

Contents

1	OMNIType Theory	3
1.1	Datatypes	3
1.2	Theorems	3
2	ssm11 Theory	4
2.1	Datatypes	4
2.2	Definitions	4
2.3	Theorems	5
3	ssm Theory	11
3.1	Datatypes	11
3.2	Definitions	12
3.3	Theorems	13
4	satList Theory	21
4.1	Definitions	21
4.2	Theorems	21
5	PBTypeIntegrated Theory	21
5.1	Datatypes	21
5.2	Theorems	22
6	PBIntegratedDef Theory	23
6.1	Definitions	23
6.2	Theorems	24
7	ssmPBIntegrated Theory	28
7.1	Theorems	28
8	ssmConductORP Theory	35
8.1	Definitions	35
8.2	Theorems	35
9	ConductORPType Theory	40
9.1	Datatypes	40
9.2	Theorems	41
10	ssmConductPB Theory	41
10.1	Definitions	42
10.2	Theorems	42

11 ConductPBType Theory	47
11.1 Datatypes	47
11.2 Theorems	47
12 ssmMoveToORP Theory	48
12.1 Definitions	48
12.2 Theorems	48
13 MoveToORPType Theory	53
13.1 Datatypes	53
13.2 Theorems	53
14 ssmMoveToPB Theory	54
14.1 Definitions	54
14.2 Theorems	54
15 MoveToPBType Theory	58
15.1 Datatypes	58
15.2 Theorems	58
16 ssmPlanPB Theory	59
16.1 Theorems	59
17 PlanPBType Theory	69
17.1 Datatypes	69
17.2 Theorems	70
18 PlanPBDef Theory	72
18.1 Definitions	72
18.2 Theorems	73

1 OMNIType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

1.1 Datatypes

```

command = ESCc escCommand | SLc 'slCommand

escCommand = returnToBase | changeMission | resupply
              | reactToContact

escOutput = ReturnToBase | ChangeMission | Resupply
            | ReactToContact

escState = RTB | CM | RESUPPLY | RTC

output = ESCo escOutput | SLo 'slOutput

principal = SR 'stateRole

state = ESCs escState | SLs 'slState

```

1.2 Theorems

[command_distinct_clauses]

$$\vdash \forall a' a. \text{ESCc } a \neq \text{SLc } a'$$

[command_one_one]

$$\vdash (\forall a a'. (\text{ESCc } a = \text{ESCc } a') \iff (a = a')) \wedge \\ \forall a a'. (\text{SLc } a = \text{SLc } a') \iff (a = a')$$

[escCommand_distinct_clauses]

$$\vdash \text{returnToBase} \neq \text{changeMission} \wedge \text{returnToBase} \neq \text{resupply} \wedge \\ \text{returnToBase} \neq \text{reactToContact} \wedge \text{changeMission} \neq \text{resupply} \wedge \\ \text{changeMission} \neq \text{reactToContact} \wedge \text{resupply} \neq \text{reactToContact}$$

[escOutput_distinct_clauses]

$$\vdash \text{ReturnToBase} \neq \text{ChangeMission} \wedge \text{ReturnToBase} \neq \text{Resupply} \wedge \\ \text{ReturnToBase} \neq \text{ReactToContact} \wedge \text{ChangeMission} \neq \text{Resupply} \wedge \\ \text{ChangeMission} \neq \text{ReactToContact} \wedge \text{Resupply} \neq \text{ReactToContact}$$

[escState_distinct_clauses]

$$\vdash \text{RTB} \neq \text{CM} \wedge \text{RTB} \neq \text{RESUPPLY} \wedge \text{RTB} \neq \text{RTC} \wedge \text{CM} \neq \text{RESUPPLY} \wedge \\ \text{CM} \neq \text{RTC} \wedge \text{RESUPPLY} \neq \text{RTC}$$

[output_distinct_clauses]

$\vdash \forall a' a. \text{ESCo } a \neq \text{SLo } a'$

[output_one_one]

$\vdash (\forall a a'. (\text{ESCo } a = \text{ESCo } a') \iff (a = a')) \wedge$
 $\quad \forall a a'. (\text{SLo } a = \text{SLo } a') \iff (a = a')$

[principal_one_one]

$\vdash \forall a a'. (\text{SR } a = \text{SR } a') \iff (a = a')$

[state_distinct_clauses]

$\vdash \forall a' a. \text{ESCs } a \neq \text{SLs } a'$

[state_one_one]

$\vdash (\forall a a'. (\text{ESCs } a = \text{ESCs } a') \iff (a = a')) \wedge$
 $\quad \forall a a'. (\text{SLs } a = \text{SLs } a') \iff (a = a')$

2 ssm11 Theory

Built: 10 June 2018

Parent Theories: satList

2.1 Datatypes

```
configuration =
  CFG (('command order, 'principal, 'd, 'e) Form -> bool)
      ('state -> ('command order, 'principal, 'd, 'e) Form)
      (('command order, 'principal, 'd, 'e) Form list)
      (('command order, 'principal, 'd, 'e) Form list) 'state
      ('output list)

order = SOME 'command | NONE

trType = discard 'command | trap 'command | exec 'command
```

2.2 Definitions

[TR_def]

$\vdash \text{TR} =$
 $\quad (\lambda a_0 a_1 a_2 a_3.$
 $\quad \quad \forall TR'.$
 $\quad \quad (\forall a_0 a_1 a_2 a_3.$
 $\quad \quad \quad (\exists \text{authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s$
 $\quad \quad \quad \quad \text{securityContext stateInterp cmd ins outs.}$
 $\quad \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec cmd}) \wedge$
 $\quad \quad \quad (a_2 =$

$$\begin{aligned}
& \text{CFG authenticationTest stateInterp} \\
& \quad \text{securityContext } (P \text{ says prop (SOME cmd)::ins) } s \\
& \quad \text{outs}) \wedge \\
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext ins (NS s (exec cmd))} \\
& \quad \quad (\text{Out s (exec cmd)::outs})) \wedge \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd))} \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext } (P \text{ says prop (SOME cmd)::ins)} \\
& \quad \quad s \text{ outs})) \vee \\
& (\exists \text{ authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s \\
& \quad \text{securityContext stateInterp cmd ins outs.} \\
& (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap cmd}) \wedge \\
& (a_2 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext } (P \text{ says prop (SOME cmd)::ins) } s \\
& \quad \quad \text{outs}) \wedge \\
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext ins (NS s (trap cmd))} \\
& \quad \quad (\text{Out s (trap cmd)::outs})) \wedge \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd))} \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext } (P \text{ says prop (SOME cmd)::ins)} \\
& \quad \quad s \text{ outs})) \vee \\
& (\exists \text{ authenticationTest NS } M \text{ Oi } Os \text{ Out } s \text{ securityContext} \\
& \quad \text{stateInterp cmd x ins outs.} \\
& (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard cmd}) \wedge \\
& (a_2 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext (x::ins) s outs}) \wedge \\
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext ins (NS s (discard cmd))} \\
& \quad \quad (\text{Out s (discard cmd)::outs})) \wedge \\
& \neg \text{authenticationTest x}) \Rightarrow \\
& TR' a_0 a_1 a_2 a_3) \Rightarrow \\
& TR' a_0 a_1 a_2 a_3)
\end{aligned}$$

2.3 Theorems

[CFGInterpret_def]

$$\begin{aligned}
& \vdash \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{input::ins) state outputStream}) \iff
\end{aligned}$$

$$(M, Oi, Os) \text{ satList } securityContext \wedge (M, Oi, Os) \text{ sat } input \wedge \\ (M, Oi, Os) \text{ sat } stateInterp \text{ state}$$

[CFGInterpret_ind]

$$\vdash \forall P. \\ (\forall M \ Oi \ Os \ authenticationTest \ stateInterp \ securityContext \\ input \ ins \ state \ outputStream. \\ P \ (M, Oi, Os) \\ (CFG \ authenticationTest \ stateInterp \ securityContext \\ (input :: ins) \ state \ outputStream)) \wedge \\ (\forall v_{15} \ v_{10} \ v_{11} \ v_{12} \ v_{13} \ v_{14}. \\ P \ v_{15} \ (CFG \ v_{10} \ v_{11} \ v_{12} \ [] \ v_{13} \ v_{14})) \Rightarrow \\ \forall v \ v_1 \ v_2 \ v_3. \ P \ (v, v_1, v_2) \ v_3$$

[configuration_one_one]

$$\vdash \forall a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5. \\ (CFG \ a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 = CFG \ a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5) \iff \\ (a_0 = a'_0) \wedge (a_1 = a'_1) \wedge (a_2 = a'_2) \wedge (a_3 = a'_3) \wedge \\ (a_4 = a'_4) \wedge (a_5 = a'_5)$$

[order_distinct_clauses]

$$\vdash \forall a. \text{ SOME } a \neq \text{ NONE}$$

[order_one_one]

$$\vdash \forall a \ a'. (\text{SOME } a = \text{SOME } a') \iff (a = a')$$

[TR_cases]

$$\vdash \forall a_0 \ a_1 \ a_2 \ a_3. \\ \text{TR } a_0 \ a_1 \ a_2 \ a_3 \iff \\ (\exists authenticationTest \ P \ NS \ M \ Oi \ Os \ Out \ s \ securityContext \\ stateInterp \ cmd \ ins \ outs. \\ (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } cmd) \wedge \\ (a_2 = \\ CFG \ authenticationTest \ stateInterp \ securityContext \\ (P \text{ says prop } (\text{SOME } cmd) :: ins) \ s \ outs) \wedge \\ (a_3 = \\ CFG \ authenticationTest \ stateInterp \ securityContext \ ins \\ (NS \ s \ (\text{exec } cmd)) \ (Out \ s \ (\text{exec } cmd) :: outs)) \wedge \\ authenticationTest \ (P \text{ says prop } (\text{SOME } cmd)) \wedge \\ CFGInterpret \ (M, Oi, Os) \\ (CFG \ authenticationTest \ stateInterp \ securityContext \\ (P \text{ says prop } (\text{SOME } cmd) :: ins) \ s \ outs)) \vee \\ (\exists authenticationTest \ P \ NS \ M \ Oi \ Os \ Out \ s \ securityContext \\ stateInterp \ cmd \ ins \ outs. \\ (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } cmd) \wedge \\ (a_2 = \\ CFG \ authenticationTest \ stateInterp \ securityContext \\ (P \text{ says prop } (\text{SOME } cmd) :: ins) \ s \ outs) \wedge$$

$$\begin{aligned}
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad (\text{NS } s \text{ (trap cmd)}) (\text{Out } s \text{ (trap cmd)::outs})) \wedge \\
& \quad \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs})) \vee \\
& \exists \text{ authenticationTest NS } M \text{ Oi Os Out } s \text{ securityContext} \\
& \quad \text{stateInterp cmd } x \text{ ins outs.} \\
& (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard cmd}) \wedge \\
& (a_2 = \\
& \quad \text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s \text{ outs}) \wedge \\
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad (\text{NS } s \text{ (discard cmd)}) (\text{Out } s \text{ (discard cmd)::outs})) \wedge \\
& \neg \text{authenticationTest } x
\end{aligned}$$

[TR_discard_cmd_rule]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) \text{ (discard cmd)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s \text{ outs}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (\text{NS } s \text{ (discard cmd)}) (\text{Out } s \text{ (discard cmd)::outs})) \iff \\
& \neg \text{authenticationTest } x
\end{aligned}$$

[TR_EQ_rules_thm]

$$\begin{aligned}
& \vdash (\text{TR } (M, Oi, Os) \text{ (exec cmd)}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (\text{NS } s \text{ (exec cmd)}) (\text{Out } s \text{ (exec cmd)::outs})) \iff \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs})) \wedge \\
& (\text{TR } (M, Oi, Os) \text{ (trap cmd)}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (\text{NS } s \text{ (trap cmd)}) (\text{Out } s \text{ (trap cmd)::outs})) \iff \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs})) \wedge \\
& (\text{TR } (M, Oi, Os) \text{ (discard cmd)}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s \text{ outs}) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins}
\end{aligned}$$

$$(NS\ s\ (\text{discard}\ cmd))\ (Out\ s\ (\text{discard}\ cmd)::outs)) \iff \neg authenticationTest\ x)$$

[TR_exec_cmd_rule]

$$\begin{aligned} &\vdash \forall authenticationTest\ securityContext\ stateInterp\ P\ cmd\ ins\ s\ outs. \\ &\quad (\forall M\ Oi\ Os. \\ &\quad \quad CFGInterpret\ (M, Oi, Os) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \Rightarrow \\ &\quad \quad \quad (M, Oi, Os)\ \text{sat}\ \text{prop}\ (SOME\ cmd)) \Rightarrow \\ &\quad \forall NS\ Out\ M\ Oi\ Os. \\ &\quad TR\ (M, Oi, Os)\ (\text{exec}\ cmd) \\ &\quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \\ &\quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext\ ins \\ &\quad \quad \quad (NS\ s\ (\text{exec}\ cmd))\ (Out\ s\ (\text{exec}\ cmd)::outs)) \iff \\ &\quad authenticationTest\ (P\ \text{says}\ \text{prop}\ (SOME\ cmd)) \wedge \\ &\quad CFGInterpret\ (M, Oi, Os) \\ &\quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \wedge \\ &\quad (M, Oi, Os)\ \text{sat}\ \text{prop}\ (SOME\ cmd)) \end{aligned}$$

[TR_ind]

$$\begin{aligned} &\vdash \forall TR'. \\ &\quad (\forall authenticationTest\ P\ NS\ M\ Oi\ Os\ Out\ s\ securityContext \\ &\quad \quad stateInterp\ cmd\ ins\ outs. \\ &\quad \quad authenticationTest\ (P\ \text{says}\ \text{prop}\ (SOME\ cmd)) \wedge \\ &\quad \quad CFGInterpret\ (M, Oi, Os) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \Rightarrow \\ &\quad \quad TR'\ (M, Oi, Os)\ (\text{exec}\ cmd) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext\ ins \\ &\quad \quad \quad \quad (NS\ s\ (\text{exec}\ cmd))\ (Out\ s\ (\text{exec}\ cmd)::outs))) \wedge \\ &\quad (\forall authenticationTest\ P\ NS\ M\ Oi\ Os\ Out\ s\ securityContext \\ &\quad \quad stateInterp\ cmd\ ins\ outs. \\ &\quad \quad authenticationTest\ (P\ \text{says}\ \text{prop}\ (SOME\ cmd)) \wedge \\ &\quad \quad CFGInterpret\ (M, Oi, Os) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \Rightarrow \\ &\quad \quad TR'\ (M, Oi, Os)\ (\text{trap}\ cmd) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext \\ &\quad \quad \quad \quad (P\ \text{says}\ \text{prop}\ (SOME\ cmd)::ins)\ s\ outs) \\ &\quad \quad \quad (CFG\ authenticationTest\ stateInterp\ securityContext\ ins \\ &\quad \quad \quad \quad (NS\ s\ (\text{trap}\ cmd))\ (Out\ s\ (\text{trap}\ cmd)::outs))) \wedge \\ &\quad (\forall authenticationTest\ NS\ M\ Oi\ Os\ Out\ s\ securityContext \\ &\quad \quad stateInterp\ cmd\ x\ ins\ outs. \end{aligned}$$

$$\begin{aligned}
& \neg \text{authenticationTest } x \Rightarrow \\
& \text{TR}' (M, Oi, Os) (\text{discard } cmd) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x :: ins) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad ins (NS s (\text{discard } cmd))) \\
& \quad (Out s (\text{discard } cmd) :: outs))) \Rightarrow \\
& \forall a_0 a_1 a_2 a_3. \text{TR } a_0 a_1 a_2 a_3 \Rightarrow \text{TR}' a_0 a_1 a_2 a_3
\end{aligned}$$

[TR_rules]

$$\begin{aligned}
& \vdash (\forall \text{authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s \text{ securityContext} \\
& \quad \text{stateInterp } cmd \text{ ins } outs. \\
& \quad \text{authenticationTest } (P \text{ says prop (SOME } cmd)) \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME } cmd) :: ins) s outs) \Rightarrow \\
& \quad \text{TR } (M, Oi, Os) (\text{exec } cmd) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME } cmd) :: ins) s outs) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (NS s (\text{exec } cmd)) (Out s (\text{exec } cmd) :: outs))) \wedge \\
& (\forall \text{authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s \text{ securityContext} \\
& \quad \text{stateInterp } cmd \text{ ins } outs. \\
& \quad \text{authenticationTest } (P \text{ says prop (SOME } cmd)) \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME } cmd) :: ins) s outs) \Rightarrow \\
& \quad \text{TR } (M, Oi, Os) (\text{trap } cmd) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME } cmd) :: ins) s outs) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (NS s (\text{trap } cmd)) (Out s (\text{trap } cmd) :: outs))) \wedge \\
& \forall \text{authenticationTest } NS \text{ M } Oi \text{ Os } Out \text{ s securityContext} \\
& \quad \text{stateInterp } cmd \text{ x ins } outs. \\
& \neg \text{authenticationTest } x \Rightarrow \\
& \text{TR } (M, Oi, Os) (\text{discard } cmd) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x :: ins) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad (NS s (\text{discard } cmd)) (Out s (\text{discard } cmd) :: outs)))
\end{aligned}$$

[TR_strongind]

$$\begin{aligned}
& \vdash \forall \text{TR}'. \\
& \quad (\forall \text{authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s \text{ securityContext} \\
& \quad \quad \text{stateInterp } cmd \text{ ins } outs. \\
& \quad \quad \text{authenticationTest } (P \text{ says prop (SOME } cmd)) \wedge \\
& \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad \quad (P \text{ says prop (SOME } cmd) :: ins) s outs) \Rightarrow
\end{aligned}$$

$$\begin{aligned}
& TR' (M, Oi, Os) (\text{exec } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \text{ins (NS } s (\text{exec } cmd)) (\text{Out } s (\text{exec } cmd)::outs))) \wedge \\
& (\forall \text{ authenticationTest } P \text{ NS } M \text{ Oi } Os \text{ Out } s \text{ securityContext} \\
& \quad \text{stateInterp } cmd \text{ ins outs.} \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd))} \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \Rightarrow \\
& TR' (M, Oi, Os) (\text{trap } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \text{ins (NS } s (\text{trap } cmd)) (\text{Out } s (\text{trap } cmd)::outs))) \wedge \\
& (\forall \text{ authenticationTest } NS \text{ M } Oi \text{ Os } Out \text{ s securityContext} \\
& \quad \text{stateInterp } cmd \text{ x ins outs.} \\
& \neg \text{authenticationTest } x \Rightarrow \\
& TR' (M, Oi, Os) (\text{discard } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s \text{ outs}) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \text{ins (NS } s (\text{discard } cmd)) \\
& \quad \quad \quad (\text{Out } s (\text{discard } cmd)::outs))) \Rightarrow \\
& \forall a_0 \ a_1 \ a_2 \ a_3. \text{ TR } a_0 \ a_1 \ a_2 \ a_3 \Rightarrow TR' a_0 \ a_1 \ a_2 \ a_3
\end{aligned}$$

[TR_trap_cmd_rule]

$$\begin{aligned}
& \vdash \forall \text{ authenticationTest stateInterp securityContext } P \text{ cmd ins s} \\
& \quad \text{outs.} \\
& (\forall M \text{ Oi } Os. \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \Rightarrow \\
& \quad \quad (M, Oi, Os) \text{ sat prop NONE}) \Rightarrow \\
& \forall NS \text{ Out } M \text{ Oi } Os. \\
& \text{TR } (M, Oi, Os) (\text{trap } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad (\text{NS } s (\text{trap } cmd)) (\text{Out } s (\text{trap } cmd)::outs)) \iff \\
& \text{authenticationTest } (P \text{ says prop (SOME cmd))} \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \wedge \\
& \quad (M, Oi, Os) \text{ sat prop NONE}
\end{aligned}$$

[TRrule0]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) (\text{exec } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext}
\end{aligned}$$

```

(P says prop (SOME cmd)::ins) s outs)
(CFG authenticationTest stateInterp securityContext ins
 (NS s (exec cmd)) (Out s (exec cmd)::outs))  $\iff$ 
authenticationTest (P says prop (SOME cmd))  $\wedge$ 
CFGInterpret (M, Oi, Os)
 (CFG authenticationTest stateInterp securityContext
  (P says prop (SOME cmd)::ins) s outs)

```

[TRrule1]

```

 $\vdash$  TR (M, Oi, Os) (trap cmd)
  (CFG authenticationTest stateInterp securityContext
   (P says prop (SOME cmd)::ins) s outs)
  (CFG authenticationTest stateInterp securityContext ins
   (NS s (trap cmd)) (Out s (trap cmd)::outs))  $\iff$ 
authenticationTest (P says prop (SOME cmd))  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG authenticationTest stateInterp securityContext
   (P says prop (SOME cmd)::ins) s outs)

```

[trType_distinct_clauses]

```

 $\vdash (\forall a' a. \text{discard } a \neq \text{trap } a') \wedge (\forall a' a. \text{discard } a \neq \text{exec } a') \wedge$ 
 $\forall a' a. \text{trap } a \neq \text{exec } a'$ 

```

[trType_one_one]

```

 $\vdash (\forall a a'. (\text{discard } a = \text{discard } a') \iff (a = a')) \wedge$ 
 $(\forall a a'. (\text{trap } a = \text{trap } a') \iff (a = a')) \wedge$ 
 $\forall a a'. (\text{exec } a = \text{exec } a') \iff (a = a')$ 

```

3 ssm Theory

Built: 10 June 2018

Parent Theories: satList

3.1 Datatypes

```

configuration =
  CFG (('command option, 'principal, 'd, 'e) Form -> bool)
    ('state ->
      ('command option, 'principal, 'd, 'e) Form list ->
        ('command option, 'principal, 'd, 'e) Form list)
      (('command option, 'principal, 'd, 'e) Form list ->
        ('command option, 'principal, 'd, 'e) Form list)
      (('command option, 'principal, 'd, 'e) Form list list)
      'state ('output list)

trType = discard 'cmdlist | trap 'cmdlist | exec 'cmdlist

```

3.2 Definitions

[authenticationTest_def]

$$\vdash \forall \text{elementTest } x. \\ \text{authenticationTest } \text{elementTest } x \iff \\ \text{FOLDR } (\lambda p \ q. \ p \wedge \ q) \ \text{T} \ (\text{MAP } \text{elementTest } x)$$

[commandList_def]

$$\vdash \forall x. \text{commandList } x = \text{MAP } \text{extractCommand } x$$

[inputList_def]

$$\vdash \forall xs. \text{inputList } xs = \text{MAP } \text{extractInput } xs$$

[propCommandList_def]

$$\vdash \forall x. \text{propCommandList } x = \text{MAP } \text{extractPropCommand } x$$

[TR_def]

$$\vdash \text{TR} = \\ (\lambda a_0 \ a_1 \ a_2 \ a_3. \\ \forall TR'. \\ (\forall a_0 \ a_1 \ a_2 \ a_3. \\ (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } \text{stateInterp } x \\ \text{ins } \text{outs}. \\ (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } (\text{inputList } x)) \wedge \\ (a_2 = \\ \text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\ \text{outs}) \wedge \\ (a_3 = \\ \text{CFG } \text{elementTest } \text{stateInterp } \text{context } \text{ins} \\ (NS \ s \ (\text{exec } (\text{inputList } x))) \\ (Out \ s \ (\text{exec } (\text{inputList } x)::\text{outs})) \wedge \\ \text{authenticationTest } \text{elementTest } x \wedge \\ \text{CFGInterpret } (M, Oi, Os) \\ (\text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\ \text{outs})) \vee \\ (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } \text{stateInterp } x \\ \text{ins } \text{outs}. \\ (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } (\text{inputList } x)) \wedge \\ (a_2 = \\ \text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\ \text{outs}) \wedge \\ (a_3 = \\ \text{CFG } \text{elementTest } \text{stateInterp } \text{context } \text{ins} \\ (NS \ s \ (\text{trap } (\text{inputList } x))) \\ (Out \ s \ (\text{trap } (\text{inputList } x)::\text{outs})) \wedge \\ \text{authenticationTest } \text{elementTest } x \wedge \\ \text{CFGInterpret } (M, Oi, Os) \\ (\text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s$$

$$\begin{aligned}
& \text{outs})) \vee \\
& (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } stateInterp \ x \\
& \quad \text{ins } outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \quad \text{CFG } \text{elementTest } stateInterp \ \text{context } (x::ins) \ s \\
& \quad \quad \text{outs}) \wedge \\
& \quad (a_3 = \\
& \quad \quad \text{CFG } \text{elementTest } stateInterp \ \text{context } ins \\
& \quad \quad (NS \ s \ (\text{discard } (\text{inputList } x))) \\
& \quad \quad (Out \ s \ (\text{discard } (\text{inputList } x))::outs)) \wedge \\
& \quad \neg \text{authenticationTest } \text{elementTest } x) \Rightarrow \\
& TR' \ a_0 \ a_1 \ a_2 \ a_3) \Rightarrow \\
& TR' \ a_0 \ a_1 \ a_2 \ a_3)
\end{aligned}$$

3.3 Theorems

[CFGInterpret_def]

$$\begin{aligned}
& \vdash \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG } \text{elementTest } stateInterp \ \text{context } (x::ins) \ state \\
& \quad \quad \text{outStream}) \iff \\
& \quad (M, Oi, Os) \ \text{satList } \text{context } x \wedge (M, Oi, Os) \ \text{satList } x \wedge \\
& \quad (M, Oi, Os) \ \text{satList } stateInterp \ state \ x
\end{aligned}$$

[CFGInterpret_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& \quad (\forall M \ Oi \ Os \ \text{elementTest } stateInterp \ \text{context } x \ \text{ins } state \\
& \quad \quad \text{outStream}. \\
& \quad \quad P \ (M, Oi, Os) \\
& \quad \quad (\text{CFG } \text{elementTest } stateInterp \ \text{context } (x::ins) \ state \\
& \quad \quad \quad \text{outStream})) \wedge \\
& \quad (\forall v_{15} \ v_{10} \ v_{11} \ v_{12} \ v_{13} \ v_{14}. \\
& \quad \quad P \ v_{15} \ (\text{CFG } v_{10} \ v_{11} \ v_{12} \ [] \ v_{13} \ v_{14})) \Rightarrow \\
& \quad \forall v \ v_1 \ v_2 \ v_3. \ P \ (v, v_1, v_2) \ v_3
\end{aligned}$$

[configuration_one_one]

$$\begin{aligned}
& \vdash \forall a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5. \\
& \quad (\text{CFG } a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 = \text{CFG } a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5) \iff \\
& \quad (a_0 = a'_0) \wedge (a_1 = a'_1) \wedge (a_2 = a'_2) \wedge (a_3 = a'_3) \wedge \\
& \quad (a_4 = a'_4) \wedge (a_5 = a'_5)
\end{aligned}$$

[extractCommand_def]

$$\vdash \text{extractCommand } (P \ \text{says prop } (\text{SOME } cmd)) = cmd$$

[extractCommand_ind]

$$\begin{aligned}
& \vdash \forall P'. \\
& \quad (\forall P \ cmd. \ P' \ (P \ \text{says prop } (\text{SOME } cmd))) \wedge P' \ \text{TT} \wedge P' \ \text{FF} \wedge \\
& \quad (\forall v_1. \ P' \ (\text{prop } v_1)) \wedge (\forall v_3. \ P' \ (\text{notf } v_3)) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_6 v_7. P' (v_6 \text{ andf } v_7)) \wedge (\forall v_{10} v_{11}. P' (v_{10} \text{ orf } v_{11})) \wedge \\
& (\forall v_{14} v_{15}. P' (v_{14} \text{ impf } v_{15})) \wedge \\
& (\forall v_{18} v_{19}. P' (v_{18} \text{ eqf } v_{19})) \wedge (\forall v_{129}. P' (v_{129} \text{ says TT})) \wedge \\
& (\forall v_{130}. P' (v_{130} \text{ says FF})) \wedge \\
& (\forall v_{132}. P' (v_{132} \text{ says prop NONE})) \wedge \\
& (\forall v_{133} v_{66}. P' (v_{133} \text{ says notf } v_{66})) \wedge \\
& (\forall v_{134} v_{69} v_{70}. P' (v_{134} \text{ says } (v_{69} \text{ andf } v_{70}))) \wedge \\
& (\forall v_{135} v_{73} v_{74}. P' (v_{135} \text{ says } (v_{73} \text{ orf } v_{74}))) \wedge \\
& (\forall v_{136} v_{77} v_{78}. P' (v_{136} \text{ says } (v_{77} \text{ impf } v_{78}))) \wedge \\
& (\forall v_{137} v_{81} v_{82}. P' (v_{137} \text{ says } (v_{81} \text{ eqf } v_{82}))) \wedge \\
& (\forall v_{138} v_{85} v_{86}. P' (v_{138} \text{ says } v_{85} \text{ says } v_{86})) \wedge \\
& (\forall v_{139} v_{89} v_{90}. P' (v_{139} \text{ says } v_{89} \text{ speaks_for } v_{90})) \wedge \\
& (\forall v_{140} v_{93} v_{94}. P' (v_{140} \text{ says } v_{93} \text{ controls } v_{94})) \wedge \\
& (\forall v_{141} v_{98} v_{99} v_{100}. P' (v_{141} \text{ says reps } v_{98} v_{99} v_{100})) \wedge \\
& (\forall v_{142} v_{103} v_{104}. P' (v_{142} \text{ says } v_{103} \text{ domi } v_{104})) \wedge \\
& (\forall v_{143} v_{107} v_{108}. P' (v_{143} \text{ says } v_{107} \text{ eqi } v_{108})) \wedge \\
& (\forall v_{144} v_{111} v_{112}. P' (v_{144} \text{ says } v_{111} \text{ doms } v_{112})) \wedge \\
& (\forall v_{145} v_{115} v_{116}. P' (v_{145} \text{ says } v_{115} \text{ eqs } v_{116})) \wedge \\
& (\forall v_{146} v_{119} v_{120}. P' (v_{146} \text{ says } v_{119} \text{ eqn } v_{120})) \wedge \\
& (\forall v_{147} v_{123} v_{124}. P' (v_{147} \text{ says } v_{123} \text{ lte } v_{124})) \wedge \\
& (\forall v_{148} v_{127} v_{128}. P' (v_{148} \text{ says } v_{127} \text{ lt } v_{128})) \wedge \\
& (\forall v_{24} v_{25}. P' (v_{24} \text{ speaks_for } v_{25})) \wedge \\
& (\forall v_{28} v_{29}. P' (v_{28} \text{ controls } v_{29})) \wedge \\
& (\forall v_{33} v_{34} v_{35}. P' (\text{reps } v_{33} v_{34} v_{35})) \wedge \\
& (\forall v_{38} v_{39}. P' (v_{38} \text{ domi } v_{39})) \wedge \\
& (\forall v_{42} v_{43}. P' (v_{42} \text{ eqi } v_{43})) \wedge \\
& (\forall v_{46} v_{47}. P' (v_{46} \text{ doms } v_{47})) \wedge \\
& (\forall v_{50} v_{51}. P' (v_{50} \text{ eqs } v_{51})) \wedge \\
& (\forall v_{54} v_{55}. P' (v_{54} \text{ eqn } v_{55})) \wedge \\
& (\forall v_{58} v_{59}. P' (v_{58} \text{ lte } v_{59})) \wedge \\
& (\forall v_{62} v_{63}. P' (v_{62} \text{ lt } v_{63})) \Rightarrow \\
& \forall v. P' v
\end{aligned}$$

[extractInput_def]

$\vdash \text{extractInput } (P \text{ says prop } x) = x$

[extractInput_ind]

$\vdash \forall P'.$

$$\begin{aligned}
& (\forall P x. P' (P \text{ says prop } x)) \wedge P' \text{ TT} \wedge P' \text{ FF} \wedge \\
& (\forall v_1. P' (\text{prop } v_1)) \wedge (\forall v_3. P' (\text{notf } v_3)) \wedge \\
& (\forall v_6 v_7. P' (v_6 \text{ andf } v_7)) \wedge (\forall v_{10} v_{11}. P' (v_{10} \text{ orf } v_{11})) \wedge \\
& (\forall v_{14} v_{15}. P' (v_{14} \text{ impf } v_{15})) \wedge \\
& (\forall v_{18} v_{19}. P' (v_{18} \text{ eqf } v_{19})) \wedge (\forall v_{129}. P' (v_{129} \text{ says TT})) \wedge \\
& (\forall v_{130}. P' (v_{130} \text{ says FF})) \wedge \\
& (\forall v_{131} v_{66}. P' (v_{131} \text{ says notf } v_{66})) \wedge \\
& (\forall v_{132} v_{69} v_{70}. P' (v_{132} \text{ says } (v_{69} \text{ andf } v_{70}))) \wedge \\
& (\forall v_{133} v_{73} v_{74}. P' (v_{133} \text{ says } (v_{73} \text{ orf } v_{74}))) \wedge \\
& (\forall v_{134} v_{77} v_{78}. P' (v_{134} \text{ says } (v_{77} \text{ impf } v_{78}))) \wedge \\
& (\forall v_{135} v_{81} v_{82}. P' (v_{135} \text{ says } (v_{81} \text{ eqf } v_{82}))) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v136 \ v85 \ v86. \ P' \ (v136 \ \text{says} \ v85 \ \text{says} \ v86)) \wedge \\
& (\forall v137 \ v89 \ v90. \ P' \ (v137 \ \text{says} \ v89 \ \text{speaks_for} \ v90)) \wedge \\
& (\forall v138 \ v93 \ v94. \ P' \ (v138 \ \text{says} \ v93 \ \text{controls} \ v94)) \wedge \\
& (\forall v139 \ v98 \ v99 \ v100. \ P' \ (v139 \ \text{says} \ \text{reps} \ v98 \ v99 \ v100)) \wedge \\
& (\forall v140 \ v103 \ v104. \ P' \ (v140 \ \text{says} \ v103 \ \text{domi} \ v104)) \wedge \\
& (\forall v141 \ v107 \ v108. \ P' \ (v141 \ \text{says} \ v107 \ \text{eqi} \ v108)) \wedge \\
& (\forall v142 \ v111 \ v112. \ P' \ (v142 \ \text{says} \ v111 \ \text{doms} \ v112)) \wedge \\
& (\forall v143 \ v115 \ v116. \ P' \ (v143 \ \text{says} \ v115 \ \text{eqs} \ v116)) \wedge \\
& (\forall v144 \ v119 \ v120. \ P' \ (v144 \ \text{says} \ v119 \ \text{eqn} \ v120)) \wedge \\
& (\forall v145 \ v123 \ v124. \ P' \ (v145 \ \text{says} \ v123 \ \text{lte} \ v124)) \wedge \\
& (\forall v146 \ v127 \ v128. \ P' \ (v146 \ \text{says} \ v127 \ \text{lt} \ v128)) \wedge \\
& (\forall v24 \ v25. \ P' \ (v24 \ \text{speaks_for} \ v25)) \wedge \\
& (\forall v28 \ v29. \ P' \ (v28 \ \text{controls} \ v29)) \wedge \\
& (\forall v33 \ v34 \ v35. \ P' \ (\text{reps} \ v33 \ v34 \ v35)) \wedge \\
& (\forall v38 \ v39. \ P' \ (v38 \ \text{domi} \ v39)) \wedge \\
& (\forall v42 \ v43. \ P' \ (v42 \ \text{eqi} \ v43)) \wedge \\
& (\forall v46 \ v47. \ P' \ (v46 \ \text{doms} \ v47)) \wedge \\
& (\forall v50 \ v51. \ P' \ (v50 \ \text{eqs} \ v51)) \wedge \\
& (\forall v54 \ v55. \ P' \ (v54 \ \text{eqn} \ v55)) \wedge \\
& (\forall v58 \ v59. \ P' \ (v58 \ \text{lte} \ v59)) \wedge \\
& (\forall v62 \ v63. \ P' \ (v62 \ \text{lt} \ v63)) \Rightarrow \\
& \forall v. \ P' \ v
\end{aligned}$$

[extractPropCommand_def]

$$\vdash \text{extractPropCommand} \ (P \ \text{says} \ \text{prop} \ (\text{SOME} \ \text{cmd})) = \text{prop} \ (\text{SOME} \ \text{cmd})$$

[extractPropCommand_ind]

$$\begin{aligned}
& \vdash \forall P'. \\
& \quad (\forall P \ \text{cmd}. \ P' \ (P \ \text{says} \ \text{prop} \ (\text{SOME} \ \text{cmd}))) \wedge P' \ \text{TT} \wedge P' \ \text{FF} \wedge \\
& \quad (\forall v_1. \ P' \ (\text{prop} \ v_1)) \wedge (\forall v_3. \ P' \ (\text{notf} \ v_3)) \wedge \\
& \quad (\forall v_6 \ v_7. \ P' \ (v_6 \ \text{andf} \ v_7)) \wedge (\forall v_{10} \ v_{11}. \ P' \ (v_{10} \ \text{orf} \ v_{11})) \wedge \\
& \quad (\forall v_{14} \ v_{15}. \ P' \ (v_{14} \ \text{impf} \ v_{15})) \wedge \\
& \quad (\forall v_{18} \ v_{19}. \ P' \ (v_{18} \ \text{eqf} \ v_{19})) \wedge (\forall v_{129}. \ P' \ (v_{129} \ \text{says} \ \text{TT})) \wedge \\
& \quad (\forall v_{130}. \ P' \ (v_{130} \ \text{says} \ \text{FF})) \wedge \\
& \quad (\forall v_{132}. \ P' \ (v_{132} \ \text{says} \ \text{prop} \ \text{NONE})) \wedge \\
& \quad (\forall v_{133} \ v_{66}. \ P' \ (v_{133} \ \text{says} \ \text{notf} \ v_{66})) \wedge \\
& \quad (\forall v_{134} \ v_{69} \ v_{70}. \ P' \ (v_{134} \ \text{says} \ (v_{69} \ \text{andf} \ v_{70}))) \wedge \\
& \quad (\forall v_{135} \ v_{73} \ v_{74}. \ P' \ (v_{135} \ \text{says} \ (v_{73} \ \text{orf} \ v_{74}))) \wedge \\
& \quad (\forall v_{136} \ v_{77} \ v_{78}. \ P' \ (v_{136} \ \text{says} \ (v_{77} \ \text{impf} \ v_{78}))) \wedge \\
& \quad (\forall v_{137} \ v_{81} \ v_{82}. \ P' \ (v_{137} \ \text{says} \ (v_{81} \ \text{eqf} \ v_{82}))) \wedge \\
& \quad (\forall v_{138} \ v_{85} \ v_{86}. \ P' \ (v_{138} \ \text{says} \ v_{85} \ \text{says} \ v_{86})) \wedge \\
& \quad (\forall v_{139} \ v_{89} \ v_{90}. \ P' \ (v_{139} \ \text{says} \ v_{89} \ \text{speaks_for} \ v_{90})) \wedge \\
& \quad (\forall v_{140} \ v_{93} \ v_{94}. \ P' \ (v_{140} \ \text{says} \ v_{93} \ \text{controls} \ v_{94})) \wedge \\
& \quad (\forall v_{141} \ v_{98} \ v_{99} \ v_{100}. \ P' \ (v_{141} \ \text{says} \ \text{reps} \ v_{98} \ v_{99} \ v_{100})) \wedge \\
& \quad (\forall v_{142} \ v_{103} \ v_{104}. \ P' \ (v_{142} \ \text{says} \ v_{103} \ \text{domi} \ v_{104})) \wedge \\
& \quad (\forall v_{143} \ v_{107} \ v_{108}. \ P' \ (v_{143} \ \text{says} \ v_{107} \ \text{eqi} \ v_{108})) \wedge \\
& \quad (\forall v_{144} \ v_{111} \ v_{112}. \ P' \ (v_{144} \ \text{says} \ v_{111} \ \text{doms} \ v_{112})) \wedge \\
& \quad (\forall v_{145} \ v_{115} \ v_{116}. \ P' \ (v_{145} \ \text{says} \ v_{115} \ \text{eqs} \ v_{116})) \wedge \\
& \quad (\forall v_{146} \ v_{119} \ v_{120}. \ P' \ (v_{146} \ \text{says} \ v_{119} \ \text{eqn} \ v_{120})) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_{147} v_{123} v_{124}. P' (v_{147} \text{ says } v_{123} \text{ lte } v_{124})) \wedge \\
& (\forall v_{148} v_{127} v_{128}. P' (v_{148} \text{ says } v_{127} \text{ lt } v_{128})) \wedge \\
& (\forall v_{24} v_{25}. P' (v_{24} \text{ speaks_for } v_{25})) \wedge \\
& (\forall v_{28} v_{29}. P' (v_{28} \text{ controls } v_{29})) \wedge \\
& (\forall v_{33} v_{34} v_{35}. P' (\text{reps } v_{33} v_{34} v_{35})) \wedge \\
& (\forall v_{38} v_{39}. P' (v_{38} \text{ domi } v_{39})) \wedge \\
& (\forall v_{42} v_{43}. P' (v_{42} \text{ eqi } v_{43})) \wedge \\
& (\forall v_{46} v_{47}. P' (v_{46} \text{ doms } v_{47})) \wedge \\
& (\forall v_{50} v_{51}. P' (v_{50} \text{ eqs } v_{51})) \wedge \\
& (\forall v_{54} v_{55}. P' (v_{54} \text{ eqn } v_{55})) \wedge \\
& (\forall v_{58} v_{59}. P' (v_{58} \text{ lte } v_{59})) \wedge \\
& (\forall v_{62} v_{63}. P' (v_{62} \text{ lt } v_{63})) \Rightarrow \\
& \forall v. P' v
\end{aligned}$$

[TR_cases]

$$\begin{aligned}
& \vdash \forall a_0 a_1 a_2 a_3. \\
& \text{TR } a_0 a_1 a_2 a_3 \iff \\
& (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 = \\
& \quad \quad \text{CFG elementTest stateInterp context } ins \\
& \quad \quad \quad (NS \ s \ (\text{exec } (\text{inputList } x))) \\
& \quad \quad \quad (Out \ s \ (\text{exec } (\text{inputList } x))::outs)) \wedge \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \\
& \quad \quad \quad outs)) \vee \\
& (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 = \\
& \quad \quad \text{CFG elementTest stateInterp context } ins \\
& \quad \quad \quad (NS \ s \ (\text{trap } (\text{inputList } x))) \\
& \quad \quad \quad (Out \ s \ (\text{trap } (\text{inputList } x))::outs)) \wedge \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \\
& \quad \quad \quad outs)) \vee \\
& \exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 =
\end{aligned}$$

CFG elementTest stateInterp context ins
 (NS s (discard (inputList x)))
 (Out s (discard (inputList x))::outs)) \wedge
 \neg authenticationTest elementTest x

[TR_discard_cmd_rule]

\vdash TR (M, Oi, Os) (discard (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (discard (inputList x)))
 (Out s (discard (inputList x))::outs)) \iff
 \neg authenticationTest elementTest x

[TR_EQ_rules_thm]

\vdash (TR (M, Oi, Os) (exec (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (exec (inputList x)))
 (Out s (exec (inputList x))::outs)) \iff
 authenticationTest elementTest x \wedge
 CFGInterpret (M, Oi, Os)
 (CFG elementTest stateInterp context (x::ins) s outs)) \wedge
 (TR (M, Oi, Os) (trap (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (trap (inputList x)))
 (Out s (trap (inputList x))::outs)) \iff
 authenticationTest elementTest x \wedge
 CFGInterpret (M, Oi, Os)
 (CFG elementTest stateInterp context (x::ins) s outs)) \wedge
 (TR (M, Oi, Os) (discard (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (discard (inputList x)))
 (Out s (discard (inputList x))::outs)) \iff
 \neg authenticationTest elementTest x)

[TR_exec_cmd_rule]

$\vdash \forall$ elementTest context stateInterp x ins s outs.
 (\forall M Oi Os.
 CFGInterpret (M, Oi, Os)
 (CFG elementTest stateInterp context (x::ins) s
 outs) \Rightarrow
 (M, Oi, Os) satList propCommandList x) \Rightarrow
 \forall NS Out M Oi Os.
 TR (M, Oi, Os) (exec (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins

$$\begin{aligned}
& (NS \ s \ (\text{exec} \ (\text{inputList} \ x))) \\
& (\text{Out} \ s \ (\text{exec} \ (\text{inputList} \ x))::\text{outs})) \iff \\
& \text{authenticationTest} \ \text{elementTest} \ x \wedge \\
& \text{CFGInterpret} \ (M, Oi, Os) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs}) \wedge \\
& (M, Oi, Os) \ \text{satList} \ \text{propCommandList} \ x
\end{aligned}$$

[TR_ind]

$\vdash \forall TR'.$

$$\begin{aligned}
& (\forall \text{elementTest} \ NS \ M \ Oi \ Os \ Out \ s \ \text{context} \ \text{stateInterp} \ x \ \text{ins} \\
& \quad \text{outs}. \\
& \text{authenticationTest} \ \text{elementTest} \ x \wedge \\
& \text{CFGInterpret} \ (M, Oi, Os) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \\
& \quad \text{outs}) \Rightarrow \\
& TR' \ (M, Oi, Os) \ (\text{exec} \ (\text{inputList} \ x)) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs}) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ \text{ins} \\
& \quad \quad (NS \ s \ (\text{exec} \ (\text{inputList} \ x))) \\
& \quad \quad (\text{Out} \ s \ (\text{exec} \ (\text{inputList} \ x))::\text{outs}))) \wedge \\
& (\forall \text{elementTest} \ NS \ M \ Oi \ Os \ Out \ s \ \text{context} \ \text{stateInterp} \ x \ \text{ins} \\
& \quad \text{outs}. \\
& \text{authenticationTest} \ \text{elementTest} \ x \wedge \\
& \text{CFGInterpret} \ (M, Oi, Os) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \\
& \quad \text{outs}) \Rightarrow \\
& TR' \ (M, Oi, Os) \ (\text{trap} \ (\text{inputList} \ x)) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs}) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ \text{ins} \\
& \quad \quad (NS \ s \ (\text{trap} \ (\text{inputList} \ x))) \\
& \quad \quad (\text{Out} \ s \ (\text{trap} \ (\text{inputList} \ x))::\text{outs}))) \wedge \\
& (\forall \text{elementTest} \ NS \ M \ Oi \ Os \ Out \ s \ \text{context} \ \text{stateInterp} \ x \ \text{ins} \\
& \quad \text{outs}. \\
& \neg \text{authenticationTest} \ \text{elementTest} \ x \Rightarrow \\
& TR' \ (M, Oi, Os) \ (\text{discard} \ (\text{inputList} \ x)) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs}) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ \text{ins} \\
& \quad \quad (NS \ s \ (\text{discard} \ (\text{inputList} \ x))) \\
& \quad \quad (\text{Out} \ s \ (\text{discard} \ (\text{inputList} \ x))::\text{outs}))) \Rightarrow \\
& \forall a_0 \ a_1 \ a_2 \ a_3. \ TR \ a_0 \ a_1 \ a_2 \ a_3 \Rightarrow TR' \ a_0 \ a_1 \ a_2 \ a_3
\end{aligned}$$

[TR_rules]

$$\begin{aligned}
& \vdash (\forall \text{elementTest} \ NS \ M \ Oi \ Os \ Out \ s \ \text{context} \ \text{stateInterp} \ x \ \text{ins} \\
& \quad \text{outs}. \\
& \text{authenticationTest} \ \text{elementTest} \ x \wedge \\
& \text{CFGInterpret} \ (M, Oi, Os) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs}) \Rightarrow \\
& TR \ (M, Oi, Os) \ (\text{exec} \ (\text{inputList} \ x)) \\
& \quad (\text{CFG} \ \text{elementTest} \ \text{stateInterp} \ \text{context} \ (x::\text{ins}) \ s \ \text{outs})
\end{aligned}$$

```

(CFG elementTest stateInterp context ins
  (NS s (exec (inputList x)))
  (Out s (exec (inputList x))::outs))) ∧
(∀ elementTest NS M Oi Os Out s context stateInterp x ins
  outs.
  authenticationTest elementTest x ∧
  CFGInterpret (M, Oi, Os)
    (CFG elementTest stateInterp context (x::ins) s outs) ⇒
  TR (M, Oi, Os) (trap (inputList x))
    (CFG elementTest stateInterp context (x::ins) s outs)
    (CFG elementTest stateInterp context ins
      (NS s (trap (inputList x)))
      (Out s (trap (inputList x))::outs))) ∧
  ∀ elementTest NS M Oi Os Out s context stateInterp x ins outs.
    ¬authenticationTest elementTest x ⇒
  TR (M, Oi, Os) (discard (inputList x))
    (CFG elementTest stateInterp context (x::ins) s outs)
    (CFG elementTest stateInterp context ins
      (NS s (discard (inputList x)))
      (Out s (discard (inputList x))::outs)))

```

[TR_strongind]

```

⊢ ∀ TR'.
  (∀ elementTest NS M Oi Os Out s context stateInterp x ins
    outs.
    authenticationTest elementTest x ∧
    CFGInterpret (M, Oi, Os)
      (CFG elementTest stateInterp context (x::ins) s
        outs) ⇒
    TR' (M, Oi, Os) (exec (inputList x))
      (CFG elementTest stateInterp context (x::ins) s outs)
      (CFG elementTest stateInterp context ins
        (NS s (exec (inputList x)))
        (Out s (exec (inputList x))::outs))) ∧
    (∀ elementTest NS M Oi Os Out s context stateInterp x ins
      outs.
      authenticationTest elementTest x ∧
      CFGInterpret (M, Oi, Os)
        (CFG elementTest stateInterp context (x::ins) s
          outs) ⇒
      TR' (M, Oi, Os) (trap (inputList x))
        (CFG elementTest stateInterp context (x::ins) s outs)
        (CFG elementTest stateInterp context ins
          (NS s (trap (inputList x)))
          (Out s (trap (inputList x))::outs))) ∧
      (∀ elementTest NS M Oi Os Out s context stateInterp x ins
        outs.
        ¬authenticationTest elementTest x ⇒
        TR' (M, Oi, Os) (discard (inputList x))

```

$$\begin{aligned}
& (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs}) \\
& (\text{CFG elementTest stateInterp context ins} \\
& \quad (\text{NS s (discard (inputList x))}) \\
& \quad (\text{Out s (discard (inputList x))::outs})) \Rightarrow \\
& \forall a_0 \ a_1 \ a_2 \ a_3. \text{ TR } a_0 \ a_1 \ a_2 \ a_3 \Rightarrow \text{TR}' a_0 \ a_1 \ a_2 \ a_3
\end{aligned}$$

[TR_trap_cmd_rule]

$$\begin{aligned}
& \vdash \forall \text{elementTest context stateInterp } x \text{ ins s outs}. \\
& \quad (\forall M \ Oi \ Os. \\
& \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s} \\
& \quad \quad \quad \text{outs}) \Rightarrow \\
& \quad \quad (M, Oi, Os) \text{ sat prop NONE}) \Rightarrow \\
& \quad \forall \text{NS Out } M \ Oi \ Os. \\
& \quad \text{TR } (M, Oi, Os) (\text{trap (inputList x)}) \\
& \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs}) \\
& \quad (\text{CFG elementTest stateInterp context ins} \\
& \quad \quad (\text{NS s (trap (inputList x))}) \\
& \quad \quad (\text{Out s (trap (inputList x))::outs})) \iff \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs}) \wedge \\
& \quad \quad (M, Oi, Os) \text{ sat prop NONE}
\end{aligned}$$

[TRrule0]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) (\text{exec (inputList x)}) \\
& \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs}) \\
& \quad (\text{CFG elementTest stateInterp context ins} \\
& \quad \quad (\text{NS s (exec (inputList x))}) \\
& \quad \quad (\text{Out s (exec (inputList x))::outs})) \iff \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs})
\end{aligned}$$

[TRrule1]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) (\text{trap (inputList x)}) \\
& \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs}) \\
& \quad (\text{CFG elementTest stateInterp context ins} \\
& \quad \quad (\text{NS s (trap (inputList x))}) \\
& \quad \quad (\text{Out s (trap (inputList x))::outs})) \iff \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::\text{ins}) \text{ s outs})
\end{aligned}$$

[trType_distinct_clauses]

$$\begin{aligned}
& \vdash (\forall a' \ a. \text{discard } a \neq \text{trap } a') \wedge (\forall a' \ a. \text{discard } a \neq \text{exec } a') \wedge \\
& \quad \forall a' \ a. \text{trap } a \neq \text{exec } a'
\end{aligned}$$

[trType_one_one]

$$\begin{aligned} \vdash (\forall a \ a'. (\text{discard } a = \text{discard } a') \iff (a = a')) \wedge \\ (\forall a \ a'. (\text{trap } a = \text{trap } a') \iff (a = a')) \wedge \\ \forall a \ a'. (\text{exec } a = \text{exec } a') \iff (a = a') \end{aligned}$$

4 satList Theory

Built: 10 June 2018

Parent Theories: aclDrules

4.1 Definitions

[satList_def]

$$\begin{aligned} \vdash \forall M \ Oi \ Os \ formList. \\ (M, Oi, Os) \text{ satList } formList \iff \\ \text{FOLDR } (\lambda x \ y. x \wedge y) \ \text{T} \ (\text{MAP } (\lambda f. (M, Oi, Os) \text{ sat } f) \ formList) \end{aligned}$$

4.2 Theorems

[satList_conj]

$$\begin{aligned} \vdash \forall l_1 \ l_2 \ M \ Oi \ Os. \\ (M, Oi, Os) \text{ satList } l_1 \wedge (M, Oi, Os) \text{ satList } l_2 \iff \\ (M, Oi, Os) \text{ satList } (l_1 ++ l_2) \end{aligned}$$

[satList_CONS]

$$\begin{aligned} \vdash \forall h \ t \ M \ Oi \ Os. \\ (M, Oi, Os) \text{ satList } (h :: t) \iff \\ (M, Oi, Os) \text{ sat } h \wedge (M, Oi, Os) \text{ satList } t \end{aligned}$$

[satList_nil]

$$\vdash (M, Oi, Os) \text{ satList } []$$

5 PBTypeIntegrated Theory

Built: 11 June 2018

Parent Theories: OMNITYPE

5.1 Datatypes

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

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

$$slCommand = PL \text{ PTypeIntegrated\$plCommand} \mid OMNI \text{ omniCommand}$$

$$slOutput = \text{PlanPB} \mid \text{MoveToORP} \mid \text{ConductORP} \mid \text{MoveToPB} \\ \mid \text{ConductPB} \mid \text{CompletePB} \mid \text{unAuthenticated} \\ \mid \text{unAuthorized}$$

$$slState = \text{PLAN_PB} \mid \text{MOVE_TO_ORP} \mid \text{CONDUCT_ORP} \mid \text{MOVE_TO_PB} \\ \mid \text{CONDUCT_PB} \mid \text{COMPLETE_PB}$$

$$stateRole = \text{PlatoonLeader} \mid \text{Omni}$$

5.2 Theorems

[omniCommand_distinct_clauses]

$$\vdash \text{ssmPlanPBComplete} \neq \text{ssmMoveToORPComplete} \wedge \\ \text{ssmPlanPBComplete} \neq \text{ssmConductORPComplete} \wedge \\ \text{ssmPlanPBComplete} \neq \text{ssmMoveToPBComplete} \wedge \\ \text{ssmPlanPBComplete} \neq \text{ssmConductPBComplete} \wedge \\ \text{ssmPlanPBComplete} \neq \text{invalidOmniCommand} \wedge \\ \text{ssmMoveToORPComplete} \neq \text{ssmConductORPComplete} \wedge \\ \text{ssmMoveToORPComplete} \neq \text{ssmMoveToPBComplete} \wedge \\ \text{ssmMoveToORPComplete} \neq \text{ssmConductPBComplete} \wedge \\ \text{ssmMoveToORPComplete} \neq \text{invalidOmniCommand} \wedge \\ \text{ssmConductORPComplete} \neq \text{ssmMoveToPBComplete} \wedge \\ \text{ssmConductORPComplete} \neq \text{ssmConductPBComplete} \wedge \\ \text{ssmConductORPComplete} \neq \text{invalidOmniCommand} \wedge \\ \text{ssmMoveToPBComplete} \neq \text{ssmConductPBComplete} \wedge \\ \text{ssmMoveToPBComplete} \neq \text{invalidOmniCommand} \wedge \\ \text{ssmConductPBComplete} \neq \text{invalidOmniCommand}$$

[plCommand_distinct_clauses]

$$\vdash \text{crossLD} \neq \text{conductORP} \wedge \text{crossLD} \neq \text{moveToPB} \wedge \\ \text{crossLD} \neq \text{conductPB} \wedge \text{crossLD} \neq \text{completePB} \wedge \\ \text{crossLD} \neq \text{incomplete} \wedge \text{conductORP} \neq \text{moveToPB} \wedge \\ \text{conductORP} \neq \text{conductPB} \wedge \text{conductORP} \neq \text{completePB} \wedge \\ \text{conductORP} \neq \text{incomplete} \wedge \text{moveToPB} \neq \text{conductPB} \wedge \\ \text{moveToPB} \neq \text{completePB} \wedge \text{moveToPB} \neq \text{incomplete} \wedge \\ \text{conductPB} \neq \text{completePB} \wedge \text{conductPB} \neq \text{incomplete} \wedge \\ \text{completePB} \neq \text{incomplete}$$

[slCommand_distinct_clauses]

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

[slCommand_one_one]

$$\vdash (\forall a \ a'. (\text{PL } a = \text{PL } a') \iff (a = a')) \wedge \\ \forall a \ a'. (\text{OMNI } a = \text{OMNI } a') \iff (a = a')$$

[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]

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

6 PBIntegratedDef Theory

Built: 11 June 2018

Parent Theories: PBTypeIntegrated, aclfoundation

6.1 Definitions

[secAuthorization_def]

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

$$\begin{aligned}
&\vdash (\forall xs. \\
&\quad \text{secContext PLAN_PB } xs = \\
&\quad \text{if } \text{getOmniCommand } xs = \text{ssmPlanPBComplete} \text{ then} \\
&\quad \quad [\text{prop } (\text{SOME } (\text{SLc } (\text{OMNI ssmPlanPBComplete}))) \text{ impf} \\
&\quad \quad \quad \text{Name PlatoonLeader controls} \\
&\quad \quad \text{prop } (\text{SOME } (\text{SLc } (\text{PL crossLD})))])
\end{aligned}$$

```

    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)))]

```

6.2 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) ∧
  (∀ xs v3. getOmniCommand (notf v3::xs) = getOmniCommand xs) ∧
  (∀ xs v5 v4.

```



```

    getOmniCommand (v4 andf v5::xs) = getOmniCommand xs) ∧
  (∀ xs v7 v6.
    getOmniCommand (v6 orf v7::xs) = getOmniCommand xs) ∧
  (∀ xs v9 v8.
    getOmniCommand (v8 impf v9::xs) = getOmniCommand xs) ∧
  (∀ xs v11 v10.
    getOmniCommand (v10 eqf v11::xs) = getOmniCommand xs) ∧
  (∀ xs v12.
    getOmniCommand (v12 says TT::xs) = getOmniCommand xs) ∧
  (∀ xs v12.
    getOmniCommand (v12 says FF::xs) = getOmniCommand xs) ∧
  (∀ xs v134.
    getOmniCommand (Name v134 says prop NONE::xs) =
    getOmniCommand xs) ∧
  (∀ xs v144.
    getOmniCommand
      (Name PlatoonLeader says prop (SOME v144)::xs) =
    getOmniCommand xs) ∧
  (∀ xs v146.
    getOmniCommand
      (Name Omni says prop (SOME (ESCc v146))::xs) =
    getOmniCommand xs) ∧
  (∀ xs v150.
    getOmniCommand
      (Name Omni says prop (SOME (SLc (PL v150)))::xs) =
    getOmniCommand xs) ∧
  (∀ xs v68 v136 v135.
    getOmniCommand (v135 meet v136 says prop v68::xs) =
    getOmniCommand xs) ∧
  (∀ xs v68 v138 v137.
    getOmniCommand (v137 quoting v138 says prop v68::xs) =
    getOmniCommand xs) ∧
  (∀ xs v69 v12.
    getOmniCommand (v12 says notf v69::xs) =
    getOmniCommand xs) ∧
  (∀ xs v71 v70 v12.
    getOmniCommand (v12 says (v70 andf v71)::xs) =
    getOmniCommand xs) ∧
  (∀ xs v73 v72 v12.
    getOmniCommand (v12 says (v72 orf v73)::xs) =
    getOmniCommand xs) ∧
  (∀ xs v75 v74 v12.
    getOmniCommand (v12 says (v74 impf v75)::xs) =
    getOmniCommand xs) ∧
  (∀ xs v77 v76 v12.
    getOmniCommand (v12 says (v76 eqf v77)::xs) =
    getOmniCommand xs) ∧
  (∀ xs v79 v78 v12.
    getOmniCommand (v12 says v78 says v79::xs) =

```

```

    getOmniCommand xs) ∧
(∀ xs v81 v80 v12.
  getOmniCommand (v12 says v80 speaks_for v81::xs) =
  getOmniCommand xs) ∧
(∀ xs v83 v82 v12.
  getOmniCommand (v12 says v82 controls v83::xs) =
  getOmniCommand xs) ∧
(∀ xs v86 v85 v84 v12.
  getOmniCommand (v12 says reps v84 v85 v86::xs) =
  getOmniCommand xs) ∧
(∀ xs v88 v87 v12.
  getOmniCommand (v12 says v87 domi v88::xs) =
  getOmniCommand xs) ∧
(∀ xs v90 v89 v12.
  getOmniCommand (v12 says v89 eqi v90::xs) =
  getOmniCommand xs) ∧
(∀ xs v92 v91 v12.
  getOmniCommand (v12 says v91 doms v92::xs) =
  getOmniCommand xs) ∧
(∀ xs v94 v93 v12.
  getOmniCommand (v12 says v93 eqs v94::xs) =
  getOmniCommand xs) ∧
(∀ xs v96 v95 v12.
  getOmniCommand (v12 says v95 eqn v96::xs) =
  getOmniCommand xs) ∧
(∀ xs v98 v97 v12.
  getOmniCommand (v12 says v97 lte v98::xs) =
  getOmniCommand xs) ∧
(∀ xs v99 v12 v100.
  getOmniCommand (v12 says v99 lt v100::xs) =
  getOmniCommand xs) ∧
(∀ xs v15 v14.
  getOmniCommand (v14 speaks_for v15::xs) =
  getOmniCommand xs) ∧
(∀ xs v17 v16.
  getOmniCommand (v16 controls v17::xs) =
  getOmniCommand xs) ∧
(∀ xs v20 v19 v18.
  getOmniCommand (reps v18 v19 v20::xs) =
  getOmniCommand xs) ∧
(∀ xs v22 v21.
  getOmniCommand (v21 domi v22::xs) = getOmniCommand xs) ∧
(∀ xs v24 v23.
  getOmniCommand (v23 eqi v24::xs) = getOmniCommand xs) ∧
(∀ xs v26 v25.
  getOmniCommand (v25 doms v26::xs) = getOmniCommand xs) ∧
(∀ xs v28 v27.
  getOmniCommand (v27 eqs v28::xs) = getOmniCommand xs) ∧
(∀ xs v30 v29.

```

$\text{getOmniCommand } (v_{29} \text{ eqn } v_{30}::xs) = \text{getOmniCommand } xs) \wedge$
 $(\forall xs \ v_{32} \ v_{31}.$
 $\text{getOmniCommand } (v_{31} \text{ lte } v_{32}::xs) = \text{getOmniCommand } xs) \wedge$
 $\forall xs \ v_{34} \ v_{33}.$
 $\text{getOmniCommand } (v_{33} \text{ lt } v_{34}::xs) = \text{getOmniCommand } xs$

[getOmniCommand_ind]

$\vdash \forall P.$
 $P \ [] \wedge$
 $(\forall cmd \ xs.$
 $P \ (\text{Name Omni says prop (SOME (SLc (OMNI cmd)))::xs})) \wedge$
 $(\forall xs. P \ xs \Rightarrow P \ (\text{TT::xs})) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{FF::xs})) \wedge$
 $(\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2::xs)) \wedge$
 $(\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3::xs)) \wedge$
 $(\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \text{ andf } v_5::xs)) \wedge$
 $(\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \text{ orf } v_7::xs)) \wedge$
 $(\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \text{ impf } v_9::xs)) \wedge$
 $(\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \text{ eqf } v_{11}::xs)) \wedge$
 $(\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says TT::xs})) \wedge$
 $(\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says FF::xs})) \wedge$
 $(\forall v_{134} \ xs. P \ xs \Rightarrow P \ (\text{Name } v_{134} \text{ says prop NONE::xs})) \wedge$
 $(\forall v_{144} \ xs.$
 $P \ xs \Rightarrow$
 $P \ (\text{Name PlatoonLeader says prop (SOME } v_{144}::xs)) \wedge$
 $(\forall v_{146} \ xs.$
 $P \ xs \Rightarrow P \ (\text{Name Omni says prop (SOME (ESCc } v_{146}::xs)) \wedge$
 $(\forall v_{150} \ xs.$
 $P \ xs \Rightarrow$
 $P \ (\text{Name Omni says prop (SOME (SLc (PL } v_{150})))::xs)) \wedge$
 $(\forall v_{135} \ v_{136} \ v_{68} \ xs.$
 $P \ xs \Rightarrow P \ (v_{135} \text{ meet } v_{136} \text{ says prop } v_{68}::xs)) \wedge$
 $(\forall v_{137} \ v_{138} \ v_{68} \ xs.$
 $P \ xs \Rightarrow P \ (v_{137} \text{ quoting } v_{138} \text{ says prop } v_{68}::xs)) \wedge$
 $(\forall v_{12} \ v_{69} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says notf } v_{69}::xs)) \wedge$
 $(\forall v_{12} \ v_{70} \ v_{71} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } (v_{70} \text{ andf } v_{71})::xs)) \wedge$
 $(\forall v_{12} \ v_{72} \ v_{73} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } (v_{72} \text{ orf } v_{73})::xs)) \wedge$
 $(\forall v_{12} \ v_{74} \ v_{75} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } (v_{74} \text{ impf } v_{75})::xs)) \wedge$
 $(\forall v_{12} \ v_{76} \ v_{77} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } (v_{76} \text{ eqf } v_{77})::xs)) \wedge$
 $(\forall v_{12} \ v_{78} \ v_{79} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{78} \text{ says } v_{79}::xs)) \wedge$
 $(\forall v_{12} \ v_{80} \ v_{81} \ xs.$
 $P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{80} \text{ speaks_for } v_{81}::xs)) \wedge$
 $(\forall v_{12} \ v_{82} \ v_{83} \ xs.$
 $P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{82} \text{ controls } v_{83}::xs)) \wedge$
 $(\forall v_{12} \ v_{84} \ v_{85} \ v_{86} \ xs.$
 $P \ xs \Rightarrow P \ (v_{12} \text{ says reps } v_{84} \ v_{85} \ v_{86}::xs)) \wedge$
 $(\forall v_{12} \ v_{87} \ v_{88} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{87} \text{ domi } v_{88}::xs)) \wedge$
 $(\forall v_{12} \ v_{89} \ v_{90} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{89} \text{ eqi } v_{90}::xs)) \wedge$
 $(\forall v_{12} \ v_{91} \ v_{92} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{91} \text{ doms } v_{92}::xs)) \wedge$
 $(\forall v_{12} \ v_{93} \ v_{94} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says } v_{93} \text{ eqs } v_{94}::xs)) \wedge$

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

7 ssmPBIntegrated Theory

Built: 11 June 2018

Parent Theories: PBIntegratedDef, ssm

7.1 Theorems

[inputOK_def]

$$\begin{aligned}
& \vdash (\text{inputOK } (\text{Name PlatoonLeader says prop } cmd) \iff T) \wedge \\
& (\text{inputOK } (\text{Name Omni says prop } cmd) \iff T) \wedge \\
& (\text{inputOK } TT \iff F) \wedge (\text{inputOK } FF \iff F) \wedge \\
& (\text{inputOK } (\text{prop } v) \iff F) \wedge (\text{inputOK } (\text{notf } v_1) \iff F) \wedge \\
& (\text{inputOK } (v_2 \text{ andf } v_3) \iff F) \wedge (\text{inputOK } (v_4 \text{ orf } v_5) \iff F) \wedge \\
& (\text{inputOK } (v_6 \text{ impf } v_7) \iff F) \wedge (\text{inputOK } (v_8 \text{ eqf } v_9) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } TT) \iff F) \wedge (\text{inputOK } (v_{10} \text{ says } FF) \iff F) \wedge \\
& (\text{inputOK } (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66}) \iff F) \wedge \\
& (\text{inputOK } (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says notf } v_{67}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } (v_{68} \text{ andf } v_{69})) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } (v_{70} \text{ orf } v_{71})) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } (v_{72} \text{ impf } v_{73})) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75})) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{76} \text{ says } v_{77}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{80} \text{ controls } v_{81}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says reps } v_{82} v_{83} v_{84}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{85} \text{ domi } v_{86}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{87} \text{ eqi } v_{88}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{89} \text{ doms } v_{90}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff F) \wedge \\
& (\text{inputOK } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge
\end{aligned}$$

$(\text{inputOK } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge$
 $(\text{inputOK } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge$
 $(\text{inputOK } (v_{14} \text{ controls } v_{15}) \iff F) \wedge$
 $(\text{inputOK } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge$
 $(\text{inputOK } (v_{19} \text{ domi } v_{20}) \iff F) \wedge$
 $(\text{inputOK } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge$
 $(\text{inputOK } (v_{23} \text{ doms } v_{24}) \iff F) \wedge$
 $(\text{inputOK } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge (\text{inputOK } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge$
 $(\text{inputOK } (v_{29} \text{ lte } v_{30}) \iff F) \wedge (\text{inputOK } (v_{31} \text{ lt } v_{32}) \iff F)$

[inputOK_ind]

$\vdash \forall P.$

$(\forall \text{cmd}. P (\text{Name PlatoonLeader says prop cmd})) \wedge$
 $(\forall \text{cmd}. P (\text{Name Omni says prop cmd})) \wedge P \text{ TT} \wedge P \text{ FF} \wedge$
 $(\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge$
 $(\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge$
 $(\forall v_6 \ v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 \ v_9. P (v_8 \text{ eqf } v_9)) \wedge$
 $(\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge$
 $(\forall v_{133} \ v_{134} \ v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{135} \ v_{136} \ v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{10} \ v_{67}. P (v_{10} \text{ says notf } v_{67})) \wedge$
 $(\forall v_{10} \ v_{68} \ v_{69}. P (v_{10} \text{ says } (v_{68} \text{ andf } v_{69}))) \wedge$
 $(\forall v_{10} \ v_{70} \ v_{71}. P (v_{10} \text{ says } (v_{70} \text{ orf } v_{71}))) \wedge$
 $(\forall v_{10} \ v_{72} \ v_{73}. P (v_{10} \text{ says } (v_{72} \text{ impf } v_{73}))) \wedge$
 $(\forall v_{10} \ v_{74} \ v_{75}. P (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75}))) \wedge$
 $(\forall v_{10} \ v_{76} \ v_{77}. P (v_{10} \text{ says } v_{76} \text{ says } v_{77})) \wedge$
 $(\forall v_{10} \ v_{78} \ v_{79}. P (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79})) \wedge$
 $(\forall v_{10} \ v_{80} \ v_{81}. P (v_{10} \text{ says } v_{80} \text{ controls } v_{81})) \wedge$
 $(\forall v_{10} \ v_{82} \ v_{83} \ v_{84}. P (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84})) \wedge$
 $(\forall v_{10} \ v_{85} \ v_{86}. P (v_{10} \text{ says } v_{85} \text{ domi } v_{86})) \wedge$
 $(\forall v_{10} \ v_{87} \ v_{88}. P (v_{10} \text{ says } v_{87} \text{ eqi } v_{88})) \wedge$
 $(\forall v_{10} \ v_{89} \ v_{90}. P (v_{10} \text{ says } v_{89} \text{ doms } v_{90})) \wedge$
 $(\forall v_{10} \ v_{91} \ v_{92}. P (v_{10} \text{ says } v_{91} \text{ eqs } v_{92})) \wedge$
 $(\forall v_{10} \ v_{93} \ v_{94}. P (v_{10} \text{ says } v_{93} \text{ eqn } v_{94})) \wedge$
 $(\forall v_{10} \ v_{95} \ v_{96}. P (v_{10} \text{ says } v_{95} \text{ lte } v_{96})) \wedge$
 $(\forall v_{10} \ v_{97} \ v_{98}. P (v_{10} \text{ says } v_{97} \text{ lt } v_{98})) \wedge$
 $(\forall v_{12} \ v_{13}. P (v_{12} \text{ speaks_for } v_{13})) \wedge$
 $(\forall v_{14} \ v_{15}. P (v_{14} \text{ controls } v_{15})) \wedge$
 $(\forall v_{16} \ v_{17} \ v_{18}. P (\text{reps } v_{16} \ v_{17} \ v_{18})) \wedge$
 $(\forall v_{19} \ v_{20}. P (v_{19} \text{ domi } v_{20})) \wedge$
 $(\forall v_{21} \ v_{22}. P (v_{21} \text{ eqi } v_{22})) \wedge$
 $(\forall v_{23} \ v_{24}. P (v_{23} \text{ doms } v_{24})) \wedge$
 $(\forall v_{25} \ v_{26}. P (v_{25} \text{ eqs } v_{26})) \wedge (\forall v_{27} \ v_{28}. P (v_{27} \text{ eqn } v_{28})) \wedge$
 $(\forall v_{29} \ v_{30}. P (v_{29} \text{ lte } v_{30})) \wedge (\forall v_{31} \ v_{32}. P (v_{31} \text{ lt } v_{32})) \Rightarrow$
 $\forall v. P \ v$

[PBNS_def]

$\vdash (\text{PBNS PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))]) =$
 $\text{MOVE_TO_ORP}) \wedge$

(PBNS MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) =
 CONDUCT_ORP) \wedge
 (PBNS CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) =
 MOVE_TO_PB) \wedge
 (PBNS MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) =
 CONDUCT_PB) \wedge
 (PBNS CONDUCT_PB (exec [SOME (SLc (PL completePB))]) =
 COMPLETE_PB) \wedge (PBNS s (trap v_0) = s) \wedge
 (PBNS s (discard v_1) = s)

[PBNS_ind]

$\vdash \forall P.$

P PLAN_PB (exec [SOME (SLc (PL crossLD))]) \wedge
 P MOVE_TO_ORP (exec [SOME (SLc (PL conductORP))]) \wedge
 P CONDUCT_ORP (exec [SOME (SLc (PL moveToPB))]) \wedge
 P MOVE_TO_PB (exec [SOME (SLc (PL conductPB))]) \wedge
 P CONDUCT_PB (exec [SOME (SLc (PL completePB))]) \wedge
 $(\forall s \ v_0. P \ s \ (\text{trap } v_0)) \wedge (\forall s \ v_1. P \ s \ (\text{discard } v_1)) \wedge$
 $(\forall v_8. P \ v_8 \ (\text{exec } [])) \wedge$
 $(\forall v_{11} \ v_{10}. P \ v_{11} \ (\text{exec } (\text{NONE}::v_{10}))) \wedge$
 $(\forall v_{16} \ v_{13} \ v_{15}. P \ v_{16} \ (\text{exec } (\text{SOME } (\text{ESCc } v_{13})::v_{15}))) \wedge$
 P MOVE_TO_ORP (exec [SOME (SLc (PL crossLD))]) \wedge
 P CONDUCT_ORP (exec [SOME (SLc (PL crossLD))]) \wedge
 P MOVE_TO_PB (exec [SOME (SLc (PL crossLD))]) \wedge
 P CONDUCT_PB (exec [SOME (SLc (PL crossLD))]) \wedge
 P COMPLETE_PB (exec [SOME (SLc (PL crossLD))]) \wedge
 P PLAN_PB (exec [SOME (SLc (PL conductORP))]) \wedge
 P CONDUCT_ORP (exec [SOME (SLc (PL conductORP))]) \wedge
 P MOVE_TO_PB (exec [SOME (SLc (PL conductORP))]) \wedge
 P CONDUCT_PB (exec [SOME (SLc (PL conductORP))]) \wedge
 P COMPLETE_PB (exec [SOME (SLc (PL conductORP))]) \wedge
 P PLAN_PB (exec [SOME (SLc (PL moveToPB))]) \wedge
 P MOVE_TO_ORP (exec [SOME (SLc (PL moveToPB))]) \wedge
 P MOVE_TO_PB (exec [SOME (SLc (PL moveToPB))]) \wedge
 P CONDUCT_PB (exec [SOME (SLc (PL moveToPB))]) \wedge
 P COMPLETE_PB (exec [SOME (SLc (PL moveToPB))]) \wedge
 P PLAN_PB (exec [SOME (SLc (PL conductPB))]) \wedge
 P MOVE_TO_ORP (exec [SOME (SLc (PL conductPB))]) \wedge
 P CONDUCT_ORP (exec [SOME (SLc (PL conductPB))]) \wedge
 P CONDUCT_PB (exec [SOME (SLc (PL conductPB))]) \wedge
 P COMPLETE_PB (exec [SOME (SLc (PL conductPB))]) \wedge
 P PLAN_PB (exec [SOME (SLc (PL completePB))]) \wedge
 P MOVE_TO_ORP (exec [SOME (SLc (PL completePB))]) \wedge
 P CONDUCT_ORP (exec [SOME (SLc (PL completePB))]) \wedge
 P MOVE_TO_PB (exec [SOME (SLc (PL completePB))]) \wedge
 P COMPLETE_PB (exec [SOME (SLc (PL completePB))]) \wedge
 $(\forall v_{24}. P \ v_{24} \ (\text{exec } [\text{SOME } (\text{SLc } (\text{PL incomplete}))])) \wedge$
 $(\forall v_{26} \ v_{25} \ v_{22} \ v_{23}.$
 $P \ v_{26} \ (\text{exec } (\text{SOME } (\text{SLc } (\text{PL } v_{25}))::v_{22}::v_{23}))) \wedge$

$$(\forall v_{28} v_{19} v_{27}. P v_{28} (\text{exec } (\text{SOME } (\text{SLc } (\text{OMNI } v_{19}))) :: v_{27})) \Rightarrow \\ \forall v v_1. P v v_1$$

[PBOut_def]

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

[PBOut_ind]

$$\begin{aligned} &\vdash \forall P. \\ &\quad P \text{ PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ MOVE_TO_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ CONDUCT_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ MOVE_TO_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ CONDUCT_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL completePB}))])) \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_8. P v_8 \text{ (exec } [])) \wedge \\ &\quad (\forall v_{11} v_{10}. P v_{11} \text{ (exec } (\text{NONE} :: v_{10}))) \wedge \\ &\quad (\forall v_{16} v_{13} v_{15}. P v_{16} \text{ (exec } (\text{SOME } (\text{ESCc } v_{13}) :: v_{15}))) \wedge \\ &\quad P \text{ MOVE_TO_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ CONDUCT_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ MOVE_TO_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ CONDUCT_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ COMPLETE_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL crossLD}))])) \wedge \\ &\quad P \text{ PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ CONDUCT_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ MOVE_TO_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ CONDUCT_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ COMPLETE_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductORP}))])) \wedge \\ &\quad P \text{ PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ MOVE_TO_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ MOVE_TO_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ CONDUCT_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ COMPLETE_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL moveToPB}))])) \wedge \\ &\quad P \text{ PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ MOVE_TO_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ CONDUCT_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ CONDUCT_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ COMPLETE_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL conductPB}))])) \wedge \\ &\quad P \text{ PLAN_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL completePB}))])) \wedge \\ &\quad P \text{ MOVE_TO_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL completePB}))])) \wedge \end{aligned}$$

$$\begin{aligned}
& P \text{ CONDUCT_ORP } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL } \text{completePB}))]) \wedge \\
& P \text{ MOVE_TO_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL } \text{completePB}))]) \wedge \\
& P \text{ COMPLETE_PB } (\text{exec } [\text{SOME } (\text{SLc } (\text{PL } \text{completePB}))]) \wedge \\
& (\forall v_{24}. P \ v_{24} \ (\text{exec } [\text{SOME } (\text{SLc } (\text{PL } \text{incomplete}))])) \wedge \\
& (\forall v_{26} \ v_{25} \ v_{22} \ v_{23}. \\
& \quad P \ v_{26} \ (\text{exec } (\text{SOME } (\text{SLc } (\text{PL } v_{25})):v_{22}:v_{23}))) \wedge \\
& (\forall v_{28} \ v_{19} \ v_{27}. P \ v_{28} \ (\text{exec } (\text{SOME } (\text{SLc } (\text{OMNI } v_{19})):v_{27}))) \Rightarrow \\
& \forall v \ v_1. P \ v \ v_1
\end{aligned}$$

[PlatoonLeader_Omni_notDiscard_slCommand_thm]

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

[PlatoonLeader_PLAN_PB_exec_justified_thm]

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


```

      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]

```

 $\vdash \text{omniCommand} \neq \text{ssmPlanPBComplete} \Rightarrow$ 
  (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]

```

 $\vdash$  omniCommand  $\neq$  ssmPlanPBComplete  $\Rightarrow$ 
  (s = PLAN_PB)  $\Rightarrow$ 
 $\forall NS$  Out M 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) ∧ (M, Oi, Os) sat prop NONE

```

[PlatoonLeader_PLAN_PB_trap_lemma]

```

⊢ omniCommand ≠ ssmPlanPBComplete ⇒
  (s = PLAN_PB) ⇒
  ∀ 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) ⇒
    (M, Oi, Os) sat prop NONE

```

8 ssmConductORP Theory

Built: 10 June 2018

Parent Theories: ConductORPType, ssm11, OMNIType

8.1 Definitions

[secContextConductORP_def]

```

⊢ ∀ plcnd psgcmd incomplete.
  secContextConductORP plcnd psgcmd incomplete =
  [Name PlatoonLeader controls prop (SOME (SLc (PL plcnd))));
   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 plcnd))) impf prop NONE]

```

[ssmConductORPStateInterp_def]

```

⊢ ∀ slState. ssmConductORPStateInterp slState = TT

```

8.2 Theorems

[authTestConductORP_cmd_reject_lemma]

```

⊢ ∀ cmd. ¬authTestConductORP (prop (SOME cmd))

```

[authTestConductORP_def]

$$\begin{aligned} \vdash & (\text{authTestConductORP } (\text{Name PlatoonLeader says prop } cmd) \iff \\ & T) \wedge \\ & (\text{authTestConductORP } (\text{Name PlatoonSergeant says prop } cmd) \iff \\ & T) \wedge (\text{authTestConductORP } TT \iff F) \wedge \\ & (\text{authTestConductORP } FF \iff F) \wedge \\ & (\text{authTestConductORP } (\text{prop } v) \iff F) \wedge \\ & (\text{authTestConductORP } (\text{notf } v_1) \iff F) \wedge \\ & (\text{authTestConductORP } (v_2 \text{ andf } v_3) \iff F) \wedge \\ & (\text{authTestConductORP } (v_4 \text{ orf } v_5) \iff F) \wedge \\ & (\text{authTestConductORP } (v_6 \text{ impf } v_7) \iff F) \wedge \\ & (\text{authTestConductORP } (v_8 \text{ eqf } v_9) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } TT) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } FF) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says notf } v_{67}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } (v_{68} \text{ andf } v_{69})) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } (v_{70} \text{ orf } v_{71})) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } (v_{72} \text{ impf } v_{73})) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75})) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{76} \text{ says } v_{77}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{80} \text{ controls } v_{81}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{85} \text{ domi } v_{86}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{87} \text{ eqi } v_{88}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{89} \text{ doms } v_{90}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{14} \text{ controls } v_{15}) \iff F) \wedge \\ & (\text{authTestConductORP } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{19} \text{ domi } v_{20}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{23} \text{ doms } v_{24}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{29} \text{ lte } v_{30}) \iff F) \wedge \\ & (\text{authTestConductORP } (v_{31} \text{ lt } v_{32}) \iff F) \end{aligned}$$

[authTestConductORP_ind]

$$\begin{aligned} \vdash & \forall P. \\ & (\forall cmd. P (\text{Name PlatoonLeader says prop } cmd)) \wedge \\ & (\forall cmd. P (\text{Name PlatoonSergeant says prop } cmd)) \wedge P \ TT \wedge \\ & P \ FF \wedge (\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge \\ & (\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge \end{aligned}$$

$$\begin{aligned}
& (\forall v_6 v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 v_9. P (v_8 \text{ eqf } v_9)) \wedge \\
& (\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge \\
& (\forall v_{133} v_{134} v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge \\
& (\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}$$

[conductORPNS_def]

$$\begin{aligned}
& \vdash (\text{conductORPNS CONDUCT_ORP (exec (PL secure))} = \text{SECURE}) \wedge \\
& (\text{conductORPNS CONDUCT_ORP (exec (PL plIncomplete))} = \\
& \quad \text{CONDUCT_ORP}) \wedge \\
& (\text{conductORPNS SECURE (exec (PSG actionsIn))} = \text{ACTIONS_IN}) \wedge \\
& (\text{conductORPNS SECURE (exec (PSG psgIncomplete))} = \text{SECURE}) \wedge \\
& (\text{conductORPNS ACTIONS_IN (exec (PL withdraw))} = \text{WITHDRAW}) \wedge \\
& (\text{conductORPNS ACTIONS_IN (exec (PL plIncomplete))} = \\
& \quad \text{ACTIONS_IN}) \wedge \\
& (\text{conductORPNS WITHDRAW (exec (PL complete))} = \text{COMPLETE}) \wedge \\
& (\text{conductORPNS WITHDRAW (exec (PL plIncomplete))} = \text{WITHDRAW}) \wedge \\
& (\text{conductORPNS } s \text{ (trap (PL cmd'))} = s) \wedge \\
& (\text{conductORPNS } s \text{ (trap (PSG cmd))} = s) \wedge \\
& (\text{conductORPNS } s \text{ (discard (PL cmd'))} = s) \wedge \\
& (\text{conductORPNS } s \text{ (discard (PSG cmd))} = s)
\end{aligned}$$

[conductORPNS_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& \quad P \text{ CONDUCT_ORP (exec (PL secure))} \wedge
\end{aligned}$$

P CONDUCT_ORP (exec (PL plIncomplete)) \wedge
 P SECURE (exec (PSG actionsIn)) \wedge
 P SECURE (exec (PSG psgIncomplete)) \wedge
 P ACTIONS_IN (exec (PL withdraw)) \wedge
 P ACTIONS_IN (exec (PL plIncomplete)) \wedge
 P WITHDRAW (exec (PL complete)) \wedge
 P WITHDRAW (exec (PL plIncomplete)) \wedge
 $(\forall s \text{ cmd}. P \ s \ (\text{trap} \ (\text{PL} \ \text{cmd}))) \wedge$
 $(\forall s \text{ cmd}. P \ s \ (\text{trap} \ (\text{PSG} \ \text{cmd}))) \wedge$
 $(\forall s \text{ cmd}. P \ s \ (\text{discard} \ (\text{PL} \ \text{cmd}))) \wedge$
 $(\forall s \text{ cmd}. P \ s \ (\text{discard} \ (\text{PSG} \ \text{cmd}))) \wedge$
 P CONDUCT_ORP (exec (PL withdraw)) \wedge
 P CONDUCT_ORP (exec (PL complete)) \wedge
 $(\forall v_{11}. P \ \text{CONDUCT_ORP} \ (\text{exec} \ (\text{PSG} \ v_{11}))) \wedge$
 $(\forall v_{13}. P \ \text{SECURE} \ (\text{exec} \ (\text{PL} \ v_{13}))) \wedge$
 P ACTIONS_IN (exec (PL secure)) \wedge
 P ACTIONS_IN (exec (PL complete)) \wedge
 $(\forall v_{17}. P \ \text{ACTIONS_IN} \ (\text{exec} \ (\text{PSG} \ v_{17}))) \wedge$
 P WITHDRAW (exec (PL secure)) \wedge
 P WITHDRAW (exec (PL withdraw)) \wedge
 $(\forall v_{20}. P \ \text{WITHDRAW} \ (\text{exec} \ (\text{PSG} \ v_{20}))) \wedge$
 $(\forall v_{21}. P \ \text{COMPLETE} \ (\text{exec} \ v_{21})) \Rightarrow$
 $\forall v \ v_1. P \ v \ v_1$

[conductORPOut_def]

$\vdash (\text{conductORPOut} \ \text{CONDUCT_ORP} \ (\text{exec} \ (\text{PL} \ \text{secure})) = \text{Secure}) \wedge$
 $(\text{conductORPOut} \ \text{CONDUCT_ORP} \ (\text{exec} \ (\text{PL} \ \text{plIncomplete})) =$
 $\text{ConductORP}) \wedge$
 $(\text{conductORPOut} \ \text{SECURE} \ (\text{exec} \ (\text{PSG} \ \text{actionsIn})) = \text{ActionsIn}) \wedge$
 $(\text{conductORPOut} \ \text{SECURE} \ (\text{exec} \ (\text{PSG} \ \text{psgIncomplete})) = \text{Secure}) \wedge$
 $(\text{conductORPOut} \ \text{ACTIONS_IN} \ (\text{exec} \ (\text{PL} \ \text{withdraw})) = \text{Withdraw}) \wedge$
 $(\text{conductORPOut} \ \text{ACTIONS_IN} \ (\text{exec} \ (\text{PL} \ \text{plIncomplete})) =$
 $\text{ActionsIn}) \wedge$
 $(\text{conductORPOut} \ \text{WITHDRAW} \ (\text{exec} \ (\text{PL} \ \text{complete})) = \text{Complete}) \wedge$
 $(\text{conductORPOut} \ \text{WITHDRAW} \ (\text{exec} \ (\text{PL} \ \text{plIncomplete})) =$
 $\text{Withdraw}) \wedge$
 $(\text{conductORPOut} \ s \ (\text{trap} \ (\text{PL} \ \text{cmd}')) = \text{unAuthorized}) \wedge$
 $(\text{conductORPOut} \ s \ (\text{trap} \ (\text{PSG} \ \text{cmd}')) = \text{unAuthorized}) \wedge$
 $(\text{conductORPOut} \ s \ (\text{discard} \ (\text{PL} \ \text{cmd}')) = \text{unAuthenticated}) \wedge$
 $(\text{conductORPOut} \ s \ (\text{discard} \ (\text{PSG} \ \text{cmd}')) = \text{unAuthenticated})$

[conductORPOut_ind]

$\vdash \forall P.$
 $P \ \text{CONDUCT_ORP} \ (\text{exec} \ (\text{PL} \ \text{secure})) \wedge$
 $P \ \text{CONDUCT_ORP} \ (\text{exec} \ (\text{PL} \ \text{plIncomplete})) \wedge$
 $P \ \text{SECURE} \ (\text{exec} \ (\text{PSG} \ \text{actionsIn})) \wedge$
 $P \ \text{SECURE} \ (\text{exec} \ (\text{PSG} \ \text{psgIncomplete})) \wedge$
 $P \ \text{ACTIONS_IN} \ (\text{exec} \ (\text{PL} \ \text{withdraw})) \wedge$
 $P \ \text{ACTIONS_IN} \ (\text{exec} \ (\text{PL} \ \text{plIncomplete})) \wedge$

$$\begin{aligned}
& P \text{ WITHDRAW } (\text{exec } (\text{PL complete})) \wedge \\
& P \text{ WITHDRAW } (\text{exec } (\text{PL plIncomplete})) \wedge \\
& (\forall s \text{ cmd. } P \ s \ (\text{trap } (\text{PL cmd}))) \wedge \\
& (\forall s \text{ cmd. } P \ s \ (\text{trap } (\text{PSG cmd}))) \wedge \\
& (\forall s \text{ cmd. } P \ s \ (\text{discard } (\text{PL cmd}))) \wedge \\
& (\forall s \text{ cmd. } P \ s \ (\text{discard } (\text{PSG cmd}))) \wedge \\
& P \text{ CONDUCT_ORP } (\text{exec } (\text{PL withdraw})) \wedge \\
& P \text{ CONDUCT_ORP } (\text{exec } (\text{PL complete})) \wedge \\
& (\forall v_{11}. P \text{ CONDUCT_ORP } (\text{exec } (\text{PSG } v_{11}))) \wedge \\
& (\forall v_{13}. P \text{ SECURE } (\text{exec } (\text{PL } v_{13}))) \wedge \\
& P \text{ ACTIONS_IN } (\text{exec } (\text{PL secure})) \wedge \\
& P \text{ ACTIONS_IN } (\text{exec } (\text{PL complete})) \wedge \\
& (\forall v_{17}. P \text{ ACTIONS_IN } (\text{exec } (\text{PSG } v_{17}))) \wedge \\
& P \text{ WITHDRAW } (\text{exec } (\text{PL secure})) \wedge \\
& P \text{ WITHDRAW } (\text{exec } (\text{PL withdraw})) \wedge \\
& (\forall v_{20}. P \text{ WITHDRAW } (\text{exec } (\text{PSG } v_{20}))) \wedge \\
& (\forall v_{21}. P \text{ COMPLETE } (\text{exec } v_{21})) \Rightarrow \\
& \forall v \ v_1. P \ v \ v_1
\end{aligned}$$

[PlatoonLeader_exec_plCommand_justified_thm]

$$\begin{aligned}
& \vdash \forall NS \text{ Out } M \ Oi \ Os. \\
& \text{TR } (M, Oi, Os) \ (\text{exec } (\text{SLc } (\text{PL } plCommand))) \\
& \quad (\text{CFG authTestConductORP ssmConductORPStateInterp} \\
& \quad \quad (\text{secContextConductORP } plCommand \ psgCommand \ incomplete) \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))::ins) \ s \ outs) \\
& \quad \quad (\text{CFG authTestConductORP ssmConductORPStateInterp} \\
& \quad \quad \quad (\text{secContextConductORP } plCommand \ psgCommand \ incomplete) \\
& \quad \quad \quad ins \ (NS \ s \ (\text{exec } (\text{SLc } (\text{PL } plCommand)))) \\
& \quad \quad \quad (\text{Out } s \ (\text{exec } (\text{SLc } (\text{PL } plCommand)))::outs)) \iff \\
& \text{authTestConductORP} \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))) \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authTestConductORP ssmConductORPStateInterp} \\
& \quad \quad (\text{secContextConductORP } plCommand \ psgCommand \ incomplete) \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))::ins) \ s \ outs) \wedge \\
& \quad (M, Oi, Os) \text{ sat } \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))
\end{aligned}$$

[PlatoonLeader_plCommand_lemma]

$$\begin{aligned}
& \vdash \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG authTestConductORP ssmConductORPStateInterp} \\
& \quad \quad (\text{secContextConductORP } plCommand \ psgCommand \ incomplete) \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))::ins) \ s \ outs) \Rightarrow \\
& \quad (M, Oi, Os) \text{ sat } \text{prop } (\text{SOME } (\text{SLc } (\text{PL } plCommand)))
\end{aligned}$$

[PlatoonSergeant_exec_psgCommand_justified_thm]

```

⊢ ∀ NS Out M Oi Os.
  TR (M, Oi, Os) (exec (SLc (PSG psgCommand)))
    (CFG authTestConductORP ssmConductORPStateInterp
      (secContextConductORP plCommand psgCommand incomplete)
      (Name PlatoonSergeant says
        prop (SOME (SLc (PSG psgCommand)))::ins) s outs)
    (CFG authTestConductORP ssmConductORPStateInterp
      (secContextConductORP plCommand psgCommand incomplete)
      ins (NS s (exec (SLc (PSG psgCommand))))
      (Out s (exec (SLc (PSG psgCommand)))::outs)) ⇔⇒
  authTestConductORP
    (Name PlatoonSergeant says
      prop (SOME (SLc (PSG psgCommand)))) ∧
  CFGInterpret (M, Oi, Os)
    (CFG authTestConductORP ssmConductORPStateInterp
      (secContextConductORP plCommand psgCommand incomplete)
      (Name PlatoonSergeant says
        prop (SOME (SLc (PSG psgCommand)))::ins) s outs) ∧
    (M, Oi, Os) sat prop (SOME (SLc (PSG psgCommand)))

```

[PlatoonSergeant_psgCommand_lemma]

```

⊢ CFGInterpret (M, Oi, Os)
  (CFG authTestConductORP ssmConductORPStateInterp
    (secContextConductORP plCommand psgCommand incomplete)
    (Name PlatoonSergeant says
      prop (SOME (SLc (PSG psgCommand)))::ins) s outs) ⇒
  (M, Oi, Os) sat prop (SOME (SLc (PSG psgCommand)))

```

9 ConductORPType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

9.1 Datatypes

```

plCommand = secure | withdraw | complete | plIncomplete
psgCommand = actionsIn | psgIncomplete
slCommand =
  PL ConductORPType$plCommand
  | PSG ConductORPType$psgCommand
slOutput = ConductORP | Secure | ActionsIn | Withdraw | Complete
           | unAuthenticated | unAuthorized
slState = CONDUCT_ORP | SECURE | ACTIONS_IN | WITHDRAW
           | COMPLETE
stateRole = PlatoonLeader | PlatoonSergeant

```


9.2 Theorems

[plCommand_distinct_clauses]

$$\vdash \text{secure} \neq \text{withdraw} \wedge \text{secure} \neq \text{complete} \wedge \\ \text{secure} \neq \text{plIncomplete} \wedge \text{withdraw} \neq \text{complete} \wedge \\ \text{withdraw} \neq \text{plIncomplete} \wedge \text{complete} \neq \text{plIncomplete}$$

[psgCommand_distinct_clauses]

$$\vdash \text{actionsIn} \neq \text{psgIncomplete}$$

[slCommand_distinct_clauses]

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

[slCommand_one_one]

$$\vdash (\forall a a'. (\text{PL } a = \text{PL } a') \iff (a = a')) \wedge \\ \forall a a'. (\text{PSG } a = \text{PSG } a') \iff (a = a')$$

[slOutput_distinct_clauses]

$$\vdash \text{ConductORP} \neq \text{Secure} \wedge \text{ConductORP} \neq \text{ActionsIn} \wedge \\ \text{ConductORP} \neq \text{Withdraw} \wedge \text{ConductORP} \neq \text{Complete} \wedge \\ \text{ConductORP} \neq \text{unAuthenticated} \wedge \text{ConductORP} \neq \text{unAuthorized} \wedge \\ \text{Secure} \neq \text{ActionsIn} \wedge \text{Secure} \neq \text{Withdraw} \wedge \text{Secure} \neq \text{Complete} \wedge \\ \text{Secure} \neq \text{unAuthenticated} \wedge \text{Secure} \neq \text{unAuthorized} \wedge \\ \text{ActionsIn} \neq \text{Withdraw} \wedge \text{ActionsIn} \neq \text{Complete} \wedge \\ \text{ActionsIn} \neq \text{unAuthenticated} \wedge \text{ActionsIn} \neq \text{unAuthorized} \wedge \\ \text{Withdraw} \neq \text{Complete} \wedge \text{Withdraw} \neq \text{unAuthenticated} \wedge \\ \text{Withdraw} \neq \text{unAuthorized} \wedge \text{Complete} \neq \text{unAuthenticated} \wedge \\ \text{Complete} \neq \text{unAuthorized} \wedge \text{unAuthenticated} \neq \text{unAuthorized}$$

[slRole_distinct_clauses]

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

[slState_distinct_clauses]

$$\vdash \text{CONDUCT_ORP} \neq \text{SECURE} \wedge \text{CONDUCT_ORP} \neq \text{ACTIONS_IN} \wedge \\ \text{CONDUCT_ORP} \neq \text{WITHDRAW} \wedge \text{CONDUCT_ORP} \neq \text{COMPLETE} \wedge \\ \text{SECURE} \neq \text{ACTIONS_IN} \wedge \text{SECURE} \neq \text{WITHDRAW} \wedge \text{SECURE} \neq \text{COMPLETE} \wedge \\ \text{ACTIONS_IN} \neq \text{WITHDRAW} \wedge \text{ACTIONS_IN} \neq \text{COMPLETE} \wedge \\ \text{WITHDRAW} \neq \text{COMPLETE}$$

10 ssmConductPB Theory

Built: 10 June 2018

Parent Theories: ConductPBType, ssm11, OMNIType

10.1 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

```

10.2 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 v1) ⇔ F) ∧
  (authTestConductPB (v2 andf v3) ⇔ F) ∧
  (authTestConductPB (v4 orf v5) ⇔ F) ∧
  (authTestConductPB (v6 impf v7) ⇔ F) ∧
  (authTestConductPB (v8 eqf v9) ⇔ F) ∧
  (authTestConductPB (v10 says TT) ⇔ F) ∧
  (authTestConductPB (v10 says FF) ⇔ F) ∧
  (authTestConductPB (v133 meet v134 says prop v66) ⇔ F) ∧
  (authTestConductPB (v135 quoting v136 says prop v66) ⇔ F) ∧
  (authTestConductPB (v10 says notf v67) ⇔ F) ∧
  (authTestConductPB (v10 says (v68 andf v69)) ⇔ F) ∧
  (authTestConductPB (v10 says (v70 orf v71)) ⇔ F) ∧
  (authTestConductPB (v10 says (v72 impf v73)) ⇔ F) ∧
  (authTestConductPB (v10 says (v74 eqf v75)) ⇔ F) ∧
  (authTestConductPB (v10 says v76 says v77) ⇔ F) ∧
  (authTestConductPB (v10 says v78 speaks_for v79) ⇔ F) ∧
  (authTestConductPB (v10 says v80 controls v81) ⇔ F) ∧
  (authTestConductPB (v10 says reps v82 v83 v84) ⇔ F) ∧
  (authTestConductPB (v10 says v85 domi v86) ⇔ F) ∧
  (authTestConductPB (v10 says v87 eqi v88) ⇔ F) ∧
  (authTestConductPB (v10 says v89 doms v90) ⇔ F) ∧

```

$(\text{authTestConductPB } (v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{14} \text{ controls } v_{15}) \iff F) \wedge$
 $(\text{authTestConductPB } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{19} \text{ domi } v_{20}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{23} \text{ doms } v_{24}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{29} \text{ lte } v_{30}) \iff F) \wedge$
 $(\text{authTestConductPB } (v_{31} \text{ lt } v_{32}) \iff F)$

$[\text{authTestConductPB_ind}]$

$\vdash \forall P.$

$(\forall \text{cmd}. P (\text{Name PlatoonLeader says prop cmd})) \wedge$
 $(\forall \text{cmd}. P (\text{Name PlatoonSergeant says prop cmd})) \wedge P \text{ TT} \wedge$
 $P \text{ FF} \wedge (\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge$
 $(\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge$
 $(\forall v_6 \ v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 \ v_9. P (v_8 \text{ eqf } v_9)) \wedge$
 $(\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge$
 $(\forall v_{133} \ v_{134} \ v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{135} \ v_{136} \ v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{10} \ v_{67}. P (v_{10} \text{ says notf } v_{67})) \wedge$
 $(\forall v_{10} \ v_{68} \ v_{69}. P (v_{10} \text{ says } (v_{68} \text{ andf } v_{69}))) \wedge$
 $(\forall v_{10} \ v_{70} \ v_{71}. P (v_{10} \text{ says } (v_{70} \text{ orf } v_{71}))) \wedge$
 $(\forall v_{10} \ v_{72} \ v_{73}. P (v_{10} \text{ says } (v_{72} \text{ impf } v_{73}))) \wedge$
 $(\forall v_{10} \ v_{74} \ v_{75}. P (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75}))) \wedge$
 $(\forall v_{10} \ v_{76} \ v_{77}. P (v_{10} \text{ says } v_{76} \text{ says } v_{77})) \wedge$
 $(\forall v_{10} \ v_{78} \ v_{79}. P (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79})) \wedge$
 $(\forall v_{10} \ v_{80} \ v_{81}. P (v_{10} \text{ says } v_{80} \text{ controls } v_{81})) \wedge$
 $(\forall v_{10} \ v_{82} \ v_{83} \ v_{84}. P (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84})) \wedge$
 $(\forall v_{10} \ v_{85} \ v_{86}. P (v_{10} \text{ says } v_{85} \text{ domi } v_{86})) \wedge$
 $(\forall v_{10} \ v_{87} \ v_{88}. P (v_{10} \text{ says } v_{87} \text{ eqi } v_{88})) \wedge$
 $(\forall v_{10} \ v_{89} \ v_{90}. P (v_{10} \text{ says } v_{89} \text{ doms } v_{90})) \wedge$
 $(\forall v_{10} \ v_{91} \ v_{92}. P (v_{10} \text{ says } v_{91} \text{ eqs } v_{92})) \wedge$
 $(\forall v_{10} \ v_{93} \ v_{94}. P (v_{10} \text{ says } v_{93} \text{ eqn } v_{94})) \wedge$
 $(\forall v_{10} \ v_{95} \ v_{96}. P (v_{10} \text{ says } v_{95} \text{ lte } v_{96})) \wedge$
 $(\forall v_{10} \ v_{97} \ v_{98}. P (v_{10} \text{ says } v_{97} \text{ lt } v_{98})) \wedge$
 $(\forall v_{12} \ v_{13}. P (v_{12} \text{ speaks_for } v_{13})) \wedge$
 $(\forall v_{14} \ v_{15}. P (v_{14} \text{ controls } v_{15})) \wedge$
 $(\forall v_{16} \ v_{17} \ v_{18}. P (\text{reps } v_{16} \ v_{17} \ v_{18})) \wedge$
 $(\forall v_{19} \ v_{20}. P (v_{19} \text{ domi } v_{20})) \wedge$
 $(\forall v_{21} \ v_{22}. P (v_{21} \text{ eqi } v_{22})) \wedge$
 $(\forall v_{23} \ v_{24}. P (v_{23} \text{ doms } v_{24})) \wedge$
 $(\forall v_{25} \ v_{26}. P (v_{25} \text{ eqs } v_{26})) \wedge (\forall v_{27} \ v_{28}. P (v_{27} \text{ eqn } v_{28})) \wedge$
 $(\forall v_{29} \ v_{30}. P (v_{29} \text{ lte } v_{30})) \wedge (\forall v_{31} \ v_{32}. P (v_{31} \text{ lt } v_{32})) \Rightarrow$

$$\forall v. P \ v$$

[conductPBNS_def]

$$\begin{aligned} \vdash & (\text{conductPBNS CONDUCT_PB (exec (PL securePB))} = \text{SECURE_PB}) \wedge \\ & (\text{conductPBNS CONDUCT_PB (exec (PL plIncompletePB))} = \\ & \quad \text{CONDUCT_PB}) \wedge \\ & (\text{conductPBNS SECURE_PB (exec (PSG actionsInPB))} = \\ & \quad \text{ACTIONS_IN_PB}) \wedge \\ & (\text{conductPBNS SECURE_PB (exec (PSG psgIncompletePB))} = \\ & \quad \text{SECURE_PB}) \wedge \\ & (\text{conductPBNS ACTIONS_IN_PB (exec (PL withdrawPB))} = \\ & \quad \text{WITHDRAW_PB}) \wedge \\ & (\text{conductPBNS ACTIONS_IN_PB (exec (PL plIncompletePB))} = \\ & \quad \text{ACTIONS_IN_PB}) \wedge \\ & (\text{conductPBNS WITHDRAW_PB (exec (PL completePB))} = \\ & \quad \text{COMPLETE_PB}) \wedge \\ & (\text{conductPBNS WITHDRAW_PB (exec (PL plIncompletePB))} = \\ & \quad \text{WITHDRAW_PB}) \wedge (\text{conductPBNS } s \text{ (trap (PL cmd'))} = s) \wedge \\ & (\text{conductPBNS } s \text{ (trap (PSG cmd))} = s) \wedge \\ & (\text{conductPBNS } s \text{ (discard (PL cmd'))} = s) \wedge \\ & (\text{conductPBNS } s \text{ (discard (PSG cmd))} = s) \end{aligned}$$

[conductPBNS_ind]

$$\begin{aligned} \vdash & \forall P. \\ & P \text{ CONDUCT_PB (exec (PL securePB))} \wedge \\ & P \text{ CONDUCT_PB (exec (PL plIncompletePB))} \wedge \\ & P \text{ SECURE_PB (exec (PSG actionsInPB))} \wedge \\ & P \text{ SECURE_PB (exec (PSG psgIncompletePB))} \wedge \\ & P \text{ ACTIONS_IN_PB (exec (PL withdrawPB))} \wedge \\ & P \text{ ACTIONS_IN_PB (exec (PL plIncompletePB))} \wedge \\ & P \text{ WITHDRAW_PB (exec (PL completePB))} \wedge \\ & P \text{ WITHDRAW_PB (exec (PL plIncompletePB))} \wedge \\ & (\forall s \text{ cmd. } P \ s \text{ (trap (PL cmd))}) \wedge \\ & (\forall s \text{ cmd. } P \ s \text{ (trap (PSG cmd))}) \wedge \\ & (\forall s \text{ cmd. } P \ s \text{ (discard (PL cmd))}) \wedge \\ & (\forall s \text{ cmd. } P \ s \text{ (discard (PSG cmd))}) \wedge \\ & P \text{ CONDUCT_PB (exec (PL withdrawPB))} \wedge \\ & P \text{ CONDUCT_PB (exec (PL completePB))} \wedge \\ & (\forall v_{11}. P \text{ CONDUCT_PB (exec (PSG } v_{11})) \wedge \\ & (\forall v_{13}. P \text{ SECURE_PB (exec (PL } v_{13})) \wedge \\ & P \text{ ACTIONS_IN_PB (exec (PL securePB))} \wedge \\ & P \text{ ACTIONS_IN_PB (exec (PL completePB))} \wedge \\ & (\forall v_{17}. P \text{ ACTIONS_IN_PB (exec (PSG } v_{17})) \wedge \\ & P \text{ WITHDRAW_PB (exec (PL securePB))} \wedge \\ & P \text{ WITHDRAW_PB (exec (PL withdrawPB))} \wedge \\ & (\forall v_{20}. P \text{ WITHDRAW_PB (exec (PSG } v_{20})) \wedge \\ & (\forall v_{21}. P \text{ COMPLETE_PB (exec } v_{21})) \Rightarrow \\ & \forall v \ v_1. P \ v \ v_1 \end{aligned}$$

[conductPBOut_def]

$$\begin{aligned}
&\vdash (\text{conductPBOut CONDUCT_PB (exec (PL securePB))} = \text{ConductPB}) \wedge \\
&\quad (\text{conductPBOut CONDUCT_PB (exec (PL plIncompletePB))} = \\
&\quad \quad \text{ConductPB}) \wedge \\
&\quad (\text{conductPBOut SECURE_PB (exec (PSG actionsInPB))} = \\
&\quad \quad \text{SecurePB}) \wedge \\
&\quad (\text{conductPBOut SECURE_PB (exec (PSG psgIncompletePB))} = \\
&\quad \quad \text{SecurePB}) \wedge \\
&\quad (\text{conductPBOut ACTIONS_IN_PB (exec (PL withdrawPB))} = \\
&\quad \quad \text{ActionsInPB}) \wedge \\
&\quad (\text{conductPBOut ACTIONS_IN_PB (exec (PL plIncompletePB))} = \\
&\quad \quad \text{ActionsInPB}) \wedge \\
&\quad (\text{conductPBOut WITHDRAW_PB (exec (PL completePB))} = \\
&\quad \quad \text{WithdrawPB}) \wedge \\
&\quad (\text{conductPBOut WITHDRAW_PB (exec (PL plIncompletePB))} = \\
&\quad \quad \text{WithdrawPB}) \wedge \\
&\quad (\text{conductPBOut } s \text{ (trap (PL cmd'))} = \text{unAuthorized}) \wedge \\
&\quad (\text{conductPBOut } s \text{ (trap (PSG cmd))} = \text{unAuthorized}) \wedge \\
&\quad (\text{conductPBOut } s \text{ (discard (PL cmd'))} = \text{unAuthenticated}) \wedge \\
&\quad (\text{conductPBOut } s \text{ (discard (PSG cmd))} = \text{unAuthenticated})
\end{aligned}$$
[conductPBOut_ind]

$$\begin{aligned}
&\vdash \forall P. \\
&\quad P \text{ CONDUCT_PB (exec (PL securePB))} \wedge \\
&\quad P \text{ CONDUCT_PB (exec (PL plIncompletePB))} \wedge \\
&\quad P \text{ SECURE_PB (exec (PSG actionsInPB))} \wedge \\
&\quad P \text{ SECURE_PB (exec (PSG psgIncompletePB))} \wedge \\
&\quad P \text{ ACTIONS_IN_PB (exec (PL withdrawPB))} \wedge \\
&\quad P \text{ ACTIONS_IN_PB (exec (PL plIncompletePB))} \wedge \\
&\quad P \text{ WITHDRAW_PB (exec (PL completePB))} \wedge \\
&\quad P \text{ WITHDRAW_PB (exec (PL plIncompletePB))} \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (trap (PL cmd))}) \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (trap (PSG cmd))}) \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (discard (PL cmd))}) \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (discard (PSG cmd))}) \wedge \\
&\quad P \text{ CONDUCT_PB (exec (PL withdrawPB))} \wedge \\
&\quad P \text{ CONDUCT_PB (exec (PL completePB))} \wedge \\
&\quad (\forall v_{11}. P \text{ CONDUCT_PB (exec (PSG } v_{11})) \wedge \\
&\quad (\forall v_{13}. P \text{ SECURE_PB (exec (PL } v_{13})) \wedge \\
&\quad P \text{ ACTIONS_IN_PB (exec (PL securePB))} \wedge \\
&\quad P \text{ ACTIONS_IN_PB (exec (PL completePB))} \wedge \\
&\quad (\forall v_{17}. P \text{ ACTIONS_IN_PB (exec (PSG } v_{17})) \wedge \\
&\quad P \text{ WITHDRAW_PB (exec (PL securePB))} \wedge \\
&\quad P \text{ WITHDRAW_PB (exec (PL withdrawPB))} \wedge \\
&\quad (\forall v_{20}. P \text{ WITHDRAW_PB (exec (PSG } v_{20})) \wedge \\
&\quad (\forall v_{21}. P \text{ COMPLETE_PB (exec } v_{21})) \Rightarrow \\
&\quad \forall v \text{ } v_1. P \text{ } v \text{ } v_1
\end{aligned}$$

[PlatoonLeader_exec_plCommandPB_justified_thm]

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

[PlatoonLeader_plCommandPB_lemma]

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

[PlatoonSergeant_exec_psgCommandPB_justified_thm]

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

```

11 ConductPBType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

11.1 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

```

11.2 Theorems

[plCommandPB_distinct_clauses]

```

⊢ securePB ≠ withdrawPB ∧ securePB ≠ completePB ∧
  securePB ≠ plIncompletePB ∧ withdrawPB ≠ completePB ∧
  withdrawPB ≠ plIncompletePB ∧ completePB ≠ plIncompletePB

```

[psgCommandPB_distinct_clauses]

```

⊢ actionsInPB ≠ psgIncompletePB

```

[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]

$$\begin{aligned}
&\vdash \text{ConductPB} \neq \text{SecurePB} \wedge \text{ConductPB} \neq \text{ActionsInPB} \wedge \\
&\quad \text{ConductPB} \neq \text{WithdrawPB} \wedge \text{ConductPB} \neq \text{CompletePB} \wedge \\
&\quad \text{ConductPB} \neq \text{unAuthenticated} \wedge \text{ConductPB} \neq \text{unAuthorized} \wedge \\
&\quad \text{SecurePB} \neq \text{ActionsInPB} \wedge \text{SecurePB} \neq \text{WithdrawPB} \wedge \\
&\quad \text{SecurePB} \neq \text{CompletePB} \wedge \text{SecurePB} \neq \text{unAuthenticated} \wedge \\
&\quad \text{SecurePB} \neq \text{unAuthorized} \wedge \text{ActionsInPB} \neq \text{WithdrawPB} \wedge \\
&\quad \text{ActionsInPB} \neq \text{CompletePB} \wedge \text{ActionsInPB} \neq \text{unAuthenticated} \wedge \\
&\quad \text{ActionsInPB} \neq \text{unAuthorized} \wedge \text{WithdrawPB} \neq \text{CompletePB} \wedge \\
&\quad \text{WithdrawPB} \neq \text{unAuthenticated} \wedge \text{WithdrawPB} \neq \text{unAuthorized} \wedge \\
&\quad \text{CompletePB} \neq \text{unAuthenticated} \wedge \text{CompletePB} \neq \text{unAuthorized} \wedge \\
&\quad \text{unAuthenticated} \neq \text{unAuthorized}
\end{aligned}$$

[slRole_distinct_clauses]

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

[slState_distinct_clauses]

$$\begin{aligned}
&\vdash \text{CONDUCT_PB} \neq \text{SECURE_PB} \wedge \text{CONDUCT_PB} \neq \text{ACTIONS_IN_PB} \wedge \\
&\quad \text{CONDUCT_PB} \neq \text{WITHDRAW_PB} \wedge \text{CONDUCT_PB} \neq \text{COMPLETE_PB} \wedge \\
&\quad \text{SECURE_PB} \neq \text{ACTIONS_IN_PB} \wedge \text{SECURE_PB} \neq \text{WITHDRAW_PB} \wedge \\
&\quad \text{SECURE_PB} \neq \text{COMPLETE_PB} \wedge \text{ACTIONS_IN_PB} \neq \text{WITHDRAW_PB} \wedge \\
&\quad \text{ACTIONS_IN_PB} \neq \text{COMPLETE_PB} \wedge \text{WITHDRAW_PB} \neq \text{COMPLETE_PB}
\end{aligned}$$

12 ssmMoveToORP Theory

Built: 10 June 2018

Parent Theories: MoveToORPType, ssm11, OMNIType

12.1 Definitions

[secContextMoveToORP_def]

$$\begin{aligned}
&\vdash \forall cmd. \\
&\quad \text{secContextMoveToORP } cmd = \\
&\quad [\text{Name PlatoonLeader controls prop (SOME (SLc cmd))}]
\end{aligned}$$

[ssmMoveToORPStateInterp_def]

$$\vdash \forall state. \text{ssmMoveToORPStateInterp } state = \text{TT}$$

12.2 Theorems

[authTestMoveToORP_cmd_reject_lemma]

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

[authTestMoveToORP_def]

$$\begin{aligned}
&\vdash (\text{authTestMoveToORP } (\text{Name PlatoonLeader says prop cmd}) \iff T) \wedge \\
&\quad (\text{authTestMoveToORP TT} \iff F) \wedge (\text{authTestMoveToORP FF} \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (\text{prop } v) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (\text{notf } v_1) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_2 \text{ andf } v_3) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_4 \text{ orf } v_5) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_6 \text{ impf } v_7) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_8 \text{ eqf } v_9) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says TT}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says FF}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says notf } v_{67}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } (v_{68} \text{ andf } v_{69})) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } (v_{70} \text{ orf } v_{71})) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } (v_{72} \text{ impf } v_{73})) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75})) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{76} \text{ says } v_{77}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{80} \text{ controls } v_{81}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{85} \text{ domi } v_{86}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{87} \text{ eqi } v_{88}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{89} \text{ doms } v_{90}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{14} \text{ controls } v_{15}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{19} \text{ domi } v_{20}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{23} \text{ doms } v_{24}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{29} \text{ lte } v_{30}) \iff F) \wedge \\
&\quad (\text{authTestMoveToORP } (v_{31} \text{ lt } v_{32}) \iff F)
\end{aligned}$$
[authTestMoveToORP_ind]

$$\begin{aligned}
&\vdash \forall P. \\
&\quad (\forall \text{cmd}. P (\text{Name PlatoonLeader says prop cmd})) \wedge P \text{ TT} \wedge \\
&\quad P \text{ FF} \wedge (\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge \\
&\quad (\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge \\
&\quad (\forall v_6 \ v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 \ v_9. P (v_8 \text{ eqf } v_9)) \wedge \\
&\quad (\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge \\
&\quad (\forall v_{133} \ v_{134} \ v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge \\
&\quad (\forall v_{135} \ v_{136} \ v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\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}$$

[moveToORPNS_def]

$$\begin{aligned}
& \vdash (\text{moveToORPNS MOVE_TO_ORP (exec (SLc pltForm))} = \text{PLT_FORM}) \wedge \\
& (\text{moveToORPNS MOVE_TO_ORP (exec (SLc incomplete))} = \\
& \quad \text{MOVE_TO_ORP}) \wedge \\
& (\text{moveToORPNS PLT_FORM (exec (SLc pltMove))} = \text{PLT_MOVE}) \wedge \\
& (\text{moveToORPNS PLT_FORM (exec (SLc incomplete))} = \text{PLT_FORM}) \wedge \\
& (\text{moveToORPNS PLT_MOVE (exec (SLc pltSecureHalt))} = \\
& \quad \text{PLT_SECURE_HALT}) \wedge \\
& (\text{moveToORPNS PLT_MOVE (exec (SLc incomplete))} = \text{PLT_MOVE}) \wedge \\
& (\text{moveToORPNS PLT_SECURE_HALT (exec (SLc complete))} = \\
& \quad \text{COMPLETE}) \wedge \\
& (\text{moveToORPNS PLT_SECURE_HALT (exec (SLc incomplete))} = \\
& \quad \text{PLT_SECURE_HALT}) \wedge (\text{moveToORPNS } s \text{ (trap (SLc cmd))} = s) \wedge \\
& (\text{moveToORPNS } s \text{ (discard (SLc cmd))} = s)
\end{aligned}$$

[moveToORPNS_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& \quad P \text{ MOVE_TO_ORP (exec (SLc pltForm))} \wedge \\
& \quad P \text{ MOVE_TO_ORP (exec (SLc incomplete))} \wedge \\
& \quad P \text{ PLT_FORM (exec (SLc pltMove))} \wedge \\
& \quad P \text{ PLT_FORM (exec (SLc incomplete))} \wedge \\
& \quad P \text{ PLT_MOVE (exec (SLc pltSecureHalt))} \wedge \\
& \quad P \text{ PLT_MOVE (exec (SLc incomplete))} \wedge
\end{aligned}$$

$P \text{ PLT_SECURE_HALT (exec (SLc complete))} \wedge$
 $P \text{ PLT_SECURE_HALT (exec (SLc incomplete))} \wedge$
 $(\forall s \text{ cmd. } P \text{ s (trap (SLc cmd))}) \wedge$
 $(\forall s \text{ cmd. } P \text{ s (discard (SLc cmd))}) \wedge$
 $(\forall s \text{ v}_6. P \text{ s (discard (ESCc v}_6\text{))}) \wedge$
 $(\forall s \text{ v}_9. P \text{ s (trap (ESCc v}_9\text{))}) \wedge$
 $(\forall v_{12}. P \text{ MOVE_TO_ORP (exec (ESCc v}_{12}\text{))}) \wedge$
 $P \text{ MOVE_TO_ORP (exec (SLc pltMove))} \wedge$
 $P \text{ MOVE_TO_ORP (exec (SLc pltSecureHalt))} \wedge$
 $P \text{ MOVE_TO_ORP (exec (SLc complete))} \wedge$
 $(\forall v_{15}. P \text{ PLT_FORM (exec (ESCc v}_{15}\text{))}) \wedge$
 $P \text{ PLT_FORM (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_FORM (exec (SLc pltSecureHalt))} \wedge$
 $P \text{ PLT_FORM (exec (SLc complete))} \wedge$
 $(\forall v_{18}. P \text{ PLT_MOVE (exec (ESCc v}_{18}\text{))}) \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc complete))} \wedge$
 $(\forall v_{21}. P \text{ PLT_SECURE_HALT (exec (ESCc v}_{21}\text{))}) \wedge$
 $P \text{ PLT_SECURE_HALT (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_SECURE_HALT (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_SECURE_HALT (exec (SLc pltSecureHalt))} \wedge$
 $(\forall v_{23}. P \text{ COMPLETE (exec v}_{23}\text{)}) \Rightarrow$
 $\forall v \text{ v}_1. P \text{ v v}_1$

[moveToORPOut_def]

$\vdash (\text{moveToORPOut MOVE_TO_ORP (exec (SLc pltForm))} = \text{PLTForm}) \wedge$
 $(\text{moveToORPOut MOVE_TO_ORP (exec (SLc incomplete))} =$
 $\text{MoveToORP}) \wedge$
 $(\text{moveToORPOut PLT_FORM (exec (SLc pltMove))} = \text{PLTMove}) \wedge$
 $(\text{moveToORPOut PLT_FORM (exec (SLc incomplete))} = \text{PLTForm}) \wedge$
 $(\text{moveToORPOut PLT_MOVE (exec (SLc pltSecureHalt))} =$
 $\text{PLTSecureHalt}) \wedge$
 $(\text{moveToORPOut PLT_MOVE (exec (SLc incomplete))} = \text{PLTMove}) \wedge$
 $(\text{moveToORPOut PLT_SECURE_HALT (exec (SLc complete))} =$
 $\text{Complete}) \wedge$
 $(\text{moveToORPOut PLT_SECURE_HALT (exec (SLc incomplete))} =$
 $\text{PLTSecureHalt}) \wedge$
 $(\text{moveToORPOut s (trap (SLc cmd))} = \text{unAuthorized}) \wedge$
 $(\text{moveToORPOut s (discard (SLc cmd))} = \text{unAuthenticated})$

[moveToORPOut_ind]

$\vdash \forall P.$
 $P \text{ MOVE_TO_ORP (exec (SLc pltForm))} \wedge$
 $P \text{ MOVE_TO_ORP (exec (SLc incomplete))} \wedge$
 $P \text{ PLT_FORM (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_FORM (exec (SLc incomplete))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltSecureHalt))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc incomplete))} \wedge$

P PLT_SECURE_HALT (exec (SLc complete)) \wedge
 P PLT_SECURE_HALT (exec (SLc incomplete)) \wedge
 $(\forall s \text{ cmd}. P \ s \ (\text{trap} \ (\text{SLc} \ \text{cmd}))) \wedge$
 $(\forall s \text{ cmd}. P \ s \ (\text{discard} \ (\text{SLc} \ \text{cmd}))) \wedge$
 $(\forall s \ v_6. P \ s \ (\text{discard} \ (\text{ESCc} \ v_6))) \wedge$
 $(\forall s \ v_9. P \ s \ (\text{trap} \ (\text{ESCc} \ v_9))) \wedge$
 $(\forall v_{12}. P \ \text{MOVE_TO_ORP} \ (\text{exec} \ (\text{ESCc} \ v_{12}))) \wedge$
 $P \ \text{MOVE_TO_ORP} \ (\text{exec} \ (\text{SLc} \ \text{pltMove})) \wedge$
 $P \ \text{MOVE_TO_ORP} \ (\text{exec} \ (\text{SLc} \ \text{pltSecureHalt})) \wedge$
 $P \ \text{MOVE_TO_ORP} \ (\text{exec} \ (\text{SLc} \ \text{complete})) \wedge$
 $(\forall v_{15}. P \ \text{PLT_FORM} \ (\text{exec} \ (\text{ESCc} \ v_{15}))) \wedge$
 $P \ \text{PLT_FORM} \ (\text{exec} \ (\text{SLc} \ \text{pltForm})) \wedge$
 $P \ \text{PLT_FORM} \ (\text{exec} \ (\text{SLc} \ \text{pltSecureHalt})) \wedge$
 $P \ \text{PLT_FORM} \ (\text{exec} \ (\text{SLc} \ \text{complete})) \wedge$
 $(\forall v_{18}. P \ \text{PLT_MOVE} \ (\text{exec} \ (\text{ESCc} \ v_{18}))) \wedge$
 $P \ \text{PLT_MOVE} \ (\text{exec} \ (\text{SLc} \ \text{pltForm})) \wedge$
 $P \ \text{PLT_MOVE} \ (\text{exec} \ (\text{SLc} \ \text{pltMove})) \wedge$
 $P \ \text{PLT_MOVE} \ (\text{exec} \ (\text{SLc} \ \text{complete})) \wedge$
 $(\forall v_{21}. P \ \text{PLT_SECURE_HALT} \ (\text{exec} \ (\text{ESCc} \ v_{21}))) \wedge$
 $P \ \text{PLT_SECURE_HALT} \ (\text{exec} \ (\text{SLc} \ \text{pltForm})) \wedge$
 $P \ \text{PLT_SECURE_HALT} \ (\text{exec} \ (\text{SLc} \ \text{pltMove})) \wedge$
 $P \ \text{PLT_SECURE_HALT} \ (\text{exec} \ (\text{SLc} \ \text{pltSecureHalt})) \wedge$
 $(\forall v_{23}. P \ \text{COMPLETE} \ (\text{exec} \ v_{23})) \Rightarrow$
 $\forall v \ v_1. P \ v \ v_1$

[PlatoonLeader_exec_slCommand_justified_thm]

$\vdash \forall NS \ Out \ M \ Oi \ Os.$
 $\text{TR} \ (M, Oi, Os) \ (\text{exec} \ (\text{SLc} \ \text{slCommand}))$
 $(\text{CFG} \ \text{authTestMoveToORP} \ \text{ssmMoveToORPStateInterp}$
 $\quad (\text{secContextMoveToORP} \ \text{slCommand})$
 $\quad (\text{Name} \ \text{PlatoonLeader} \ \text{says} \ \text{prop} \ (\text{SOME} \ (\text{SLc} \ \text{slCommand})))::$
 $\quad \text{ins}) \ s \ \text{outs})$
 $(\text{CFG} \ \text{authTestMoveToORP} \ \text{ssmMoveToORPStateInterp}$
 $\quad (\text{secContextMoveToORP} \ \text{slCommand}) \ \text{ins}$
 $\quad (NS \ s \ (\text{exec} \ (\text{SLc} \ \text{slCommand}))))$
 $\quad (\text{Out} \ s \ (\text{exec} \ (\text{SLc} \ \text{slCommand}))):: \text{outs}) \iff$
 authTestMoveToORP
 $\quad (\text{Name} \ \text{PlatoonLeader} \ \text{says} \ \text{prop} \ (\text{SOME} \ (\text{SLc} \ \text{slCommand}))) \wedge$
 $\text{CFGInterpret} \ (M, Oi, Os)$
 $\quad (\text{CFG} \ \text{authTestMoveToORP} \ \text{ssmMoveToORPStateInterp}$
 $\quad (\text{secContextMoveToORP} \ \text{slCommand})$
 $\quad (\text{Name} \ \text{PlatoonLeader} \ \text{says} \ \text{prop} \ (\text{SOME} \ (\text{SLc} \ \text{slCommand})))::$
 $\quad \text{ins}) \ s \ \text{outs}) \wedge$
 $(M, Oi, Os) \ \text{sat} \ \text{prop} \ (\text{SOME} \ (\text{SLc} \ \text{slCommand}))$

[PlatoonLeader_slCommand_lemma]

$\vdash \text{CFGInterpret} \ (M, Oi, Os)$
 $\quad (\text{CFG} \ \text{authTestMoveToORP} \ \text{ssmMoveToORPStateInterp}$
 $\quad (\text{secContextMoveToORP} \ \text{slCommand})$

$(\text{Name PlatoonLeader says prop (SOME (SLc slCommand)))::$
 $\text{ins) } s \text{ outs) } \Rightarrow$
 $(M, Oi, Os) \text{ sat prop (SOME (SLc slCommand))}$

13 MoveToORPType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

13.1 Datatypes

$\text{slCommand} = \text{pltForm} \mid \text{pltMove} \mid \text{pltSecureHalt} \mid \text{complete}$
 $\quad \mid \text{incomplete}$
 $\text{slOutput} = \text{MoveToORP} \mid \text{PLTForm} \mid \text{PLTMove} \mid \text{PLTSecureHalt}$
 $\quad \mid \text{Complete} \mid \text{unAuthorized} \mid \text{unAuthenticated}$
 $\text{slState} = \text{MOVE_TO_ORP} \mid \text{PLT_FORM} \mid \text{PLT_MOVE} \mid \text{PLT_SECURE_HALT}$
 $\quad \mid \text{COMPLETE}$
 $\text{stateRole} = \text{PlatoonLeader}$

13.2 Theorems

[slCommand_distinct_clauses]

$\vdash \text{pltForm} \neq \text{pltMove} \wedge \text{pltForm} \neq \text{pltSecureHalt} \wedge$
 $\text{pltForm} \neq \text{complete} \wedge \text{pltForm} \neq \text{incomplete} \wedge$
 $\text{pltMove} \neq \text{pltSecureHalt} \wedge \text{pltMove} \neq \text{complete} \wedge$
 $\text{pltMove} \neq \text{incomplete} \wedge \text{pltSecureHalt} \neq \text{complete} \wedge$
 $\text{pltSecureHalt} \neq \text{incomplete} \wedge \text{complete} \neq \text{incomplete}$

[slOutput_distinct_clauses]

$\vdash \text{MoveToORP} \neq \text{PLTForm} \wedge \text{MoveToORP} \neq \text{PLTMove} \wedge$
 $\text{MoveToORP} \neq \text{PLTSecureHalt} \wedge \text{MoveToORP} \neq \text{Complete} \wedge$
 $\text{MoveToORP} \neq \text{unAuthorized} \wedge \text{MoveToORP} \neq \text{unAuthenticated} \wedge$
 $\text{PLTForm} \neq \text{PLTMove} \wedge \text{PLTForm} \neq \text{PLTSecureHalt} \wedge$
 $\text{PLTForm} \neq \text{Complete} \wedge \text{PLTForm} \neq \text{unAuthorized} \wedge$
 $\text{PLTForm} \neq \text{unAuthenticated} \wedge \text{PLTMove} \neq \text{PLTSecureHalt} \wedge$
 $\text{PLTMove} \neq \text{Complete} \wedge \text{PLTMove} \neq \text{unAuthorized} \wedge$
 $\text{PLTMove} \neq \text{unAuthenticated} \wedge \text{PLTSecureHalt} \neq \text{Complete} \wedge$
 $\text{PLTSecureHalt} \neq \text{unAuthorized} \wedge$
 $\text{PLTSecureHalt} \neq \text{unAuthenticated} \wedge \text{Complete} \neq \text{unAuthorized} \wedge$
 $\text{Complete} \neq \text{unAuthenticated} \wedge \text{unAuthorized} \neq \text{unAuthenticated}$

[slState_distinct_clauses]

$\vdash \text{MOVE_TO_ORP} \neq \text{PLT_FORM} \wedge \text{MOVE_TO_ORP} \neq \text{PLT_MOVE} \wedge$
 $\text{MOVE_TO_ORP} \neq \text{PLT_SECURE_HALT} \wedge \text{MOVE_TO_ORP} \neq \text{COMPLETE} \wedge$
 $\text{PLT_FORM} \neq \text{PLT_MOVE} \wedge \text{PLT_FORM} \neq \text{PLT_SECURE_HALT} \wedge$
 $\text{PLT_FORM} \neq \text{COMPLETE} \wedge \text{PLT_MOVE} \neq \text{PLT_SECURE_HALT} \wedge$
 $\text{PLT_MOVE} \neq \text{COMPLETE} \wedge \text{PLT_SECURE_HALT} \neq \text{COMPLETE}$

14 ssmMoveToPB Theory

Built: 10 June 2018

Parent Theories: MoveToPBType, ssm11, OMNIType

14.1 Definitions

[secContextMoveToPB_def]

$$\vdash \forall cmd. \\ \text{secContextMoveToPB } cmd = \\ [\text{Name PlatoonLeader controls prop (SOME (SLc cmd))}]$$

[ssmMoveToPBStateInterp_def]

$$\vdash \forall state. \text{ssmMoveToPBStateInterp } state = \text{TT}$$

14.2 Theorems

[authTestMoveToPB_cmd_reject_lemma]

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

[authTestMoveToPB_def]

$$\vdash (\text{authTestMoveToPB (Name PlatoonLeader says prop cmd)} \iff \text{T}) \wedge \\ (\text{authTestMoveToPB TT} \iff \text{F}) \wedge (\text{authTestMoveToPB FF} \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (prop } v) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (notf } v_1) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_2 \text{ andf } v_3) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_4 \text{ orf } v_5) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_6 \text{ impf } v_7) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_8 \text{ eqf } v_9) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says TT) } \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says FF) } \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{133} \text{ meet } v_{134} \text{ says prop } v_{66}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says notf } v_{67}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says (} v_{68} \text{ andf } v_{69}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says (} v_{70} \text{ orf } v_{71}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says (} v_{72} \text{ impf } v_{73}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says (} v_{74} \text{ eqf } v_{75}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{76} \text{ says } v_{77}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{80} \text{ controls } v_{81}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says reps } v_{82} \text{ } v_{83} \text{ } v_{84}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{85} \text{ domi } v_{86}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{87} \text{ eqi } v_{88}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{89} \text{ doms } v_{90}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff \text{F}) \wedge \\ (\text{authTestMoveToPB (} v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff \text{F}) \wedge$$

$(\text{authTestMoveToPB } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{14} \text{ controls } v_{15}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{19} \text{ domi } v_{20}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{23} \text{ doms } v_{24}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{29} \text{ lte } v_{30}) \iff F) \wedge$
 $(\text{authTestMoveToPB } (v_{31} \text{ lt } v_{32}) \iff F)$

[authTestMoveToPB_ind]

$\vdash \forall P.$

$(\forall \text{cmd}. P (\text{Name PlatoonLeader says prop cmd})) \wedge P \text{ TT} \wedge$
 $P \text{ FF} \wedge (\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge$
 $(\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge$
 $(\forall v_6 \ v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 \ v_9. P (v_8 \text{ eqf } v_9)) \wedge$
 $(\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge$
 $(\forall v_{133} \ v_{134} \ v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{135} \ v_{136} \ v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{10} \ v_{67}. P (v_{10} \text{ says notf } v_{67})) \wedge$
 $(\forall v_{10} \ v_{68} \ v_{69}. P (v_{10} \text{ says } (v_{68} \text{ andf } v_{69}))) \wedge$
 $(\forall v_{10} \ v_{70} \ v_{71}. P (v_{10} \text{ says } (v_{70} \text{ orf } v_{71}))) \wedge$
 $(\forall v_{10} \ v_{72} \ v_{73}. P (v_{10} \text{ says } (v_{72} \text{ impf } v_{73}))) \wedge$
 $(\forall v_{10} \ v_{74} \ v_{75}. P (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75}))) \wedge$
 $(\forall v_{10} \ v_{76} \ v_{77}. P (v_{10} \text{ says } v_{76} \text{ says } v_{77})) \wedge$
 $(\forall v_{10} \ v_{78} \ v_{79}. P (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79})) \wedge$
 $(\forall v_{10} \ v_{80} \ v_{81}. P (v_{10} \text{ says } v_{80} \text{ controls } v_{81})) \wedge$
 $(\forall v_{10} \ v_{82} \ v_{83} \ v_{84}. P (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84})) \wedge$
 $(\forall v_{10} \ v_{85} \ v_{86}. P (v_{10} \text{ says } v_{85} \text{ domi } v_{86})) \wedge$
 $(\forall v_{10} \ v_{87} \ v_{88}. P (v_{10} \text{ says } v_{87} \text{ eqi } v_{88})) \wedge$
 $(\forall v_{10} \ v_{89} \ v_{90}. P (v_{10} \text{ says } v_{89} \text{ doms } v_{90})) \wedge$
 $(\forall v_{10} \ v_{91} \ v_{92}. P (v_{10} \text{ says } v_{91} \text{ eqs } v_{92})) \wedge$
 $(\forall v_{10} \ v_{93} \ v_{94}. P (v_{10} \text{ says } v_{93} \text{ eqn } v_{94})) \wedge$
 $(\forall v_{10} \ v_{95} \ v_{96}. P (v_{10} \text{ says } v_{95} \text{ lte } v_{96})) \wedge$
 $(\forall v_{10} \ v_{97} \ v_{98}. P (v_{10} \text{ says } v_{97} \text{ lt } v_{98})) \wedge$
 $(\forall v_{12} \ v_{13}. P (v_{12} \text{ speaks_for } v_{13})) \wedge$
 $(\forall v_{14} \ v_{15}. P (v_{14} \text{ controls } v_{15})) \wedge$
 $(\forall v_{16} \ v_{17} \ v_{18}. P (\text{reps } v_{16} \ v_{17} \ v_{18})) \wedge$
 $(\forall v_{19} \ v_{20}. P (v_{19} \text{ domi } v_{20})) \wedge$
 $(\forall v_{21} \ v_{22}. P (v_{21} \text{ eqi } v_{22})) \wedge$
 $(\forall v_{23} \ v_{24}. P (v_{23} \text{ doms } v_{24})) \wedge$
 $(\forall v_{25} \ v_{26}. P (v_{25} \text{ eqs } v_{26})) \wedge (\forall v_{27} \ v_{28}. P (v_{27} \text{ eqn } v_{28})) \wedge$
 $(\forall v_{29} \ v_{30}. P (v_{29} \text{ lte } v_{30})) \wedge (\forall v_{31} \ v_{32}. P (v_{31} \text{ lt } v_{32})) \Rightarrow$
 $\forall v. P \ v$

[moveToPBNS_def]

$$\begin{aligned}
&\vdash (\text{moveToPBNS MOVE_TO_PB (exec (SLc pltForm))} = \text{PLT_FORM}) \wedge \\
&\quad (\text{moveToPBNS MOVE_TO_PB (exec (SLc incomplete))} = \\
&\quad \quad \text{MOVE_TO_PB}) \wedge \\
&\quad (\text{moveToPBNS PLT_FORM (exec (SLc pltMove))} = \text{PLT_MOVE}) \wedge \\
&\quad (\text{moveToPBNS PLT_FORM (exec (SLc incomplete))} = \text{PLT_FORM}) \wedge \\
&\quad (\text{moveToPBNS PLT_MOVE (exec (SLc pltHalt))} = \text{PLT_HALT}) \wedge \\
&\quad (\text{moveToPBNS PLT_MOVE (exec (SLc incomplete))} = \text{PLT_MOVE}) \wedge \\
&\quad (\text{moveToPBNS PLT_HALT (exec (SLc complete))} = \text{COMPLETE}) \wedge \\
&\quad (\text{moveToPBNS PLT_HALT (exec (SLc incomplete))} = \text{PLT_HALT}) \wedge \\
&\quad (\text{moveToPBNS } s \text{ (trap (SLc cmd))} = s) \wedge \\
&\quad (\text{moveToPBNS } s \text{ (discard (SLc cmd))} = s)
\end{aligned}$$

[moveToPBNS_ind]

$$\begin{aligned}
&\vdash \forall P. \\
&\quad P \text{ MOVE_TO_PB (exec (SLc pltForm))} \wedge \\
&\quad P \text{ MOVE_TO_PB (exec (SLc incomplete))} \wedge \\
&\quad P \text{ PLT_FORM (exec (SLc pltMove))} \wedge \\
&\quad P \text{ PLT_FORM (exec (SLc incomplete))} \wedge \\
&\quad P \text{ PLT_MOVE (exec (SLc pltHalt))} \wedge \\
&\quad P \text{ PLT_MOVE (exec (SLc incomplete))} \wedge \\
&\quad P \text{ PLT_HALT (exec (SLc complete))} \wedge \\
&\quad P \text{ PLT_HALT (exec (SLc incomplete))} \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (trap (SLc cmd))}) \wedge \\
&\quad (\forall s \text{ cmd. } P \text{ } s \text{ (discard (SLc cmd))}) \wedge \\
&\quad (\forall s \text{ } v_6. P \text{ } s \text{ (discard (ESCc } v_6))) \wedge \\
&\quad (\forall s \text{ } v_9. P \text{ } s \text{ (trap (ESCc } v_9))) \wedge \\
&\quad (\forall v_{12}. P \text{ MOVE_TO_PB (exec (ESCc } v_{12}))) \wedge \\
&\quad P \text{ MOVE_TO_PB (exec (SLc pltMove))} \wedge \\
&\quad P \text{ MOVE_TO_PB (exec (SLc pltHalt))} \wedge \\
&\quad P \text{ MOVE_TO_PB (exec (SLc complete))} \wedge \\
&\quad (\forall v_{15}. P \text{ PLT_FORM (exec (ESCc } v_{15}))) \wedge \\
&\quad P \text{ PLT_FORM (exec (SLc pltForm))} \wedge \\
&\quad P \text{ PLT_FORM (exec (SLc pltHalt))} \wedge \\
&\quad P \text{ PLT_FORM (exec (SLc complete))} \wedge \\
&\quad (\forall v_{18}. P \text{ PLT_MOVE (exec (ESCc } v_{18}))) \wedge \\
&\quad P \text{ PLT_MOVE (exec (SLc pltForm))} \wedge \\
&\quad P \text{ PLT_MOVE (exec (SLc pltMove))} \wedge \\
&\quad P \text{ PLT_MOVE (exec (SLc complete))} \wedge \\
&\quad (\forall v_{21}. P \text{ PLT_HALT (exec (ESCc } v_{21}))) \wedge \\
&\quad P \text{ PLT_HALT (exec (SLc pltForm))} \wedge \\
&\quad P \text{ PLT_HALT (exec (SLc pltMove))} \wedge \\
&\quad P \text{ PLT_HALT (exec (SLc pltHalt))} \wedge \\
&\quad (\forall v_{23}. P \text{ COMPLETE (exec } v_{23})) \Rightarrow \\
&\quad \forall v \text{ } v_1. P \text{ } v \text{ } v_1
\end{aligned}$$

[moveToPBOut_def]

$$\begin{aligned}
&\vdash (\text{moveToPBOut MOVE_TO_PB (exec (SLc pltForm))} = \text{PLTForm}) \wedge \\
&\quad (\text{moveToPBOut MOVE_TO_PB (exec (SLc incomplete))} = \text{MoveToPB}) \wedge \\
&\quad (\text{moveToPBOut PLT_FORM (exec (SLc pltMove))} = \text{PLTMove}) \wedge
\end{aligned}$$

$(\text{moveToPBOut PLT_FORM (exec (SLc incomplete))} = \text{PLTForm}) \wedge$
 $(\text{moveToPBOut PLT_MOVE (exec (SLc pltHalt))} = \text{PLTHalt}) \wedge$
 $(\text{moveToPBOut PLT_MOVE (exec (SLc incomplete))} = \text{PLTMove}) \wedge$
 $(\text{moveToPBOut PLT_HALT (exec (SLc complete))} = \text{Complete}) \wedge$
 $(\text{moveToPBOut PLT_HALT (exec (SLc incomplete))} = \text{PLTHalt}) \wedge$
 $(\text{moveToPBOut } s \text{ (trap (SLc cmd))} = \text{unAuthorized}) \wedge$
 $(\text{moveToPBOut } s \text{ (discard (SLc cmd))} = \text{unAuthenticated})$

[moveToPBOut_ind]

$\vdash \forall P.$
 $P \text{ MOVE_TO_PB (exec (SLc pltForm))} \wedge$
 $P \text{ MOVE_TO_PB (exec (SLc incomplete))} \wedge$
 $P \text{ PLT_FORM (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_FORM (exec (SLc incomplete))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltHalt))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc incomplete))} \wedge$
 $P \text{ PLT_HALT (exec (SLc complete))} \wedge$
 $P \text{ PLT_HALT (exec (SLc incomplete))} \wedge$
 $(\forall s \text{ cmd. } P \text{ } s \text{ (trap (SLc cmd))}) \wedge$
 $(\forall s \text{ cmd. } P \text{ } s \text{ (discard (SLc cmd))}) \wedge$
 $(\forall s \text{ } v_6. P \text{ } s \text{ (discard (ESCc } v_6))) \wedge$
 $(\forall s \text{ } v_9. P \text{ } s \text{ (trap (ESCc } v_9))) \wedge$
 $(\forall v_{12}. P \text{ MOVE_TO_PB (exec (ESCc } v_{12}))) \wedge$
 $P \text{ MOVE_TO_PB (exec (SLc pltMove))} \wedge$
 $P \text{ MOVE_TO_PB (exec (SLc pltHalt))} \wedge$
 $P \text{ MOVE_TO_PB (exec (SLc complete))} \wedge$
 $(\forall v_{15}. P \text{ PLT_FORM (exec (ESCc } v_{15}))) \wedge$
 $P \text{ PLT_FORM (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_FORM (exec (SLc pltHalt))} \wedge$
 $P \text{ PLT_FORM (exec (SLc complete))} \wedge$
 $(\forall v_{18}. P \text{ PLT_MOVE (exec (ESCc } v_{18}))) \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_MOVE (exec (SLc complete))} \wedge$
 $(\forall v_{21}. P \text{ PLT_HALT (exec (ESCc } v_{21}))) \wedge$
 $P \text{ PLT_HALT (exec (SLc pltForm))} \wedge$
 $P \text{ PLT_HALT (exec (SLc pltMove))} \wedge$
 $P \text{ PLT_HALT (exec (SLc pltHalt))} \wedge$
 $(\forall v_{23}. P \text{ COMPLETE (exec } v_{23})) \Rightarrow$
 $\forall v \text{ } v_1. P \text{ } v \text{ } v_1$

[PlatoonLeader_exec_slCommand_justified_thm]

$\vdash \forall NS \text{ Out } M \text{ } O_i \text{ } O_s.$
 $\text{TR } (M, O_i, O_s) \text{ (exec (SLc slCommand))}$
 $(\text{CFG authTestMoveToPB ssmMoveToPBStateInterp}$
 $\quad (\text{secContextMoveToPB slCommand})$
 $\quad (\text{Name PlatoonLeader says prop (SOME (SLc slCommand))} ::$
 $\quad \quad \text{ins) } s \text{ outs})$
 $\quad (\text{CFG authTestMoveToPB ssmMoveToPBStateInterp}$

```

(secContextMoveToPB slCommand) ins
(NS s (exec (SLc slCommand)))
(Out s (exec (SLc slCommand))::outs))  $\iff$ 
authTestMoveToPB
(Name PlatoonLeader says prop (SOME (SLc slCommand)))  $\wedge$ 
CFGInterpret (M, Oi, Os)
(CFG authTestMoveToPB ssmMoveToPBStateInterp
(secContextMoveToPB slCommand)
(Name PlatoonLeader says prop (SOME (SLc slCommand))::
ins) s outs)  $\wedge$ 
(M, Oi, Os) sat prop (SOME (SLc slCommand))

```

[PlatoonLeader_slCommand_lemma]

```

 $\vdash$  CFGInterpret (M, Oi, Os)
(CFG authTestMoveToPB ssmMoveToPBStateInterp
(secContextMoveToPB slCommand)
(Name PlatoonLeader says prop (SOME (SLc slCommand))::
ins) s outs)  $\Rightarrow$ 
(M, Oi, Os) sat prop (SOME (SLc slCommand))

```

15 MoveToPBType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

15.1 Datatypes

slCommand = pltForm | pltMove | pltHalt | complete | incomplete

slOutput = MoveToPB | PLTForm | PLTMove | PLTHalt | Complete
 | unauthorized | unAuthenticated

slState = MOVE_TO_PB | PLT_FORM | PLT_MOVE | PLT_HALT | COMPLETE

stateRole = PlatoonLeader

15.2 Theorems

[slCommand_distinct_clauses]

```

 $\vdash$  pltForm  $\neq$  pltMove  $\wedge$  pltForm  $\neq$  pltHalt  $\wedge$  pltForm  $\neq$  complete  $\wedge$ 
pltForm  $\neq$  incomplete  $\wedge$  pltMove  $\neq$  pltHalt  $\wedge$ 
pltMove  $\neq$  complete  $\wedge$  pltMove  $\neq$  incomplete  $\wedge$ 
pltHalt  $\neq$  complete  $\wedge$  pltHalt  $\neq$  incomplete  $\wedge$ 
complete  $\neq$  incomplete

```

[slOutput_distinct_clauses]

$\vdash \text{MoveToPB} \neq \text{PLTForm} \wedge \text{MoveToPB} \neq \text{PLTMove} \wedge$
 $\text{MoveToPB} \neq \text{PLTHalt} \wedge \text{MoveToPB} \neq \text{Complete} \wedge$
 $\text{MoveToPB} \neq \text{unAuthorized} \wedge \text{MoveToPB} \neq \text{unAuthenticated} \wedge$
 $\text{PLTForm} \neq \text{PLTMove} \wedge \text{PLTForm} \neq \text{PLTHalt} \wedge \text{PLTForm} \neq \text{Complete} \wedge$
 $\text{PLTForm} \neq \text{unAuthorized} \wedge \text{PLTForm} \neq \text{unAuthenticated} \wedge$
 $\text{PLTMove} \neq \text{PLTHalt} \wedge \text{PLTMove} \neq \text{Complete} \wedge$
 $\text{PLTMove} \neq \text{unAuthorized} \wedge \text{PLTMove} \neq \text{unAuthenticated} \wedge$
 $\text{PLTHalt} \neq \text{Complete} \wedge \text{PLTHalt} \neq \text{unAuthorized} \wedge$
 $\text{PLTHalt} \neq \text{unAuthenticated} \wedge \text{Complete} \neq \text{unAuthorized} \wedge$
 $\text{Complete} \neq \text{unAuthenticated} \wedge \text{unAuthorized} \neq \text{unAuthenticated}$

[slState_distinct_clauses]

$\vdash \text{MOVE_TO_PB} \neq \text{PLT_FORM} \wedge \text{MOVE_TO_PB} \neq \text{PLT_MOVE} \wedge$
 $\text{MOVE_TO_PB} \neq \text{PLT_HALT} \wedge \text{MOVE_TO_PB} \neq \text{COMPLETE} \wedge$
 $\text{PLT_FORM} \neq \text{PLT_MOVE} \wedge \text{PLT_FORM} \neq \text{PLT_HALT} \wedge$
 $\text{PLT_FORM} \neq \text{COMPLETE} \wedge \text{PLT_MOVE} \neq \text{PLT_HALT} \wedge$
 $\text{PLT_MOVE} \neq \text{COMPLETE} \wedge \text{PLT_HALT} \neq \text{COMPLETE}$

16 ssmPlanPB Theory

Built: 10 June 2018

Parent Theories: PlanPBDef, ssm

16.1 Theorems

[inputOK_def]

$\vdash (\text{inputOK} (\text{Name PlatoonLeader says prop } cmd) \iff T) \wedge$
 $(\text{inputOK} (\text{Name PlatoonSergeant says prop } cmd) \iff T) \wedge$
 $(\text{inputOK } TT \iff F) \wedge (\text{inputOK } FF \iff F) \wedge$
 $(\text{inputOK} (\text{prop } v) \iff F) \wedge (\text{inputOK} (\text{notf } v_1) \iff F) \wedge$
 $(\text{inputOK} (v_2 \text{ andf } v_3) \iff F) \wedge (\text{inputOK} (v_4 \text{ orf } v_5) \iff F) \wedge$
 $(\text{inputOK} (v_6 \text{ impf } v_7) \iff F) \wedge (\text{inputOK} (v_8 \text{ eqf } v_9) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } TT) \iff F) \wedge (\text{inputOK} (v_{10} \text{ says } FF) \iff F) \wedge$
 $(\text{inputOK} (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66}) \iff F) \wedge$
 $(\text{inputOK} (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says notf } v_{67}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } (v_{68} \text{ andf } v_{69})) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } (v_{70} \text{ orf } v_{71})) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } (v_{72} \text{ impf } v_{73})) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75})) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } v_{76} \text{ says } v_{77}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } v_{80} \text{ controls } v_{81}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } v_{85} \text{ domi } v_{86}) \iff F) \wedge$
 $(\text{inputOK} (v_{10} \text{ says } v_{87} \text{ eqi } v_{88}) \iff F) \wedge$

$(\text{inputOK } (v_{10} \text{ says } v_{89} \text{ doms } v_{90}) \iff F) \wedge$
 $(\text{inputOK } (v_{10} \text{ says } v_{91} \text{ eqs } v_{92}) \iff F) \wedge$
 $(\text{inputOK } (v_{10} \text{ says } v_{93} \text{ eqn } v_{94}) \iff F) \wedge$
 $(\text{inputOK } (v_{10} \text{ says } v_{95} \text{ lte } v_{96}) \iff F) \wedge$
 $(\text{inputOK } (v_{10} \text{ says } v_{97} \text{ lt } v_{98}) \iff F) \wedge$
 $(\text{inputOK } (v_{12} \text{ speaks_for } v_{13}) \iff F) \wedge$
 $(\text{inputOK } (v_{14} \text{ controls } v_{15}) \iff F) \wedge$
 $(\text{inputOK } (\text{reps } v_{16} \ v_{17} \ v_{18}) \iff F) \wedge$
 $(\text{inputOK } (v_{19} \text{ domi } v_{20}) \iff F) \wedge$
 $(\text{inputOK } (v_{21} \text{ eqi } v_{22}) \iff F) \wedge$
 $(\text{inputOK } (v_{23} \text{ doms } v_{24}) \iff F) \wedge$
 $(\text{inputOK } (v_{25} \text{ eqs } v_{26}) \iff F) \wedge (\text{inputOK } (v_{27} \text{ eqn } v_{28}) \iff F) \wedge$
 $(\text{inputOK } (v_{29} \text{ lte } v_{30}) \iff F) \wedge (\text{inputOK } (v_{31} \text{ lt } v_{32}) \iff F)$

[inputOK_ind]

$\vdash \forall P.$

$(\forall \text{cmd}. P (\text{Name PlatoonLeader says prop cmd})) \wedge$
 $(\forall \text{cmd}. P (\text{Name PlatoonSergeant says prop cmd})) \wedge P \text{ TT} \wedge$
 $P \text{ FF} \wedge (\forall v. P (\text{prop } v)) \wedge (\forall v_1. P (\text{notf } v_1)) \wedge$
 $(\forall v_2 \ v_3. P (v_2 \text{ andf } v_3)) \wedge (\forall v_4 \ v_5. P (v_4 \text{ orf } v_5)) \wedge$
 $(\forall v_6 \ v_7. P (v_6 \text{ impf } v_7)) \wedge (\forall v_8 \ v_9. P (v_8 \text{ eqf } v_9)) \wedge$
 $(\forall v_{10}. P (v_{10} \text{ says TT})) \wedge (\forall v_{10}. P (v_{10} \text{ says FF})) \wedge$
 $(\forall v_{133} \ v_{134} \ v_{66}. P (v_{133} \text{ meet } v_{134} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{135} \ v_{136} \ v_{66}. P (v_{135} \text{ quoting } v_{136} \text{ says prop } v_{66})) \wedge$
 $(\forall v_{10} \ v_{67}. P (v_{10} \text{ says notf } v_{67})) \wedge$
 $(\forall v_{10} \ v_{68} \ v_{69}. P (v_{10} \text{ says } (v_{68} \text{ andf } v_{69}))) \wedge$
 $(\forall v_{10} \ v_{70} \ v_{71}. P (v_{10} \text{ says } (v_{70} \text{ orf } v_{71}))) \wedge$
 $(\forall v_{10} \ v_{72} \ v_{73}. P (v_{10} \text{ says } (v_{72} \text{ impf } v_{73}))) \wedge$
 $(\forall v_{10} \ v_{74} \ v_{75}. P (v_{10} \text{ says } (v_{74} \text{ eqf } v_{75}))) \wedge$
 $(\forall v_{10} \ v_{76} \ v_{77}. P (v_{10} \text{ says } v_{76} \text{ says } v_{77})) \wedge$
 $(\forall v_{10} \ v_{78} \ v_{79}. P (v_{10} \text{ says } v_{78} \text{ speaks_for } v_{79})) \wedge$
 $(\forall v_{10} \ v_{80} \ v_{81}. P (v_{10} \text{ says } v_{80} \text{ controls } v_{81})) \wedge$
 $(\forall v_{10} \ v_{82} \ v_{83} \ v_{84}. P (v_{10} \text{ says reps } v_{82} \ v_{83} \ v_{84})) \wedge$
 $(\forall v_{10} \ v_{85} \ v_{86}. P (v_{10} \text{ says } v_{85} \text{ domi } v_{86})) \wedge$
 $(\forall v_{10} \ v_{87} \ v_{88}. P (v_{10} \text{ says } v_{87} \text{ eqi } v_{88})) \wedge$
 $(\forall v_{10} \ v_{89} \ v_{90}. P (v_{10} \text{ says } v_{89} \text{ doms } v_{90})) \wedge$
 $(\forall v_{10} \ v_{91} \ v_{92}. P (v_{10} \text{ says } v_{91} \text{ eqs } v_{92})) \wedge$
 $(\forall v_{10} \ v_{93} \ v_{94}. P (v_{10} \text{ says } v_{93} \text{ eqn } v_{94})) \wedge$
 $(\forall v_{10} \ v_{95} \ v_{96}. P (v_{10} \text{ says } v_{95} \text{ lte } v_{96})) \wedge$
 $(\forall v_{10} \ v_{97} \ v_{98}. P (v_{10} \text{ says } v_{97} \text{ lt } v_{98})) \wedge$
 $(\forall v_{12} \ v_{13}. P (v_{12} \text{ speaks_for } v_{13})) \wedge$
 $(\forall v_{14} \ v_{15}. P (v_{14} \text{ controls } v_{15})) \wedge$
 $(\forall v_{16} \ v_{17} \ v_{18}. P (\text{reps } v_{16} \ v_{17} \ v_{18})) \wedge$
 $(\forall v_{19} \ v_{20}. P (v_{19} \text{ domi } v_{20})) \wedge$
 $(\forall v_{21} \ v_{22}. P (v_{21} \text{ eqi } v_{22})) \wedge$
 $(\forall v_{23} \ v_{24}. P (v_{23} \text{ doms } v_{24})) \wedge$
 $(\forall v_{25} \ v_{26}. P (v_{25} \text{ eqs } v_{26})) \wedge (\forall v_{27} \ v_{28}. P (v_{27} \text{ eqn } v_{28})) \wedge$
 $(\forall v_{29} \ v_{30}. P (v_{29} \text{ lte } v_{30})) \wedge (\forall v_{31} \ v_{32}. P (v_{31} \text{ lt } v_{32})) \Rightarrow$
 $\forall v. P \ v$

[planPBNS_def]

```

⊢ (planPBNS WARN0 (exec x) =
  if
    (getRecon x = [SOME (SLc (PL recon))]) ∧
    (getTentativePlan x = [SOME (SLc (PL tentativePlan))]) ∧
    (getReport x = [SOME (SLc (PL report1))]) ∧
    (getInitMove x = [SOME (SLc (PSG initiateMovement))])
  then
    REPORT1
  else WARN0) ∧
(planPBNS PLAN_PB (exec x) =
  if getPlCom x = receiveMission then RECEIVE_MISSION
  else PLAN_PB) ∧
(planPBNS RECEIVE_MISSION (exec x) =
  if getPlCom x = warn0 then WARN0 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 WARN0 (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 WARN0 (exec x) =
  if
    (getRecon x = [SOME (SLc (PL recon))]) ∧
    (getTentativePlan 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]

$\vdash \forall P.$

$$\begin{aligned}
& (\forall x. P \text{ WARNO (exec } x)) \wedge (\forall x. P \text{ PLAN_PB (exec } x)) \wedge \\
& (\forall x. P \text{ RECEIVE_MISSION (exec } x)) \wedge \\
& (\forall x. P \text{ REPORT1 (exec } x)) \wedge (\forall x. P \text{ COMPLETE_PLAN (exec } x)) \wedge \\
& (\forall x. P \text{ OPOID (exec } x)) \wedge (\forall x. P \text{ SUPERVISE (exec } x)) \wedge \\
& (\forall x. P \text{ REPORT2 (exec } x)) \wedge (\forall s \ v_0. P \ s \ (\text{trap } v_0)) \wedge \\
& (\forall s \ v_1. P \ s \ (\text{discard } v_1)) \wedge \\
& (\forall v_6. P \text{ TENTATIVE_PLAN (exec } v_6)) \wedge \\
& (\forall v_7. P \text{ INITIATE_MOVEMENT (exec } v_7)) \wedge \\
& (\forall v_8. P \text{ RECON (exec } v_8)) \wedge (\forall v_9. P \text{ COMPLETE (exec } v_9)) \Rightarrow \\
& \forall v \ v_1. P \ v \ v_1
\end{aligned}$$

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_justified_lemma]

$\vdash s \neq \text{WARNO} \Rightarrow$
 $plCommand \neq \text{invalidPlCommand} \Rightarrow$
 $plCommand \neq \text{report1} \Rightarrow$
 $\forall NS \ Out \ M \ Oi \ Os.$
 $\text{TR } (M, Oi, Os)$
 $(\text{exec}$
 $\quad (\text{inputList}$
 $\quad \quad [\text{Name PlatoonLeader says}$
 $\quad \quad \quad \text{prop (SOME (SLc (PL } plCommand)))]])$
 $\quad (\text{CFG inputOK secContext secContextNull}$
 $\quad \quad ([\text{Name PlatoonLeader says}$
 $\quad \quad \quad \text{prop (SOME (SLc (PL } plCommand)))]::ins) \ s \ outs)$

```

(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))  $\iff$ 
authenticationTest inputOK
  [Name PlatoonLeader says
    prop (SOME (SLc (PL plCommand)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG inputOK secContext secContextNull
    ([Name PlatoonLeader says
      prop (SOME (SLc (PL plCommand)))]::ins) s outs)  $\wedge$ 
  (M, Oi, Os) satList
  propCommandList
  [Name PlatoonLeader says
    prop (SOME (SLc (PL plCommand)))]

```

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_justified_thm]

```

 $\vdash s \neq \text{WARNO} \Rightarrow$ 
  plCommand  $\neq$  invalidPlCommand  $\Rightarrow$ 
  plCommand  $\neq$  report1  $\Rightarrow$ 
 $\forall NS \text{ Out } M \text{ Oi } Os.$ 
  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))  $\iff$ 
authenticationTest inputOK
  [Name PlatoonLeader says
    prop (SOME (SLc (PL plCommand)))]  $\wedge$ 
CFGInterpret (M, Oi, Os)
  (CFG inputOK secContext secContextNull
    ([Name PlatoonLeader says
      prop (SOME (SLc (PL plCommand)))]::ins) s outs)  $\wedge$ 
  (M, Oi, Os) satList [prop (SOME (SLc (PL plCommand)))]

```

[PlatoonLeader_notWARNO_notreport1_exec_plCommand_lemma]

```

 $\vdash s \neq \text{WARNO} \Rightarrow$ 
  plCommand  $\neq$  invalidPlCommand  $\Rightarrow$ 
  plCommand  $\neq$  report1  $\Rightarrow$ 

```

$\forall M \ O_i \ O_s.$
 CFGInterpret (M, O_i, O_s)
 (CFG inputOK secContext secContextNull
 ([Name PlatoonLeader says
 prop (SOME (SLc (PL $plCommand$))))] $::ins$) $s \ outs$) \Rightarrow
 (M, O_i, O_s) satList
 propCommandList
 [Name PlatoonLeader says
 prop (SOME (SLc (PL $plCommand$)))]

[PlatoonLeader_psgCommand_notDiscard_thm]

$\vdash \forall NS \ Out \ M \ O_i \ O_s.$
 \neg TR (M, O_i, O_s) (discard [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 [SOME (SLc (PSG $psgCommand$))]))
 ($Out \ s$ (discard [SOME (SLc (PSG $psgCommand$))])) $::$
 $outs$))

[PlatoonLeader_trap_psgCommand_justified_lemma]

$\vdash \forall NS \ Out \ M \ O_i \ O_s.$
 TR (M, O_i, O_s)
 (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 $psgCommand$)))]))) $::$
 $outs$)) \iff
 authenticationTest inputOK
 [Name PlatoonLeader says
 prop (SOME (SLc (PSG $psgCommand$)))] \wedge
 CFGInterpret (M, O_i, O_s)
 (CFG inputOK secContext secContextNull
 ([Name PlatoonLeader says

prop (SOME (SLc (PSG psgCommand))))]::ins) s outs) \wedge
 (M, Oi, Os) sat prop NONE

[PlatoonLeader_trap_psgCommand_lemma]

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

[PlatoonLeader_WARNO_exec_report1_justified_lemma]

$\vdash \forall NS \ Out \ M \ Oi \ Os.$
 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))  $\iff$ 
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)))]  $\wedge$ 
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) WARN0 outs)  $\wedge$ 
( $M, Oi, Os$ ) 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_WARN0_exec_report1_justified_thm]
 $\vdash \forall NS \text{ Out } M \text{ } Oi \text{ } Os.$ 
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) WARNNO outs)
(CFG inputOK secContext secContextNull ins
  (NS WARNNO
    (exec
      [SOME (SLc (PL recon));
       SOME (SLc (PL tentativePlan));
       SOME (SLc (PSG initiateMovement));
       SOME (SLc (PL report1))]))
  (Out WARNNO
    (exec
      [SOME (SLc (PL recon));
       SOME (SLc (PL tentativePlan));
       SOME (SLc (PSG initiateMovement));
       SOME (SLc (PL report1))]::outs))  $\iff$ 
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)))]  $\wedge$ 
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) WARNNO outs)  $\wedge$ 
(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_WARNNO_exec_report1_lemma]

$\vdash \forall M \ Oi \ Os.$

```

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) WARN0 outs)  $\Rightarrow$ 
(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]

```

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

```

[PlatoonSergeant_trap_plCommand_justified_thm]

```

 $\vdash \forall NS \text{ Out } M \text{ Oi } Os.$ 
TR (M, Oi, Os) (trap [SOME (SLc (PL plCommand))])
  (CFG inputOK secContext secContextNull
    ([Name PlatoonSergeant says
     prop (SOME (SLc (PL plCommand)))]::ins) s outs)
  (CFG inputOK secContext secContextNull ins
    (NS s (trap [SOME (SLc (PL plCommand))])))

```

```

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

```

[PlatoonSergeant_trap_plCommand_lemma]

```

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

```

17 PlanPBType Theory

Built: 10 June 2018

Parent Theories: indexedLists, patternMatches

17.1 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

```

17.2 Theorems

[plCommand_distinct_clauses]

$$\begin{aligned}
&\vdash \text{receiveMission} \neq \text{warno} \wedge \text{receiveMission} \neq \text{tentativePlan} \wedge \\
&\quad \text{receiveMission} \neq \text{recon} \wedge \text{receiveMission} \neq \text{report1} \wedge \\
&\quad \text{receiveMission} \neq \text{completePlan} \wedge \text{receiveMission} \neq \text{opoid} \wedge \\
&\quad \text{receiveMission} \neq \text{supervise} \wedge \text{receiveMission} \neq \text{report2} \wedge \\
&\quad \text{receiveMission} \neq \text{complete} \wedge \text{receiveMission} \neq \text{plIncomplete} \wedge \\
&\quad \text{receiveMission} \neq \text{invalidPlCommand} \wedge \text{warno} \neq \text{tentativePlan} \wedge \\
&\quad \text{warno} \neq \text{recon} \wedge \text{warno} \neq \text{report1} \wedge \text{warno} \neq \text{completePlan} \wedge \\
&\quad \text{warno} \neq \text{opoid} \wedge \text{warno} \neq \text{supervise} \wedge \text{warno} \neq \text{report2} \wedge \\
&\quad \text{warno} \neq \text{complete} \wedge \text{warno} \neq \text{plIncomplete} \wedge \\
&\quad \text{warno} \neq \text{invalidPlCommand} \wedge \text{tentativePlan} \neq \text{recon} \wedge \\
&\quad \text{tentativePlan} \neq \text{report1} \wedge \text{tentativePlan} \neq \text{completePlan} \wedge \\
&\quad \text{tentativePlan} \neq \text{opoid} \wedge \text{tentativePlan} \neq \text{supervise} \wedge \\
&\quad \text{tentativePlan} \neq \text{report2} \wedge \text{tentativePlan} \neq \text{complete} \wedge \\
&\quad \text{tentativePlan} \neq \text{plIncomplete} \wedge \\
&\quad \text{tentativePlan} \neq \text{invalidPlCommand} \wedge \text{recon} \neq \text{report1} \wedge \\
&\quad \text{recon} \neq \text{completePlan} \wedge \text{recon} \neq \text{opoid} \wedge \text{recon} \neq \text{supervise} \wedge \\
&\quad \text{recon} \neq \text{report2} \wedge \text{recon} \neq \text{complete} \wedge \text{recon} \neq \text{plIncomplete} \wedge \\
&\quad \text{recon} \neq \text{invalidPlCommand} \wedge \text{report1} \neq \text{completePlan} \wedge \\
&\quad \text{report1} \neq \text{opoid} \wedge \text{report1} \neq \text{supervise} \wedge \text{report1} \neq \text{report2} \wedge \\
&\quad \text{report1} \neq \text{complete} \wedge \text{report1} \neq \text{plIncomplete} \wedge \\
&\quad \text{report1} \neq \text{invalidPlCommand} \wedge \text{completePlan} \neq \text{opoid} \wedge \\
&\quad \text{completePlan} \neq \text{supervise} \wedge \text{completePlan} \neq \text{report2} \wedge \\
&\quad \text{completePlan} \neq \text{complete} \wedge \text{completePlan} \neq \text{plIncomplete} \wedge \\
&\quad \text{completePlan} \neq \text{invalidPlCommand} \wedge \text{opoid} \neq \text{supervise} \wedge \\
&\quad \text{opoid} \neq \text{report2} \wedge \text{opoid} \neq \text{complete} \wedge \text{opoid} \neq \text{plIncomplete} \wedge \\
&\quad \text{opoid} \neq \text{invalidPlCommand} \wedge \text{supervise} \neq \text{report2} \wedge \\
&\quad \text{supervise} \neq \text{complete} \wedge \text{supervise} \neq \text{plIncomplete} \wedge \\
&\quad \text{supervise} \neq \text{invalidPlCommand} \wedge \text{report2} \neq \text{complete} \wedge \\
&\quad \text{report2} \neq \text{plIncomplete} \wedge \text{report2} \neq \text{invalidPlCommand} \wedge \\
&\quad \text{complete} \neq \text{plIncomplete} \wedge \text{complete} \neq \text{invalidPlCommand} \wedge \\
&\quad \text{plIncomplete} \neq \text{invalidPlCommand}
\end{aligned}$$

[psgCommand_distinct_clauses]

$$\begin{aligned}
&\vdash \text{initiateMovement} \neq \text{psgIncomplete} \wedge \\
&\quad \text{initiateMovement} \neq \text{invalidPsgCommand} \wedge \\
&\quad \text{psgIncomplete} \neq \text{invalidPsgCommand}
\end{aligned}$$

[slCommand_distinct_clauses]

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

[slCommand_one_one]

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

[slOutput_distinct_clauses]

$$\begin{aligned}
&\vdash \text{PlanPB} \neq \text{ReceiveMission} \wedge \text{PlanPB} \neq \text{Warno} \wedge \\
&\text{PlanPB} \neq \text{TentativePlan} \wedge \text{PlanPB} \neq \text{InitiateMovement} \wedge \\
&\text{PlanPB} \neq \text{Recon} \wedge \text{PlanPB} \neq \text{Report1} \wedge \text{PlanPB} \neq \text{CompletePlan} \wedge \\
&\text{PlanPB} \neq \text{Opoid} \wedge \text{PlanPB} \neq \text{Supervise} \wedge \text{PlanPB} \neq \text{Report2} \wedge \\
&\text{PlanPB} \neq \text{Complete} \wedge \text{PlanPB} \neq \text{unAuthenticated} \wedge \\
&\text{PlanPB} \neq \text{unAuthorized} \wedge \text{ReceiveMission} \neq \text{Warno} \wedge \\
&\text{ReceiveMission} \neq \text{TentativePlan} \wedge \\
&\text{ReceiveMission} \neq \text{InitiateMovement} \wedge \text{ReceiveMission} \neq \text{Recon} \wedge \\
&\text{ReceiveMission} \neq \text{Report1} \wedge \text{ReceiveMission} \neq \text{CompletePlan} \wedge \\
&\text{ReceiveMission} \neq \text{Opoid} \wedge \text{ReceiveMission} \neq \text{Supervise} \wedge \\
&\text{ReceiveMission} \neq \text{Report2} \wedge \text{ReceiveMission} \neq \text{Complete} \wedge \\
&\text{ReceiveMission} \neq \text{unAuthenticated} \wedge \\
&\text{ReceiveMission} \neq \text{unAuthorized} \wedge \text{Warno} \neq \text{TentativePlan} \wedge \\
&\text{Warno} \neq \text{InitiateMovement} \wedge \text{Warno} \neq \text{Recon} \wedge \text{Warno} \neq \text{Report1} \wedge \\
&\text{Warno} \neq \text{CompletePlan} \wedge \text{Warno} \neq \text{Opoid} \wedge \text{Warno} \neq \text{Supervise} \wedge \\
&\text{Warno} \neq \text{Report2} \wedge \text{Warno} \neq \text{Complete} \wedge \\
&\text{Warno} \neq \text{unAuthenticated} \wedge \text{Warno} \neq \text{unAuthorized} \wedge \\
&\text{TentativePlan} \neq \text{InitiateMovement} \wedge \text{TentativePlan} \neq \text{Recon} \wedge \\
&\text{TentativePlan} \neq \text{Report1} \wedge \text{TentativePlan} \neq \text{CompletePlan} \wedge \\
&\text{TentativePlan} \neq \text{Opoid} \wedge \text{TentativePlan} \neq \text{Supervise} \wedge \\
&\text{TentativePlan} \neq \text{Report2} \wedge \text{TentativePlan} \neq \text{Complete} \wedge \\
&\text{TentativePlan} \neq \text{unAuthenticated} \wedge \\
&\text{TentativePlan} \neq \text{unAuthorized} \wedge \text{InitiateMovement} \neq \text{Recon} \wedge \\
&\text{InitiateMovement} \neq \text{Report1} \wedge \\
&\text{InitiateMovement} \neq \text{CompletePlan} \wedge \text{InitiateMovement} \neq \text{Opoid} \wedge \\
&\text{InitiateMovement} \neq \text{Supervise} \wedge \text{InitiateMovement} \neq \text{Report2} \wedge \\
&\text{InitiateMovement} \neq \text{Complete} \wedge \\
&\text{InitiateMovement} \neq \text{unAuthenticated} \wedge \\
&\text{InitiateMovement} \neq \text{unAuthorized} \wedge \text{Recon} \neq \text{Report1} \wedge \\
&\text{Recon} \neq \text{CompletePlan} \wedge \text{Recon} \neq \text{Opoid} \wedge \text{Recon} \neq \text{Supervise} \wedge \\
&\text{Recon} \neq \text{Report2} \wedge \text{Recon} \neq \text{Complete} \wedge \\
&\text{Recon} \neq \text{unAuthenticated} \wedge \text{Recon} \neq \text{unAuthorized} \wedge \\
&\text{Report1} \neq \text{CompletePlan} \wedge \text{Report1} \neq \text{Opoid} \wedge \\
&\text{Report1} \neq \text{Supervise} \wedge \text{Report1} \neq \text{Report2} \wedge \\
&\text{Report1} \neq \text{Complete} \wedge \text{Report1} \neq \text{unAuthenticated} \wedge \\
&\text{Report1} \neq \text{unAuthorized} \wedge \text{CompletePlan} \neq \text{Opoid} \wedge \\
&\text{CompletePlan} \neq \text{Supervise} \wedge \text{CompletePlan} \neq \text{Report2} \wedge \\
&\text{CompletePlan} \neq \text{Complete} \wedge \text{CompletePlan} \neq \text{unAuthenticated} \wedge \\
&\text{CompletePlan} \neq \text{unAuthorized} \wedge \text{Opoid} \neq \text{Supervise} \wedge \\
&\text{Opoid} \neq \text{Report2} \wedge \text{Opoid} \neq \text{Complete} \wedge \\
&\text{Opoid} \neq \text{unAuthenticated} \wedge \text{Opoid} \neq \text{unAuthorized} \wedge \\
&\text{Supervise} \neq \text{Report2} \wedge \text{Supervise} \neq \text{Complete} \wedge \\
&\text{Supervise} \neq \text{unAuthenticated} \wedge \text{Supervise} \neq \text{unAuthorized} \wedge \\
&\text{Report2} \neq \text{Complete} \wedge \text{Report2} \neq \text{unAuthenticated} \wedge \\
&\text{Report2} \neq \text{unAuthorized} \wedge \text{Complete} \neq \text{unAuthenticated} \wedge \\
&\text{Complete} \neq \text{unAuthorized} \wedge \text{unAuthenticated} \neq \text{unAuthorized}
\end{aligned}$$
[slRole_distinct_clauses]

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

[slState_distinct_clauses]

$\vdash \text{PLAN_PB} \neq \text{RECEIVE_MISSION} \wedge \text{PLAN_PB} \neq \text{WARNO} \wedge$
 $\text{PLAN_PB} \neq \text{TENTATIVE_PLAN} \wedge \text{PLAN_PB} \neq \text{INITIATE_MOVEMENT} \wedge$
 $\text{PLAN_PB} \neq \text{RECON} \wedge \text{PLAN_PB} \neq \text{REPORT1} \wedge$
 $\text{PLAN_PB} \neq \text{COMPLETE_PLAN} \wedge \text{PLAN_PB} \neq \text{OPOID} \wedge$
 $\text{PLAN_PB} \neq \text{SUPERVISE} \wedge \text{PLAN_PB} \neq \text{REPORT2} \wedge$
 $\text{PLAN_PB} \neq \text{COMPLETE} \wedge \text{RECEIVE_MISSION} \neq \text{WARNO} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{TENTATIVE_PLAN} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{INITIATE_MOVEMENT} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{RECON} \wedge \text{RECEIVE_MISSION} \neq \text{REPORT1} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{COMPLETE_PLAN} \wedge \text{RECEIVE_MISSION} \neq \text{OPOID} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{SUPERVISE} \wedge \text{RECEIVE_MISSION} \neq \text{REPORT2} \wedge$
 $\text{RECEIVE_MISSION} \neq \text{COMPLETE} \wedge \text{WARNO} \neq \text{TENTATIVE_PLAN} \wedge$
 $\text{WARNO} \neq \text{INITIATE_MOVEMENT} \wedge \text{WARNO} \neq \text{RECON} \wedge \text{WARNO} \neq \text{REPORT1} \wedge$
 $\text{WARNO} \neq \text{COMPLETE_PLAN} \wedge \text{WARNO} \neq \text{OPOID} \wedge \text{WARNO} \neq \text{SUPERVISE} \wedge$
 $\text{WARNO} \neq \text{REPORT2} \wedge \text{WARNO} \neq \text{COMPLETE} \wedge$
 $\text{TENTATIVE_PLAN} \neq \text{INITIATE_MOVEMENT} \wedge \text{TENTATIVE_PLAN} \neq \text{RECON} \wedge$
 $\text{TENTATIVE_PLAN} \neq \text{REPORT1} \wedge \text{TENTATIVE_PLAN} \neq \text{COMPLETE_PLAN} \wedge$
 $\text{TENTATIVE_PLAN} \neq \text{OPOID} \wedge \text{TENTATIVE_PLAN} \neq \text{SUPERVISE} \wedge$
 $\text{TENTATIVE_PLAN} \neq \text{REPORT2} \wedge \text{TENTATIVE_PLAN} \neq \text{COMPLETE} \wedge$
 $\text{INITIATE_MOVEMENT} \neq \text{RECON} \wedge \text{INITIATE_MOVEMENT} \neq \text{REPORT1} \wedge$
 $\text{INITIATE_MOVEMENT} \neq \text{COMPLETE_PLAN} \wedge$
 $\text{INITIATE_MOVEMENT} \neq \text{OPOID} \wedge \text{INITIATE_MOVEMENT} \neq \text{SUPERVISE} \wedge$
 $\text{INITIATE_MOVEMENT} \neq \text{REPORT2} \wedge \text{INITIATE_MOVEMENT} \neq \text{COMPLETE} \wedge$
 $\text{RECON} \neq \text{REPORT1} \wedge \text{RECON} \neq \text{COMPLETE_PLAN} \wedge \text{RECON} \neq \text{OPOID} \wedge$
 $\text{RECON} \neq \text{SUPERVISE} \wedge \text{RECON} \neq \text{REPORT2} \wedge \text{RECON} \neq \text{COMPLETE} \wedge$
 $\text{REPORT1} \neq \text{COMPLETE_PLAN} \wedge \text{REPORT1} \neq \text{OPOID} \wedge$
 $\text{REPORT1} \neq \text{SUPERVISE} \wedge \text{REPORT1} \neq \text{REPORT2} \wedge$
 $\text{REPORT1} \neq \text{COMPLETE} \wedge \text{COMPLETE_PLAN} \neq \text{OPOID} \wedge$
 $\text{COMPLETE_PLAN} \neq \text{SUPERVISE} \wedge \text{COMPLETE_PLAN} \neq \text{REPORT2} \wedge$
 $\text{COMPLETE_PLAN} \neq \text{COMPLETE} \wedge \text{OPOID} \neq \text{SUPERVISE} \wedge$
 $\text{OPOID} \neq \text{REPORT2} \wedge \text{OPOID} \neq \text{COMPLETE} \wedge \text{SUPERVISE} \neq \text{REPORT2} \wedge$
 $\text{SUPERVISE} \neq \text{COMPLETE} \wedge \text{REPORT2} \neq \text{COMPLETE}$

18 PlanPBDef Theory

Built: 10 June 2018

Parent Theories: PlanPBType, aclfoundation, OMNIType

18.1 Definitions

[PL_notWARNO_Auth_def]

$\vdash \forall cmd.$
 $\text{PL_notWARNO_Auth } cmd =$
if $cmd = \text{report1}$ **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))]) ∧
        (getTentativePlan 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]

```

18.2 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) ∧

```

$$\begin{aligned}
& (\forall xs \ v_{32} \ v_{31}. \\
& \quad \text{getInitMove } (v_{31} \text{ lte } v_{32} :: xs) = \text{getInitMove } xs) \wedge \\
& \quad \forall xs \ v_{34} \ v_{33}. \text{getInitMove } (v_{33} \text{ lt } v_{34} :: xs) = \text{getInitMove } xs \\
& [\text{getInitMove_ind}] \\
& \vdash \forall P. \\
& \quad P \ [] \ \wedge \\
& \quad (\forall xs. \\
& \quad \quad P \\
& \quad \quad (\text{Name PlatoonSergeant says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PSG initiateMovement))) :: xs})) \wedge \\
& \quad (\forall xs. P \ xs \Rightarrow P \ (\text{TT} :: xs)) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{FF} :: xs)) \wedge \\
& \quad (\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2 :: xs)) \wedge \\
& \quad (\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3 :: xs)) \wedge \\
& \quad (\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \ \text{andf } v_5 :: xs)) \wedge \\
& \quad (\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \ \text{orf } v_7 :: xs)) \wedge \\
& \quad (\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \ \text{impf } v_9 :: xs)) \wedge \\
& \quad (\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \ \text{eqf } v_{11} :: xs)) \wedge \\
& \quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says TT} :: xs)) \wedge \\
& \quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says FF} :: xs)) \wedge \\
& \quad (\forall v_{134} \ xs. P \ xs \Rightarrow P \ (\text{Name } v_{134} \ \text{says prop NONE} :: xs)) \wedge \\
& \quad (\forall v_{144} \ xs. \\
& \quad \quad P \ xs \Rightarrow \\
& \quad \quad P \ (\text{Name PlatoonLeader says prop (SOME } v_{144}) :: xs)) \wedge \\
& \quad (\forall v_{146} \ xs. \\
& \quad \quad P \ xs \Rightarrow \\
& \quad \quad P \\
& \quad \quad (\text{Name PlatoonSergeant says prop (SOME (ESCc } v_{146})) :: \\
& \quad \quad \quad xs)) \wedge \\
& \quad (\forall v_{150} \ xs. \\
& \quad \quad P \ xs \Rightarrow \\
& \quad \quad P \\
& \quad \quad (\text{Name PlatoonSergeant says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL } v_{150})) :: xs)) \wedge \\
& \quad (\forall xs. \\
& \quad \quad P \ xs \Rightarrow \\
& \quad \quad P \\
& \quad \quad (\text{Name PlatoonSergeant says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PSG psgIncomplete))) :: xs)) \wedge \\
& \quad (\forall xs. \\
& \quad \quad P \ xs \Rightarrow \\
& \quad \quad P \\
& \quad \quad (\text{Name PlatoonSergeant says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PSG invalidPsgCommand))) :: xs)) \wedge \\
& \quad (\forall v_{135} \ v_{136} \ v_{68} \ xs. \\
& \quad \quad P \ xs \Rightarrow P \ (v_{135} \ \text{meet } v_{136} \ \text{says prop } v_{68} :: xs)) \wedge \\
& \quad (\forall v_{137} \ v_{138} \ v_{68} \ xs. \\
& \quad \quad P \ xs \Rightarrow P \ (v_{137} \ \text{quoting } v_{138} \ \text{says prop } v_{68} :: xs)) \wedge \\
& \quad (\forall v_{12} \ v_{69} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says notf } v_{69} :: xs)) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_{12} v_{70} v_{71} xs. P xs \Rightarrow P (v_{12} \text{ says } (v_{70} \text{ andf } v_{71}) :: xs)) \wedge \\
& (\forall v_{12} v_{72} v_{73} xs. P xs \Rightarrow P (v_{12} \text{ says } (v_{72} \text{ orf } v_{73}) :: xs)) \wedge \\
& (\forall v_{12} v_{74} v_{75} xs. P xs \Rightarrow P (v_{12} \text{ says } (v_{74} \text{ impf } v_{75}) :: xs)) \wedge \\
& (\forall v_{12} v_{76} v_{77} xs. P xs \Rightarrow P (v_{12} \text{ says } (v_{76} \text{ eqf } v_{77}) :: xs)) \wedge \\
& (\forall v_{12} v_{78} v_{79} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{78} \text{ says } v_{79} :: xs)) \wedge \\
& (\forall v_{12} v_{80} v_{81} xs. \\
& \quad P xs \Rightarrow P (v_{12} \text{ says } v_{80} \text{ speaks_for } v_{81} :: xs)) \wedge \\
& (\forall v_{12} v_{82} v_{83} xs. \\
& \quad P xs \Rightarrow P (v_{12} \text{ says } v_{82} \text{ controls } v_{83} :: xs)) \wedge \\
& (\forall v_{12} v_{84} v_{85} v_{86} xs. \\
& \quad P xs \Rightarrow P (v_{12} \text{ says reps } v_{84} v_{85} v_{86} :: xs)) \wedge \\
& (\forall v_{12} v_{87} v_{88} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{87} \text{ domi } v_{88} :: xs)) \wedge \\
& (\forall v_{12} v_{89} v_{90} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{89} \text{ eqi } v_{90} :: xs)) \wedge \\
& (\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]

$$\begin{aligned}
& \vdash (\text{getPlCom } [] = \text{invalidPlCommand}) \wedge \\
& (\forall xs \text{ cmd}. \\
& \quad \text{getPlCom} \\
& \quad \quad (\text{Name PlatoonLeader says prop (SOME (SLc (PL cmd)))}) :: \\
& \quad \quad \quad xs) = \\
& \quad \text{cmd}) \wedge (\forall xs. \text{getPlCom } (\text{TT} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs. \text{getPlCom } (\text{FF} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_2. \text{getPlCom } (\text{prop } v_2 :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_3. \text{getPlCom } (\text{notf } v_3 :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_5 \ v_4. \text{getPlCom } (v_4 \text{ andf } v_5 :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_7 \ v_6. \text{getPlCom } (v_6 \text{ orf } v_7 :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_9 \ v_8. \text{getPlCom } (v_8 \text{ impf } v_9 :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_{11} \ v_{10}. \text{getPlCom } (v_{10} \text{ eqf } v_{11} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_{12}. \text{getPlCom } (v_{12} \text{ says TT} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_{12}. \text{getPlCom } (v_{12} \text{ says FF} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_{134}. \\
& \quad \text{getPlCom } (\text{Name } v_{134} \text{ says prop NONE} :: xs) = \text{getPlCom } xs) \wedge \\
& (\forall xs \ v_{146}.
\end{aligned}$$

```

    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) ∧

```

$(\forall xs \ v_{99} \ v_{12} \ v_{100}.$
 $\quad \text{getPlCom } (v_{12} \text{ says } v_{99} \text{ lt } v_{100}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{15} \ v_{14}.$
 $\quad \text{getPlCom } (v_{14} \text{ speaks_for } v_{15}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{17} \ v_{16}.$
 $\quad \text{getPlCom } (v_{16} \text{ controls } v_{17}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{20} \ v_{19} \ v_{18}.$
 $\quad \text{getPlCom } (\text{reps } v_{18} \ v_{19} \ v_{20}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{22} \ v_{21}.$
 $\quad \text{getPlCom } (v_{21} \text{ domi } v_{22}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{24} \ v_{23}.$
 $\quad \text{getPlCom } (v_{23} \text{ eqi } v_{24}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{26} \ v_{25}.$
 $\quad \text{getPlCom } (v_{25} \text{ doms } v_{26}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{28} \ v_{27}.$
 $\quad \text{getPlCom } (v_{27} \text{ eqs } v_{28}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{30} \ v_{29}.$
 $\quad \text{getPlCom } (v_{29} \text{ eqn } v_{30}::xs) = \text{getPlCom } xs) \wedge$
 $(\forall xs \ v_{32} \ v_{31}.$
 $\quad \text{getPlCom } (v_{31} \text{ lte } v_{32}::xs) = \text{getPlCom } xs) \wedge$
 $\forall xs \ v_{34} \ v_{33}.$
 $\quad \text{getPlCom } (v_{33} \text{ lt } v_{34}::xs) = \text{getPlCom } xs$

[getPlCom_ind]

$\vdash \forall P.$
 $\quad P \ \square \wedge$
 $\quad (\forall \text{cmd } xs.$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonLeader says prop (SOME (SLc (PL cmd)))::}$
 $\quad \quad \quad xs)) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{TT}::xs)) \wedge$
 $\quad (\forall xs. P \ xs \Rightarrow P \ (\text{FF}::xs)) \wedge$
 $\quad (\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2::xs)) \wedge$
 $\quad (\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3::xs)) \wedge$
 $\quad (\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \text{ andf } v_5::xs)) \wedge$
 $\quad (\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \text{ orf } v_7::xs)) \wedge$
 $\quad (\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \text{ impf } v_9::xs)) \wedge$
 $\quad (\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \text{ eqf } v_{11}::xs)) \wedge$
 $\quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says TT}::xs)) \wedge$
 $\quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \text{ says FF}::xs)) \wedge$
 $\quad (\forall v_{134} \ xs. P \ xs \Rightarrow P \ (\text{Name } v_{134} \text{ says prop NONE}::xs)) \wedge$
 $\quad (\forall v_{146} \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonLeader says prop (SOME (ESCc } v_{146}))::$
 $\quad \quad \quad xs)) \wedge$
 $\quad (\forall v_{151} \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonLeader says}$
 $\quad \quad \quad \text{prop (SOME (SLc (PSG } v_{151})))::xs)) \wedge$
 $\quad (\forall v_{144} \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P \ (\text{Name PlatoonSergeant says prop (SOME } v_{144})::xs)) \wedge$
 $\quad (\forall v_{135} \ v_{136} \ v_{68} \ xs.$
 $\quad \quad P \ xs \Rightarrow P \ (v_{135} \text{ meet } v_{136} \text{ says prop } v_{68}::xs)) \wedge$
 $\quad (\forall v_{137} \ v_{138} \ v_{68} \ xs.$

$$\begin{aligned}
& P \text{ } xs \Rightarrow P \text{ } (v137 \text{ quoting } v138 \text{ says prop } v68::xs)) \wedge \\
& (\forall v_{12} \ v_{69} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says notf } v_{69}::xs)) \wedge \\
& (\forall v_{12} \ v_{70} \ v_{71} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } (v_{70} \text{ andf } v_{71})::xs)) \wedge \\
& (\forall v_{12} \ v_{72} \ v_{73} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } (v_{72} \text{ orf } v_{73})::xs)) \wedge \\
& (\forall v_{12} \ v_{74} \ v_{75} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } (v_{74} \text{ impf } v_{75})::xs)) \wedge \\
& (\forall v_{12} \ v_{76} \ v_{77} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } (v_{76} \text{ eqf } v_{77})::xs)) \wedge \\
& (\forall v_{12} \ v_{78} \ v_{79} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{78} \text{ says } v_{79}::xs)) \wedge \\
& (\forall v_{12} \ v_{80} \ v_{81} \ xs. \\
& \quad P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{80} \text{ speaks_for } v_{81}::xs)) \wedge \\
& (\forall v_{12} \ v_{82} \ v_{83} \ xs. \\
& \quad P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{82} \text{ controls } v_{83}::xs)) \wedge \\
& (\forall v_{12} \ v_{84} \ v_{85} \ v_{86} \ xs. \\
& \quad P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says reps } v_{84} \ v_{85} \ v_{86}::xs)) \wedge \\
& (\forall v_{12} \ v_{87} \ v_{88} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{87} \text{ domi } v_{88}::xs)) \wedge \\
& (\forall v_{12} \ v_{89} \ v_{90} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{89} \text{ eqi } v_{90}::xs)) \wedge \\
& (\forall v_{12} \ v_{91} \ v_{92} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{91} \text{ doms } v_{92}::xs)) \wedge \\
& (\forall v_{12} \ v_{93} \ v_{94} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{93} \text{ eqs } v_{94}::xs)) \wedge \\
& (\forall v_{12} \ v_{95} \ v_{96} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{95} \text{ eqn } v_{96}::xs)) \wedge \\
& (\forall v_{12} \ v_{97} \ v_{98} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{97} \text{ lte } v_{98}::xs)) \wedge \\
& (\forall v_{12} \ v_{99} \ v100 \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{12} \text{ says } v_{99} \text{ lt } v100::xs)) \wedge \\
& (\forall v_{14} \ v_{15} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{14} \text{ speaks_for } v_{15}::xs)) \wedge \\
& (\forall v_{16} \ v_{17} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{16} \text{ controls } v_{17}::xs)) \wedge \\
& (\forall v_{18} \ v_{19} \ v_{20} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (reps \ v_{18} \ v_{19} \ v_{20}::xs)) \wedge \\
& (\forall v_{21} \ v_{22} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{21} \text{ domi } v_{22}::xs)) \wedge \\
& (\forall v_{23} \ v_{24} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{23} \text{ eqi } v_{24}::xs)) \wedge \\
& (\forall v_{25} \ v_{26} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{25} \text{ doms } v_{26}::xs)) \wedge \\
& (\forall v_{27} \ v_{28} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{27} \text{ eqs } v_{28}::xs)) \wedge \\
& (\forall v_{29} \ v_{30} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{29} \text{ eqn } v_{30}::xs)) \wedge \\
& (\forall v_{31} \ v_{32} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{31} \text{ lte } v_{32}::xs)) \wedge \\
& (\forall v_{33} \ v_{34} \ xs. \ P \text{ } xs \Rightarrow P \text{ } (v_{33} \text{ lt } v_{34}::xs)) \Rightarrow \\
& \forall v. \ P \ v
\end{aligned}$$

[getPsgCom_def]

$$\begin{aligned}
& \vdash (\text{getPsgCom } [] = \text{invalidPsgCommand}) \wedge \\
& (\forall xs \text{ cmd.} \\
& \quad \text{getPsgCom} \\
& \quad \quad (\text{Name PlatoonSergeant says prop (SOME (SLc (PSG cmd)))}):: \\
& \quad \quad \quad xs) = \\
& \quad \quad \text{cmd}) \wedge (\forall xs. \text{getPsgCom (TT::xs)} = \text{getPsgCom } xs) \wedge \\
& (\forall xs. \text{getPsgCom (FF::xs)} = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_2. \text{getPsgCom (prop } v_2::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_3. \text{getPsgCom (notf } v_3::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_5 \ v_4. \text{getPsgCom (} v_4 \text{ andf } v_5::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_7 \ v_6. \text{getPsgCom (} v_6 \text{ orf } v_7::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_9 \ v_8. \text{getPsgCom (} v_8 \text{ impf } v_9::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_{11} \ v_{10}. \text{getPsgCom (} v_{10} \text{ eqf } v_{11}::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_{12}. \text{getPsgCom (} v_{12} \text{ says TT::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v_{12}. \text{getPsgCom (} v_{12} \text{ says FF::xs) = \text{getPsgCom } xs) \wedge \\
& (\forall xs \ v134.
\end{aligned}$$

```

    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) ∧

```

$(\forall xs \ v_{96} \ v_{95} \ v_{12}. \text{getPsgCom } (v_{12} \text{ says } v_{95} \text{ eqn } v_{96}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{98} \ v_{97} \ v_{12}. \text{getPsgCom } (v_{12} \text{ says } v_{97} \text{ lte } v_{98}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{99} \ v_{12} \ v100. \text{getPsgCom } (v_{12} \text{ says } v_{99} \text{ lt } v100::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{15} \ v_{14}. \text{getPsgCom } (v_{14} \text{ speaks_for } v_{15}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{17} \ v_{16}. \text{getPsgCom } (v_{16} \text{ controls } v_{17}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{20} \ v_{19} \ v_{18}. \text{getPsgCom } (\text{reps } v_{18} \ v_{19} \ v_{20}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{22} \ v_{21}. \text{getPsgCom } (v_{21} \text{ domi } v_{22}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{24} \ v_{23}. \text{getPsgCom } (v_{23} \text{ eqi } v_{24}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{26} \ v_{25}. \text{getPsgCom } (v_{25} \text{ doms } v_{26}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{28} \ v_{27}. \text{getPsgCom } (v_{27} \text{ eqs } v_{28}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{30} \ v_{29}. \text{getPsgCom } (v_{29} \text{ eqn } v_{30}::xs) = \text{getPsgCom } xs) \wedge$
 $(\forall xs \ v_{32} \ v_{31}. \text{getPsgCom } (v_{31} \text{ lte } v_{32}::xs) = \text{getPsgCom } xs) \wedge$
 $\forall xs \ v_{34} \ v_{33}. \text{getPsgCom } (v_{33} \text{ lt } v_{34}::xs) = \text{getPsgCom } xs$

[getPsgCom_ind]

$\vdash \forall P.$
 $\quad P \ [] \wedge$
 $\quad (\forall cmd \ xs.$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonSergeant says}$
 $\quad \quad \quad \text{prop (SOME (SLc (PSG cmd)))::xs)) \wedge$
 $\quad (\forall xs. P \ xs \Rightarrow P \ (\text{TT}::xs)) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{FF}::xs)) \wedge$
 $\quad (\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2::xs)) \wedge$
 $\quad (\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3::xs)) \wedge$
 $\quad (\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \ \text{andf } v_5::xs)) \wedge$
 $\quad (\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \ \text{orf } v_7::xs)) \wedge$
 $\quad (\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \ \text{impf } v_9::xs)) \wedge$
 $\quad (\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \ \text{eqf } v_{11}::xs)) \wedge$
 $\quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says TT}::xs)) \wedge$
 $\quad (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says FF}::xs)) \wedge$
 $\quad (\forall v134 \ xs. P \ xs \Rightarrow P \ (\text{Name } v134 \ \text{says prop NONE}::xs)) \wedge$
 $\quad (\forall v144 \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P \ (\text{Name PlatoonLeader says prop (SOME } v144)::xs)) \wedge$
 $\quad (\forall v146 \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonSergeant says prop (SOME (ESCc } v146)::$
 $\quad \quad \quad xs)) \wedge$
 $\quad (\forall v150 \ xs.$
 $\quad \quad P \ xs \Rightarrow$
 $\quad \quad P$
 $\quad \quad (\text{Name PlatoonSergeant says}$

$$\begin{aligned}
& \text{prop (SOME (SLc (PL v150)))::xs)} \wedge \\
& (\forall v135 v136 v68 xs. \\
& \quad P xs \Rightarrow P (v135 \text{ meet } v136 \text{ says prop } v68::xs)) \wedge \\
& (\forall v137 v138 v68 xs. \\
& \quad P xs \Rightarrow P (v137 \text{ quoting } v138 \text{ says prop } v68::xs)) \wedge \\
& (\forall v12 v69 xs. P xs \Rightarrow P (v12 \text{ says notf } v69::xs)) \wedge \\
& (\forall v12 v70 v71 xs. P xs \Rightarrow P (v12 \text{ says (v70 andf v71)::xs)) \wedge \\
& (\forall v12 v72 v73 xs. P xs \Rightarrow P (v12 \text{ says (v72 orf v73)::xs)) \wedge \\
& (\forall v12 v74 v75 xs. P xs \Rightarrow P (v12 \text{ says (v74 impf v75)::xs)) \wedge \\
& (\forall v12 v76 v77 xs. P xs \Rightarrow P (v12 \text{ says (v76 eqf v77)::xs)) \wedge \\
& (\forall v12 v78 v79 xs. P xs \Rightarrow P (v12 \text{ says } v78 \text{ says } v79::xs)) \wedge \\
& (\forall v12 v80 v81 xs. \\
& \quad P xs \Rightarrow P (v12 \text{ says } v80 \text{ speaks_for } v81::xs)) \wedge \\
& (\forall v12 v82 v83 xs. \\
& \quad P xs \Rightarrow P (v12 \text{ says } v82 \text{ controls } v83::xs)) \wedge \\
& (\forall v12 v84 v85 v86 xs. \\
& \quad P xs \Rightarrow P (v12 \text{ says reps } v84 v85 v86::xs)) \wedge \\
& (\forall v12 v87 v88 xs. P xs \Rightarrow P (v12 \text{ says } v87 \text{ domi } v88::xs)) \wedge \\
& (\forall v12 v89 v90 xs. P xs \Rightarrow P (v12 \text{ says } v89 \text{ eqi } v90::xs)) \wedge \\
& (\forall v12 v91 v92 xs. P xs \Rightarrow P (v12 \text{ says } v91 \text{ doms } v92::xs)) \wedge \\
& (\forall v12 v93 v94 xs. P xs \Rightarrow P (v12 \text{ says } v93 \text{ eqs } v94::xs)) \wedge \\
& (\forall v12 v95 v96 xs. P xs \Rightarrow P (v12 \text{ says } v95 \text{ eqn } v96::xs)) \wedge \\
& (\forall v12 v97 v98 xs. P xs \Rightarrow P (v12 \text{ says } v97 \text{ lte } v98::xs)) \wedge \\
& (\forall v12 v99 v100 xs. P xs \Rightarrow P (v12 \text{ says } v99 \text{ lt } v100::xs)) \wedge \\
& (\forall v14 v15 xs. P xs \Rightarrow P (v14 \text{ speaks_for } v15::xs)) \wedge \\
& (\forall v16 v17 xs. P xs \Rightarrow P (v16 \text{ controls } v17::xs)) \wedge \\
& (\forall v18 v19 v20 xs. P xs \Rightarrow P (\text{reps } v18 v19 v20::xs)) \wedge \\
& (\forall v21 v22 xs. P xs \Rightarrow P (v21 \text{ domi } v22::xs)) \wedge \\
& (\forall v23 v24 xs. P xs \Rightarrow P (v23 \text{ eqi } v24::xs)) \wedge \\
& (\forall v25 v26 xs. P xs \Rightarrow P (v25 \text{ doms } v26::xs)) \wedge \\
& (\forall v27 v28 xs. P xs \Rightarrow P (v27 \text{ eqs } v28::xs)) \wedge \\
& (\forall v29 v30 xs. P xs \Rightarrow P (v29 \text{ eqn } v30::xs)) \wedge \\
& (\forall v31 v32 xs. P xs \Rightarrow P (v31 \text{ lte } v32::xs)) \wedge \\
& (\forall v33 v34 xs. P xs \Rightarrow P (v33 \text{ lt } v34::xs)) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

[getRecon_def]

$$\begin{aligned}
& \vdash (\text{getRecon } [] = [\text{NONE}]) \wedge \\
& (\forall xs. \\
& \quad \text{getRecon} \\
& \quad \quad (\text{Name PlatoonLeader says prop (SOME (SLc (PL recon)))::} \\
& \quad \quad \quad xs) = \\
& \quad \quad [\text{SOME (SLc (PL recon))}] \wedge \\
& (\forall xs. \text{getRecon (TT::xs)} = \text{getRecon } xs) \wedge \\
& (\forall xs. \text{getRecon (FF::xs)} = \text{getRecon } xs) \wedge \\
& (\forall xs v2. \text{getRecon (prop } v2::xs) = \text{getRecon } xs) \wedge \\
& (\forall xs v3. \text{getRecon (notf } v3::xs) = \text{getRecon } xs) \wedge \\
& (\forall xs v5 v4. \text{getRecon (v4 andf v5::xs)} = \text{getRecon } xs) \wedge \\
& (\forall xs v7 v6. \text{getRecon (v6 orf v7::xs)} = \text{getRecon } xs) \wedge
\end{aligned}$$

```

(∀ 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

```

$$\begin{aligned}
& (\text{Name PlatoonLeader says prop (SOME (ESCc } v146)) :: \\
& \quad xs)) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL receiveMission))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL warno))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL tentativePlan))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL report1))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL completePlan))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL opoid))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL supervise))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL report2))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL complete))) :: xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow
\end{aligned}$$

$$\begin{aligned}
& P \\
& \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \text{prop (SOME (SLc (PL plIncomplete))))::xs)) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL invalidPlCommand))))::xs)) \wedge \\
& (\forall v151 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PSG v151))))::xs)) \wedge \\
& (\forall v144 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \ (\text{Name PlatoonSergeant says prop (SOME v144)::xs)) \wedge \\
& (\forall v135 \ v136 \ v_{68} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v135 \ \text{meet} \ v136 \ \text{says prop } v_{68}::xs)) \wedge \\
& (\forall v137 \ v138 \ v_{68} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v137 \ \text{quoting } v138 \ \text{says prop } v_{68}::xs)) \wedge \\
& (\forall v_{12} \ v_{69} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says notf } v_{69}::xs)) \wedge \\
& (\forall v_{12} \ v_{70} \ v_{71} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says (v}_{70} \ \text{andf } v_{71})::xs)) \wedge \\
& (\forall v_{12} \ v_{72} \ v_{73} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says (v}_{72} \ \text{orf } v_{73})::xs)) \wedge \\
& (\forall v_{12} \ v_{74} \ v_{75} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says (v}_{74} \ \text{impf } v_{75})::xs)) \wedge \\
& (\forall v_{12} \ v_{76} \ v_{77} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says (v}_{76} \ \text{eqf } v_{77})::xs)) \wedge \\
& (\forall v_{12} \ v_{78} \ v_{79} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says } v_{78} \ \text{says } v_{79}::xs)) \wedge \\
& (\forall v_{12} \ v_{80} \ v_{81} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says } v_{80} \ \text{speaks_for } v_{81}::xs)) \wedge \\
& (\forall v_{12} \ v_{82} \ v_{83} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says } v_{82} \ \text{controls } v_{83}::xs)) \wedge \\
& (\forall v_{12} \ v_{84} \ v_{85} \ v_{86} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says reps } v_{84} \ v_{85} \ v_{86}::xs)) \wedge \\
& (\forall v_{12} \ v_{87} \ v_{88} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says } v_{87} \ \text{domi } v_{88}::xs)) \wedge \\
& (\forall v_{12} \ v_{89} \ v_{90} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says } v_{89} \ \text{eqi } v_{90}::xs)) \wedge \\
& (\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
\end{aligned}$$

$$\forall v. P \ v$$

[getReport_def]

$$\begin{aligned} & \vdash (\text{getReport } [] = [\text{NONE}]) \wedge \\ & (\forall xs. \\ & \quad \text{getReport} \\ & \quad \quad (\text{Name PlatoonLeader says} \\ & \quad \quad \quad \text{prop (SOME (SLc (PL report1)))::xs} = \\ & \quad \quad \quad [\text{SOME (SLc (PL report1))}] \wedge \\ & \quad \quad (\forall xs. \text{getReport (TT::xs)} = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \text{getReport (FF::xs)} = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_2. \text{getReport (prop } v_2::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_3. \text{getReport (notf } v_3::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_5 \ v_4. \text{getReport (v}_4 \text{ andf v}_5::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_7 \ v_6. \text{getReport (v}_6 \text{ orf v}_7::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_9 \ v_8. \text{getReport (v}_8 \text{ impf v}_9::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_{11} \ v_{10}. \text{getReport (v}_{10} \text{ eqf v}_{11}::xs) = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_{12}. \text{getReport (v}_{12} \text{ says TT::xs)} = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_{12}. \text{getReport (v}_{12} \text{ says FF::xs)} = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_{134}. \\ & \quad \quad \quad \text{getReport (Name v}_{134} \text{ says prop NONE::xs)} = \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs \ v_{146}. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says prop (SOME (ESCc v}_{146}))::xs} = \\ & \quad \quad \quad \quad \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says} \\ & \quad \quad \quad \quad \quad \text{prop (SOME (SLc (PL receiveMission)))::xs} = \\ & \quad \quad \quad \quad \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says prop (SOME (SLc (PL warno)))::} \\ & \quad \quad \quad \quad \quad \text{xs}) = \\ & \quad \quad \quad \quad \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says} \\ & \quad \quad \quad \quad \quad \text{prop (SOME (SLc (PL tentativePlan)))::xs} = \\ & \quad \quad \quad \quad \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says prop (SOME (SLc (PL recon)))::} \\ & \quad \quad \quad \quad \quad \text{xs}) = \\ & \quad \quad \quad \quad \text{getReport xs}) \wedge \\ & \quad \quad (\forall xs. \\ & \quad \quad \quad \text{getReport} \\ & \quad \quad \quad \quad (\text{Name PlatoonLeader says} \\ & \quad \quad \quad \quad \quad \text{prop (SOME (SLc (PL completePlan)))::xs} = \end{aligned}$$

```

    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 notif v69::xs) = getReport xs) ∧
  (∀ xs v71 v70 v12.

```

$$\begin{aligned}
& \text{getReport } (v_{12} \text{ says } (v_{70} \text{ andf } v_{71}) :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{73} \ v_{72} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } (v_{72} \text{ orf } v_{73}) :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{75} \ v_{74} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } (v_{74} \text{ impf } v_{75}) :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{77} \ v_{76} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } (v_{76} \text{ eqf } v_{77}) :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{79} \ v_{78} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{78} \text{ says } v_{79} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{81} \ v_{80} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{80} \text{ speaks_for } v_{81} :: xs) = \\
& \quad \text{getReport } xs) \wedge \\
& (\forall xs \ v_{83} \ v_{82} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{82} \text{ controls } v_{83} :: xs) = \\
& \quad \text{getReport } xs) \wedge \\
& (\forall xs \ v_{86} \ v_{85} \ v_{84} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says reps } v_{84} \ v_{85} \ v_{86} :: xs) = \\
& \quad \text{getReport } xs) \wedge \\
& (\forall xs \ v_{88} \ v_{87} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{87} \text{ domi } v_{88} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{90} \ v_{89} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{89} \text{ eqi } v_{90} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{92} \ v_{91} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{91} \text{ doms } v_{92} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{94} \ v_{93} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{93} \text{ eqs } v_{94} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{96} \ v_{95} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{95} \text{ eqn } v_{96} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{98} \ v_{97} \ v_{12}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{97} \text{ lte } v_{98} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{99} \ v_{12} \ v_{100}. \\
& \quad \text{getReport } (v_{12} \text{ says } v_{99} \text{ lt } v_{100} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{15} \ v_{14}. \\
& \quad \text{getReport } (v_{14} \text{ speaks_for } v_{15} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{17} \ v_{16}. \\
& \quad \text{getReport } (v_{16} \text{ controls } v_{17} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{20} \ v_{19} \ v_{18}. \\
& \quad \text{getReport } (\text{reps } v_{18} \ v_{19} \ v_{20} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{22} \ v_{21}. \text{getReport } (v_{21} \text{ domi } v_{22} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{24} \ v_{23}. \text{getReport } (v_{23} \text{ eqi } v_{24} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{26} \ v_{25}. \text{getReport } (v_{25} \text{ doms } v_{26} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{28} \ v_{27}. \text{getReport } (v_{27} \text{ eqs } v_{28} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{30} \ v_{29}. \text{getReport } (v_{29} \text{ eqn } v_{30} :: xs) = \text{getReport } xs) \wedge \\
& (\forall xs \ v_{32} \ v_{31}. \text{getReport } (v_{31} \text{ lte } v_{32} :: xs) = \text{getReport } xs) \wedge \\
& \forall xs \ v_{34} \ v_{33}. \text{getReport } (v_{33} \text{ lt } v_{34} :: xs) = \text{getReport } xs
\end{aligned}$$

[getReport_ind]

$\vdash \forall P.$
 $P \ [] \ \wedge$

$$\begin{aligned}
& (\forall xs. \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL report1))))::xs})) \wedge \\
& (\forall xs. P \ xs \Rightarrow P \ (\text{TT::xs})) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{FF::xs})) \wedge \\
& (\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2::xs)) \wedge \\
& (\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3::xs)) \wedge \\
& (\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \ \text{andf } v_5::xs)) \wedge \\
& (\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \ \text{orf } v_7::xs)) \wedge \\
& (\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \ \text{impf } v_9::xs)) \wedge \\
& (\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \ \text{eqf } v_{11}::xs)) \wedge \\
& (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says TT::xs})) \wedge \\
& (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says FF::xs})) \wedge \\
& (\forall v_{134} \ xs. P \ xs \Rightarrow P \ (\text{Name } v_{134} \ \text{says prop NONE::xs})) \wedge \\
& (\forall v_{146} \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says prop (SOME (ESCc } v_{146}))::} \\
& \quad \quad \quad xs)) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL receiveMission))))::xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL warno))))::xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL tentativePlan))))::xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL recon))))::xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL completePlan))))::xs})) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL opoid))))::xs})) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL supervise)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL report2)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL complete)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL plIncomplete)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL invalidPlCommand)))::xs))} \wedge \\
& (\forall v151 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PSG v151)))::xs))} \wedge \\
& (\forall v144 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \text{ (Name PlatoonSergeant says prop (SOME v144)::xs))} \wedge \\
& (\forall v135 \ v136 \ v68 \ xs. \\
& \quad P \ xs \Rightarrow P \text{ (v135 meet v136 says prop v68::xs))} \wedge \\
& (\forall v137 \ v138 \ v68 \ xs. \\
& \quad P \ xs \Rightarrow P \text{ (v137 quoting v138 says prop v68::xs))} \wedge \\
& (\forall v12 \ v69 \ xs. P \ xs \Rightarrow P \text{ (v12 says notif v69::xs))} \wedge \\
& (\forall v12 \ v70 \ v71 \ xs. P \ xs \Rightarrow P \text{ (v12 says (v70 andf v71)::xs))} \wedge \\
& (\forall v12 \ v72 \ v73 \ xs. P \ xs \Rightarrow P \text{ (v12 says (v72 orf v73)::xs))} \wedge \\
& (\forall v12 \ v74 \ v75 \ xs. P \ xs \Rightarrow P \text{ (v12 says (v74 impf v75)::xs))} \wedge \\
& (\forall v12 \ v76 \ v77 \ xs. P \ xs \Rightarrow P \text{ (v12 says (v76 eqf v77)::xs))} \wedge \\
& (\forall v12 \ v78 \ v79 \ xs. P \ xs \Rightarrow P \text{ (v12 says v78 says v79::xs))} \wedge \\
& (\forall v12 \ v80 \ v81 \ xs. \\
& \quad P \ xs \Rightarrow P \text{ (v12 says v80 speaks_for v81::xs))} \wedge \\
& (\forall v12 \ v82 \ v83 \ xs. \\
& \quad P \ xs \Rightarrow P \text{ (v12 says v82 controls v83::xs))} \wedge \\
& (\forall v12 \ v84 \ v85 \ v86 \ xs. \\
& \quad P \ xs \Rightarrow P \text{ (v12 says reps v84 v85 v86::xs))} \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_{12} v_{87} v_{88} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{87} \text{ domi } v_{88} :: xs)) \wedge \\
& (\forall v_{12} v_{89} v_{90} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{89} \text{ eqi } v_{90} :: xs)) \wedge \\
& (\forall v_{12} v_{91} v_{92} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{91} \text{ doms } v_{92} :: xs)) \wedge \\
& (\forall v_{12} v_{93} v_{94} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{93} \text{ eqs } v_{94} :: xs)) \wedge \\
& (\forall v_{12} v_{95} v_{96} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{95} \text{ eqn } v_{96} :: xs)) \wedge \\
& (\forall v_{12} v_{97} v_{98} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{97} \text{ lte } v_{98} :: xs)) \wedge \\
& (\forall v_{12} v_{99} v_{100} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{99} \text{ lt } v_{100} :: xs)) \wedge \\
& (\forall v_{14} v_{15} xs. P xs \Rightarrow P (v_{14} \text{ speaks_for } v_{15} :: xs)) \wedge \\
& (\forall v_{16} v_{17} xs. P xs \Rightarrow P (v_{16} \text{ controls } v_{17} :: xs)) \wedge \\
& (\forall v_{18} v_{19} v_{20} xs. P xs \Rightarrow P (\text{reps } v_{18} v_{19} v_{20} :: xs)) \wedge \\
& (\forall v_{21} v_{22} xs. P xs \Rightarrow P (v_{21} \text{ domi } v_{22} :: xs)) \wedge \\
& (\forall v_{23} v_{24} xs. P xs \Rightarrow P (v_{23} \text{ eqi } v_{24} :: xs)) \wedge \\
& (\forall v_{25} v_{26} xs. P xs \Rightarrow P (v_{25} \text{ doms } v_{26} :: xs)) \wedge \\
& (\forall v_{27} v_{28} xs. P xs \Rightarrow P (v_{27} \text{ eqs } v_{28} :: xs)) \wedge \\
& (\forall v_{29} v_{30} xs. P xs \Rightarrow P (v_{29} \text{ eqn } v_{30} :: xs)) \wedge \\
& (\forall v_{31} v_{32} xs. P xs \Rightarrow P (v_{31} \text{ lte } v_{32} :: xs)) \wedge \\
& (\forall v_{33} v_{34} xs. P xs \Rightarrow P (v_{33} \text{ lt } v_{34} :: xs)) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

[getTentativePlan_def]

$$\begin{aligned}
& \vdash (\text{getTentativePlan } [] = [\text{NONE}]) \wedge \\
& (\forall xs. \\
& \quad \text{getTentativePlan} \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL tentativePlan))) :: xs}) = \\
& \quad \quad \quad [\text{SOME (SLc (PL tentativePlan))}] \wedge \\
& (\forall xs. \text{getTentativePlan (TT :: xs)} = \text{getTentativePlan } xs) \wedge \\
& (\forall xs. \text{getTentativePlan (FF :: xs)} = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_2. \\
& \quad \text{getTentativePlan (prop } v_2 :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_3. \\
& \quad \text{getTentativePlan (notf } v_3 :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_5 \ v_4. \\
& \quad \text{getTentativePlan (} v_4 \text{ andf } v_5 :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_7 \ v_6. \\
& \quad \text{getTentativePlan (} v_6 \text{ orf } v_7 :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_9 \ v_8. \\
& \quad \text{getTentativePlan (} v_8 \text{ impf } v_9 :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_{11} \ v_{10}. \\
& \quad \text{getTentativePlan (} v_{10} \text{ eqf } v_{11} :: xs) = \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_{12}. \\
& \quad \text{getTentativePlan (} v_{12} \text{ says TT :: xs) = getTentativePlan } xs) \wedge \\
& (\forall xs \ v_{12}. \\
& \quad \text{getTentativePlan (} v_{12} \text{ says FF :: xs) = getTentativePlan } xs) \wedge \\
& (\forall xs \ v_{134}. \\
& \quad \text{getTentativePlan (Name } v_{134} \text{ says prop NONE :: xs) =} \\
& \quad \text{getTentativePlan } xs) \wedge \\
& (\forall xs \ v_{146}. \\
& \quad \text{getTentativePlan}
\end{aligned}$$

```

      (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) =

```



```

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

```

[getTentativePlan_ind]

⊢ ∀ P.
 P [] ∧

$$\begin{aligned}
& (\forall xs. \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL tentativePlan)))::xs}) \wedge \\
& (\forall xs. P \ xs \Rightarrow P \ (\text{TT::xs})) \wedge (\forall xs. P \ xs \Rightarrow P \ (\text{FF::xs})) \wedge \\
& (\forall v_2 \ xs. P \ xs \Rightarrow P \ (\text{prop } v_2::xs)) \wedge \\
& (\forall v_3 \ xs. P \ xs \Rightarrow P \ (\text{notf } v_3::xs)) \wedge \\
& (\forall v_4 \ v_5 \ xs. P \ xs \Rightarrow P \ (v_4 \ \text{andf } v_5::xs)) \wedge \\
& (\forall v_6 \ v_7 \ xs. P \ xs \Rightarrow P \ (v_6 \ \text{orf } v_7::xs)) \wedge \\
& (\forall v_8 \ v_9 \ xs. P \ xs \Rightarrow P \ (v_8 \ \text{impf } v_9::xs)) \wedge \\
& (\forall v_{10} \ v_{11} \ xs. P \ xs \Rightarrow P \ (v_{10} \ \text{eqf } v_{11}::xs)) \wedge \\
& (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says TT::xs})) \wedge \\
& (\forall v_{12} \ xs. P \ xs \Rightarrow P \ (v_{12} \ \text{says FF::xs})) \wedge \\
& (\forall v_{134} \ xs. P \ xs \Rightarrow P \ (\text{Name } v_{134} \ \text{says prop NONE::xs})) \wedge \\
& (\forall v_{146} \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says prop (SOME (ESCc } v_{146})):: \\
& \quad \quad \quad xs)) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL receiveMission)))::xs}) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL warno)))::xs}) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL recon)))::xs}) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL report1)))::xs}) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL completePlan)))::xs}) \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \quad (\text{Name PlatoonLeader says} \\
& \quad \quad \quad \text{prop (SOME (SLc (PL opoid)))::xs}) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL supervise)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL report2)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL complete)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL plIncomplete)))::xs))} \wedge \\
& (\forall xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PL invalidPlCommand)))::xs))} \wedge \\
& (\forall v151 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \\
& \quad \text{(Name PlatoonLeader says} \\
& \quad \text{prop (SOME (SLc (PSG v151)))::xs))} \wedge \\
& (\forall v144 \ xs. \\
& \quad P \ xs \Rightarrow \\
& \quad P \ \text{(Name PlatoonSergeant says prop (SOME v144)::xs))} \wedge \\
& (\forall v135 \ v136 \ v68 \ xs. \\
& \quad P \ xs \Rightarrow P \ (v135 \ \text{meet} \ v136 \ \text{says prop} \ v68::xs)) \wedge \\
& (\forall v137 \ v138 \ v68 \ xs. \\
& \quad P \ xs \Rightarrow P \ (v137 \ \text{quoting} \ v138 \ \text{says prop} \ v68::xs)) \wedge \\
& (\forall v_{12} \ v_{69} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ \text{notif} \ v_{69}::xs)) \wedge \\
& (\forall v_{12} \ v_{70} \ v_{71} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ (v_{70} \ \text{andf} \ v_{71})::xs)) \wedge \\
& (\forall v_{12} \ v_{72} \ v_{73} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ (v_{72} \ \text{orf} \ v_{73})::xs)) \wedge \\
& (\forall v_{12} \ v_{74} \ v_{75} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ (v_{74} \ \text{impf} \ v_{75})::xs)) \wedge \\
& (\forall v_{12} \ v_{76} \ v_{77} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ (v_{76} \ \text{eqf} \ v_{77})::xs)) \wedge \\
& (\forall v_{12} \ v_{78} \ v_{79} \ xs. \ P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ v_{78} \ \text{says} \ v_{79}::xs)) \wedge \\
& (\forall v_{12} \ v_{80} \ v_{81} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ v_{80} \ \text{speaks_for} \ v_{81}::xs)) \wedge \\
& (\forall v_{12} \ v_{82} \ v_{83} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ v_{82} \ \text{controls} \ v_{83}::xs)) \wedge \\
& (\forall v_{12} \ v_{84} \ v_{85} \ v_{86} \ xs. \\
& \quad P \ xs \Rightarrow P \ (v_{12} \ \text{says} \ \text{reps} \ v_{84} \ v_{85} \ v_{86}::xs)) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_{12} v_{87} v_{88} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{87} \text{ domi } v_{88} :: xs)) \wedge \\
& (\forall v_{12} v_{89} v_{90} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{89} \text{ eqi } v_{90} :: xs)) \wedge \\
& (\forall v_{12} v_{91} v_{92} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{91} \text{ doms } v_{92} :: xs)) \wedge \\
& (\forall v_{12} v_{93} v_{94} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{93} \text{ eqs } v_{94} :: xs)) \wedge \\
& (\forall v_{12} v_{95} v_{96} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{95} \text{ eqn } v_{96} :: xs)) \wedge \\
& (\forall v_{12} v_{97} v_{98} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{97} \text{ lte } v_{98} :: xs)) \wedge \\
& (\forall v_{12} v_{99} v_{100} xs. P xs \Rightarrow P (v_{12} \text{ says } v_{99} \text{ lt } v_{100} :: xs)) \wedge \\
& (\forall v_{14} v_{15} xs. P xs \Rightarrow P (v_{14} \text{ speaks_for } v_{15} :: xs)) \wedge \\
& (\forall v_{16} v_{17} xs. P xs \Rightarrow P (v_{16} \text{ controls } v_{17} :: xs)) \wedge \\
& (\forall v_{18} v_{19} v_{20} xs. P xs \Rightarrow P (\text{reps } v_{18} v_{19} v_{20} :: xs)) \wedge \\
& (\forall v_{21} v_{22} xs. P xs \Rightarrow P (v_{21} \text{ domi } v_{22} :: xs)) \wedge \\
& (\forall v_{23} v_{24} xs. P xs \Rightarrow P (v_{23} \text{ eqi } v_{24} :: xs)) \wedge \\
& (\forall v_{25} v_{26} xs. P xs \Rightarrow P (v_{25} \text{ doms } v_{26} :: xs)) \wedge \\
& (\forall v_{27} v_{28} xs. P xs \Rightarrow P (v_{27} \text{ eqs } v_{28} :: xs)) \wedge \\
& (\forall v_{29} v_{30} xs. P xs \Rightarrow P (v_{29} \text{ eqn } v_{30} :: xs)) \wedge \\
& (\forall v_{31} v_{32} xs. P xs \Rightarrow P (v_{31} \text{ lte } v_{32} :: xs)) \wedge \\
& (\forall v_{33} v_{34} xs. P xs \Rightarrow P (v_{33} \text{ lt } v_{34} :: xs)) \Rightarrow \\
& \forall v. P v
\end{aligned}$$

Index

ConductORPType Theory, 40

Datatypes, 40

Theorems, 41

plCommand_distinct_clauses, 41

psgCommand_distinct_clauses, 41

slCommand_distinct_clauses, 41

slCommand_one_one, 41

slOutput_distinct_clauses, 41

slRole_distinct_clauses, 41

slState_distinct_clauses, 41

ConductPBType Theory, 47

Datatypes, 47

Theorems, 47

plCommandPB_distinct_clauses, 47

psgCommandPB_distinct_clauses, 47

slCommand_distinct_clauses, 47

slCommand_one_one, 47

slOutput_distinct_clauses, 48

slRole_distinct_clauses, 48

slState_distinct_clauses, 48

MoveToORPType Theory, 53

Datatypes, 53

Theorems, 53

slCommand_distinct_clauses, 53

slOutput_distinct_clauses, 53

slState_distinct_clauses, 53

MoveToPBType Theory, 58

Datatypes, 58

Theorems, 58

slCommand_distinct_clauses, 58

slOutput_distinct_clauses, 59

slState_distinct_clauses, 59

OMNIType Theory, 3

Datatypes, 3

Theorems, 3

command_distinct_clauses, 3

command_one_one, 3

escCommand_distinct_clauses, 3

escOutput_distinct_clauses, 3

escState_distinct_clauses, 3

output_distinct_clauses, 4

output_one_one, 4

principal_one_one, 4

state_distinct_clauses, 4

state_one_one, 4

PBIntegratedDef Theory, 23

Definitions, 23

secAuthorization_def, 23

secContext_def, 23

secHelper_def, 24

Theorems, 24

getOmniCommand_def, 24

getOmniCommand_ind, 27

PBTypeIntegrated Theory, 21

Datatypes, 21

Theorems, 22

omniCommand_distinct_clauses, 22

plCommand_distinct_clauses, 22

slCommand_distinct_clauses, 22

slCommand_one_one, 22

slOutput_distinct_clauses, 23

slState_distinct_clauses, 23

stateRole_distinct_clauses, 23

PlanPBDef Theory, 72

Definitions, 72

PL_notWARNO_Auth_def, 72

PL_WARNO_Auth_def, 73

secContext_def, 73

secContextNull_def, 73

Theorems, 73

getInitMove_def, 73

getInitMove_ind, 76

getPlCom_def, 77

getPlCom_ind, 79

getPsgCom_def, 80

getPsgCom_ind, 82

getRecon_def, 83

getRecon_ind, 86

getReport_def, 89

- getReport_ind, 91
- getTenativePlan_def, 94
- getTenativePlan_ind, 97
- PlanPBType Theory**, 69
 - Datatypes, 69
 - Theorems, 70
 - plCommand_distinct_clauses, 70
 - psgCommand_distinct_clauses, 70
 - slCommand_distinct_clauses, 70
 - slCommand_one_one, 70
 - slOutput_distinct_clauses, 71
 - slRole_distinct_clauses, 71
 - slState_distinct_clauses, 72
- satList Theory**, 21
 - Definitions, 21
 - satList_def, 21
 - Theorems, 21
 - satList_conj, 21
 - satList_CONS, 21
 - satList_nil, 21
- ssm Theory**, 11
 - Datatypes, 11
 - Definitions, 12
 - authenticationTest_def, 12
 - commandList_def, 12
 - inputList_def, 12
 - propCommandList_def, 12
 - TR_def, 12
 - Theorems, 13
 - CFGInterpret_def, 13
 - CFGInterpret_ind, 13
 - configuration_one_one, 13
 - extractCommand_def, 13
 - extractCommand_ind, 13
 - extractInput_def, 14
 - extractInput_ind, 14
 - extractPropCommand_def, 15
 - extractPropCommand_ind, 15
 - TR_cases, 16
 - TR_discard_cmd_rule, 17
 - TR_EQ_rules_thm, 17
 - TR_exec_cmd_rule, 17
 - TR_ind, 18
 - TR_rules, 18
 - TR_strongind, 19
 - TR_trap_cmd_rule, 20
 - TRrule0, 20
 - TRrule1, 20
 - trType_distinct_clauses, 20
 - trType_one_one, 21
- ssm11 Theory**, 4
 - Datatypes, 4
 - Definitions, 4
 - TR_def, 4
 - Theorems, 5
 - CFGInterpret_def, 5
 - CFGInterpret_ind, 6
 - configuration_one_one, 6
 - order_distinct_clauses, 6
 - order_one_one, 6
 - TR_cases, 6
 - TR_discard_cmd_rule, 7
 - TR_EQ_rules_thm, 7
 - TR_exec_cmd_rule, 8
 - TR_ind, 8
 - TR_rules, 9
 - TR_strongind, 9
 - TR_trap_cmd_rule, 10
 - TRrule0, 10
 - TRrule1, 11
 - trType_distinct_clauses, 11
 - trType_one_one, 11
- ssmConductORP Theory**, 35
 - Definitions, 35
 - secContextConductORP_def, 35
 - ssmConductORPStateInterp_def, 35
 - Theorems, 35
 - authTestConductORP_cmd_reject_lemma, 35
 - authTestConductORP_def, 36
 - authTestConductORP_ind, 36
 - conductORPNS_def, 37
 - conductORPNS_ind, 37
 - conductORPOut_def, 38
 - conductORPOut_ind, 38
 - PlatoonLeader_exec_plCommand_jus-

tified.thm, 39
 PlatoonLeader_plCommand.lemma, 39
 PlatoonSergeant_exec_psgCommand_-
 justified.thm, 40
 PlatoonSergeant_psgCommand.lemma,
 40
ssmConductPB Theory, 41
 Definitions, 42
 secContextConductPB_def, 42
 ssmConductPBStateInterp_def, 42
 Theorems, 42
 authTestConductPB_cmd_reject.lemma,
 42
 authTestConductPB_def, 42
 authTestConductPB_ind, 43
 conductPBNS_def, 44
 conductPBNS_ind, 44
 conductPBOut_def, 45
 conductPBOut_ind, 45
 PlatoonLeader_exec_plCommandPB_-
 justified.thm, 46
 PlatoonLeader_plCommandPB.lemma,
 46
 PlatoonSergeant_exec_psgCommandPB_-
 justified.thm, 46
 PlatoonSergeant_psgCommandPB.lemma,
 47
ssmMoveToORP Theory, 48
 Definitions, 48
 secContextMoveToORP_def, 48
 ssmMoveToORPStateInterp_def, 48
 Theorems, 48
 authTestMoveToORP_cmd_reject.lemma,
 48
 authTestMoveToORP_def, 49
 authTestMoveToORP_ind, 49
 moveToORPNS_def, 50
 moveToORPNS_ind, 50
 moveToORPOut_def, 51
 moveToORPOut_ind, 51
 PlatoonLeader_exec_slCommand_jus-
 tified.thm, 52
 PlatoonLeader_slCommand.lemma, 52
ssmMoveToPB Theory, 54
 Definitions, 54
 secContextMoveToPB_def, 54
 ssmMoveToPBStateInterp_def, 54
 Theorems, 54
 authTestMoveToPB_cmd_reject.lemma,
 54
 authTestMoveToPB_def, 54
 authTestMoveToPB_ind, 55
 moveToPBNS_def, 55
 moveToPBNS_ind, 56
 moveToPBOut_def, 56
 moveToPBOut_ind, 57
 PlatoonLeader_exec_slCommand_jus-
 tified.thm, 57
 PlatoonLeader_slCommand.lemma, 58
ssmPBIntegrated Theory, 28
 Theorems, 28
 inputOK_def, 28
 inputOK_ind, 29
 PBNS_def, 29
 PBNS_ind, 30
 PBOut_def, 31
 PBOut_ind, 31
 PlatoonLeader_Omni_notDiscard_slCom-
 mand.thm, 32
 PlatoonLeader_PLAN_PB_exec_justi-
 fied.thm, 32
 PlatoonLeader_PLAN_PB_exec.lemma,
 33
 PlatoonLeader_PLAN_PB_trap_justi-
 fied.lemma, 33
 PlatoonLeader_PLAN_PB_trap_justi-
 fied.thm, 34
 PlatoonLeader_PLAN_PB_trap.lemma,
 35
ssmPlanPB Theory, 59
 Theorems, 59
 inputOK_def, 59
 inputOK_ind, 60
 planPBNS_def, 61
 planPBNS_ind, 61
 planPBOut_def, 61

planPBOut_ind, 62
 PlatoonLeader_notWARNO_notreport1_-
 exec_plCommand_justified_lemma, 62
 PlatoonLeader_notWARNO_notreport1_-
 exec_plCommand_justified_thm, 63
 PlatoonLeader_notWARNO_notreport1_-
 exec_plCommand_lemma, 63
 PlatoonLeader_psgCommand_notDis-
 card_thm, 64
 PlatoonLeader_trap_psgCommand_jus-
 tified_lemma, 64
 PlatoonLeader_trap_psgCommand_lemma,
 65
 PlatoonLeader_WARNO_exec_report1_-
 justified_lemma, 65
 PlatoonLeader_WARNO_exec_report1_-
 justified_thm, 66
 PlatoonLeader_WARNO_exec_report1_-
 lemma, 67
 PlatoonSergeant_trap_plCommand_jus-
 tified_lemma, 68
 PlatoonSergeant_trap_plCommand_jus-
 tified_thm, 68
 PlatoonSergeant_trap_plCommand_lemma,
 69