1 Abstraction Soundness

Theorem 1 (Simulation).
If $\varsigma \to \varsigma'$ and $|\varsigma\rangle_{ca} \subseteq \varsigma$, then there exists $\varsigma''$ such that $\varsigma \to \varsigma''$ and $|\varsigma''\rangle_{ca} \subseteq \varsigma'$.

Proof. By cases on $\varsigma$.

1. Case $\varsigma = UE$:
   \begin{align*}
   UE &= \{ (f \ q \ t \ v \ e) \} \ (\beta \ u \ v \ t \ s \ st) \ and \\
   \bar{UE} &= \{ (f \ q) \ (\beta \ u \ v \ t \ s \ st) \} \ where \\
   |ve|_{ca} \subseteq h & \ so \ |proc = A_{u}(f, \beta \ u \ v)\rangle_{ca} \subseteq \hat{A}(f, h) \ \exists \ ulam \ \ and \ |d = A_{u}(e, \beta \ u \ v)\rangle_{ca} \subseteq \hat{A}(e, h) = \hat{d}; \\
   c &= A_{k}(q, \beta \ k, st), \ reconstruct(CP(\gamma, \beta \ k, st) = \hat{st}, \ \ and \ \ Lemma 1 \ \ so \ \ pop(c, (\beta \ u \ v) :: st) = st', (\hat{q}, \hat{st}') = reconstruct^{*}(c, st'), \ and \ (\hat{q}, \hat{st}') = \hat{pop}(q, st); \\
   so \ |(proc, d, c, st', ve, t')\rangle_{ca} & \subseteq (ulam, \hat{d}, \hat{q}, \hat{st}', h)
   \end{align*}

2. Case $\varsigma = CE$:
   \begin{align*}
   CE &= \{ (q \ e) \} \ (\beta \ u \ v \ t \ s \ st) \ and \\
   \bar{CE} &= \{ (q \ e) \} \ (\beta \ u \ v \ t \ s \ st) \ where \\
   |ve|_{ca} \subseteq h & \ so \ |d = A_{u}(e, \beta \ u \ v)\rangle_{ca} \subseteq \hat{A}(e, h) = \hat{d}; \\
   (cp, fp) &= c = A_{k}(q, \beta \ k, st), \ reconstruct(CP(\gamma, \beta \ k, st) = \hat{st}, \ \ and \ \ Lemma 1 \ \ so \ \ pop(c, (\beta \ u \ v) :: st) = st', (\hat{cp}, \hat{st}') = reconstruct^{*}(c, st'), \ and \ (\hat{cp}, \hat{st}') = \hat{pop}(cp, st); \\
   so \ |(cp, d, st', ve, t')\rangle_{ca} & \subseteq (cp, \hat{d}, \hat{st}', h)
   \end{align*}

3. Case $\varsigma = UA$:
   \begin{align*}
   UA &= \{ (proc, d, c, st, ve, t) \} \ and \\
   \bar{UA} &= \{ (ulam, \hat{d}, \hat{q}, \hat{st}, h) \} \ where \\
   |proc|_{ca} &= \{ ulam = (\lambda(u \ k) \ call) \}, \\
   (\hat{q}, \hat{st}) &= reconstruct^{*}(c, st) \ so \ reconstruct(CP(call), \beta \ k, st) = reconstruct(k, \beta \ k \ st) = \hat{st}, \ \ and \\
   |ve|_{ca} \subseteq h & \ and \ |d|_{ca} \subseteq \bar{d} \ so \ |ve'|_{ca} = |ve|_{ca} \cup |(u, t') \rightarrow d|_{ca} = |ve|_{ca} \cup |(u, t') \rightarrow d|_{ca} = |d|_{ca} \subseteq h \cup |u \rightarrow d| = h' \\
   so \ |(call, \beta \ u \ v' \ k', st, ve, t')|_{ca} & \subseteq (call, \hat{st}', h')
   \end{align*}

4. Case $\varsigma = CA$:
   \begin{align*}
   CA &= \{ clam, d, st, ve, t \} \ and \\
   \bar{CA} &= \{ clam, d, \hat{st}, h \} \ where
   \end{align*}
clam = (λ(u) call)γ,
(⟨cp⟩, ˆst) = reconstruct∗(⟨cp, [st]|⟩, st), and
|ve|ca ⊆ h and |d|ca ⊆ d so |ve'|ca = |ve[[u, t'] → d]|ca = |ve|ca ∪ |[(u, t') → d]|ca
|call, βu, βk, st, ve', t'|ca ⊆ (call, ˆst, h')

Definition 1 A stack st is well-formed if, for every continuation environment βk at stack level n, the frame pointer fp of each continuation c in βk is less than n.

Lemma 1. Suppose |UE|ca ⊆ UE where UE = ((f e qγ), βu, βk, st, ve, t) and UE is well-formed. If k = CP(γ), A_k(q, β_u, st) = c, reconstruct(k, β_k, st) = ˆst, and pop(c, (β_u, β_k) :: st) = st', then reconstruct∗(c, st') = ˆpop(q, ˆst).

Proof. By induction on st.

1. Base case st = 0:
reconstruct(k, β_k, 0) = [k → halt] :: 0 ⇐ reconstruct∗(halt, 0), 0) =
(halt, 0)
If q = k', then pop(c, (β_u, β_k) :: 0) = pop((halt, 0), (β_u, β_k) :: 0) = 0 and reconstruct∗(halt, 0), 0) = (halt, 0) = ˆpop(k', [k → halt] :: 0).
Otherwise, pop(c, (β_u, β_k) :: 0) = (β_u, β_k) :: 0 and reconstruct∗(c, (β_u, β_k) :: 0) = (q, [k → halt] :: 0) = ˆpop(q, [k → halt] :: 0).

2. Inductive case st = β_u, β_k :: st_k:
reconstruct(k, β_k, β_u, β_k :: st_k) = [k → ˆq] :: ˆst_k ⇐ reconstruct∗(c, st_k) = (ˆq, ˆst_k)
If q = k', then pop(c, (β_u, β_k) :: st_k) = pop(c, (β_u', β_k') :: st_k) = st' and reconstruct∗(c, st_k) = (ˆq, ˆst_k) = ˆpop(q', [k → ˆq] :: ˆst_k).
Otherwise, pop(c, (β_u, β_k) :: β_u', β_k' :: st_k) = (β_u, β_k) :: (β_u', β_k') :: st_k and reconstruct∗(c, (β_u, β_k) :: (β_u', β_k') :: st_k) = (q, [k → halt] :: st_k) = ˆpop(q, [k → ˆq] :: st_k).

The following lemma establishes that calls are more conservative than exits:
a user call with a continuation argument q will pop at most as many frames as a continuation call with operator q; moreover, the positional continuation mapping is preserved on the stack.

Lemma 2 (Conservative Pop).
Let ˆq = π_i(ˆq). If ˆpop(ˆq, ˆst) = ((clam), ˆst_0) and ˆpop(ˆq, ˆs_0) = (q', ˆs'), then
ˆpop(π_i(q'), ˆs') = ((clam), ˆs_0).

Proof. By cases on ˆq.

– Case ˆq = clam: By definition, ˆpop(ˆq, ˆst) = (q, ˆst). Then ˆpop(π_i(q'), ˆs') = ˆpop(π_i(ˆq), ˆs) = ˆpop(ˆq, ˆs) = ((clam), ˆst_0), by assumption.

– Case ˆq = k: By induction on whether π_i(ˆq) = clam for some i. If so, then ˆpop(ˆq, ˆs) = (q, ˆst). If not, then ˆpop(ˆq, ˆs) = ˆpop(ˆq, sm :: ˆs'') = ˆpop(sm(ˆq), ˆs'') and the result follows by induction.
Lemma 3 (Conservative Path).

Suppose $\mathcal{U}A \equiv_p \text{CEE}$ by $n$ where $\mathcal{U}A = (\ulam, \vec{d}, \vec{q}, \vec{s}, h)$ and $CV(\text{CEE}) = k$.
If $\overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, \vec{s}) = (\langle \text{clam} \rangle, \vec{s'} \rangle)$, then $\overline{\text{pop}}(k, \vec{s}_{\text{CEE}}) = (\langle \text{clam} \rangle, \vec{s'})$.

Proof. By induction on the definition of $\cdot \equiv_p \cdot$ by $\cdot$.

1. Case $p \equiv \mathcal{U}A \twoheadrightarrow \vec{q}' \twoheadrightarrow \text{CEE}$: By $\mathcal{U}A \twoheadrightarrow \vec{q}'$, $\vec{s}_{\vec{q}'} = sm :: \vec{s}_{\mathcal{U}A}$ where $sm(k) = \pi_n(\vec{q})$ where $CP(\mathcal{U}A, k) = n$. By Lemma 5, $\vec{s}_{\text{CEE}} = sm :: \vec{s}_{\mathcal{U}A}$. Then $\overline{\text{pop}}(k, \vec{s}_{\text{CEE}}) = \overline{\text{pop}}(k, sm :: \vec{s}_{\mathcal{U}A})$. By definition, $\overline{\text{pop}}(k, sm :: \vec{s}_{\mathcal{U}A}) = \overline{\text{pop}}(\langle sm(k) \rangle, \vec{s}_{\mathcal{U}A})$. By the above, $\overline{\text{pop}}(\langle sm(k) \rangle, \vec{s}_{\mathcal{U}A}) = \overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, \vec{s}_{\mathcal{U}A})$. 
By assumption, $\overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, \vec{s}_{\mathcal{U}A}) = (\langle \text{clam} \rangle, \vec{s'})$.

2. Case $p \equiv \mathcal{U}A \twoheadrightarrow \vec{q}' \twoheadrightarrow^* \mathcal{U}E \twoheadrightarrow \mathcal{U}A_0 \twoheadrightarrow^+ \text{CEE}$ where the operator of $\mathcal{U}A$ is $(\lambda_{\psi}(u k_1 \ldots k_N) \text{call})$, the call of $\mathcal{U}E$ is $(f \ e \ q_1 \ldots q_N)_{\psi_0}$, and $\mathcal{U}A_0 \equiv_p \text{CEE}$ by $n_0$:
Let $\vec{q}' = \langle q_1, \ldots, q_N \rangle$. By $\mathcal{U}A \twoheadrightarrow \vec{q}'$, $\vec{s}_{\vec{q}'} = sm :: \vec{s}_{\mathcal{U}A}$ where $sm(k) = \pi_n(\vec{q})$.
By Lemma 5, $\vec{s}_{\text{CEE}} = sm :: \vec{s}_{\mathcal{U}A}$. By assumption, $\overline{\text{pop}}(\langle \pi_n(\vec{q}') \rangle, \vec{s}_{\text{CEE}}) = \overline{\text{pop}}(\langle \pi_n(\vec{q}') \rangle, sm :: \vec{s}_{\mathcal{U}A})$. By above, $\overline{\text{pop}}(\langle \pi_n(\vec{q}') \rangle, sm :: \vec{s}_{\mathcal{U}A}) = \overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, sm :: \vec{s}_{\mathcal{U}A})$.
By assumption, $\overline{\text{pop}}(\langle sm(k_n) \rangle, sm :: \vec{s}_{\mathcal{U}A}) = \overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, sm :: \vec{s}_{\mathcal{U}A})$. 
By definition, $\mathcal{U}E \twoheadrightarrow (\ulam, \vec{d}_0, \vec{q}_0, \vec{s}_0, h_0)$ where $(\vec{q}_0, \vec{s}_0) = \overline{\text{pop}}(\vec{q}', \vec{s}_{\mathcal{U}E})$. By Lemma 2, $\overline{\text{pop}}(\langle \pi_n(\vec{q}_0) \rangle, \vec{s}_0) = (\langle \text{clam} \rangle, \vec{s'})$.
By induction, $\overline{\text{pop}}(\langle \pi_n(\vec{q}_0) \rangle, \vec{s}_{\text{CEE}}) = (\langle \text{clam} \rangle, \vec{s'})$.

Lemma 4 (Same Stack).

If $p \equiv \mathcal{U}E \twoheadrightarrow \mathcal{U}A \twoheadrightarrow^+ \text{CEE} \twoheadrightarrow \vec{q} \text{ where call}_{\vec{q}} = (f \ e \ q_1 \ldots q_n)_{\psi}$, $q_n \in \text{CLam}$, and $\mathcal{U}A \equiv_p \text{CEE}$ by $n$, then $\vec{s}_{\vec{q}} = sm :: \vec{s}$ and $\vec{s}_{\text{CEE}} = sm :: \vec{s}$.

Proof. Let $\vec{q} = \langle q_1, \ldots, q_n \rangle$ so that $\pi_n(\vec{q}) = \text{clam}$. By $\mathcal{U}E \twoheadrightarrow \mathcal{U}A$, if $S(f)$. Then $\overline{\text{pop}}(\langle \pi_n(\vec{q}) \rangle, \vec{s}_{\text{CEE}}) = \overline{\text{pop}}(\langle \text{clam} \rangle, \vec{s}_{\text{CEE}})$ and, by definition, $\overline{\text{pop}}(\langle \text{clam} \rangle, \vec{s}_{\text{CEE}}) = (\langle \text{clam} \rangle, \vec{s}_{\text{CEE}})$.
By $\text{CEE} \twoheadrightarrow \vec{q}$, $\vec{q} = (\text{clam}', \vec{d}, \vec{s}, h)$ where $(\langle \text{clam}' \rangle, \vec{s}) = \overline{\text{pop}}(\langle CV(\text{CEE}) \rangle, \vec{s}_{\text{CEE}})$. By the above and Lemma 3, $\overline{\text{pop}}(\langle CV(\text{CEE}) \rangle, \vec{s}_{\text{CEE}}) = (\langle \text{clam} \rangle, \vec{s}_{\text{CEE}})$.

Lemma 5 (Single Frame). If $p \equiv \mathcal{U}A \twoheadrightarrow^+ \vec{q}$, then there exists sm such that, for all $\vec{q}$, if $\mathcal{U}A = CE_p(\vec{q})$, then $\vec{s}_{\vec{q}} = sm :: \vec{s}_{\mathcal{U}A}$.

Proof. By induction on the definition of $CE_p$.

1. Path composition doesn’t satisfy the premise.
2. By induction on $|p|$.
   (a) Base case of $p \equiv \mathcal{U}A \twoheadrightarrow^0 \vec{q} \twoheadrightarrow \vec{q} \equiv_p \text{CEE}$ holds by definition of $\twoheadrightarrow$; instantiate $sm$ thereby.
   (b) Inductive case of $p \equiv \mathcal{U}A \twoheadrightarrow^+ \vec{q} \twoheadrightarrow \vec{q}$ where $\mathcal{U}A = CE_p(\vec{q})$, $\vec{q} \not\in \text{UEval}$, $\vec{q} \not\in C\text{EvalExit}$, and $\vec{s}_{\vec{q}} = sm :: \vec{s}_{\mathcal{U}A}$; $\vec{s}_{\vec{q}} = sm :: \vec{s}_{\mathcal{U}A}$ by cases of $\vec{q}$ in $\vec{q}$.
3. By induction, $\vec{s}_{\mathcal{U}A} = sm :: \vec{s}_{\mathcal{U}A}$. By Lemma 4, $\vec{s}_{\vec{q}} = sm :: \vec{s}_{\mathcal{U}A}$.
Lemma 6 (Local Simulation Soundness).

If \( \xi \rightsquigarrow \xi' \) and \( \text{succ}(|\xi|_{al}) \neq \emptyset \), then \( |\xi'|_{al} \in \text{succ}(|\xi|_{al}) \).

Proof. By cases on \( \xi \).

1. Case \( \xi = ((\lambda_\gamma (u_1 \ldots u_n k_1 \ldots k_m) \text{call}), \hat{d}, \hat{q}, \hat{s}, h) \):

   In the abstract, we have \( \hat{\xi} \rightsquigarrow (\text{call}, \hat{\delta}, h') \) where \( \hat{\delta} = \text{sm} :: \hat{s} \).

   Locally, we have \( \text{succ}(|\hat{\xi}|_{al}) = \text{succ}((\text{ulum}, \hat{d}, h)) = \{(\text{call}, h')\} \). Since \( |\xi'|_{al} = (\text{call}, h') \), we get \( |\xi'|_{al} \in \{|\xi'|_{al}\} \).

2. Case \( \xi = ((f e_1 \ldots e_n q_1 \ldots q_m), \gamma, \hat{s}, h) \):

   In the abstract, we have \( \hat{\xi} \rightsquigarrow (\text{ulum}, \hat{d}, \hat{q}', \hat{s}'') \) for \( \text{ulum} \in \hat{\mathcal{A}}(f, h) \) where \( \hat{d} = \langle \hat{d}_1, \ldots, \hat{d}_n \rangle \) for \( \hat{\delta}_i = \hat{\mathcal{A}}(e_i, h) \) and \( \langle \hat{q}', \hat{s}' \rangle = \hat{\text{pop}}(\hat{q}, \hat{s}) \) for \( \hat{q} = \langle \hat{q}_1, \ldots, \hat{q}_m \rangle \).

   Locally, we have \( \text{succ}(|\hat{\xi}|_{al}) = \text{succ}(((f e_1 \ldots e_n q_1 \ldots q_m), h)) = \{|\text{ulum}, \hat{d}, h\} \).

3. Case \( \xi = ((\lambda_\gamma (u_1 \ldots u_n) \text{call}), \hat{d}, \text{sm} :: \hat{s}, h) \):

   In the abstract, we have \( \hat{\xi} \rightsquigarrow (\text{call}, \text{sm} :: \hat{s}, h') \) where \( \hat{d} = \langle \hat{d}_1, \ldots, \hat{d}_n \rangle \).

   Locally, we have \( \text{succ}(|\hat{\xi}|_{al}) = \text{succ}((\text{clam}, \hat{d}, h')) = \{(\text{call}, h')\} \) where \( \hat{d} = \langle \hat{d}_1, \ldots, \hat{d}_n \rangle \). Since \( |\xi'|_{al} = (\text{call}, h') \), we get \( |\xi'|_{al} \in \{|\xi'|_{al}\} \).

4. Case \( \xi = ((\text{clam} e_1 \ldots e_n), \gamma, \hat{s}, h) \):

   In the abstract, we have \( \hat{\xi} \rightsquigarrow (\text{clam}, \hat{d}, \hat{s}, h) \) where \( \hat{d} = \langle \hat{d}_1, \ldots, \hat{d}_n \rangle \) for \( \hat{\delta}_i = \hat{\mathcal{A}}(e_i, h) \) since \( (\text{clam}, \hat{s}) \).

   Locally, we have \( \text{succ}(|\hat{\xi}|_{al}) = \text{succ}((\text{clam} e_1 \ldots e_n), h)) = \{|\text{clam}, \hat{d}, h\} \)

   where \( \hat{d} = \langle \hat{d}_1, \ldots, \hat{d}_n \rangle \) for \( \hat{\delta}_i \) since \( (\text{clam}, \hat{s}) \). Since \( |\xi'|_{al} = (\text{clam}, \hat{d}, h) \), we get \( |\xi'|_{al} \in \{|\xi'|_{al}\} \).

5. \( \xi = ((k e_1 \ldots e_n), \gamma, h) \):

   \( \text{succ}(|\xi|_{al}) = \emptyset \) so the premise doesn’t hold.

Lemma 7 (Local Simulation Completeness).

If \( \xi \rightsquigarrow \xi' \), then, for each \( \xi \) such that \( \xi = |\xi|_{al} \), there exists \( \xi' \) such that \( \xi' = |\xi'|_{al} \) and \( \xi \rightsquigarrow \xi' \).

Proof. By similar arguments as the proof for local simulation soundness.
4 Path Decomposition

Lemma 8 (Path Decomposition).

All paths can be decomposed as follows:

1. If \( p \equiv \hat{I}(pr, d) \sim^+ CEE \), then \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+ CEE \) where \( UA_i = CE_p(UE_i) \) and \( UA \equiv_p CEE \) by \( m \) for some \( m \) and the \( m \)th continuation argument of \( UEn \) is some clause.

2. If \( p \equiv \hat{I}(pr, d) \sim^* \xi \) where \( \xi \notin CEvalExit \), then \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^* \xi \) where \( UA_1 = CE_p(UE_1) \) and \( UA = CE_p(\xi) \).

Proof. By induction on \(|p|\).

- Base case \( p \equiv \hat{I}(pr, d) \sim^0 UA \): The path matches form 2 with \( n = 0 \). By definition of \( CE_p \), \( \hat{I}(pr, =)CE_p(UA) \).

- Inductive case \( p \equiv \hat{I}(pr, d) \sim^* \xi \sim \xi \): By cases on \( \xi \).

1. Case \( \xi = UA \): Then \( \xi' = UE \) and we have \( \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+ CA \). By definition of \( CE_p \), we have \( UA = CE_p(\xi) \). Thus, \( p \) matches form 2.

2. Case \( \xi = CA \): By cases on \( \xi' \).

   (a) Case \( \xi' = CEE \): We have \( \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+ CEE \). By definition of \( CE_p \), we have \( UA = CE_p(\xi) \). Then \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+ CA \). Thus, \( p \) matches form 2.

   (b) Case \( \xi' = CEE \): We have \( \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+ CEE \). By Lemma 4, \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ CA \) where \( UAn = CE_p(CA) \). Thus, \( p \) matches form 2.

3. Case \( \xi = UE \): Then \( \xi' = A \) and we have \( \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^* A \). By definition of \( CE_p \), \( UA = CE_p(\xi) \). By definition of \( \sim^* \), \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UEn \sim UA \sim^+UE \). Thus, \( p \) matches form 2.

4. Case \( \xi = CEE \): Similar to previous case.

5. Case \( \xi = CEE \): For \( m = CP(UA, CV(CEE)) \), we have \( UA \equiv_p CEE \) by \( m \). By induction on \( n \).

   (a) Base case \( UA_{i+1} \equiv_p CEE \) by \( m_{i+1} \) and \( CA(UE_i, m) \in CLam \): Then \( p \equiv \hat{I}(pr, d) \sim^0 UA_1 \sim^+ UE_1 \sim \ldots \sim UAn \sim^+ UE_i \sim UA_{i+1} \sim^+ CEE \).

   (b) Inductive case \( UA_{i+1} \equiv_p CEE \) by \( m_{i+1} \) and \( CA(UE_i, m_{i+1}) \in CVar \): Then \( UA_i \equiv_p CEE \) by \( m_i \) for \( m_i = CP(UA_i, m_{i+1}) \).
5 Path Normalization

Definition 2 (Push Monotonicity) A path \( p \equiv \Upsilon \Rightarrow^+ \zeta \) is push monotonic if \( \tilde{s}_1 \Upsilon_1 \) is a suffix of \( \tilde{s}_2 \zeta \); for each \( \zeta' \) in \( p \).

For \( p \equiv \Upsilon \Rightarrow^+ \zeta \) \( \equiv p \) CEE, even if \( \Upsilon \equiv \zeta \) CEE by \( n \), \( p \) isn’t necessarily push monotonic: a tail call within might pop the stack below the point of entry. However, such a path can be normalized to remove incidental stack, and the result is push monotonic.

Definition 3 (Path Normalization) \( F(p) = F_1(p, \langle \rangle) \) for \( p \equiv \Upsilon \Rightarrow^+ \zeta \) CEE where \( \Upsilon \equiv \zeta \) CEE by \( n \)

\[
F_1(p, \tilde{s}) = F_2(p, \tilde{s}, \tilde{s}', \langle \text{halt}, \ldots, \text{halt} \rangle) \text{ where } p \equiv \Upsilon \Rightarrow^+ \zeta \text{ CEE and } \Upsilon = (\ulam, \tilde{d}, \tilde{q}, \tilde{s}', h) \text{ where } |\tilde{q}| = |\langle \text{halt}, \ldots, \text{halt} \rangle|,
\]

\[
F_2(p, \tilde{s}, \tilde{s}', \tilde{q}') = G_2(\Upsilon \Upsilon_1, \tilde{s}, \tilde{s}', \tilde{q}') \Rightarrow^+ G_2(\zeta \Upsilon_1, \tilde{s}, \tilde{s}', \tilde{q}') \text{ if } \Upsilon \equiv \zeta \zeta \zeta \Upsilon \Upsilon_1 \Rightarrow^+ \zeta \zeta \text{ CEE}
\]

\[
F_3(p, \tilde{s}, \tilde{s}', \tilde{q}) = F_2(p, \tilde{s}, \tilde{s}', \tilde{q}') \text{ if } \tilde{s}' \text{ is a suffix of } \tilde{s}_1 \Upsilon \Upsilon_1 \text{ where } p \equiv \Upsilon \Rightarrow^+ \zeta \zeta \zeta \text{ CEE}
\]

\[
F_3(p, \tilde{s}, \tilde{s}', \tilde{q}) = F_1(p, \langle \rangle) \text{ if } \tilde{s}' \text{ is not a suffix of } \tilde{s}_1 \Upsilon \Upsilon_1 \text{ where } p \equiv \Upsilon \Rightarrow^+ \zeta \zeta \zeta \text{ CEE}
\]

\[
G_2((\ulam, \tilde{d}, \tilde{q}, \tilde{s}, h), \tilde{s}, \tilde{s}', \tilde{q}') = (\ulam, \tilde{d}, \tilde{q}', \tilde{s}', h)
\]

\[
G_2((\ldots, \tilde{s}_0, h), \tilde{s}, \tilde{s}', \tilde{q}') = (\ldots, \tilde{s}', h).
\]

\[
\tilde{s}'' = fr_1 :: \cdots :: fr_n :: fr'' :: \tilde{s}'
\]

\[
fr'' = sm''
\]

\[
sm'' = \left[ k_1 \mapsto \tilde{q}'_1, \ldots, k_m \mapsto \tilde{q}'_m \right]
\]

\[
\tilde{q}' = \langle \tilde{q}'_1, \ldots, \tilde{q}'_m \rangle
\]

\[
sm = \left[ k_1 \mapsto \tilde{q}_1, \ldots, k_m \mapsto \tilde{q}_m \right]
\]

\[
fr = sm
\]

\[
\tilde{s}_0 = fr_1 :: \cdots :: fr_n :: fr :: \tilde{s}
\]

Lemma 9 (Stack Irrelevance).

\[
\text{If } p \equiv \Upsilon \Rightarrow^+ \zeta \zeta \zeta \text{ CEE where } \Upsilon \equiv (\ulam, \tilde{d}, \tilde{q}, \tilde{s}, h), \Upsilon \equiv \zeta \zeta \zeta \text{ CEE by } n, \text{ and } \tilde{p} = (\pi_n(\tilde{q})), \tilde{s}). \text{ Then, for any stack } \tilde{s}'' = F_{\Upsilon \Upsilon_1} \tilde{s}'' \equiv p F_{\zeta \zeta \zeta} \tilde{s}' \zeta'' \text{ by } n.
\]

Proof. After application of Definition 3, by induction on \( \cdot \equiv p \cdot \) by \( \cdot \).

6 Summarization Soundness

We prove that summarization is sound by induction on path length. In the inductive step, we discriminate the penultimate state in the path. By the quasicompleteness of the local semantics and the explicit handling of returns by the algorithm, every possible ultimate state of the path is considered.
Theorem 2 (Summarization Soundness).

After summarization,

1. if \( p \equiv T(pr, d) \rightarrow* \forall A \rightarrow* \xi \) such that \( \forall A = CE_{\xi}(I) \), \( (|\forall A|_{al}, |\xi|_{al}) \in \text{Summary} \);
2. if \( p \equiv T(pr, d) \rightarrow* \forall A \rightarrow+ \xi \)\text{EE} such that \( \exists p \in \xi \text{EE} \) by \( n \), then \( (|\forall A|_{al}, |\xi|_{al}) \in \text{Summary} \); and
3. if \( p \equiv T(pr, d) \rightarrow+ \xi \) such that \( \xi \) is a final state, then \( |\xi|_{al} \in \text{Final} \).

Proof. By induction on \( |p| \).

Base case \( p \equiv T(pr, d) \rightarrow^0 T(pr, d) \): At summarization commencement, \( (T(pr, )T(pr, )) \in \text{Summary} \).

Inductive case \( p \equiv T(pr, d) \rightarrow^+ \xi \rightarrow \xi' \):

By cases on \( \xi \).

1. Case \( \xi = \forall A \) By induction, \( (|\forall A|_{al}, |\xi|_{al}) \) is added to \( \text{Work} \), since \( \xi = CE_{\xi}(I) \).

By Lemma 7, the first case of the main loop calls \( \text{Propagate}(|\forall A|_{al}, |\xi'|_{al}) \).

The result follows from the soundness of \( \text{Propagate} \).

2. Case \( \xi = C \) or \( \xi = C \cdot E \) By induction, \( (|\forall A|_{al}, |\xi|_{al}) \) is added to \( \text{Work} \), where \( \forall A = CE_{\xi}(I) \).

By Lemma 7, the first case of the main loop calls \( \text{Propagate}(|\forall A|_{al}, |\xi'|_{al}) \).

The result follows from the soundness of \( \text{Propagate} \).

3. Case \( \xi = CE \) By induction, \( (|\forall A|_{al}, |\xi|_{al}) \) is added to \( \text{Work} \), where \( \forall A = CE_{\xi}(I) \).

By Lemma 7, the second case of the main loop calls \( \text{Propagate}(|\forall A|_{al}, |\xi'|_{al}) \), since \( \xi' = CE_{\xi'}(I) \).

If a summary exists, then it holds by Lemma 9. If a summary doesn’t exist, then it holds by Lemma 4.

4. Case \( \xi = \xi \cdot E \) By induction, \( (|\forall A|_{al}, |\xi|_{al}) \) is added to \( \text{Work} \), where \( \forall A = CE_{\xi}(I) \).

The third case of the main loop calls \( \text{Return}(|\forall A|_{al}, |\xi|_{al}, CP(|\forall A|_{al}, CV(|\xi|_{al}))) \).

The result follows by the soundness of \( \text{Return} \).

Lemma 10 (Return Sound).

If

1. \( p \equiv T(pr, d) \rightarrow^0 U A_1 \rightarrow* U E_1 \rightarrow \ldots \rightarrow U A_n \rightarrow* U E_n \rightarrow \forall A \rightarrow+ \xi \)\text{EE} such that \( \forall A = CE_{\xi}(I) \);
2. \( \forall A \equiv_p CE \) by \( j \);
3. \( (|U A_2|_{al}, |U E_2|_{al}, |U A_{i+1}|_{al}) \in Call \);
4. \( (|U A_{n}|_{al}, |U E_{n}|_{al}, |U A|_{al}) \in Call \); and
5. if \( (|U A|_{al}, |CE|_{al}, j) \in \text{Summary} \), then
   (a) if \( U A_i \equiv_p CE \) by \( j_i \), then \( (|U A|_{al}, |CE|_{al}, j_i) \in \text{Summary} \); and
   (b) if \( T(pr, d) \equiv_p CE \) by \( 1 \) and \( CE \rightarrow \xi \), then \( |\xi|_{al} \in \text{Final} \).

then, after \( \text{Return}(|\forall A|_{al}, |CE|_{al}, j) \),

1. \( (|U A|_{al}, |CE|_{al}, j) \in \text{Summary} \);
2. if \( U A_i \equiv_p CE \) by \( j_i \), then \( (|U A|_{al}, |CE|_{al}, j_i) \in \text{Summary} \); and
3. if \( T(pr, d) \equiv_p CE \) by \( 1 \) and \( CE \rightarrow \xi \), then \( |\xi|_{al} \in \text{Final} \).

Proof. By case analysis on \( \text{Summary} \) and induction on Lemma 11.
Lemma 11 (Link Sound). 

If

1. \( p \equiv \tilde{I}(pr, \tilde{d}) \xrightarrow{0} \tilde{u}_A^1 \xrightarrow{\oplus} \tilde{u}_E^1 \xrightarrow{\ldots} \tilde{u}_A^n \xrightarrow{\oplus} \tilde{u}_E^n \xrightarrow{\oplus} \tilde{u}_A \xrightarrow{\oplus} \tilde{C} \tilde{E} \tilde{E} \) such that \( \tilde{u}_A = CE_{\tilde{u}_E}(j) \);
2. \( \tilde{u}_A \equiv_p \tilde{C} \tilde{E} \tilde{E} \) by \( j \);
3. \( (|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al+1}) \in Call; \)
4. \( (|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al}) \in Call; \) and
5. \( (|\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}, j) \in Summary. \)

then, after \( Link(|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}, j) \),

1. if \( CA(|\tilde{u}_E|_{al}, j) = k \), then preconditions for \( Return(|\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}, CP(|\tilde{u}_A|_{al}, k)) \) are met and its postconditions hold; and
2. if \( CA(|\tilde{u}_E|_{al}, j) = clam \), then preconditions for \( Update(|\tilde{u}_A|_{al}, |\tilde{u}_A|_{al}, |\tilde{u}_E|_{al})|\tilde{C} \tilde{E} \tilde{E}|_{al} j \) are met and its postconditions hold.

Proof. By cases on \( CA(|\tilde{u}_E|_{al}, j) \), induction on Lemma 10, and Lemma 12.

Lemma 12 (Update Sound).

If

1. \( p \equiv \tilde{I}(pr, \tilde{d}) \xrightarrow{0} \tilde{u}_A^1 \xrightarrow{\oplus} \tilde{u}_E^1 \xrightarrow{\ldots} \tilde{u}_A^n \xrightarrow{\oplus} \tilde{u}_E^n \xrightarrow{\oplus} \tilde{u}_A \xrightarrow{\oplus} \tilde{u}_E \)
2. \( \tilde{u}_A \equiv_p \tilde{C} \tilde{E} \tilde{E} \) by \( j \);
3. \( (|\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}, j) \in Summary; \)
4. \( (|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al+1}) \in Call; \)
5. \( (|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al}) \in Call; \) and
6. \( CA(|\tilde{u}_E|_{al}, j) = clam \)

then, after \( Link(|\tilde{u}_A|_{al}, |\tilde{u}_E|_{al}, |\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}, j) \), the postconditions of \( Propagate(|\tilde{u}_A|_{al}, |\tilde{C} \tilde{E} \tilde{E}|_{al}) \) hold, where \( \tilde{C} \tilde{E} \tilde{E} \xrightarrow{\oplus} \tilde{C} \).

Proof. By Lemma 4, Lemma 3, and the definition of \( CE \).

Lemma 13 (Final Sound).

If \( p \equiv \tilde{I}(pr, \tilde{d}) \xrightarrow{\oplus} \tilde{C} \tilde{E} \tilde{E} \) such that \( \tilde{I}(pr, \tilde{d}) \equiv_p \tilde{C} \tilde{E} \tilde{E} \) by \( 1 \), then, after \( Final(|\tilde{C} \tilde{E} \tilde{E}|_{al}) \), \( |\tilde{C}|_{al} \in Final \), where \( \tilde{C} \tilde{E} \tilde{E} \xrightarrow{\oplus} \tilde{C} \).

Proof. By Lemma 3.

7 Summarization Soundness

Theorem 3 (Summarization Completeness).

After summarization,

1. if \( (\tilde{u}_A, \tilde{C}) \in Seen \), then there exists \( p \equiv \tilde{I}(pr, \tilde{d}) \xrightarrow{\ast} \tilde{u}_A \xrightarrow{\ast} \tilde{C} \) such that
   \( \tilde{u}_A = |\tilde{u}_A|_{al}, \tilde{C} = |\tilde{C}|_{al} \), and \( \tilde{u}_A = CE_{\tilde{C}}(j) \);
2. if \((\bar{u}, \bar{c}, \bar{e}, n) \in \text{Summary}\) then there exists \(p \equiv \tilde{I}(pr, \bar{d}) \rightsquigarrow^* \bar{u}_A \rightsquigarrow^+ \bar{c}, \bar{e}\) such that \(\bar{u} = |\bar{u}|_al\), \(\bar{c}, \bar{e} = |\bar{c}, \bar{e}|_al\), and \(\bar{u} \equiv_p \bar{c}, \bar{e}\) by \(n\); and

3. if \(\xi \in \text{Final}\), then there exists \(p \equiv \tilde{I}(pr, \bar{d}) \rightsquigarrow^+ \xi\) such that \(\xi = |\xi|_al\) and \(\xi\) is a final state.

**Proof.** By induction on the number of iterations \(n\) through the loop.

Base case \(n = 0\):

At summarization commencement, \((\tilde{I}(pr, \cdot) \tilde{I}(pr, \cdot)) \in \text{Seen}\) and \(\tilde{I}(pr, \bar{d}) \rightsquigarrow \tilde{I}(pr, \bar{d})\).

Inductive case \(n = i\):

Each iteration commences by considering \((\bar{u}, \xi)\) such that there is a path \(p \equiv \tilde{I}(pr, \bar{d}) \rightsquigarrow^* \bar{u}_A \rightsquigarrow^* \xi\) such that \(\bar{u}_A = |\bar{u}|_al\) and \(\xi = |\xi|_al\).

By cases on \(\xi\).

1. Case \(\xi = \bar{u}:\)

   By Lemma 7, there exists \(\xi\) such that \(\xi \rightsquigarrow \xi\) and \(|\xi|_al = \xi\). Then there exists path \(\tilde{I}(pr, \bar{d}) \rightsquigarrow^* \bar{u}_A \rightsquigarrow^* \xi \rightsquigarrow \xi\).

2. Case \(\xi = \bar{v}\):

   By Lemma 7, for each \(\xi' \in \text{succ}(\xi)\), there exists \(\xi'\) such that \(\xi \rightsquigarrow \xi'\) and \(|\xi'|_al = \xi'\). Then there exists path \(\tilde{I}(pr, \bar{d}) \rightsquigarrow^* \bar{u}_A \rightsquigarrow^* \xi \rightsquigarrow \xi'\) and the preconditions for \(\text{Propagate}(\xi', \xi')\) are met. Suppose \((\xi', \bar{c}, \bar{e}, j) \in \text{Summary}\). By Lemma 9, there exists path \(\tilde{I}(pr, \bar{d}) \rightsquigarrow^* \bar{u}_A \rightsquigarrow^* \xi \rightsquigarrow \xi' \rightsquigarrow^+ \bar{c}, \bar{e}\) such that \(|\bar{c}, \bar{e}|_al = \bar{c}, \bar{e}\) and \(\xi' \equiv_p \bar{c}, \bar{e}\) by \(j\). With \((\bar{u}, \xi, \xi') \in \text{Call}\), the preconditions for \(\text{Link}(\bar{u}, \xi, \xi', \bar{c}, \bar{e}, j)\) are met and its postconditions hold.

3. Case \(\xi = \bar{e}\):

   By definition, \(\bar{u}_A \equiv_p \xi\) by \(CP(\bar{u}_A, CV(\xi))\). Then the preconditions for \(\text{Return}(\bar{u}_A, \xi, CP(\bar{u}_A, CV(\xi)))\) are met and its postconditions hold.

**Lemma 14 (Return Complete).**

If

1. there exists \(p \equiv \tilde{I}(pr, \bar{d}) \rightsquigarrow 0 \bar{u}_A_1 \rightsquigarrow^+ \bar{u}_E_1 \rightsquigarrow \ldots \rightsquigarrow \bar{u}_A_n \rightsquigarrow^+ \bar{u}_E_n \rightsquigarrow \bar{u}_A \rightsquigarrow^+ \bar{c}, \bar{e}\) such that \(\bar{u}_A = CE_{\bar{u}_E}(\cdot)\);
2. \(\bar{u}_A \equiv_p \bar{c}, \bar{e}\) by \(j\);
3. \(|\bar{u}_A_1|_al, |\bar{u}_E_1|_al, |\bar{u}_A_{i+1}|_al\) \in \text{Call};
4. \(|\bar{u}_A_n|_al, |\bar{u}_E_n|_al, |\bar{u}_A|_al\) \in \text{Call}; and

then, after \(\text{Return}(|\bar{u}_A|_al, |\bar{c}, \bar{e}|_al, j)\),

1. if \(|\bar{u}_A|_al, |\bar{c}, \bar{e}|_al, ji\) \in \text{Summary}, then there exists path with \(\bar{u}_A \equiv_p \bar{c}, \bar{e}\) by \(ji\); and
2. if \(|\xi|_al \in \text{Final}\), then there exists path with \(\tilde{I}(pr, \bar{d}) \equiv_p \bar{c}, \bar{e}\) by \(1\) and \(\bar{c}, \bar{e} \rightsquigarrow \xi\).

**Proof.** By Lemma 5 and Lemma 3.

**Lemma 15 (Link Complete).**

If
1. there exists path \( p \equiv \hat{I}(pr, \hat{d}) \xrightarrow{*} \hat{U}A \xrightarrow{+} \hat{U}E \xrightarrow{*} \hat{U}A^* \xrightarrow{+} \text{CEE} \) such that \( \hat{U}A_i = CE_{\hat{U}E}(j) \);
2. \( \hat{U}A \equiv \hat{A} \text{CEE by } j \);
3. \( (|\hat{U}A_i|_{at}, |\hat{U}E_i|_{at}, |\hat{U}A_{i+1}|_{at}) \in \text{Call} \);
4. \( (|\hat{U}A_n|_{at}, |\hat{U}E_n|_{at}, |\hat{U}A|_{at}) \in \text{Call} \); and
5. \( (|\hat{U}A|_{at}, |\text{CEE}|_{at}, j) \in \text{Summary} \).

then, after \( \text{Link}(|\hat{U}A_n|_{at}, |\hat{U}E_n|_{at}, |\hat{U}A|_{at}, |\text{CEE}|_{at}, j) \),

1. if \( \text{CA}(\hat{U}E_n|_{at}, j) = k \), then preconditions for \( \text{Return}(\hat{U}A_n|_{at}, |\text{CEE}|_{at}, CP(|\hat{U}A_n|_{at}, k)) \) are met and its postconditions hold; and
2. if \( \text{CA}(\hat{U}E_n|_{at}, j) = \text{clam} \), then preconditions for \( \text{Update}(\hat{U}A_n|_{at}, |\hat{U}A|_{at}, |\hat{U}E_n|_{at})|\text{CEE}|_{at}) \) are met and its postconditions hold.

Proof. By cases on \( \text{CA}(|\hat{U}E_n|_{at}, j) \), induction on Lemma 14, and Lemma 16.

Lemma 16 (Update Complete).
If there exists path \( p \equiv \hat{I}(pr, \hat{d}) \xrightarrow{*} \hat{U}A \xrightarrow{+} \hat{U}E \xrightarrow{*} \hat{U}A^* \xrightarrow{+} \text{CEE} \) such that \( \hat{U}A^* \equiv_p \hat{A} \text{CEE by } 1 \) and \( \text{CA}(\hat{U}E, j) = \text{clam} \), then, after \( \text{Update}(\hat{U}A, \hat{U}A^*, \hat{U}E)|\text{CEE}|_{at} \) such that \( |\hat{U}A|_{at} = \hat{U}A, |\hat{U}E|_{at} = \hat{U}E, |\hat{U}A^*|_{at} = \hat{U}A^* \), and \( |\text{CEE}|_{at} = \text{CEE} \) \( (\hat{U}A, \hat{E}, \hat{C}) \in \text{Seen and there exists } p' \equiv p \xrightarrow{\hat{c}} \hat{c} \) such that \( \hat{c} = |\hat{c}|_{at} \).


Lemma 17 (Final Complete).
If, for \( \text{CEE} \), there exists path \( p \equiv \hat{I}(pr, \hat{d}) \xrightarrow{+} \text{CEE} \) such that \( \hat{I}(pr, \hat{d}) \equiv_p \text{CEE by } 1 \) and \( |\text{CEE}|_{at} = \text{CEE} \), then, after \( \text{Final}(\text{CEE}) \), \( \hat{c} \in \text{Final and } \hat{I}(pr, \hat{d}) \xrightarrow{+}\text{CEE} \xrightarrow{\hat{c}} \hat{c} \) where \( |\hat{c}|_{at} = \hat{c} \).

Proof. By Lemma 3.