# Contents

# 1 MoveToPBType Theory

**Built:** 10 June 2018

**Parent Theories:** indexedLists, patternMatches

## 1.1 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
```

## 1.2 Theorems

[slCommand_distinct_clauses]

```
⊢ pltForm ≠ pltMove ∧ pltForm ≠ pltHalt ∧ pltForm ≠ complete ∧
  pltForm ≠ incomplete ∧ pltMove ≠ pltHalt ∧
  pltMove ≠ complete ∧ pltMove ≠ incomplete ∧
  pltHalt ≠ complete ∧ pltHalt ≠ incomplete ∧
  complete ≠ incomplete
```

[slOutput_distinct_clauses]

```
⊢ MoveToPB ≠ PLTForm ∧ MoveToPB ≠ PLTMove ∧
  MoveToPB ≠ PLTHalt ∧ MoveToPB ≠ Complete ∧
  MoveToPB ≠ unAuthorized ∧ MoveToPB ≠ unAuthenticated ∧
  PLTForm ≠ PLTMove ∧ PLTForm ≠ PLTHalt ∧ PLTForm ≠ Complete ∧
  PLTForm ≠ unAuthorized ∧ PLTForm ≠ unAuthenticated ∧
  PLTMove ≠ PLTHalt ∧ PLTMove ≠ Complete ∧
  PLTMove ≠ unAuthorized ∧ PLTMove ≠ unAuthenticated ∧
  PLTHalt ≠ Complete ∧ PLTHalt ≠ unAuthorized ∧
  PLTHalt ≠ unAuthenticated ∧ Complete ≠ unAuthorized ∧
  Complete ≠ unAuthenticated ∧ unAuthorized ≠ unAuthenticated
```

[slState_distinct_clauses]

```
⊢ MOVE_TO_PB ≠ PLT_FORM ∧ MOVE_TO_PB ≠ PLT_MOVE ∧
  MOVE_TO_PB ≠ PLT_HALT ∧ MOVE_TO_PB ≠ COMPLETE ∧
  PLT_FORM ≠ PLT_MOVE ∧ PLT_FORM ≠ PLT_HALT ∧
  PLT_FORM ≠ COMPLETE ∧ PLT_MOVE ≠ PLT_HALT ∧
  PLT_MOVE ≠ COMPLETE ∧ PLT_HALT ≠ COMPLETE
```

# 2 ssmMoveToPB Theory

**Built:** 10 June 2018

**Parent Theories:** MoveToPBType, ssm11, OMNIType

## 2.1 Definitions

[secContextMoveToPB_def]

```
⊢ ∀ cmd.
    secContextMoveToPB cmd =
    [Name PlatoonLeader controls prop (SOME (SLc cmd))]
```

[ssmMoveToPBStateInterp_def]

```
⊢ ∀ state. ssmMoveToPBStateInterp state = TT
```

## 2.2 Theorems

[authTestMoveToPB_cmd_reject_lemma]

```
⊢ ∀ cmd. ¬authTestMoveToPB (prop (SOME cmd))
```

[authTestMoveToPB_def]

```
⊢ (authTestMoveToPB (Name PlatoonLeader says prop cmd) ⇔ T) ∧
  (authTestMoveToPB TT ⇔ F) ∧ (authTestMoveToPB FF ⇔ F) ∧
  (authTestMoveToPB (prop v) ⇔ F) ∧
  (authTestMoveToPB (notf $v_{1}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{2}$ andf $v_{3}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{4}$ orf $v_{5}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{6}$ impf $v_{7}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{8}$ eqf $v_{9}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says TT) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says FF) ⇔ F) ∧
  (authTestMoveToPB (v133 meet v134 says prop $v_{66}$) ⇔ F) ∧
  (authTestMoveToPB (v135 quoting v136 says prop $v_{66}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says notf $v_{67}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says ($v_{68}$ andf $v_{69}$)) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says ($v_{70}$ orf $v_{71}$)) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says ($v_{72}$ impf $v_{73}$)) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says ($v_{74}$ eqf $v_{75}$)) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{76}$ says $v_{77}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{78}$ speaks_for $v_{79}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{80}$ controls $v_{81}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says reps $v_{82}$ $v_{83}$ $v_{84}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{85}$ domi $v_{86}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{87}$ eqi $v_{88}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{89}$ doms $v_{90}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{91}$ eqs $v_{92}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{93}$ eqn $v_{94}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{95}$ lte $v_{96}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{10}$ says $v_{97}$ lt $v_{98}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{12}$ speaks_for $v_{13}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{14}$ controls $v_{15}$) ⇔ F) ∧
  (authTestMoveToPB (reps $v_{16}$ $v_{17}$ $v_{18}$) ⇔ F) ∧
  (authTestMoveToPB ($v_{19}$ domi $v_{20}$) ⇔ F) ∧
```

(authTestMoveToPB ($v_{21}$ eqi $v_{22}$) $\iff$ F) $\wedge$<br>
(authTestMoveToPB ($v_{23}$ doms $v_{24}$) $\iff$ F) $\wedge$<br>
(authTestMoveToPB ($v_{25}$ eqs $v_{26}$) $\iff$ F) $\wedge$<br>
(authTestMoveToPB ($v_{27}$ eqn $v_{28}$) $\iff$ F) $\wedge$<br>
(authTestMoveToPB ($v_{29}$ lte $v_{30}$) $\iff$ F) $\wedge$<br>
(authTestMoveToPB ($v_{31}$ lt $v_{32}$) $\iff$ F)

[authTestMoveToPB_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 $v_{79}$)) $\wedge$
  ($\forall v_{10}\ v_{80}\ v_{81}$. $P$ ($v_{10}$ says $v_{80}$ controls $v_{81}$)) $\wedge$
  ($\forall v_{10}\ v_{82}\ v_{83}\ v_{84}$. $P$ ($v_{10}$ says reps $v_{82}$ $v_{83}$ $v_{84}$)) $\wedge$
  ($\forall v_{10}\ v_{85}\ v_{86}$. $P$ ($v_{10}$ says $v_{85}$ domi $v_{86}$)) $\wedge$
  ($\forall v_{10}\ v_{87}\ v_{88}$. $P$ ($v_{10}$ says $v_{87}$ eqi $v_{88}$)) $\wedge$
  ($\forall v_{10}\ v_{89}\ v_{90}$. $P$ ($v_{10}$ says $v_{89}$ doms $v_{90}$)) $\wedge$
  ($\forall v_{10}\ v_{91}\ v_{92}$. $P$ ($v_{10}$ says $v_{91}$ eqs $v_{92}$)) $\wedge$
  ($\forall v_{10}\ v_{93}\ v_{94}$. $P$ ($v_{10}$ says $v_{93}$ eqn $v_{94}$)) $\wedge$
  ($\forall v_{10}\ v_{95}\ v_{96}$. $P$ ($v_{10}$ says $v_{95}$ lte $v_{96}$)) $\wedge$
  ($\forall v_{10}\ v_{97}\ v_{98}$. $P$ ($v_{10}$ says $v_{97}$ lt $v_{98}$)) $\wedge$
  ($\forall v_{12}\ v_{13}$. $P$ ($v_{12}$ speaks_for $v_{13}$)) $\wedge$
  ($\forall v_{14}\ v_{15}$. $P$ ($v_{14}$ controls $v_{15}$)) $\wedge$
  ($\forall v_{16}\ v_{17}\ v_{18}$. $P$ (reps $v_{16}$ $v_{17}$ $v_{18}$)) $\wedge$
  ($\forall v_{19}\ v_{20}$. $P$ ($v_{19}$ domi $v_{20}$)) $\wedge$
  ($\forall v_{21}\ v_{22}$. $P$ ($v_{21}$ eqi $v_{22}$)) $\wedge$
  ($\forall v_{23}\ v_{24}$. $P$ ($v_{23}$ doms $v_{24}$)) $\wedge$
  ($\forall v_{25}\ v_{26}$. $P$ ($v_{25}$ eqs $v_{26}$)) $\wedge$ ($\forall v_{27}\ v_{28}$. $P$ ($v_{27}$ eqn $v_{28}$)) $\wedge$
  ($\forall v_{29}\ v_{30}$. $P$ ($v_{29}$ lte $v_{30}$)) $\wedge$ ($\forall v_{31}\ v_{32}$. $P$ ($v_{31}$ lt $v_{32}$)) $\Rightarrow$
  $\forall v$. $P$ $v$

[moveToPBNS_def]

$\vdash$ (moveToPBNS MOVE_TO_PB (exec (SLc pltForm)) = PLT_FORM) $\wedge$
 (moveToPBNS MOVE_TO_PB (exec (SLc incomplete)) =
  MOVE_TO_PB) $\wedge$
 (moveToPBNS PLT_FORM (exec (SLc pltMove)) = PLT_MOVE) $\wedge$
 (moveToPBNS PLT_FORM (exec (SLc incomplete)) = PLT_FORM) $\wedge$
 (moveToPBNS PLT_MOVE (exec (SLc pltHalt)) = PLT_HALT) $\wedge$

```
  (moveToPBNS PLT_MOVE (exec (SLc incomplete)) = PLT_MOVE) ∧
  (moveToPBNS PLT_HALT (exec (SLc complete)) = COMPLETE) ∧
  (moveToPBNS PLT_HALT (exec (SLc incomplete)) = PLT_HALT) ∧
  (moveToPBNS s (trap (SLc cmd)) = s) ∧
  (moveToPBNS s (discard (SLc cmd)) = s)
```

[moveToPBNS_ind]

```
⊢ ∀P.
    P MOVE_TO_PB (exec (SLc pltForm)) ∧
    P MOVE_TO_PB (exec (SLc incomplete)) ∧
    P PLT_FORM (exec (SLc pltMove)) ∧
    P PLT_FORM (exec (SLc incomplete)) ∧
    P PLT_MOVE (exec (SLc pltHalt)) ∧
    P PLT_MOVE (exec (SLc incomplete)) ∧
    P PLT_HALT (exec (SLc complete)) ∧
    P PLT_HALT (exec (SLc incomplete)) ∧
    (∀s cmd. P s (trap (SLc cmd))) ∧
    (∀s cmd. P s (discard (SLc cmd))) ∧
    (∀s v6. P s (discard (ESCc v6))) ∧
    (∀s v9. P s (trap (ESCc v9))) ∧
    (∀v12. P MOVE_TO_PB (exec (ESCc v12))) ∧
    P MOVE_TO_PB (exec (SLc pltMove)) ∧
    P MOVE_TO_PB (exec (SLc pltHalt)) ∧
    P MOVE_TO_PB (exec (SLc complete)) ∧
    (∀v15. P PLT_FORM (exec (ESCc v15))) ∧
    P PLT_FORM (exec (SLc pltForm)) ∧
    P PLT_FORM (exec (SLc pltHalt)) ∧
    P PLT_FORM (exec (SLc complete)) ∧
    (∀v18. P PLT_MOVE (exec (ESCc v18))) ∧
    P PLT_MOVE (exec (SLc pltForm)) ∧
    P PLT_MOVE (exec (SLc pltMove)) ∧
    P PLT_MOVE (exec (SLc complete)) ∧
    (∀v21. P PLT_HALT (exec (ESCc v21))) ∧
    P PLT_HALT (exec (SLc pltForm)) ∧
    P PLT_HALT (exec (SLc pltMove)) ∧
    P PLT_HALT (exec (SLc pltHalt)) ∧
    (∀v23. P COMPLETE (exec v23)) ⇒
    ∀v v1. P v v1
```

[moveToPBOut_def]

```
⊢ (moveToPBOut MOVE_TO_PB (exec (SLc pltForm)) = PLTForm) ∧
  (moveToPBOut MOVE_TO_PB (exec (SLc incomplete)) = MoveToPB) ∧
  (moveToPBOut PLT_FORM (exec (SLc pltMove)) = PLTMove) ∧
  (moveToPBOut PLT_FORM (exec (SLc incomplete)) = PLTForm) ∧
  (moveToPBOut PLT_MOVE (exec (SLc pltHalt)) = PLTHalt) ∧
  (moveToPBOut PLT_MOVE (exec (SLc incomplete)) = PLTMove) ∧
  (moveToPBOut PLT_HALT (exec (SLc complete)) = Complete) ∧
  (moveToPBOut PLT_HALT (exec (SLc incomplete)) = PLTHalt) ∧
  (moveToPBOut s (trap (SLc cmd)) = unAuthorized) ∧
  (moveToPBOut s (discard (SLc cmd)) = unAuthenticated)
```

[moveToPBOut_ind]

```
⊢ ∀P.
    P MOVE_TO_PB (exec (SLc pltForm)) ∧
    P MOVE_TO_PB (exec (SLc incomplete)) ∧
    P PLT_FORM (exec (SLc pltMove)) ∧
    P PLT_FORM (exec (SLc incomplete)) ∧
    P PLT_MOVE (exec (SLc pltHalt)) ∧
    P PLT_MOVE (exec (SLc incomplete)) ∧
    P PLT_HALT (exec (SLc complete)) ∧
    P PLT_HALT (exec (SLc incomplete)) ∧
    (∀s cmd. P s (trap (SLc cmd))) ∧
    (∀s cmd. P s (discard (SLc cmd))) ∧
    (∀s v6. P s (discard (ESCc v6))) ∧
    (∀s v9. P s (trap (ESCc v9))) ∧
    (∀v12. P MOVE_TO_PB (exec (ESCc v12))) ∧
    P MOVE_TO_PB (exec (SLc pltMove)) ∧
    P MOVE_TO_PB (exec (SLc pltHalt)) ∧
    P MOVE_TO_PB (exec (SLc complete)) ∧
    (∀v15. P PLT_FORM (exec (ESCc v15))) ∧
    P PLT_FORM (exec (SLc pltForm)) ∧
    P PLT_FORM (exec (SLc pltHalt)) ∧
    P PLT_FORM (exec (SLc complete)) ∧
    (∀v18. P PLT_MOVE (exec (ESCc v18))) ∧
    P PLT_MOVE (exec (SLc pltForm)) ∧
    P PLT_MOVE (exec (SLc pltMove)) ∧
    P PLT_MOVE (exec (SLc complete)) ∧
    (∀v21. P PLT_HALT (exec (ESCc v21))) ∧
    P PLT_HALT (exec (SLc pltForm)) ∧
    P PLT_HALT (exec (SLc pltMove)) ∧
    P PLT_HALT (exec (SLc pltHalt)) ∧
    (∀v23. P COMPLETE (exec v23)) ⇒
    ∀v v1. P v v1
```

[PlatoonLeader_exec_slCommand_justified_thm]

```
⊢ ∀NS Out M Oi Os.
    TR (M,Oi,Os) (exec (SLc slCommand))
      (CFG authTestMoveToPB ssmMoveToPBStateInterp
         (secContextMoveToPB slCommand)
         (Name PlatoonLeader says prop (SOME (SLc slCommand))::
              ins) s outs)
      (CFG authTestMoveToPB ssmMoveToPBStateInterp
         (secContextMoveToPB slCommand) ins
         (NS s (exec (SLc slCommand)))
         (Out s (exec (SLc slCommand))::outs)) ⟺
    authTestMoveToPB
      (Name PlatoonLeader says prop (SOME (SLc slCommand))) ∧
    CFGInterpret (M,Oi,Os)
      (CFG authTestMoveToPB ssmMoveToPBStateInterp
         (secContextMoveToPB slCommand)
```

```
          (Name PlatoonLeader says prop (SOME (SLc slCommand))::
               ins) s outs) ∧
     (M,Oi,Os) sat prop (SOME (SLc slCommand))
```

[PlatoonLeader_slCommand_lemma]

```
 ⊢ CFGInterpret (M,Oi,Os)
     (CFG authTestMoveToPB ssmMoveToPBStateInterp
        (secContextMoveToPB slCommand)
        (Name PlatoonLeader says prop (SOME (SLc slCommand))::
             ins) s outs) ⇒
   (M,Oi,Os) sat prop (SOME (SLc slCommand))
```

# Index