Class Notes for CS 819 C - Spring 2002

5. Trees and Dags

John C. Reynolds
Department of Computer Science
Carnegie Mellon University

revised March 5, 2002

©2002 John C. Reynolds

S-expressions (a la LISP)

$\tau \in$ S-exps iff

$$\tau \in \text{Atoms}$$

$$\text{or } \tau = (\tau_1 \cdot \tau_2) \text{ where } \tau_1, \tau_2 \in \text{S-exps.}$$

Representing S-expressions by Trees

For $\tau \in$ S-exps, we define the assertion

$$\mathsf{tree}\, \tau\, (i)$$

by structural induction:

$$\mathsf{tree}\, a\, (i) \text{ iff } \mathbf{emp} \wedge i = a$$

$$\mathsf{tree}\, (\tau_1 \cdot \tau_2)\, (i) \text{ iff}$$
$$\exists i_1, i_2.\ \ i \mapsto i_1, i_2\ *\ \mathsf{tree}\, \tau_1\, (i_1)\ *\ \mathsf{tree}\, \tau_2\, (i_2)$$

Representing S-expressions by Dags

For $\tau \in$ S-exps, we define

$$\mathsf{dag}\,\tau\,(i)$$

by:

$$\mathsf{dag}\,a\,(i) \text{ iff } i = a$$

$$\mathsf{dag}\,(\tau_1 \cdot \tau_2)\,(i) \text{ iff}$$
$$\exists i_1, i_2.\ \ i \mapsto i_1, i_2\ *\ (\mathsf{dag}\,\tau_1\,(i_1) \wedge \mathsf{dag}\,\tau_2\,(i_2))$$

In fact, $\mathsf{dag}\,\tau\,(i)$ is intuitionistic. The proof is by induction on the structure of τ. If τ is an atom a,

$$\begin{aligned}
&\mathsf{dag}\,a\,(i)\ *\ \mathbf{true}\\
&\quad \Rightarrow i = a\ *\ \mathbf{true}\\
&\quad \Rightarrow i = a\\
&\quad \Rightarrow \mathsf{dag}\,a\,(i).
\end{aligned}$$

since $i = a$ is pure. Otherwise, $\tau = (\tau_1 \cdot \tau_2)$, and

$$\begin{aligned}
&\mathsf{dag}\,(\tau_1 \cdot \tau_2)\,(i)\ *\ \mathbf{true}\\
&\quad \Rightarrow \exists i_1, i_2.\ \ i \mapsto i_1, i_2\ *\ (\mathsf{dag}\,\tau_1\,(i_1) \wedge \mathsf{dag}\,\tau_2\,(i_2))\ *\ \mathbf{true}\\
&\quad \Rightarrow \exists i_1, i_2.\ \ i \mapsto i_1, i_2\ *\\
&\qquad ((\mathsf{dag}\,\tau_1\,(i_1)\ *\ \mathbf{true}) \wedge (\mathsf{dag}\,\tau_2\,(i_2)\ *\ \mathbf{true}))\\
&\quad \Rightarrow \exists i_1, i_2.\ \ i \mapsto i_1, i_2\ *\ (\mathsf{dag}\,\tau_1\,(i_1) \wedge \mathsf{dag}\,\tau_2\,(i_2))\\
&\quad \Rightarrow \mathsf{dag}\,(\tau_1 \cdot \tau_2)\,(i),
\end{aligned}$$

by the induction hypothesis for τ_1 and τ_2.

A Problem

Suppose we wish to prove that a recursively defined procedure satisfies

$$\{\mathsf{dag}\,\tau(\mathsf{i})\}\ \mathsf{copytree}(\mathsf{i};\mathsf{j})\ \{\mathsf{dag}\,\tau(\mathsf{i})\ *\ \mathsf{tree}\,\tau(\mathsf{j})\}$$

Then, we must use this specification as a hypothesis in proving that the recursive calls in the procedure body satisfy:

$$\begin{array}{l}\{\mathsf{dag}\,\tau_1(\mathsf{i}_1) \wedge \mathsf{dag}\,\tau_2(\mathsf{i}_2)\}\\ \mathsf{copytree}(\mathsf{i}_1;\mathsf{j}_1)\\ \{(\mathsf{dag}\,\tau_1(\mathsf{i}_1) \wedge \mathsf{dag}\,\tau_2(\mathsf{i}_2))\ *\ \mathsf{tree}\,\tau_1(\mathsf{j}_1)\}\\ \mathsf{copytree}(\mathsf{i}_2;\mathsf{j}_2)\\ \{(\mathsf{dag}\,\tau_1(\mathsf{i}_1) \wedge \mathsf{dag}\,\tau_2(\mathsf{i}_2))\ *\ \mathsf{tree}\,\tau_1(\mathsf{j}_1)\ *\ \mathsf{tree}\,\tau_2(\mathsf{j}_2)\}\end{array}$$

But the hypothesis is not strong enough to imply this. For example, suppose

$$\tau_1 = ((3 \cdot 4) \cdot (5 \cdot 6)) \qquad\qquad \tau_2 = (5 \cdot 6).$$

Then $\mathsf{copytree}(\mathsf{i}_1;\mathsf{j}_1)$ might change the state

$$\begin{array}{l}s = [\,\mathsf{i}_1\!:10 \mid \mathsf{i}_2\!:30\,]\\ h = [\,10\!:20 \mid 11\!:30 \mid 20\!:3 \mid 21\!:4 \mid 30\!:5 \mid 31\!:6\,]\end{array}$$

into the state

$$\begin{array}{l}s = [\,\mathsf{i}_1\!:10 \mid \mathsf{i}_2\!:30 \mid \mathsf{j}_1\!:40\,]\\ h = [\,10\!:30 \mid 11\!:20 \mid 20\!:5 \mid 21\!:6 \mid 30\!:3 \mid 31\!:4\ \mid\\ \qquad\qquad \mid\ 40\!:50 \mid 41\!:60 \mid 50\!:3 \mid 51\!:4 \mid 60\!:5 \mid 61\!:6\,],\end{array}$$

where $\mathsf{dag}\,\tau_2(\mathsf{i}_2)$ is false.

Possible solutions

- Introduce ghost (auxiliary) variables denoting heaps, e.g.

$$\{\mathbf{this}(\mathsf{h}) \wedge \mathsf{dag}\,\tau(\mathsf{i})\}$$
$$\mathsf{copytree}(\mathsf{i};\mathsf{j})$$
$$\{\mathbf{this}(\mathsf{h}) \;*\; \mathsf{tree}\,\tau(\mathsf{j})\}$$

- Introduce ghost (auxiliary) variables denoting assertions, e.g.

$$\{\mathsf{p} \wedge \mathsf{dag}\,\tau(\mathsf{i})\}$$
$$\mathsf{copytree}(\mathsf{i};\mathsf{j})$$
$$\{\mathsf{p} \;*\; \mathsf{tree}\,\tau(\mathsf{j})\}$$

- Introduce passivity, i.e., the ability to assert that portions of the heap remain unchanged throughout the execution of a command.

Assertion Variables

$$\langle \text{assert} \rangle ::= \langle \text{avar} \rangle \mid \cdots$$

$$\text{AStores}_A \;=\; A \to (\text{Heaps} \to \mathbf{B})$$

$$\text{AStates}_{AV} \;=\; \text{AStores}_A \times \text{Stores}_V \times \text{Heaps}$$

$$[\![p \in \langle \text{assert} \rangle]\!]_{\text{assert}} \;\in\; (\bigcup_{A \supseteq^{\text{fin}} \text{AV}(p)} \text{AStores}_A) \to (\bigcup_{V \supseteq^{\text{fin}} \text{FV}(p)} \text{Stores}_V) \to \text{Heaps} \to \mathbf{B},$$

where $\text{AV}(p)$ is the set of assertion variables occurring in p.

When a is an assertion variable,

$$[\![a]\!]_{\text{assert}}\, as\, s\, h \;\text{ iff }\; as\, a\, h.$$

The Substitution Law for Assertions, Revisited

Proposition 9 *Let δ abbreviate the substitution*

$$a_1 \to p_1, \ldots, a_m \to p_m, v_1 \to e_1, \ldots, v_n \to e_n,$$

let as and s be such that $\text{AV}(p_1) \cup \cdots \cup \text{AV}(p_m) \subseteq \text{dom}\, as$ and $\text{FV}(p_1) \cup \cdots \cup \text{FV}(p_m) \cup \text{FV}(e_1) \cup \cdots \cup \text{FV}(e_n) \subseteq \text{dom}\, s$, and let

$$\widehat{as} = [\, a_1 \colon [\![p_1]\!]_{\text{exp}}\, as\, s \mid \ldots \mid a_m \colon [\![p_m]\!]_{\text{exp}}\, as\, s \,]$$

$$\widehat{s} = [\, v_1 \colon [\![e_1]\!]_{\text{exp}}\, s \mid \ldots \mid v_n \colon [\![e_n]\!]_{\text{exp}}\, s \,].$$

If p is an assertion such that $\text{AV}(p) \subseteq \{a_1, \ldots, a_n\}$ and $\text{FV}(p) \subseteq \{v_1, \ldots, v_n\}$, then

$$[\![p/\delta]\!]_{\text{assert}}\, as\, s\, h = [\![p]\!]_{\text{assert}}\, \widehat{as}\, \widehat{s}\, h.$$

Specifications Revisited

Let $A = \mathrm{AV}(p) \cup \mathrm{AV}(q)$ and $V = \mathrm{FV}(p) \cup \mathrm{FV}(c) \cup \mathrm{FV}(q)$. Then

Partial correctness:

$\{p\}\ c\ \{q\}$ holds iff $\forall (as, s, h) \in \mathrm{AStates}_{AV}.\ [\![p]\!]_{\mathrm{assert}}\, as\, s\, h$ implies

$\neg\, (c, (s, h) \rightsquigarrow^* \mathbf{abort})$

and $(\forall (s', h') \in \mathrm{States}_V.\ c, (s, h) \rightsquigarrow^* (s', h')$ implies $[\![q]\!]_{\mathrm{assert}}\, as\, s'\, h')$

Total correctness:

$[\,p\,]\ c\ [\,q\,]$ holds iff $\forall (as, s, h) \in \mathrm{AStates}_{AV}.\ [\![p]\!]_{\mathrm{assert}}\, as\, s\, h$ implies

$\neg\, (c, (s, h) \rightsquigarrow^* \mathbf{abort})$

and $\neg\, (c, (s, h)\uparrow)$

and $(\forall (s', h') \in \mathrm{States}_V.\ c, (s, h) \rightsquigarrow^* (s', h')$ implies $[\![q]\!]_{\mathrm{assert}}\, as\, s'\, h')$

Free Variable Substitution Revisited

$$\frac{\{p\}\ c\ \{q\}}{(\{p\}\ c\ \{q\})/a_1 \to p_1, \ldots, a_m \to p_m, v_1 \to e_1, \ldots, v_n \to e_n}$$

where $a_1, \ldots, a_m$ are the assertion variables occurring in p or q, and $v_1, \ldots, v_n$ are the variables occurring free in p, c, or q, and, if v_i is assigned by c, then e_i is a variable that does not occur free in any other e_j or in any p_j.

For example, in

$$\{\mathsf{a}\}\ \mathsf{x} := \mathsf{y}\ \{\mathsf{a}\},$$

we can substitute $\mathsf{a} \to \mathsf{y} = \mathsf{z}, \mathsf{x} \to \mathsf{x}, \mathsf{y} \to \mathsf{y}$ to obtain

$$\{\mathsf{y} = \mathsf{z}\}\ \mathsf{x} := \mathsf{y}\ \{\mathsf{y} = \mathsf{z}\},$$

but we cannot substitute $\mathsf{a} \to \mathsf{x} = \mathsf{z}, \mathsf{x} \to \mathsf{x}, \mathsf{y} \to \mathsf{y}$ to obtain

$$\{\mathsf{x} = \mathsf{z}\}\ \mathsf{x} := \mathsf{y}\ \{\mathsf{x} = \mathsf{z}\}.$$

A Tree-Copying Procedure

The procedure

$$
\begin{array}{l}
\mathsf{copytree}(\mathsf{i};\mathsf{j}) = \\
\quad \textbf{if } \mathsf{isatom}(\mathsf{i}) \textbf{ then } \mathsf{j} := \mathsf{i} \textbf{ else} \\
\quad\quad \textbf{newvar } \mathsf{i}_1, \mathsf{i}_2, \mathsf{j}_1, \mathsf{j}_2 \textbf{ in} \\
\quad\quad\quad \Big(\mathsf{i}_1 := [\mathsf{i}] \; ; \; \mathsf{i}_2 := [\mathsf{i}+1] \; ; \\
\quad\quad\quad \textbf{copytree}(\mathsf{i}_1;\mathsf{j}_1) \; ; \; \textbf{copytree}(\mathsf{i}_2;\mathsf{j}_2) \; ; \; \mathsf{j} := \textbf{cons}(\mathsf{j}_1,\mathsf{j}_2)\Big)
\end{array}
$$

satisfies

$$
\begin{array}{l}
\{\mathsf{p} \wedge \mathsf{dag}\,\tau(\mathsf{i})\} \\
\mathsf{copytree}(\mathsf{i};\mathsf{j}) \\
\{\mathsf{p} \;*\; \mathsf{tree}\,\tau(\mathsf{j})\}
\end{array}
$$

since

$\{\mathsf{p} \wedge \mathsf{dag}\ \tau(\mathsf{i})\}$
if $\mathsf{isatom}(\mathsf{i})$ **then**
$\{\mathsf{p} \wedge \mathsf{isatom}(\tau) \wedge \tau = \mathsf{i}\}$
$\{\mathsf{p} \ * \ (\mathsf{isatom}(\tau) \wedge \tau = \mathsf{i} \wedge \mathbf{emp})\}$
$\mathsf{j} := \mathsf{i}$
$\{\mathsf{p} \ * \ (\mathsf{isatom}(\tau) \wedge \tau = \mathsf{j} \wedge \mathbf{emp})\}$
else
$\{\exists \tau_1, \tau_2.\ \ \tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p} \wedge \mathsf{dag}\ (\tau_1 \cdot \tau_2)(\mathsf{i})\}$
newvar $\mathsf{i}_1, \mathsf{i}_2, \mathsf{j}_1, \mathsf{j}_2$ **in**
$\Big(\mathsf{i}_1 := [\mathsf{i}]\ ;\ \mathsf{i}_2 := [\mathsf{i}+1]\ ;$
$\{\exists \tau_1, \tau_2.\ \ \tau = (\tau_1 \cdot \tau_2) \wedge$
$\quad \mathsf{p} \wedge (\mathsf{i} \mapsto \mathsf{i}_1, \mathsf{i}_2 \ * \ (\mathsf{dag}\ \tau_1\ (\mathsf{i}_1) \wedge \mathsf{dag}\ \tau_2\ (\mathsf{i}_2)))\}$
$\{\exists \tau_1, \tau_2.\ \ \tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p} \wedge \mathsf{dag}\ \tau_1\ (\mathsf{i}_1) \wedge \mathsf{dag}\ \tau_2\ (\mathsf{i}_2)\}$
$\mathbf{copytree}(\mathsf{i}_1; \mathsf{j}_1)\ ;$
$\{\exists \tau_1, \tau_2.\ \ (\tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p} \wedge \mathsf{dag}\ \tau_2\ (\mathsf{i}_2)) \ * \ \mathsf{tree}\ \tau_1\ (\mathsf{j}_1)\}$
$\mathbf{copytree}(\mathsf{i}_2; \mathsf{j}_2)\ ;$
$\{\exists \tau_1, \tau_2.\ \ (\tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p}) \ * \ \mathsf{tree}\ \tau_1\ (\mathsf{j}_1) \ * \ \mathsf{tree}\ \tau_2\ (\mathsf{j}_2)\}$
$\mathsf{j} := \mathbf{cons}(\mathsf{j}_1, \mathsf{j}_2)$
$\{\exists \tau_1, \tau_2.\ \ (\tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p}) \ *$
$\quad \mathsf{j} \mapsto \mathsf{j}_1, \mathsf{j}_2 \ * \ \mathsf{tree}\ \tau_1\ (\mathsf{j}_1) \ * \ \mathsf{tree}\ \tau_2\ (\mathsf{j}_2)\}$
$\{\exists \tau_1, \tau_2.\ \ (\tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{p}) \ * \ \mathsf{tree}\ (\tau_1 \cdot \tau_2)\ (\mathsf{j})\}\Big)$
$\{\mathsf{p} \ * \ \mathsf{tree}\ \tau(\mathsf{j})\}$

The General Case

Suppose

$$\{P * Q\}\, C\, \{P * R\} \tag{1}$$

holds, and C does not mutate or deallocate the part of the heap described by P. Then the frame rule gives

$$\{P * \mathbf{true} * Q\}\, C\, \{P * \mathbf{true} * R\} \tag{2}$$

and C does not mutate or deallocate the part of the heap described by $P * \mathbf{true}$. Thus, if the assertion variable a describes any property of a heap, then

$$\{(a \wedge (P * \mathbf{true})) * Q\}\, C\, \{a * R\} \tag{3}$$

holds.

From (3), since $P \Rightarrow P * \mathbf{true}$ is valid, one can obtain

$$\{(a \wedge P) * Q\}\, C\, \{a * R\} \tag{4}$$

(If P is intuitionistic, so that $P * \mathbf{true} \Rightarrow P$ is valid, then one can also obtain (3) from (4).)

Substitution of an S-expression for an Atom

$$\begin{aligned} a/a \to \tau' &= \tau' \\ b/a \to \tau' &= b \qquad \text{when } b \in \text{Atoms} - \{a\} \\ (\tau_1 \cdot \tau_2)/a \to \tau' &= ((\tau_1/a \to \tau') \cdot (\tau_2/a \to \tau')) \end{aligned}$$

Substitution by Copying

$$\begin{array}{l}
\mathsf{subst1}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}) = \\
\quad \textbf{if } \mathsf{isatom}(\mathsf{i}) \textbf{ then if } \mathsf{i} = \mathsf{a} \textbf{ then } \mathsf{copytree}(\mathsf{j}\,;\,\mathsf{i}) \textbf{ else skip} \\
\quad \textbf{else newvar } \mathsf{i}_1, \mathsf{i}_2 \textbf{ in } (\mathsf{i}_1 := [\mathsf{i}]\,;\,\mathsf{i}_2 := [\mathsf{i}+1]\,; \\
\qquad \mathsf{subst1}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}_1)\,;\,\mathsf{subst1}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}_2)\,;\,[\mathsf{i}] := \mathsf{i}_1\,;\,[\mathsf{i}+1] := \mathsf{i}_2)
\end{array}$$

satisfies

$$\begin{array}{l}
\{\mathsf{tree}\,\tau\,(\mathsf{i}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\mathsf{subst1}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}) \\
\{\mathsf{tree}\,(\tau/\mathsf{a} \to \tau')\,(\mathsf{i}) \;*\; \mathsf{p}\}
\end{array}$$

since

$$\begin{array}{l}
\{\mathsf{tree}\,\tau\,(\mathsf{i}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\textbf{if } \mathsf{isatom}(\mathsf{i}) \textbf{ then} \\
\quad \{(\mathsf{isatom}(\tau) \wedge \tau = \mathsf{i} \wedge \mathbf{emp}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\quad \textbf{if } \mathsf{i} = \mathsf{a} \textbf{ then} \\
\qquad \{(\mathsf{isatom}(\tau) \wedge \tau = \mathsf{a} \wedge \mathbf{emp}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\qquad \{((\tau/\mathsf{a} \to \tau') = \tau' \wedge \mathbf{emp}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\qquad \mathsf{copytree}(\mathsf{j}\,;\,\mathsf{i}) \\
\qquad \{((\tau/\mathsf{a} \to \tau') = \tau' \wedge \mathbf{emp}) \;*\; \mathsf{tree}\,\tau'\,(\mathsf{i}) \;*\; \mathsf{p}\} \\
\quad \textbf{else} \\
\qquad \{(\mathsf{isatom}(\tau) \wedge \tau \neq \mathsf{a} \wedge \tau = \mathsf{i} \wedge \mathbf{emp}) \;*\; \mathsf{p}\} \\
\qquad \{((\tau/\mathsf{a} \to \tau') = \tau \wedge \mathsf{isatom}(\tau) \wedge \tau = \mathsf{i} \wedge \mathbf{emp}) \;*\; \mathsf{p}\} \\
\qquad \textbf{skip} \\
\qquad \vdots
\end{array}$$

$$\vdots$$

$\mathbf{else\ newvar}\ \mathsf{i}_1, \mathsf{i}_2\ \mathbf{in}\ \Big($

$\{\exists \tau_1, \tau_2, \mathsf{i}_1, \mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto \mathsf{i}_1, \mathsf{i}_2\ *$

$\quad \mathsf{tree}\ \tau_1\ (\mathsf{i}_1)\ *\ \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{i}_1 := [\mathsf{i}]\ ;\ \mathsf{i}_2 := [\mathsf{i} + 1]\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$

$\quad \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ \mathsf{tree}\ \tau_1\ (\mathsf{i}_1)\ *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j}) \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{subst1}(\mathsf{a}, \mathsf{j}\ ;\ \mathsf{i}_1)\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$

$\quad \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ \mathsf{tree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1)\ *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{subst1}(\mathsf{a}, \mathsf{j}\ ;\ \mathsf{i}_2)\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$

$\quad \mathsf{tree}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{i}_2)\ *\ \mathsf{tree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1)\ *\ \mathsf{p})\}$

$[\mathsf{i}] := \mathsf{i}_1\ ;\ [\mathsf{i} + 1] := \mathsf{i}_2$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto \mathsf{i}_1, \mathsf{i}_2\ *$

$\quad \mathsf{tree}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{i}_2)\ *\ \mathsf{tree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1)\ *\ \mathsf{p})\}$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{tree}\ ((\tau_1/\mathsf{a} \to \tau') \cdot (\tau_2/\mathsf{a} \to \tau'))\ (\mathsf{i})\ *\ \mathsf{p})\}\Big)$

$\{\mathsf{tree}\ (\tau/\mathsf{a} \to \tau')\ (\mathsf{i})\ *\ \mathsf{p}\}$

Substitution without Copying

$$
\begin{array}{l}
\mathsf{subst2}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}) = \\
\textbf{if } \mathsf{isatom}(\mathsf{i}) \textbf{ then if } \mathsf{i} = \mathsf{a} \textbf{ then } \mathsf{i} := \mathsf{j} \textbf{ else skip} \\
\textbf{else newvar } \mathsf{i}_1, \mathsf{i}_2 \textbf{ in } (\mathsf{i}_1 := [\mathsf{i}]\,;\, \mathsf{i}_2 := [\mathsf{i}+1]\,; \\
\quad \mathsf{subst2}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}_1)\,;\,\mathsf{subst2}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}_2)\,;\,[\mathsf{i}] := \mathsf{i}_1\,;\,[\mathsf{i}+1] := \mathsf{i}_2)
\end{array}
$$

satisfies

$$
\begin{array}{l}
\{\mathsf{tree}\,\tau\,(\mathsf{i}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\mathsf{subst2}(\mathsf{a},\mathsf{j}\,;\,\mathsf{i}) \\
\{\mathsf{dag}\,(\tau/\mathsf{a} \to \tau')\,(\mathsf{i}) \wedge (\mathsf{p} \;*\; \textbf{true})\}
\end{array}
$$

since

$$
\begin{array}{l}
\{\mathsf{tree}\,\tau\,(\mathsf{i}) \;*\; (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\textbf{if } \mathsf{isatom}(\mathsf{i}) \textbf{ then} \\
\quad \{\mathsf{isatom}(\tau) \wedge \tau = \mathsf{i} \wedge \mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j})\} \\
\quad \textbf{if } \mathsf{i} = \mathsf{a} \textbf{ then} \\
\quad\quad \{\mathsf{isatom}(\tau) \wedge \tau = \mathsf{a} \wedge \mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j})\} \\
\quad\quad \{(\tau/\mathsf{a} \to \tau') = \tau' \wedge \mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j})\} \\
\quad\quad \mathsf{i} := \mathsf{j} \\
\quad \textbf{else} \\
\quad\quad \{\mathsf{isatom}(\tau) \wedge \tau \neq \mathsf{a} \wedge \tau = \mathsf{i} \wedge \mathsf{p}\} \\
\quad\quad \{(\tau/\mathsf{a} \to \tau') = \tau \wedge \mathsf{isatom}(\tau) \wedge \tau = \mathsf{i} \wedge \mathsf{p}\} \\
\quad\quad \textbf{skip} \\
\quad \{\mathsf{dag}\,(\tau/\mathsf{a} \to \tau')\,(\mathsf{i}) \wedge \mathsf{p}\} \\
\quad\quad \vdots
\end{array}
$$

$\vdots$

else newvar $\mathsf{i}_1, \mathsf{i}_2$ **in** $\Big($

$\{\exists \tau_1, \tau_2, \mathsf{i}_1, \mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto \mathsf{i}_1, \mathsf{i}_2\ *$
$\quad \mathsf{tree}\ \tau_1\ (\mathsf{i}_1)\ *\ \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{i}_1 := [\mathsf{i}]\ ;\ \mathsf{i}_2 := [\mathsf{i}+1]\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ \mathsf{tree}\ \tau_1\ (\mathsf{i}_1)\ *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j}) \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{subst2}(\mathsf{a}, \mathsf{j}\ ;\ \mathsf{i}_1)\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ (\mathsf{dag}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1) \wedge ((\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j}))\ *\ \mathbf{true})))\}$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad \mathsf{tree}\ \tau_2\ (\mathsf{i}_2)\ *\ (\mathsf{dag}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1) \wedge (\mathsf{p}\ *\ \mathbf{true}) \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\}$

$\mathsf{subst2}(\mathsf{a}, \mathsf{j}\ ;\ \mathsf{i}_2)\ ;$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad (\mathsf{dag}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{i}_2) \wedge$
$\qquad ((\mathsf{dag}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1) \wedge (\mathsf{p}\ *\ \mathbf{true}))\ *\ \mathbf{true})))\}$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad (\mathsf{dag}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{i}_2) \wedge \mathsf{dag}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1) \wedge (\mathsf{p}\ *\ \mathbf{true})))\}$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge (\mathsf{i} \mapsto -, -\ *$
$\quad (\mathsf{dag}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{i}_2) \wedge \mathsf{dag}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1))) \wedge (\mathsf{p}\ *\ \mathbf{true})\}$

$[\mathsf{i}] := \mathsf{i}_1\ ;\ [\mathsf{i}+1] := \mathsf{i}_2$

$\{\exists \tau_1, \tau_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge \mathsf{dag}\ ((\tau_1/\mathsf{a} \to \tau') \cdot (\tau_2/\mathsf{a} \to \tau'))\ (\mathsf{i}) \wedge$
$\qquad (\mathsf{p}\ *\ \mathbf{true})\}\Big)$

$\{\mathsf{dag}\ (\tau/\mathsf{a} \to \tau')\ (\mathsf{i}) \wedge (\mathsf{p}\ *\ \mathbf{true})\}$

A Problem: Skewed Sharing

Consider a state where

$$\begin{aligned} s(\mathsf{i}) &= 10 \\ h(10) &= 20 \\ h(11) &= 21 \\ h(20) &= 1 \\ h(21) &= 2 \\ h(22) &= 3. \end{aligned}$$

This is an abnormal form of sharing, but it still satisfies

$$\mathsf{dag}\,((1 \cdot 2) \cdot (2 \cdot 3))\,(\mathsf{i}).$$

Representing S-expressions by Back-Linked Trees

For $\tau \in$ S-exps, we define

$$\mathsf{bktree}\,\tau\,(i,b,t)$$

by structural induction:

$$\mathsf{bktree}\,a\,(i,b,t) \text{ iff } i \mapsto b,t,a,-$$

$\mathsf{bktree}\,(\tau_1 \cdot \tau_2)\,(i,b,t)$ iff

$$\exists i_1,i_2.\ \ i \mapsto b,t,i_1,i_2 \ *\ \mathsf{bktree}\,\tau_1\,(i_1,i,0)\ *\ \mathsf{bktree}\,\tau_2\,(i_2,i,1).$$

Then

$\mathsf{copybktree(i,j)} =$

$\quad$ **if** $\mathsf{isatom(i)}$ **then** $[\mathsf{j}+2] := \mathsf{i}$

$\quad$ **else newvar** i',j' **in**

$\quad\quad \Big(\mathsf{i}' := [\mathsf{i}]\ ;\ \mathsf{j}' := \mathbf{cons}(\mathsf{j},0,0,0)\ ;\ \mathsf{copybktree(i',j')}\ ;\ [\mathsf{j}+2] := \mathsf{j}'\ ;$

$\quad\quad \mathsf{i}' := [\mathsf{i}+1]\ ;\ \mathsf{j}' := \mathbf{cons}(\mathsf{j},1,0,0)\ ;\ \mathsf{copybktree(i',j')}\ ;\ [\mathsf{j}+3] := \mathsf{j}'\Big)$

satisfies

$$\{(\mathsf{p} \wedge \mathsf{dag}\,\tau\,(\mathsf{i}))\ *\ \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\}$$
$$\mathsf{copybktree(i,j)}$$
$$\{\mathsf{p}\ *\ \mathsf{bktree}\,\tau\,(\mathsf{j},\mathsf{b},\mathsf{t})\}$$

since

$\{(\mathsf{p} \wedge \mathsf{dag}\,\tau\,(\mathsf{i})) \;*\; \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\}$

if $\mathsf{isatom}(\mathsf{i})$ **then**

$\{(\mathsf{p} \wedge \mathsf{isatom}(\tau) \wedge \tau = i) \;*\; \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\}$

$[\mathsf{j}+2] := \mathsf{i}$

$\{(\mathsf{p} \wedge \mathsf{isatom}(\tau) \wedge \tau = i) \;*\; \mathsf{j} \mapsto \mathsf{b},\mathsf{t},\mathsf{i},-\}$

else newvar i',j' **in**

$\Big(\{\exists \tau_1,\tau_2,\mathsf{i}_1,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\quad \Big(\Big(p \wedge (\mathsf{i} \mapsto \mathsf{i}_1,\mathsf{i}_2 \;*\; (\mathsf{dag}\,\tau_1\,(\mathsf{i}_1) \wedge \mathsf{dag}\,\tau_2\,(\mathsf{i}_2)))\Big) * \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\Big)\}$

$\mathsf{i}' := [\mathsf{i}]\,;$

$\{\exists \tau_1,\tau_2,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\quad \Big(\Big(p \wedge (\mathsf{i} \mapsto \mathsf{i}',\mathsf{i}_2 \;*\; (\mathsf{dag}\,\tau_1\,(\mathsf{i}') \wedge \mathsf{dag}\,\tau_2\,(\mathsf{i}_2)))\Big) * \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\Big)\}$

$\mathsf{j}' := \mathbf{cons}(\mathsf{j},0,0,0)\,;$

$\{\exists \tau_1,\tau_2,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\quad \Big(\Big(p \wedge (\mathsf{i} \mapsto \mathsf{i}',\mathsf{i}_2 \;*\; (\mathsf{dag}\,\tau_1\,(\mathsf{i}') \wedge \mathsf{dag}\,\tau_2\,(\mathsf{i}_2)))\Big) * \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-$

$\qquad * \mathsf{j}' \mapsto \mathsf{j},0,-,-\Big)\}$

$\{\exists \tau_1,\tau_2,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\quad \Big(\Big(p \wedge (\mathsf{i} \mapsto -,\mathsf{i}_2 \;*\; \mathsf{dag}\,\tau_2\,(\mathsf{i}_2)) \wedge \mathsf{dag}\,\tau_1\,(\mathsf{i}')\Big) * \mathsf{j}' \mapsto \mathsf{j},0,-,-$

$\qquad * \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\Big)\}$

$\mathsf{copybktree}(\mathsf{i}',\mathsf{j}')\,;$

$\{\exists \tau_1,\tau_2,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\quad \Big(\Big(p \wedge (\mathsf{i} \mapsto -,\mathsf{i}_2 \;*\; \mathsf{dag}\,\tau_2\,(\mathsf{i}_2))\Big) * \mathsf{bktree}\,\tau_1\,(\mathsf{j}',\mathsf{j},0) \;*\; \mathsf{j} \mapsto \mathsf{b},\mathsf{t},-,-\Big)\}$

$[\mathsf{j}+2] := \mathsf{j}'\,;$

$\quad \vdots$

$$\vdots$$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1, \mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\Big(\big(p \wedge (\mathsf{i} \mapsto -, \mathsf{i}_2\ *\ \mathsf{dag}\ \tau_2\ (\mathsf{i}_2))\big) *\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, -\Big)\}$

$\mathsf{i}' := [\mathsf{i} + 1]\ ;$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\Big(\big(p \wedge (\mathsf{i} \mapsto -, \mathsf{i}'\ *\ \mathsf{dag}\ \tau_2\ (\mathsf{i}'))\big) *\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, -\Big)\}$

$\mathsf{j}' := \mathbf{cons}(\mathsf{j}, 1, 0, 0)\ ;$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\Big(\big(p \wedge (\mathsf{i} \mapsto -, \mathsf{i}'\ *\ \mathsf{dag}\ \tau_2\ (\mathsf{i}'))\big) *\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, -$

$*\ \mathsf{j}' \mapsto \mathsf{j}, 1, -, -\Big)\}$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$\Big(\big(p \wedge \mathsf{dag}\ \tau_2\ (\mathsf{i}')\big) *\ \mathsf{j}' \mapsto \mathsf{j}, 1, -, -$

$*\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, -\Big)\}$

$\mathsf{copybktree}(\mathsf{i}', \mathsf{j}')\ ;$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$(p\ *\ \mathsf{bktree}\ \tau_2\ (\mathsf{j}', \mathsf{j}, 1)\ *\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, -)\}$

$[\mathsf{j} + 3] := \mathsf{j}'$

$\{\exists \tau_1, \tau_2, \mathsf{j}_1, \mathsf{j}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge$

$(p\ *\ \mathsf{bktree}\ \tau_2\ (\mathsf{j}_2, \mathsf{j}, 1)\ *\ \mathsf{bktree}\ \tau_1\ (\mathsf{j}_1, \mathsf{j}, 0)\ *\ \mathsf{j} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{j}_1, \mathsf{j}_2)\}\Big)$

$\{\mathsf{p}\ *\ \mathsf{bktree}\ \tau\ (\mathsf{j}, \mathsf{b}, \mathsf{t})\}.$

Substitution in Back-Linked Trees

We first note that

$$\begin{array}{l}
\mathsf{bktree}\,\tau\,(\mathsf{i},\mathsf{b},\mathsf{t}) \Rightarrow \\
\quad \exists \mathsf{k}.\ \mathsf{i} \hookrightarrow \mathsf{b},\mathsf{t},\mathsf{k},- \\
\qquad \wedge\,(\mathsf{isatom}(\mathsf{k}) \Rightarrow \tau = \mathsf{k} \wedge \mathsf{i} \mapsto \mathsf{b},\mathsf{t},\mathsf{k},-) \\
\qquad \wedge\,(\neg\,\mathsf{isatom}(\mathsf{k}) \Rightarrow \exists \tau_1,\tau_2,\mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge \\
\qquad\quad (\mathsf{i} \mapsto \mathsf{b},\mathsf{t},\mathsf{k},\mathsf{i}_2\ *\ \mathsf{bktree}\,\tau_1\,(\mathsf{k},\mathsf{i},0)\ *\ \mathsf{bktree}\,\tau_2\,(\mathsf{i}_2,\mathsf{i},1)).
\end{array}$$

Then

$$\begin{array}{l}
\mathsf{subst3}(\mathsf{a},\mathsf{j},\mathsf{i}) = \\
\quad \mathbf{newvar}\ \mathsf{k}\ \mathbf{in}\ \Big(\mathsf{k} := [\mathsf{i}+2]\ ; \\
\qquad \mathbf{if}\ \mathsf{isatom}(\mathsf{k})\ \mathbf{then} \\
\qquad\quad \mathbf{if}\ \mathsf{k} = \mathsf{a}\ \mathbf{then}\ \mathsf{copybktree}(\mathsf{j},\mathsf{i})\ \mathbf{else\ skip} \\
\qquad \mathbf{else}\ (\mathsf{subst3}(\mathsf{a},\mathsf{j},\mathsf{k})\ ;\ \mathsf{k} := [\mathsf{i}+3]\ ;\ \mathsf{subst3}(\mathsf{a},\mathsf{j},\mathsf{k}))\Big)
\end{array}$$

satisfies

$$\begin{array}{l}
\{\mathsf{bktree}\,\tau\,(\mathsf{i},\mathsf{b},\mathsf{t})\ *\ (\mathsf{p} \wedge \mathsf{dag}\,\tau'\,(\mathsf{j}))\} \\
\mathsf{subst3}(\mathsf{a},\mathsf{j},\mathsf{i}) \\
\{\mathsf{bktree}\,(\tau/\mathsf{a} \to \tau')\,(\mathsf{i},\mathsf{b},\mathsf{t})\ *\ \mathsf{p}\}
\end{array}$$

since

$\{\mathsf{bktree}\,\tau\,(\mathsf{i},\mathsf{b},\mathsf{t})\ *\ (\mathsf{p}\wedge\mathsf{dag}\,\tau'\,(\mathsf{j}))\}$

$\mathbf{newvar}\ \mathsf{k}\ \mathbf{in}\ \Big(\mathsf{k}:=[\mathsf{i}+2]\ ;$

$\quad\{(\mathsf{i}\hookrightarrow\mathsf{b},\mathsf{t},\mathsf{k},-\wedge(\mathsf{isatom}(\mathsf{k})\Rightarrow\tau=\mathsf{k}\wedge\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},-)$

$\qquad\wedge(\neg\,\mathsf{isatom}(\mathsf{k})\Rightarrow\exists\tau_1,\tau_2,\mathsf{i}_2.\ \tau=(\tau_1\cdot\tau_2)\wedge$

$\qquad\quad(\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},\mathsf{i}_2\ *\ \mathsf{bktree}\,\tau_1\,(\mathsf{k},\mathsf{i},0)\ *\ \mathsf{bktree}\,\tau_2\,(\mathsf{i}_2,\mathsf{i},1))))$

$\quad *\ (\mathsf{p}\wedge\mathsf{dag}\,\tau'\,(\mathsf{j}))\}$

$\quad\mathbf{if}\ \mathsf{isatom}(\mathsf{k})\ \mathbf{then}$

$\qquad\{(\mathsf{isatom}(\tau)\wedge\tau=\mathsf{k}\wedge\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},-)\ *\ (\mathsf{p}\wedge\mathsf{dag}\,\tau'\,(\mathsf{j}))\}$

$\qquad\mathbf{if}\ \mathsf{k}=\mathsf{a}\ \mathbf{then}$

$\qquad\quad\{(\mathsf{isatom}(\tau)\wedge\tau=\mathsf{a}\wedge\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},-)\ *\ (\mathsf{p}\wedge\mathsf{dag}\,\tau'\,(\mathsf{j}))\}$

$\qquad\quad\{(\tau/\mathsf{a}\to\tau')=\tau'\wedge(\mathsf{i}\mapsto\mathsf{b},\mathsf{t},-,-\ *\ (\mathsf{p}\wedge\mathsf{dag}\,\tau'\,(\mathsf{j})))\}$

$\qquad\quad\mathsf{copybktree}(\mathsf{j},\mathsf{i})$

$\qquad\quad\{(\tau/\mathsf{a}\to\tau')=\tau'\wedge(\mathsf{bktree}\,\tau'\,(\mathsf{i},\mathsf{b},\mathsf{t})\ *\ \mathsf{p})\}$

$\qquad\mathbf{else}$

$\qquad\quad\{(\mathsf{isatom}(\tau)\wedge\tau\neq\mathsf{a}\wedge\tau=\mathsf{k}\wedge\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},-)\ *\ \mathsf{p}\}$

$\qquad\quad\{((\tau/\mathsf{a}\to\tau')=\tau\wedge\mathsf{isatom}(\tau)\wedge\tau=\mathsf{k}\wedge\mathsf{i}\mapsto\mathsf{b},\mathsf{t},\mathsf{k},-)\ *\ \mathsf{p}\}$

$\qquad\quad\mathbf{skip}$

$\qquad\vdots$

$$
\begin{array}{l}
\quad \vdots \\
\quad \textbf{else} \\
\quad \Big(\{\exists \tau_1, \tau_2, \mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge \\
\qquad (\mathsf{i} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{k}, \mathsf{i}_2\ *\ \mathsf{bktree}\ \tau_1\ (\mathsf{k}, \mathsf{i}, 0)\ *\ \mathsf{bktree}\ \tau_2\ (\mathsf{i}_2, \mathsf{i}, 1) \\
\qquad *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j}) \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\} \\
\qquad \mathsf{subst3}(\mathsf{a}, \mathsf{j}, \mathsf{k})\ ; \\
\qquad \{\exists \tau_1, \tau_2, \mathsf{i}_2.\ \tau = (\tau_1 \cdot \tau_2) \wedge \\
\qquad (\mathsf{i} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{k}, \mathsf{i}_2\ *\ \mathsf{bktree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{k}, \mathsf{i}, 0)\ *\ \mathsf{bktree}\ \tau_2\ (\mathsf{i}_2, \mathsf{i}, 1) \\
\qquad *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\} \\
\qquad \mathsf{k} := [\mathsf{i} + 3]\ ; \\
\qquad \{\exists \tau_1, \tau_2, \mathsf{i}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge \\
\qquad (\mathsf{i} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{i}_1, \mathsf{k}\ *\ \mathsf{bktree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1, \mathsf{i}, 0)\ *\ \mathsf{bktree}\ \tau_2\ (\mathsf{k}, \mathsf{i}, 1) \\
\qquad *\ (\mathsf{p} \wedge \mathsf{dag}\ \tau'\ (\mathsf{j})))\} \\
\qquad \mathsf{subst3}(\mathsf{a}, \mathsf{j}, \mathsf{k}) \\
\qquad \{\exists \tau_1, \tau_2, \mathsf{i}_1.\ \tau = (\tau_1 \cdot \tau_2) \wedge \\
\qquad (\mathsf{i} \mapsto \mathsf{b}, \mathsf{t}, \mathsf{i}_1, \mathsf{k}\ *\ \mathsf{bktree}\ (\tau_1/\mathsf{a} \to \tau')\ (\mathsf{i}_1, \mathsf{i}, 0) \\
\qquad *\ \mathsf{bktree}\ (\tau_2/\mathsf{a} \to \tau')\ (\mathsf{k}, \mathsf{i}, 1)\ *\ \mathsf{p})\}\Big)\Big) \\
\{\mathsf{bktree}\ (\tau/\mathsf{a} \to \tau')\ (\mathsf{i}, \mathsf{b}, \mathsf{t})\ *\ \mathsf{p}\}
\end{array}
$$