
(∀ xs v146 .
   getReport
     (Name PlatoonLeader says prop (SOME (ESCc v146))::xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL receiveMission)))::xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says prop (SOME (SLc (PL warno)))::
          xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL tentativePlan)))::xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says prop (SOME (SLc (PL recon)))::
          xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says prop (SOME (SLc (PL opoid)))::
          xs) =
   getReport xs) ∧
(∀ xs .
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs) =
   getReport xs) ∧
```

```
(∀ xs.
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs) =
   getReport xs) ∧
(∀ xs.
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL complete)))::xs) =
   getReport xs) ∧
(∀ xs.
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL plIncomplete)))::xs) =
   getReport xs) ∧
(∀ xs.
   getReport
     (Name PlatoonLeader says
      prop (SOME (SLc (PL invalidPlCommand)))::xs) =
   getReport xs) ∧
(∀ xs v151.
   getReport
     (Name PlatoonLeader says prop (SOME (SLc (PSG v151)))::
          xs) =
   getReport xs) ∧
(∀ xs v144.
   getReport
     (Name PlatoonSergeant says prop (SOME v144)::xs) =
   getReport xs) ∧
(∀ xs v68 v136 v135.
   getReport (v135 meet v136 says prop v68::xs) =
   getReport xs) ∧
(∀ xs v68 v138 v137.
   getReport (v137 quoting v138 says prop v68::xs) =
   getReport xs) ∧
(∀ xs v69 v12.
   getReport (v12 says notf v69::xs) = getReport xs) ∧
(∀ xs v71 v70 v12.
   getReport (v12 says (v70 andf v71)::xs) = getReport xs) ∧
(∀ xs v73 v72 v12.
   getReport (v12 says (v72 orf v73)::xs) = getReport xs) ∧
(∀ xs v75 v74 v12.
   getReport (v12 says (v74 impf v75)::xs) = getReport xs) ∧
(∀ xs v77 v76 v12.
   getReport (v12 says (v76 eqf v77)::xs) = getReport xs) ∧
(∀ xs v79 v78 v12.
   getReport (v12 says v78 says v79::xs) = getReport xs) ∧
(∀ xs v81 v80 v12.
   getReport (v12 says v80 speaks_for v81::xs) =
```

```
      getReport xs) ∧
   (∀ xs v83 v82 v12.
      getReport (v12 says v82 controls v83::xs) =
      getReport xs) ∧
   (∀ xs v86 v85 v84 v12.
      getReport (v12 says reps v84 v85 v86::xs) =
      getReport xs) ∧
   (∀ xs v88 v87 v12.
      getReport (v12 says v87 domi v88::xs) = getReport xs) ∧
   (∀ xs v90 v89 v12.
      getReport (v12 says v89 eqi v90::xs) = getReport xs) ∧
   (∀ xs v92 v91 v12.
      getReport (v12 says v91 doms v92::xs) = getReport xs) ∧
   (∀ xs v94 v93 v12.
      getReport (v12 says v93 eqs v94::xs) = getReport xs) ∧
   (∀ xs v96 v95 v12.
      getReport (v12 says v95 eqn v96::xs) = getReport xs) ∧
   (∀ xs v98 v97 v12.
      getReport (v12 says v97 lte v98::xs) = getReport xs) ∧
   (∀ xs v99 v12 v100.
      getReport (v12 says v99 lt v100::xs) = getReport xs) ∧
   (∀ xs v15 v14.
      getReport (v14 speaks_for v15::xs) = getReport xs) ∧
   (∀ xs v17 v16.
      getReport (v16 controls v17::xs) = getReport xs) ∧
   (∀ xs v20 v19 v18.
      getReport (reps v18 v19 v20::xs) = getReport xs) ∧
   (∀ xs v22 v21. getReport (v21 domi v22::xs) = getReport xs) ∧
   (∀ xs v24 v23. getReport (v23 eqi v24::xs) = getReport xs) ∧
   (∀ xs v26 v25. getReport (v25 doms v26::xs) = getReport xs) ∧
   (∀ xs v28 v27. getReport (v27 eqs v28::xs) = getReport xs) ∧
   (∀ xs v30 v29. getReport (v29 eqn v30::xs) = getReport xs) ∧
   (∀ xs v32 v31. getReport (v31 lte v32::xs) = getReport xs) ∧
  ∀ xs v34 v33. getReport (v33 lt v34::xs) = getReport xs
```

[getReport_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ xs.
       P
         (Name PlatoonLeader says
          prop (SOME (SLc (PL report1)))::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 xs. P xs ⇒ P (v12 says TT::xs)) ∧
(∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
(∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
(∀ v146 xs.
   P xs ⇒
   P
     (Name PlatoonLeader says prop (SOME (ESCc v146))::
          xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL receiveMission)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL warno)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL tentativePlan)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL recon)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL opoid)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs)) ∧
(∀ xs.
```

$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL complete)))::xs)) $\wedge$
$(\forall xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL plIncomplete)))::xs)) $\wedge$
$(\forall xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL invalidPlCommand)))::xs)) $\wedge$
$(\forall v151\ xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PSG $v151$)))::xs)) $\wedge$
$(\forall v144\ xs.$
$P\ xs \Rightarrow$
P (Name PlatoonSergeant says prop (SOME $v144$)::xs)) $\wedge$
$(\forall v135\ v136\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v135$ meet $v136$ says prop v_{68}::xs)) $\wedge$
$(\forall v137\ v138\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v137$ quoting $v138$ says prop v_{68}::xs)) $\wedge$
$(\forall v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says notf v_{69}::xs)) $\wedge$
$(\forall v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{70} andf v_{71})::xs)) $\wedge$
$(\forall v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{72} orf v_{73})::xs)) $\wedge$
$(\forall v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{74} impf v_{75})::xs)) $\wedge$
$(\forall v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{76} eqf v_{77})::xs)) $\wedge$
$(\forall v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{78} says v_{79}::xs)) $\wedge$
$(\forall v_{12}\ v_{80}\ v_{81}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says v_{80} speaks_for v_{81}::xs)) $\wedge$
$(\forall v_{12}\ v_{82}\ v_{83}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says v_{82} controls v_{83}::xs)) $\wedge$
$(\forall v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says reps $v_{84}\ v_{85}\ v_{86}$::xs)) $\wedge$
$(\forall v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{87} domi v_{88}::xs)) $\wedge$
$(\forall v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{89} eqi v_{90}::xs)) $\wedge$
$(\forall v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{91} doms v_{92}::xs)) $\wedge$
$(\forall v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{93} eqs v_{94}::xs)) $\wedge$
$(\forall v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{95} eqn v_{96}::xs)) $\wedge$
$(\forall v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{97} lte v_{98}::xs)) $\wedge$
$(\forall v_{12}\ v_{99}\ v100\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{99} lt $v100$::xs)) $\wedge$
$(\forall v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P\ (v_{14}$ speaks_for v_{15}::xs)) $\wedge$
$(\forall v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P\ (v_{16}$ controls v_{17}::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)) $\wedge$

```
   (∀ v23 v24 xs. P xs ⇒ P (v23 eqi v24::xs)) ∧
   (∀ v25 v26 xs. P xs ⇒ P (v25 doms v26::xs)) ∧
   (∀ 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
```

[getTenativePlan_def]

```
⊢ (getTenativePlan [] = [NONE]) ∧
  (∀ xs.
     getTenativePlan
       (Name PlatoonLeader says
        prop (SOME (SLc (PL tentativePlan)))::xs) =
     [SOME (SLc (PL tentativePlan))]) ∧
  (∀ xs. getTenativePlan (TT::xs) = getTenativePlan xs) ∧
  (∀ xs. getTenativePlan (FF::xs) = getTenativePlan xs) ∧
  (∀ xs v2.
     getTenativePlan (prop v2::xs) = getTenativePlan xs) ∧
  (∀ xs v3.
     getTenativePlan (notf v3::xs) = getTenativePlan xs) ∧
  (∀ xs v5 v4.
     getTenativePlan (v4 andf v5::xs) = getTenativePlan xs) ∧
  (∀ xs v7 v6.
     getTenativePlan (v6 orf v7::xs) = getTenativePlan xs) ∧
  (∀ xs v9 v8.
     getTenativePlan (v8 impf v9::xs) = getTenativePlan xs) ∧
  (∀ xs v11 v10.
     getTenativePlan (v10 eqf v11::xs) = getTenativePlan xs) ∧
  (∀ xs v12.
     getTenativePlan (v12 says TT::xs) = getTenativePlan xs) ∧
  (∀ xs v12.
     getTenativePlan (v12 says FF::xs) = getTenativePlan xs) ∧
  (∀ xs v134.
     getTenativePlan (Name v134 says prop NONE::xs) =
     getTenativePlan xs) ∧
  (∀ xs v146.
     getTenativePlan
       (Name PlatoonLeader says prop (SOME (ESCc v146))::xs) =
     getTenativePlan xs) ∧
  (∀ xs.
     getTenativePlan
       (Name PlatoonLeader says
        prop (SOME (SLc (PL receiveMission)))::xs) =
     getTenativePlan xs) ∧
  (∀ xs.
     getTenativePlan
       (Name PlatoonLeader says prop (SOME (SLc (PL warno)))::
            xs) =
```

```
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says prop (SOME (SLc (PL recon)))::
          xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report1)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says prop (SOME (SLc (PL opoid)))::
          xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL complete)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL plIncomplete)))::xs) =
   getTenativePlan xs) ∧
(∀ xs.
   getTenativePlan
     (Name PlatoonLeader says
      prop (SOME (SLc (PL invalidPlCommand)))::xs) =
   getTenativePlan xs) ∧
(∀ xs v151.
   getTenativePlan
     (Name PlatoonLeader says prop (SOME (SLc (PSG v151)))::
```

```
          xs) =
   getTenativePlan xs) ∧
(∀ xs v144 .
   getTenativePlan
     (Name PlatoonSergeant says prop (SOME v144)::xs) =
   getTenativePlan xs) ∧
(∀ xs v68 v136 v135 .
   getTenativePlan (v135 meet v136 says prop v68::xs) =
   getTenativePlan xs) ∧
(∀ xs v68 v138 v137 .
   getTenativePlan (v137 quoting v138 says prop v68::xs) =
   getTenativePlan xs) ∧
(∀ xs v69 v12 .
   getTenativePlan (v12 says notf v69::xs) =
   getTenativePlan xs) ∧
(∀ xs v71 v70 v12 .
   getTenativePlan (v12 says (v70 andf v71)::xs) =
   getTenativePlan xs) ∧
(∀ xs v73 v72 v12 .
   getTenativePlan (v12 says (v72 orf v73)::xs) =
   getTenativePlan xs) ∧
(∀ xs v75 v74 v12 .
   getTenativePlan (v12 says (v74 impf v75)::xs) =
   getTenativePlan xs) ∧
(∀ xs v77 v76 v12 .
   getTenativePlan (v12 says (v76 eqf v77)::xs) =
   getTenativePlan xs) ∧
(∀ xs v79 v78 v12 .
   getTenativePlan (v12 says v78 says v79::xs) =
   getTenativePlan xs) ∧
(∀ xs v81 v80 v12 .
   getTenativePlan (v12 says v80 speaks_for v81::xs) =
   getTenativePlan xs) ∧
(∀ xs v83 v82 v12 .
   getTenativePlan (v12 says v82 controls v83::xs) =
   getTenativePlan xs) ∧
(∀ xs v86 v85 v84 v12 .
   getTenativePlan (v12 says reps v84 v85 v86::xs) =
   getTenativePlan xs) ∧
(∀ xs v88 v87 v12 .
   getTenativePlan (v12 says v87 domi v88::xs) =
   getTenativePlan xs) ∧
(∀ xs v90 v89 v12 .
   getTenativePlan (v12 says v89 eqi v90::xs) =
   getTenativePlan xs) ∧
(∀ xs v92 v91 v12 .
   getTenativePlan (v12 says v91 doms v92::xs) =
   getTenativePlan xs) ∧
(∀ xs v94 v93 v12 .
```

```
      getTenativePlan (v12 says v93 eqs v94::xs) =
      getTenativePlan xs) ∧
   (∀ xs v96 v95 v12.
      getTenativePlan (v12 says v95 eqn v96::xs) =
      getTenativePlan xs) ∧
   (∀ xs v98 v97 v12.
      getTenativePlan (v12 says v97 lte v98::xs) =
      getTenativePlan xs) ∧
   (∀ xs v99 v12 v100.
      getTenativePlan (v12 says v99 lt v100::xs) =
      getTenativePlan xs) ∧
   (∀ xs v15 v14.
      getTenativePlan (v14 speaks_for v15::xs) =
      getTenativePlan xs) ∧
   (∀ xs v17 v16.
      getTenativePlan (v16 controls v17::xs) =
      getTenativePlan xs) ∧
   (∀ xs v20 v19 v18.
      getTenativePlan (reps v18 v19 v20::xs) =
      getTenativePlan xs) ∧
   (∀ xs v22 v21.
      getTenativePlan (v21 domi v22::xs) = getTenativePlan xs) ∧
   (∀ xs v24 v23.
      getTenativePlan (v23 eqi v24::xs) = getTenativePlan xs) ∧
   (∀ xs v26 v25.
      getTenativePlan (v25 doms v26::xs) = getTenativePlan xs) ∧
   (∀ xs v28 v27.
      getTenativePlan (v27 eqs v28::xs) = getTenativePlan xs) ∧
   (∀ xs v30 v29.
      getTenativePlan (v29 eqn v30::xs) = getTenativePlan xs) ∧
   (∀ xs v32 v31.
      getTenativePlan (v31 lte v32::xs) = getTenativePlan xs) ∧
   ∀ xs v34 v33.
     getTenativePlan (v33 lt v34::xs) = getTenativePlan xs
```

[getTenativePlan_ind]

```
⊢ ∀ P.
    P [] ∧
    (∀ xs.
       P
         (Name PlatoonLeader says
          prop (SOME (SLc (PL tentativePlan)))::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 xs. P xs ⇒ P (v12 says TT::xs)) ∧
(∀ v12 xs. P xs ⇒ P (v12 says FF::xs)) ∧
(∀ v134 xs. P xs ⇒ P (Name v134 says prop NONE::xs)) ∧
(∀ v146 xs.
   P xs ⇒
   P
     (Name PlatoonLeader says prop (SOME (ESCc v146))::
          xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL receiveMission)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL warno)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL recon)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report1)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL completePlan)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL opoid)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL supervise)))::xs)) ∧
(∀ xs.
   P xs ⇒
   P
     (Name PlatoonLeader says
      prop (SOME (SLc (PL report2)))::xs)) ∧
(∀ xs.
```

$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL complete)))::xs)) $\wedge$
$(\forall xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL plIncomplete)))::xs)) $\wedge$
$(\forall xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PL invalidPlCommand)))::xs)) $\wedge$
$(\forall v151\ xs.$
$P\ xs \Rightarrow$
P
(Name PlatoonLeader says
prop (SOME (SLc (PSG $v151$)))::xs)) $\wedge$
$(\forall v144\ xs.$
$P\ xs \Rightarrow$
P (Name PlatoonSergeant says prop (SOME $v144$)::xs)) $\wedge$
$(\forall v135\ v136\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v135$ meet $v136$ says prop v_{68}::xs)) $\wedge$
$(\forall v137\ v138\ v_{68}\ xs.$
$P\ xs \Rightarrow P$ ($v137$ quoting $v138$ says prop v_{68}::xs)) $\wedge$
$(\forall v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says notf v_{69}::xs)) $\wedge$
$(\forall v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{70} andf v_{71})::xs)) $\wedge$
$(\forall v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{72} orf v_{73})::xs)) $\wedge$
$(\forall v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{74} impf v_{75})::xs)) $\wedge$
$(\forall v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says (v_{76} eqf v_{77})::xs)) $\wedge$
$(\forall v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{78} says v_{79}::xs)) $\wedge$
$(\forall v_{12}\ v_{80}\ v_{81}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says v_{80} speaks_for v_{81}::xs)) $\wedge$
$(\forall v_{12}\ v_{82}\ v_{83}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says v_{82} controls v_{83}::xs)) $\wedge$
$(\forall v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$P\ xs \Rightarrow P\ (v_{12}$ says reps $v_{84}\ v_{85}\ v_{86}$::xs)) $\wedge$
$(\forall v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{87} domi v_{88}::xs)) $\wedge$
$(\forall v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{89} eqi v_{90}::xs)) $\wedge$
$(\forall v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{91} doms v_{92}::xs)) $\wedge$
$(\forall v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{93} eqs v_{94}::xs)) $\wedge$
$(\forall v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{95} eqn v_{96}::xs)) $\wedge$
$(\forall v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{97} lte v_{98}::xs)) $\wedge$
$(\forall v_{12}\ v_{99}\ v100\ xs.\ P\ xs \Rightarrow P\ (v_{12}$ says v_{99} lt $v100$::xs)) $\wedge$
$(\forall v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P\ (v_{14}$ speaks_for v_{15}::xs)) $\wedge$
$(\forall v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P\ (v_{16}$ controls v_{17}::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)) $\wedge$

$$
\begin{array}{l}
(\forall\, v_{23}\ \ v_{24}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{23}\ \texttt{eqi}\ \ v_{24}\texttt{::}xs))\ \wedge \\
(\forall\, v_{25}\ \ v_{26}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{25}\ \texttt{doms}\ \ v_{26}\texttt{::}xs))\ \wedge \\
(\forall\, v_{27}\ \ v_{28}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{27}\ \texttt{eqs}\ \ v_{28}\texttt{::}xs))\ \wedge \\
(\forall\, v_{29}\ \ v_{30}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{29}\ \texttt{eqn}\ \ v_{30}\texttt{::}xs))\ \wedge \\
(\forall\, v_{31}\ \ v_{32}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{31}\ \texttt{lte}\ \ v_{32}\texttt{::}xs))\ \wedge \\
(\forall\, v_{33}\ \ v_{34}\ \ xs.\ \ P\ \ xs\ \Rightarrow\ P\ \ (v_{33}\ \texttt{lt}\ \ v_{34}\texttt{::}xs))\ \Rightarrow \\
\forall\, v.\ \ P\ \ v
\end{array}
$$

Index