A Posteriori Environment Analysis via Pushdown $\Delta$CFA
Technical Report

Kimball Germane  Matthew Might
University of Utah

\begin{figure}[h]
\centering
\begin{align*}
\zeta \in \text{State} &= \text{Eval} + \text{Apply} \\
\xi \in \text{Eval} &= \text{Call} \times \text{BEnv} \times \text{VEnv} \times \text{Log} \times \text{Time} \\
\kappa \in \text{Apply} &= \text{Proc} \times \overrightarrow{\text{D}} \times \text{VEnv} \times \text{Log} \times \text{Time} \\
\overline{\psi} \in \text{VEnv} &= \text{Var} \times \text{Time} \rightarrow \overrightarrow{\text{D}} \\
\delta \in \text{Log} &= \text{Time} \rightarrow \mathcal{F} \\
\overrightarrow{\delta}, \overline{\psi} \in \overrightarrow{\text{D}} &= \text{Proc} \\
\text{proc} \in \text{Proc} &= \text{Clos} + \{\text{halt}\} \\
\overline{\text{ clos}} \in \text{Clos} &= \text{Lam} \times \text{BEnv} \times \text{Time} \\
\end{align*}
\caption{Log state space}
\end{figure}

1. Introduction
This brief technical report contains theorems of correctness for the approach described in “A Posteriori Environment Analysis via Pushdown $\Delta$CFA”. Correctness is grounded in the log semantics of $\Delta$CFA which we now present.

2. Log Semantics
The log semantics augments the standard semantics in two ways: First, a closure is the triple $(\lambda m, \beta, t)$ instead of the pair $(\lambda m, \beta)$ where $t$ is the timestamp of its birth. Second, each state acquires a log $\delta$ component, a map from times to delta frame strings. The effect of these additions can be seen in the log state space, presented in Figure 1: the Log domain was added and each other barred domain was altered to accommodate them. The log machine relation $\Rightarrow$ augments the transitions of the standard semantics as seen in Figure 2.

The bulk of the work of the log semantics is in keeping the log up-to-date, which requires calculating the delta frame string on each transition and recording it in the log. For $\text{Apply} \rightarrow \text{Eval}$ transitions, the delta frame string is merely the single frame push $(\delta')$. For $\text{Eval} \rightarrow \text{Apply}$ transitions, the birth time of the continuation $\overline{\tau}$ is used to determine which frames to pop. (The net operation $\overline{\tau} \in \text{Eval}$ ensures that the frames of tail calls, which pass their continuations unmodified, are popped only once.)

$\overline{A}$ alters $A$ to take an extra parameter, the timestamp $t$ of the transition source state.

3. Delta Frame String Recovery
The delta frame string of an $\text{Eval} \rightarrow \text{Apply}$ step is derived from the delta frame string since the birth of the call’s continuation. The log semantics obtains this birth time by consulting the closure’s timestamp. However, we can obtain the equivalent birth state from the path’s decomposition.

3.1 Continuation Birth Time Recovery
Given a path $\overline{P} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ for which we want the birth state $\overline{E}_{\text{birth}}$ of $\overline{I}$'s continuation $\overline{\tau}$, we define $\text{btime}$ over $\overline{P}$ to produce $\overline{P}_{\text{birth}} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{E}_{\text{birth}}$.

Definition 1 (Continuation Birth Time). If $\overline{P} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$, then $\text{btime}(\overline{P})$ is

1. $\overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ if $\overline{P} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$,
2. $\overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ if $\overline{P} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ where $\overline{U_{A}} \in CE^*(\overline{E})$; and
3. $\overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ if $\overline{P} \equiv \overline{I} (pr, \overline{d}) \Rightarrow \overline{\epsilon}$ where $\overline{U_{A}} \in CE^*(\overline{E})$.
Case 1 reflects that an `EvalInner` state has a λ-term `clam` for its continuation expression meaning that its continuation is born in that state, at that time. Cases 2 and 3 reflect that an `EvalExit` state has a reference `k` for its continuation. By Theorem 2, the path can be decomposed into a form required by case 2 or case 3. In case 2, `π` was born at inner call `UEI` and passed in an arbitrary number of tail calls before the tail or continuation call of `UEI`. Case 3 is like case 2 except that `π` is `halt`.

The function `btime` is correct if the timestamp on the terminal state of the path it produces is the same as the birth time of the continuation of the terminal state of the path it’s given. We define some notation before we express it formally.

**Definition 2.** Let `qcall = q` for call = `(f e` `q)` or call = `(q e`) - `eval`.

**Definition 3** (Continuation Birth Timestamp).

\[ btime((lam, \beta, t)) = t \]

**Definition 4** (Continuation Procedure).

\[ CP(UA) = \epsilon \text{ where } UA = \langle \text{prcmd}, (\delta_1, \ldots, \delta_n), \delta, \delta, t \rangle \]

\[ CP(\epsilon) = \epsilon \text{ where } \epsilon = (\epsilon, \delta, \epsilon, \delta, t) \]

**Definition 5** (Invocation Continuation).

\[ IC(E) = \epsilon(k, \beta(k)) \text{ where } E = (call, \beta, \epsilon, \delta, t) \]

\[ IC(\epsilon) = \epsilon(k, \beta(k)) \text{ where } \epsilon = (\epsilon, \delta, \epsilon, \delta, t) \]

where \( k \in dom(\beta) \)

**Lemma 1.** If \( UA \Rightarrow^* \xi \text{ where } UA \in CE(\xi) \), then \( IC(\xi) = CP(UA) \).

**Proof.** By induction on \( CE(\cdot) \).

1. Case \( UA \Rightarrow^0 UA \): Doesn’t apply.

2. Case \( UA \Rightarrow^+ \xi \Rightarrow \xi \text{ where } \xi' \notin UEval \cup CEvalExit \):
   - (a) Case \( \xi' = \epsilon \): By \( \epsilon \Rightarrow \epsilon \), the continuation is bound.
   - (b) Case \( \xi' = \epsilon \): By \( \epsilon \Rightarrow \epsilon \), the environment is restored.
   - (c) Case \( \xi' = CEI \):
     - By \( CEI \Rightarrow CEI \), the environment is not binding.

3. Case \( UA \Rightarrow^+ \xi \Rightarrow UA_0 \Rightarrow CE \Rightarrow \xi \text{ where } UA_0 = CE(\xi) \), \( \xi' = \epsilon \): By \( \epsilon \Rightarrow \epsilon \), the environment is restored.

By Lemma 2 and \( CE \Rightarrow \epsilon \), \( CP(\epsilon) = \epsilon(k, \beta, \epsilon, t) \). Then \( IC(\epsilon) = CP(UA) \).

\[ \square \]

**Lemma 2.** If \( UA \Rightarrow^+ CE \text{ where } UA \in CE^*(CE) \text{ and } CE = (k e_1), \beta, \epsilon, \delta, t) \), then \( \lambda k (k, \beta, \epsilon, t) = CP(UA) \).

**Proof.** By induction on \( CE^*(\cdot) \).

1. Case \( UA = CE(CE) \):
   - By Lemma 1, \( IC(CE) = CP(UA) \). By definition, \( \lambda k (k, \beta, \epsilon, t) = \epsilon(k, \beta(k)) \). By definition, \( IC(CE) = \epsilon(k, \beta(k)) \). Then \( \lambda k (k, \beta, \epsilon, t) = CP(UA) \).

2. Cases \( UA \Rightarrow^+ UEI \Rightarrow UA_0 \Rightarrow CE \text{ where } UA \in CE(UEI) \) and \( UA_0 \in CE^*(UEI) \):
   - By Lemma 1 and above reasoning, \( IC(UEI) = CP(UA) \).
   - By \( UEI \Rightarrow UA_0 \), \( CP(UA) = IC(UEI) \). By induction, \( \lambda k (k, \beta, \epsilon, t) = CP(UA) \). Then \( \lambda k (k, \beta, \epsilon, t) = CP(UA) \).

**Theorem 1** (Continuation Birth Time Correctness). If \( \bar{P} \equiv \bar{P}(pr, d) \Rightarrow^+ E \) where \( E = (call, \beta, \epsilon, t) \) and \( P' = btime(\bar{P}) \), then \( btime(A(qcall, \beta, \epsilon, t)) = t_0 \).

**Proof.** Let \( E = (call, \beta, \epsilon, t) \). Consider cases of \( btime(\cdot) \).

1. Case \( \bar{P} \equiv \bar{P}(pr, d) \Rightarrow^+ E \):
   - By assumption, \( qcall = \text{clam} \Rightarrow A(qcall, \beta, \epsilon, t) = \text{clam} \).
   - By definition, \( btime(clam, \beta, t) = t \). By definition, \( btime(\bar{P}) = t \).

2. Case \( \bar{P} \equiv \bar{P}(pr, d) \Rightarrow^+ E \Rightarrow CE \):
   - By assumption, \( qcall = \beta \) so, by Lemma 2, \( \lambda k (k, \beta, t) = CP(UA) \).
   - By definition, \( btime(\bar{P}) = t' \Rightarrow \bar{T}(pr, d) \Rightarrow^0 E \).
   - By \( \bar{T} \Rightarrow \bar{U} \), \( CP(\bar{T}) = \epsilon \).

3. Case \( \bar{P} \equiv \bar{T}(pr, d) \Rightarrow^0 E \Rightarrow CE \):
   - By assumption, \( qcall = \beta \) so, by Lemma 2, \( \lambda k (k, \beta, t) = CP(UA) \).
   - By definition, \( btime(\bar{P}) = t' \Rightarrow \bar{T}(pr, d) \Rightarrow^0 E \).

3.2 Log Recovery

We now show that the ability to recover continuation birth times allows us to recover the log itself. We recover logs inductively on an evaluation path: to recover the log \( \delta' \) of \( \xi \), we first recover the log of the predecessor \( \xi' \) of \( \xi \); the log of \( \bar{T}(pr, d) \) is known by definition.

**Definition 6** (Log Recovery). \( Rec(\bar{P}) = \perp[t_0 \Rightarrow \epsilon] \).

- If \( \bar{P} \equiv \bar{T}(pr, d) \Rightarrow^0 \xi \):
  - \( Rec(\bar{P}) = (\lambda a, \delta[t]) \Rightarrow \epsilon \).
  - If \( \bar{P} \equiv \bar{T}(pr, d) \Rightarrow^* \xi \) and \( \bar{P'} \equiv \bar{P} \Rightarrow \xi' \) where \( \delta = Rec(\bar{P}) \):
    - \( p_a = \{ \} \).
    - If \( \xi = \epsilon \) then \( \delta = \epsilon \).

Log recovery is correct if the log we construct is the same as the log of the state. We express this condition in the following theorem.

**Theorem 2** (Genuine Log Recovery). For all \( n \in \mathbb{N} \), if \( \bar{P} \equiv \bar{T}(pr, d) \Rightarrow^0 \xi \), then \( Rec(\bar{P}) = \delta, \xi \).

The proof proceeds inductively over path length and by cases on the terminal state at each inductive step. Unsurprisingly, Theorem 2 is fundamental to the \( E \) case.

**Proof.** By induction on \( n \).

1. Base case \( \bar{P} \equiv \bar{T}(pr, d) \Rightarrow^0 \xi \):
   - \( \delta = \perp[t_0 \Rightarrow \epsilon] \).

2. Inductive case \( P \equiv P \Rightarrow \xi \) and \( P' \equiv P \Rightarrow \xi' \):
   - Let \( \delta = Rec(P') \). By induction, \( \delta = \xi' \).

By cases on \( \xi' \):

(a) Case \( \xi' = \Lambda \):
   - By definition, \( Rec(P') = (\lambda \delta[t] + p_a)[\xi' \Rightarrow \epsilon] \) where \( p_a = \{ \} \).

\[ \square \]
Lemma 3. If \( A \Rightarrow E \), then \( A \rightarrow E \).

Lemma 4. If \( UA \Rightarrow E \Rightarrow CA_1 \Rightarrow E_1 \Rightarrow \cdots \Rightarrow CA_n \Rightarrow E_n \), then 
\[
[\langle t_{UA}, t_{E_n} \rangle] = \left[ \left( \gamma_1 \right)_{E_1} \cdots \left( \gamma_n \right)_{E_n} \right] \text{ where } \gamma_i \equiv (L(t_{i}), \omega, \omega) \text{ and } CA_i \equiv (L(\gamma_i), \omega, \omega) \text{ for } 1 \leq i \leq n.
\]

Proof. By induction on \( n \).

4. Binding- and Birth-State Resolution Correctness

Lemma 6 (BirthBP Correctness). If

1. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{UA} \) where \( \tilde{UA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \), or
2. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{CA} \) where \( \tilde{CA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \),

then \( \text{BirthBP}(\tilde{P}, u) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{E} \) such that \( \text{blime}(\tilde{d}_i) = t_{\tilde{E}} \) where \( i = \text{BP}(\tilde{P}, u) \).

Proof. By induction on \( \text{BirthIP}(\tilde{P}, i) \).

Lemma 7 (BirthIP Correctness). If

1. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{UA} \) where \( \tilde{UA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \), or
2. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{CA} \) where \( \tilde{CA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \),

then \( \text{BirthIP}(\tilde{P}, i) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{E} \) such that \( \text{blime}(\tilde{d}_i) = t_{\tilde{E}} \).

Proof. If \( n = 0 \), then it is vacuously true. If \( n > 0 \), then \( \tilde{P} \equiv \tilde{P'} \Rightarrow \tilde{U} \) where \( \tilde{U} = ((f e_1 \ldots e_n q), \tilde{\beta}, \tilde{\nu}, \tilde{\delta}, t) \). If \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^* \tilde{U} \) then, by \( \tilde{U} \Rightarrow \tilde{U}, \tilde{d}_1 = \tilde{A}(e_i, \tilde{\beta}, \tilde{\nu}, t) \) where \( \tilde{U} = ((f e_1 \ldots e_n q), \tilde{\beta}, \tilde{\nu}, \tilde{\delta}, t) \).

Lemma 8 (BirthIE Correctness). If

1. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{UA} \) where \( \tilde{UA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \), or
2. \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{CA} \) where \( \tilde{CA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \),

then \( \text{BirthIP}(\tilde{P}, i) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{E} \) such that \( \text{blime}(\tilde{A}(e_i, \tilde{\beta}, \tilde{\nu}, t)) = t_{\tilde{E}} \) for \( 1 \leq i \leq n \).

Proof. If \( n = 0 \), then it is vacuously true. If \( n > 0 \), then, by definition, \( \text{BirthIE}(\tilde{P}, i) = \text{Birth}(\tilde{P}, e_i) \). The result follows by induction on \( \text{Birth}(\tilde{P}, e_i) \).

Lemma 9 (BirthOP Correctness). If

\( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{UA} \) where \( \tilde{UA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \), then \( \text{BirthOP}(\tilde{P}) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{E} \) such that \( \text{blime}(\tilde{proc}) = t_{\tilde{E}} \).

Proof. If \( n = 0 \), then it is vacuously true. If \( n > 0 \), then \( \tilde{P} \equiv \tilde{P'} \Rightarrow \tilde{U} \) where \( \tilde{U} = ((f e_1 \ldots e_n q), \tilde{\beta}, \tilde{\nu}, \tilde{\delta}, t) \). Then the result follows by induction on \( \text{BirthOP}(\tilde{P'}) \).

Lemma 10 (BirthOE Correctness). If \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{UA} \) where \( \tilde{UA} = (\tilde{proc}, (d_1, \ldots, d_n), \tilde{v}, \tilde{\delta}, t) \), then \( \text{BirthIP}(\tilde{P}, i) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{E} \) such that \( \text{blime}(\tilde{A}(f, \tilde{\beta}, \tilde{v}, t)) = t_{\tilde{E}} \).

Proof. By definition, \( \text{BirthOE}(\tilde{P}) = \text{Birth}(\tilde{P}, f) \). The result follows by induction on \( \text{Birth}(\tilde{P}, f) \).

Lemma 11 (Birth Correctness). If \( \tilde{P} \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{E} \) where \( \tilde{E} = (\text{call}, \beta, \tilde{\nu}, \tilde{\delta}, t) \), and \( \text{Birth}(\tilde{P}, h) = \tilde{P}_* \equiv \tilde{T}(pr, d) \Rightarrow^+ \tilde{E} \), then \( \text{blime}(\tilde{A}(h, \beta, \tilde{v}, t)) = t_{\tilde{E}} \).

Proof. By cases on \( h \).
• Case $h = ulam$: By definition, $Birth(\bar{P}, ulam) = \bar{P}$. Then $t_{\bar{P}} = t_\bar{P} = t$. By definition, $\bar{A}(ulam, \beta, \bar{w}, t) = (ulam, \beta, t)$. By definition, $\text{btame}(ulam, \beta, t) = t$. Then $\text{btame}(\bar{A}(ulam, \beta, \bar{w}, t)) = t_{\bar{P}}$.

• Case $h = u$: By definition, $Birth(\bar{P}, u) = BirthBP(\bar{P}_2, u)$ where $P_1 = \text{Bind}(\bar{P}, u)$ such that $P_1 = \bar{P}_2 \Rightarrow \text{t}_{\bar{P}}$ and $P_2 = \bar{P}_1 \Rightarrow \bar{X}_1$. By induction on $\text{Bind}(\bar{P}, u)$, $\beta(u) = t_{\bar{P}}$. Then $\bar{w}(u, \beta(u)) = \bar{w}_{\bar{P}}(u, t_{\bar{P}})$. By $\bar{X}_1 \Rightarrow \bar{E}_1$, $\bar{w}_{\bar{P}}(u, t_{\bar{P}}) = \pi_i(\bar{d})$ where $\bar{X}_1 = (\text{prd}, \bar{d}, \bar{w}_1, \delta_1, t_1)$ and $i = BP_{prd}(u)$.

By induction, $BirthBP(\bar{P}_2, u) = \bar{P}_2 \equiv T(pr, \bar{d}) \Rightarrow \bar{E}_2$ such that $\text{btame}(\bar{X}_i(\bar{d})) = t_{\bar{P}}$. As $\pi_i(\bar{d}) = \bar{w}_{\bar{P}}(u, t_{\bar{P}}) = \bar{w}(u, \beta(u)) = \bar{A}(u, \beta, \bar{w}, t)$, $\text{btame}(\bar{A}(u, \beta, \bar{w}, t)) = t_{\bar{P}}$ and we have our result.

**Lemma 12 (Bind Correctness).** If $\bar{P}' \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E}$ where $E = (\text{call}, \beta, \bar{w}, \delta, t)$ and $\text{Bind}(\bar{P}', u) = \bar{P}_2 \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E}$, then $\beta(u) = t_{\bar{P}}$.

**Proof.** Let $\bar{P} \equiv T(pr, \bar{d}) \Rightarrow^* \bar{X}$ such that $\bar{P}' \equiv \text{logpath} \Rightarrow \bar{X}$. By cases on $\text{Bind}(\bar{P}, u)$.

• Case $u \in B(\bar{P})$: By definition, $\text{Bind}(\bar{P}', u) = \bar{P}'$ with $\bar{E}_2 = \bar{E}$. By $\bar{X} \Rightarrow \bar{E}$, $\beta(u) = t_{\bar{P}}$.

• Case $u \notin B(\bar{P})$: By definition, $\text{Bind}(\bar{P}', u) = \text{Find}(\bar{P}', u)$. By induction, $\text{Find}(\bar{P}', u) = \bar{P}_2 \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E}_2$ where $\beta(u) = t_{\bar{P}}$.

**Lemma 13 (Find Correctness).** If $\bar{P} \equiv T(pr, \bar{d}) \Rightarrow^* \bar{X}$ such that $u \notin B(\bar{P})$, $\bar{P}' \equiv \bar{P} \Rightarrow \bar{E}$ where $E = (\text{call}, \beta, \bar{w}, \delta, t)$, and $\text{Find}(\bar{P}', u) = \bar{P}_2 \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E}$, then $\beta(u) = t_{\bar{P}}$.

**Proof.** By cases on $\text{Find}(\bar{P}, u)$.

1. Case $\bar{P} \equiv T(pr, \bar{d}) \Rightarrow^* \bar{U} \bar{A}$: Let $\bar{U} \bar{A} = (\text{prw}, \bar{d}, \bar{w}, \delta', t')$. By assumption, $u \notin B(\bar{P})$, so $\beta(u) = \beta_{prw}(u)$. By definition, $\text{Find}(\bar{P}', u) = \text{Bind}(BirthOP(\bar{P}'), u)$. By induction, $BirthOP(\bar{P}') = \bar{P}_1 \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E}$ such that $\text{btame}(\bar{W}) = t_{\bar{P}}$. By definition of $\Rightarrow$, $\beta_{prw}(u) = \beta_{\bar{P}_1}(u)$.

The result follows by induction on $\text{Bind}(\bar{P}_1, u)$.

2. Case $\bar{P} \equiv T(pr, \bar{d}) \Rightarrow^* \bar{E} \Rightarrow^* \bar{C} \bar{A}$: Let $\bar{C} \bar{A} = (\tau, \bar{G}, \bar{w}, \delta', t')$.

By assumption, $\bar{E} \Rightarrow \bar{C} \bar{A}$ and $u \notin B(\bar{P})$, so $\beta(u) = \beta_{\bar{C} \bar{A}}(u)$. The result follows by induction on $\text{Bind}(\bar{P}_1, u)$ where $\bar{P}_1 \equiv T(pr, \bar{d}) \Rightarrow \bar{E}_1$. 

□