Contents

1 example1 Theory

Built: 19 March 2019

Parent Theories: aclDrules

1.1 Datatypes

```
commands = go | nogo | launch | abort

staff = Alice | Bob | Carol | Dan
```

1.2 Theorems

[example1Theorem]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example1TheoremA]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example1TheoremB]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example2Theorem]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice speaks_for Name Bob ⇒
  (M,Oi,Os) sat Name Bob controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example2TheoremA]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice speaks_for Name Bob ⇒
  (M,Oi,Os) sat Name Bob controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example2TheoremB]

```
⊢ (M,Oi,Os) sat Name Alice says prop go ⇒
  (M,Oi,Os) sat Name Alice speaks_for Name Bob ⇒
  (M,Oi,Os) sat Name Bob controls prop go ⇒
  (M,Oi,Os) sat prop go
```

[example3Theorem]

```
⊢ (M,Oi,Os) sat prop go impf prop launch ⇒
  (M,Oi,Os) sat prop go ⇒
  (M,Oi,Os) sat Name Carol says prop launch
```

[example3TheoremA]

```
⊢ (M,Oi,Os) sat prop go impf prop launch ⇒
  (M,Oi,Os) sat prop go ⇒
  (M,Oi,Os) sat Name Carol says prop launch
```

[Mono_Reps_Theorem]

```
⊢ (M,Oi,Os) sat Q controls f ⇒
  (M,Oi,Os) sat reps P Q f ⇒
  (M,Oi,Os) sat P' quoting Q' says f ⇒
  (M,Oi,Os) sat P' speaks_for P ⇒
  (M,Oi,Os) sat Q' speaks_for Q ⇒
  (M,Oi,Os) sat f
```

2 solutions1 Theory

Built: 19 March 2019
Parent Theories: example1

2.1 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
```

[aclExercise1B]

```
⊢ (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
```

[aclExercise2]

```
⊢ (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
```

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

3 conops0Solution Theory

Built: 19 March 2019
Parent Theories: aclDrules

3.1 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
```

3.2 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 (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 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
```

Index