

Contents

1	m1 Theory	3
1.1	Datatypes	3
1.2	Theorems	3
2	sm Theory	4
2.1	Datatypes	5
2.2	Definitions	5
2.3	Theorems	5

1 m1 Theory

Built: 02 March 2020

Parent Theories: sm

1.1 Datatypes

command = i0 | i1

output = o0 | o1

state = S0 | S1 | S2

1.2 Theorems

[command_distinct_clauses]

$\vdash i0 \neq i1$

[m1_rules]

$\vdash (\forall ins\ outs.$
 TR i0 (CFG (i0::ins) S0 outs) (CFG ins S1 (o0::outs))) \wedge
 $(\forall ins\ outs.$
 TR i1 (CFG (i1::ins) S0 outs) (CFG ins S2 (o1::outs))) \wedge
 $(\forall ins\ outs.$
 TR i0 (CFG (i0::ins) S1 outs) (CFG ins S0 (o0::outs))) \wedge
 $(\forall ins\ outs.$
 TR i1 (CFG (i1::ins) S1 outs) (CFG ins S0 (o0::outs))) \wedge
 $(\forall ins\ outs.$
 TR i0 (CFG (i0::ins) S2 outs) (CFG ins S2 (o1::outs))) \wedge
 $\forall ins\ outs.$
 TR i1 (CFG (i1::ins) S2 outs) (CFG ins S2 (o1::outs)))

[M1ns_def]

$\vdash (M1ns\ S0\ i0 = S1) \wedge (M1ns\ S0\ i1 = S2) \wedge (M1ns\ S1\ i0 = S0) \wedge$
 $(M1ns\ S1\ i1 = S0) \wedge (M1ns\ S2\ i0 = S2) \wedge (M1ns\ S2\ i1 = S2)$

[M1ns_ind]

$\vdash \forall P.$
 $P\ S0\ i0 \wedge P\ S0\ i1 \wedge P\ S1\ i0 \wedge P\ S1\ i1 \wedge P\ S2\ i0 \wedge P\ S2\ i1 \Rightarrow$
 $\forall v\ v_1. P\ v\ v_1$

[M1out_def]

$$\vdash (\text{M1out } S0 \text{ i0} = \text{o0}) \wedge (\text{M1out } S0 \text{ i1} = \text{o1}) \wedge \\ (\text{M1out } S1 \text{ i0} = \text{o0}) \wedge (\text{M1out } S1 \text{ i1} = \text{o0}) \wedge \\ (\text{M1out } S2 \text{ i0} = \text{o1}) \wedge (\text{M1out } S2 \text{ i1} = \text{o1})$$

[M1out_ind]

$$\vdash \forall P. \\ P \text{ S0 i0} \wedge P \text{ S0 i1} \wedge P \text{ S1 i0} \wedge P \text{ S1 i1} \wedge P \text{ S2 i0} \wedge P \text{ S2 i1} \Rightarrow \\ \forall v \ v_1. \ P \ v \ v_1$$

[m1TR_clauses]

$$\vdash (\forall x \ x1s \ s_1 \ out1s \ x2s \ out2s \ s_2. \\ \text{TR } x \ (\text{CFG } x1s \ s_1 \ out1s) \ (\text{CFG } x2s \ s_2 \ out2s) \iff \\ \exists NS \ Out \ ins. \\ (x1s = x::ins) \wedge (x2s = ins) \wedge (s_2 = NS \ s_1 \ x) \wedge \\ (out2s = Out \ s_1 \ x::out1s)) \wedge \\ \forall x \ x1s \ s_1 \ out1s \ x2s \ out2s. \\ \text{TR } x \ (\text{CFG } x1s \ s_1 \ out1s) \\ (\text{CFG } x2s \ (\text{Mins } s_1 \ x) \ (\text{M1out } s_1 \ x::out2s)) \iff \\ \exists ins. (x1s = x::ins) \wedge (x2s = ins) \wedge (out2s = out1s))$$

[m1TR_rules]

$$\vdash \forall s \ x \ ins \ outs. \\ \text{TR } x \ (\text{CFG } (x::ins) \ s \ outs) \\ (\text{CFG } ins \ (\text{Mins } s \ x) \ (\text{M1out } s \ x::outs))$$

[m1Trans_Equiv_TR]

$$\vdash \text{TR } x \ (\text{CFG } (x::ins) \ s \ outs) \\ (\text{CFG } ins \ (\text{Mins } s \ x) \ (\text{M1out } s \ x::outs)) \iff \\ \text{Trans } x \ s \ (\text{Mins } s \ x)$$

[output_distinct_clauses]

$$\vdash \text{o0} \neq \text{o1}$$

[state_distinct_clauses]

$$\vdash S0 \neq S1 \wedge S0 \neq S2 \wedge S1 \neq S2$$

2 sm Theory

Built: 02 March 2020

Parent Theories: indexedLists, patternMatches

2.1 Datatypes

configuration = CFG ('input list) 'state ('output list)

2.2 Definitions

[TR_def]

$$\begin{aligned} \vdash \text{TR} = & \\ & (\lambda a_0 a_1 a_2. \\ & \quad \forall TR'. \\ & \quad (\forall a_0 a_1 a_2. \\ & \quad \quad (\exists NS \text{ Out } s \text{ ins } \text{ outs}. \\ & \quad \quad \quad (a_1 = \text{CFG } (a_0 :: \text{ins}) \text{ } s \text{ } \text{outs}) \wedge \\ & \quad \quad \quad (a_2 = \text{CFG } \text{ins } (NS \text{ } s \text{ } a_0) (\text{Out } s \text{ } a_0 :: \text{outs}))) \Rightarrow \\ & \quad \quad TR' a_0 a_1 a_2) \Rightarrow \\ & \quad TR' a_0 a_1 a_2) \end{aligned}$$

[Trans_def]

$$\begin{aligned} \vdash \text{Trans} = & \\ & (\lambda a_0 a_1 a_2. \\ & \quad \forall Trans'. \\ & \quad (\forall a_0 a_1 a_2. (\exists NS. a_2 = NS a_1 a_0) \Rightarrow Trans' a_0 a_1 a_2) \Rightarrow \\ & \quad Trans' a_0 a_1 a_2) \end{aligned}$$

2.3 Theorems

[configuration_one_one]

$$\begin{aligned} \vdash \forall a_0 a_1 a_2 a'_0 a'_1 a'_2. \\ & (\text{CFG } a_0 a_1 a_2 = \text{CFG } a'_0 a'_1 a'_2) \iff \\ & (a_0 = a'_0) \wedge (a_1 = a'_1) \wedge (a_2 = a'_2) \end{aligned}$$

[TR_cases]

$$\begin{aligned} \vdash \forall a_0 a_1 a_2. \\ & \text{TR } a_0 a_1 a_2 \iff \\ & \exists NS \text{ Out } s \text{ ins } \text{ outs}. \\ & \quad (a_1 = \text{CFG } (a_0 :: \text{ins}) \text{ } s \text{ } \text{outs}) \wedge \\ & \quad (a_2 = \text{CFG } \text{ins } (NS \text{ } s \text{ } a_0) (\text{Out } s \text{ } a_0 :: \text{outs})) \end{aligned}$$

[TR_clauses]

$$\begin{aligned} \vdash (\forall x \text{ x1s } s_1 \text{ out1s } x2s \text{ out2s } s_2. \\ & \text{TR } x (\text{CFG } x1s \text{ } s_1 \text{ } \text{out1s}) (\text{CFG } x2s \text{ } s_2 \text{ } \text{out2s}) \iff \\ & \exists NS \text{ Out } \text{ins}. \\ & \quad (x1s = x :: \text{ins}) \wedge (x2s = \text{ins}) \wedge (s_2 = NS \text{ } s_1 \text{ } x) \wedge \end{aligned}$$

$$\begin{aligned}
& (out2s = Out\ s_1\ x :: out1s)) \wedge \\
& \forall NS\ Out\ x\ x1s\ s_1\ out1s\ x2s\ out2s. \\
& TR\ x\ (CFG\ x1s\ s_1\ out1s) \\
& (CFG\ x2s\ (NS\ s_1\ x)\ (Out\ s_1\ x :: out2s)) \iff \\
& \exists ins. (x1s = x :: ins) \wedge (x2s = ins) \wedge (out2s = out1s)
\end{aligned}$$

[TR_complete]

$$\begin{aligned}
& \vdash \forall s\ x\ ins\ outs. \\
& \quad \exists s'\ out. \\
& \quad TR\ x\ (CFG\ (x :: ins)\ s\ outs)\ (CFG\ ins\ s'\ (out :: outs))
\end{aligned}$$

[TR_deterministic]

$$\begin{aligned}
& \vdash \forall NS\ Out\ x_1\ ins_1\ s_1\ outs_1\ ins'_2\ outs_2\ outs'_2. \\
& TR\ x_1\ (CFG\ (x_1 :: ins_1)\ s_1\ outs_1) \\
& (CFG\ ins_2\ (NS\ s_1\ x_1)\ (Out\ s_1\ x_1 :: outs_2)) \wedge \\
& TR\ x_1\ (CFG\ (x_1 :: ins_1)\ s_1\ outs_1) \\
& (CFG\ ins'_2\ (NS\ s_1\ x_1)\ (Out\ s_1\ x_1 :: outs'_2)) \iff \\
& (CFG\ ins_2\ (NS\ s_1\ x_1)\ (Out\ s_1\ x_1 :: outs_2)) = \\
& CFG\ ins'_2\ (NS\ s_1\ x_1)\ (Out\ s_1\ x_1 :: outs'_2)) \wedge \\
& TR\ x_1\ (CFG\ (x_1 :: ins_1)\ s_1\ outs_1) \\
& (CFG\ ins_2\ (NS\ s_1\ x_1)\ (Out\ s_1\ x_1 :: outs_2))
\end{aligned}$$

[TR_ind]

$$\begin{aligned}
& \vdash \forall TR'. \\
& (\forall NS\ Out\ s\ x\ ins\ outs. \\
& \quad TR'\ x\ (CFG\ (x :: ins)\ s\ outs) \\
& \quad (CFG\ ins\ (NS\ s\ x)\ (Out\ s\ x :: outs))) \Rightarrow \\
& \forall a_0\ a_1\ a_2. TR\ a_0\ a_1\ a_2 \Rightarrow TR'\ a_0\ a_1\ a_2
\end{aligned}$$

[TR_rules]

$$\begin{aligned}
& \vdash \forall NS\ Out\ s\ x\ ins\ outs. \\
& TR\ x\ (CFG\ (x :: ins)\ s\ outs) \\
& (CFG\ ins\ (NS\ s\ x)\ (Out\ s\ x :: outs))
\end{aligned}$$

[TR_strongind]

$$\begin{aligned}
& \vdash \forall TR'. \\
& (\forall NS\ Out\ s\ x\ ins\ outs. \\
& \quad TR'\ x\ (CFG\ (x :: ins)\ s\ outs) \\
& \quad (CFG\ ins\ (NS\ s\ x)\ (Out\ s\ x :: outs))) \Rightarrow \\
& \forall a_0\ a_1\ a_2. TR\ a_0\ a_1\ a_2 \Rightarrow TR'\ a_0\ a_1\ a_2
\end{aligned}$$

[TR_Trans_lemma]

$$\begin{aligned}
& \vdash TR\ x\ (CFG\ (x :: ins)\ s\ outs) \\
& (CFG\ ins\ (NS\ s\ x)\ (Out\ s\ x :: outs)) \Rightarrow \\
& Trans\ x\ s\ (NS\ s\ x)
\end{aligned}$$

[Trans_cases]

$$\vdash \forall a_0 \ a_1 \ a_2. \text{Trans } a_0 \ a_1 \ a_2 \iff \exists NS. \ a_2 = NS \ a_1 \ a_0$$

[Trans_Equiv_TR]

$$\vdash \text{TR } x \ (\text{CFG } (x::\text{ins}) \ s \ \text{outs}) \\ (\text{CFG } \text{ins} \ (NS \ s \ x) \ (\text{Out } s \ x::\text{outs})) \iff \text{Trans } x \ s \ (NS \ s \ x)$$

[Trans_ind]

$$\vdash \forall \text{Trans}'. \\ (\forall NS \ s \ x. \ \text{Trans}' \ x \ s \ (NS \ s \ x)) \Rightarrow \\ \forall a_0 \ a_1 \ a_2. \ \text{Trans } a_0 \ a_1 \ a_2 \Rightarrow \text{Trans}' \ a_0 \ a_1 \ a_2$$

[Trans_rules]

$$\vdash \forall NS \ s \ x. \ \text{Trans } x \ s \ (NS \ s \ x)$$

[Trans_strongind]

$$\vdash \forall \text{Trans}'. \\ (\forall NS \ s \ x. \ \text{Trans}' \ x \ s \ (NS \ s \ x)) \Rightarrow \\ \forall a_0 \ a_1 \ a_2. \ \text{Trans } a_0 \ a_1 \ a_2 \Rightarrow \text{Trans}' \ a_0 \ a_1 \ a_2$$

[Trans_TR_lemma]

$$\vdash \text{Trans } x \ s \ (NS \ s \ x) \Rightarrow \\ \text{TR } x \ (\text{CFG } (x::\text{ins}) \ s \ \text{outs}) \ (\text{CFG } \text{ins} \ (NS \ s \ x) \ (\text{Out } s \ x::\text{outs}))$$

Index

m1 Theory, 3

- Datatypes, 3
- Theorems, 3
 - command_distinct_clauses, 3
 - m1_rules, 3
 - M1ns_def, 3
 - M1ns_ind, 3
 - M1out_def, 4
 - M1out_ind, 4
 - m1TR_clauses, 4
 - m1TR_rules, 4
 - m1Trans_Equiv_TR, 4
 - output_distinct_clauses, 4
 - state_distinct_clauses, 4

sm Theory, 4

- Datatypes, 5
- Definitions, 5
 - TR_def, 5
 - Trans_def, 5
- Theorems, 5
 - configuration_one_one, 5
 - TR_cases, 5
 - TR_clauses, 5
 - TR_complete, 6
 - TR_deterministic, 6
 - TR_ind, 6
 - TR_rules, 6
 - TR_strongind, 6
 - TR_Trans_lemma, 6
 - Trans_cases, 7
 - Trans_Equiv_TR, 7
 - Trans_ind, 7
 - Trans_rules, 7
 - Trans_strongind, 7
 - Trans_TR_lemma, 7