

Contents

1	ssm1 Theory	3
1.1	Datatypes	3
1.2	Definitions	3
1.3	Theorems	4

1 ssm1 Theory

Built: 09 April 2019

Parent Theories: satList

1.1 Datatypes

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

```
inst = SOME 'command | NONE
```

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

1.2 Definitions

[TR_def]

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

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

1.3 Theorems

[CFGInterpret_def]

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

[CFGInterpret_ind]

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

[configuration_one_one]

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

[inst_distinct_clauses]

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

[inst_one_one]

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

[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 \\
& \quad (\exists \text{inputTest } P \ NS \ M \ Oi \ Os \ Out \ s \ \text{certList } \text{stateInterp } \text{cmd } ins \\
& \quad \quad \text{outs}. \\
& \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{exec } \text{cmd}) \wedge \\
& \quad \quad (a_2 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \ \text{outs}) \wedge \\
& \quad \quad (a_3 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList ins} \\
& \quad \quad \quad (NS \ s \ (\text{exec } \text{cmd})) \ (Out \ s \ (\text{exec } \text{cmd}::outs)) \wedge \\
& \quad \quad \text{inputTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad \quad (\text{CFG inputTest stateInterp certList} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \ \text{outs})) \vee \\
& \quad (\exists \text{inputTest } P \ NS \ M \ Oi \ Os \ Out \ s \ \text{certList } \text{stateInterp } \text{cmd } ins \\
& \quad \quad \text{outs}. \\
& \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{trap } \text{cmd}) \wedge \\
& \quad \quad (a_2 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \ \text{outs}) \wedge \\
& \quad \quad (a_3 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList ins} \\
& \quad \quad \quad (NS \ s \ (\text{trap } \text{cmd})) \ (Out \ s \ (\text{trap } \text{cmd}::outs)) \wedge \\
& \quad \quad \text{inputTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \quad \quad \text{CFGInterpret } (M, Oi, Os) \\
& \quad \quad \quad (\text{CFG inputTest stateInterp certList} \\
& \quad \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \ \text{outs})) \vee \\
& \quad \exists \text{inputTest } NS \ M \ Oi \ Os \ Out \ s \ \text{certList } \text{stateInterp } \text{cmd } x \ ins \\
& \quad \quad \text{outs}. \\
& \quad \quad (a_0 = (M, Oi, Os)) \wedge (a_1 = \text{discard}) \wedge \\
& \quad \quad (a_2 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList } (x::ins) \ s \ \text{outs}) \wedge \\
& \quad \quad (a_3 = \\
& \quad \quad \quad \text{CFG inputTest stateInterp certList ins } (NS \ s \ \text{discard}) \\
& \quad \quad \quad (Out \ s \ \text{discard}::outs)) \wedge \neg \text{inputTest } x
\end{aligned}$$
[TR_discard_cmd_rule]

$$\begin{aligned}
& \vdash \text{TR } (M, Oi, Os) \ \text{discard} \\
& \quad (\text{CFG inputTest stateInterp certList } (x::ins) \ s \ \text{outs}) \\
& \quad (\text{CFG inputTest stateInterp certList ins } (NS \ s \ \text{discard}) \\
& \quad \quad (Out \ s \ \text{discard}::outs)) \iff \neg \text{inputTest } x
\end{aligned}$$
[TR_EQ_rules_thm]

$$\begin{aligned}
& \vdash (\text{TR } (M, Oi, Os) \ (\text{exec } \text{cmd}) \\
& \quad (\text{CFG inputTest stateInterp certList} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \ \text{outs})
\end{aligned}$$

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

[TR_exec_cmd_rule]

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

[TR_ind]

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

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

[TR_rules]

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

$\text{TR } (M, Oi, Os) \text{ discard}$
 $(\text{CFG } \text{inputTest } \text{stateInterp } \text{certList } (x::\text{ins}) \text{ } s \text{ } \text{outs})$
 $(\text{CFG } \text{inputTest } \text{stateInterp } \text{certList } \text{ins } (\text{NS } s \text{ discard})$
 $(\text{Out } s \text{ discard}::\text{outs}))$

[TR_strongind]

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

[TR_trap_cmd_rule]

$\vdash \forall \text{inputTest } \text{stateInterp } \text{certList } P \text{ cmd } \text{ins } s \text{ } \text{outs}.$
 $(\forall M \text{ } Oi \text{ } Os.$
 $\text{CFGInterpret } (M, Oi, Os)$
 $(\text{CFG } \text{inputTest } \text{stateInterp } \text{certList}$
 $(P \text{ says prop } (\text{SOME } \text{cmd})::\text{ins}) \text{ } s \text{ } \text{outs}) \Rightarrow$
 $(M, Oi, Os) \text{ sat prop NONE}) \Rightarrow$
 $\forall \text{NS } \text{Out } M \text{ } Oi \text{ } Os.$
 $\text{TR } (M, Oi, Os) (\text{trap } \text{cmd})$
 $(\text{CFG } \text{inputTest } \text{stateInterp } \text{certList}$
 $(P \text{ says prop } (\text{SOME } \text{cmd})::\text{ins}) \text{ } s \text{ } \text{outs})$

$$\begin{aligned}
& (\text{CFG } \text{inputTest } \text{stateInterp } \text{certList } \text{ins} \\
& \quad (\text{NS } s \text{ (trap cmd)}) (\text{Out } s \text{ (trap cmd)::outs})) \iff \\
& \text{inputTest } (P \text{ says prop (SOME cmd)}) \wedge \\
& \text{CFGInterpret } (M, Oi, Os) \\
& \quad (\text{CFG } \text{inputTest } \text{stateInterp } \text{certList} \\
& \quad \quad (P \text{ says prop (SOME cmd)::ins) } s \text{ outs}) \wedge \\
& (M, Oi, Os) \text{ sat prop NONE}
\end{aligned}$$

[TRrule0]

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

[TRrule1]

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

[trType_distinct_clauses]

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

[trType_one_one]

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

Index

ssm1 Theory, 3
 Datatypes, 3
 Definitions, 3
 TR_def, 3
 Theorems, 4
 CFGInterpret_def, 4
 CFGInterpret_ind, 4
 configuration_one_one, 4
 inst_distinct_clauses, 4
 inst_one_one, 4
 TR_cases, 5
 TR_discard_cmd_rule, 5
 TR_EQ_rules_thm, 5
 TR_exec_cmd_rule, 6
 TR_ind, 6
 TR_rules, 7
 TR_strongind, 8
 TR_trap_cmd_rule, 8
 TRrule0, 9
 TRrule1, 9
 trType_distinct_clauses, 9
 trType_one_one, 9