

Contents

1	ssm11 Theory	3
1.1	Datatypes	3
1.2	Definitions	3
1.3	Theorems	4
2	ssm Theory	10
2.1	Datatypes	10
2.2	Definitions	10
2.3	Theorems	11
3	satList Theory	19
3.1	Definitions	19
3.2	Theorems	20

1 ssm11 Theory

Built: 16 May 2018

Parent Theories: satList

1.1 Datatypes

```
configuration =
  CFG (('command order, 'principal, 'd, 'e) Form -> bool)
    ('state -> ('command order, 'principal, 'd, 'e) Form)
    (('command order, 'principal, 'd, 'e) Form list)
    (('command order, 'principal, 'd, 'e) Form list) 'state
    ('output list)

order = SOME 'command | NONE

trType = discard 'command | trap 'command | exec 'command
```

1.2 Definitions

[\[TR_def\]](#)

```
⊢ TR =
  (λ a0 a1 a2 a3.
    ∀ TR'.
      (∀ a0 a1 a2 a3.
        (∃ authenticationTest P NS M Oi Os Out s
          securityContext stateInterp cmd ins outs.
            (a0 = (M, Oi, Os)) ∧ (a1 = exec cmd) ∧
            (a2 =
              CFG authenticationTest stateInterp
                securityContext (P says prop (SOME cmd)::ins) s
                outs) ∧
            (a3 =
              CFG authenticationTest stateInterp
                securityContext ins (NS s (exec cmd))
                (Out s (exec cmd)::outs)) ∧
            authenticationTest (P says prop (SOME cmd)) ∧
            CFGInterpret (M, Oi, Os)
              (CFG authenticationTest stateInterp
                securityContext (P says prop (SOME cmd)::ins)
                s outs)) ∨
        (∃ authenticationTest P NS M Oi Os Out s
          securityContext stateInterp cmd ins outs.
            (a0 = (M, Oi, Os)) ∧ (a1 = trap cmd) ∧
            (a2 =
              CFG authenticationTest stateInterp
                securityContext (P says prop (SOME cmd)::ins) s
                outs) ∧
```

$$\begin{aligned}
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext ins (NS s (trap cmd))} \\
& \quad \quad (\text{Out s (trap cmd)::outs})) \wedge \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad \text{CFGInterpret (M, Oi, Os)} \\
& \quad \quad (\text{CFG authenticationTest stateInterp} \\
& \quad \quad \quad \text{securityContext (P says prop (SOME cmd)::ins)} \\
& \quad \quad \quad \text{s outs})) \vee \\
& (\exists \text{ authenticationTest NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd x ins outs.} \\
& (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard cmd}) \wedge \\
& (a_2 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext (x::ins) s outs}) \wedge \\
& (a_3 = \\
& \quad \text{CFG authenticationTest stateInterp} \\
& \quad \quad \text{securityContext ins (NS s (discard cmd))} \\
& \quad \quad (\text{Out s (discard cmd)::outs})) \wedge \\
& \neg \text{authenticationTest x}) \Rightarrow \\
& TR' a_0 a_1 a_2 a_3) \Rightarrow \\
& TR' a_0 a_1 a_2 a_3)
\end{aligned}$$

1.3 Theorems

[CFGInterpret_def]

$$\begin{aligned}
& \vdash \text{CFGInterpret (M, Oi, Os)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{input::ins) state outputStream}) \iff \\
& \quad (M, Oi, Os) \text{ satList securityContext} \wedge (M, Oi, Os) \text{ sat input} \wedge \\
& \quad (M, Oi, Os) \text{ sat stateInterp state}
\end{aligned}$$

[CFGInterpret_ind]

$$\begin{aligned}
& \vdash \forall P. \\
& \quad (\forall M Oi Os \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \text{input ins state outputStream.} \\
& \quad P (M, Oi, Os) \\
& \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (\text{input::ins) state outputStream})) \wedge \\
& \quad (\forall v_{15} v_{10} v_{11} v_{12} v_{13} v_{14}. \\
& \quad \quad P v_{15} (\text{CFG } v_{10} v_{11} v_{12} [] v_{13} v_{14})) \Rightarrow \\
& \quad \forall v v_1 v_2 v_3. P (v, v_1, v_2) v_3
\end{aligned}$$

[configuration_one_one]

$$\begin{aligned}
& \vdash \forall a_0 a_1 a_2 a_3 a_4 a_5 a'_0 a'_1 a'_2 a'_3 a'_4 a'_5. \\
& \quad (\text{CFG } a_0 a_1 a_2 a_3 a_4 a_5 = \text{CFG } a'_0 a'_1 a'_2 a'_3 a'_4 a'_5) \iff \\
& \quad (a_0 = a'_0) \wedge (a_1 = a'_1) \wedge (a_2 = a'_2) \wedge (a_3 = a'_3) \wedge \\
& \quad (a_4 = a'_4) \wedge (a_5 = a'_5)
\end{aligned}$$

[order_distinct_clauses]

$\vdash \forall a. \text{SOME } a \neq \text{NONE}$

[order_one_one]

$\vdash \forall a \ a'. (\text{SOME } a = \text{SOME } a') \iff (a = a')$

[TR_cases]

$\vdash \forall a_0 \ a_1 \ a_2 \ a_3.$

$\text{TR } a_0 \ a_1 \ a_2 \ a_3 \iff$
 $(\exists \text{authenticationTest } P \ NS \ M \ Oi \ Os \ Out \ s \ \text{securityContext}$
 $\text{stateInterp } cmd \ ins \ outs.$
 $(a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } cmd) \wedge$
 $(a_2 =$
 $\text{CFG authenticationTest stateInterp securityContext}$
 $(P \text{ says prop (SOME } cmd)::ins) \ s \ outs) \wedge$
 $(a_3 =$
 $\text{CFG authenticationTest stateInterp securityContext ins}$
 $(NS \ s (\text{exec } cmd)) (Out \ s (\text{exec } cmd)::outs)) \wedge$
 $\text{authenticationTest } (P \text{ says prop (SOME } cmd)) \wedge$
 $\text{CFGInterpret } (M, Oi, Os)$
 $(\text{CFG authenticationTest stateInterp securityContext}$
 $(P \text{ says prop (SOME } cmd)::ins) \ s \ outs)) \vee$
 $(\exists \text{authenticationTest } P \ NS \ M \ Oi \ Os \ Out \ s \ \text{securityContext}$
 $\text{stateInterp } cmd \ ins \ outs.$
 $(a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } cmd) \wedge$
 $(a_2 =$
 $\text{CFG authenticationTest stateInterp securityContext}$
 $(P \text{ says prop (SOME } cmd)::ins) \ s \ outs) \wedge$
 $(a_3 =$
 $\text{CFG authenticationTest stateInterp securityContext ins}$
 $(NS \ s (\text{trap } cmd)) (Out \ s (\text{trap } cmd)::outs)) \wedge$
 $\text{authenticationTest } (P \text{ says prop (SOME } cmd)) \wedge$
 $\text{CFGInterpret } (M, Oi, Os)$
 $(\text{CFG authenticationTest stateInterp securityContext}$
 $(P \text{ says prop (SOME } cmd)::ins) \ s \ outs)) \vee$
 $\exists \text{authenticationTest } NS \ M \ Oi \ Os \ Out \ s \ \text{securityContext}$
 $\text{stateInterp } cmd \ x \ ins \ outs.$
 $(a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard } cmd) \wedge$
 $(a_2 =$
 $\text{CFG authenticationTest stateInterp securityContext}$
 $(x::ins) \ s \ outs) \wedge$
 $(a_3 =$
 $\text{CFG authenticationTest stateInterp securityContext ins}$
 $(NS \ s (\text{discard } cmd)) (Out \ s (\text{discard } cmd)::outs)) \wedge$
 $\neg \text{authenticationTest } x$

[TR_discard_cmd_rule]

$$\begin{aligned} &\vdash \text{TR } (M, Oi, Os) \text{ (discard cmd)} \\ &\quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad (x::ins) s outs) \\ &\quad (\text{CFG authenticationTest stateInterp securityContext ins} \\ &\quad \quad (\text{NS } s \text{ (discard cmd)}) (\text{Out } s \text{ (discard cmd)::outs})) \iff \\ &\quad \neg \text{authenticationTest } x \end{aligned}$$

[TR_EQ_rules_thm]

$$\begin{aligned} &\vdash (\text{TR } (M, Oi, Os) \text{ (exec cmd)}) \\ &\quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\ &\quad (\text{CFG authenticationTest stateInterp securityContext ins} \\ &\quad \quad (\text{NS } s \text{ (exec cmd)}) (\text{Out } s \text{ (exec cmd)::outs})) \iff \\ &\quad \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\ &\quad \text{CFGInterpret } (M, Oi, Os) \\ &\quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs})) \wedge \\ &\quad (\text{TR } (M, Oi, Os) \text{ (trap cmd)}) \\ &\quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\ &\quad \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\ &\quad \quad \quad (\text{NS } s \text{ (trap cmd)}) (\text{Out } s \text{ (trap cmd)::outs})) \iff \\ &\quad \quad \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\ &\quad \quad \text{CFGInterpret } (M, Oi, Os) \\ &\quad \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs})) \wedge \\ &\quad (\text{TR } (M, Oi, Os) \text{ (discard cmd)}) \\ &\quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad (x::ins) s outs) \\ &\quad \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\ &\quad \quad \quad (\text{NS } s \text{ (discard cmd)}) (\text{Out } s \text{ (discard cmd)::outs})) \iff \\ &\quad \quad \neg \text{authenticationTest } x \end{aligned}$$

[TR_exec_cmd_rule]

$$\begin{aligned} &\vdash \forall \text{authenticationTest securityContext stateInterp } P \text{ cmd ins } s \\ &\quad \text{outs.} \\ &\quad (\forall M \text{ } Oi \text{ } Os. \\ &\quad \quad \text{CFGInterpret } (M, Oi, Os) \\ &\quad \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \Rightarrow \\ &\quad \quad \quad (M, Oi, Os) \text{ sat prop (SOME cmd)}) \Rightarrow \\ &\quad \forall \text{NS Out } M \text{ } Oi \text{ } Os. \\ &\quad \quad \text{TR } (M, Oi, Os) \text{ (exec cmd)} \\ &\quad \quad \quad (\text{CFG authenticationTest stateInterp securityContext} \\ &\quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \\ &\quad \quad \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\ &\quad \quad \quad \quad (\text{NS } s \text{ (exec cmd)}) (\text{Out } s \text{ (exec cmd)::outs})) \iff \\ &\quad \quad \quad \text{authenticationTest } (P \text{ says prop (SOME cmd)}) \wedge \\ &\quad \quad \quad \text{CFGInterpret } (M, Oi, Os) \end{aligned}$$

(CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs) \wedge
 (M, Oi, Os) sat prop (SOME cmd)

[TR_ind]

$\vdash \forall TR'.$

(\forall authenticationTest P NS M Oi Os Out s securityContext
 stateInterp cmd ins outs.
 authenticationTest (P says prop (SOME cmd)) \wedge
 CFGInterpret (M, Oi, Os)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs) \Rightarrow
 TR' (M, Oi, Os) (exec cmd)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs)
 (CFG authenticationTest stateInterp securityContext
 ins (NS s (exec cmd)) (Out s (exec cmd)::outs))) \wedge
 (\forall authenticationTest P NS M Oi Os Out s securityContext
 stateInterp cmd ins outs.
 authenticationTest (P says prop (SOME cmd)) \wedge
 CFGInterpret (M, Oi, Os)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs) \Rightarrow
 TR' (M, Oi, Os) (trap cmd)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs)
 (CFG authenticationTest stateInterp securityContext
 ins (NS s (trap cmd)) (Out s (trap cmd)::outs))) \wedge
 (\forall authenticationTest NS M Oi Os Out s securityContext
 stateInterp cmd x ins outs.
 \neg authenticationTest x \Rightarrow
 TR' (M, Oi, Os) (discard cmd)
 (CFG authenticationTest stateInterp securityContext
 (x::ins) s outs)
 (CFG authenticationTest stateInterp securityContext
 ins (NS s (discard cmd))
 (Out s (discard cmd)::outs))) \Rightarrow
 $\forall a_0 a_1 a_2 a_3. \text{TR } a_0 a_1 a_2 a_3 \Rightarrow \text{TR}' a_0 a_1 a_2 a_3$

[TR_rules]

$\vdash (\forall$ authenticationTest P NS M Oi Os Out s securityContext
 stateInterp cmd ins outs.
 authenticationTest (P says prop (SOME cmd)) \wedge
 CFGInterpret (M, Oi, Os)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs) \Rightarrow
 TR (M, Oi, Os) (exec cmd)
 (CFG authenticationTest stateInterp securityContext
 (P says prop (SOME cmd)::ins) s outs)

$$\begin{aligned}
& (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad (\text{NS s (exec cmd)}) (\text{Out s (exec cmd)::outs}})) \wedge \\
& (\forall \text{authenticationTest P NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd ins outs.} \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad \text{CFGInterpret (M, Oi, Os)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \Rightarrow \\
& \text{TR (M, Oi, Os) (trap cmd)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad (\text{NS s (trap cmd)}) (\text{Out s (trap cmd)::outs}})) \wedge \\
& \forall \text{authenticationTest NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd x ins outs.} \\
& \neg \text{authenticationTest x} \Rightarrow \\
& \text{TR (M, Oi, Os) (discard cmd)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext ins} \\
& \quad \quad (\text{NS s (discard cmd)}) (\text{Out s (discard cmd)::outs}}))
\end{aligned}$$

[TR_strongind]

$\vdash \forall TR'.$

$$\begin{aligned}
& (\forall \text{authenticationTest P NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd ins outs.} \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad \text{CFGInterpret (M, Oi, Os)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \Rightarrow \\
& \text{TR' (M, Oi, Os) (exec cmd)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \text{ins (NS s (exec cmd)}) (\text{Out s (exec cmd)::outs}})) \wedge \\
& (\forall \text{authenticationTest P NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd ins outs.} \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad \text{CFGInterpret (M, Oi, Os)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \Rightarrow \\
& \text{TR' (M, Oi, Os) (trap cmd)} \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad (\text{P says prop (SOME cmd)::ins}) s outs) \\
& \quad (\text{CFG authenticationTest stateInterp securityContext} \\
& \quad \quad \text{ins (NS s (trap cmd)}) (\text{Out s (trap cmd)::outs}})) \wedge \\
& (\forall \text{authenticationTest NS M Oi Os Out s securityContext} \\
& \quad \text{stateInterp cmd x ins outs.} \\
& \neg \text{authenticationTest x} \Rightarrow
\end{aligned}$$

$$\begin{aligned}
& TR' (M, Oi, Os) (\text{discard } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (x::ins) s outs) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad ins (NS s (\text{discard } cmd))) \\
& \quad \quad (Out s (\text{discard } cmd)::outs))) \Rightarrow \\
& \forall a_0 a_1 a_2 a_3. TR a_0 a_1 a_2 a_3 \Rightarrow TR' a_0 a_1 a_2 a_3
\end{aligned}$$

[TR_trap_cmd_rule]

$$\begin{aligned}
& \vdash \forall \text{ authenticationTest stateInterp securityContext } P \text{ cmd ins s} \\
& \quad outs. \\
& \quad (\forall M Oi Os. \\
& \quad \quad CFGInterpret (M, Oi, Os) \\
& \quad \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs}) \Rightarrow \\
& \quad \quad \quad (M, Oi, Os) \text{ sat prop NONE}) \Rightarrow \\
& \quad \forall NS Out M Oi Os. \\
& \quad \quad TR (M, Oi, Os) (\text{trap } cmd) \\
& \quad \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs) \\
& \quad \quad \quad (CFG \text{ authenticationTest stateInterp securityContext ins} \\
& \quad \quad \quad \quad (NS s (\text{trap } cmd)) (Out s (\text{trap } cmd)::outs))) \iff \\
& \quad \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad \quad CFGInterpret (M, Oi, Os) \\
& \quad \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs}) \wedge \\
& \quad \quad (M, Oi, Os) \text{ sat prop NONE}
\end{aligned}$$

[TRrule0]

$$\begin{aligned}
& \vdash TR (M, Oi, Os) (\text{exec } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext ins} \\
& \quad \quad (NS s (\text{exec } cmd)) (Out s (\text{exec } cmd)::outs))) \iff \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad CFGInterpret (M, Oi, Os) \\
& \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs)
\end{aligned}$$

[TRrule1]

$$\begin{aligned}
& \vdash TR (M, Oi, Os) (\text{trap } cmd) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs) \\
& \quad (CFG \text{ authenticationTest stateInterp securityContext ins} \\
& \quad \quad (NS s (\text{trap } cmd)) (Out s (\text{trap } cmd)::outs))) \iff \\
& \quad \text{authenticationTest (P says prop (SOME cmd))} \wedge \\
& \quad CFGInterpret (M, Oi, Os) \\
& \quad \quad (CFG \text{ authenticationTest stateInterp securityContext} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) s outs)
\end{aligned}$$

[trType_distinct_clauses]

$$\vdash (\forall a' a. \text{discard } a \neq \text{trap } a') \wedge (\forall a' a. \text{discard } a \neq \text{exec } a') \wedge \\ \forall a' a. \text{trap } a \neq \text{exec } a'$$

[trType_one_one]

$$\vdash (\forall a a'. (\text{discard } a = \text{discard } a') \iff (a = a')) \wedge \\ (\forall a a'. (\text{trap } a = \text{trap } a') \iff (a = a')) \wedge \\ \forall a a'. (\text{exec } a = \text{exec } a') \iff (a = a')$$

2 ssm Theory

Built: 16 May 2018

Parent Theories: satList

2.1 Datatypes

```
configuration =
  CFG (('command option, 'principal, 'd, 'e) Form -> bool)
    ('state ->
      ('command option, 'principal, 'd, 'e) Form list ->
        ('command option, 'principal, 'd, 'e) Form list)
    (('command option, 'principal, 'd, 'e) Form list ->
      ('command option, 'principal, 'd, 'e) Form list)
    (('command option, 'principal, 'd, 'e) Form list list)
    'state ('output list)
```

```
trType = discard 'cmdlist | trap 'cmdlist | exec 'cmdlist
```

2.2 Definitions

[authenticationTest_def]

$$\vdash \forall \text{elementTest } x. \\ \text{authenticationTest } \text{elementTest } x \iff \\ \text{FOLDR } (\lambda p q. p \wedge q) \text{ T } (\text{MAP } \text{elementTest } x)$$

[commandList_def]

$$\vdash \forall x. \text{commandList } x = \text{MAP } \text{extractCommand } x$$

[inputList_def]

$$\vdash \forall xs. \text{inputList } xs = \text{MAP } \text{extractInput } xs$$

[propCommandList_def]

$$\vdash \forall x. \text{propCommandList } x = \text{MAP } \text{extractPropCommand } x$$

[TR_def]

$$\begin{aligned}
&\vdash \text{TR} = \\
&\quad (\lambda a_0 a_1 a_2 a_3. \\
&\quad \quad \forall TR'. \\
&\quad \quad (\forall a_0 a_1 a_2 a_3. \\
&\quad \quad \quad (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } \text{stateInterp } x \\
&\quad \quad \quad \quad \text{ins } \text{outs}. \\
&\quad \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } (\text{inputList } x)) \wedge \\
&\quad \quad \quad (a_2 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\
&\quad \quad \quad \quad \text{outs}) \wedge \\
&\quad \quad \quad (a_3 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } \text{ins} \\
&\quad \quad \quad \quad \quad (NS \ s \ (\text{exec } (\text{inputList } x))) \\
&\quad \quad \quad \quad \quad (Out \ s \ (\text{exec } (\text{inputList } x))::\text{outs})) \wedge \\
&\quad \quad \quad \text{authenticationTest } \text{elementTest } x \wedge \\
&\quad \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
&\quad \quad \quad \quad (\text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\
&\quad \quad \quad \quad \text{outs})) \vee \\
&\quad \quad \quad (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } \text{stateInterp } x \\
&\quad \quad \quad \quad \text{ins } \text{outs}. \\
&\quad \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } (\text{inputList } x)) \wedge \\
&\quad \quad \quad (a_2 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\
&\quad \quad \quad \quad \text{outs}) \wedge \\
&\quad \quad \quad (a_3 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } \text{ins} \\
&\quad \quad \quad \quad \quad (NS \ s \ (\text{trap } (\text{inputList } x))) \\
&\quad \quad \quad \quad \quad (Out \ s \ (\text{trap } (\text{inputList } x))::\text{outs})) \wedge \\
&\quad \quad \quad \text{authenticationTest } \text{elementTest } x \wedge \\
&\quad \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
&\quad \quad \quad \quad (\text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\
&\quad \quad \quad \quad \text{outs})) \vee \\
&\quad \quad \quad (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ \text{context } \text{stateInterp } x \\
&\quad \quad \quad \quad \text{ins } \text{outs}. \\
&\quad \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard } (\text{inputList } x)) \wedge \\
&\quad \quad \quad (a_2 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } (x::\text{ins}) \ s \\
&\quad \quad \quad \quad \text{outs}) \wedge \\
&\quad \quad \quad (a_3 = \\
&\quad \quad \quad \quad \text{CFG } \text{elementTest } \text{stateInterp } \text{context } \text{ins} \\
&\quad \quad \quad \quad \quad (NS \ s \ (\text{discard } (\text{inputList } x))) \\
&\quad \quad \quad \quad \quad (Out \ s \ (\text{discard } (\text{inputList } x))::\text{outs})) \wedge \\
&\quad \quad \quad \neg \text{authenticationTest } \text{elementTest } x) \Rightarrow \\
&\quad \quad TR' \ a_0 \ a_1 \ a_2 \ a_3) \Rightarrow \\
&\quad TR' \ a_0 \ a_1 \ a_2 \ a_3)
\end{aligned}$$

2.3 Theorems

[CFGInterpret_def]

$$\begin{aligned} &\vdash \text{CFGInterpret } (M, Oi, Os) \\ &\quad (\text{CFG elementTest stateInterp context } (x::ins) \text{ state} \\ &\quad \quad \text{outStream}) \iff \\ &\quad (M, Oi, Os) \text{ satList context } x \wedge (M, Oi, Os) \text{ satList } x \wedge \\ &\quad (M, Oi, Os) \text{ satList stateInterp state } x \end{aligned}$$
[CFGInterpret_ind]

$$\begin{aligned} &\vdash \forall P. \\ &\quad (\forall M \ Oi \ Os \ \text{elementTest stateInterp context } x \ ins \ \text{state} \\ &\quad \quad \text{outStream}. \\ &\quad \quad P \ (M, Oi, Os) \\ &\quad \quad (\text{CFG elementTest stateInterp context } (x::ins) \ \text{state} \\ &\quad \quad \quad \text{outStream})) \wedge \\ &\quad (\forall v_{15} \ v_{10} \ v_{11} \ v_{12} \ v_{13} \ v_{14}. \\ &\quad \quad P \ v_{15} \ (\text{CFG } v_{10} \ v_{11} \ v_{12} \ [] \ v_{13} \ v_{14})) \Rightarrow \\ &\quad \forall v \ v_1 \ v_2 \ v_3. \ P \ (v, v_1, v_2) \ v_3 \end{aligned}$$
[configuration_one_one]

$$\begin{aligned} &\vdash \forall a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5. \\ &\quad (\text{CFG } a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 = \text{CFG } a'_0 \ a'_1 \ a'_2 \ a'_3 \ a'_4 \ a'_5) \iff \\ &\quad (a_0 = a'_0) \wedge (a_1 = a'_1) \wedge (a_2 = a'_2) \wedge (a_3 = a'_3) \wedge \\ &\quad (a_4 = a'_4) \wedge (a_5 = a'_5) \end{aligned}$$
[extractCommand_def]

$$\vdash \text{extractCommand } (P \text{ says prop } (\text{SOME } cmd)) = cmd$$
[extractCommand_ind]

$$\begin{aligned} &\vdash \forall P'. \\ &\quad (\forall P \ cmd. \ P' \ (P \text{ says prop } (\text{SOME } cmd))) \wedge P' \ \text{TT} \wedge P' \ \text{FF} \wedge \\ &\quad (\forall v_1. \ P' \ (\text{prop } v_1)) \wedge (\forall v_3. \ P' \ (\text{notf } v_3)) \wedge \\ &\quad (\forall v_6 \ v_7. \ P' \ (v_6 \ \text{andf } v_7)) \wedge (\forall v_{10} \ v_{11}. \ P' \ (v_{10} \ \text{orf } v_{11})) \wedge \\ &\quad (\forall v_{14} \ v_{15}. \ P' \ (v_{14} \ \text{impf } v_{15})) \wedge \\ &\quad (\forall v_{18} \ v_{19}. \ P' \ (v_{18} \ \text{eqf } v_{19})) \wedge (\forall v_{129}. \ P' \ (v_{129} \ \text{says TT})) \wedge \\ &\quad (\forall v_{130}. \ P' \ (v_{130} \ \text{says FF})) \wedge \\ &\quad (\forall v_{132}. \ P' \ (v_{132} \ \text{says prop NONE})) \wedge \\ &\quad (\forall v_{133} \ v_{66}. \ P' \ (v_{133} \ \text{says notf } v_{66})) \wedge \\ &\quad (\forall v_{134} \ v_{69} \ v_{70}. \ P' \ (v_{134} \ \text{says } (v_{69} \ \text{andf } v_{70}))) \wedge \\ &\quad (\forall v_{135} \ v_{73} \ v_{74}. \ P' \ (v_{135} \ \text{says } (v_{73} \ \text{orf } v_{74}))) \wedge \\ &\quad (\forall v_{136} \ v_{77} \ v_{78}. \ P' \ (v_{136} \ \text{says } (v_{77} \ \text{impf } v_{78}))) \wedge \\ &\quad (\forall v_{137} \ v_{81} \ v_{82}. \ P' \ (v_{137} \ \text{says } (v_{81} \ \text{eqf } v_{82}))) \wedge \\ &\quad (\forall v_{138} \ v_{85} \ v_{86}. \ P' \ (v_{138} \ \text{says } v_{85} \ \text{says } v_{86})) \wedge \\ &\quad (\forall v_{139} \ v_{89} \ v_{90}. \ P' \ (v_{139} \ \text{says } v_{89} \ \text{speaks_for } v_{90})) \wedge \\ &\quad (\forall v_{140} \ v_{93} \ v_{94}. \ P' \ (v_{140} \ \text{says } v_{93} \ \text{controls } v_{94})) \wedge \\ &\quad (\forall v_{141} \ v_{98} \ v_{99} \ v_{100}. \ P' \ (v_{141} \ \text{says reps } v_{98} \ v_{99} \ v_{100})) \wedge \\ &\quad (\forall v_{142} \ v_{103} \ v_{104}. \ P' \ (v_{142} \ \text{says } v_{103} \ \text{domi } v_{104})) \wedge \\ &\quad (\forall v_{143} \ v_{107} \ v_{108}. \ P' \ (v_{143} \ \text{says } v_{107} \ \text{eqi } v_{108})) \wedge \\ &\quad (\forall v_{144} \ v_{111} \ v_{112}. \ P' \ (v_{144} \ \text{says } v_{111} \ \text{doms } v_{112})) \wedge \end{aligned}$$

$$\begin{aligned}
& (\forall v145 \ v115 \ v116. \ P' \ (v145 \ \text{says} \ v115 \ \text{eqs} \ v116)) \wedge \\
& (\forall v146 \ v119 \ v120. \ P' \ (v146 \ \text{says} \ v119 \ \text{eqn} \ v120)) \wedge \\
& (\forall v147 \ v123 \ v124. \ P' \ (v147 \ \text{says} \ v123 \ \text{lte} \ v124)) \wedge \\
& (\forall v148 \ v127 \ v128. \ P' \ (v148 \ \text{says} \ v127 \ \text{lt} \ v128)) \wedge \\
& (\forall v24 \ v25. \ P' \ (v24 \ \text{speaks_for} \ v25)) \wedge \\
& (\forall v28 \ v29. \ P' \ (v28 \ \text{controls} \ v29)) \wedge \\
& (\forall v33 \ v34 \ v35. \ P' \ (\text{reps} \ v33 \ v34 \ v35)) \wedge \\
& (\forall v38 \ v39. \ P' \ (v38 \ \text{domi} \ v39)) \wedge \\
& (\forall v42 \ v43. \ P' \ (v42 \ \text{eqi} \ v43)) \wedge \\
& (\forall v46 \ v47. \ P' \ (v46 \ \text{doms} \ v47)) \wedge \\
& (\forall v50 \ v51. \ P' \ (v50 \ \text{eqs} \ v51)) \wedge \\
& (\forall v54 \ v55. \ P' \ (v54 \ \text{eqn} \ v55)) \wedge \\
& (\forall v58 \ v59. \ P' \ (v58 \ \text{lte} \ v59)) \wedge \\
& (\forall v62 \ v63. \ P' \ (v62 \ \text{lt} \ v63)) \Rightarrow \\
& \forall v. \ P' \ v
\end{aligned}$$

[extractInput_def]

$$\vdash \text{extractInput} \ (P \ \text{says} \ \text{prop} \ x) = x$$

[extractInput_ind]

$$\begin{aligned}
& \vdash \forall P'. \\
& \quad (\forall P \ x. \ P' \ (P \ \text{says} \ \text{prop} \ x)) \wedge P' \ \text{TT} \wedge P' \ \text{FF} \wedge \\
& \quad (\forall v_1. \ P' \ (\text{prop} \ v_1)) \wedge (\forall v_3. \ P' \ (\text{notf} \ v_3)) \wedge \\
& \quad (\forall v_6 \ v_7. \ P' \ (v_6 \ \text{andf} \ v_7)) \wedge (\forall v_{10} \ v_{11}. \ P' \ (v_{10} \ \text{orf} \ v_{11})) \wedge \\
& \quad (\forall v_{14} \ v_{15}. \ P' \ (v_{14} \ \text{impf} \ v_{15})) \wedge \\
& \quad (\forall v_{18} \ v_{19}. \ P' \ (v_{18} \ \text{eqf} \ v_{19})) \wedge (\forall v_{129}. \ P' \ (v_{129} \ \text{says} \ \text{TT})) \wedge \\
& \quad (\forall v_{130}. \ P' \ (v_{130} \ \text{says} \ \text{FF})) \wedge \\
& \quad (\forall v_{131} \ v_{66}. \ P' \ (v_{131} \ \text{says} \ \text{notf} \ v_{66})) \wedge \\
& \quad (\forall v_{132} \ v_{69} \ v_{70}. \ P' \ (v_{132} \ \text{says} \ (v_{69} \ \text{andf} \ v_{70}))) \wedge \\
& \quad (\forall v_{133} \ v_{73} \ v_{74}. \ P' \ (v_{133} \ \text{says} \ (v_{73} \ \text{orf} \ v_{74}))) \wedge \\
& \quad (\forall v_{134} \ v_{77} \ v_{78}. \ P' \ (v_{134} \ \text{says} \ (v_{77} \ \text{impf} \ v_{78}))) \wedge \\
& \quad (\forall v_{135} \ v_{81} \ v_{82}. \ P' \ (v_{135} \ \text{says} \ (v_{81} \ \text{eqf} \ v_{82}))) \wedge \\
& \quad (\forall v_{136} \ v_{85} \ v_{86}. \ P' \ (v_{136} \ \text{says} \ v_{85} \ \text{says} \ v_{86})) \wedge \\
& \quad (\forall v_{137} \ v_{89} \ v_{90}. \ P' \ (v_{137} \ \text{says} \ v_{89} \ \text{speaks_for} \ v_{90})) \wedge \\
& \quad (\forall v_{138} \ v_{93} \ v_{94}. \ P' \ (v_{138} \ \text{says} \ v_{93} \ \text{controls} \ v_{94})) \wedge \\
& \quad (\forall v_{139} \ v_{98} \ v_{99} \ v_{100}. \ P' \ (v_{139} \ \text{says} \ \text{reps} \ v_{98} \ v_{99} \ v_{100})) \wedge \\
& \quad (\forall v_{140} \ v_{103} \ v_{104}. \ P' \ (v_{140} \ \text{says} \ v_{103} \ \text{domi} \ v_{104})) \wedge \\
& \quad (\forall v_{141} \ v_{107} \ v_{108}. \ P' \ (v_{141} \ \text{says} \ v_{107} \ \text{eqi} \ v_{108})) \wedge \\
& \quad (\forall v_{142} \ v_{111} \ v_{112}. \ P' \ (v_{142} \ \text{says} \ v_{111} \ \text{doms} \ v_{112})) \wedge \\
& \quad (\forall v_{143} \ v_{115} \ v_{116}. \ P' \ (v_{143} \ \text{says} \ v_{115} \ \text{eqs} \ v_{116})) \wedge \\
& \quad (\forall v_{144} \ v_{119} \ v_{120}. \ P' \ (v_{144} \ \text{says} \ v_{119} \ \text{eqn} \ v_{120})) \wedge \\
& \quad (\forall v_{145} \ v_{123} \ v_{124}. \ P' \ (v_{145} \ \text{says} \ v_{123} \ \text{lte} \ v_{124})) \wedge \\
& \quad (\forall v_{146} \ v_{127} \ v_{128}. \ P' \ (v_{146} \ \text{says} \ v_{127} \ \text{lt} \ v_{128})) \wedge \\
& \quad (\forall v_{24} \ v_{25}. \ P' \ (v_{24} \ \text{speaks_for} \ v_{25})) \wedge \\
& \quad (\forall v_{28} \ v_{29}. \ P' \ (v_{28} \ \text{controls} \ v_{29})) \wedge \\
& \quad (\forall v_{33} \ v_{34} \ v_{35}. \ P' \ (\text{reps} \ v_{33} \ v_{34} \ v_{35})) \wedge \\
& \quad (\forall v_{38} \ v_{39}. \ P' \ (v_{38} \ \text{domi} \ v_{39})) \wedge \\
& \quad (\forall v_{42} \ v_{43}. \ P' \ (v_{42} \ \text{eqi} \ v_{43})) \wedge \\
& \quad (\forall v_{46} \ v_{47}. \ P' \ (v_{46} \ \text{doms} \ v_{47})) \wedge
\end{aligned}$$

$$\begin{aligned}
& (\forall v_{50} v_{51}. P' (v_{50} \text{ eqs } v_{51})) \wedge \\
& (\forall v_{54} v_{55}. P' (v_{54} \text{ eqn } v_{55})) \wedge \\
& (\forall v_{58} v_{59}. P' (v_{58} \text{ lte } v_{59})) \wedge \\
& (\forall v_{62} v_{63}. P' (v_{62} \text{ lt } v_{63})) \Rightarrow \\
& \forall v. P' v
\end{aligned}$$

[extractPropCommand_def]

$$\vdash \text{extractPropCommand } (P \text{ says prop (SOME cmd)}) = \text{prop (SOME cmd)}$$

[extractPropCommand_ind]

$$\begin{aligned}
& \vdash \forall P'. \\
& \quad (\forall P \text{ cmd}. P' (P \text{ says prop (SOME cmd)})) \wedge P' \text{ TT} \wedge P' \text{ FF} \wedge \\
& \quad (\forall v_1. P' (\text{prop } v_1)) \wedge (\forall v_3. P' (\text{notf } v_3)) \wedge \\
& \quad (\forall v_6 v_7. P' (v_6 \text{ andf } v_7)) \wedge (\forall v_{10} v_{11}. P' (v_{10} \text{ orf } v_{11})) \wedge \\
& \quad (\forall v_{14} v_{15}. P' (v_{14} \text{ impf } v_{15})) \wedge \\
& \quad (\forall v_{18} v_{19}. P' (v_{18} \text{ eqf } v_{19})) \wedge (\forall v_{129}. P' (v_{129} \text{ says TT})) \wedge \\
& \quad (\forall v_{130}. P' (v_{130} \text{ says FF})) \wedge \\
& \quad (\forall v_{132}. P' (v_{132} \text{ says prop NONE})) \wedge \\
& \quad (\forall v_{133} v_{66}. P' (v_{133} \text{ says notf } v_{66})) \wedge \\
& \quad (\forall v_{134} v_{69} v_{70}. P' (v_{134} \text{ says (v}_{69} \text{ andf } v_{70}))) \wedge \\
& \quad (\forall v_{135} v_{73} v_{74}. P' (v_{135} \text{ says (v}_{73} \text{ orf } v_{74}))) \wedge \\
& \quad (\forall v_{136} v_{77} v_{78}. P' (v_{136} \text{ says (v}_{77} \text{ impf } v_{78}))) \wedge \\
& \quad (\forall v_{137} v_{81} v_{82}. P' (v_{137} \text{ says (v}_{81} \text{ eqf } v_{82}))) \wedge \\
& \quad (\forall v_{138} v_{85} v_{86}. P' (v_{138} \text{ says v}_{85} \text{ says v}_{86})) \wedge \\
& \quad (\forall v_{139} v_{89} v_{90}. P' (v_{139} \text{ says v}_{89} \text{ speaks_for } v_{90})) \wedge \\
& \quad (\forall v_{140} v_{93} v_{94}. P' (v_{140} \text{ says v}_{93} \text{ controls } v_{94})) \wedge \\
& \quad (\forall v_{141} v_{98} v_{99} v_{100}. P' (v_{141} \text{ says reps v}_{98} v_{99} v_{100})) \wedge \\
& \quad (\forall v_{142} v_{103} v_{104}. P' (v_{142} \text{ says v}_{103} \text{ domi } v_{104})) \wedge \\
& \quad (\forall v_{143} v_{107} v_{108}. P' (v_{143} \text{ says v}_{107} \text{ eqi } v_{108})) \wedge \\
& \quad (\forall v_{144} v_{111} v_{112}. P' (v_{144} \text{ says v}_{111} \text{ doms } v_{112})) \wedge \\
& \quad (\forall v_{145} v_{115} v_{116}. P' (v_{145} \text{ says v}_{115} \text{ eqs } v_{116})) \wedge \\
& \quad (\forall v_{146} v_{119} v_{120}. P' (v_{146} \text{ says v}_{119} \text{ eqn } v_{120})) \wedge \\
& \quad (\forall v_{147} v_{123} v_{124}. P' (v_{147} \text{ says v}_{123} \text{ lte } v_{124})) \wedge \\
& \quad (\forall v_{148} v_{127} v_{128}. P' (v_{148} \text{ says v}_{127} \text{ lt } v_{128})) \wedge \\
& \quad (\forall v_{24} v_{25}. P' (v_{24} \text{ speaks_for } v_{25})) \wedge \\
& \quad (\forall v_{28} v_{29}. P' (v_{28} \text{ controls } v_{29})) \wedge \\
& \quad (\forall v_{33} v_{34} v_{35}. P' (\text{reps } v_{33} v_{34} v_{35})) \wedge \\
& \quad (\forall v_{38} v_{39}. P' (v_{38} \text{ domi } v_{39})) \wedge \\
& \quad (\forall v_{42} v_{43}. P' (v_{42} \text{ eqi } v_{43})) \wedge \\
& \quad (\forall v_{46} v_{47}. P' (v_{46} \text{ doms } v_{47})) \wedge \\
& \quad (\forall v_{50} v_{51}. P' (v_{50} \text{ eqs } v_{51})) \wedge \\
& \quad (\forall v_{54} v_{55}. P' (v_{54} \text{ eqn } v_{55})) \wedge \\
& \quad (\forall v_{58} v_{59}. P' (v_{58} \text{ lte } v_{59})) \wedge \\
& \quad (\forall v_{62} v_{63}. P' (v_{62} \text{ lt } v_{63})) \Rightarrow \\
& \quad \forall v. P' v
\end{aligned}$$

[TR_cases]

$$\begin{aligned}
& \vdash \forall a_0 a_1 a_2 a_3. \\
& \quad \text{TR } a_0 a_1 a_2 a_3 \iff
\end{aligned}$$

$$\begin{aligned}
& (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 = \\
& \quad \text{CFG elementTest stateInterp context ins} \\
& \quad \quad (NS \ s \ (\text{exec } (\text{inputList } x))) \\
& \quad \quad (Out \ s \ (\text{exec } (\text{inputList } x))::outs)) \wedge \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \\
& \quad \quad \quad outs)) \vee \\
& (\exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 = \\
& \quad \text{CFG elementTest stateInterp context ins} \\
& \quad \quad (NS \ s \ (\text{trap } (\text{inputList } x))) \\
& \quad \quad (Out \ s \ (\text{trap } (\text{inputList } x))::outs)) \wedge \\
& \quad \text{authenticationTest elementTest } x \wedge \\
& \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \\
& \quad \quad \quad outs)) \vee \\
& \exists \text{elementTest } NS \ M \ Oi \ Os \ Out \ s \ context \ stateInterp \ x \ ins \\
& \quad outs. \\
& \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard } (\text{inputList } x)) \wedge \\
& \quad (a_2 = \\
& \quad \text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \wedge \\
& \quad (a_3 = \\
& \quad \text{CFG elementTest stateInterp context ins} \\
& \quad \quad (NS \ s \ (\text{discard } (\text{inputList } x))) \\
& \quad \quad (Out \ s \ (\text{discard } (\text{inputList } x))::outs)) \wedge \\
& \quad \neg \text{authenticationTest elementTest } x
\end{aligned}$$

[TR_discard_cmd_rule]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) \ (\text{discard } (\text{inputList } x)) \\
& \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \\
& \quad (\text{CFG elementTest stateInterp context ins} \\
& \quad \quad (NS \ s \ (\text{discard } (\text{inputList } x))) \\
& \quad \quad (Out \ s \ (\text{discard } (\text{inputList } x))::outs)) \iff \\
& \neg \text{authenticationTest elementTest } x
\end{aligned}$$

[TR_EQ_rules_thm]

$$\begin{aligned}
& \vdash (\text{TR } (M, Oi, Os) \ (\text{exec } (\text{inputList } x)) \\
& \quad (\text{CFG elementTest stateInterp context } (x::ins) \ s \ outs) \\
& \quad (\text{CFG elementTest stateInterp context ins}
\end{aligned}$$

$$\begin{aligned}
& (NS\ s\ (\text{exec}\ (\text{inputList}\ x))) \\
& (Out\ s\ (\text{exec}\ (\text{inputList}\ x))::outs)) \iff \\
& \text{authenticationTest}\ elementTest\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs)) \wedge \\
& (TR\ (M, Oi, Os)\ (\text{trap}\ (\text{inputList}\ x)) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ ins \\
& \quad \quad (NS\ s\ (\text{trap}\ (\text{inputList}\ x))) \\
& \quad \quad (Out\ s\ (\text{trap}\ (\text{inputList}\ x))::outs)) \iff \\
& \text{authenticationTest}\ elementTest\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs)) \wedge \\
& (TR\ (M, Oi, Os)\ (\text{discard}\ (\text{inputList}\ x)) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ ins \\
& \quad \quad (NS\ s\ (\text{discard}\ (\text{inputList}\ x))) \\
& \quad \quad (Out\ s\ (\text{discard}\ (\text{inputList}\ x))::outs)) \iff \\
& \neg \text{authenticationTest}\ elementTest\ x
\end{aligned}$$

[TR_exec_cmd_rule]

$$\begin{aligned}
& \vdash \forall elementTest\ context\ stateInterp\ x\ ins\ s\ outs. \\
& \quad (\forall M\ Oi\ Os. \\
& \quad \quad \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad \quad \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s \\
& \quad \quad \quad \quad outs) \Rightarrow \\
& \quad \quad (M, Oi, Os)\ \text{satList}\ \text{propCommandList}\ x) \Rightarrow \\
& \forall NS\ Out\ M\ Oi\ Os. \\
& \quad TR\ (M, Oi, Os)\ (\text{exec}\ (\text{inputList}\ x)) \\
& \quad \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad \quad (CFG\ elementTest\ stateInterp\ context\ ins \\
& \quad \quad \quad (NS\ s\ (\text{exec}\ (\text{inputList}\ x))) \\
& \quad \quad \quad (Out\ s\ (\text{exec}\ (\text{inputList}\ x))::outs)) \iff \\
& \text{authenticationTest}\ elementTest\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \wedge \\
& (M, Oi, Os)\ \text{satList}\ \text{propCommandList}\ x
\end{aligned}$$

[TR_ind]

$$\begin{aligned}
& \vdash \forall TR'. \\
& \quad (\forall elementTest\ NS\ M\ Oi\ Os\ Out\ s\ context\ stateInterp\ x\ ins \\
& \quad \quad outs. \\
& \quad \quad \text{authenticationTest}\ elementTest\ x \wedge \\
& \quad \quad \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad \quad \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s \\
& \quad \quad \quad \quad outs) \Rightarrow \\
& \quad \quad TR'\ (M, Oi, Os)\ (\text{exec}\ (\text{inputList}\ x)) \\
& \quad \quad \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad \quad \quad (CFG\ elementTest\ stateInterp\ context\ ins
\end{aligned}$$

$$\begin{aligned}
& (NS\ s\ (\text{exec}\ (\text{inputList}\ x))) \\
& (\text{Out}\ s\ (\text{exec}\ (\text{inputList}\ x))::\text{outs})) \wedge \\
& (\forall \text{elementTest}\ NS\ M\ Oi\ Os\ Out\ s\ \text{context}\ \text{stateInterp}\ x\ ins \\
& \quad \text{outs}. \\
& \text{authenticationTest}\ \text{elementTest}\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s \\
& \quad \quad \text{outs}) \Rightarrow \\
& TR' (M, Oi, Os) (\text{trap}\ (\text{inputList}\ x)) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ ins \\
& \quad \quad (NS\ s\ (\text{trap}\ (\text{inputList}\ x)))) \\
& \quad (\text{Out}\ s\ (\text{trap}\ (\text{inputList}\ x))::\text{outs})) \wedge \\
& (\forall \text{elementTest}\ NS\ M\ Oi\ Os\ Out\ s\ \text{context}\ \text{stateInterp}\ x\ ins \\
& \quad \text{outs}. \\
& \neg \text{authenticationTest}\ \text{elementTest}\ x \Rightarrow \\
& TR' (M, Oi, Os) (\text{discard}\ (\text{inputList}\ x)) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ ins \\
& \quad \quad (NS\ s\ (\text{discard}\ (\text{inputList}\ x)))) \\
& \quad (\text{Out}\ s\ (\text{discard}\ (\text{inputList}\ x))::\text{outs})) \Rightarrow \\
& \forall a_0\ a_1\ a_2\ a_3. TR\ a_0\ a_1\ a_2\ a_3 \Rightarrow TR'\ a_0\ a_1\ a_2\ a_3
\end{aligned}$$

[TR_rules]

$$\begin{aligned}
& \vdash (\forall \text{elementTest}\ NS\ M\ Oi\ Os\ Out\ s\ \text{context}\ \text{stateInterp}\ x\ ins \\
& \quad \text{outs}. \\
& \text{authenticationTest}\ \text{elementTest}\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \Rightarrow \\
& TR (M, Oi, Os) (\text{exec}\ (\text{inputList}\ x)) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ ins \\
& \quad \quad (NS\ s\ (\text{exec}\ (\text{inputList}\ x)))) \\
& \quad (\text{Out}\ s\ (\text{exec}\ (\text{inputList}\ x))::\text{outs})) \wedge \\
& (\forall \text{elementTest}\ NS\ M\ Oi\ Os\ Out\ s\ \text{context}\ \text{stateInterp}\ x\ ins \\
& \quad \text{outs}. \\
& \text{authenticationTest}\ \text{elementTest}\ x \wedge \\
& \text{CFGInterpret}\ (M, Oi, Os) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \Rightarrow \\
& TR (M, Oi, Os) (\text{trap}\ (\text{inputList}\ x)) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ ins \\
& \quad \quad (NS\ s\ (\text{trap}\ (\text{inputList}\ x)))) \\
& \quad (\text{Out}\ s\ (\text{trap}\ (\text{inputList}\ x))::\text{outs})) \wedge \\
& \forall \text{elementTest}\ NS\ M\ Oi\ Os\ Out\ s\ \text{context}\ \text{stateInterp}\ x\ ins\ \text{outs}. \\
& \neg \text{authenticationTest}\ \text{elementTest}\ x \Rightarrow \\
& TR (M, Oi, Os) (\text{discard}\ (\text{inputList}\ x)) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ (x::ins)\ s\ \text{outs}) \\
& \quad (\text{CFG}\ \text{elementTest}\ \text{stateInterp}\ \text{context}\ ins
\end{aligned}$$

(NS s (discard (inputList x)))
 (Out s (discard (inputList x))::outs))

[TR_strongind]

$\vdash \forall TR'.$

(\forall elementTest NS M Oi Os Out s context stateInterp x ins
 outs.
 authenticationTest elementTest x \wedge
 CFGInterpret (M,Oi,Os)
 (CFG elementTest stateInterp context (x::ins) s
 outs) \Rightarrow
 TR' (M,Oi,Os) (exec (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (exec (inputList x)))
 (Out s (exec (inputList x))::outs))) \wedge
 (\forall elementTest NS M Oi Os Out s context stateInterp x ins
 outs.
 authenticationTest elementTest x \wedge
 CFGInterpret (M,Oi,Os)
 (CFG elementTest stateInterp context (x::ins) s
 outs) \Rightarrow
 TR' (M,Oi,Os) (trap (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (trap (inputList x)))
 (Out s (trap (inputList x))::outs))) \wedge
 (\forall elementTest NS M Oi Os Out s context stateInterp x ins
 outs.
 \neg authenticationTest elementTest x \Rightarrow
 TR' (M,Oi,Os) (discard (inputList x))
 (CFG elementTest stateInterp context (x::ins) s outs)
 (CFG elementTest stateInterp context ins
 (NS s (discard (inputList x)))
 (Out s (discard (inputList x))::outs))) \Rightarrow
 $\forall a_0 a_1 a_2 a_3. \text{TR } a_0 a_1 a_2 a_3 \Rightarrow \text{TR}' a_0 a_1 a_2 a_3$

[TR_trap_cmd_rule]

$\vdash \forall$ elementTest context stateInterp x ins s outs.

(\forall M Oi Os.

CFGInterpret (M,Oi,Os)

(CFG elementTest stateInterp context (x::ins) s
 outs) \Rightarrow

(M,Oi,Os) sat prop NONE) \Rightarrow

\forall NS Out M Oi Os.

TR (M,Oi,Os) (trap (inputList x))

(CFG elementTest stateInterp context (x::ins) s outs)

(CFG elementTest stateInterp context ins

(NS s (trap (inputList x)))

$$\begin{aligned}
& (Out\ s\ (trap\ (inputList\ x))::outs)) \iff \\
& authenticationTest\ elementTest\ x \wedge \\
& CFGInterpret\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \wedge \\
& (M, Oi, Os)\ sat\ prop\ NONE
\end{aligned}$$

[TRrule0]

$$\begin{aligned}
& \vdash TR\ (M, Oi, Os)\ (exec\ (inputList\ x)) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ ins \\
& \quad \quad (NS\ s\ (exec\ (inputList\ x))) \\
& \quad \quad (Out\ s\ (exec\ (inputList\ x))::outs)) \iff \\
& authenticationTest\ elementTest\ x \wedge \\
& CFGInterpret\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs)
\end{aligned}$$

[TRrule1]

$$\begin{aligned}
& \vdash TR\ (M, Oi, Os)\ (trap\ (inputList\ x)) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ ins \\
& \quad \quad (NS\ s\ (trap\ (inputList\ x))) \\
& \quad \quad (Out\ s\ (trap\ (inputList\ x))::outs)) \iff \\
& authenticationTest\ elementTest\ x \wedge \\
& CFGInterpret\ (M, Oi, Os) \\
& \quad (CFG\ elementTest\ stateInterp\ context\ (x::ins)\ s\ outs)
\end{aligned}$$

[trType_distinct_clauses]

$$\begin{aligned}
& \vdash (\forall a'\ a.\ discard\ a \neq trap\ a') \wedge (\forall a'\ a.\ discard\ a \neq exec\ a') \wedge \\
& \quad \forall a'\ a.\ trap\ a \neq exec\ a'
\end{aligned}$$

[trType_one_one]

$$\begin{aligned}
& \vdash (\forall a\ a'. (discard\ a = discard\ a') \iff (a = a')) \wedge \\
& \quad (\forall a\ a'. (trap\ a = trap\ a') \iff (a = a')) \wedge \\
& \quad \forall a\ a'. (exec\ a = exec\ a') \iff (a = a')
\end{aligned}$$

3 satList Theory

Built: 16 May 2018

Parent Theories: aclDrules

3.1 Definitions

[satList_def]

$$\begin{aligned}
& \vdash \forall M\ Oi\ Os\ formList. \\
& \quad (M, Oi, Os)\ satList\ formList \iff \\
& \quad FOLDR\ (\lambda x\ y.\ x \wedge y)\ T\ (MAP\ (\lambda f.\ (M, Oi, Os)\ sat\ f)\ formList)
\end{aligned}$$

3.2 Theorems

[satList_conj]

$$\begin{aligned} &\vdash \forall l_1 \ l_2 \ M \ Oi \ Os. \\ &\quad (M, Oi, Os) \text{ satList } l_1 \wedge (M, Oi, Os) \text{ satList } l_2 \iff \\ &\quad (M, Oi, Os) \text{ satList } (l_1 ++ l_2) \end{aligned}$$

[satList_CONS]

$$\begin{aligned} &\vdash \forall h \ t \ M \ Oi \ Os. \\ &\quad (M, Oi, Os) \text{ satList } (h :: t) \iff \\ &\quad (M, Oi, Os) \text{ sat } h \wedge (M, Oi, Os) \text{ satList } t \end{aligned}$$

[satList_nil]

$$\vdash (M, Oi, Os) \text{ satList } []$$

Index

satList Theory, 19

Definitions, 19

satList_def, 19

Theorems, 20

satList_conj, 20

satList_CONS, 20

satList_nil, 20

ssm Theory, 10

Datatypes, 10

Definitions, 10

authenticationTest_def, 10

commandList_def, 10

inputList_def, 10

propCommandList_def, 10

TR_def, 11

Theorems, 11

CFGInterpret_def, 12

CFGInterpret_ind, 12

configuration_one_one, 12

extractCommand_def, 12

extractCommand_ind, 12

extractInput_def, 13

extractInput_ind, 13

extractPropCommand_def, 14

extractPropCommand_ind, 14

TR_cases, 14

TR_discard_cmd_rule, 15

TR_EQ_rules_thm, 15

TR_exec_cmd_rule, 16

TR_ind, 16

TR_rules, 17

TR_strongind, 18

TR_trap_cmd_rule, 18

TRrule0, 19

TRrule1, 19

trType_distinct_clauses, 19

trType_one_one, 19

ssm11 Theory, 3

Datatypes, 3

Definitions, 3

TR_def, 3

Theorems, 4

CFGInterpret_def, 4

CFGInterpret_ind, 4

configuration_one_one, 4

order_distinct_clauses, 5

order_one_one, 5

TR_cases, 5

TR_discard_cmd_rule, 5

TR_EQ_rules_thm, 6

TR_exec_cmd_rule, 6

TR_ind, 7

TR_rules, 7

TR_strongind, 8

TR_trap_cmd_rule, 9

TRrule0, 9

TRrule1, 9

trType_distinct_clauses, 10

trType_one_one, 10