Contents

1 OMNIType Theory

Built: 02 July 2017
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]

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

[command_one_one]

$\vdash (\forall a\ a'.\ (\texttt{ESCc}\ a = \texttt{ESCc}\ a') \iff (a = a')) \land$
$\forall a\ a'.\ (\texttt{SLc}\ a = \texttt{SLc}\ a') \iff (a = a')$

[escCommand_distinct_clauses]

$\vdash \texttt{returnToBase} \neq \texttt{changeMission} \land \texttt{returnToBase} \neq \texttt{resupply} \land$
$\texttt{returnToBase} \neq \texttt{reactToContact} \land \texttt{changeMission} \neq \texttt{resupply} \land$
$\texttt{changeMission} \neq \texttt{reactToContact} \land \texttt{resupply} \neq \texttt{reactToContact}$

[escOutput_distinct_clauses]

$\vdash \texttt{ReturnToBase} \neq \texttt{ChangeMission} \land \texttt{ReturnToBase} \neq \texttt{Resupply} \land$
$\texttt{ReturnToBase} \neq \texttt{ReactToContact} \land \texttt{ChangeMission} \neq \texttt{Resupply} \land$
$\texttt{ChangeMission} \neq \texttt{ReactToContact} \land \texttt{Resupply} \neq \texttt{ReactToContact}$

[escState_distinct_clauses]

$\vdash \texttt{RTB} \neq \texttt{CM} \land \texttt{RTB} \neq \texttt{RESUPPLY} \land \texttt{RTB} \neq \texttt{RTC} \land \texttt{CM} \neq \texttt{RESUPPLY} \land$
$\texttt{CM} \neq \texttt{RTC} \land \texttt{RESUPPLY} \neq \texttt{RTC}$

[output_distinct_clauses]

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

[output_one_one]

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

[principal_one_one]

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

[state_distinct_clauses]

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

[state_one_one]

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

2 ssmPB Theory

Built: 02 July 2017

Parent Theories: PBType, ssm11, OMNIType

2.1 Definitions

[secContext_def]

$\vdash \forall \mathit{slCommand}.$
$\quad \texttt{secContext}\ \mathit{slCommand} =$
$\quad [\texttt{Name PlatoonLeader controls prop (SOME (SLc}\ \mathit{slCommand}\texttt{))}]$

[ssmPBStateInterp_def]

$\vdash \forall \mathit{state}.\ \texttt{ssmPBStateInterp}\ \mathit{state} = \texttt{TT}$

2.2 Theorems

[authenticationTest_cmd_reject_lemma]

$\vdash \forall \mathit{cmd}.\ \neg\texttt{authenticationTest (prop (SOME}\ \mathit{cmd}\texttt{))}$

[authenticationTest_def]

$\vdash (\texttt{authenticationTest (Name PlatoonLeader says prop}\ \mathit{cmd}) \iff$
$\quad \texttt{T}) \land (\texttt{authenticationTest TT} \iff \texttt{F}) \land$
$\quad (\texttt{authenticationTest FF} \iff \texttt{F}) \land$
$\quad (\texttt{authenticationTest (prop}\ v) \iff \texttt{F}) \land$
$\quad (\texttt{authenticationTest (notf}\ v_1) \iff \texttt{F}) \land$
$\quad (\texttt{authenticationTest}\ (v_2\ \texttt{andf}\ v_3) \iff \texttt{F}) \land$
$\quad (\texttt{authenticationTest}\ (v_4\ \texttt{orf}\ v_5) \iff \texttt{F}) \land$

(authenticationTest (v_6 impf v_7) $\iff$ F) $\wedge$
(authenticationTest (v_8 eqf v_9) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says TT) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says FF) $\iff$ F) $\wedge$
(authenticationTest ($v133$ meet $v134$ says prop v_{66}) $\iff$ F) $\wedge$
(authenticationTest ($v135$ quoting $v136$ says prop v_{66}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says notf v_{67}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{68} andf v_{69})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{70} orf v_{71})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{72} impf v_{73})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says (v_{74} eqf v_{75})) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{76} says v_{77}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{78} speaks_for v_{79}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{80} controls v_{81}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says reps v_{82} v_{83} v_{84}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{85} domi v_{86}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{87} eqi v_{88}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{89} doms v_{90}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{91} eqs v_{92}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{93} eqn v_{94}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{95} lte v_{96}) $\iff$ F) $\wedge$
(authenticationTest (v_{10} says v_{97} lt v_{98}) $\iff$ F) $\wedge$
(authenticationTest (v_{12} speaks_for v_{13}) $\iff$ F) $\wedge$
(authenticationTest (v_{14} controls v_{15}) $\iff$ F) $\wedge$
(authenticationTest (reps v_{16} v_{17} v_{18}) $\iff$ F) $\wedge$
(authenticationTest (v_{19} domi v_{20}) $\iff$ F) $\wedge$
(authenticationTest (v_{21} eqi v_{22}) $\iff$ F) $\wedge$
(authenticationTest (v_{23} doms v_{24}) $\iff$ F) $\wedge$
(authenticationTest (v_{25} eqs v_{26}) $\iff$ F) $\wedge$
(authenticationTest (v_{27} eqn v_{28}) $\iff$ F) $\wedge$
(authenticationTest (v_{29} lte v_{30}) $\iff$ F) $\wedge$
(authenticationTest (v_{31} lt v_{32}) $\iff$ F)

[authenticationTest_ind]

$\vdash \forall P$.
($\forall cmd$. P (Name PlatoonLeader says prop cmd)) $\wedge$ P TT $\wedge$
P FF $\wedge$ ($\forall v$. P (prop v)) $\wedge$ ($\forall v_1$. P (notf v_1)) $\wedge$
($\forall v_2$ v_3. P (v_2 andf v_3)) $\wedge$ ($\forall v_4$ v_5. P (v_4 orf v_5)) $\wedge$
($\forall v_6$ v_7. P (v_6 impf v_7)) $\wedge$ ($\forall v_8$ v_9. P (v_8 eqf v_9)) $\wedge$
($\forall v_{10}$. P (v_{10} says TT)) $\wedge$ ($\forall v_{10}$. P (v_{10} says FF)) $\wedge$
($\forall v133$ $v134$ v_{66}. P ($v133$ meet $v134$ says prop v_{66})) $\wedge$
($\forall v135$ $v136$ v_{66}. P ($v135$ quoting $v136$ says prop v_{66})) $\wedge$
($\forall v_{10}$ v_{67}. P (v_{10} says notf v_{67})) $\wedge$
($\forall v_{10}$ v_{68} v_{69}. P (v_{10} says (v_{68} andf v_{69}))) $\wedge$
($\forall v_{10}$ v_{70} v_{71}. P (v_{10} says (v_{70} orf v_{71}))) $\wedge$
($\forall v_{10}$ v_{72} v_{73}. P (v_{10} says (v_{72} impf v_{73}))) $\wedge$
($\forall v_{10}$ v_{74} v_{75}. P (v_{10} says (v_{74} eqf v_{75}))) $\wedge$
($\forall v_{10}$ v_{76} v_{77}. P (v_{10} says v_{76} says v_{77})) $\wedge$
($\forall v_{10}$ v_{78} v_{79}. P (v_{10} says v_{78} speaks_for v_{79})) $\wedge$

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

[authenticationTest_TT_reject_lemma]

⊢ ¬authenticationTest TT

[PBNS_def]

```
⊢ (PBNS PLAN_PB (exec (SLc crossLD)) = MOVE_TO_ORP) ∧
  (PBNS PLAN_PB (exec (SLc incomplete)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (exec (SLc conductORP)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_ORP (exec (SLc incomplete)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (exec (SLc moveToPB)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_ORP (exec (SLc incomplete)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_PB (exec (SLc conductPB)) = CONDUCT_PB) ∧
  (PBNS MOVE_TO_PB (exec (SLc incomplete)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_PB (exec (SLc completePB)) = COMPLETE_PB) ∧
  (PBNS CONDUCT_PB (exec (SLc incomplete)) = CONDUCT_PB) ∧
  (PBNS PLAN_PB (trap (SLc crossLD)) = PLAN_PB) ∧
  (PBNS PLAN_PB (trap (SLc incomplete)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (trap (SLc moveToORP)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (trap (SLc moveToPB)) = CONDUCT_ORP) ∧
  (PBNS CONDUCT_ORP (trap (SLc incomplete)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_PB (trap (SLc conductPB)) = MOVE_TO_PB) ∧
  (PBNS MOVE_TO_PB (trap (SLc incomplete)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_PB (trap (SLc completePB)) = CONDUCT_PB) ∧
  (PBNS CONDUCT_PB (trap (SLc incomplete)) = CONDUCT_PB) ∧
  (PBNS PLAN_PB (discard (SLc crossLD)) = PLAN_PB) ∧
  (PBNS MOVE_TO_ORP (discard (SLc moveToORP)) = MOVE_TO_ORP) ∧
  (PBNS CONDUCT_ORP (discard (SLc moveToPB)) = CONDUCT_ORP) ∧
  (PBNS MOVE_TO_PB (discard (SLc conductPB)) = MOVE_TO_PB) ∧
  (PBNS CONDUCT_PB (discard (SLc completePB)) = CONDUCT_PB)
```

[PBNS_ind]

```
⊢ ∀P.
    P PLAN_PB (exec (SLc crossLD)) ∧
    P PLAN_PB (exec (SLc incomplete)) ∧
    P MOVE_TO_ORP (exec (SLc conductORP)) ∧
    P MOVE_TO_ORP (exec (SLc incomplete)) ∧
    P CONDUCT_ORP (exec (SLc moveToPB)) ∧
    P CONDUCT_ORP (exec (SLc incomplete)) ∧
    P MOVE_TO_PB (exec (SLc conductPB)) ∧
    P MOVE_TO_PB (exec (SLc incomplete)) ∧
    P CONDUCT_PB (exec (SLc completePB)) ∧
    P CONDUCT_PB (exec (SLc incomplete)) ∧
    P PLAN_PB (trap (SLc crossLD)) ∧
    P PLAN_PB (trap (SLc incomplete)) ∧
    (∀moveToORP. P MOVE_TO_ORP (trap (SLc moveToORP))) ∧
    P CONDUCT_ORP (trap (SLc moveToPB)) ∧
    P CONDUCT_ORP (trap (SLc incomplete)) ∧
    P MOVE_TO_PB (trap (SLc conductPB)) ∧
    P MOVE_TO_PB (trap (SLc incomplete)) ∧
    P CONDUCT_PB (trap (SLc completePB)) ∧
    P CONDUCT_PB (trap (SLc incomplete)) ∧
    P PLAN_PB (discard (SLc crossLD)) ∧
    (∀moveToORP. P MOVE_TO_ORP (discard (SLc moveToORP))) ∧
    P CONDUCT_ORP (discard (SLc moveToPB)) ∧
    P MOVE_TO_PB (discard (SLc conductPB)) ∧
    P CONDUCT_PB (discard (SLc completePB)) ∧
    (∀v8 v6. P v8 (discard (ESCc v6))) ∧
    P PLAN_PB (discard (SLc conductORP)) ∧
    P PLAN_PB (discard (SLc moveToPB)) ∧
    P PLAN_PB (discard (SLc conductPB)) ∧
    P PLAN_PB (discard (SLc completePB)) ∧
    P PLAN_PB (discard (SLc incomplete)) ∧
    P CONDUCT_ORP (discard (SLc crossLD)) ∧
    P CONDUCT_ORP (discard (SLc conductORP)) ∧
    P CONDUCT_ORP (discard (SLc conductPB)) ∧
    P CONDUCT_ORP (discard (SLc completePB)) ∧
    P CONDUCT_ORP (discard (SLc incomplete)) ∧
    P MOVE_TO_PB (discard (SLc crossLD)) ∧
    P MOVE_TO_PB (discard (SLc conductORP)) ∧
    P MOVE_TO_PB (discard (SLc moveToPB)) ∧
    P MOVE_TO_PB (discard (SLc completePB)) ∧
    P MOVE_TO_PB (discard (SLc incomplete)) ∧
    P CONDUCT_PB (discard (SLc crossLD)) ∧
    P CONDUCT_PB (discard (SLc conductORP)) ∧
    P CONDUCT_PB (discard (SLc moveToPB)) ∧
    P CONDUCT_PB (discard (SLc conductPB)) ∧
    P CONDUCT_PB (discard (SLc incomplete)) ∧
    (∀v9. P COMPLETE_PB (discard (SLc v9))) ∧
    (∀v13 v11. P v13 (trap (ESCc v11))) ∧
```

P PLAN_PB (trap (SLc conductORP)) $\wedge$
P PLAN_PB (trap (SLc moveToPB)) $\wedge$
P PLAN_PB (trap (SLc conductPB)) $\wedge$
P PLAN_PB (trap (SLc completePB)) $\wedge$
P CONDUCT_ORP (trap (SLc crossLD)) $\wedge$
P CONDUCT_ORP (trap (SLc conductORP)) $\wedge$
P CONDUCT_ORP (trap (SLc conductPB)) $\wedge$
P CONDUCT_ORP (trap (SLc completePB)) $\wedge$
P MOVE_TO_PB (trap (SLc crossLD)) $\wedge$
P MOVE_TO_PB (trap (SLc conductORP)) $\wedge$
P MOVE_TO_PB (trap (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (trap (SLc completePB)) $\wedge$
P CONDUCT_PB (trap (SLc crossLD)) $\wedge$
P CONDUCT_PB (trap (SLc conductORP)) $\wedge$
P CONDUCT_PB (trap (SLc moveToPB)) $\wedge$
P CONDUCT_PB (trap (SLc conductPB)) $\wedge$
($\forall v_{14}$. P COMPLETE_PB (trap (SLc v_{14}))) $\wedge$
($\forall v_{18}\ v_{16}$. $P\ v_{18}$ (exec (ESCc v_{16}))) $\wedge$
P MOVE_TO_ORP (exec (SLc crossLD)) $\wedge$
P CONDUCT_ORP (exec (SLc crossLD)) $\wedge$
P MOVE_TO_PB (exec (SLc crossLD)) $\wedge$
P CONDUCT_PB (exec (SLc crossLD)) $\wedge$
P COMPLETE_PB (exec (SLc crossLD)) $\wedge$
P PLAN_PB (exec (SLc conductORP)) $\wedge$
P CONDUCT_ORP (exec (SLc conductORP)) $\wedge$
P MOVE_TO_PB (exec (SLc conductORP)) $\wedge$
P CONDUCT_PB (exec (SLc conductORP)) $\wedge$
P COMPLETE_PB (exec (SLc conductORP)) $\wedge$
P PLAN_PB (exec (SLc moveToPB)) $\wedge$
P MOVE_TO_ORP (exec (SLc moveToPB)) $\wedge$
P MOVE_TO_PB (exec (SLc moveToPB)) $\wedge$
P CONDUCT_PB (exec (SLc moveToPB)) $\wedge$
P COMPLETE_PB (exec (SLc moveToPB)) $\wedge$
P PLAN_PB (exec (SLc conductPB)) $\wedge$
P MOVE_TO_ORP (exec (SLc conductPB)) $\wedge$
P CONDUCT_ORP (exec (SLc conductPB)) $\wedge$
P CONDUCT_PB (exec (SLc conductPB)) $\wedge$
P COMPLETE_PB (exec (SLc conductPB)) $\wedge$
P PLAN_PB (exec (SLc completePB)) $\wedge$
P MOVE_TO_ORP (exec (SLc completePB)) $\wedge$
P CONDUCT_ORP (exec (SLc completePB)) $\wedge$
P MOVE_TO_PB (exec (SLc completePB)) $\wedge$
P COMPLETE_PB (exec (SLc completePB)) $\wedge$
P COMPLETE_PB (exec (SLc incomplete)) $\Rightarrow$
$\forall v\ v_1$. $P\ v\ v_1$

[PBOut_def]

$\vdash$ (PBOut PLAN_PB (exec (SLc crossLD)) = MoveToORP) $\wedge$
(PBOut PLAN_PB (exec (SLc incomplete)) = PlanPB) $\wedge$

```
(PBOut MOVE_TO_ORP (exec (SLc conductORP)) = ConductORP) ∧
(PBOut MOVE_TO_ORP (exec (SLc incomplete)) = MoveToORP) ∧
(PBOut CONDUCT_ORP (exec (SLc moveToPB)) = MoveToPB) ∧
(PBOut CONDUCT_ORP (exec (SLc incomplete)) = ConductORP) ∧
(PBOut MOVE_TO_PB (exec (SLc conductPB)) = ConductPB) ∧
(PBOut MOVE_TO_PB (exec (SLc incomplete)) = MoveToPB) ∧
(PBOut CONDUCT_PB (exec (SLc completePB)) = CompletePB) ∧
(PBOut CONDUCT_PB (exec (SLc incomplete)) = ConductPB) ∧
(PBOut PLAN_PB (trap (SLc crossLD)) = PlanPB) ∧
(PBOut PLAN_PB (trap (SLc incomplete)) = PlanPB) ∧
(PBOut MOVE_TO_ORP (trap (SLc moveToORP)) = MoveToORP) ∧
(PBOut CONDUCT_ORP (trap (SLc moveToPB)) = ConductORP) ∧
(PBOut CONDUCT_ORP (trap (SLc incomplete)) = ConductORP) ∧
(PBOut MOVE_TO_PB (trap (SLc conductPB)) = MoveToPB) ∧
(PBOut MOVE_TO_PB (trap (SLc incomplete)) = MoveToPB) ∧
(PBOut CONDUCT_PB (trap (SLc completePB)) = ConductPB) ∧
(PBOut CONDUCT_PB (trap (SLc incomplete)) = ConductPB) ∧
(PBOut PLAN_PB (discard (SLc crossLD)) = unAuthenticated) ∧
(PBOut MOVE_TO_ORP (discard (SLc moveToORP)) =
 unAuthenticated) ∧
(PBOut CONDUCT_ORP (discard (SLc moveToPB)) =
 unAuthenticated) ∧
(PBOut MOVE_TO_PB (discard (SLc conductPB)) =
 unAuthenticated) ∧
(PBOut CONDUCT_PB (discard (SLc completePB)) =
 unAuthenticated)
```

[PBOut_ind]

```
⊢ ∀P.
    P PLAN_PB (exec (SLc crossLD)) ∧
    P PLAN_PB (exec (SLc incomplete)) ∧
    P MOVE_TO_ORP (exec (SLc conductORP)) ∧
    P MOVE_TO_ORP (exec (SLc incomplete)) ∧
    P CONDUCT_ORP (exec (SLc moveToPB)) ∧
    P CONDUCT_ORP (exec (SLc incomplete)) ∧
    P MOVE_TO_PB (exec (SLc conductPB)) ∧
    P MOVE_TO_PB (exec (SLc incomplete)) ∧
    P CONDUCT_PB (exec (SLc completePB)) ∧
    P CONDUCT_PB (exec (SLc incomplete)) ∧
    P PLAN_PB (trap (SLc crossLD)) ∧
    P PLAN_PB (trap (SLc incomplete)) ∧
    (∀moveToORP. P MOVE_TO_ORP (trap (SLc moveToORP))) ∧
    P CONDUCT_ORP (trap (SLc moveToPB)) ∧
    P CONDUCT_ORP (trap (SLc incomplete)) ∧
    P MOVE_TO_PB (trap (SLc conductPB)) ∧
    P MOVE_TO_PB (trap (SLc incomplete)) ∧
    P CONDUCT_PB (trap (SLc completePB)) ∧
    P CONDUCT_PB (trap (SLc incomplete)) ∧
    P PLAN_PB (discard (SLc crossLD)) ∧
```

$(\forall\, moveToORP.\ P$ MOVE_TO_ORP (discard (SLc $moveToORP$))) $\land$
P CONDUCT_ORP (discard (SLc moveToPB)) $\land$
P MOVE_TO_PB (discard (SLc conductPB)) $\land$
P CONDUCT_PB (discard (SLc completePB)) $\land$
$(\forall\, v_8\ v_6.\ P\ v_8$ (discard (ESCc v_6))) $\land$
P PLAN_PB (discard (SLc conductORP)) $\land$
P PLAN_PB (discard (SLc moveToPB)) $\land$
P PLAN_PB (discard (SLc conductPB)) $\land$
P PLAN_PB (discard (SLc completePB)) $\land$
P PLAN_PB (discard (SLc incomplete)) $\land$
P CONDUCT_ORP (discard (SLc crossLD)) $\land$
P CONDUCT_ORP (discard (SLc conductORP)) $\land$
P CONDUCT_ORP (discard (SLc conductPB)) $\land$
P CONDUCT_ORP (discard (SLc completePB)) $\land$
P CONDUCT_ORP (discard (SLc incomplete)) $\land$
P MOVE_TO_PB (discard (SLc crossLD)) $\land$
P MOVE_TO_PB (discard (SLc conductORP)) $\land$
P MOVE_TO_PB (discard (SLc moveToPB)) $\land$
P MOVE_TO_PB (discard (SLc completePB)) $\land$
P MOVE_TO_PB (discard (SLc incomplete)) $\land$
P CONDUCT_PB (discard (SLc crossLD)) $\land$
P CONDUCT_PB (discard (SLc conductORP)) $\land$
P CONDUCT_PB (discard (SLc moveToPB)) $\land$
P CONDUCT_PB (discard (SLc conductPB)) $\land$
P CONDUCT_PB (discard (SLc incomplete)) $\land$
$(\forall\, v_9.\ P$ COMPLETE_PB (discard (SLc v_9))) $\land$
$(\forall\, v_{13}\ v_{11}.\ P\ v_{13}$ (trap (ESCc v_{11}))) $\land$
P PLAN_PB (trap (SLc conductORP)) $\land$
P PLAN_PB (trap (SLc moveToPB)) $\land$
P PLAN_PB (trap (SLc conductPB)) $\land$
P PLAN_PB (trap (SLc completePB)) $\land$
P CONDUCT_ORP (trap (SLc crossLD)) $\land$
P CONDUCT_ORP (trap (SLc conductORP)) $\land$
P CONDUCT_ORP (trap (SLc conductPB)) $\land$
P CONDUCT_ORP (trap (SLc completePB)) $\land$
P MOVE_TO_PB (trap (SLc crossLD)) $\land$
P MOVE_TO_PB (trap (SLc conductORP)) $\land$
P MOVE_TO_PB (trap (SLc moveToPB)) $\land$
P MOVE_TO_PB (trap (SLc completePB)) $\land$
P CONDUCT_PB (trap (SLc crossLD)) $\land$
P CONDUCT_PB (trap (SLc conductORP)) $\land$
P CONDUCT_PB (trap (SLc moveToPB)) $\land$
P CONDUCT_PB (trap (SLc conductPB)) $\land$
$(\forall\, v_{14}.\ P$ COMPLETE_PB (trap (SLc v_{14}))) $\land$
$(\forall\, v_{18}\ v_{16}.\ P\ v_{18}$ (exec (ESCc v_{16}))) $\land$
P MOVE_TO_ORP (exec (SLc crossLD)) $\land$
P CONDUCT_ORP (exec (SLc crossLD)) $\land$
P MOVE_TO_PB (exec (SLc crossLD)) $\land$
P CONDUCT_PB (exec (SLc crossLD)) $\land$

```
P COMPLETE_PB (exec (SLc crossLD)) ∧
P PLAN_PB (exec (SLc conductORP)) ∧
P CONDUCT_ORP (exec (SLc conductORP)) ∧
P MOVE_TO_PB (exec (SLc conductORP)) ∧
P CONDUCT_PB (exec (SLc conductORP)) ∧
P COMPLETE_PB (exec (SLc conductORP)) ∧
P PLAN_PB (exec (SLc moveToPB)) ∧
P MOVE_TO_ORP (exec (SLc moveToPB)) ∧
P MOVE_TO_PB (exec (SLc moveToPB)) ∧
P CONDUCT_PB (exec (SLc moveToPB)) ∧
P COMPLETE_PB (exec (SLc moveToPB)) ∧
P PLAN_PB (exec (SLc conductPB)) ∧
P MOVE_TO_ORP (exec (SLc conductPB)) ∧
P CONDUCT_ORP (exec (SLc conductPB)) ∧
P CONDUCT_PB (exec (SLc conductPB)) ∧
P COMPLETE_PB (exec (SLc conductPB)) ∧
P PLAN_PB (exec (SLc completePB)) ∧
P MOVE_TO_ORP (exec (SLc completePB)) ∧
P CONDUCT_ORP (exec (SLc completePB)) ∧
P MOVE_TO_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc completePB)) ∧
P COMPLETE_PB (exec (SLc incomplete)) ⇒
∀ v v1. P v v1
```

3 PBType Theory

Built: 02 July 2017

Parent Theories: indexedLists, patternMatches

3.1 Datatypes

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

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

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

stateRole = PlatoonLeader
```

3.2 Theorems

[slCommand_distinct_clauses]

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

crossLD $\neq$ incomplete $\wedge$ conductORP $\neq$ moveToPB $\wedge$
conductORP $\neq$ conductPB $\wedge$ conductORP $\neq$ completePB $\wedge$
conductORP $\neq$ incomplete $\wedge$ moveToPB $\neq$ conductPB $\wedge$
moveToPB $\neq$ completePB $\wedge$ moveToPB $\neq$ incomplete $\wedge$
conductPB $\neq$ completePB $\wedge$ conductPB $\neq$ incomplete $\wedge$
completePB $\neq$ incomplete

[slOutput_distinct_clauses]

$\vdash$ PlanPB $\neq$ MoveToORP $\wedge$ PlanPB $\neq$ ConductORP $\wedge$
PlanPB $\neq$ MoveToPB $\wedge$ PlanPB $\neq$ ConductPB $\wedge$
PlanPB $\neq$ CompletePB $\wedge$ PlanPB $\neq$ unAuthenticated $\wedge$
MoveToORP $\neq$ ConductORP $\wedge$ MoveToORP $\neq$ MoveToPB $\wedge$
MoveToORP $\neq$ ConductPB $\wedge$ MoveToORP $\neq$ CompletePB $\wedge$
MoveToORP $\neq$ unAuthenticated $\wedge$ ConductORP $\neq$ MoveToPB $\wedge$
ConductORP $\neq$ ConductPB $\wedge$ ConductORP $\neq$ CompletePB $\wedge$
ConductORP $\neq$ unAuthenticated $\wedge$ MoveToPB $\neq$ ConductPB $\wedge$
MoveToPB $\neq$ CompletePB $\wedge$ MoveToPB $\neq$ unAuthenticated $\wedge$
ConductPB $\neq$ CompletePB $\wedge$ ConductPB $\neq$ unAuthenticated $\wedge$
CompletePB $\neq$ unAuthenticated

[slState_distinct_clauses]

$\vdash$ PLAN_PB $\neq$ MOVE_TO_ORP $\wedge$ PLAN_PB $\neq$ CONDUCT_ORP $\wedge$
PLAN_PB $\neq$ MOVE_TO_PB $\wedge$ PLAN_PB $\neq$ CONDUCT_PB $\wedge$
PLAN_PB $\neq$ COMPLETE_PB $\wedge$ MOVE_TO_ORP $\neq$ CONDUCT_ORP $\wedge$
MOVE_TO_ORP $\neq$ MOVE_TO_PB $\wedge$ MOVE_TO_ORP $\neq$ CONDUCT_PB $\wedge$
MOVE_TO_ORP $\neq$ COMPLETE_PB $\wedge$ CONDUCT_ORP $\neq$ MOVE_TO_PB $\wedge$
CONDUCT_ORP $\neq$ CONDUCT_PB $\wedge$ CONDUCT_ORP $\neq$ COMPLETE_PB $\wedge$
MOVE_TO_PB $\neq$ CONDUCT_PB $\wedge$ MOVE_TO_PB $\neq$ COMPLETE_PB $\wedge$
CONDUCT_PB $\neq$ COMPLETE_PB

Index