Technical Report COMP-001-2009
Computing Department
Lancaster University

From stack traces to call-trees: outline of a proof

François Taïani

January 22, 2009

Abstract

This report outlines a proof of the theorem described in COSMOPEN*: Dynamic reverse engineering on a budget, How cheap observation techniques can be used to reconstruct complex multi-level behaviour* [1] on the construction of a minimal call-trees from stack traces. Please refer to original publication for details regarding the context of the work, the notation used, and definitions. We repeat below the pseudo-code of the call-tree construction algorithm, and present some key definitions in more detail. We then outline a proof of the characterisation of the call-tree constructed by the algorithm.

Computing Department
Infolab21, South Drive
Lancaster University
LANCASTER LA1 4WA
United Kingdom
Tl.: +44 (0) 1524 51 03 38
mail: francois.taiani@comp.lancs.ac.uk

1 Introduction

This short document outlines a proof of the theorem described in [1] on the construction of a minimal call-trees from stack traces. Please refer to [1] for details regarding the context of the work, the notation used, and definitions. We repeat below the pseudo-code of the call-tree construction algorithm, and present some key definitions in more detail. We then outline a proof of the characterisation of the call-tree constructed by the algorithm.

2 Call-tree construction algorithm

Algorithm 1 Transforming a stack trace sequence into a call-tree

```
 1: ActiveTreePath ← (); V ← E ← ∅
 2: traceSequence = (trace_1, trace_2, .., trace_N)
 3: for i = 1 to |traceSequence| do
 4:     AddTraceToGraph(traceSequence[i], V, E, ActiveTreePath)
 5: end for

 6: procedure AddTraceToGraph(trace, V, E, ActiveTreePath)
 7:     for j = 1 to min( |ActiveTreePath|, |trace| − 1 ) do
 8:         if symbol(ActiveTreePath[j]) ≠ trace[j] then break
 9:     end for
10:     truncate(ActiveTreePath, j)
11:     previousNode ← last(ActiveTreePath)
12:     for k = j to |trace| do
13:         newNode ← new Node(trace[k])
14:         ActiveTreePath ← ActiveTreePath + (newNode)
15:         V ← V ∪ {newNode}
16:         if previousNode ≠ NIL then E ← E ∪ { (previousNode, newNode) }
17:         previousNode ← newNode
18:     end for
19: end procedure
```

In algorithm 1, time stamps (also known as *sequence numbers*) are allocated by the constructor **new** $Node(\text{symbol})$ (line 13) in the order in which nodes are created. The same timestamp is never allocated more than once.

3 Definitions and theorem

Call tree A *call-tree* is a directed tree (V, E) that is associated with two labelling functions: $symbol$ and $timeStamp$.

$symbol$ associates each node $x \in \mathsf{V}$ of the tree with the name of a function (its *symbol*). Multiple nodes might be associated with the same symbol.

$timeStamp$ associates each node $x \in \mathsf{V}$ with an integer (its *time stamp*). Time stamps are unique (no two nodes have the same time stamp) and the time stamp of a child node is strictly greater than that of its parent:

$$\forall x, y \in \mathsf{V}: \quad timeStamp(x) = timeStamp(y) \;\Rightarrow\; x = y \tag{1}$$

$$\forall x, y \in \mathsf{V}: \quad (x, y) \in \mathsf{E} \;\Rightarrow\; timeStamp(x) < timeStamp(y) \tag{2}$$

Trace compatibility We say that a call-tree (V, E) is *trace-compatible* (compatible for short) with a sequence of observed stack traces $\mathsf{traceSequence} = (\mathsf{trace}_1, \mathsf{trace}_2, .., \mathsf{trace}_N)$ if and only if there exists a surjective

mapping Φ from the frames $\langle \mathsf{trace}_i, j \rangle$ of the observed traces onto the nodes of V, such that Φ fulfils the following properties P0, P1, P2 and P3.

PROPERTY P0: SYMBOL CONSISTENCY Stack frames are mapped onto nodes with the same symbol:

$$\forall\, \mathsf{trace}_i \in \mathsf{traceSequence}, 1 \leq k \leq |\mathsf{trace}_i| : symbol(\Phi(\langle \mathsf{trace}_i, k \rangle)) = \mathsf{trace}_i[k] \tag{3}$$

PROPERTY P1: TRACE INCLUSION Each observed trace is mapped by Φ onto a path of the tree that starts from the tree's root:

$$\begin{aligned} &\forall\, \mathsf{t}_i \in \mathsf{traceSequence} : \\ &\quad \Phi(\langle \mathsf{t}_i, 1 \rangle) \text{ is the root of } (\mathsf{V}, \mathsf{E}) \wedge \\ &\quad \forall\, 1 \leq j < |\mathsf{t}_i| : \Big(\Phi(\langle \mathsf{t}_i, j \rangle),\ \Phi(\langle \mathsf{t}_i, j+1 \rangle) \Big) \in \mathsf{E} \end{aligned} \tag{4}$$

PROPERTY P2: BREAKPOINT DISCRIMINATION The last frame of every stack trace (e.g. the frame that corresponds to a breakpoint activation) is mapped to a node that is distinct from those of the stack traces that precede it:

$$\forall\, \mathsf{t}_i, \mathsf{t}_j \in \mathsf{traceSequence} : i < j \Rightarrow \forall\, 1 \leq k \leq |\mathsf{t}_i| : \Phi(\langle \mathsf{t}_j, |\mathsf{t}_j| \rangle) \neq \Phi(\langle \mathsf{t}_i, k \rangle) \tag{5}$$

PROPERTY P3: ORDER CONSERVATION The sequence numbers of the nodes of the tree reflect the order in which traces are observed. More precisely: if a trace trace_j is observed after another trace trace_i, then the tail of trace_j that does not overlap with trace_i is mapped onto nodes with higher sequence numbers than the nodes of trace_i.

$$\begin{aligned} &\forall\, \mathsf{trace}_i, \mathsf{trace}_j \in \mathsf{traceSequence}, \forall k \in \Big[1, min(|\mathsf{trace}_i|, |\mathsf{trace}_j|)\Big] : \\ &\quad (j > i) \wedge \Phi(\langle \mathsf{trace}_i, k \rangle) \neq \Phi(\langle \mathsf{trace}_j, k \rangle) \Rightarrow \\ &\qquad \forall l \in \Big[k, |\mathsf{trace}_j|\Big], \forall m \in \Big[1, |\mathsf{trace}_i|\Big] : \\ &\qquad\quad timeStamp\Big(\Phi(\langle \mathsf{trace}_j, l \rangle)\Big) > timeStamp\Big(\Phi(\langle \mathsf{trace}_i, m \rangle)\Big) \end{aligned} \tag{6}$$

Homomorphism between call-trees A *call-tree homomorphism* h from a call-tree (V, E) onto another call-tree (W, F) is a graph homomorphism from (V, E) to (W, F) that fulfils the following additional properties:

$$\forall x \in \mathsf{V} : symbol\Big(h(x)\Big) = symbol(x) \tag{7}$$

$$\begin{aligned} \forall r, s \in h[\mathsf{V}] : &\Big(\forall x \in h^{-1}[\{r\}], \forall y \in h^{-1}[\{s\}] : timeStamp(x) \leq timeStamp(y)\Big) \\ &\Rightarrow timeStamp(r) \leq timeStamp(s) \end{aligned} \tag{8}$$

where $f[\mathsf{A}]$ (resp. $f^{-1}[\mathsf{A}]$) denotes the image (resp. the pre-image) of the set A by the function f.
A *call-tree isomorphism* is a bijective call-tree homomorphism. Note that for a call-tree isomorphism (8) is equivalent to

$$\forall x, y \in \mathsf{V} : timeStamp(x) \leq timeStamp(y) \Rightarrow timeStamp\Big(h(x)\Big) \leq timeStamp\Big(h(y)\Big) \tag{9}$$

Theorem 3.1. *The set of call-trees that are trace-compatible with a sequence of observed stack traces is non-empty. The call-tree of this set with the smallest number of nodes is unique (modulo a call-tree isomorphism) and is the call-tree constructed by algorithm 1.*

4 Proof

Let's consider a trace sequence $\mathsf{traceSequence} = (\mathsf{trace}_1, \mathsf{trace}_2, .., \mathsf{trace}_N)$, and let's call $\mathbf{G_0} = (\mathsf{V}_0, \mathsf{E}_0)$ the call-tree constructed by algorithm 1 from this trace sequence.

Lemma 4.1. $\mathbf{G_0}$ *is trace compatible with* $\mathsf{traceSequence}$.

Proof. Let's note $\mathsf{ActiveTreePath}_i$ the value of $\mathsf{ActiveTreePath}$ just after trace_i has been processed by $AddTrace\text{-}ToGraph()$ at line 5. A mapping Φ_0 from the frames of $\mathsf{traceSequence}$ onto $\mathbf{G_0}$ can be defined as:

$$\forall \mathsf{trace}_i \in \mathsf{traceSequence}, \forall j \in \Big[1, |\mathsf{trace}_i|\Big] : \Phi_0\Big(\langle \mathsf{trace}_i, j\rangle\Big) \equiv \mathsf{ActiveTreePath}_i[j] \tag{10}$$

By construction Φ_0 is a function (i.e. it associates each frame $\langle \mathsf{trace_i}, j\rangle$ with exactly one node of $\mathbf{G_0}$), and by construction of algorithm 1, Φ_0 is surjective.
Also because of lines 8, 13, and 14 and the constructor property of the **new** $Node(\mathsf{trace}[k])$, we have

$$symbol\Big(\Phi_0(\langle \mathsf{trace}_i, j\rangle)\Big) = symbol\Big(\mathsf{ActiveTreePath}_i[j]\Big) = \mathsf{trace}_i[j]$$

and consequently Φ_0 fulfils property P0.
Property P1 follows from the creation of new edges at line 16 and the fact that all traces start with the same symbol (the program's entry point): $\forall \mathsf{t}_i \in \mathsf{traceSequence} : \mathsf{t}_i[1] = \mathsf{trace}_1[1]$.
Property P2 follows from the bound $|\mathsf{trace}| - 1$ put on the loop that compares $\mathsf{ActiveTreePath}$ with the current trace at line 7. '-1' forces the last frame of the trace (and hence the symbol corresponding to a breakpoint activation) to be included as a new node in the call-tree.
The proof of property P3 is slightly more technical. In this proof we will use the following sublemma:

Sublemma 4.1.1.

$$\begin{aligned}\forall\, \mathsf{trace}_i \in \mathsf{traceSequence}, \forall k, l \in \Big[1, |\mathsf{trace}_i|\Big] :\\ k < l \Rightarrow timeStamp\Big(\Phi_0(\langle \mathsf{trace}_i, k\rangle)\Big) < timeStamp\Big(\Phi_0(\langle \mathsf{trace}_i, l\rangle)\Big)\end{aligned} \tag{11}$$

Proof. This sublemma directly follows from

$$\begin{aligned}\forall\, i \in \Big[1, |\mathsf{traceSequence}|\Big], \forall k, l \in \Big[1, |\mathsf{traceSequence}[i]|\Big] :\\ k < l \Rightarrow timeStamp\Big(\mathsf{ActiveTreePath}_i[k]\Big) < timeStamp\Big(\mathsf{ActiveTreePath}_i[l]\Big)\end{aligned} \tag{12}$$

which is itself an invariant of the loop at line 3. This invariant can be proved by recursion using the properties of time stamps and node creation. (The details of this short proof are left out for place reason.) □

To prove P3, let's now assume $\Phi_0(\langle \mathsf{trace}_i, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_j, k\rangle)$ for some $i > j$ and $k \in \Big[1, min(|\mathsf{trace}_i|, |\mathsf{trace}_j|)\Big]$.
The proof needs to consider two cases depending whether all traces between i and j are at least of length k or not.

Case 1 $\forall r \in [i, j] : |\mathsf{trace}_r| \geq k$

In this case we can consider the largest $l \in [i, j-1]$ such that $\Phi_0(\langle \mathsf{trace}_l, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_{l+1}, k\rangle)$. Let's note this value l_0:

$$l_0 \equiv max\left(\left\{l \in [i, j-1] \;\middle|\; \Phi_0(\langle \mathsf{trace}_l, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_{l+1}, k\rangle)\right\}\right) \tag{13}$$

By construction of l_0 we have

$$\Phi_0(\langle \mathsf{trace}_{l_0}, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_{l_0+1}, k\rangle) = \Phi_0(\langle \mathsf{trace}_{l_0+2}, k\rangle) = ... = \Phi_0(\langle \mathsf{trace}_j, k\rangle) \tag{14}$$

This implies (by definition of Φ_0 in equation (10))

$$\mathsf{ActiveTreePath}_{l_0}[k] \neq \mathsf{ActiveTreePath}_{l_0+1}[k] \tag{15}$$

This again implies that a new node was constructed for $\mathsf{trace}_{l_0+1}[k]$ at line 13. Because timestamps are allocated in the order in which nodes are created, this node's timestamp is higher than those of any previously created nodes, and in particular than the timestamp of $\mathsf{ActiveTreePath}_i[m] = \Phi_0(\langle \mathsf{trace}_i, m\rangle)$ for all $m \in \Big[1, |\mathsf{trace}_i|\Big]$.

$$\forall m \in \Big[1, |\mathsf{trace}_i|\Big] : timeStamp\Big(\Phi_0(\langle \mathsf{trace}_i, m)\Big) < timeStamp\Big(\Phi_0(\langle \mathsf{trace}_{l_0+1}, k\rangle)\Big) \tag{16}$$

From (14) and (16) we thus get (since $\Phi_0(\langle \mathsf{trace}_{l_0+1}, k\rangle) = \Phi_0(\langle \mathsf{trace}_j, k\rangle)$):

$$\forall m \in \Big[1, |\mathsf{trace}_i|\Big] : timeStamp\Big(\Phi_0(\langle \mathsf{trace}_i, m)\Big) < timeStamp\Big(\Phi_0(\langle \mathsf{trace}_j, k\rangle)\Big) \tag{17}$$

Combined with sublemma 4.1.1, equation (17) gives

$$\begin{aligned} &\forall m \in \Big[1, |\mathsf{trace}_i|\Big], l \in \Big[k, |\mathsf{trace}_j|\Big] : \\ &\quad timeStamp\Big(\Phi_0(\langle \mathsf{trace}_i, m\rangle)\Big) < timeStamp\Big(\Phi_0(\langle \mathsf{trace}_j, l\rangle)\Big) \end{aligned} \tag{18}$$

which concludes the proof of P3 in this case.

Case 2 $\exists r \in [i, j] : |\mathsf{trace}_r| < k$

In that case we can consider

$$r_0 \equiv max\left(\left\{r \in [i+1, j-1] \;\middle|\; |\mathsf{trace}_r| < k\right\}\right) \tag{19}$$

By construction, note that we have:

$$\forall r \in [r_0 + 1, j] : |\mathsf{trace}_r| \geq k \tag{20}$$

Two subcases follow.

Case 2.1

If $\exists l \in [r_0 + 1, j-1] : \Phi_0(\langle \mathsf{trace}_l, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_{l+1}, k\rangle)$ then we can define

$$l_0 \equiv max\left(\left\{l \in [r_0 + 1, j-1] \;\middle|\; \Phi_0(\langle \mathsf{trace}_l, k\rangle) \neq \Phi_0(\langle \mathsf{trace}_{l+1}, k\rangle)\right\}\right) \tag{21}$$

and we are back to case 1.

Case 2.2

If $\forall l \in [r_0 + 1, j-1] : \Phi_0(\langle \mathsf{trace}_l, k\rangle) = \Phi_0(\langle \mathsf{trace}_{l+1}, k\rangle)$ then we note that because $|\mathsf{trace}_{r_0}| < k$ and $|\mathsf{trace}_{r_0+1}| \geq k$ then a new node was constructed for $\mathsf{trace}_{r_0+1}[k]$ at line 13. Because $\Phi_0(\langle \mathsf{trace}_{r_0+1}, k\rangle) = \Phi_0(\langle \mathsf{trace}_j, k\rangle)$, we are again back to case 1.

This concludes the proof of P3, and more generally the proof of lemma 4.1. □

Lemma 4.2. *If a call-tree* $\mathbf{H} = (\mathsf{W}, \mathsf{F})$ *is trace compatible with* traceSequence *through a mapping* $\Phi_{\mathbf{H}}$*, then*

$$\forall w \in \mathsf{W} : \Phi_0\left[\Phi_{\mathbf{H}}^{-1}\Big[\{w\}\Big]\right] \textit{ is a singleton}$$

where $f[\mathsf{A}]$ *(resp.* $f^{-1}[\mathsf{A}]$*) denotes the image (resp. the pre-image) of the set* A *by the function* f*.*

Proof. Let's first note that because $\Phi_{\mathbf{H}}$ is by definition surjective, $\Phi_{\mathbf{H}}^{-1}[\{x\}]$ is non-empty, and $\Phi_0[\Phi_{\mathbf{H}}^{-1}[\{x\}]]$ therefore contains at least one element.

Let's now consider

$$x, y \in \Phi_0\left[\Phi_{\mathbf{H}}^{-1}\left[\{w\}\right]\right] \tag{22}$$

Our goal is to prove that $x = y$

(22) means that there exists $\mathsf{trace}_i, \mathsf{trace}_j \in \mathsf{traceSequence}$, $k \in [1, |\mathsf{trace}_i|]$ and $l \in [1, |\mathsf{trace}_j|]$ such that

$$\begin{aligned} \langle \mathsf{trace}_i, k\rangle, \langle \mathsf{trace}_j, l\rangle &\in \Phi_{\mathbf{H}}^{-1}\left[\{w\}\right] && (23)\\ \Phi_0(\langle \mathsf{trace}_i, k\rangle) &= x && (24)\\ \Phi_0(\langle \mathsf{trace}_j, l\rangle) &= y && (25) \end{aligned}$$

(23) means in turn that:

$$\Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, k\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{trace}_j, l\rangle) = w \tag{26}$$

Because $\mathbf{H}$ is a tree, Property P1 of $\mathbf{H}$ and (26) imply that $l = k$ and that

$$\forall m \in [1, k] : \Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, m\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{trace}_j, m\rangle) \tag{27}$$

If $i = j$, (24) and (25), and $l = k$ imply trivially that $x = y$. Without loss of generality, we will therefore assume in the remaining that $i < j$.

We will now prove by contradiction that

$$\begin{aligned} \forall r \in [i+1, j] : &\quad |\mathsf{trace}_r| > k && (28)\\ \forall r \in [i, j], \forall m \in [1, k] : &\quad \Phi_{\mathbf{H}}(\langle \mathsf{trace}_r, m\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, m\rangle) && (29) \end{aligned}$$

Let's first consider (28) and assume there exists $r \in [i+1, j]$ such that $|\mathsf{trace}_r| \leq k$. Let's note $m_r \equiv |\mathsf{trace}_r|$. By applying property P2 of $\mathbf{H}$ on trace_i and trace_r we get

$$\Phi_{\mathbf{H}}(\langle \mathsf{trace}_r, m_r\rangle) \neq \Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, m_r\rangle) \tag{30}$$

Because of (27) and $m_r \equiv |\mathsf{trace}_r| \leq k$ we also have

$$\Phi_{\mathbf{H}}(\langle \mathsf{trace}_r, m_r\rangle) \neq \Phi_{\mathbf{H}}(\langle \mathsf{trace}_j, m_r\rangle) \tag{31}$$

From this we also conclude that $r < j$. Since $i < r < j$, we can apply the property P3 of $\mathbf{H}$ twice, first on trace_i and trace_r and then on trace_r and trace_j, and we eventually get:

$$timeStamp\Big(\Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, m_r\rangle)\Big) < timeStamp\Big(\Phi_{\mathbf{H}}(\langle \mathsf{trace}_r, m_r\rangle)\Big) < timeStamp\Big(\Phi_{\mathbf{H}}(\langle \mathsf{trace}_j, m_r\rangle)\Big) \tag{32}$$

(32) implies that $\Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, m_r\rangle \neq \Phi_{\mathbf{H}}(\langle \mathsf{trace}_j, m_r\rangle)$, which contradicts (27), thus proving (28).

Turning to (29), let's assume there exists $r \in [i, j]$ and $n \in [1, k]$ such that

$$\Phi_{\mathbf{H}}(\langle \mathsf{trace}_r, n\rangle) \neq \Phi_{\mathbf{H}}(\langle \mathsf{trace}_i, n\rangle) \tag{33}$$

We are back to equation (30) with m_r replaced by n, and we derive the same contradiction, thus proving (29).

Using the property P0 of $\mathbf{H}$, we derive from (29)

$$\forall r \in [i, j], \forall m \in [1, k] : \mathsf{trace}_r[m] = \mathsf{trace}_i[m] \tag{34}$$

By construction of the loop at lines 3- 5 of algorithm 1, we derive from (34) and (28) that

$$\forall r \in [i, j], \forall m \in [1, k] : \mathsf{ActiveTreePath}_r[m] = \mathsf{ActiveTreePath}_i[m] \tag{35}$$

and by definition of Φ_0 (equation (10)) that

$$\Phi_0(\langle \mathsf{trace}_i, k\rangle) = \Phi_0(\langle \mathsf{trace}_j, k\rangle) \tag{36}$$

which with (24) and (25) and $l = k$ leads us to $x = y$ and concludes the proof of lemma 4.2.

□

Lemma 4.3. *If a call-tree* $\mathbf{H} = (\mathsf{W}, \mathsf{F})$ *is trace compatible with* traceSequence*, then there exists a surjective call-tree homomorphism from* $\mathbf{H}$ *onto* $\mathbf{G_0} = (\mathsf{V_0}, \mathsf{E_0})$.

Proof. Consider the following mapping h from $\mathbf{H}$ onto $\mathbf{G_0}$:

$$\forall w \in \mathsf{W} : h(w) \equiv \text{ the unique element of } \Phi_0\Big[\Phi_{\mathbf{H}}^{-1}\Big[\{w\}\Big]\Big] \tag{37}$$

Because of lemma 4.2, h is well-defined. Furthermore, because of the properties of set images and pre-images by a function, and because Φ_0 is surjective, we have

$$\forall x \in \mathsf{V_0} : \exists\, w \in \Phi_{\mathbf{H}}\Big[\Phi_0^{-1}[x]\Big] : h(w) = x \tag{38}$$

h is therefore surjective.

In the following we first prove that h is a graph homomorphism, and then that h fulfils (7) and (8), and is thus a call tree homomorphism.

Let's consider $v, w \in \mathsf{W}$ such that $(v, w) \in \mathsf{F}$. We need to prove that $(h(v), h(w)) \in \mathsf{E_0}$.

Because $\Phi_{\mathbf{H}}$ is surjective, there exists a stack frame $\langle \mathsf{t}, k\rangle$ such that

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}, k\rangle) = w \tag{39}$$

Because $(v, w) \in \mathsf{F}$, w is not the root of $\mathbf{H}$, and by applying property P1 if $\mathbf{H}$ on trace t, we conclude that $k > 1$. Again from property P1 we derive

$$\Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}, k-1\rangle), \Phi_{\mathbf{H}}(\langle \mathsf{t}, k\rangle)\Big) \in \mathsf{F} \tag{40}$$

Because $\mathbf{H}$ is a tree, w has at most one parent. $(v, w) \in \mathsf{F}$ and (40) thus imply:

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}, k-1\rangle) = v \tag{41}$$

From (39) and (41) we derive

$$\langle \mathsf{t}, k\rangle \in \Phi_{\mathbf{H}}^{-1}[w] \tag{42}$$
$$\langle \mathsf{t}, k-1\rangle \in \Phi_{\mathbf{H}}^{-1}[v] \tag{43}$$

And thus after applying the mapping Φ_0 and using the definition of h (37):

$$\Phi_0\Big(\langle \mathsf{t}, k\rangle\Big) \in \Phi_0\Big[\Phi_{\mathbf{H}}^{-1}[w]\Big] = \Big\{h(w)\Big\} \tag{44}$$
$$\Phi_0\Big(\langle \mathsf{t}, k-1\rangle\Big) \in \Phi_0\Big[\Phi_{\mathbf{H}}^{-1}[v]\Big] = \Big\{h(v)\Big\} \tag{45}$$

We finally get:

$$\Phi_0\Big(\langle \mathsf{t}, k\rangle\Big) = h(w) \tag{46}$$
$$\Phi_0\Big(\langle \mathsf{t}, k-1\rangle\Big) = h(v) \tag{47}$$

By applying property P1 of $\mathbf{G_0}$ on trace t we conclude:

$$(h(v), h(w)) = \Big(\Phi_0(\langle \mathsf{t}, k-1\rangle), \Phi_0(\langle \mathsf{t}, k\rangle)\Big) \in \mathsf{E_0} \tag{48}$$

which shows that h is a graph homomorphism from $\mathbf{H}$ onto $\mathbf{G_0}$.
We will now turn to equations (7) and (8) to prove that h is a *call-tree homomorphism* and complete the lemma.
Let's consider $w \in \mathsf{W}$. Using property P0 of $\mathbf{H}$ we have

$$\forall \langle \mathsf{t}, k\rangle \in \Phi_{\mathbf{H}}^{-1}[w] : symbol(w) = symbol\Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}, k\rangle)\Big) = \mathsf{t}[k] \tag{49}$$

Using now the definition of h (37) and the property P0 of $\mathbf{G_0}$ we have

$$\forall \langle \mathsf{t}, k\rangle \in \Phi_{\mathbf{H}}^{-1}[w] : symbol\Big(h(w)\Big) = symbol\Big(\Phi_0(\langle \mathsf{t}, k\rangle)\Big) = \mathsf{t}[k] \tag{50}$$

Because $\Phi_{\mathbf{H}}$ is surjective, and therefore $\Phi_{\mathbf{H}}^{-1}[w]$ is non-empty, we derive from (49) and (50) that:

$$symbol(w) = symbol\Big(h(w)\Big) \tag{51}$$

which proves (7) for h.
Turning now to the proof of property (8), let's consider $r, s \in h(\mathsf{W}) = \mathsf{V_0}$, and let's assume

$$\forall u \in h^{-1}[r], \forall v \in h^{-1}[s] : timeStamp(u) \leq timeStamp(v) \tag{52}$$

Let's note R and S the sets of stack frames such that

$$R \equiv \Phi_0^{-1}[r] \wedge S \equiv \Phi_0^{-1}[s] \tag{53}$$

By definition of h (37) and lemma 4.2 we have

$$h^{-1}[r] = \Phi_{\mathbf{H}}[R] \tag{54}$$
$$h^{-1}[s] = \Phi_{\mathbf{H}}[S] \tag{55}$$

Because by definition of R we have $\forall \langle \mathsf{t}_i, k\rangle \in R : \Phi_0(\langle \mathsf{t}_i, k\rangle) = r$, property P1 of $\mathbf{G_0}$ and the fact that $\mathbf{G_0}$ is a tree means all frames of R are at the same stack depth k_r:

$$\forall \langle \mathsf{t}_i, k\rangle \in R : k = k_r \tag{56}$$

The same is true of S. There exists a stack depth m_s such that:

$$\forall \langle \mathsf{t}_j, m\rangle \in S : m = m_s \tag{57}$$

Let's consider two stack frames from R and S:

$$\langle \mathsf{t}_i, k_r\rangle \in R \wedge \langle \mathsf{t}_j, m_s\rangle \in S \tag{58}$$

Note that by definition of R and S (53), we have

$$\begin{array}{rcl} \Phi_0(\langle \mathsf{t}_i, k_r\rangle) & = & r \\ \Phi_0(\langle \mathsf{t}_j, m_s\rangle) & = & s \end{array} \tag{59}$$

We first prove by contradiction that the following holds:

$$(j \geq i) \vee \Big(\forall k \leq k_r : \Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k\rangle)\Big) \tag{60}$$

Let's assume the negation of (60):

$$(j < i) \wedge \Big(\exists k \leq k_r : \Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k\rangle) \neq \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k\rangle)\Big) \tag{61}$$

We can apply property P3 of $\mathbf{H}$ on t_j and t_i, and we get

$$timeStamp\Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, m_s\rangle)\Big) < timeStamp\Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)\Big) \tag{62}$$

Because $\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, m_s\rangle) \in \Phi_{\mathbf{H}}[S] = h^{-1}[s]$ and $\Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle) \in \Phi_{\mathbf{H}}[R] = h^{-1}[r]$, (62) contradicts (52). We therefore conclude that (60) must be true.
Because

$$(i = j) \Rightarrow \Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)\Big)$$

and

$$\Big(\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)\Big) \Rightarrow \Big(\forall k \leq k_r : \Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)\Big)$$

(using the fact that $\mathbf{H}$ is a tree and property P1), (60) is furthermore equivalent to

$$(j > i) \vee \Big(\forall k \leq k_r : \Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)\Big) \tag{63}$$

The remaining of the proof considers two cases, depending whether $m_s < k_r$.

Case 1 $m_s < k_r$

If we assume $\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, m_s\rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, m_s\rangle)$, then $m_s < k_r$, property P1 of $\mathbf{H}$, and (2) give us

$$\begin{aligned} timeStamp(\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, m_s\rangle)) &= timeStamp(\Phi_{\mathbf{H}}(\langle \mathsf{t}_i, m_s\rangle)) && (64)\\ &< timeStamp(\Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r\rangle)) && (65)\end{aligned}$$

The same equation as (62), which yields the same contradiction with (52). From this we conclude that:

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, m_s\rangle) \neq \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, m_s\rangle) \tag{66}$$

With (63) and $m_s < k_r$ (case assumption), this implies that

$$j > i \tag{67}$$

(66) and (67) mean we can apply property P3 of Φ_0 to t_i and t_j. We get

$$timeStamp(\Phi_0(\langle \mathsf{t}_j, m_s\rangle)) > timeStamp(\Phi_0(\langle \mathsf{t}_i, k_r\rangle)) \tag{68}$$

Using (59), this yields

$$timeStamp(s) > timeStamp(r) \tag{69}$$

and proves (8) in this case.

Case 2 $m_s \geq k_r$

We consider in turn the two alternatives of (63).

If $j > i$, then by construction of algorithm 1 and the property of the node constructor **new** $Node(\mathsf{trace}[k])$ at line 13, we have

$$timeStamp(\Phi_0(\langle \mathsf{t}_i, k_r\rangle)) \leq timeStamp(\Phi_0(\langle \mathsf{t}_j, k_r\rangle)) \tag{70}$$

Because of sublemma 4.1.1 on Φ_0 and $m_s \geq k_r$ we also have

$$timeStamp(\Phi_0(\langle \mathsf{t}_j, k_r \rangle)) \leq timeStamp(\Phi_0(\langle \mathsf{t}_j, m_s \rangle)) \tag{71}$$

From (70) and (71) we derive

$$timeStamp(\Phi_0(\langle \mathsf{t}_i, k_r \rangle)) \leq timeStamp(\Phi_0(\langle \mathsf{t}_j, m_s \rangle)) \tag{72}$$

which using (59), gives us

$$timeStamp(r) \leq timeStamp(s) \tag{73}$$

and proves (8) when $j > i$.

If we turn to the second term of (63) and assume

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r \rangle) = \Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r \rangle) \tag{74}$$

Because of (58) and (54) we have

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}_i, k_r \rangle) \in \Phi_{\mathbf{H}}[R] = h^{-1}[r] \tag{75}$$

With (74), we thus have

$$\Phi_{\mathbf{H}}(\langle \mathsf{t}_j, k_r \rangle) \in h^{-1}[r] \tag{76}$$

From which we derive

$$\langle \mathsf{t}_j, k_r \rangle \in \Phi_{\mathbf{H}}^{-1}\Big[h^{-1}[r]\Big] \tag{77}$$

$$\Phi_0\Big(\langle \mathsf{t}_j, k_r \rangle\Big) \in \Phi_0\Big[\Phi_{\mathbf{H}}^{-1}\Big[h^{-1}[r]\Big]\Big] \tag{78}$$

By definition of h (37) and lemma 4.2 this yields

$$\Phi_0\Big(\langle \mathsf{t}_j, k_r \rangle\Big) = r \tag{79}$$

Because $m_s \geq k_r$, sublemma 4.1.1 on t_j gives us

$$timeStamp(\Phi_0(\langle \mathsf{t}_j, m_s \rangle)) \geq timeStamp(\Phi_0(\langle \mathsf{t}_j, k_r \rangle)) \tag{80}$$

This, with (59) and (79) leads to

$$timeStamp(s) \geq timeStamp(r) \tag{81}$$

and concludes the proof (8) for h.

h is thus a surjective graph homomorphism that fulfils (7) and (8), which proves lemma 4.3. □

Theorem 3.1. *The set of call-trees that are trace-compatible with a sequence of observed stack traces is non-empty. The call-tree of this set with the smallest number of nodes is unique (modulo a call-tree isomorphism) and is the call-tree constructed by algorithm 1.*

Proof. As before, let traceSequence be a sequence of observed stack traces, and $\mathbf{G_0} = (\mathsf{E}_0, \mathsf{V}_0)$ the call-tree constructed by algorithm 1 for traceSequence.

Because of lemma 4.1, the set of call-trees that are trace-compatible with traceSequence contains $\mathbf{G_0}$ and is therefore non-empty.

Let's consider $\mathbf{H} = (\mathsf{W}, \mathsf{F})$ a call tree that is trace-compatible with traceSequence. Because of lemma 4.3, there exists a surjective call-tree homomorphism h from $\mathbf{H}$ onto $\mathbf{G_0}$. Because h is surjective we have

$$|\mathsf{E}_0| \leq |\mathsf{W}| \tag{82}$$

i.e., $\mathbf{G_0}$ has the smallest number of nodes of all call-trees compatible with traceSequence.

Let's now consider $\mathbf{H_0} = (\mathsf{W}_0, \mathsf{F}_0)$, a call tree that is trace-compatible with traceSequence and has the same number of nodes as $\mathbf{G_0}$.

$$|\mathsf{E}_0| = |\mathsf{W}_0| \tag{83}$$

Because of lemma 4.3, there exists a surjective call-tree homomorphism h_0 from $\mathbf{H_0}$ onto $\mathbf{G_0}$.

Because h_0 is surjective, and its domain and co-domain are finite and contain the same number of elements, then h_0 is a bijection. $\mathbf{H_0}$ onto $\mathbf{G_0}$ are therefore identical modulo a call-tree isomorphism.

$\mathbf{G_0}$ is therefore unique (modulo a call-tree isomorphism), which concludes the proof of theorem 3.1.

□

References

[1] François Taïani, Marc-Olivier Killijian, and Jean-Charles Fabre. Cosmopen: Dynamic reverse engineering on a budget, how cheap observation techniques can be used to reconstruct complex multi-level behaviour. Technical report, Lancaster University, Computing Departement, 2008. submitted for publication.