Contents

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)) ∧
```

$$\begin{array}{l}
\qquad\qquad\texttt{CFGInterpret}\ (M\texttt{,}Oi\texttt{,}Os)\\
\qquad\qquad\quad(\texttt{CFG}\ inputTest\ stateInterp\ certList\\
\qquad\qquad\qquad(P\ \texttt{says}\ \texttt{prop}\ (\texttt{SOME}\ cmd)\texttt{::}ins)\ s\ outs))\ \lor\\
\qquad\quad(\exists\, inputTest\ NS\ M\ Oi\ Os\ Out\ s\ certList\ stateInterp\ cmd\\
\qquad\qquad x\ ins\ outs\texttt{.}\\
\qquad\qquad(a_0\ \texttt{=}\ (M\texttt{,}Oi\texttt{,}Os))\ \land\ (a_1\ \texttt{=}\ \texttt{discard})\ \land\\
\qquad\qquad(a_2\ \texttt{=}\\
\qquad\qquad\ \texttt{CFG}\ inputTest\ stateInterp\ certList\ (x\texttt{::}ins)\ s\\
\qquad\qquad\qquad outs)\ \land\\
\qquad\qquad(a_3\ \texttt{=}\\
\qquad\qquad\ \texttt{CFG}\ inputTest\ stateInterp\ certList\ ins\\
\qquad\qquad\qquad(NS\ s\ \texttt{discard})\ (Out\ s\ \texttt{discard::}outs))\ \land\\
\qquad\qquad\neg inputTest\ x)\ \Rightarrow\\
\qquad\quad TR'\ a_0\ a_1\ a_2\ a_3)\ \Rightarrow\\
\qquad TR'\ a_0\ a_1\ a_2\ a_3)
\end{array}$$

1.3 Theorems

[CFGInterpret_def]

$$\begin{array}{l}
\vdash\ \texttt{CFGInterpret}\ (M\texttt{,}Oi\texttt{,}Os)\\
\qquad(\texttt{CFG}\ inputTest\ stateInterp\ context\ (x\texttt{::}ins)\ state\\
\qquad\quad outStream)\ \iff\\
\quad(M\texttt{,}Oi\texttt{,}Os)\ \texttt{satList}\ context\ \land\ (M\texttt{,}Oi\texttt{,}Os)\ \texttt{sat}\ x\ \land\\
\quad(M\texttt{,}Oi\texttt{,}Os)\ \texttt{sat}\ stateInterp\ state
\end{array}$$

[CFGInterpret_ind]

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

[configuration_one_one]

$$\begin{array}{l}
\vdash\ \forall\, a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ a_0'\ a_1'\ a_2'\ a_3'\ a_4'\ a_5'\texttt{.}\\
\qquad(\texttt{CFG}\ a_0\ a_1\ a_2\ a_3\ a_4\ a_5\ \texttt{=}\ \texttt{CFG}\ a_0'\ a_1'\ a_2'\ a_3'\ a_4'\ a_5')\ \iff\\
\qquad(a_0\ \texttt{=}\ a_0')\ \land\ (a_1\ \texttt{=}\ a_1')\ \land\ (a_2\ \texttt{=}\ a_2')\ \land\ (a_3\ \texttt{=}\ a_3')\ \land\\
\qquad(a_4\ \texttt{=}\ a_4')\ \land\ (a_5\ \texttt{=}\ a_5')
\end{array}$$

[inst_distinct_clauses]

$$\vdash\ \forall\, a\texttt{.}\ \texttt{SOME}\ a\ \neq\ \texttt{NONE}$$

[inst_one_one]

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

[TR_cases]

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

[TR_discard_cmd_rule]

```
⊢ TR (M,Oi,Os) discard
    (CFG inputTest stateInterp certList (x::ins) s outs)
    (CFG inputTest stateInterp certList ins (NS s discard)
       (Out s discard::outs)) ⇔ ¬inputTest x
```

[TR_EQ_rules_thm]

```
⊢ (TR (M,Oi,Os) (exec cmd)
     (CFG inputTest stateInterp certList
        (P says prop (SOME cmd)::ins) s outs)
        (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)) ∧
    (TR (M,Oi,Os) (trap cmd)
        (CFG inputTest stateInterp certList
           (P says prop (SOME cmd)::ins) s outs)
        (CFG inputTest stateInterp certList ins (NS s (trap cmd))
           (Out s (trap cmd)::outs)) ⇔
      inputTest (P says prop (SOME cmd)) ∧
      CFGInterpret (M,Oi,Os)
        (CFG inputTest stateInterp certList
           (P says prop (SOME cmd)::ins) s outs)) ∧
    (TR (M,Oi,Os) discard
        (CFG inputTest stateInterp certList (x::ins) s outs)
        (CFG inputTest stateInterp certList ins (NS s discard)
           (Out s discard::outs)) ⇔ ¬inputTest x)
```

[TR_exec_cmd_rule]

```
 ⊢ ∀inputTest certList stateInterp P cmd ins s outs.
       (∀M Oi Os.
          CFGInterpret (M,Oi,Os)
            (CFG inputTest stateInterp certList
               (P says prop (SOME cmd)::ins) s outs) ⇒
          (M,Oi,Os) sat prop (SOME cmd)) ⇒
       ∀NS Out M Oi Os.
         TR (M,Oi,Os) (exec cmd)
           (CFG inputTest stateInterp certList
              (P says prop (SOME cmd)::ins) s outs)
           (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) ∧
         (M,Oi,Os) sat prop (SOME cmd)
```

[TR_ind]

```
 ⊢ ∀TR′.
       (∀inputTest P NS M Oi Os Out s certList stateInterp cmd ins
           outs.
          inputTest (P says prop (SOME cmd)) ∧
          CFGInterpret (M,Oi,Os)
            (CFG inputTest stateInterp certList
               (P says prop (SOME cmd)::ins) s outs) ⇒
          TR′ (M,Oi,Os) (exec cmd)
            (CFG inputTest stateInterp certList
           (P says prop (SOME cmd)::ins) s outs)
        (CFG inputTest stateInterp certList ins
           (NS s (exec cmd)) (Out s (exec cmd)::outs))) ∧
  (∀ inputTest P NS M Oi Os Out s certList stateInterp cmd ins
      outs.
     inputTest (P says prop (SOME cmd)) ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs) ⇒
     TR′ (M,Oi,Os) (trap cmd)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs)
       (CFG inputTest stateInterp certList ins
          (NS s (trap cmd)) (Out s (trap cmd)::outs))) ∧
  (∀ inputTest NS M Oi Os Out s certList stateInterp cmd x ins
      outs.
     ¬inputTest x ⇒
     TR′ (M,Oi,Os) discard
       (CFG inputTest stateInterp certList (x::ins) s outs)
       (CFG inputTest stateInterp certList ins (NS s discard)
          (Out s discard::outs))) ⇒
  ∀ a0 a1 a2 a3. TR a0 a1 a2 a3 ⇒ TR′ a0 a1 a2 a3
```

[TR_rules]

```
⊢ (∀ inputTest P NS M Oi Os Out s certList stateInterp cmd ins
      outs.
     inputTest (P says prop (SOME cmd)) ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs) ⇒
     TR (M,Oi,Os) (exec cmd)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs)
       (CFG inputTest stateInterp certList ins
          (NS s (exec cmd)) (Out s (exec cmd)::outs))) ∧
  (∀ inputTest P NS M Oi Os Out s certList stateInterp cmd ins
      outs.
     inputTest (P says prop (SOME cmd)) ∧
     CFGInterpret (M,Oi,Os)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs) ⇒
     TR (M,Oi,Os) (trap cmd)
       (CFG inputTest stateInterp certList
          (P says prop (SOME cmd)::ins) s outs)
       (CFG inputTest stateInterp certList ins
          (NS s (trap cmd)) (Out s (trap cmd)::outs))) ∧
  ∀ inputTest NS M Oi Os Out s certList stateInterp cmd x ins
    outs.
    ¬inputTest x ⇒
```

```
     TR (M,Oi,Os) discard
       (CFG inputTest stateInterp certList (x::ins) s outs)
       (CFG inputTest stateInterp certList ins (NS s discard)
          (Out s discard::outs))
```

[TR_strongind]

```
 ⊢ ∀ TR′.
     (∀ inputTest P NS M Oi Os Out s certList stateInterp cmd ins
         outs.
        inputTest (P says prop (SOME cmd)) ∧
        CFGInterpret (M,Oi,Os)
          (CFG inputTest stateInterp certList
             (P says prop (SOME cmd)::ins) s outs) ⇒
        TR′ (M,Oi,Os) (exec cmd)
          (CFG inputTest stateInterp certList
             (P says prop (SOME cmd)::ins) s outs)
          (CFG inputTest stateInterp certList ins
             (NS s (exec cmd)) (Out s (exec cmd)::outs))) ∧
     (∀ inputTest P NS M Oi Os Out s certList stateInterp cmd ins
         outs.
        inputTest (P says prop (SOME cmd)) ∧
        CFGInterpret (M,Oi,Os)
          (CFG inputTest stateInterp certList
             (P says prop (SOME cmd)::ins) s outs) ⇒
        TR′ (M,Oi,Os) (trap cmd)
          (CFG inputTest stateInterp certList
             (P says prop (SOME cmd)::ins) s outs)
          (CFG inputTest stateInterp certList ins
             (NS s (trap cmd)) (Out s (trap cmd)::outs))) ∧
     (∀ inputTest NS M Oi Os Out s certList stateInterp x ins
        outs.
        ¬inputTest x ⇒
        TR′ (M,Oi,Os) discard
          (CFG inputTest stateInterp certList (x::ins) s outs)
          (CFG inputTest stateInterp certList ins (NS s discard)
             (Out s discard::outs))) ⇒
     ∀ a0 a1 a2 a3. TR a0 a1 a2 a3 ⇒ TR′ a0 a1 a2 a3
```

[TR_trap_cmd_rule]

```
 ⊢ ∀ inputTest stateInterp certList P cmd ins s outs.
     (∀ M Oi Os.
        CFGInterpret (M,Oi,Os)
          (CFG inputTest stateInterp certList
             (P says prop (SOME cmd)::ins) s outs) ⇒
        (M,Oi,Os) sat prop NONE) ⇒
     ∀ NS Out M Oi Os.
       TR (M,Oi,Os) (trap cmd)
         (CFG inputTest stateInterp certList
            (P says prop (SOME cmd)::ins) s outs)
```

```
      (CFG inputTest stateInterp certList ins
         (NS s (trap cmd)) (Out s (trap cmd)::outs)) ⟺
    inputTest (P says prop (SOME cmd)) ∧
    CFGInterpret (M,Oi,Os)
      (CFG inputTest stateInterp certList
         (P says prop (SOME cmd)::ins) s outs) ∧
    (M,Oi,Os) sat prop NONE
```

[TRrule0]

```
⊢ TR (M,Oi,Os) (exec cmd)
    (CFG inputTest stateInterp certList
       (P says prop (SOME cmd)::ins) s outs)
    (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)
```

[TRrule1]

```
⊢ TR (M,Oi,Os) (trap cmd)
    (CFG inputTest stateInterp certList
       (P says prop (SOME cmd)::ins) s outs)
    (CFG inputTest stateInterp certList ins (NS s (trap cmd))
       (Out s (trap cmd)::outs)) ⟺
  inputTest (P says prop (SOME cmd)) ∧
  CFGInterpret (M,Oi,Os)
    (CFG inputTest stateInterp certList
       (P says prop (SOME cmd)::ins) s outs)
```

[trType_distinct_clauses]

```
⊢ (∀ a. discard ≠ trap a) ∧ (∀ a. discard ≠ exec a) ∧
  ∀ a′ a. trap a ≠ exec a′
```

[trType_one_one]

```
⊢ (∀ a a′. (trap a = trap a′) ⟺ (a = a′)) ∧
  ∀ a a′. (exec a = exec a′) ⟺ (a = a′)
```

Index