

Contents

1	PBTypeIntegrated Theory	3
1.1	Datatypes	3
1.2	Theorems	3
2	ssmPBIntegrated Theory	4
2.1	Theorems	5
3	PBIntegratedDef Theory	11
3.1	Definitions	11
3.2	Theorems	12

1 PBTYPESINTEGRATED Theory

Built: 11 June 2018

Parent Theories: OMNITYPE

1.1 Datatypes

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

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

```
slCommand = PL plCommand | OMNI omniCommand
```

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

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

```
stateRole = PlatoonLeader | Omni
```

1.2 Theorems

[omniCommand_distinct_clauses]

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

[plCommand_distinct_clauses]

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

$$\begin{aligned} & \text{conductORP} \neq \text{conductPB} \wedge \text{conductORP} \neq \text{completePB} \wedge \\ & \text{conductORP} \neq \text{incomplete} \wedge \text{moveToPB} \neq \text{conductPB} \wedge \\ & \text{moveToPB} \neq \text{completePB} \wedge \text{moveToPB} \neq \text{incomplete} \wedge \\ & \text{conductPB} \neq \text{completePB} \wedge \text{conductPB} \neq \text{incomplete} \wedge \\ & \text{completePB} \neq \text{incomplete} \end{aligned}$$

[slCommand_distinct_clauses]

$$\vdash \forall a' a. \text{PL } a \neq \text{OMNI } a'$$

[slCommand_one_one]

$$\begin{aligned} & \vdash (\forall a a'. (\text{PL } a = \text{PL } a') \iff (a = a')) \wedge \\ & \quad \forall a a'. (\text{OMNI } a = \text{OMNI } a') \iff (a = a') \end{aligned}$$

[slOutput_distinct_clauses]

$$\begin{aligned} & \vdash \text{PlanPB} \neq \text{MoveToORP} \wedge \text{PlanPB} \neq \text{ConductORP} \wedge \\ & \quad \text{PlanPB} \neq \text{MoveToPB} \wedge \text{PlanPB} \neq \text{ConductPB} \wedge \\ & \quad \text{PlanPB} \neq \text{CompletePB} \wedge \text{PlanPB} \neq \text{unAuthenticated} \wedge \\ & \quad \text{PlanPB} \neq \text{unAuthorized} \wedge \text{MoveToORP} \neq \text{ConductORP} \wedge \\ & \quad \text{MoveToORP} \neq \text{MoveToPB} \wedge \text{MoveToORP} \neq \text{ConductPB} \wedge \\ & \quad \text{MoveToORP} \neq \text{CompletePB} \wedge \text{MoveToORP} \neq \text{unAuthenticated} \wedge \\ & \quad \text{MoveToORP} \neq \text{unAuthorized} \wedge \text{ConductORP} \neq \text{MoveToPB} \wedge \\ & \quad \text{ConductORP} \neq \text{ConductPB} \wedge \text{ConductORP} \neq \text{CompletePB} \wedge \\ & \quad \text{ConductORP} \neq \text{unAuthenticated} \wedge \text{ConductORP} \neq \text{unAuthorized} \wedge \\ & \quad \text{MoveToPB} \neq \text{ConductPB} \wedge \text{MoveToPB} \neq \text{CompletePB} \wedge \\ & \quad \text{MoveToPB} \neq \text{unAuthenticated} \wedge \text{MoveToPB} \neq \text{unAuthorized} \wedge \\ & \quad \text{ConductPB} \neq \text{CompletePB} \wedge \text{ConductPB} \neq \text{unAuthenticated} \wedge \\ & \quad \text{ConductPB} \neq \text{unAuthorized} \wedge \text{CompletePB} \neq \text{unAuthenticated} \wedge \\ & \quad \text{CompletePB} \neq \text{unAuthorized} \wedge \text{unAuthenticated} \neq \text{unAuthorized} \end{aligned}$$

[slState_distinct_clauses]

$$\begin{aligned} & \vdash \text{PLAN_PB} \neq \text{MOVE_TO_ORP} \wedge \text{PLAN_PB} \neq \text{CONDUCT_ORP} \wedge \\ & \quad \text{PLAN_PB} \neq \text{MOVE_TO_PB} \wedge \text{PLAN_PB} \neq \text{CONDUCT_PB} \wedge \\ & \quad \text{PLAN_PB} \neq \text{COMPLETE_PB} \wedge \text{MOVE_TO_ORP} \neq \text{CONDUCT_ORP} \wedge \\ & \quad \text{MOVE_TO_ORP} \neq \text{MOVE_TO_PB} \wedge \text{MOVE_TO_ORP} \neq \text{CONDUCT_PB} \wedge \\ & \quad \text{MOVE_TO_ORP} \neq \text{COMPLETE_PB} \wedge \text{CONDUCT_ORP} \neq \text{MOVE_TO_PB} \wedge \\ & \quad \text{CONDUCT_ORP} \neq \text{CONDUCT_PB} \wedge \text{CONDUCT_ORP} \neq \text{COMPLETE_PB} \wedge \\ & \quad \text{MOVE_TO_PB} \neq \text{CONDUCT_PB} \wedge \text{MOVE_TO_PB} \neq \text{COMPLETE_PB} \wedge \\ & \quad \text{CONDUCT_PB} \neq \text{COMPLETE_PB} \end{aligned}$$

[stateRole_distinct_clauses]

$$\vdash \text{PlatoonLeader} \neq \text{Omni}$$

2 ssmPBIntegrated Theory

Built: 11 June 2018

Parent Theories: PBIntegratedDef, ssm

2.1 Theorems

[inputOK_cmd_reject_lemma]

$\vdash \forall cmd. \neg \text{inputOK} (\text{prop} (\text{SOME } cmd))$

[inputOK_def]

\vdash (inputOK (Name PlatoonLeader says prop cmd) \iff T) \wedge
 (inputOK (Name Omni says prop cmd) \iff T) \wedge
 (inputOK TT \iff F) \wedge (inputOK FF \iff F) \wedge
 (inputOK (prop v) \iff F) \wedge (inputOK (notf v₁) \iff F) \wedge
 (inputOK (v₂ andf v₃) \iff F) \wedge (inputOK (v₄ orf v₅) \iff F) \wedge
 (inputOK (v₆ impf v₇) \iff F) \wedge (inputOK (v₈ eqf v₉) \iff F) \wedge
 (inputOK (v₁₀ says TT) \iff F) \wedge (inputOK (v₁₀ says FF) \iff F) \wedge
 (inputOK (v₁₃₃ meet v₁₃₄ says prop v₆₆) \iff F) \wedge
 (inputOK (v₁₃₅ quoting v₁₃₆ says prop v₆₆) \iff F) \wedge
 (inputOK (v₁₀ says notf v₆₇) \iff F) \wedge
 (inputOK (v₁₀ says (v₆₈ andf v₆₉)) \iff F) \wedge
 (inputOK (v₁₀ says (v₇₀ orf v₇₁)) \iff F) \wedge
 (inputOK (v₁₀ says (v₇₂ impf v₇₃)) \iff F) \wedge
 (inputOK (v₁₀ says (v₇₄ eqf v₇₅)) \iff F) \wedge
 (inputOK (v₁₀ says v₇₆ says v₇₇) \iff F) \wedge
 (inputOK (v₁₀ says v₇₈ speaks_for v₇₉) \iff F) \wedge
 (inputOK (v₁₀ says v₈₀ controls v₈₁) \iff F) \wedge
 (inputOK (v₁₀ says reps v₈₂ v₈₃ v₈₄) \iff F) \wedge
 (inputOK (v₁₀ says v₈₅ domi v₈₆) \iff F) \wedge
 (inputOK (v₁₀ says v₈₇ eqi v₈₈) \iff F) \wedge
 (inputOK (v₁₀ says v₈₉ doms v₉₀) \iff F) \wedge
 (inputOK (v₁₀ says v₉₁ eqs v₉₂) \iff F) \wedge
 (inputOK (v₁₀ says v₉₃ eqn v₉₄) \iff F) \wedge
 (inputOK (v₁₀ says v₉₅ lte v₉₆) \iff F) \wedge
 (inputOK (v₁₀ says v₉₇ lt v₉₈) \iff F) \wedge
 (inputOK (v₁₂ speaks_for v₁₃) \iff F) \wedge
 (inputOK (v₁₄ controls v₁₅) \iff F) \wedge
 (inputOK (reps v₁₆ v₁₇ v₁₈) \iff F) \wedge
 (inputOK (v₁₉ domi v₂₀) \iff F) \wedge
 (inputOK (v₂₁ eqi v₂₂) \iff F) \wedge
 (inputOK (v₂₃ doms v₂₄) \iff F) \wedge
 (inputOK (v₂₅ eqs v₂₆) \iff F) \wedge (inputOK (v₂₇ eqn v₂₈) \iff F) \wedge
 (inputOK (v₂₉ lte v₃₀) \iff F) \wedge (inputOK (v₃₁ lt v₃₂) \iff F)

[inputOK_ind]

$\vdash \forall P.$
 ($\forall cmd. P$ (Name PlatoonLeader says prop cmd)) \wedge
 ($\forall cmd. P$ (Name Omni says prop cmd)) $\wedge P$ TT $\wedge P$ FF \wedge
 ($\forall v. P$ (prop v)) \wedge ($\forall v_1. P$ (notf v₁)) \wedge
 ($\forall v_2 v_3. P$ (v₂ andf v₃)) \wedge ($\forall v_4 v_5. P$ (v₄ orf v₅)) \wedge
 ($\forall v_6 v_7. P$ (v₆ impf v₇)) \wedge ($\forall v_8 v_9. P$ (v₈ eqf v₉)) \wedge
 ($\forall v_{10}. P$ (v₁₀ says TT)) \wedge ($\forall v_{10}. P$ (v₁₀ says FF)) \wedge
 ($\forall v_{133} v_{134} v_{66}. P$ (v₁₃₃ meet v₁₃₄ says prop v₆₆)) \wedge

$$\begin{aligned}
& (\forall v_{135} v_{136} v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge \\
& (\forall v_{10} v_{67}. P (v_{10} \text{ says notf } v_{67})) \wedge \\
& (\forall v_{10} v_{68} v_{69}. P (v_{10} \text{ says } (v_{68} \text{ andf } v_{69}))) \wedge \\
& (\forall v_{10} v_{70} v_{71}. P (v_{10} \text{ says } (v_{70} \text{ orf } v_{71}))) \wedge \\
& (\forall v_{10} v_{72} v_{73}. P (v_{10} \text{ says } (v_{72} \text{ impf } v_{73}))) \wedge \\
& (\forall v_{10} v_{74} v_{75}. P (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75}))) \wedge \\
& (\forall v_{10} v_{76} v_{77}. P (v_{10} \text{ says } v_{76} \text{ says } v_{77})) \wedge \\
& (\forall v_{10} v_{78} v_{79}. P (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79})) \wedge \\
& (\forall v_{10} v_{80} v_{81}. P (v_{10} \text{ says } v_{80} \text{ controls } v_{81})) \wedge \\
& (\forall v_{10} v_{82} v_{83} v_{84}. P (v_{10} \text{ says reps } v_{82} v_{83} v_{84})) \wedge \\
& (\forall v_{10} v_{85} v_{86}. P (v_{10} \text{ says } v_{85} \text{ domi } v_{86})) \wedge \\
& (\forall v_{10} v_{87} v_{88}. P (v_{10} \text{ says } v_{87} \text{ eqi } v_{88})) \wedge \\
& (\forall v_{10} v_{89} v_{90}. P (v_{10} \text{ says } v_{89} \text{ doms } v_{90})) \wedge \\
& (\forall v_{10} v_{91} v_{92}. P (v_{10} \text{ says } v_{91} \text{ eqs } v_{92})) \wedge \\
& (\forall v_{10} v_{93} v_{94}. P (v_{10} \text{ says } v_{93} \text{ eqn } v_{94})) \wedge \\
& (\forall v_{10} v_{95} v_{96}. P (v_{10} \text{ says } v_{95} \text{ lte } v_{96})) \wedge \\
& (\forall v_{10} v_{97} v_{98}. P (v_{10} \text{ says } v_{97} \text{ lt } v_{98})) \wedge \\
& (\forall v_{12} v_{13}. P (v_{12} \text{ speaks_for } v_{13})) \wedge \\
& (\forall v_{14} v_{15}. P (v_{14} \text{ controls } v_{15})) \wedge \\
& (\forall v_{16} v_{17} v_{18}. P (\text{reps } v_{16} v_{17} v_{18})) \wedge \\
& (\forall v_{19} v_{20}. P (v_{19} \text{ domi } v_{20})) \wedge \\
& (\forall v_{21} v_{22}. P (v_{21} \text{ eqi } v_{22})) \wedge \\
& (\forall v_{23} v_{24}. P (v_{23} \text{ doms } v_{24})) \wedge \\
& (\forall v_{25} v_{26}. P (v_{25} \text{ eqs } v_{26})) \wedge (\forall v_{27} v_{28}. P (v_{27} \text{ eqn } v_{28})) \wedge \\
& (\forall v_{29} v_{30}. P (v_{29} \text{ lte } v_{30})) \wedge (\forall v_{31} v_{32}. P (v_{31} \text{ lt } v_{32})) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

[PBNS_def]

$$\begin{aligned}
& \vdash (\text{PBNS PLAN_PB (exec } x) = \\
& \quad \text{if getPlCom } x = \text{crossLD then MOVE_TO_ORP else PLAN_PB}) \wedge \\
& (\text{PBNS MOVE_TO_ORP (exec } x) = \\
& \quad \text{if getPlCom } x = \text{conductORP then CONDUCT_ORP} \\
& \quad \text{else MOVE_TO_ORP}) \wedge \\
& (\text{PBNS CONDUCT_ORP (exec } x) = \\
& \quad \text{if getPlCom } x = \text{moveToPB then MOVE_TO_PB else CONDUCT_ORP}) \wedge \\
& (\text{PBNS MOVE_TO_PB (exec } x) = \\
& \quad \text{if getPlCom } x = \text{conductPB then CONDUCT_PB else MOVE_TO_PB}) \wedge \\
& (\text{PBNS CONDUCT_PB (exec } x) = \\
& \quad \text{if getPlCom } x = \text{completePB then COMPLETE_PB} \\
& \quad \text{else CONDUCT_PB}) \wedge (\text{PBNS } s (\text{trap } v_0) = s) \wedge \\
& (\text{PBNS } s (\text{discard } v_1) = s)
\end{aligned}$$

[PBNS_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& (\forall x. P \text{ PLAN_PB (exec } x)) \wedge (\forall x. P \text{ MOVE_TO_ORP (exec } x)) \wedge \\
& (\forall x. P \text{ CONDUCT_ORP (exec } x)) \wedge \\
& (\forall x. P \text{ MOVE_TO_PB (exec } x)) \wedge (\forall x. P \text{ CONDUCT_PB (exec } x)) \wedge \\
& (\forall s v_0. P s (\text{trap } v_0)) \wedge (\forall s v_1. P s (\text{discard } v_1)) \wedge \\
& (\forall v_6. P \text{ COMPLETE_PB (exec } v_6)) \Rightarrow \\
& \forall v v_1. P v v_1
\end{aligned}$$

[PBOut_def]

$$\begin{aligned}
&\vdash (\text{PBOut PLAN_PB (exec } x) = \\
&\quad \text{if getPlCom } x = \text{crossLD then MoveToORP else PlanPB}) \wedge \\
&(\text{PBOut MOVE_TO_ORP (exec } x) = \\
&\quad \text{if getPlCom } x = \text{conductORP then ConductORP else MoveToORP}) \wedge \\
&(\text{PBOut CONDUCT_ORP (exec } x) = \\
&\quad \text{if getPlCom } x = \text{moveToPB then MoveToORP else ConductORP}) \wedge \\
&(\text{PBOut MOVE_TO_PB (exec } x) = \\
&\quad \text{if getPlCom } x = \text{conductPB then ConductPB else MoveToPB}) \wedge \\
&(\text{PBOut CONDUCT_PB (exec } x) = \\
&\quad \text{if getPlCom } x = \text{completePB then CompletePB else ConductPB}) \wedge \\
&(\text{PBOut } s \text{ (trap } v_0) = \text{unAuthorized}) \wedge \\
&(\text{PBOut } s \text{ (discard } v_1) = \text{unAuthenticated})
\end{aligned}$$
[PBOut_ind]

$$\begin{aligned}
&\vdash \forall P. \\
&\quad (\forall x. P \text{ PLAN_PB (exec } x)) \wedge (\forall x. P \text{ MOVE_TO_ORP (exec } x)) \wedge \\
&\quad (\forall x. P \text{ CONDUCT_ORP (exec } x)) \wedge \\
&\quad (\forall x. P \text{ MOVE_TO_PB (exec } x)) \wedge (\forall x. P \text{ CONDUCT_PB (exec } x)) \wedge \\
&\quad (\forall s \ v_0. P \ s \text{ (trap } v_0)) \wedge (\forall s \ v_1. P \ s \text{ (discard } v_1)) \wedge \\
&\quad (\forall v_6. P \text{ COMPLETE_PB (exec } v_6)) \Rightarrow \\
&\quad \forall v \ v_1. P \ v \ v_1
\end{aligned}$$
[PlatoonLeader_Omni_notDiscard_slCommand_thm]

$$\begin{aligned}
&\vdash \forall NS \ Out \ M \ Oi \ Os. \\
&\quad \neg \text{TR } (M, Oi, Os) \\
&\quad (\text{discard} \\
&\quad \quad [\text{SOME (SLc (PL } plCommand)); \\
&\quad \quad \quad \text{SOME (SLc (OMNI } omniCommand))]) \\
&\quad (\text{CFG inputOK secContext secAuthorization} \\
&\quad \quad ([\text{Name Omni says prop (SOME (SLc (PL } plCommand))}); \\
&\quad \quad \quad \text{Name PlatoonLeader says} \\
&\quad \quad \quad \text{prop (SOME (SLc (OMNI } omniCommand)))]::ins) \text{ PLAN_PB} \\
&\quad \quad outs) \\
&\quad (\text{CFG inputOK secContext secAuthorization ins} \\
&\quad \quad (NS \text{ PLAN_PB} \\
&\quad \quad \quad (\text{discard} \\
&\quad \quad \quad \quad [\text{SOME (SLc (PL } plCommand)); \\
&\quad \quad \quad \quad \quad \text{SOME (SLc (OMNI } omniCommand))]) \\
&\quad \quad \quad (\text{Out PLAN_PB} \\
&\quad \quad \quad \quad (\text{discard} \\
&\quad \quad \quad \quad \quad [\text{SOME (SLc (PL } plCommand)); \\
&\quad \quad \quad \quad \quad \quad \text{SOME (SLc (OMNI } omniCommand)))]::outs))
\end{aligned}$$
[PlatoonLeader_PLAN_PB_exec_justified_lemma]

$$\begin{aligned}
&\vdash \forall NS \ Out \ M \ Oi \ Os. \\
&\quad \text{TR } (M, Oi, Os) \\
&\quad (\text{exec}
\end{aligned}$$

```

      (inputList
        [Name Omni says
          prop (SOME (SLc (OMNI ssmPlanPBComplete)));
          Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD))))])
    (CFG inputOK secContext secAuthorization
      ([Name Omni says
        prop (SOME (SLc (OMNI ssmPlanPBComplete)));
        Name PlatoonLeader says
          prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB outs)
    (CFG inputOK secContext secAuthorization ins
      (NS PLAN_PB
        (exec
          (inputList
            [Name Omni says
              prop (SOME (SLc (OMNI ssmPlanPBComplete)));
              Name PlatoonLeader says
                prop (SOME (SLc (PL crossLD))))]))
        (Out PLAN_PB
          (exec
            (inputList
              [Name Omni says
                prop (SOME (SLc (OMNI ssmPlanPBComplete)));
                Name PlatoonLeader says
                  prop (SOME (SLc (PL crossLD)))]::outs))  $\iff$ 
authenticationTest inputOK
      [Name Omni says
        prop (SOME (SLc (OMNI ssmPlanPBComplete)));
        Name PlatoonLeader says
          prop (SOME (SLc (PL crossLD)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
      (CFG inputOK secContext secAuthorization
        ([Name Omni says
          prop (SOME (SLc (OMNI ssmPlanPBComplete)));
          Name PlatoonLeader says
            prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
          outs)  $\wedge$ 
      (M, Oi, Os) satList
      propCommandList
        [Name Omni says
          prop (SOME (SLc (OMNI ssmPlanPBComplete)));
          Name PlatoonLeader says prop (SOME (SLc (PL crossLD)))]

```

[PlatoonLeader_PLAN_PB_exec_justified_thm]

```

 $\vdash \forall NS \text{ Out } M \text{ Oi } Os.$ 
  TR (M, Oi, Os)
    (exec
      [SOME (SLc (OMNI ssmPlanPBComplete));
       SOME (SLc (PL crossLD))])

```



```

(CFG inputOK secContext secAuthorization
  ([Name Omni says
    prop (SOME (SLc (OMNI ssmPlanPBComplete)));
    Name PlatoonLeader says
    prop (SOME (SLc (PL crossLD)))])::ins) PLAN_PB outs)
(CFG inputOK secContext secAuthorization ins
  (NS PLAN_PB
    (exec
      [SOME (SLc (OMNI ssmPlanPBComplete));
        SOME (SLc (PL crossLD))]))
  (Out PLAN_PB
    (exec
      [SOME (SLc (OMNI ssmPlanPBComplete));
        SOME (SLc (PL crossLD))])::outs))  $\iff$ 
authenticationTest inputOK
  [Name Omni says
    prop (SOME (SLc (OMNI ssmPlanPBComplete)));
    Name PlatoonLeader says
    prop (SOME (SLc (PL crossLD)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG inputOK secContext secAuthorization
    ([Name Omni says
      prop (SOME (SLc (OMNI ssmPlanPBComplete)));
      Name PlatoonLeader says
      prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
    outs)  $\wedge$ 
  (M, Oi, Os) satList
  [prop (SOME (SLc (OMNI ssmPlanPBComplete)));
    prop (SOME (SLc (PL crossLD)))]

```

[PlatoonLeader_PLAN_PB_exec_lemma]

```

 $\vdash \forall M \ Oi \ Os.$ 
  CFGInterpret (M, Oi, Os)
    (CFG inputOK secContext secAuthorization
      ([Name Omni says
        prop (SOME (SLc (OMNI ssmPlanPBComplete)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
      outs)  $\Rightarrow$ 
    (M, Oi, Os) satList
  propCommandList
    [Name Omni says
      prop (SOME (SLc (OMNI ssmPlanPBComplete)));
      Name PlatoonLeader says prop (SOME (SLc (PL crossLD)))]

```

[PlatoonLeader_PLAN_PB_trap_justified_lemma]

```

 $\vdash omniCommand \neq ssmPlanPBComplete \Rightarrow$ 
  (s = PLAN_PB)  $\Rightarrow$ 
   $\forall NS \ Out \ M \ Oi \ Os.$ 

```

```

TR (M, Oi, Os)
  (trap
    (inputList
      [Name Omni says
        prop (SOME (SLc (OMNI omniCommand)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD))))])
    (CFG inputOK secContext secAuthorization
      ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB outs)
    (CFG inputOK secContext secAuthorization ins
      (NS PLAN_PB
        (trap
          (inputList
            [Name Omni says
              prop (SOME (SLc (OMNI omniCommand)));
              Name PlatoonLeader says
              prop (SOME (SLc (PL crossLD)))])))
        (Out PLAN_PB
          (trap
            (inputList
              [Name Omni says
                prop (SOME (SLc (OMNI omniCommand)));
                Name PlatoonLeader says
                prop (SOME (SLc (PL crossLD)))]))::outs))  $\iff$ 
authenticationTest inputOK
  [Name Omni says prop (SOME (SLc (OMNI omniCommand)));
    Name PlatoonLeader says
    prop (SOME (SLc (PL crossLD)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG inputOK secContext secAuthorization
    ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
      Name PlatoonLeader says
      prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
    outs)  $\wedge$  (M, Oi, Os) sat prop NONE

```

[PlatoonLeader_PLAN_PB_trap_justified_thm]

```

 $\vdash \text{omniCommand} \neq \text{ssmPlanPBComplete} \Rightarrow$ 
 $(s = \text{PLAN\_PB}) \Rightarrow$ 
 $\forall NS \text{ Out } M \text{ Oi } Os.$ 
  TR (M, Oi, Os)
    (trap
      [SOME (SLc (OMNI omniCommand));
        SOME (SLc (PL crossLD))])
    (CFG inputOK secContext secAuthorization
      ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
        Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB outs)

```

```

(CFG inputOK secContext secAuthorization ins
  (NS PLAN_PB
    (trap
      [SOME (SLc (OMNI omniCommand));
       SOME (SLc (PL crossLD))]))
  (Out PLAN_PB
    (trap
      [SOME (SLc (OMNI omniCommand));
       SOME (SLc (PL crossLD))]::outs))  $\iff$ 
authenticationTest inputOK
  [Name Omni says prop (SOME (SLc (OMNI omniCommand)));
   Name PlatoonLeader says
    prop (SOME (SLc (PL crossLD)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG inputOK secContext secAuthorization
    ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
     Name PlatoonLeader says
      prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
    outs)  $\wedge$  (M, Oi, Os) sat prop NONE

```

[PlatoonLeader_PLAN_PB_trap_lemma]

```

 $\vdash$  omniCommand  $\neq$  ssmPlanPBComplete  $\Rightarrow$ 
  (s = PLAN_PB)  $\Rightarrow$ 
 $\forall M \ Oi \ Os.$ 
  CFGInterpret (M, Oi, Os)
    (CFG inputOK secContext secAuthorization
      ([Name Omni says prop (SOME (SLc (OMNI omniCommand)));
       Name PlatoonLeader says
        prop (SOME (SLc (PL crossLD)))]::ins) PLAN_PB
      outs)  $\Rightarrow$ 
    (M, Oi, Os) sat prop NONE

```

3 PBIntegratedDef Theory

Built: 11 June 2018

Parent Theories: PBTypeIntegrated, aclfoundation

3.1 Definitions

[secAuthorization_def]

```

 $\vdash \forall xs. \text{secAuthorization } xs = \text{secHelper } (\text{getOmniCommand } xs)$ 

```

[secContext_def]

```

 $\vdash (\forall xs.$ 
  secContext PLAN_PB xs =
  if getOmniCommand xs = ssmPlanPBComplete then
    [prop (SOME (SLc (OMNI ssmPlanPBComplete)))] impf

```

```

    Name PlatoonLeader controls
    prop (SOME (SLc (PL crossLD))))]
  else [prop NONE]] ∧
(∀ xs.
  secContext MOVE_TO_ORP xs =
  if getOmniCommand xs = ssmMoveToORPComplete then
    [prop (SOME (SLc (OMNI ssmMoveToORPComplete))) impf
      Name PlatoonLeader controls
      prop (SOME (SLc (PL conductORP)))]
  else [prop NONE]] ∧
(∀ xs.
  secContext CONDUCT_ORP xs =
  if getOmniCommand xs = ssmConductORPComplete then
    [prop (SOME (SLc (OMNI ssmConductORPComplete))) impf
      Name PlatoonLeader controls
      prop (SOME (SLc (PL moveToPB)))]
  else [prop NONE]] ∧
(∀ xs.
  secContext MOVE_TO_PB xs =
  if getOmniCommand xs = ssmConductORPComplete then
    [prop (SOME (SLc (OMNI ssmMoveToPBComplete))) impf
      Name PlatoonLeader controls
      prop (SOME (SLc (PL conductPB)))]
  else [prop NONE]] ∧
∀ xs.
  secContext CONDUCT_PB xs =
  if getOmniCommand xs = ssmConductPBComplete then
    [prop (SOME (SLc (OMNI ssmConductPBComplete))) impf
      Name PlatoonLeader controls
      prop (SOME (SLc (PL completePB)))]
  else [prop NONE]

```

[secHelper_def]

```

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

```

3.2 Theorems

[getOmniCommand_def]

```

⊢ (getOmniCommand [] = invalidOmniCommand) ∧
(∀ xs cmd.
  getOmniCommand
    (Name Omni says prop (SOME (SLc (OMNI cmd))))::xs) =
  cmd) ∧
(∀ xs. getOmniCommand (TT::xs) = getOmniCommand xs) ∧
(∀ xs. getOmniCommand (FF::xs) = getOmniCommand xs) ∧
(∀ xs v2. getOmniCommand (prop v2::xs) = getOmniCommand xs) ∧

```

$$\begin{aligned}
& (\forall xs \ v_3. \text{getOmniCommand} (\text{notf } v_3::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_5 \ v_4. \\
& \quad \text{getOmniCommand} (v_4 \text{ andf } v_5::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_7 \ v_6. \\
& \quad \text{getOmniCommand} (v_6 \text{ orf } v_7::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_9 \ v_8. \\
& \quad \text{getOmniCommand} (v_8 \text{ impf } v_9::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{11} \ v_{10}. \\
& \quad \text{getOmniCommand} (v_{10} \text{ eqf } v_{11}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says TT}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says FF}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{134}. \\
& \quad \text{getOmniCommand} (\text{Name } v_{134} \text{ says prop NONE}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{144}. \\
& \quad \text{getOmniCommand} \\
& \quad \quad (\text{Name PlatoonLeader says prop (SOME } v_{144})::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{146}. \\
& \quad \text{getOmniCommand} \\
& \quad \quad (\text{Name Omni says prop (SOME (ESCC } v_{146}))::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{150}. \\
& \quad \text{getOmniCommand} \\
& \quad \quad (\text{Name Omni says prop (SOME (SLC (PL } v_{150}))::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{68} \ v_{136} \ v_{135}. \\
& \quad \text{getOmniCommand} (v_{135} \text{ meet } v_{136} \text{ says prop } v_{68}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{68} \ v_{138} \ v_{137}. \\
& \quad \text{getOmniCommand} (v_{137} \text{ quoting } v_{138} \text{ says prop } v_{68}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{69} \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says notf } v_{69}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{71} \ v_{70} \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says } (v_{70} \text{ andf } v_{71})::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{73} \ v_{72} \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says } (v_{72} \text{ orf } v_{73})::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{75} \ v_{74} \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says } (v_{74} \text{ impf } v_{75})::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{77} \ v_{76} \ v_{12}. \\
& \quad \text{getOmniCommand} (v_{12} \text{ says } (v_{76} \text{ eqf } v_{77})::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall xs \ v_{79} \ v_{78} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{78} \text{ says } v_{79}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{81} \ v_{80} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{80} \text{ speaks_for } v_{81}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{83} \ v_{82} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{82} \text{ controls } v_{83}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{86} \ v_{85} \ v_{84} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says reps } v_{84} \ v_{85} \ v_{86}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{88} \ v_{87} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{87} \text{ domi } v_{88}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{90} \ v_{89} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{89} \text{ eqi } v_{90}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{92} \ v_{91} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{91} \text{ doms } v_{92}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{94} \ v_{93} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{93} \text{ eqs } v_{94}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{96} \ v_{95} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{95} \text{ eqn } v_{96}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{98} \ v_{97} \ v_{12}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{97} \text{ lte } v_{98}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{99} \ v_{12} \ v_{100}. \\
& \quad \text{getOmniCommand } (v_{12} \text{ says } v_{99} \text{ lt } v_{100}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{15} \ v_{14}. \\
& \quad \text{getOmniCommand } (v_{14} \text{ speaks_for } v_{15}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{17} \ v_{16}. \\
& \quad \text{getOmniCommand } (v_{16} \text{ controls } v_{17}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{20} \ v_{19} \ v_{18}. \\
& \quad \text{getOmniCommand } (\text{reps } v_{18} \ v_{19} \ v_{20}::xs) = \\
& \quad \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{22} \ v_{21}. \\
& \quad \text{getOmniCommand } (v_{21} \text{ domi } v_{22}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{24} \ v_{23}. \\
& \quad \text{getOmniCommand } (v_{23} \text{ eqi } v_{24}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{26} \ v_{25}. \\
& \quad \text{getOmniCommand } (v_{25} \text{ doms } v_{26}::xs) = \text{getOmniCommand } xs) \wedge \\
& (\forall xs \ v_{28} \ v_{27}.
\end{aligned}$$

```

  getOmniCommand (v27 eqs v28::xs) = getOmniCommand xs) ∧
  (∀ xs v30 v29.
    getOmniCommand (v29 eqn v30::xs) = getOmniCommand xs) ∧
  (∀ xs v32 v31.
    getOmniCommand (v31 lte v32::xs) = getOmniCommand xs) ∧
  ∀ xs v34 v33.
    getOmniCommand (v33 lt v34::xs) = getOmniCommand xs

```

[getOmniCommand_ind]

```

⊢ ∀ P.
  P [] ∧
  (∀ cmd xs.
    P (Name Omni says prop (SOME (SLc (OMNI cmd)))::xs)) ∧
    (∀ xs. P xs ⇒ P (TT::xs)) ∧ (∀ xs. P xs ⇒ P (FF::xs)) ∧
    (∀ v2 xs. P xs ⇒ P (prop v2::xs)) ∧
    (∀ v3 xs. P xs ⇒ P (notf v3::xs)) ∧
    (∀ v4 v5 xs. P xs ⇒ P (v4 andf v5::xs)) ∧
    (∀ v6 v7 xs. P xs ⇒ P (v6 orf v7::xs)) ∧
    (∀ v8 v9 xs. P xs ⇒ P (v8 impf v9::xs)) ∧
    (∀ v10 v11 xs. P xs ⇒ P (v10 eqf v11::xs)) ∧
    (∀ v12 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 Omni says prop (SOME (ESCc v146))::xs)) ∧
    (∀ v150 xs.
      P xs ⇒
      P (Name Omni 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)) ∧

```

$$\begin{aligned}
& (\forall v_{12} v_{91} v_{92} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{91} \text{ doms } v_{92} :: xs)) \wedge \\
& (\forall v_{12} v_{93} v_{94} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{93} \text{ eqs } v_{94} :: xs)) \wedge \\
& (\forall v_{12} v_{95} v_{96} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{95} \text{ eqn } v_{96} :: xs)) \wedge \\
& (\forall v_{12} v_{97} v_{98} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{97} \text{ lte } v_{98} :: xs)) \wedge \\
& (\forall v_{12} v_{99} v_{100} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{99} \text{ lt } v_{100} :: xs)) \wedge \\
& (\forall v_{14} v_{15} xs. P xs \Rightarrow P (v_{14} \text{ speaks_for } v_{15} :: xs)) \wedge \\
& (\forall v_{16} v_{17} xs. P xs \Rightarrow P (v_{16} \text{ controls } v_{17} :: xs)) \wedge \\
& (\forall v_{18} v_{19} v_{20} xs. P xs \Rightarrow P (\text{reps } v_{18} v_{19} v_{20} :: xs)) \wedge \\
& (\forall v_{21} v_{22} xs. P xs \Rightarrow P (v_{21} \text{ domi } v_{22} :: xs)) \wedge \\
& (\forall v_{23} v_{24} xs. P xs \Rightarrow P (v_{23} \text{ eqi } v_{24} :: xs)) \wedge \\
& (\forall v_{25} v_{26} xs. P xs \Rightarrow P (v_{25} \text{ doms } v_{26} :: xs)) \wedge \\
& (\forall v_{27} v_{28} xs. P xs \Rightarrow P (v_{27} \text{ eqs } v_{28} :: xs)) \wedge \\
& (\forall v_{29} v_{30} xs. P xs \Rightarrow P (v_{29} \text{ eqn } v_{30} :: xs)) \wedge \\
& (\forall v_{31} v_{32} xs. P xs \Rightarrow P (v_{31} \text{ lte } v_{32} :: xs)) \wedge \\
& (\forall v_{33} v_{34} xs. P xs \Rightarrow P (v_{33} \text{ lt } v_{34} :: xs)) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

[getPlCom_def]

$$\begin{aligned}
& \vdash (\text{getPlCom } [] = \text{incomplete}) \wedge \\
& (\forall xs \text{ cmd}. \text{getPlCom } (\text{SOME } (\text{SLc } (\text{PL } \text{cmd}))) :: xs) = \text{cmd}) \wedge \\
& (\forall xs. \text{getPlCom } (\text{NONE} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs v_4. \text{getPlCom } (\text{SOME } (\text{ESCc } v_4)) :: xs) = \text{getPlCom } xs) \wedge \\
& \forall xs v_9. \text{getPlCom } (\text{SOME } (\text{SLc } (\text{OMNI } v_9))) :: xs) = \text{getPlCom } xs
\end{aligned}$$

[getPlCom_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& P [] \wedge (\forall \text{cmd } xs. P (\text{SOME } (\text{SLc } (\text{PL } \text{cmd}))) :: xs)) \wedge \\
& (\forall xs. P xs \Rightarrow P (\text{NONE} :: xs)) \wedge \\
& (\forall v_4 xs. P xs \Rightarrow P (\text{SOME } (\text{ESCc } v_4)) :: xs)) \wedge \\
& (\forall v_9 xs. P xs \Rightarrow P (\text{SOME } (\text{SLc } (\text{OMNI } v_9))) :: xs)) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

Index

PBIntegratedDef Theory, 11

Definitions, 11

secAuthorization_def, 11

secContext_def, 11

secHelper_def, 12

Theorems, 12

getOmniCommand_def, 12

getOmniCommand_ind, 15

getPlCom_def, 16

getPlCom_ind, 16

PBTypeIntegrated Theory, 3

Datatypes, 3

Theorems, 3

omniCommand_distinct_clauses, 3

plCommand_distinct_clauses, 3

slCommand_distinct_clauses, 4

slCommand_one_one, 4

slOutput_distinct_clauses, 4

slState_distinct_clauses, 4

stateRole_distinct_clauses, 4

ssmPBIntegrated Theory, 4

Theorems, 5

inputOK_cmd_reject_lemma, 5

inputOK_def, 5

inputOK_ind, 5

PBNS_def, 6

PBNS_ind, 6

PBOut_def, 7

PBOut_ind, 7

PlatoonLeader_Omni_notDiscard_slCom-
mand_thm, 7

PlatoonLeader_PLAN_PB_exec_justi-
fied_lemma, 7

PlatoonLeader_PLAN_PB_exec_justi-
fied_thm, 8

PlatoonLeader_PLAN_PB_exec_lemma,
9

PlatoonLeader_PLAN_PB_trap_justi-
fied_lemma, 9

PlatoonLeader_PLAN_PB_trap_justi-
fied_thm, 10

PlatoonLeader_PLAN_PB_trap_lemma,
11