Alone Together: Compositional Reasoning and Inference for Weak Isolation

ANONYMOUS AUTHOR(S)

ACM Reference format:
Anonymous Author(s). 2017. Alone Together: Compositional Reasoning and Inference for Weak Isolation. *PACM Progr. Lang.* 1, 1, Article 1 (January 2017), 13 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 OPERATIONAL SEMANTICS

Syntax

$x, y \in \texttt{Variables} \quad f \in \texttt{Field Names} \quad i, j \in \mathbb{N} \quad \odot \in \{+, -, \leq, \geq, =\} \quad k \in \mathbb{Z} \cup \mathbb{B} \quad r \in \{\bar{f} = \bar{k}\}$

$$
\begin{array}{llll}
\delta, \Delta, s & \in & \texttt{State} & := \mathcal{P}\left(\{\bar{f} = \bar{k}\}\right) \\
\mathbb{I}_e, \mathbb{I}_c & \in & \texttt{IsolationSpec} & := (\delta, \Delta, \Delta') \to \mathbb{P} \\
v & \in & \texttt{Values} & := k \mid r \mid s \\
e & \in & \texttt{Expressions} & := k \mid x \mid x.f \mid \{\bar{f} = \bar{e}\} \mid e_1 \odot e_2 \\
c & \in & \texttt{Commands} & := \mathsf{SKIP} \mid \mathsf{LET}\ x = e\ \mathsf{IN}\ c \mid \mathsf{IF}\ e\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2 \mid c_1; c_2 \mid \mathsf{INSERT}\ x \\
 & & & \mid \mathsf{DELETE}\ \lambda x.e \mid \mathsf{LET}\ x = \mathsf{SELECT}\ \lambda x.e\ \mathsf{IN}\ c \mid \mathsf{UPDATE}\ \lambda x.e_1\ \lambda x.e_2 \\
 & & & \mid \mathsf{FOREACH}\ x\ \mathsf{DO}\ \lambda y.\lambda z.c \mid \mathsf{foreach}\langle s_1 \rangle\ s_2\ \mathsf{do}\ \lambda x.\lambda y.e \\
 & & & \mid \mathsf{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c \rangle\{c\} \mid \mathsf{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \delta, \Delta \rangle\{c\} \mid c1 || c2 \\
\mathcal{E} & \in & \texttt{Eval Ctx} & ::= \bullet \mid \bullet || c_2 \mid c_1 || \bullet \mid \bullet; c_2 \mid \mathsf{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \delta, \Delta \rangle\{\bullet\}
\end{array}
$$

Local Reduction $\boxed{\Delta \vdash (c, \delta) \longrightarrow (c', \delta')}$

E-INSERT

$$\frac{i \notin \mathsf{dom}(\delta \cup \Delta) \qquad r = \{\bar{f} = \bar{k}; \mathtt{id} = i; \mathtt{del} = \mathtt{false}\}}{\Delta \vdash (\mathsf{INSERT}\ \{\bar{f} = \bar{k}\}, \delta) \longrightarrow (\mathsf{SKIP}, \delta \cup \{r\})}$$

E-DELETE

$$\frac{s = \{r' \mid \exists (r \in \Delta).\ \mathsf{eval}([r/x]e) = \mathtt{true} \wedge r' = \{\bar{f} = r.\bar{f}; \mathtt{id} = r.\mathtt{id}; \mathtt{del} = \mathtt{true}\}\}}{\Delta \vdash (\mathsf{DELETE}\ \lambda x.e, \delta) \longrightarrow (\mathsf{SKIP}, \delta \cup s)}$$

E-SELECT

$$\frac{s = \{r \in \Delta \mid \mathsf{eval}([r/x]e) = \mathtt{true}\} \quad c' = [s/y]c}{\Delta \vdash (\mathsf{LET}\ y = \mathsf{SELECT}\ \lambda x.e\ \mathsf{IN}\ c, \delta) \longrightarrow (c', \delta)}$$

E-UPDATE

$$\frac{s = \{r' \mid \exists (r \in \Delta).\ \mathsf{eval}([r/x]e_2) = \mathtt{true} \ \wedge\ r' = [r/x]e_1\}}{\Delta \vdash (\mathsf{UPDATE}\ \lambda x.e_1\ \lambda x.e_2, \delta) \longrightarrow (\mathsf{SKIP}, \delta \cup s)}$$

E-SEQ1

$$\frac{\Delta \vdash (c1, \delta) \longrightarrow (c1', \delta') \quad c1 \neq \mathsf{SKIP}}{\Delta \vdash (c1; c2, \delta) \longrightarrow (c1'; c2, \delta')}$$

E-SEQ2

$$\frac{\Delta \vdash (c1, \delta) \longrightarrow (\mathsf{SKIP}, \delta')}{\Delta \vdash (c1; c2, \delta) \longrightarrow (c2, \delta')}$$

E-IFTRUE

$$\frac{\mathsf{eval}(e) = \mathtt{true}}{\Delta \vdash (\mathsf{IF}\ e\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2, \delta) \longrightarrow (c1, \delta)}$$

E-IFFALSE

$$\frac{\mathsf{eval}(e) = \mathtt{false}}{\Delta \vdash (\mathsf{IF}\ e\ \mathsf{THEN}\ c_1\ \mathsf{ELSE}\ c_2, \delta) \longrightarrow (c2, \delta)}$$

2017. 2475-1421/2017/1-ART1 $15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

E-Foreach1 $\Delta \vdash (\texttt{FOREACH}\ s\ \texttt{DO}\ \lambda y.\lambda z.c, \delta) \longrightarrow (\text{foreach}\langle\emptyset\rangle\ s\ \text{do}\ \lambda y.\lambda z.c)$

E-Foreach2 $\Delta \vdash (\text{foreach}\langle s_1\rangle\ \{r\} \uplus s_2\ \text{do}\ \lambda y.\lambda z.c, \delta) \longrightarrow ([r/z][s_1/y]c;\ \text{foreach}\langle s_1 \cup \{r\}\rangle\ s_2\ \text{do}\ \lambda y.\lambda z.c)$

E-Foreach3 $\Delta \vdash (\text{foreach}\langle s\rangle\ \emptyset\ \text{do}\ \lambda y.\lambda z.c, \delta) \longrightarrow (\texttt{SKIP}, \delta)$

Top-Level Reduction $\boxed{(c, \Delta) \longrightarrow (c', \Delta')}$

E-Txn-Start

$$\frac{}{(\text{txn}i\mathbb{I}_e, \mathbb{I}_c c, \Delta) \longrightarrow (\texttt{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \emptyset, \Delta\rangle\{c\}, \Delta)}$$

E-Txn

$$\frac{\mathbb{I}_e\ (\delta, \Delta, \Delta') \quad \Delta \vdash (c, \delta) \longrightarrow (c', \delta')}{(\texttt{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \delta, \Delta\rangle\{c\}, \Delta') \longrightarrow (\texttt{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \delta', \Delta'\rangle\{c'\}, \Delta')}$$

E-Commit

$$\frac{\mathbb{I}_c\ (\delta, \Delta, \Delta')}{(\texttt{TXN}_i\langle \mathbb{I}_e, \mathbb{I}_c, \delta, \Delta\rangle\{\texttt{SKIP}\}, \Delta') \longrightarrow (\texttt{SKIP}, \delta \gg \Delta')}$$

2 RELY-GUARANTEE REASONING

Txn-Local Reasoning $\boxed{R \vdash \{P\}\,[c]_i\,\{Q\}}$

RG-Insert

$$\frac{\begin{array}{c}\texttt{stable}(R, P) \\ \forall \delta, \delta', \Delta, i.\ P(\delta, \Delta)\ \wedge\ i \notin \text{dom}(\delta \cup \Delta) \\ \wedge\ \delta' = \delta \cup \{\{\bar{f} = x.\bar{f};\ \texttt{id} = i;\ \texttt{del} = \texttt{false}\} \Rightarrow Q(\delta', \Delta)\end{array}}{R \vdash \{P\}\,[\texttt{INSERT}\ x]_i\,\{Q\}}$$

RG-Delete

$$\frac{\begin{array}{c}\texttt{stable}(R, P) \\ \forall \delta, \delta', \Delta.\ P(\delta, \Delta)\ \wedge\ \delta' = \delta \cup \{r' \mid \exists (r \in \Delta).\ [r/x]e = \texttt{true} \\ \wedge\ r' = \{\bar{f} = r.\bar{f};\ \texttt{id} = r.\texttt{id};\ \texttt{del} = \texttt{true}\}\} \Rightarrow Q(\delta', \Delta)\end{array}}{R \vdash \{P\}\,[\texttt{DELETE}\ \lambda x.e]_i\,\{Q\}}$$

RG-Update

$$\frac{\begin{array}{c}\texttt{stable}(R, P) \\ \forall \delta, \delta', \Delta.\ P(\delta, \Delta)\ \wedge\ \delta' = \delta \cup \{r' \mid \exists (r \in \Delta).\ [r/x]e_2 = \texttt{true} \\ \wedge\ r' = [r/x]e_1\} \Rightarrow Q(\delta', \Delta)\end{array}}{R \vdash \{P\}\,[\texttt{UPDATE}\ \lambda x.e_1\ \lambda x.e_2]_i\,\{Q\}}$$

RG-Select

$$\frac{\begin{array}{c}R \vdash \{P'\}\,[c]_i\,\{Q\} \quad \texttt{stable}(R, P) \\ P'(\delta, \Delta) \Leftrightarrow P(\delta, \Delta) \wedge x = \{r' \mid \exists (r \in \Delta).\ [r/x]e_2 = \texttt{true}\}\end{array}}{R \vdash \{P\}\,[\texttt{LET}\ y = \texttt{SELECT}\ \lambda x.e\ \texttt{IN}\ c]_i\,\{Q\}}$$

RG-Foreach

$$\frac{\begin{array}{c}\texttt{stable}(\mathbb{R}, Q) \quad \texttt{stable}(\mathbb{R}, \psi) \\ P \Rightarrow [y/\phi]\psi \quad \mathbb{R} \vdash \{\psi \wedge z \in x\}\,[c]_i\,\{Q_c\} \\ Q_c \Rightarrow [y \cup \{z\}/y]\psi \quad [x/y]\psi \Rightarrow Q\end{array}}{\mathbb{R} \vdash \{P\}\,[\texttt{FOREACH}\ x\ \texttt{DO}\ \lambda y.\lambda z.c]_i\,\{Q\}}$$

RG-Seq

$$\frac{\begin{array}{c}\{P\}\,[c1]_i\,\{Q^{'}\} \quad \{Q^{'}\}\,[c2]_i\,\{Q\} \\ \texttt{stable}(R, Q^{'})\end{array}}{R \vdash \{P\}\,[c1; c2]_i\,\{Q\}}$$

RG-If

$$\frac{\begin{array}{c}\{P \wedge e\}\,[c1]_i\,\{Q\} \quad \{P \wedge \neg e\}\,[c2]_i\,\{Q\} \\ \texttt{stable}(R, P)\end{array}}{R \vdash \{P\}\,[\texttt{IF}\ e\ \texttt{THEN}\ c_1\ \texttt{ELSE}\ c_2]_i\,\{Q\}}$$

RG-Conseq

$$\frac{\begin{array}{c}R \vdash \{P\}\,[c]_i\,\{Q\} \\ P' \Rightarrow P \quad Q \Rightarrow Q' \quad \texttt{stable}(R, P') \quad \texttt{stable}(R, Q')\end{array}}{R \vdash \{P'\}\,[c]_i\,\{Q'\}}$$

Top-Level Reasoning $\boxed{\{I,R\}\, c\, \{G,I\}}$

RG-Txn

$$\frac{\begin{array}{c}\texttt{stable}(R,I) \quad \texttt{stable}(R,\mathbb{I}) \quad P(\delta,\Delta) \Leftrightarrow \delta = \emptyset \wedge I(\Delta) \\ \mathbb{R}_e(\delta,\Delta,\Delta') \Leftrightarrow \exists \Delta_1.R(\Delta,\Delta') \wedge \mathbb{I}_e(\delta,\Delta_1,\Delta') \quad \mathbb{R}_e \vdash \{P\}\, c\, \{Q\} \\ \mathbb{R}_c(\delta,\Delta,\Delta') \Leftrightarrow \exists \Delta_1.R(\Delta,\Delta') \wedge \mathbb{I}_c(\delta,\Delta_1,\Delta') \quad \texttt{stable}(\mathbb{R}_c,Q) \\ \forall \delta,\Delta.\ Q(\delta,\Delta) \Rightarrow G(\Delta,\delta \gg \Delta) \quad \forall \Delta,\Delta'.\ I(\Delta) \wedge G(\Delta,\Delta') \Rightarrow I(\Delta')\end{array}}{\{I,R\}\, \texttt{TXN}_i\langle \mathbb{I}\rangle\{c\}\, \{G \cup ID, I\}}$$

RG-Par

$$\frac{\begin{array}{c}\{I, R \cup G_2 \cup ID\}\, t_1\, \{G_1 \cup ID, I\} \\ \{I, R \cup G_1 \cup ID\}\, t_2\, \{G_2 \cup ID, I\}\end{array}}{\{I,R\}\, t_1 || t_2\, \{G_1 \cup G_2 \cup ID, I\}}$$

RG-Conseq2

$$\frac{\begin{array}{c}\{I,R\}\, \texttt{TXN}_i\langle \mathbb{I}\rangle\{c\}\, \{G,I\} \\ \mathbb{I}' \Rightarrow \mathbb{I} \quad R' \subseteq R \quad \texttt{stable}(R',\mathbb{I}') \\ G \subseteq G' \quad \forall \Delta,\Delta'.\ I(\Delta) \wedge G'(\Delta,\Delta') \Rightarrow I(\Delta')\end{array}}{\{I,R'\}\, \texttt{TXN}_i\langle \mathbb{I}'\rangle\{c\}\, \{G',I\}}$$

3 SOUNDNESS OF RG-REASONING

Definition 3.1 (Step-indexed reflexive transitive closure). For all A : Type, $R : A \to A \to \mathbb{P}$, and $n : \mathbb{N}$, the step-indexed reflexive transitive closure R^n of R is the smallest relation satisfying the following properties:

- $\forall (x : A).\, R^0(x,x)$
- $\forall (x,y,z : A).\, R(x,y) \;\wedge\; R^{n-1}(y,z) \Rightarrow R^n(x,z)$

Definition 3.2 (Interleaved step relation). The interleaved step relation (denoted as $\to_R$) interleaves transaction local reduction with interference from concurrent transactions captured as the Rely relation (R). It is defined as follows:

$$(t,\Delta) \to_R (t',\Delta') \quad \stackrel{def}{=} \quad (t = t' \wedge R(\sigma,\sigma')) \vee ((t,\Delta) \to (t',\sigma'))$$

The interleaved multistep relation (denoted as $\to^n_R$) is the step-indexed reflexive transitive closure of $\to_R$.

Given a transaction $t = \texttt{txn}\langle \mathbb{I},\delta,\Delta\rangle\{c\}$, we use the notation $t.\delta$, $t.\Delta$, $t.\mathbb{I}$ and $t.c$ to denote the various components of t. Below, we provide a more precise definition of the transaction-local RG judgement:

$$\mathbb{R} \vdash \{P\}\, [c]_i\, \{Q\} \quad \stackrel{def}{=} \quad \forall t,\Delta,\Delta',\bar{v}.P(t.\delta,\Delta) \;\wedge\; t.c = c \wedge (t,\Delta) \to^n_{\mathbb{R}} (t_2,\Delta') \to (t',\Delta') \wedge t'.c = \texttt{SKIP} \wedge Q(t'.\delta,\Delta')$$

Here, we have explicitly stated that the last step in the reduction sequence is taken by the transaction (and not by the environment), finishing in the state satisfying the assertion Q. The nature of interference before and after the last step of the transaction are different (after the last step and before the commit step, the interference is controlled by $\mathbb{I}_c$, while before the last step, the interference is controlled by $\mathbb{I}_e$). Also, $\bar{v}$ denote valuations to all free variables in c.

Lemma 3.3. *If* $\texttt{stable}(R,Q)$*, then* $\forall \delta,\Delta,\Delta',k.Q(\delta,\Delta) \wedge R^k(\Delta,\Delta') \Rightarrow Q(\delta,\Delta')$

Proof. We use induction on k.
Base Case: For $k = 0$, $\Delta = \Delta'$ and hence $Q(\delta,\Delta')$.
Inductive Case: For the inductive case, assume that for k', $\forall \delta,\Delta,\Delta'.Q(\delta,\Delta) \wedge R^{k'}(\Delta,\Delta') \Rightarrow Q(\delta,\Delta')$. Given δ,Δ,Δ_1 such that $Q(\delta,\Delta)$, $R^{k'+1}(\delta,\Delta_1)$, we have to show $Q(\delta,\Delta_1)$. There exists Δ' such that $R^{k'}(\Delta,\Delta')$ and $R(\Delta',\Delta_1)$. By the inductive hypothesis, $Q(\delta,\Delta')$. $\texttt{stable}(R,Q)$ is defined as follows:

$$\texttt{stable}(R,Q) = \forall \delta,\Delta,\Delta'.Q(\delta,\Delta) \wedge R(\Delta,\Delta') \Rightarrow Q(\delta,\Delta')$$

Instantiating the above statement with δ,Δ',Δ_1, we get $Q(\delta,\Delta_1)$ □

Theorem 3.4. *RG-Txn is sound.*

Proof.

$$
\begin{array}{ll}
\texttt{stable}(R, I) & HI \\
\texttt{stable}(R, \mathbb{I}) & H\mathbb{I} \\
P(\delta, \Delta) \Leftrightarrow \delta = \emptyset \wedge I(\Delta) & HP \\
\mathbb{R}_e(\delta, \Delta, \Delta') \Leftrightarrow \exists \Delta_1 . R(\Delta, \Delta') \wedge \mathbb{I}_e(\delta, \Delta_1, \Delta') & H\mathbb{R}_l \\
\mathbb{R}_e \vdash \{P\}\, c\, \{Q\} & Hc \\
\mathbb{R}_c(\delta, \Delta, \Delta') \Leftrightarrow \exists \Delta_1 . R(\Delta, \Delta') \wedge \mathbb{I}_c(\delta, \Delta_1, \Delta') & H\mathbb{R}_c \\
\texttt{stable}(\mathbb{R}_c, Q) & HQ \\
\forall \delta, \Delta.\ Q(\delta, \Delta) \Rightarrow G(\Delta, \delta \gg \Delta) & HQG \\
\forall \Delta, \Delta'.\ I(\Delta) \wedge G(\Delta, \Delta') \Rightarrow I(\Delta') & HG \\
\forall \Delta . G(\Delta, \Delta) & HID
\end{array}
$$

Let $t_s = \texttt{TXN}_i\langle\mathbb{I}\rangle\{c\}$. Consider Δ such that $I(\Delta)$, and let $(t_s, \Delta) \rightarrow_R^n (\texttt{SKIP}, \Delta^{'})$. We have to show (1) $I(\Delta^{'})$ and (2) $\texttt{step-guaranteed}(R, G, t_s, \Delta)$. We break down the sequence of reductions into four parts :

- $\pi_1 = (t_s, \Delta) \rightarrow_R^{n_1} (t_s, \Delta_1) \rightarrow (t, \Delta_1)$, where initially only the environment takes steps and the last step in the sequence is the start of the transaction using the rule E-Txn-Start.
- $(t, \Delta_1) \rightarrow_R^{n_2} (t^{'}, \Delta_2)$, which begins from t taking its first step at state Δ_1 and ends at the first configuration where $t^{'}.c = \texttt{SKIP}$. We denote this sub-sequence by π.
- $(t^{'}, \Delta_2) \rightarrow_R^{n_3} (\texttt{SKIP}, \Delta_3)$ which ends at the step where t commits.
- $(\texttt{SKIP}, \Delta_3) \rightarrow_R^{n_4} (\texttt{SKIP}, \Delta^{'})$ where only the environment takes a step.

In the sequence π_1, $R^{n_1}(\Delta, \Delta_1)$. By $I(\Delta)$, HI and Lemma 3.3, $I(\Delta_1)$. By the rule E-Txn-Start, $t.\delta = \phi$, $t.\Delta = \Delta_1$ and $t.c = c$. Hence $P(t.\delta, \Delta_1)$.

Expanding the definition of the assertion Hc and instantiating it with $\Delta = \Delta_1$ and $\Delta^{'} = \Delta_2$, we would get $Q(t^{'}.\delta, \Delta_2)$. However, the environment steps in sequence π are in R, while the environment steps in assertion Hc are in $\mathbb{R}_e$. Hence, we will now show that only environment steps in $\mathbb{R}_e$ can actually happen in the sequence π. We will show this in two steps. In the first step, we will prove that for all configurations (t_p, Δ_p) in the sequence π except possibly the last configuration, $\mathbb{I}_e(t_p.\delta, t_p.\Delta, \Delta_p)$.

We will prove this by contradiction. Assume that there is a configuration (t_1, Δ_b) such that $\neg\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b)$. Let $(t_1, \Delta_b^{'}) \rightarrow (t_1^{'}, \Delta_b^{'})$ be the next step in π taken by the transaction. We know that this step always exists because the last step in π is taken by the transaction. Then $\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b^{'})$. All steps between (t_1, Δ_b) and $(t_1, \Delta_b^{'})$ are taken by the environment, i.e. $R^k(\Delta_b, \Delta_b^{'})$ for some k. However, $\neg\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b)$, the assertion $H\mathbb{I}$ and a simple induction on k shows that $\neg\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b^{'})$. This is a contradiction. Hence, $\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b)$.

Now, we will show that every environment step in π is in $\mathbb{R}_e$. Assume that $(t_1, \Delta_a) \rightarrow_R (t_1, \Delta_b)$ is an environment step such that $R(\Delta_a, \Delta_b)$. Then, we know that $\mathbb{I}_e(t_1.\delta, t_1.\Delta, \Delta_b)$. Hence, $t_1.\Delta$ provides the existence of Δ_1 in the definition of $\mathbb{R}_e$. Thus, $\mathbb{R}_e(t_1.\delta, \Delta_a, \Delta_b)$. We can use Hc and make the assertion $Q(t^{'}.\delta, \Delta_2)$.

Note that $t^{'}.\Delta = \Delta_2$. Also, since all the changes in the global database state have so far been made by the environment, $I(\Delta_2)$.

$(t^{'}, \Delta_2) \rightarrow_R^{n_3 - 1} (t^{'}, \Delta_2^{'}) \rightarrow (\texttt{SKIP}, \Delta_3)$, where the first $n_3 - 1$ steps are only performed by the environment. Since the transaction commits at state $\Delta_2^{'}$, by the E-Commit rule, $\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_2^{'})$. We will now show that all environment steps in the above sequence must be in $\mathbb{R}_c$. Again, we will show this in two steps. Let $m = n_3 - 1$ and $(t^{'}, \Delta_2) \rightarrow_R (t^{'}, \Delta_{21}) \rightarrow_R (t^{'}, \Delta_{22}) \ldots \rightarrow_R (t^{'}, \Delta_{2m}) \rightarrow (\texttt{SKIP}, \Delta_3)$. We will show that $\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_{2k})$ for all $k, 1 \leq k \leq m$.

We will prove this by contradiction. Suppose for some i, $\neg\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_{2i})$. Clearly, $R^j(\Delta_{2i}, \Delta^{'})$. Then, by $H\mathbb{I}$ and a simple induction on j, we can show that $\neg\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_2^{'})$. However, this is a contradiction. Hence, $\forall k$, $\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_{2k})$.

Now, we will show that every environment step is in $\mathbb{R}_c$. Consider the step $(t^{'}, \Delta_{2k}) \rightarrow_R (t^{'}, \Delta_{2(k+1)})$. We have $\mathbb{I}_c(t^{'}.\delta, \Delta_2, \Delta_{2(k+1)})$. Hence, Δ_2 provides the existence of Δ_1 in the definition of $\mathbb{R}_c$. Thus, $\mathbb{R}_c(t^{'}.\delta, \Delta_{2k}, \Delta_{2(k+1)})$.

By HQ, $Q(t^{'}.\delta, \Delta_2)$ and Lemma 3.3 we have $Q(t^{'}.\delta, \Delta_2^{'})$. Since all state changes so far have been made by the environment, $I(\Delta_2^{'})$. By HQG, $G(\Delta_2^{'}, t^{'}.\delta \gg \Delta_2^{'})$. By the E-Commit rule, $\Delta_3 = (t^{'}.\delta \gg \Delta_2^{'})$. Hence, $G(\Delta_2^{'}, \Delta_3)$. All the steps of the transaction except the commit step do not change the global database state and by HID belong to G. The commit step satisfies G. This proves the `step-guaranteed` assertion. Finally, by HG, $I(\Delta_3)$.

All the steps in $(\texttt{SKIP}, \Delta_3) \rightarrow_R^{n_4} (\texttt{SKIP}, \Delta^{'})$ are performed by the environment. Since $I(\Delta_3)$, by HI and Lemma 3.3, $I(\Delta^{'})$. □

Theorem 3.5. *RG-Select is sound*

Proof. Given the premise of RG-Select, t, Δ such that $t.c = \texttt{LET}\ x = \texttt{SELECT}\ \lambda y.e\ \texttt{IN}\ c$, $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$. The reduction sequence can be broken down into following parts:

- $\pi_1 = (t, \Delta) \rightarrow_R^{n_1} (t, \Delta_1) \rightarrow (t_1, \Delta_1)$ where initially only the environment takes steps, and ends with the application of the E-Select rule.
- $\pi_2 = (t_1, \Delta_1) \rightarrow_R^{n_2} (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$ which corresponds to the execution of c

In π_1, $R^{n_1}(\Delta, \Delta_1)$. By $P(t.\delta, \Delta)$ and $\texttt{stable}(R, P)$, we get $P(t.\delta, \Delta_1)$. By applying the E-Select rule, $t_1.\delta = t.\delta$, $t_1.c = [s/x]c$, where $s = \{r \in \Delta_1 \mid \mathsf{eval}([r/y]e) = \texttt{true}\}$. By definition of P', $P'(t_1.\delta, \Delta_1)$. The following property holds trivially:

$$R \vdash \{P \wedge x = s\}\,[c]_i\,\{Q\} \Leftrightarrow R \vdash \{P\}\,[[s/x]c]_i\,\{Q\}$$

Since $R \vdash \{P'\}\,[c]_i\,\{Q\}$, by the above property, $R \vdash \{P\}\,[[s/x]c]_i\,\{Q\}$. Since $P(t_1.\delta, \Delta_1)$, by definition of $R \vdash \{P\}\,[[s/x]c]_i\,\{Q\}$, we get $Q(t^{'}.\delta, \Delta')$.

□

Theorem 3.6. *RG-Update is sound*

Proof. Given the premise of RG-Update, t, Δ such that $t.c = \texttt{UPDATE}\ \lambda x.e_1\ \lambda x.e_2$, $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$.

Since only a single step needs to be taken by the transaction (by applying the E-Update rule), $t_2.c = t.c$, $t_2.\delta = t.\delta$ and $R^m(\Delta, \Delta')$. By $\texttt{stable}(R, P)$, $P(t_2.\delta, \Delta')$. According to E-Update, $t^{'}.\delta = t_2.\delta \cup \{r' \mid \exists(r \in \Delta').\ \mathsf{eval}([r/x]e_2) = \texttt{true} \wedge r' = [r/x]e_1\}$. From the premise of RG-Update, we know that

$$\forall \delta, \delta', \Delta.\ P(\delta, \Delta)\ \wedge\ \delta' = \delta \cup \{r' \mid \exists(r \in \Delta).\ [r/x]e_2 = \texttt{true}\ \wedge\ r' = [r/x]e_1\} \Rightarrow Q(\delta', \Delta)$$

Instantiating the above statement with $\delta = t_2.\delta$ and $\Delta = \Delta'$, we get $Q(\delta', \Delta')$. However, $\delta^{'} = t^{'}.\delta$. Hence, $Q(t^{'}.\delta, \Delta')$. □

Theorem 3.7. *RG-Insert is sound*

Proof. Given the premise of RG-Insert, t, Δ such that $t.c = \texttt{INSERT}\ x$, $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$.

Since only a single step needs to be taken by the transaction (by applying the E-Insert rule), $t_2.c = t.c$, $t_2.\delta = t.\delta$ and $R^m(\Delta, \Delta')$. By $\texttt{stable}(R, P)$, $P(t_2.\delta, \Delta')$. According to E-Insert, $t^{'}.\delta = t_2.\delta \cup \{\bar{f} = \bar{k};\ \texttt{id} = i;\ \texttt{del} = \texttt{false}\}$ and $i \notin \mathsf{dom}(t_2.\delta \cup \Delta')$. From the premise of RG-Insert, we know that

$$\forall \delta, \delta', \Delta, i.\ P(\delta, \Delta)\ \wedge\ i \notin \mathsf{dom}(\delta \cup \Delta)\ \wedge\ \delta' = \delta \cup \{\{\bar{f} = x.\bar{f};\ \texttt{id} = i;\ \texttt{del} = \texttt{false}\} \Rightarrow Q(\delta', \Delta)$$

Instantiating the above statement with $\delta = t_2.\delta$ and $\Delta = stg'$, we get $Q(\delta', \Delta')$. However, $\delta^{'} = t^{'}.\delta$. Hence, $Q(t^{'}.\delta, \Delta)$. □

Theorem 3.8. *RG-Delete is sound*

Proof. Given the premise of RG-Delete t, Δ such that $t.c = \texttt{DELETE}\ \lambda x.e$, $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \sigma^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$.

Since only a single step needs to be taken by the transaction (by applying the E-Delete rule), $t_2.c = t.c$, $t_2.\delta = t.\delta$ and $R^m(\Delta, \Delta')$. By $\texttt{stable}(R, P)$, $P(t_2.\delta, \Delta')$. According to E-Delete, $t^{'}.\delta = t_2.\delta \cup \{r' \mid \exists (r \in \Delta').\ \mathsf{eval}([r/x]e) = \texttt{true}\ \wedge\ r' = \{\bar{f} = r.\bar{f}; \texttt{id} = r.\texttt{id}; \texttt{del} = \texttt{true}\}\}$. From the premise of RG-Delete, we know that

$$\forall \delta, \delta', \Delta.\ P(\delta, \Delta)\ \wedge\ \delta' = \delta \cup \{r' \mid \exists (r \in \Delta).\ [r/x]e = \texttt{true}\ \wedge\ r' = \{\bar{f} = r.\bar{f}; \texttt{id} = r.\texttt{id}; \texttt{del} = \texttt{true}\}\} \Rightarrow Q(\delta', \Delta)$$

Instantiating the above statement with $\delta = t_2.\delta$ and $\Delta = \Delta'$, we get $Q(\delta', \Delta')$. However, $\delta^{'} = t^{'}.\delta$. Hence, $Q(t^{'}.\delta, \Delta')$. □

Theorem 3.9. *RG-Foreach is sound*

Proof.

$\texttt{stable}(R, Q)$	HQ
$\texttt{stable}(R, \psi)$	HI
$\texttt{stable}(R, P)$	HP
$P \Rightarrow [\phi/y]\psi$	$H1$
$R \vdash \{\psi \wedge z \in x\}\, [c]_i\, \{Q_c\}$	Hc
$Q_c \Rightarrow [y \cup \{z\}/y]\psi$	$H2$

Given t, Δ such that $t.c = \texttt{FOREACH}\ x\ \texttt{DO}\ \lambda y.\lambda z.c$, $(t, \Delta) \rightarrow_R^n (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$.

The operational semantics of `foreach` (E-Foreach1, E-Foreach2, E-Foreach3) essentially execute the command `c` for a number of iterations, where in each iteration, `z` is bound to a record $r \in$ `x`, while `y` is bound to a set containing records bound to `z` in previous iterations. `z` is bound to a different record in each iteration, and the loop stops when all records in `x` are iterated over.

Assuming that $|x| = s$, the reduction sequence for `foreach` will have the following structure :

$$(t, \Delta) \rightarrow_R^m (t_1, \Delta_1) \rightarrow_R^{n_1} (t_2^{'}, \Delta_2^{'}) \rightarrow_R^{n_1^{'}} (t_2, \Delta_2) \rightarrow_R^{n_2} (t_3^{'}, \Delta_3^{'}) \rightarrow_R^{n_2^{'}} (t_3, \Delta_3) \ldots (t_s, \Delta_s) \rightarrow_R^{n_s} (t_{s+1}^{'}, \Delta_{s+1}^{'}) \rightarrow_R^{l} (t^{'}, \Delta^{'})$$

The reduction sequence $\pi_i = (t_i, \Delta_i) \rightarrow_R^{n_i} (t_{i+1}^{'}, \Delta_{i+1}^{'})$ corresponds to the execution of the command `c` in the ith iteration, such that the first and last steps in π_i are not environment steps. The sequence $\pi_0 = (t, \Delta) \rightarrow_R^m (t_1, \Delta_1)$ corresponds to the steps E-Foreach1 and E-Foreach2 along with environment steps. Similarly, the sequence $\pi_i^{'} = (t_{i+1}^{'}, \Delta_{i+1}^{'}) \rightarrow_R^{n_1^{'}} (t_{i+1}, \Delta_2)$ corresponds to the execution of the E-Foreach2 step required to prepare the $(i+1)$th iteration along with environment steps.

Let $x = \{r_1, \ldots, r_s\}$, and assume that the records are picked in the increasing order. Then at the start of the ith iteration, z is bound to r_i, while y is bound to $\{r_1, \ldots, r_{i-1}\}$. We will show that $[\{r_1, \ldots, r_i\}/y]\psi$ holds at the end of iteration i, for all $1 \leq i \leq s$. More precisely, we will show $[\{r_1, \ldots, r_i\}/y]\psi(t^{'}_{i+1}.\delta, \Delta^{'}_{i+1})$. We will use induction on i.

Base Case: The steps E-Foreach1 and E-Foreach2 do not change δ. Also, $P(t.\delta, \Delta)$ and `stable`(R, P). Hence, at the end of the sequence π_0, $P(t_1.\delta, \Delta_1)$. By $H1$, this implies $[\phi/y]\psi(t_1.\delta, \Delta_1)$. The sequence $\pi_1 = (t_1, \Delta_1) \rightarrow^{n_1}_R (t^{'}_2, \Delta^{'}_2)$ corresponds the execution of c in the first iteration with z bound to r_1 and y bound to ϕ. Clearly, $\psi(t_1.\delta, \Delta_1) \wedge z \in x$ holds. Hence, by Hc, $Q_c(t^{'}_2.\delta, \Delta^{'}_2)$. By H2, this implies $[\{r_1\}/y]\psi(t^{'}_2.\delta, \Delta^{'}_2)$.

Inductive Case: Assume that $[\{r_1, \ldots, r_{k-1}\}/y]\psi(t^{'}_k.\delta, \Delta^{'}_k)$. The next sequence of reductions $(t^{'}_k, \Delta^{'}_k) \rightarrow^{n^{'}_k}_R (t_k, \Delta_k)$ only corresponds to the execution of the E-Foreach2 step for the kth iteration and environment steps. E-Foreach2 does not change δ, and since `stable`(R, ψ), we get $[\{r_1, \ldots, r_k\}/y]\psi(t_k.\delta, \Delta_k)$. At the start of the next iteration, z is bound to r_k, and y is bound to $\{r_1, \ldots, r_{k-1}\}$. Hence, $\psi(t_k.\delta, \Delta_k) \wedge z \in x$. By Hc, this implies $Q_c(t^{'}_{k+1}.\delta, \Delta^{'}_{k+1})$. By $H2$, this implies $[y \cup z/y]\psi(t^{'}_{k+1}.\delta, \Delta^{'}_{k+1}) = [\{r_1, \ldots, r_k\}/y]\psi(t^{'}_{k+1}.\delta, \Delta^{'}_{k+1})$. This proves the inductive step.

Hence, at the end of the sth iteration, $[x/y]\psi(t^{'}_{s+1}.\delta, \Delta^{'}_{s+1})$. This implies $Q(t^{'}_{s+1}.\delta, \Delta^{'}_{s+1})$. Finally, the last part of the reduction, $(t^{'}_{s+1}, \Delta^{'}_{s+1}) \rightarrow^{l}_R (t^{'}, \Delta^{'})$ corresponds environment steps and E-Foreach3 (as the last step). Since `stable`(R, Q) and E-Foreach3 does not change δ, we have $Q(t^{'}.\delta, \Delta)$.

□

Theorem 3.10. *RG-Seq is sound*

Proof.

$$\begin{array}{ll} \{P\}\,[c1]_i\,\{Q^{'}\} & H1 \\ \{Q^{'}\}\,[c2]_i\,\{Q\} & H2 \\ \texttt{stable}(R, Q^{'}) & H3 \end{array}$$

Given t, Δ such that $t.c = c1; c2$, $(t, \Delta) \rightarrow^{m}_R (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q(t^{'}.\delta, \Delta^{'})$. We can divide the reduction sequence into three parts :

- $(t, \Delta) \rightarrow^{m_1}_R (t^{'}_m, \Delta_1) \rightarrow (t_m, \Delta_1)$, where $t_m.c = c2$. We denote this sequence as π_1.
- $(t_m, \Delta_1) \rightarrow^{m_2}_R (t_m, \Delta^{'}_1)$ where all steps are taken by the environment. This sequence is denoted as π_2.
- $(t_m, \Delta^{'}_1) \rightarrow^{m_3}_R (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$. This sequence is denoted as π_3.

By the premise of the E-Seq1 and E-Seq2 rules, all the reductions in the sequence π_1 are also applicable to $c1$. Hence, consider transaction s such that $s.c = c1$, $s.\delta = t.\delta$. Then, there exists the sequence $(s, \Delta) \rightarrow^{m_1}_R (s_2, \Delta_1) \rightarrow (s^{'}, \Delta_1)$ with $s^{'}.c = \texttt{SKIP}$, $s^{'}.\delta = t_m.\delta$. Since $P(s.\delta, \Delta)$, by H1, $Q^{'}(s^{'}.\delta, \Delta_1)$. This implies $Q^{'}(t_m.\delta, \Delta_1)$.

In the sequence π_2, all steps are taken by the environment. By H3, $Q^{'}(t_m.\delta, \Delta^{'}_1)$.

Since $t_m.c = c2$, by H3, $Q(t^{'}.\delta, \Delta^{'})$.

□

Theorem 3.11. *RG-If is sound*

Proof.

$$\begin{array}{ll} \{P \wedge e\}\,[c1]_i\,\{Q\} & H1 \\ \{P \wedge \neg e\}\,[c2]_i\,\{Q\} & H2 \\ \texttt{stable}(R, P) & H3 \end{array}$$

Given t, Δ such that $t.c$ = `IF` e `THEN` c_1 `ELSE` c_2, $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P(t.\delta, \Delta)$ and $t^{'}.c$ = `SKIP`, we have to show that $Q(t^{'}.\delta, \Delta^{'})$. Assume that $\texttt{eval}(e) = \texttt{true}$. We divide the sequence of steps into two parts:

- $\pi_1 = (t, \Delta) \rightarrow_R^{n_1} (t, \Delta_1) \rightarrow (t_1, \Delta_1)$ where initially only the environment takes steps, and the last step is taken by the transaction using E-IfTrue.
- $\pi_2 = (t_1, \Delta_1) \rightarrow_R^{n_2} (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$

Since $P(t.\delta, \Delta)$ and $R^{n_1}(\Delta, \Delta_1)$, by H3, we have $P(t.\delta, \Delta_1)$. By applying the rule E-IfTrue, we have $t_1.\delta = t.\delta$, $t_1.c = c1$. Hence, $P(t_1.\delta, \Delta_1)$. By the definition of $H1$, $Q(t^{'}, \Delta')$. A similar proof follows for the case $\texttt{eval}(e) =$ `false`

□

Lemma 3.12. *If* $\texttt{stable}(R, Q)$ *and* $R' \subseteq R$*, then* $\texttt{stable}(R', Q)$

Proof. Given δ, Δ, Δ' such that $Q(\delta, \Delta)$ and $R'(\Delta, \Delta')$, we have to show that $Q(\delta, \Delta')$. Since $R' \subseteq R$, $R(\Delta, \Delta')$. Hence, by $\texttt{stable}(R, Q)$, $Q(\delta, \Delta')$. □

Lemma 3.13. *If* $\{I, R\}\, \texttt{TXN}_i\langle\mathbb{I}\rangle\{c\}\, \{G \cup ID, I\}$ *and* $R^{'} \subseteq R$*, then* $\{I, R'\}\, \texttt{TXN}_i\langle\mathbb{I}\rangle\{c\}\, \{G \cup ID, I\}$

Proof. Let $t = \texttt{TXN}_i\langle\mathbb{I}\rangle\{c\}$. Then, given Δ such that $I(\Delta)$ and $(t, \Delta) \rightarrow_{R'}^n (\texttt{SKIP}, \Delta^{'})$, we have to show (1) $I(\Delta^{'})$ and (2) $\texttt{step-guaranteed}(R', G \cup ID, t, \Delta)$. Since $R' \subseteq R$, every environment step in the above reduction sequence is in R. Thus, $(t, \Delta) \rightarrow_{R'}^n (\texttt{SKIP}, \Delta^{'})$, which by definition of $\{I, R\}\, \texttt{TXN}_i\langle\mathbb{I}\rangle\{c\}\, \{G \cup ID, I\}$ implies $I(\Delta')$. The same argument holds for $\texttt{step-guaranteed}(R', G \cup ID, t, \Delta)$. □

Theorem 3.14. *RG-Par is sound*

Proof.

$$\begin{array}{ll} \{I, R \cup G_2 \cup ID\}\, t_1\, \{G_1 \cup ID, I\} & H1 \\ \{I, R \cup G_1 \cup ID\}\, t_2\, \{G_2 \cup ID, I\} & H2 \end{array}$$

Consider Δ such that $I(\Delta)$, and let $(t_1 || t_2, \Delta) \rightarrow_R^n (\texttt{SKIP}, \Delta^{'})$. We have to show (1) $I(\Delta^{'})$ and (2) $\texttt{step-guaranteed}(R, G_1 \cup G_2 \cup ID, t_1 || t_2, \Delta)$.

Suppose that t_1 commits before t_2 in the execution sequence. Consider the sequence upto (and including) the commit step of t_1, i.e. $(t_1 || t_2, \Delta) \rightarrow_R^{n_1} (t_1^{'} || t_2^{'}, \Delta_1) \rightarrow (t_2^{'}, \Delta_1^{'})$. In this sequence, all steps apart from the steps taken by t_1 belong to $R \cup ID$, since any step taken by t_2 cannot change the global database state. Hence, there exists the sequence $(t_1, \Delta) \rightarrow_{R \cup ID}^{n_1} (t_1^{'}, \Delta_1) \rightarrow (\texttt{SKIP}, \Delta_1^{'})$. Since $R \cup ID \subseteq R \cup G_2 \cup ID$, by H1 and Lemma 3.13, $I(\Delta_1^{'})$ and $G_1(\Delta_1, \Delta_1^{'})$. Now, consider the entire sequence from the perspective of t_2. All steps taken by t_1 except the commit step do not change the global database state, and the change during the commit step belongs to G_1. Hence, all steps in the sequence apart from the steps taken by t_2 belong to $R \cup G_1 \cup ID$. Hence, there exists a sequence $(t_2, \Delta) \rightarrow_{R \cup G_1 \cup ID}^n (\texttt{SKIP}, \Delta^{'})$. By H2 $I(\Delta^{'})$.

Finally, the commit step of t_1 belongs to G_1, while the commit step of t_2 belongs to G_2, and every other step of either transaction does not change the global database state. Hence, $\texttt{step-guaranteed}(R, G_1 \cup G_2 \cup ID, t1 || t2, \Delta)$. The proof for the case where t_2 commits before t_1 would be similar. □

Theorem 3.15. *RG-Conseq is sound*

Proof.

$$\begin{array}{ll} R \vdash \{P\}\, [t]_i\, \{Q\} & H1 \\ P' \Rightarrow P & H2 \\ Q \Rightarrow Q' & H3 \end{array}$$

Given t, Δ such that $(t, \Delta) \rightarrow_R^m (t_2, \Delta^{'}) \rightarrow (t^{'}, \Delta^{'})$, $P'(t.\delta, \Delta)$ and $t^{'}.c = \texttt{SKIP}$, we have to show that $Q'(t^{'}.\delta, \Delta^{'})$. By H2, $P(t.\delta, \Delta)$. Then, expanding the definition in H1, we get $Q(t^{'}.\delta, \Delta')$. By H3, $Q'(t^{'}.\delta, \Delta^{'})$. □

Theorem 3.16. *RG-Conseq2 is sound*

Proof.

$$
\begin{array}{ll}
\{I, R\}\, \texttt{TXN}_i\langle \mathbb{I} \rangle\{c\}\, \{G, I\} & H1 \\
\mathbb{I}' \Rightarrow \mathbb{I} & H2 \\
R' \subseteq R & H2 \\
\texttt{stable}(R', \mathbb{I}') & H3 \\
G \subseteq G' & H4 \\
\forall \Delta, \Delta'.\ I(\Delta) \wedge G'(\Delta, \Delta') \Rightarrow I(\Delta') & H5
\end{array}
$$

Let $t = \texttt{TXN}_i\langle \mathbb{I}' \rangle\{c\}$. Given Δ such that $I(\Delta)$ and reduction sequence $\pi = (t, \Delta) \rightarrow_{R'}^n (\texttt{SKIP}, \Delta')$, we have to show that $I(\Delta')$ and $\texttt{step-guaranteed}(R', G', t, \Delta)$. First, we will show that the above reduction sequence is valid even if the isolation level of t is changed to $\mathbb{I}$. Assume that the transaction performs m steps in π. We will use induction on m to show that every step of the transaction is valid for isolation level $\mathbb{I}$.

For the base case, the first step is always valid irrespective of any isolation level. For the inductive case, assume that all steps upto the kth step of the transaction in t are valid with isolation level $\mathbb{I}$. Let the $(k+1)$th step of the transaction be $(t_1, \Delta_1) \rightarrow (t_2, \Delta_1)$. Then $\mathbb{I}'(t_1.\delta, t_1.\Delta, \Delta_1)$. By H2, $\mathbb{I}(t_1.\delta, t_1.\Delta, \Delta_1)$. Hence, the $k+1$th step is also valid for isolation level $\mathbb{I}$. This shows that the entire reduction sequence is valid even if the isolation level of t is changed to $\mathbb{I}$. Let $t' = \texttt{TXN}_i\langle \mathbb{I}' \rangle\{c\}$. Since $R' \subseteq R$, it follows that the reduction sequence $\pi' = (t', \Delta) \rightarrow_R^n (\texttt{SKIP}, \Delta')$ comprising of the same steps as π is valid. By H1, $I(\Delta')$. Finally, by $\texttt{step-guaranteed}(R, G, t', \Delta)$, all global database state changes caused by t' in π' are in G. But these are the same global database stage changes in π. Since $G \subseteq G'$, these state changes are also in G'. □

Theorem 3.17. *If*

$$\forall \delta, \Delta, \Delta'.\mathbb{R}_e(\delta, \Delta, \Delta') \Leftrightarrow \exists \Delta_1.R(\Delta, \Delta') \wedge \mathbb{I}_e(\delta, \Delta_1, \Delta')$$

and

$$\forall \delta, \Delta, \Delta', \Delta''.\ \neg\mathbb{I}_e(\delta, \Delta, \Delta')\ \wedge\ R(\Delta', \Delta'') \Rightarrow \neg\mathbb{I}_e(\delta, \Delta, \Delta'')$$

then $\forall \delta, \Delta, \Delta'.\exists \Delta_1.R^*(\Delta, \Delta') \wedge \mathbb{I}_e(\delta, \Delta_1, \Delta') \Rightarrow \mathbb{R}_e^*(\delta, \Delta, \Delta')$

Proof. Consider $\delta, \Delta, \Delta', \Delta_1$ and n such that $R^{n-1}(\Delta, \Delta')$ and $\mathbb{I}_e(\delta, \Delta_1, \Delta')$. Then $R(\Delta'_1, \Delta'_2), R(\Delta'_2, \Delta'_3), \ldots, R(\Delta'_{n-1}, \Delta'_n)$, where $\Delta'_1 = \Delta$ and $\Delta'_n = \Delta'$. We will show that $\mathbb{R}_e^{n-1}(\delta, \Delta, \Delta')$. Clearly, $\mathbb{R}_e(\delta, \Delta'_{n-1}, \Delta'_n)$. Also, $\mathbb{I}_e(\delta, \Delta_1, \Delta'_{n-1})$. Because otherwise, if $\neg\mathbb{I}_e(\delta, \Delta_1, \Delta'_{n-1})$, then with $R(\Delta'_{n-1}, \Delta'_n)$ would imply $\neg\mathbb{I}_e(\delta, \Delta_1, \Delta'_n)$ which contradicts our assumption that $\mathbb{I}_e(\delta, \Delta_1, \Delta')$. Hence, by definition $\mathbb{R}_e(\delta, \Delta'_{n-2}, \Delta'_{n-1})$. In this manner, we can show that $\forall i, 1 \leq i \leq n-1.\mathbb{R}_e(\delta, \Delta'_i, \Delta'_{i+1})$, because $\mathbb{I}_e(\delta, \Delta_1, \Delta'_{i+1})$. Hence, $\mathbb{R}_e^{n-1}(\delta, \Delta, \Delta')$. □

Theorem 3.18. *If* $\mathbb{I}_e = \mathbb{I}_{ss}$, *then* $\forall \delta, \Delta, \Delta'.\mathbb{R}_e(\delta, \Delta, \Delta') \Rightarrow \Delta = \Delta'$

Proof. First, we show that

$$\forall \delta, \Delta, \Delta', \Delta''.\ \neg\mathbb{I}_e(\delta, \Delta, \Delta')\ \wedge\ R(\Delta', \Delta'') \Rightarrow \neg\mathbb{I}_e(\delta, \Delta, \Delta'')$$

Given δ, Δ, Δ' such that $\neg\mathbb{I}_e(\delta, \Delta, \Delta')$, $\Delta \neq \Delta'$. Since $R(\Delta', \Delta'')$ corresponds to the commit of a transaction, either the transaction is read-only, in which case $\Delta' = \Delta''$ and hence $\Delta \neq \Delta''$ which implies $\neg\mathbb{I}_e(\delta, \Delta, \Delta'')$, or the transaction modifies/inserts a record, in which case it will also add its own unique transaction id to the record, so that $\Delta \neq \Delta''$, which again implies the result.

$\boxed{c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}}$

$$\frac{}{\texttt{INSERT}\ x \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \lambda(\delta,\Delta).\ \{y \mid y = \{x\ \texttt{with}\ r.\texttt{del} = \mathit{false};\ \texttt{txn} = i\}\} \rrbracket_{\langle \mathbb{R},I\rangle}}$$

$$\frac{G = \lambda y.\ \mathsf{if}\ [y/x]e_2\ \mathsf{then}\ \{[y/x]e_1\ \texttt{with}\ \texttt{id} = r.\texttt{id};\ \texttt{del} = y.\texttt{del};\ \texttt{txn} = i\}\ \mathsf{else}\ \emptyset}{\texttt{UPDATE}\ \lambda x.e_1\ \lambda x.e_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \lambda(\delta,\Delta).\ \Delta \gg= G \rrbracket_{\langle \mathbb{R},I\rangle}}$$

$$\frac{G = \lambda y.\ \mathsf{if}\ [y/x]e\ \mathsf{then}\ \{y\ \texttt{with}\ \texttt{del} = \mathit{true};\ \texttt{txn} = i\}\ \mathsf{else}\ \emptyset}{\texttt{DELETE}\ \lambda x.e \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \lambda(\delta,\Delta).\ \Delta \gg= G\} \rrbracket_{\langle \mathbb{R},I\rangle}}$$

$$\frac{c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}}{\texttt{LET}\ x = e\ \texttt{IN}\ c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \lambda(\delta,\Delta).\ [e/x]\,\mathsf{F}(\delta,\Delta) \rrbracket_{\langle \mathbb{R},I\rangle}}$$

$$\frac{c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F} \quad G = \lambda r.\ \mathsf{if}\ [r/x]e\ \mathsf{then}\ \{r\}\ \mathsf{else}\ \emptyset}{\texttt{LET}\ y = \texttt{SELECT}\ \lambda x.e\ \texttt{IN}\ c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \lambda(\delta,\Delta).\ [(\Delta \gg= G)/y]\,\mathsf{F}(\delta,\Delta) \rrbracket_{\langle \mathbb{R},I\rangle}}$$

$$\frac{c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1 \quad c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_2}{\texttt{IF}\ e\ \texttt{THEN}\ c_1\ \texttt{ELSE}\ c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \lambda(\delta,\Delta).\ \mathsf{if}\ e\ \mathsf{then}\ \mathsf{F}_1(\delta,\Delta)\ \mathsf{else}\ \mathsf{F}_2(\delta,\Delta)}$$

$$\frac{c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1 \quad c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_2}{c_1; c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \lambda(\delta,\Delta).\ \mathsf{F}_1(\delta,\Delta) \cup \mathsf{F}_2(\delta \cup \mathsf{F}_1(\delta,\Delta),\Delta)}$$

$$\frac{c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}}{\texttt{FOREACH}\ x\ \texttt{DO}\ \lambda y.\lambda z.\ c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \lambda(\delta,\Delta).\ x \gg= (\lambda z.\ \mathsf{F}(\delta,\Delta))}$$

Fig. 1. $\mathcal{T}$: State transformer semantics.

Now, consider δ, Δ, Δ' such that $\mathbb{R}_e(\delta, \Delta, \Delta')$.By definition of $\mathbb{R}_e$, there exists Δ_1 such that $\mathbb{I}_e(\delta, \Delta_1, \Delta')$. Hence, $\Delta_1 = \Delta'$. Now, $\mathbb{I}_e(\delta, \Delta_1, \Delta)$, because otherwise, if $\neg\mathbb{I}_e(\delta, \Delta_1, \Delta)$, then by the earlier result, $\neg\mathbb{I}_e((\delta, \Delta_1, \Delta')$ which is a contradiction. Hence, $\Delta_1 = \Delta$. This implies that $\Delta = \Delta'$. □

THEOREM 3.19. *If* $\mathbb{I}_e = \mathbb{I}_{ww}$, *then* $\forall \delta, \Delta, \Delta'.\mathbb{R}_e(\delta, \Delta, \Delta') \Rightarrow \Delta = \Delta'$

THEOREM 3.20. *Forall* i,s,$\mathbb{R}$,I,c,F, *if* $\mathtt{stable}(\mathbb{R}, I)$ *and* $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$, *then:*

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [c]_i\, \{\lambda(\delta,\Delta).\delta = s \cup \mathsf{F}(s,\Delta)\}$$

PROOF. Hypothesis:

$$\begin{array}{ll} \mathtt{stable}(\mathbb{R}, I) & H1 \\ c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F} & H2 \end{array}$$

Proof by induction on $H2$.

We prove the statement separately for every type of c. The base cases correspond to the SQL statements INSERT, UPDATE and DELETE.

Case : INSERT. We have to show that $\forall \mathbb{R}, I$ if $\mathtt{stable}(\mathbb{R}, I)$ and $\mathit{INSERT}\ x \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}$, then

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [\mathit{INSERT}\ x]_i\, \{\lambda(\delta,\Delta).\delta = s \cup \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}(s,\Delta)\}$$

We will prove the premises of the RG-INSERT rule. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$. By $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. Note that $\mathtt{stable}(\mathbb{R}, F)$ and hence, $\llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle} = \mathsf{F}$. $Q \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup F(s,\Delta)$. Given δ, Δ, i such that $P(\delta,\Delta)$ and $\delta' = \delta \cup \{x\ \mathtt{with}\ \mathtt{del} = \mathit{false};\ \mathtt{txn} = i\}$, it follows from definition of F that $Q(\delta',\Delta)$. Thus, all premises of RG-INSERT are satisfied.

Case : UPDATE. We have to show that $\forall \mathbb{R}, I$ if $\mathtt{stable}(\mathbb{R}, I)$ and $\mathit{UPDATE}\ \lambda x.e_1\ \lambda x.e_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}$, then

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [\mathit{UPDATE}\ \lambda x.e_1\ \lambda x.e_2]_i\, \{\lambda(\delta,\Delta).\delta = s \cup \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}(s,\Delta)\}$$

We will prove the premises of the RG-UPDATE rule. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$ and $Q \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s \cup \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}(\delta,\Delta)$. By $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. We can have either $\mathtt{stable}(\mathbb{R}, F)$ or $\neg\mathtt{stable}(\mathbb{R}, F)$. In either case, we will show that all premises of RG-UPDATE are satisfied.

Suppose $\mathtt{stable}(\mathbb{R}, F)$. Then $\llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle} = \mathsf{F}$. Then, given δ, Δ such that $P(\delta,\Delta)$ and $\delta' = \delta \cup \{r' \mid \exists (r \in \Delta).\ [r/x]e_2 = \mathtt{true}\ \wedge\ r' = [r/x]e_1\ \mathtt{with}\ \mathtt{id} = r.\mathtt{id};\ \mathtt{del} = y.\mathtt{del};\ \mathtt{txn} = i\}$, it follows from definition of F that $Q(\delta',\Delta)$. Suppose $\neg\mathtt{stable}(\mathbb{R}, F)$. Then, $\llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle} = \lambda(\delta,\Delta).\mathrm{exists}(\Delta', I(\Delta'), \mathsf{F}(\delta,\Delta'))$. Also, since $P(\delta,\Delta)$, we have $I(\Delta)$. Hence, $Q(\delta',\Delta)$, since Δ provides the existential Δ', and $s \cup F(s,\Delta)$ is δ'.

Case: DELETE. We have to show that $\forall \mathbb{R}, I$ if $\mathtt{stable}(\mathbb{R}, I)$ and $\mathit{DELETE}\ \lambda x.e \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}$, then

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [\mathit{DELETE}\ \lambda x.e]_i\, \{\lambda(\delta,\Delta).\delta = s \cup \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}(s,\Delta)\}$$

We will prove the premises of the RG-DELETE rule. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$ and $Q \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s \cup \llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle}(\delta,\Delta)$. By $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. We can have either $\mathtt{stable}(\mathbb{R}, F)$ or $\neg\mathtt{stable}(\mathbb{R}, F)$. In either case, we will show that all premises of RG-DELETE are satisfied.

Suppose $\mathtt{stable}(\mathbb{R}, F)$. Then $\llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle} = \mathsf{F}$. Then, given δ, Δ such that $P(\delta,\Delta)$ and $\delta' = \delta \cup \{r' \mid \exists (r \in \Delta).\ [r/x]e = \mathtt{true}\ \wedge\ r' = \{\bar{f} = r.\bar{f}; \mathtt{id} = r.\mathtt{id}; \mathtt{del} = \mathtt{true}\}\}$, it follows from definition of F that $Q(\delta',\Delta)$. Suppose $\neg\mathtt{stable}(\mathbb{R}, F)$. Then, $\llbracket \mathsf{F} \rrbracket_{\langle \mathbb{R},I\rangle} = \lambda(\delta,\Delta).\mathrm{exists}(\Delta', I(\Delta'), \mathsf{F}(\delta,\Delta'))$.Also, since $P(\delta,\Delta)$, we have $I(\Delta)$. Hence, $Q(\delta',\Delta)$, since Δ provides the existential Δ', and $s \cup F(s,\Delta)$ is δ'.

Case: SELECT. Given $\mathbb{R}, I$ such that $\mathtt{stable}(\mathbb{R}, I)$, $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$, $G = \lambda r.\ \mathtt{if}\ [r/x]e\ \mathtt{then}\ \{r\}\ \mathtt{else}\ \emptyset$, and $H = \lambda(\delta,\Delta).\ [(\Delta \gg= G)/y]\, \mathsf{F}(\delta,\Delta)$, we have to show that

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [\mathsf{LET}\ y = \mathsf{SELECT}\ \lambda x.e\ \mathsf{IN}\ c]_i\, \{\lambda(\delta,\Delta).\delta = s \cup \llbracket H \rrbracket_{\langle \mathbb{R},I\rangle}(s,\Delta)\}$$

We will prove all the premises of RG-SELECT. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$, while $Q \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup \llbracket H \rrbracket_{\langle \mathbb{R},I\rangle}(s,\Delta)$. By $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. By inductive hypothesis and $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$ we have

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\, [c]_i\, \{\lambda(\delta,\Delta).\delta = s \cup F(s,\Delta)\}$$

Let

$$P'(\delta,\Delta) \Leftrightarrow P(\delta,\Delta) \wedge y = \{r | \exists (r \in \Delta)[r/x]e\}$$

Given δ, Δ, P' just binds y to a set of records which depend on Δ. We now have the following from the inductive hypothesis:

$$\mathbb{R} \vdash \{P'\}\,[c]_i\,\{\lambda(\delta,\Delta).\delta = s \cup \llbracket[(\Delta \gg= G)/y]F(\delta,\Delta)\rrbracket_{\langle\mathbb{R},I\rangle}\}$$

The reason is that y occurs free in c and by the inductive hypothesis, any binding of y can be used. Note that if $P'(\delta,\Delta)$, then $y = (\Delta \gg= G)$. If $\mathtt{stable}(\mathbb{R}, \lambda(\delta,\Delta).[(\Delta \gg= G)/y]F(\delta,\Delta))$, then given $P'(s,\Delta_1)$,

$$\lambda(\delta,\Delta).[(\Delta \gg= G)/y]F(s,\Delta) = [(\Delta_1 \gg= G)/y]F(s,\Delta_1)$$

If $\neg\mathtt{stable}(\mathbb{R}, \lambda(\delta,\Delta).[(\Delta \gg= G)/y]F(\delta,\Delta))$, then

$$\llbracket\lambda(\delta,\Delta).[(\Delta \gg= G)/y]F(\delta,\Delta)\rrbracket_{\langle\mathbb{R},I\rangle} = \lambda(\delta,\Delta).\mathsf{exists}(\Delta', I, [(\Delta' \gg= G)/y]F(\delta,\Delta'))$$

Then, given $P'(s,\Delta_1)$, $I(\Delta_1)$ and hence Δ_1 gives the existential Δ'.

Case : IF-THEN-ELSE. Given $\mathbb{R}, I$ such that $\mathtt{stable}(\mathbb{R}, I)$, $c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1$, $c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_2$, we have to show that

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\,[\mathtt{IF}\ e\ \mathtt{THEN}\ c_1\ \mathtt{ELSE}\ c_2]_i\,\{\lambda(\delta,\Delta).\delta = s \cup (\text{if } e \text{ then } \mathsf{F}_1(s,\Delta) \text{ else } \mathsf{F}_2(s,\Delta))\}$$

We will prove all the premises of RG-If. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$, while $Q \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup (\text{if } e \text{ then } \mathsf{F}_1(s,\Delta) \text{ else } \mathsf{F}_2(s,\Delta))$. By the inductive hypothesis and $c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1$, we know that

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\,[c_1]_i\,\{\lambda(\delta,\Delta).\delta = s \cup \mathsf{F}_1(s,\Delta)\}$$

The post-condition in the above statement can also be written as $Q \wedge e$. Since e does not access the global or local database, the above statement can be written as $\mathbb{R} \vdash \{P \wedge e\}\,[c_1]_i\,\{Q \wedge e\}$. Similarly, $\mathbb{R} \vdash \{P \wedge \neg e\}\,[c_2]_i\,\{Q \wedge \neg e\}$. By $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. Thus, all the premises of RG-If are satisfied.

Case : SEQ. Given $\mathbb{R}, I$ such that $\mathtt{stable}(\mathbb{R}, I)$, $c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1$, $c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_2$, we have to show that

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\,[c_1; c_2]_i\,\{\lambda(\delta,\Delta).\delta = s \cup F_1(s,\Delta) \cup F_2(s \cup F_1(s,\Delta),\Delta)\}$$

We will prove all the premises of RG-Seq. Here, $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$, while $Q \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup F_1(s,\Delta) \cup F_2(s \cup F_1(s,\Delta),\Delta)$. Let $Q' \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup F_1(s,\Delta)$. Then, by the inductive hypothesis and $c_1 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_1$, we have $\mathbb{R} \vdash \{P\}\,[c1]_i\,\{Q'\}$. Further, by the inductive hypothesis and $c_2 \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}_2$, we have $\mathbb{R} \vdash \{Q'\}\,[c_2]_i\,\{Q\}$. Finally, since the stabilization operator ($\llbracket\rrbracket_{\langle\mathbb{R},I\rangle}$) is always applied on F_1, we have $\mathtt{stable}(\mathbb{R}, Q')$. Thus, all premises of RG-Seq are satisfied.

Case : FOREACH. Given $\mathbb{R}, I$ such that $\mathtt{stable}(\mathbb{R}, I)$, $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$, we have to show that

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)\}\,[\mathtt{FOREACH}\ x\ \mathtt{DO}\ \lambda y.\lambda z.\ c]_i\,\{\lambda(\delta,\Delta).\delta = s \cup x \gg= (\lambda z.\ \mathsf{F}(s,\Delta)\}$$

We will prove all the premises of RG-ForEach using the loop invariant $\psi(\delta,\Delta) \Leftrightarrow \delta = s \cup y \gg= (\lambda z.\ \mathsf{F}(s,\Delta))$. Here $P \Leftrightarrow \lambda(\delta,\Delta).\ \delta = s\ \wedge\ I(\Delta)$, while $Q \Leftrightarrow \lambda(\delta,\Delta).\delta = s \cup x \gg= (\lambda z.\ \mathsf{F}(s,\Delta)$. Since $[\phi/y]\psi(\delta,\Delta) \Leftrightarrow \delta = s$, $P \to [\phi/y]\psi$. By the inductive hypothesis and $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$, we have

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = s \cup y \gg= (\lambda z.\ \mathsf{F}(s,\Delta)) \wedge I(\Delta)\}\,[c]_i\,\{\lambda(\delta,\Delta).\delta = s \cup y \gg= (\lambda z.\ \mathsf{F}(s,\Delta)) \cup F(s \cup y \gg= (\lambda z.\ \mathsf{F}(s,\Delta)),\Delta)\}$$

Now, since all iterations are independent of each other, $F(s \cup y \gg= (\lambda z.\ \mathsf{F}(s,\Delta)),\Delta) = F(s,\Delta)$. Binding z (which is free in c) to a record in x (i.e. $z \in x$) in the pre-condition, the post condition in the above statement implies $\delta = s \cup (y \cup \{z\}) \gg= (\lambda z.\ \mathsf{F}(s,\Delta))$, which is nothing but $[y \cup \{z\}/y]\psi(\delta,\Delta)$. Hence, ψ is a loop invariant. Finally, $[x/y]\psi \to Q$.

From $\mathtt{stable}(\mathbb{R}, I)$, we have $\mathtt{stable}(\mathbb{R}, P)$. Since F has been stabilized using the $\llbracket\rrbracket_{\langle\mathbb{R},I\rangle}$ function, and ψ is an assertion on the union of multiple applications of F, it follows that $\mathtt{stable}(\mathbb{R}, \psi)$. Using the same reasoning, $\mathtt{stable}(\mathbb{R}, Q)$. Thus, all the premises of RG-ForEach are satisfied.

□

THEOREM 3.21. Forall *i,R,I,c,*F*, if* $\texttt{stable}(\mathbb{R}, I)$ *and* $c \Longrightarrow_{\langle i,\mathbb{R},I\rangle} \mathsf{F}$*, then:*

$$\mathbb{R} \vdash \{\lambda(\delta,\Delta).\ \delta = \emptyset\ \wedge\ I(\Delta)\}\, [c]_i\, \{\lambda(\delta,\Delta).\delta = \mathsf{F}(\emptyset,\Delta)\}$$

PROOF. Follows from the stronger version of this theorem (Theorem 3.20) by substituting $\emptyset$ for s. □