Contents

1 projectTypes Theory

Built: 27 December 2018
Parent Theories: indexedLists, patternMatches

1.1 Datatypes

commands = SquadLeaderCOM squadLeaderCom | OmniCOM omniCom

omniCom = none | omniNA

output = RtPosition | RtOrient | RtAlert | Complete
| NoActionTaken | UnAuthenticated | UnAuthorized

principal = SquadLeader | Omni

squadLeaderCom = rtPosition | rtOrient | rtAlert | complete

state = RT_FORM | RT_POSITION | RT_ORIENT | RT_ALERT | COMPLETE

1.2 Theorems

[commands_distinct_clauses]

$\vdash \forall a'\ a.$ SquadLeaderCOM $a \neq$ OmniCOM a'

[commands_one_one]

$\vdash$ $(\forall a\ a'.$ (SquadLeaderCOM a = SquadLeaderCOM $a') \iff (a = a')) \wedge$
$\forall a\ a'.$ (OmniCOM a = OmniCOM $a') \iff (a = a')$

[omniCom_distinct_clauses]

$\vdash$ none $\neq$ omniNA

[output_distinct_clauses]

$\vdash$ RtPosition $\neq$ RtOrient $\wedge$ RtPosition $\neq$ RtAlert $\wedge$
RtPosition $\neq$ Complete $\wedge$ RtPosition $\neq$ NoActionTaken $\wedge$
RtPosition $\neq$ UnAuthenticated $\wedge$ RtPosition $\neq$ UnAuthorized $\wedge$
RtOrient $\neq$ RtAlert $\wedge$ RtOrient $\neq$ Complete $\wedge$
RtOrient $\neq$ NoActionTaken $\wedge$ RtOrient $\neq$ UnAuthenticated $\wedge$
RtOrient $\neq$ UnAuthorized $\wedge$ RtAlert $\neq$ Complete $\wedge$
RtAlert $\neq$ NoActionTaken $\wedge$ RtAlert $\neq$ UnAuthenticated $\wedge$
RtAlert $\neq$ UnAuthorized $\wedge$ Complete $\neq$ NoActionTaken $\wedge$
Complete $\neq$ UnAuthenticated $\wedge$ Complete $\neq$ UnAuthorized $\wedge$
NoActionTaken $\neq$ UnAuthenticated $\wedge$
NoActionTaken $\neq$ UnAuthorized $\wedge$ UnAuthenticated $\neq$ UnAuthorized

[principal_distinct_clauses]

$\vdash$ SquadLeader $\neq$ Omni

[squadLeaderCom_distinct_clauses]

⊢ rtPosition ≠ rtOrient ∧ rtPosition ≠ rtAlert ∧
rtPosition ≠ complete ∧ rtOrient ≠ rtAlert ∧
rtOrient ≠ complete ∧ rtAlert ≠ complete

[state_distinct_clauses]

⊢ RT_FORM ≠ RT_POSITION ∧ RT_FORM ≠ RT_ORIENT ∧
RT_FORM ≠ RT_ALERT ∧ RT_FORM ≠ COMPLETE ∧
RT_POSITION ≠ RT_ORIENT ∧ RT_POSITION ≠ RT_ALERT ∧
RT_POSITION ≠ COMPLETE ∧ RT_ORIENT ≠ RT_ALERT ∧
RT_ORIENT ≠ COMPLETE ∧ RT_ALERT ≠ COMPLETE

2 projectUtilities Theory

Built: 27 December 2018
Parent Theories: projectTypes, satList

2.1 Theorems

[getOmniCOM_def]

⊢ (getOmniCOM [] = NONE) ∧
(∀ *xs* *cmd*.
getOmniCOM (SOME (OmniCOM *cmd*)::*xs*) =
SOME (OmniCOM *cmd*)) ∧
(∀ *xs*. getOmniCOM (NONE::*xs*) = getOmniCOM *xs*) ∧
∀ *xs* v_4.
getOmniCOM (SOME (SquadLeaderCOM v_4)::*xs*) = getOmniCOM *xs*

[getOmniCOM_ind]

⊢ ∀ *P*.
P [] ∧ (∀ *cmd* *xs*. *P* (SOME (OmniCOM *cmd*)::*xs*)) ∧
(∀ *xs*. *P* *xs* ⇒ *P* (NONE::*xs*)) ∧
(∀ v_4 *xs*. *P* *xs* ⇒ *P* (SOME (SquadLeaderCOM v_4)::*xs*)) ⇒
∀ *v*. *P* *v*

[getOmniCOMx_def]

⊢ (getOmniCOMx [] = NONE) ∧
(∀ *xs* *cmd*.
getOmniCOMx
(Name Omni says prop (SOME (OmniCOM *cmd*))::*xs*) =
SOME (OmniCOM *cmd*)) ∧
(∀ *xs*. getOmniCOMx (TT::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs*. getOmniCOMx (FF::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_2. getOmniCOMx (prop v_2::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_3. getOmniCOMx (notf v_3::*xs*) = getOmniCOMx *xs*) ∧
(∀ *xs* v_5 v_4. getOmniCOMx (v_4 andf v_5::*xs*) = getOmniCOMx *xs*) ∧

```
(∀ xs v7 v6. getOmniCOMx (v6 orf v7::xs) = getOmniCOMx xs) ∧
(∀ xs v9 v8. getOmniCOMx (v8 impf v9::xs) = getOmniCOMx xs) ∧
(∀ xs v11 v10.
   getOmniCOMx (v10 eqf v11::xs) = getOmniCOMx xs) ∧
(∀ xs v12. getOmniCOMx (v12 says TT::xs) = getOmniCOMx xs) ∧
(∀ xs v12. getOmniCOMx (v12 says FF::xs) = getOmniCOMx xs) ∧
(∀ xs v134.
   getOmniCOMx (Name v134 says prop NONE::xs) =
   getOmniCOMx xs) ∧
(∀ xs v144.
   getOmniCOMx (Name SquadLeader says prop (SOME v144)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v146.
   getOmniCOMx
     (Name Omni says prop (SOME (SquadLeaderCOM v146))::xs) =
   getOmniCOMx xs) ∧
(∀ xs v68 v136 v135.
   getOmniCOMx (v135 meet v136 says prop v68::xs) =
   getOmniCOMx xs) ∧
(∀ xs v68 v138 v137.
   getOmniCOMx (v137 quoting v138 says prop v68::xs) =
   getOmniCOMx xs) ∧
(∀ xs v69 v12.
   getOmniCOMx (v12 says notf v69::xs) = getOmniCOMx xs) ∧
(∀ xs v71 v70 v12.
   getOmniCOMx (v12 says (v70 andf v71)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v73 v72 v12.
   getOmniCOMx (v12 says (v72 orf v73)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v75 v74 v12.
   getOmniCOMx (v12 says (v74 impf v75)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v77 v76 v12.
   getOmniCOMx (v12 says (v76 eqf v77)::xs) =
   getOmniCOMx xs) ∧
(∀ xs v79 v78 v12.
   getOmniCOMx (v12 says v78 says v79::xs) =
   getOmniCOMx xs) ∧
(∀ xs v81 v80 v12.
   getOmniCOMx (v12 says v80 speaks_for v81::xs) =
   getOmniCOMx xs) ∧
(∀ xs v83 v82 v12.
   getOmniCOMx (v12 says v82 controls v83::xs) =
   getOmniCOMx xs) ∧
(∀ xs v86 v85 v84 v12.
   getOmniCOMx (v12 says reps v84 v85 v86::xs) =
   getOmniCOMx xs) ∧
(∀ xs v88 v87 v12.
```

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

[getOmniCOMx_ind]

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

$(\forall\, v134\ xs.\ P\ xs \Rightarrow P\ (\texttt{Name}\ v134\ \texttt{says prop NONE::}xs)) \wedge$
$(\forall\, v144\ xs.$
$\quad P\ xs \Rightarrow P\ (\texttt{Name SquadLeader says prop (SOME}\ v144\texttt{)::}xs)) \wedge$
$(\forall\, v146\ xs.$
$\quad P\ xs \Rightarrow$
$\quad P$
$\quad\quad (\texttt{Name Omni says prop (SOME (SquadLeaderCOM}\ v146\texttt{))::}$
$\quad\quad\quad xs)) \wedge$
$(\forall\, v135\ v136\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P\ (v135\ \texttt{meet}\ v136\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge$
$(\forall\, v137\ v138\ v_{68}\ xs.$
$\quad P\ xs \Rightarrow P\ (v137\ \texttt{quoting}\ v138\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says notf}\ v_{69}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{70}\ \texttt{andf}\ v_{71}\texttt{)::}xs)) \wedge$
$(\forall\, v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{72}\ \texttt{orf}\ v_{73}\texttt{)::}xs)) \wedge$
$(\forall\, v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{74}\ \texttt{impf}\ v_{75}\texttt{)::}xs)) \wedge$
$(\forall\, v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{76}\ \texttt{eqf}\ v_{77}\texttt{)::}xs)) \wedge$
$(\forall\, v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{78}\ \texttt{says}\ v_{79}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{80}\ v_{81}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{80}\ \texttt{speaks\_for}\ v_{81}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{82}\ v_{83}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{82}\ \texttt{controls}\ v_{83}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs.$
$\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says reps}\ v_{84}\ v_{85}\ v_{86}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{87}\ \texttt{domi}\ v_{88}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{89}\ \texttt{eqi}\ v_{90}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{91}\ \texttt{doms}\ v_{92}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{93}\ \texttt{eqs}\ v_{94}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{95}\ \texttt{eqn}\ v_{96}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{97}\ \texttt{lte}\ v_{98}\texttt{::}xs)) \wedge$
$(\forall\, v_{12}\ v_{99}\ v100\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{99}\ \texttt{lt}\ v100\texttt{::}xs)) \wedge$
$(\forall\, v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P\ (v_{14}\ \texttt{speaks\_for}\ v_{15}\texttt{::}xs)) \wedge$
$(\forall\, v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P\ (v_{16}\ \texttt{controls}\ v_{17}\texttt{::}xs)) \wedge$
$(\forall\, v_{18}\ v_{19}\ v_{20}\ xs.\ P\ xs \Rightarrow P\ (\texttt{reps}\ v_{18}\ v_{19}\ v_{20}\texttt{::}xs)) \wedge$
$(\forall\, v_{21}\ v_{22}\ xs.\ P\ xs \Rightarrow P\ (v_{21}\ \texttt{domi}\ v_{22}\texttt{::}xs)) \wedge$
$(\forall\, v_{23}\ v_{24}\ xs.\ P\ xs \Rightarrow P\ (v_{23}\ \texttt{eqi}\ v_{24}\texttt{::}xs)) \wedge$
$(\forall\, v_{25}\ v_{26}\ xs.\ P\ xs \Rightarrow P\ (v_{25}\ \texttt{doms}\ v_{26}\texttt{::}xs)) \wedge$
$(\forall\, v_{27}\ v_{28}\ xs.\ P\ xs \Rightarrow P\ (v_{27}\ \texttt{eqs}\ v_{28}\texttt{::}xs)) \wedge$
$(\forall\, v_{29}\ v_{30}\ xs.\ P\ xs \Rightarrow P\ (v_{29}\ \texttt{eqn}\ v_{30}\texttt{::}xs)) \wedge$
$(\forall\, v_{31}\ v_{32}\ xs.\ P\ xs \Rightarrow P\ (v_{31}\ \texttt{lte}\ v_{32}\texttt{::}xs)) \wedge$
$(\forall\, v_{33}\ v_{34}\ xs.\ P\ xs \Rightarrow P\ (v_{33}\ \texttt{lt}\ v_{34}\texttt{::}xs)) \Rightarrow$
$\forall\, v.\ P\ v$

[getSquadLeaderCOM_def]

$\vdash$ $(\texttt{getSquadLeaderCOM []} = \texttt{NONE}) \wedge$
$(\forall\, xs\ cmd.$
$\quad \texttt{getSquadLeaderCOM (SOME (SquadLeaderCOM}\ cmd\texttt{)::}xs\texttt{)} =$
$\quad \texttt{SOME (SquadLeaderCOM}\ cmd\texttt{))} \wedge$
$(\forall\, xs.\ \texttt{getSquadLeaderCOM (NONE::}xs\texttt{)} = \texttt{getSquadLeaderCOM}\ xs) \wedge$

```
  ∀xs v5.
    getSquadLeaderCOM (SOME (OmniCOM v5)::xs) =
    getSquadLeaderCOM xs
```

[getSquadLeaderCOM_ind]

```
⊢ ∀P.
    P [] ∧ (∀cmd xs. P (SOME (SquadLeaderCOM cmd)::xs)) ∧
    (∀xs. P xs ⇒ P (NONE::xs)) ∧
    (∀v5 xs. P xs ⇒ P (SOME (OmniCOM v5)::xs)) ⇒
    ∀v. P v
```

[getSquadLeaderCOMx_def]

```
⊢ (getSquadLeaderCOMx [] = NONE) ∧
  (∀xs cmd.
     getSquadLeaderCOMx
       (Name SquadLeader says
        prop (SOME (SquadLeaderCOM cmd))::xs) =
     SOME (SquadLeaderCOM cmd)) ∧
  (∀xs. getSquadLeaderCOMx (TT::xs) = getSquadLeaderCOMx xs) ∧
  (∀xs. getSquadLeaderCOMx (FF::xs) = getSquadLeaderCOMx xs) ∧
  (∀xs v2.
     getSquadLeaderCOMx (prop v2::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v3.
     getSquadLeaderCOMx (notf v3::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v5 v4.
     getSquadLeaderCOMx (v4 andf v5::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v7 v6.
     getSquadLeaderCOMx (v6 orf v7::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v9 v8.
     getSquadLeaderCOMx (v8 impf v9::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v11 v10.
     getSquadLeaderCOMx (v10 eqf v11::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v12.
     getSquadLeaderCOMx (v12 says TT::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v12.
     getSquadLeaderCOMx (v12 says FF::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v134.
     getSquadLeaderCOMx (Name v134 says prop NONE::xs) =
     getSquadLeaderCOMx xs) ∧
  (∀xs v147.
     getSquadLeaderCOMx
```

```
     (Name SquadLeader says prop (SOME (OmniCOM v147))::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v144 .
   getSquadLeaderCOMx (Name Omni says prop (SOME v144)::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v68 v136 v135 .
   getSquadLeaderCOMx (v135 meet v136 says prop v68::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v68 v138 v137 .
   getSquadLeaderCOMx (v137 quoting v138 says prop v68::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v69 v12 .
   getSquadLeaderCOMx (v12 says notf v69::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v71 v70 v12 .
   getSquadLeaderCOMx (v12 says (v70 andf v71)::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v73 v72 v12 .
   getSquadLeaderCOMx (v12 says (v72 orf v73)::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v75 v74 v12 .
   getSquadLeaderCOMx (v12 says (v74 impf v75)::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v77 v76 v12 .
   getSquadLeaderCOMx (v12 says (v76 eqf v77)::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v79 v78 v12 .
   getSquadLeaderCOMx (v12 says v78 says v79::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v81 v80 v12 .
   getSquadLeaderCOMx (v12 says v80 speaks_for v81::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v83 v82 v12 .
   getSquadLeaderCOMx (v12 says v82 controls v83::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v86 v85 v84 v12 .
   getSquadLeaderCOMx (v12 says reps v84 v85 v86::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v88 v87 v12 .
   getSquadLeaderCOMx (v12 says v87 domi v88::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v90 v89 v12 .
   getSquadLeaderCOMx (v12 says v89 eqi v90::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v92 v91 v12 .
   getSquadLeaderCOMx (v12 says v91 doms v92::xs) =
   getSquadLeaderCOMx xs) ∧
(∀ xs v94 v93 v12 .
   getSquadLeaderCOMx (v12 says v93 eqs v94::xs) =
```

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

[getSquadLeaderCOMx_ind]

```
⊢ ∀ P .
    P [] ∧
    (∀ cmd xs .
       P
         (Name SquadLeader says
          prop (SOME (SquadLeaderCOM cmd))::xs)) ∧
    (∀ xs . P xs ⇒ P (TT::xs)) ∧ (∀ xs . P xs ⇒ P (FF::xs)) ∧
    (∀ v2 xs . P xs ⇒ P (prop v2::xs)) ∧
```

$$
\begin{array}{l}
(\forall\, v_3\ xs.\ P\ xs \Rightarrow P\ (\texttt{notf}\ v_3\texttt{::}xs)) \wedge \\
(\forall\, v_4\ v_5\ xs.\ P\ xs \Rightarrow P\ (v_4\ \texttt{andf}\ v_5\texttt{::}xs)) \wedge \\
(\forall\, v_6\ v_7\ xs.\ P\ xs \Rightarrow P\ (v_6\ \texttt{orf}\ v_7\texttt{::}xs)) \wedge \\
(\forall\, v_8\ v_9\ xs.\ P\ xs \Rightarrow P\ (v_8\ \texttt{impf}\ v_9\texttt{::}xs)) \wedge \\
(\forall\, v_{10}\ v_{11}\ xs.\ P\ xs \Rightarrow P\ (v_{10}\ \texttt{eqf}\ v_{11}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says TT::}xs)) \wedge \\
(\forall\, v_{12}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says FF::}xs)) \wedge \\
(\forall\, \mathit{v134}\ xs.\ P\ xs \Rightarrow P\ (\texttt{Name}\ \mathit{v134}\ \texttt{says prop NONE::}xs)) \wedge \\
(\forall\, \mathit{v147}\ xs. \\
\quad P\ xs \Rightarrow \\
\quad P \\
\qquad (\texttt{Name SquadLeader says prop (SOME (OmniCOM}\ \mathit{v147}\texttt{))::} \\
\qquad\quad xs)) \wedge \\
(\forall\, \mathit{v144}\ xs. \\
\quad P\ xs \Rightarrow P\ (\texttt{Name Omni says prop (SOME}\ \mathit{v144}\texttt{)::}xs)) \wedge \\
(\forall\, \mathit{v135}\ \mathit{v136}\ v_{68}\ xs. \\
\quad P\ xs \Rightarrow P\ (\mathit{v135}\ \texttt{meet}\ \mathit{v136}\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge \\
(\forall\, \mathit{v137}\ \mathit{v138}\ v_{68}\ xs. \\
\quad P\ xs \Rightarrow P\ (\mathit{v137}\ \texttt{quoting}\ \mathit{v138}\ \texttt{says prop}\ v_{68}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{69}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says notf}\ v_{69}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{70}\ v_{71}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{70}\ \texttt{andf}\ v_{71}\texttt{)::}xs)) \wedge \\
(\forall\, v_{12}\ v_{72}\ v_{73}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{72}\ \texttt{orf}\ v_{73}\texttt{)::}xs)) \wedge \\
(\forall\, v_{12}\ v_{74}\ v_{75}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{74}\ \texttt{impf}\ v_{75}\texttt{)::}xs)) \wedge \\
(\forall\, v_{12}\ v_{76}\ v_{77}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says (}v_{76}\ \texttt{eqf}\ v_{77}\texttt{)::}xs)) \wedge \\
(\forall\, v_{12}\ v_{78}\ v_{79}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{78}\ \texttt{says}\ v_{79}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{80}\ v_{81}\ xs. \\
\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{80}\ \texttt{speaks\_for}\ v_{81}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{82}\ v_{83}\ xs. \\
\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{82}\ \texttt{controls}\ v_{83}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{84}\ v_{85}\ v_{86}\ xs. \\
\quad P\ xs \Rightarrow P\ (v_{12}\ \texttt{says reps}\ v_{84}\ v_{85}\ v_{86}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{87}\ v_{88}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{87}\ \texttt{domi}\ v_{88}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{89}\ v_{90}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{89}\ \texttt{eqi}\ v_{90}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{91}\ v_{92}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{91}\ \texttt{doms}\ v_{92}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{93}\ v_{94}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{93}\ \texttt{eqs}\ v_{94}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{95}\ v_{96}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{95}\ \texttt{eqn}\ v_{96}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{97}\ v_{98}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{97}\ \texttt{lte}\ v_{98}\texttt{::}xs)) \wedge \\
(\forall\, v_{12}\ v_{99}\ \mathit{v100}\ xs.\ P\ xs \Rightarrow P\ (v_{12}\ \texttt{says}\ v_{99}\ \texttt{lt}\ \mathit{v100}\texttt{::}xs)) \wedge \\
(\forall\, v_{14}\ v_{15}\ xs.\ P\ xs \Rightarrow P\ (v_{14}\ \texttt{speaks\_for}\ v_{15}\texttt{::}xs)) \wedge \\
(\forall\, v_{16}\ v_{17}\ xs.\ P\ xs \Rightarrow P\ (v_{16}\ \texttt{controls}\ v_{17}\texttt{::}xs)) \wedge \\
(\forall\, v_{18}\ v_{19}\ v_{20}\ xs.\ P\ xs \Rightarrow P\ (\texttt{reps}\ v_{18}\ v_{19}\ v_{20}\texttt{::}xs)) \wedge \\
(\forall\, v_{21}\ v_{22}\ xs.\ P\ xs \Rightarrow P\ (v_{21}\ \texttt{domi}\ v_{22}\texttt{::}xs)) \wedge \\
(\forall\, v_{23}\ v_{24}\ xs.\ P\ xs \Rightarrow P\ (v_{23}\ \texttt{eqi}\ v_{24}\texttt{::}xs)) \wedge \\
(\forall\, v_{25}\ v_{26}\ xs.\ P\ xs \Rightarrow P\ (v_{25}\ \texttt{doms}\ v_{26}\texttt{::}xs)) \wedge \\
(\forall\, v_{27}\ v_{28}\ xs.\ P\ xs \Rightarrow P\ (v_{27}\ \texttt{eqs}\ v_{28}\texttt{::}xs)) \wedge \\
(\forall\, v_{29}\ v_{30}\ xs.\ P\ xs \Rightarrow P\ (v_{29}\ \texttt{eqn}\ v_{30}\texttt{::}xs)) \wedge \\
(\forall\, v_{31}\ v_{32}\ xs.\ P\ xs \Rightarrow P\ (v_{31}\ \texttt{lte}\ v_{32}\texttt{::}xs)) \wedge \\
(\forall\, v_{33}\ v_{34}\ xs.\ P\ xs \Rightarrow P\ (v_{33}\ \texttt{lt}\ v_{34}\texttt{::}xs)) \Rightarrow \\
\forall\, v.\ P\ v
\end{array}
$$

3 projectSM Theory

Built: 27 December 2018
Parent Theories: projectUtilities, ssm

3.1 Theorems

[NOut_def]

```
⊢ (NOut RT_FORM (exec x) =
   if
     getSquadLeaderCOM x = SOME (SquadLeaderCOM rtPosition)
   then
     RtPosition
   else NoActionTaken) ∧
  (NOut RT_POSITION (exec x) =
   if getSquadLeaderCOM x = SOME (SquadLeaderCOM rtOrient) then
     RtOrient
   else NoActionTaken) ∧
  (NOut RT_ORIENT (exec x) =
   if getSquadLeaderCOM x = SOME (SquadLeaderCOM rtAlert) then
     RtAlert
   else NoActionTaken) ∧
  (NOut RT_ALERT (exec x) =
   if getSquadLeaderCOM x = SOME (SquadLeaderCOM complete) then
     Complete
   else NoActionTaken) ∧ (NOut s (trap v0) = UnAuthorized) ∧
  (NOut s (discard v1) = UnAuthenticated)
```

[NOut_ind]

```
⊢ ∀P.
    (∀x. P RT_FORM (exec x)) ∧ (∀x. P RT_POSITION (exec x)) ∧
    (∀x. P RT_ORIENT (exec x)) ∧ (∀x. P RT_ALERT (exec x)) ∧
    (∀s v0. P s (trap v0)) ∧ (∀s v1. P s (discard v1)) ∧
    (∀v6. P COMPLETE (exec v6)) ⇒
    ∀v v1. P v v1
```

[NS_def]

```
⊢ (NS RT_FORM (exec x) =
   if
     getSquadLeaderCOM x = SOME (SquadLeaderCOM rtPosition)
   then
     RT_POSITION
   else RT_FORM) ∧
  (NS RT_POSITION (exec x) =
   if getSquadLeaderCOM x = SOME (SquadLeaderCOM rtOrient) then
     RT_ORIENT
   else RT_POSITION) ∧
  (NS RT_ORIENT (exec x) =
```

```
    if getSquadLeaderCOM x = SOME (SquadLeaderCOM rtAlert) then
      RT_ALERT
    else RT_ORIENT) ∧
   (NS RT_ALERT (exec x) =
    if getSquadLeaderCOM x = SOME (SquadLeaderCOM complete) then
      COMPLETE
    else RT_ALERT) ∧ (NS s (trap v0) = s) ∧
   (NS s (discard v1) = s)
```

[NS_ind]

```
⊢ ∀P.
    (∀x. P RT_FORM (exec x)) ∧ (∀x. P RT_POSITION (exec x)) ∧
    (∀x. P RT_ORIENT (exec x)) ∧ (∀x. P RT_ALERT (exec x)) ∧
    (∀s v0. P s (trap v0)) ∧ (∀s v1. P s (discard v1)) ∧
    (∀v6. P COMPLETE (exec v6)) ⇒
    ∀v v1. P v v1
```

4 projectSecurity Theory

Built: 27 December 2018
Parent Theories: projectUtilities, ssm

4.1 Definitions

[globalAuth_def]

```
⊢ ∀x. globalAuth x = [TT]
```

[stateAuth_def]

```
⊢ ∀s x.
    stateAuth s x =
    if s = RT_FORM then
      if
        getSquadLeaderCOMx x = SOME (SquadLeaderCOM rtPosition)
      then
        [Name SquadLeader controls
         prop (SOME (SquadLeaderCOM rtPosition))]
      else [prop NONE]
    else if s = RT_POSITION then
      if
        getSquadLeaderCOMx x = SOME (SquadLeaderCOM rtOrient)
      then
        [Name SquadLeader controls
         prop (SOME (SquadLeaderCOM rtOrient))]
      else [prop NONE]
    else if s = RT_ORIENT then
      if
        getSquadLeaderCOMx x = SOME (SquadLeaderCOM rtAlert)
```

```
      then
        [Name SquadLeader controls
         prop (SOME (SquadLeaderCOM rtAlert))]
      else [prop NONE]
    else if s = RT_ALERT then
      if
        getSquadLeaderCOMx x = SOME (SquadLeaderCOM complete)
      then
        [Name SquadLeader controls
         prop (SOME (SquadLeaderCOM complete))]
      else [prop NONE]
    else [prop NONE]
```

4.2 Theorems

[authentication_def]

```
⊢ (authentication
     (Name SquadLeader says prop (SOME (SquadLeaderCOM x'))) ⟺
   T) ∧
  (authentication (Name Omni says prop (SOME (OmniCOM x))) ⟺
   T) ∧ (authentication TT ⟺ F) ∧ (authentication FF ⟺ F) ∧
  (authentication (prop v) ⟺ F) ∧
  (authentication (notf v1) ⟺ F) ∧
  (authentication (v2 andf v3) ⟺ F) ∧
  (authentication (v4 orf v5) ⟺ F) ∧
  (authentication (v6 impf v7) ⟺ F) ∧
  (authentication (v8 eqf v9) ⟺ F) ∧
  (authentication (Name v66 says TT) ⟺ F) ∧
  (authentication (Name v66 says FF) ⟺ F) ∧
  (authentication (Name v66 says prop NONE) ⟺ F) ∧
  (authentication
     (Name Omni says prop (SOME (SquadLeaderCOM v144))) ⟺ F) ∧
  (authentication
     (Name SquadLeader says prop (SOME (OmniCOM v145))) ⟺ F) ∧
  (authentication (Name v66 says notf v77) ⟺ F) ∧
  (authentication (Name v66 says (v78 andf v79)) ⟺ F) ∧
  (authentication (Name v66 says (v80 orf v81)) ⟺ F) ∧
  (authentication (Name v66 says (v82 impf v83)) ⟺ F) ∧
  (authentication (Name v66 says (v84 eqf v85)) ⟺ F) ∧
  (authentication (Name v66 says v86 says v87) ⟺ F) ∧
  (authentication (Name v66 says v88 speaks_for v89) ⟺ F) ∧
  (authentication (Name v66 says v90 controls v91) ⟺ F) ∧
  (authentication (Name v66 says reps v92 v93 v94) ⟺ F) ∧
  (authentication (Name v66 says v95 domi v96) ⟺ F) ∧
  (authentication (Name v66 says v97 eqi v98) ⟺ F) ∧
  (authentication (Name v66 says v99 doms v100) ⟺ F) ∧
  (authentication (Name v66 says v101 eqs v102) ⟺ F) ∧
  (authentication (Name v66 says v103 eqn v104) ⟺ F) ∧
  (authentication (Name v66 says v105 lte v106) ⟺ F) ∧
```

(authentication (Name v_{66} says *v107* lt *v108*) $\iff$ F) $\wedge$
(authentication (v_{67} meet v_{68} says v_{11}) $\iff$ F) $\wedge$
(authentication (v_{69} quoting v_{70} says v_{11}) $\iff$ F) $\wedge$
(authentication (v_{12} speaks_for v_{13}) $\iff$ F) $\wedge$
(authentication (v_{14} controls v_{15}) $\iff$ F) $\wedge$
(authentication (reps v_{16} v_{17} v_{18}) $\iff$ F) $\wedge$
(authentication (v_{19} domi v_{20}) $\iff$ F) $\wedge$
(authentication (v_{21} eqi v_{22}) $\iff$ F) $\wedge$
(authentication (v_{23} doms v_{24}) $\iff$ F) $\wedge$
(authentication (v_{25} eqs v_{26}) $\iff$ F) $\wedge$
(authentication (v_{27} eqn v_{28}) $\iff$ F) $\wedge$
(authentication (v_{29} lte v_{30}) $\iff$ F) $\wedge$
(authentication (v_{31} lt v_{32}) $\iff$ F)

[authentication_ind]

$\vdash \forall P.$
(∀ *x*.
P
(Name SquadLeader says
prop (SOME (SquadLeaderCOM *x*)))) $\wedge$
(∀ *x*. *P* (Name Omni says prop (SOME (OmniCOM *x*)))) $\wedge$ *P* TT $\wedge$
P FF $\wedge$ (∀ *v*. *P* (prop *v*)) $\wedge$ (∀ v_1. *P* (notf v_1)) $\wedge$
(∀ v_2 v_3. *P* (v_2 andf v_3)) $\wedge$ (∀ v_4 v_5. *P* (v_4 orf v_5)) $\wedge$
(∀ v_6 v_7. *P* (v_6 impf v_7)) $\wedge$ (∀ v_8 v_9. *P* (v_8 eqf v_9)) $\wedge$
(∀ v_{66}. *P* (Name v_{66} says TT)) $\wedge$
(∀ v_{66}. *P* (Name v_{66} says FF)) $\wedge$
(∀ v_{66}. *P* (Name v_{66} says prop NONE)) $\wedge$
(∀ *v144*.
P (Name Omni says prop (SOME (SquadLeaderCOM *v144*)))) $\wedge$
(∀ *v145*.
P (Name SquadLeader says prop (SOME (OmniCOM *v145*)))) $\wedge$
(∀ v_{66} v_{77}. *P* (Name v_{66} says notf v_{77})) $\wedge$
(∀ v_{66} v_{78} v_{79}. *P* (Name v_{66} says (v_{78} andf v_{79}))) $\wedge$
(∀ v_{66} v_{80} v_{81}. *P* (Name v_{66} says (v_{80} orf v_{81}))) $\wedge$
(∀ v_{66} v_{82} v_{83}. *P* (Name v_{66} says (v_{82} impf v_{83}))) $\wedge$
(∀ v_{66} v_{84} v_{85}. *P* (Name v_{66} says (v_{84} eqf v_{85}))) $\wedge$
(∀ v_{66} v_{86} v_{87}. *P* (Name v_{66} says v_{86} says v_{87})) $\wedge$
(∀ v_{66} v_{88} v_{89}. *P* (Name v_{66} says v_{88} speaks_for v_{89})) $\wedge$
(∀ v_{66} v_{90} v_{91}. *P* (Name v_{66} says v_{90} controls v_{91})) $\wedge$
(∀ v_{66} v_{92} v_{93} v_{94}. *P* (Name v_{66} says reps v_{92} v_{93} v_{94})) $\wedge$
(∀ v_{66} v_{95} v_{96}. *P* (Name v_{66} says v_{95} domi v_{96})) $\wedge$
(∀ v_{66} v_{97} v_{98}. *P* (Name v_{66} says v_{97} eqi v_{98})) $\wedge$
(∀ v_{66} v_{99} *v100*. *P* (Name v_{66} says v_{99} doms *v100*)) $\wedge$
(∀ v_{66} *v101* *v102*. *P* (Name v_{66} says *v101* eqs *v102*)) $\wedge$
(∀ v_{66} *v103* *v104*. *P* (Name v_{66} says *v103* eqn *v104*)) $\wedge$
(∀ v_{66} *v105* *v106*. *P* (Name v_{66} says *v105* lte *v106*)) $\wedge$
(∀ v_{66} *v107* *v108*. *P* (Name v_{66} says *v107* lt *v108*)) $\wedge$
(∀ v_{67} v_{68} v_{11}. *P* (v_{67} meet v_{68} says v_{11})) $\wedge$
(∀ v_{69} v_{70} v_{11}. *P* (v_{69} quoting v_{70} says v_{11})) $\wedge$

```
(∀ v12 v13. P (v12 speaks_for v13)) ∧
(∀ v14 v15. P (v14 controls v15)) ∧
(∀ v16 v17 v18. P (reps v16 v17 v18)) ∧
(∀ v19 v20. P (v19 domi v20)) ∧
(∀ v21 v22. P (v21 eqi v22)) ∧
(∀ v23 v24. P (v23 doms v24)) ∧
(∀ v25 v26. P (v25 eqs v26)) ∧ (∀ v27 v28. P (v27 eqn v28)) ∧
(∀ v29 v30. P (v29 lte v30)) ∧ (∀ v31 v32. P (v31 lt v32)) ⇒
∀ v. P v
```

5 projectAssuranceExec Theory

Built: 27 December 2018

Parent Theories: projectSecurity

5.1 Theorems

[RT_ALERT_exec_complete_lemma1]

```
⊢ ∀ M Oi Os.
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM complete))]::ins)
         RT_ALERT outs) ⇒
    (M,Oi,Os) satList
    propCommandList
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM complete))]
```

[RT_ALERT_exec_complete_lemma2]

```
⊢ ∀ NS Out M Oi Os.
    TR (M,Oi,Os)
      (exec
         (inputList
            [Name SquadLeader says
             prop (SOME (SquadLeaderCOM complete))]))
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM complete))]::ins)
         RT_ALERT outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_ALERT
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM complete))])))
         (Out RT_ALERT
```

```
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM complete))])::
              outs)) ⟺
     authenticationTest authentication
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM complete))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM complete))]::ins)
          RT_ALERT outs) ∧
     (M,Oi,Os) satList
     propCommandList
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM complete))]
```

[RT_ALERT_exec_complete_thm]

```
 ⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os) (exec [SOME (SquadLeaderCOM complete)])
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM complete))]::ins)
          RT_ALERT outs)
       (CFG authentication stateAuth globalAuth ins
          (NS RT_ALERT (exec [SOME (SquadLeaderCOM complete)]))
          (Out RT_ALERT
             (exec [SOME (SquadLeaderCOM complete)])::outs)) ⟺
     authenticationTest authentication
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM complete))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM complete))]::ins)
          RT_ALERT outs) ∧
     (M,Oi,Os) satList [prop (SOME (SquadLeaderCOM complete))]
```

[RT_FORM_exec_rtPosition_lemma1]

```
 ⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM rtPosition))]::ins)
          RT_FORM outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM rtPosition))]
```

[RT_FORM_exec_rtPosition_lemma2]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os)
      (exec
         (inputList
            [Name SquadLeader says
             prop (SOME (SquadLeaderCOM rtPosition))]))
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtPosition))]::ins)
         RT_FORM outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_FORM
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtPosition))])))
         (Out RT_FORM
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtPosition))])::
              outs)) ⟺
    authenticationTest authentication
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtPosition))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtPosition))]::ins)
         RT_FORM outs) ∧
    (M,Oi,Os) satList
    propCommandList
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtPosition))]
```

[RT_FORM_exec_rtPosition_thm]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os) (exec [SOME (SquadLeaderCOM rtPosition)])
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtPosition))]::ins)
         RT_FORM outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_FORM (exec [SOME (SquadLeaderCOM rtPosition)]))
         (Out RT_FORM
            (exec [SOME (SquadLeaderCOM rtPosition)])::
              outs)) ⟺
    authenticationTest authentication
```

```
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtPosition))] ∧
    CFGInterpret (M, Oi, Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtPosition))]::ins)
         RT_FORM outs) ∧
    (M, Oi, Os) satList [prop (SOME (SquadLeaderCOM rtPosition))]
```

[RT_ORIENT_exec_rtAlert_lemma1]

```
⊢ ∀ M Oi Os.
    CFGInterpret (M, Oi, Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtAlert))]::ins)
         RT_ORIENT outs) ⇒
    (M, Oi, Os) satList
    propCommandList
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtAlert))]
```

[RT_ORIENT_exec_rtAlert_lemma2]

```
⊢ ∀ NS Out M Oi Os.
    TR (M, Oi, Os)
      (exec
         (inputList
            [Name SquadLeader says
             prop (SOME (SquadLeaderCOM rtAlert))]))
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtAlert))]::ins)
         RT_ORIENT outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_ORIENT
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtAlert))])))
         (Out RT_ORIENT
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtAlert))]))::
              outs)) ⟺
    authenticationTest authentication
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtAlert))] ∧
    CFGInterpret (M, Oi, Os)
      (CFG authentication stateAuth globalAuth
```

```
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM rtAlert))]::ins)
          RT_ORIENT outs) ∧
     (M,Oi,Os) satList
     propCommandList
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM rtAlert))]
```

[RT_ORIENT_exec_rtAlert_thm]

```
⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os) (exec [SOME (SquadLeaderCOM rtAlert)])
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM rtAlert))]::ins)
          RT_ORIENT outs)
       (CFG authentication stateAuth globalAuth ins
          (NS RT_ORIENT (exec [SOME (SquadLeaderCOM rtAlert)]))
          (Out RT_ORIENT
             (exec [SOME (SquadLeaderCOM rtAlert)])::outs)) ⟺
     authenticationTest authentication
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM rtAlert))] ∧
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM rtAlert))]::ins)
          RT_ORIENT outs) ∧
     (M,Oi,Os) satList [prop (SOME (SquadLeaderCOM rtAlert))]
```

[RT_POSITION_exec_rtOrient_lemma1]

```
⊢ ∀M Oi Os.
     CFGInterpret (M,Oi,Os)
       (CFG authentication stateAuth globalAuth
          ([Name SquadLeader says
            prop (SOME (SquadLeaderCOM rtOrient))]::ins)
          RT_POSITION outs) ⇒
     (M,Oi,Os) satList
     propCommandList
       [Name SquadLeader says
        prop (SOME (SquadLeaderCOM rtOrient))]
```

[RT_POSITION_exec_rtOrient_lemma2]

```
⊢ ∀NS Out M Oi Os.
     TR (M,Oi,Os)
       (exec
          (inputList
             [Name SquadLeader says
              prop (SOME (SquadLeaderCOM rtOrient))]))
```

```
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtOrient))]::ins)
         RT_POSITION outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_POSITION
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtOrient))])))
         (Out RT_POSITION
            (exec
               (inputList
                  [Name SquadLeader says
                   prop (SOME (SquadLeaderCOM rtOrient))])))::
           outs)) ⟺
    authenticationTest authentication
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtOrient))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtOrient))]::ins)
         RT_POSITION outs) ∧
    (M,Oi,Os) satList
    propCommandList
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtOrient))]
```

[RT_POSITION_exec_rtOrient_thm]

```
⊢ ∀ NS Out M Oi Os.
    TR (M,Oi,Os) (exec [SOME (SquadLeaderCOM rtOrient)])
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtOrient))]::ins)
         RT_POSITION outs)
      (CFG authentication stateAuth globalAuth ins
         (NS RT_POSITION
            (exec [SOME (SquadLeaderCOM rtOrient)]))
         (Out RT_POSITION
            (exec [SOME (SquadLeaderCOM rtOrient)])::outs)) ⟺
    authenticationTest authentication
      [Name SquadLeader says
       prop (SOME (SquadLeaderCOM rtOrient))] ∧
    CFGInterpret (M,Oi,Os)
      (CFG authentication stateAuth globalAuth
         ([Name SquadLeader says
           prop (SOME (SquadLeaderCOM rtOrient))]::ins)
         RT_POSITION outs) ∧
```

(*M*, *Oi*, *Os*) satList [prop (SOME (SquadLeaderCOM rtOrient))]

Index