Contents

1 conops0Solution Theory

Built: 29 October 2017
Parent Theories: aclDrules

1.1 Datatypes

commands = go | nogo | launch | abort | activate | stand_down

keyPrinc = Staff people | Role roles | Ap num

people = Alice | Bob

principals = PR keyPrinc | Key keyPrinc

roles = Commander | Operator | CA

1.2 Theorems

[ApRuleActivate_thm]

```
⊢ (M, Oi, Os) sat
  Name (PR (Role Operator)) controls prop launch ⇒
  (M, Oi, Os) sat
  reps (Name (PR (Staff Bob))) (Name (PR (Role Operator)))
    (prop launch) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Bob)) quoting Name (PR (Role Operator)) says
  prop launch ⇒
  (M, Oi, Os) sat prop launch impf prop activate ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) speaks_for Name (PR (Role CA)) ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) says
  Name (Key (Staff Bob)) speaks_for Name (PR (Staff Bob)) ⇒
  (M, Oi, Os) sat
  Name (PR (Role CA)) controls
  Name (Key (Staff Bob)) speaks_for Name (PR (Staff Bob)) ⇒
  (M, Oi, Os) sat prop activate
```

[ApRuleStandDown_thm]

```
⊢ (M, Oi, Os) sat Name (PR (Role Operator)) controls prop abort ⇒
  (M, Oi, Os) sat
  reps (Name (PR (Staff Bob))) (Name (PR (Role Operator)))
    (prop abort) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Bob)) quoting Name (PR (Role Operator)) says
  prop abort ⇒
  (M, Oi, Os) sat prop abort impf prop stand_down ⇒
```

```
(M, Oi, Os) sat
Name (Key (Role CA)) speaks_for Name (PR (Role CA)) ⇒
(M, Oi, Os) sat
Name (Key (Role CA)) says
Name (Key (Staff Bob)) speaks_for Name (PR (Staff Bob)) ⇒
(M, Oi, Os) sat
Name (PR (Role CA)) controls
Name (Key (Staff Bob)) speaks_for Name (PR (Staff Bob)) ⇒
(M, Oi, Os) sat prop stand_down
```

[OpRuleAbort_thm]

```
⊢ (M, Oi, Os) sat Name (PR (Role Commander)) controls prop nogo ⇒
  (M, Oi, Os) sat
  reps (Name (PR (Staff Alice))) (Name (PR (Role Commander)))
    (prop nogo) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Alice)) quoting
  Name (PR (Role Commander)) says prop nogo ⇒
  (M, Oi, Os) sat prop nogo impf prop abort ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) speaks_for Name (PR (Role CA)) ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) says
  Name (Key (Staff Alice)) speaks_for Name (PR (Staff Alice)) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Bob)) quoting Name (PR (Role Operator)) says
  prop abort
```

[OpRuleLaunch_thm]

```
⊢ (M, Oi, Os) sat Name (PR (Role Commander)) controls prop go ⇒
  (M, Oi, Os) sat
  reps (Name (PR (Staff Alice))) (Name (PR (Role Commander)))
    (prop go) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Alice)) quoting
  Name (PR (Role Commander)) says prop go ⇒
  (M, Oi, Os) sat prop go impf prop launch ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) speaks_for Name (PR (Role CA)) ⇒
  (M, Oi, Os) sat
  Name (Key (Role CA)) says
  Name (Key (Staff Alice)) speaks_for Name (PR (Staff Alice)) ⇒
  (M, Oi, Os) sat
  Name (PR (Role CA)) controls
  Name (Key (Staff Alice)) speaks_for Name (PR (Staff Alice)) ⇒
  (M, Oi, Os) sat
  Name (Key (Staff Bob)) quoting Name (PR (Role Operator)) says
  prop launch
```

2 example1 Theory

Built: 29 October 2017
Parent Theories: aclDrules

2.1 Datatypes

commands = go | nogo | launch | abort

staff = Alice | Bob | Carol | Dan

2.2 Theorems

[aclExercise1]

⊢ (*M*, *Oi*, *Os*) sat Name Alice says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Bob says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Alice meet Name Bob says prop go

[aclExercise1A]

⊢ (*M*, *Oi*, *Os*) sat Name Alice says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Bob says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Alice meet Name Bob says prop go

[aclExercise2A]

⊢ (*M*, *Oi*, *Os*) sat Name Alice says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Alice controls prop go ⇒
(*M*, *Oi*, *Os*) sat prop go impf prop launch ⇒
(*M*, *Oi*, *Os*) sat Name Bob says prop launch

[aclExercise2B]

⊢ (*M*, *Oi*, *Os*) sat Name Alice says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Bob says prop go ⇒
(*M*, *Oi*, *Os*) sat Name Alice meet Name Bob says prop go

[aclExerciseTwo]

⊢ (*M*, *Oi*, *Os*) sat Name Bob says prop launch

Index