Models of Computation Coursework 2

Honglei Gu, Zhizhong Ren, Patrick Ma, Mohammad Abbas

December 2, 2024

1 Question 1

MoC Coursework (due Dec 2nd)

Q1

a.

$$\text{(L-HALT)}\ \frac{}{C \rightsquigarrow C[0 \mapsto 0]}\ \begin{array}{l} C(0)=k, \\ C(k)=0 \end{array}$$

$$\text{(L-ADD)}\ \frac{}{C \rightsquigarrow C[0 \mapsto L'][R \mapsto C(R)+1]}\ \begin{array}{l} C(0)=k,\ C(k)=1, \\ C(k+1)=R,\ C(k+2)=L' \end{array}$$

$$\text{(L-SUB-FAIL)}\ \frac{}{C \rightsquigarrow C[0 \mapsto L'']}\ \begin{array}{l} C(0)=k,\ C(k)=2,\ C(k+1)=R, \\ C(R)=0,\ C(k+3)=L'' \end{array}$$

$$\text{(L-SUB-SUCCEED)}\ \frac{}{C \rightsquigarrow C[0 \mapsto L'][R \mapsto C(R)-1]}\ \begin{array}{l} C(0)=k,\ C(k)=2, \\ C(k+1)=R,\ C(R)>0, \\ C(k+2)=L' \end{array}$$

b.

1. Answer configuration C satisfies $C(0)=0$.

2. Stuck configuration C satisfy:

$$C(0)\uparrow \vee C(C(0))\uparrow \vee C(C(0)) > 2$$
$$\vee [C(C(0)) > 0 \wedge C(C(0)+1)\uparrow]$$
$$\vee [C(C(0)) = 2 \wedge C(C(0)+1) > 0 \wedge C(C(0)+2)\uparrow]$$
$$\vee [C(C(0)) = 2 \wedge C(C(0)+1) = 0 \wedge C(C(0)+3)\uparrow]$$

c,

C_{st} terminates $\forall x, y \in \mathbb{N}$

d,

Let $N_P = [0, 18] \setminus \{16, 17\}$, $N_Q = [0, 18] \setminus \{16, 17, 18\}$

Then $P(C_{st}, C) \triangleq C(16) + C(17) = C_{st}(17) \wedge C_{st} \sim_{N_P} C$

$$Q(C_{st}, C) \triangleq C(16) + C(18) = C_{st}(17) + C_{st}(18)$$
$$\wedge\, C(17) = 0 \wedge C_{st} \sim_{N_Q} C$$

e,

P and Q are analogous to loop invariants.

Executions from C_{st} to C_1, and from C_2 to C_3 are analogous to one execution of ~~their respective~~ associated loop.

f,

Take arbitrary $C, C' \in Cfg$

Assume

① $C(16) + C(18) = C_{st}(17) + C_{st}(18)$ [from $Q(C_{st}, C)$]

② $C(17) = 0$ [from $Q(C_{st}, C)$]

③ $C_{st} \sim_{N_Q} C$ [from $Q(C_{st}, C)$]

④ $C(0) = 8$

⑤ $C \rightsquigarrow C'$

⑥ $C'(0) = 15$

To show: $C'(16) = C_{st}(17) + C_{st}(18)$

⑦ C' can be obtained from C with an application of one of the rules we defined (by def of $\rightsquigarrow$)

⑧ $C(8) = C_{st}(8) = 2$ (by ③)

⑨ $C(8+2) = C(10) = C_{st}(10) = 12$ (by ③ and arithmetic)

⑩ $C(8+3) = C(11) = C_{st}(11) = 15$ (by ③ and arithmetic)

⑪ either (L-SUB-FAILS or (L-SUB-SUCCEED) is applicable to C (by ④ and ⑧)

⑫ C' can only be obtained from (L-SUB-FAIL)
(by ⑪ and by the fact that we have ⑨ and ⑩, so only (L-SUB-FAIL) produce ⑥)

⑬ $C(8+1) = C(9) = C_{st}(9) = 18$ (by ③ and arithmetic)

⑭ $C(18) = C(C(8+1)) = 0$ (by ⑫ (side-condition of L-SUB-FAIL))

⑮ $C(16) = C_{st}(17) + C_{st}(18)$ (by ⑭, ① and arithmetic)

2 Question 2

Consider a small language of expressions, which is very similar to the languages we discussed below, and whose syntax is defined below:

$$E \in Exp ::= n \mid b \mid E \oplus E$$

where $n \in \mathbb{N}$, $b \in \mathbb{B}$ and $\mathbb{B} = \{\mathrm{ff}, \mathrm{tt}\}$.

2.1 Part A

2.1.1 a

Answer: it terminates to tt

$$\begin{aligned}
(\mathrm{tt} \oplus 3) \oplus 0 &= \mathrm{tt} \oplus 3 && \text{(by E.4)} \\
&= \mathrm{tt} && \text{(by E.4)}
\end{aligned}$$

2.1.2 b

Answer:it terminates to 3

$$\begin{aligned}
(\mathrm{ff} \oplus 3) \oplus 0 &= 3 \oplus 0 && \text{(by E.3)} \\
&= 3 && \text{(by E.6)}
\end{aligned}$$

2.1.3 c

Answer:it terminates to 8

$$\begin{aligned}
(3 \oplus 4) \oplus 1 &= (4 \oplus 3) \oplus 1 && \text{(by E.7)} \\
&= (5 \oplus 2) \oplus 1 && \text{(by E.7)} \\
&= (6 \oplus 1) \oplus 1 && \text{(by E.7)} \\
&= (7 \oplus 0) \oplus 1 && \text{(by E.7)} \\
&= 7 \oplus 1 && \text{(by E.6)} \\
&= 8 \oplus 0 && \text{(by E.7)} \\
&= 8 && \text{(by E.6)}
\end{aligned}$$

2.1.4 d

Answer:it does not terminate since we cannot find a rule to evaluate $4 \oplus \mathrm{ff}$

2.2 Part B

2.2.1 a

Answer: it does not have a type

$$
\begin{aligned}
\mathrm{tt} :: \mathrm{Bool} && \text{by T.2} \\
3 :: \mathrm{Int} && \text{by T.1}
\end{aligned}
$$

at this point we have no rule to evaluate further

2.2.2 b

Answer: it does not have a type

$$
\begin{aligned}
\mathrm{ff} :: \mathrm{Bool} && \text{by T.2} \\
3 :: \mathrm{Int} && \text{by T.1}
\end{aligned}
$$

at this point we have no rule to evaluate further

2.2.3 c

Answer:it has type Int

$$
\begin{aligned}
3 :: \mathrm{Int} && \text{by T.1} \\
4 :: \mathrm{Int} && \text{by T.1} \\
3 \oplus 4 :: \mathrm{Int} && \text{by T.3} \\
1 :: \mathrm{Int} && \text{by T.1} \\
(3 \oplus 4) \oplus 1 :: \mathrm{Int} && \text{by T.3}
\end{aligned}
$$

2.2.4 d

Answer:it does not have a type

$$
\begin{aligned}
4 :: \mathrm{Int} && \text{by T.1} \\
\mathrm{ff} :: \mathrm{Bool} && \text{by T.2}
\end{aligned}
$$

at this point we have no rule to evaluate further

2.3 Part C

Answer:

$$\begin{aligned}\mathbf{S}(\mathrm{ff} \oplus (3 \oplus 4)) &= \mathbf{S}(\mathrm{ff}) \cup \mathbf{S}(3 \oplus 4) \cup \{\mathrm{ff} \oplus (3 \oplus 4)\} \\ &= \{\mathrm{ff}\} \cup (\{3\} \cup \{4\} \cup \{3 \oplus 4\}) \cup \{\mathrm{ff} \oplus (3 \oplus 4)\} \\ &= \{\mathrm{ff}, 3, 4, 3 \oplus 4, \mathrm{ff} \oplus (3 \oplus 4)\}\end{aligned}$$

2.4 Part D

Proof: By induction on $E \in Exp$. By the definition of Exp, the induction principle of (**A8**) is then (**IP**):

$$\begin{aligned}&\forall n \in \mathbb{N}.\forall E' \in Exp.\forall T \in Type.[n :: T \wedge E' \in \mathbf{S}(n) \Rightarrow E' :: T] \\ &\wedge \forall b \in \mathbb{B}.\forall E' \in Exp.\forall T \in Type.[b :: T \wedge E' \in \mathbf{S}(b) \Rightarrow E' :: T] \\ &\wedge \forall E_1, E_2 \in Exp.[\\ &\quad (\forall E' \in Exp.\forall T \in Type.[E_1 :: T \wedge E' \in \mathbf{S}(E_1) \Rightarrow E' :: T]) \\ &\quad \wedge (\forall E' \in Exp.\forall T \in Type.[E_2 :: T \wedge E' \in \mathbf{S}(E_2) \Rightarrow E' :: T]) \\ &\quad \Rightarrow (\forall E' \in Exp.\forall T \in Type.[E_1 \oplus E_2 :: T \wedge E' \in \mathbf{S}(E_1 \oplus E_2) \Rightarrow E' :: T]) \\ &\quad] \\ &\Rightarrow \forall E \in Exp.\forall E' \in Exp.\forall T \in Type.[E :: T \wedge E' \in \mathbf{S}(E) \Rightarrow E' :: T]\end{aligned} \tag{\textbf{IP}}$$

2.4.1 Proof, part A, Base Case 1

To prove (**BC1**):

$$\forall n \in \mathbb{N}.\forall E' \in Exp.\forall T \in Type.[n :: T \wedge E' \in \mathbf{S}(n) \Rightarrow E' :: T] \tag{\textbf{BC1}}$$

Take $n' \in \mathbb{N}$, $e' \in Exp$, $t \in Type$ arbitrarily and assume

$$n' :: t \tag{\textbf{asm1}}$$

$$e' \in \mathbf{S}(n') \tag{\textbf{asm2}}$$

To show

$$e' :: t \tag{α}$$

We have:

$$\begin{aligned} n' &:: Int && \text{by T.1} && (1) \\ t &= Int && \text{by (1)} && (2) \\ \mathbf{S}(n') &= \{n'\} && \text{by def. } \mathbf{S} \text{ and (1)} && (3) \\ e' &= n' && \text{by } (\textbf{asm2}) \text{ and (3)} && (4) \\ e' &:: Int && \text{by (1) and (4)} && (5) \end{aligned}$$

So (α) follows directly. Then by (showing implication using (**asm1**), (**asm2**), (α)), and (showing 'for all') as $n' \in \mathbb{N}$, $e' \in Exp$, $t \in Type$ arbitrarily, (**BC1**) holds.

2.4.2 Proof, part B, Base Case 1

To prove (**BC2**):

$$\forall b \in \mathbb{B}.\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[b :: T \land E' \in \mathbf{S}(b) \Rightarrow E' :: T] \qquad (\mathbf{BC2})$$

The proof follows the exact same structure as that (**BC1**) with the following textual substitutions:

- $\mathbb{N}$ with $\mathbb{B}$
- n' with b'
- T.1 with T.2
- Int with Bool

2.4.3 Proof, part C, Inductive Step

To prove (**IC**):

$$\begin{aligned}
&\forall E_1, E_2 \in \mathit{Exp}.[\\
&\quad (\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[E_1 :: T \land E' \in \mathbf{S}(E_1) \Rightarrow E' :: T])\\
&\land (\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[E_2 :: T \land E' \in \mathbf{S}(E_2) \Rightarrow E' :: T])\\
&\Rightarrow (\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[E_1 \oplus E_2 :: T \land E' \in \mathbf{S}(E_1 \oplus E_2) \Rightarrow E' :: T])\\
&\quad]
\end{aligned} \qquad (\mathbf{IC})$$

Take $e_1, e_2 \in \mathit{Exp}$ arbitrarily, and assume

$$\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[e_1 :: T \land E' \in \mathbf{S}(e_1) \Rightarrow E' :: T] \qquad (\mathbf{IH1})$$
$$\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[e_2 :: T \land E' \in \mathbf{S}(e_2) \Rightarrow E' :: T] \qquad (\mathbf{IH2})$$

To show

$$\forall E' \in \mathit{Exp}.\forall T \in \mathit{Type}.[e_1 \oplus e_2 :: T \land E' \in \mathbf{S}(e_1 \oplus e_2) \Rightarrow e_1 \oplus e_2 :: T] \qquad (\mathbf{IS})$$

Take $e' \in \mathit{Exp}$, $t \in \mathit{Type}$ arbitrarily, and assume

$$e_1 \oplus e_2 :: t \qquad (\mathbf{asm1})$$
$$e' \in \mathbf{S}(e_1 \oplus e_2) \qquad (\mathbf{asm2})$$

To show

$$e' :: t \qquad (\alpha)$$

Proof:

$e_1 :: t$	by (**asm1**) and T.3	(1)
$e_2 :: t$	by (**asm2**) and T.3	(2)
$e_1 :: t \land e' \in \mathbf{S}(e_1) \Rightarrow e' :: t$	by (**IH1**)	(3)
$e_2 :: t \land e' \in \mathbf{S}(e_2) \Rightarrow e' :: t$	by (**IH2**)	(4)
$\mathbf{S}(e_1 \oplus e_2) = \mathbf{S}(e_1) \cup \mathbf{S}(e_2) \cup \{e_1 \oplus e_2\}$	by def. $\mathbf{S}$	(5)
$\forall E \in \mathbf{S}(e_1 \oplus e_2).[E \in \mathbf{S}(e_1) \lor E \in \mathbf{S}(e_2) \lor E = e_1 \oplus e_2]$	by (5) and def. $(\cup)$	(6)
$e' \in \mathbf{S}(e_1) \lor e' \in \mathbf{S}(e_2) \lor e' = e_1 \oplus e_2$	by (6) and (**asm1**)	(7)

By cases on (7):

Case I

$$
\begin{aligned}
& e' \in \mathbf{S}(e_1) && \text{(assumption)} && (8)\\
& e_1 :: t \land e' \in \mathbf{S}(e_2) && \text{by (2) and (8)} && (9)\\
& e' :: t && \text{by (9) and (3)} && (10)
\end{aligned}
$$

Case II

$$
\begin{aligned}
& e' \in \mathbf{S}(e_2) && \text{(assumption)} && (11)\\
& e_2 :: t \land e' \in \mathbf{S}(e_2) && \text{by (2) and (11)} && (12)\\
& e' :: t && \text{by (12) and (4)} && (13)
\end{aligned}
$$

Case III

$$
\begin{aligned}
& e' = e_1 \oplus e_2 && \text{(assumption)} && (14)\\
& e_1 :: t \land e_2 :: t && \text{by (1), (2) and (14)} && (15)\\
& e' :: t && \text{by (15) and T.3} && (16)
\end{aligned}
$$

So by (10), (13) and (16) and (using or on (7)),

$$e' :: t \tag{17}$$

and (α) follows directly. Then by (showing implication using (**asm1**), (**asm2**) and (α)) and that $e' \in \mathit{Exp}$, $t \in \mathit{Type}$ arbitrarily, (**IS**) holds. By (**IH1**), (**IH2**) and (showing implication), (**IC**) follows directly.

3 Part E

Proof. By induction on the definition of expression evaluation ($\Downarrow$). The induction principle for an arbitrary predicate $P \subseteq \mathit{Exp} \times \mathbb{V}$ is then (**IP**):

$$
\begin{aligned}
&\ \ \forall n, n' \in \mathbb{N}.[P(\mathtt{n}, n)]\\
&\land \forall b, b' \in \mathbb{B}.[P(\mathtt{b}, b)]\\
&\land \forall E_1, E_2 \in \mathit{Exp}.\forall v \in \mathbb{V}.[\ P(E_1, \mathrm{ff}) \land P(E_2, v) \Rightarrow P(E_1 \oplus E_2, v)\ ]\\
&\land \forall E_1, E_2 \in \mathit{Exp}.[\ P(E_1, \mathrm{tt}) \Rightarrow P(E_1 \oplus E_2, \mathrm{tt})\ ]\\
&\land \forall E_1, E_2 \in \mathit{Exp}.\forall v_1, v_2, v \in \mathbb{V}.[\\
&\quad P(E_1, v_1) \land P(E_2, v_2) \land P(v_1 \oplus v_2, v) \Rightarrow P(E_1 \oplus E_2, v)\\
&\ \ ]\\
&\land \forall v \in \mathbb{V}.[P(v \oplus 0, v)]\\
&\land \forall n_1, n_2 \in \mathbb{N}.\forall v \in \mathbb{V}.[P((n_1 + 1) \oplus n_2, v) \Rightarrow P(n_1 \oplus (n_2 + 1), v)]\\
&\Rightarrow \forall E \in \mathit{Exp}.\forall v \in \mathbb{V}.[E \Downarrow v \Rightarrow P(E, v)]
\end{aligned}
\tag{\textbf{IP}}
$$

Define $P \subseteq \mathit{Exp} \times \mathbb{V}$ as $P(E, v) \triangleq (E :: \mathrm{Int} \Rightarrow v \in \mathbb{N})$.

3.1 Part I: Last applied rule is E.1

To show a base case, (**C-E.1**):

$$\forall n \in \mathbb{N}.[\mathtt{n} :: \text{Int} \Rightarrow n \in \mathbb{N}] \tag{\textbf{C-E.1}}$$

Take $n' \in \mathbb{N}$ arbitrarily, and assume $\mathtt{n}' :: \text{Int}$. Then by T.1 being the only possible rule, $n' \in \mathbb{N}$ (since $\mathtt{n}'$ is the numerical representation of n'). So the implication holds for arbitrary $n' \in \mathbb{N}$, and (**C-E.1**) follows directly.

3.2 Part II: Last applied rule is E.2

To show a base case, (**C-E.2**):

$$\forall b \in \mathbb{B}.[\mathtt{b} :: \text{Int} \Rightarrow b \in \mathbb{N}] \tag{\textbf{C-E.2}}$$

Take $b' \in \mathbb{B}$ arbitrarily, and assume $\mathtt{b}' :: \text{Int}$. But we have already $\mathtt{b}' :: \text{Bool}$ by T.2. So $\text{Int} = \text{Bool}$, which is a contradiction under the definition of *Type* as containing two disjoint types. As such,

$$\mathtt{b}' :: \text{Int} \Rightarrow b' \in \mathbb{N}$$

holds vacuously for arbitrary $b' \in \mathbb{B}$, and (**C-E.2**) follows directly.

3.3 Part III: Last applied rule is E.5

To show an inductive case, (**C-E.5**):

$$\begin{aligned} &\forall E_1, E_2 \in \mathit{Exp}.\forall v_1, v_2, v \in \mathbb{V}.[\\ &\quad (E_1 :: \text{Int} \Rightarrow v_1 \in \mathbb{N})\\ &\quad \wedge (E_2 :: \text{Int} \Rightarrow v_2 \in \mathbb{N})\\ &\quad \wedge (v_1 \oplus v_2 :: \text{Int} \Rightarrow v_1 \oplus v_2 \in \mathbb{N})\\ &\quad \Rightarrow (E_1 \oplus E_2 :: \text{Int} \Rightarrow v \in \mathbb{N})] \end{aligned} \tag{\textbf{C-E.5}}$$

Take $e_1, e_2 \in \mathit{Exp}$, $v_1', v_2', v' \in \mathbb{V}$ arbitrarily, and assume induction hypotheses

$$e_1 :: \text{Int} \Rightarrow v_1' \in \mathbb{N} \tag{\textbf{IH1}}$$
$$e_2 :: \text{Int} \Rightarrow v_2' \in \mathbb{N} \tag{\textbf{IH2}}$$
$$v_1 \oplus v_2 :: \text{Int} \Rightarrow v' \in \mathbb{N} \tag{\textbf{IH3}}$$

To show (**IS**):

$$e_1 \oplus e_2 :: \text{Int} \Rightarrow v' \in \mathbb{N}$$

Assume $e_1 \oplus e_2 :: \text{Int}$.
Then by T.3 being the only possible rule for $(\oplus)$ and T.3's definition, $e_1 :: \text{Int}$ and $e_2 :: \text{Int}$.
Then by (**IH1**) and (**IH2**), $v_1' \in \mathbb{N}$ and $v_2' \in \mathbb{N}$.
Then by T.1, $v_1' :: \text{Int}$ and $v_2' :: \text{Int}$.
Then by T.3, $v_1' \oplus v_2' :: \text{Int}$.
Then by (**IH3**), $v' \in \mathbb{N}$, so by (showing implication), (**IS**) follows directly.
So since $e_1, e_2 \in \mathit{Exp}$, $v_1', v_2', v' \in \mathbb{V}$ arbitrarily, (**C-E.5**) holds.

3.4 Part IV: Last applied rule is E.7

To show an inductive case, (**C-E.7**):

$$\forall n_1, n_2 \in \mathbb{N}.\forall v \in \mathbb{V}.[((n_1{+}1)\oplus n_2 :: \text{Int} \Rightarrow v \in \mathbb{N}) \Rightarrow (n_1\oplus(n_2{+}1) :: \text{Int} \Rightarrow v \in \mathbb{N})] \quad (\textbf{C-E.7})$$

Take $n_1', n_2' \in \mathbb{N}, v' \in \mathbb{V}$ arbitrarily, and assume the induction hypothesis

$$(n_1' + 1) \oplus n_2' :: \text{Int} \Rightarrow v' \in \mathbb{N} \quad (\textbf{IH})$$

To show

$$n_1' \oplus (n_2' + 1) :: \text{Int} \Rightarrow v' \in \mathbb{N} \quad (\textbf{IS})$$

Assume $n_1' \oplus (n_2' + 1) :: \text{Int}$.
We have $n_1' + 1, n_2' \in \mathbb{N}$, so by T.1, $n_1' + 1, n_2' :: \text{Int}$.
Then by T.3, $(n_1' + 1) \oplus n_2' :: \text{Int}$.
Then by (**IH**), $v' \in \mathbb{N}$, and by (showing implication), (**IS**) follows directly.
By (showing implication from (**IH**) to (**IS**)) and (showing for-all) since $n_1', n_2' \in \mathbb{N}, v' \in \mathbb{V}$ arbitrarily, (**C-E.7**) follows directly.

4 Question 3

4.1 a

Zhizhong Ren an Patrick Ma worked on Question 1, and Honglei Gu and Mohammad Abbas worked on Question 2

4.2 b

Question 2 part e requires more consideration on what should we apply induction to.

4.3 c

about 10 hours

4.4 d

it helps clarify induction, Register machines, and also to consider different approaches of induction.