Contents

1 ConductPBType Theory **3**
1.1 Datatypes 3
1.2 Theorems 3

2 ssmConductPB Theory **4**
2.1 Definitions 4
2.2 Theorems 4

1 ConductPBType Theory

Built: 16 May 2018
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

plCommandPB = securePB | withdrawPB | completePB
| plIncompletePB

psgCommandPB = actionsInPB | psgIncompletePB

slCommand = PL plCommandPB | PSG psgCommandPB

slOutput = ConductPB | SecurePB | ActionsInPB | WithdrawPB
| CompletePB | unAuthenticated | unAuthorized

slState = CONDUCT_PB | SECURE_PB | ACTIONS_IN_PB | WITHDRAW_PB
| COMPLETE_PB

stateRole = PlatoonLeader | PlatoonSergeant

1.2 Theorems

[plCommandPB_distinct_clauses]
$\vdash$ securePB $\neq$ withdrawPB $\wedge$ securePB $\neq$ completePB $\wedge$
securePB $\neq$ plIncompletePB $\wedge$ withdrawPB $\neq$ completePB $\wedge$
withdrawPB $\neq$ plIncompletePB $\wedge$ completePB $\neq$ plIncompletePB

[psgCommandPB_distinct_clauses]
$\vdash$ actionsInPB $\neq$ psgIncompletePB

[slCommand_distinct_clauses]
$\vdash \forall a'\ a.$ PL $a \neq$ PSG a'

[slCommand_one_one]
$\vdash$ $(\forall a\ a'.$ (PL a = PL a') $\iff$ $(a = a')) \wedge$
$\forall a\ a'.$ (PSG a = PSG a') $\iff$ $(a = a')$

[slOutput_distinct_clauses]
$\vdash$ ConductPB $\neq$ SecurePB $\wedge$ ConductPB $\neq$ ActionsInPB $\wedge$
ConductPB $\neq$ WithdrawPB $\wedge$ ConductPB $\neq$ CompletePB $\wedge$
ConductPB $\neq$ unAuthenticated $\wedge$ ConductPB $\neq$ unAuthorized $\wedge$
SecurePB $\neq$ ActionsInPB $\wedge$ SecurePB $\neq$ WithdrawPB $\wedge$
SecurePB $\neq$ CompletePB $\wedge$ SecurePB $\neq$ unAuthenticated $\wedge$
SecurePB $\neq$ unAuthorized $\wedge$ ActionsInPB $\neq$ WithdrawPB $\wedge$
ActionsInPB $\neq$ CompletePB $\wedge$ ActionsInPB $\neq$ unAuthenticated $\wedge$
ActionsInPB $\neq$ unAuthorized $\wedge$ WithdrawPB $\neq$ CompletePB $\wedge$
WithdrawPB $\neq$ unAuthenticated $\wedge$ WithdrawPB $\neq$ unAuthorized $\wedge$
CompletePB $\neq$ unAuthenticated $\wedge$ CompletePB $\neq$ unAuthorized $\wedge$
unAuthenticated $\neq$ unAuthorized

[slRole_distinct_clauses]

⊢ PlatoonLeader ≠ PlatoonSergeant

[slState_distinct_clauses]

⊢ CONDUCT_PB ≠ SECURE_PB ∧ CONDUCT_PB ≠ ACTIONS_IN_PB ∧
CONDUCT_PB ≠ WITHDRAW_PB ∧ CONDUCT_PB ≠ COMPLETE_PB ∧
SECURE_PB ≠ ACTIONS_IN_PB ∧ SECURE_PB ≠ WITHDRAW_PB ∧
SECURE_PB ≠ COMPLETE_PB ∧ ACTIONS_IN_PB ≠ WITHDRAW_PB ∧
ACTIONS_IN_PB ≠ COMPLETE_PB ∧ WITHDRAW_PB ≠ COMPLETE_PB

2 ssmConductPB Theory

Built: 16 May 2018
Parent Theories: ConductPBType, ssm11, OMNIType

2.1 Definitions

[secContextConductPB_def]

⊢ ∀ *plcmd psgcmd incomplete*.
secContextConductPB *plcmd psgcmd incomplete* =
[Name PlatoonLeader controls prop (SOME (SLc (PL *plcmd*)));
Name PlatoonSergeant controls
prop (SOME (SLc (PSG *psgcmd*)));
Name PlatoonLeader says
prop (SOME (SLc (PSG *psgcmd*))) impf prop NONE;
Name PlatoonSergeant says
prop (SOME (SLc (PL *plcmd*))) impf prop NONE]

[ssmConductPBStateInterp_def]

⊢ ∀ *slState*. ssmConductPBStateInterp *slState* = TT

2.2 Theorems

[authTestConductPB_cmd_reject_lemma]

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

[authTestConductPB_def]

⊢ (authTestConductPB (Name PlatoonLeader says prop *cmd*) ⟺ T) ∧
(authTestConductPB (Name PlatoonSergeant says prop *cmd*) ⟺
T) ∧ (authTestConductPB TT ⟺ F) ∧
(authTestConductPB FF ⟺ F) ∧
(authTestConductPB (prop *v*) ⟺ F) ∧
(authTestConductPB (notf v_1) ⟺ F) ∧
(authTestConductPB (v_2 andf v_3) ⟺ F) ∧
(authTestConductPB (v_4 orf v_5) ⟺ F) ∧
(authTestConductPB (v_6 impf v_7) ⟺ F) ∧

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

[authTestConductPB_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})) ∧

$(\forall\, v_{10}\ v_{80}\ v_{81}.\ P\ (v_{10}\ \texttt{says}\ v_{80}\ \texttt{controls}\ v_{81})) \wedge$
$(\forall\, v_{10}\ v_{82}\ v_{83}\ v_{84}.\ P\ (v_{10}\ \texttt{says}\ \texttt{reps}\ v_{82}\ v_{83}\ v_{84})) \wedge$
$(\forall\, v_{10}\ v_{85}\ v_{86}.\ P\ (v_{10}\ \texttt{says}\ v_{85}\ \texttt{domi}\ v_{86})) \wedge$
$(\forall\, v_{10}\ v_{87}\ v_{88}.\ P\ (v_{10}\ \texttt{says}\ v_{87}\ \texttt{eqi}\ v_{88})) \wedge$
$(\forall\, v_{10}\ v_{89}\ v_{90}.\ P\ (v_{10}\ \texttt{says}\ v_{89}\ \texttt{doms}\ v_{90})) \wedge$
$(\forall\, v_{10}\ v_{91}\ v_{92}.\ P\ (v_{10}\ \texttt{says}\ v_{91}\ \texttt{eqs}\ v_{92})) \wedge$
$(\forall\, v_{10}\ v_{93}\ v_{94}.\ P\ (v_{10}\ \texttt{says}\ v_{93}\ \texttt{eqn}\ v_{94})) \wedge$
$(\forall\, v_{10}\ v_{95}\ v_{96}.\ P\ (v_{10}\ \texttt{says}\ v_{95}\ \texttt{lte}\ v_{96})) \wedge$
$(\forall\, v_{10}\ v_{97}\ v_{98}.\ P\ (v_{10}\ \texttt{says}\ v_{97}\ \texttt{lt}\ v_{98})) \wedge$
$(\forall\, v_{12}\ v_{13}.\ P\ (v_{12}\ \texttt{speaks\_for}\ v_{13})) \wedge$
$(\forall\, v_{14}\ v_{15}.\ P\ (v_{14}\ \texttt{controls}\ v_{15})) \wedge$
$(\forall\, v_{16}\ v_{17}\ v_{18}.\ P\ (\texttt{reps}\ v_{16}\ v_{17}\ v_{18})) \wedge$
$(\forall\, v_{19}\ v_{20}.\ P\ (v_{19}\ \texttt{domi}\ v_{20})) \wedge$
$(\forall\, v_{21}\ v_{22}.\ P\ (v_{21}\ \texttt{eqi}\ v_{22})) \wedge$
$(\forall\, v_{23}\ v_{24}.\ P\ (v_{23}\ \texttt{doms}\ v_{24})) \wedge$
$(\forall\, v_{25}\ v_{26}.\ P\ (v_{25}\ \texttt{eqs}\ v_{26})) \wedge (\forall\, v_{27}\ v_{28}.\ P\ (v_{27}\ \texttt{eqn}\ v_{28})) \wedge$
$(\forall\, v_{29}\ v_{30}.\ P\ (v_{29}\ \texttt{lte}\ v_{30})) \wedge (\forall\, v_{31}\ v_{32}.\ P\ (v_{31}\ \texttt{lt}\ v_{32})) \Rightarrow$
$\forall\, v.\ P\ v$

[conductPBNS_def]

$\vdash$ (conductPBNS CONDUCT_PB (exec (PL securePB)) = SECURE_PB) $\wedge$
(conductPBNS CONDUCT_PB (exec (PL plIncompletePB)) =
CONDUCT_PB) $\wedge$
(conductPBNS SECURE_PB (exec (PSG actionsInPB)) =
ACTIONS_IN_PB) $\wedge$
(conductPBNS SECURE_PB (exec (PSG psgIncompletePB)) =
SECURE_PB) $\wedge$
(conductPBNS ACTIONS_IN_PB (exec (PL withdrawPB)) =
WITHDRAW_PB) $\wedge$
(conductPBNS ACTIONS_IN_PB (exec (PL plIncompletePB)) =
ACTIONS_IN_PB) $\wedge$
(conductPBNS WITHDRAW_PB (exec (PL completePB)) =
COMPLETE_PB) $\wedge$
(conductPBNS WITHDRAW_PB (exec (PL plIncompletePB)) =
WITHDRAW_PB) $\wedge$ (conductPBNS s (trap (PL cmd')) = s) $\wedge$
(conductPBNS s (trap (PSG cmd)) = s) $\wedge$
(conductPBNS s (discard (PL cmd')) = s) $\wedge$
(conductPBNS s (discard (PSG cmd)) = s)

[conductPBNS_ind]

$\vdash \forall\, P.$
P CONDUCT_PB (exec (PL securePB)) $\wedge$
P CONDUCT_PB (exec (PL plIncompletePB)) $\wedge$
P SECURE_PB (exec (PSG actionsInPB)) $\wedge$
P SECURE_PB (exec (PSG psgIncompletePB)) $\wedge$
P ACTIONS_IN_PB (exec (PL withdrawPB)) $\wedge$
P ACTIONS_IN_PB (exec (PL plIncompletePB)) $\wedge$
P WITHDRAW_PB (exec (PL completePB)) $\wedge$
P WITHDRAW_PB (exec (PL plIncompletePB)) $\wedge$

```
     (∀ s cmd. P s (trap (PL cmd))) ∧
     (∀ s cmd. P s (trap (PSG cmd))) ∧
     (∀ s cmd. P s (discard (PL cmd))) ∧
     (∀ s cmd. P s (discard (PSG cmd))) ∧
     P CONDUCT_PB (exec (PL withdrawPB)) ∧
     P CONDUCT_PB (exec (PL completePB)) ∧
     (∀ v11. P CONDUCT_PB (exec (PSG v11))) ∧
     (∀ v13. P SECURE_PB (exec (PL v13))) ∧
     P ACTIONS_IN_PB (exec (PL securePB)) ∧
     P ACTIONS_IN_PB (exec (PL completePB)) ∧
     (∀ v17. P ACTIONS_IN_PB (exec (PSG v17))) ∧
     P WITHDRAW_PB (exec (PL securePB)) ∧
     P WITHDRAW_PB (exec (PL withdrawPB)) ∧
     (∀ v20. P WITHDRAW_PB (exec (PSG v20))) ∧
     (∀ v21. P COMPLETE_PB (exec v21)) ⇒
     ∀ v v1. P v v1
```

[conductPBOut_def]

```
⊢ (conductPBOut CONDUCT_PB (exec (PL securePB)) = ConductPB) ∧
  (conductPBOut CONDUCT_PB (exec (PL plIncompletePB)) =
   ConductPB) ∧
  (conductPBOut SECURE_PB (exec (PSG actionsInPB)) =
   SecurePB) ∧
  (conductPBOut SECURE_PB (exec (PSG psgIncompletePB)) =
   SecurePB) ∧
  (conductPBOut ACTIONS_IN_PB (exec (PL withdrawPB)) =
   ActionsInPB) ∧
  (conductPBOut ACTIONS_IN_PB (exec (PL plIncompletePB)) =
   ActionsInPB) ∧
  (conductPBOut WITHDRAW_PB (exec (PL completePB)) =
   WithdrawPB) ∧
  (conductPBOut WITHDRAW_PB (exec (PL plIncompletePB)) =
   WithdrawPB) ∧
  (conductPBOut s (trap (PL cmd')) = unAuthorized) ∧
  (conductPBOut s (trap (PSG cmd)) = unAuthorized) ∧
  (conductPBOut s (discard (PL cmd')) = unAuthenticated) ∧
  (conductPBOut s (discard (PSG cmd)) = unAuthenticated)
```

[conductPBOut_ind]

```
⊢ ∀ P.
     P CONDUCT_PB (exec (PL securePB)) ∧
     P CONDUCT_PB (exec (PL plIncompletePB)) ∧
     P SECURE_PB (exec (PSG actionsInPB)) ∧
     P SECURE_PB (exec (PSG psgIncompletePB)) ∧
     P ACTIONS_IN_PB (exec (PL withdrawPB)) ∧
     P ACTIONS_IN_PB (exec (PL plIncompletePB)) ∧
     P WITHDRAW_PB (exec (PL completePB)) ∧
     P WITHDRAW_PB (exec (PL plIncompletePB)) ∧
     (∀ s cmd. P s (trap (PL cmd))) ∧
```

$(\forall s\ cmd.\ P\ s\ (\texttt{trap (PSG}\ cmd\texttt{)})) \wedge$
$(\forall s\ cmd.\ P\ s\ (\texttt{discard (PL}\ cmd\texttt{)})) \wedge$
$(\forall s\ cmd.\ P\ s\ (\texttt{discard (PSG}\ cmd\texttt{)})) \wedge$
$P\ \texttt{CONDUCT\_PB (exec (PL withdrawPB))} \wedge$
$P\ \texttt{CONDUCT\_PB (exec (PL completePB))} \wedge$
$(\forall v_{11}.\ P\ \texttt{CONDUCT\_PB (exec (PSG}\ v_{11}\texttt{))}) \wedge$
$(\forall v_{13}.\ P\ \texttt{SECURE\_PB (exec (PL}\ v_{13}\texttt{))}) \wedge$
$P\ \texttt{ACTIONS\_IN\_PB (exec (PL securePB))} \wedge$
$P\ \texttt{ACTIONS\_IN\_PB (exec (PL completePB))} \wedge$
$(\forall v_{17}.\ P\ \texttt{ACTIONS\_IN\_PB (exec (PSG}\ v_{17}\texttt{))}) \wedge$
$P\ \texttt{WITHDRAW\_PB (exec (PL securePB))} \wedge$
$P\ \texttt{WITHDRAW\_PB (exec (PL withdrawPB))} \wedge$
$(\forall v_{20}.\ P\ \texttt{WITHDRAW\_PB (exec (PSG}\ v_{20}\texttt{))}) \wedge$
$(\forall v_{21}.\ P\ \texttt{COMPLETE\_PB (exec}\ v_{21}\texttt{)}) \Rightarrow$
$\forall v\ v_1.\ P\ v\ v_1$

[PlatoonLeader_exec_plCommandPB_justified_thm]

$\vdash \forall NS\ Out\ M\ Oi\ Os.$
$\texttt{TR}\ (M, Oi, Os)\ \texttt{(exec (SLc (PL}\ plCommand\texttt{)))}$
$\texttt{(CFG authTestConductPB ssmConductPBStateInterp}$
$\texttt{(secContextConductPB}\ plCommand\ psgCommand\ incomplete\texttt{)}$
$\texttt{(Name PlatoonLeader says}$
$\texttt{prop (SOME (SLc (PL}\ plCommand\texttt{)))::}ins\texttt{)}\ s\ outs\texttt{)}$
$\texttt{(CFG authTestConductPB ssmConductPBStateInterp}$
$\texttt{(secContextConductPB}\ plCommand\ psgCommand\ incomplete\texttt{)}$
$ins\ (NS\ s\ \texttt{(exec (SLc (PL}\ plCommand\texttt{))))}$
$(Out\ s\ \texttt{(exec (SLc (PL}\ plCommand\texttt{)))::}outs\texttt{))} \iff$
$\texttt{authTestConductPB}$
$\texttt{(Name PlatoonLeader says}$
$\texttt{prop (SOME (SLc (PL}\ plCommand\texttt{))))} \wedge$
$\texttt{CFGInterpret}\ (M, Oi, Os)$
$\texttt{(CFG authTestConductPB ssmConductPBStateInterp}$
$\texttt{(secContextConductPB}\ plCommand\ psgCommand\ incomplete\texttt{)}$
$\texttt{(Name PlatoonLeader says}$
$\texttt{prop (SOME (SLc (PL}\ plCommand\texttt{)))::}ins\texttt{)}\ s\ outs\texttt{)} \wedge$
$(M, Oi, Os)\ \texttt{sat prop (SOME (SLc (PL}\ plCommand\texttt{)))}$

[PlatoonLeader_plCommandPB_lemma]

$\vdash \texttt{CFGInterpret}\ (M, Oi, Os)$
$\texttt{(CFG authTestConductPB ssmConductPBStateInterp}$
$\texttt{(secContextConductPB}\ plCommand\ psgCommand\ incomplete\texttt{)}$
$\texttt{(Name PlatoonLeader says}$
$\texttt{prop (SOME (SLc (PL}\ plCommand\texttt{)))::}ins\texttt{)}\ s\ outs\texttt{)} \Rightarrow$
$(M, Oi, Os)\ \texttt{sat prop (SOME (SLc (PL}\ plCommand\texttt{)))}$

[PlatoonSergeant_exec_psgCommandPB_justified_thm]

$\vdash \forall NS\ Out\ M\ Oi\ Os.$
$\texttt{TR}\ (M, Oi, Os)\ \texttt{(exec (SLc (PSG}\ psgCommand\texttt{)))}$

```
      (CFG authTestConductPB ssmConductPBStateInterp
         (secContextConductPB plCommand psgCommand incomplete)
         (Name PlatoonSergeant says
          prop (SOME (SLc (PSG psgCommand)))::ins) s outs)
      (CFG authTestConductPB ssmConductPBStateInterp
         (secContextConductPB plCommand psgCommand incomplete)
         ins (NS s (exec (SLc (PSG psgCommand))))
         (Out s (exec (SLc (PSG psgCommand)))::outs)) ⟺
    authTestConductPB
      (Name PlatoonSergeant says
       prop (SOME (SLc (PSG psgCommand)))) ∧
    CFGInterpret (M,Oi,Os)
      (CFG authTestConductPB ssmConductPBStateInterp
         (secContextConductPB plCommand psgCommand incomplete)
         (Name PlatoonSergeant says
          prop (SOME (SLc (PSG psgCommand)))::ins) s outs) ∧
    (M,Oi,Os) sat prop (SOME (SLc (PSG psgCommand)))
```

[PlatoonSergeant_psgCommandPB_lemma]

```
⊢ CFGInterpret (M,Oi,Os)
    (CFG authTestConductPB ssmConductPBStateInterp
       (secContextConductPB plCommand psgCommand incomplete)
       (Name PlatoonSergeant says
        prop (SOME (SLc (PSG psgCommand)))::ins) s outs) ⇒
  (M,Oi,Os) sat prop (SOME (SLc (PSG psgCommand)))
```

Index

ConductPBType Theory, 3
Datatypes, 3
Theorems, 3
plCommandPB_distinct_clauses, 3
psgCommandPB_distinct_clauses, 3
slCommand_distinct_clauses, 3
slCommand_one_one, 3
slOutput_distinct_clauses, 3
slRole_distinct_clauses, 4
slState_distinct_clauses, 4

ssmConductPB Theory, 4
Definitions, 4
secContextConductPB_def, 4
ssmConductPBStateInterp_def, 4
Theorems, 4
authTestConductPB_cmd_reject_lemma, 4
authTestConductPB_def, 4
authTestConductPB_ind, 5
conductPBNS_def, 6
conductPBNS_ind, 6
conductPBOut_def, 7
conductPBOut_ind, 7
PlatoonLeader_exec_plCommandPB_-justified_thm, 8
PlatoonLeader_plCommandPB_lemma, 8
PlatoonSergeant_exec_psgCommandPB_-justified_thm, 8
PlatoonSergeant_psgCommandPB_lemma, 9