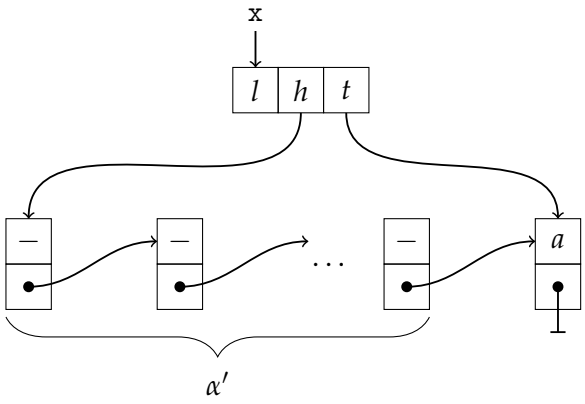
Collections-C Singly Linked Lists

1 The Collections-C (CC) library is a well-used data-structure library on GitHub. This question is inspired by a singly-linked list data structure and algorithms from CC, which have been ported from C to the small while language of the course. Consider a (simplified) version of the CC singly-linked list which represents a mathematical list α by a data structure represented in memory at address x with shape:



where h, t are internal addresses, $\alpha = \alpha' \cdot [a]$ and $l = |\alpha|$, the length of list α. CC singly-linked lists can be described using the following predicates:

$$\mathsf{CClist}(E,\alpha) \triangleq \exists h,t.\mathsf{CClistht}(E,h,t,\alpha)$$

$$\mathsf{CClistht}(x,h,t,\alpha) \triangleq \exists l.\ x \mapsto l,h,t * (P(l,h,t,\alpha) \vee Q(l,h,t,\alpha))$$

$$P(l,h,t,\alpha) \triangleq (l \doteq 0 * h \doteq \texttt{null} * t \doteq \texttt{null} * \alpha \doteq [\,])$$

$$Q(l,h,t,\alpha) \triangleq (\exists \alpha', a.\ l \doteq |\alpha'| + 1 * \mathsf{lseg}(h,t,\alpha') * t \mapsto a,\texttt{null} * \alpha \doteq \alpha' \cdot [a])$$

$$\mathsf{lseg}(x,h,\alpha) \triangleq (x \doteq h * \alpha \doteq [\,]) \vee (\exists a, \alpha', z.\ x \mapsto a,z * \mathsf{lseg}(z,h,\alpha') * \alpha \doteq a : \alpha')$$

a Prove by induction on the size of α that the following implication is valid: $\vDash (\mathsf{lseg}(x,h,\alpha) * h \mapsto a,t) \Rightarrow \mathsf{lseg}(x,t,\alpha \cdot [a])$.

Answer:

This is simple bookwork that the students will know from the lectures. Proof by induction on the length of α.

Base Case: $|\alpha| = 0$, that is, $\alpha = [\,]$.

$\models \mathsf{lseg}(x,h,[\,]) * h \mapsto a,t$
$\Rightarrow x \doteq h * [\,] \doteq [\,] * h \mapsto a,t$
$\Rightarrow x \mapsto a,t * t \doteq t * [\,] \doteq [\,]$
$\Rightarrow x \mapsto a,t * \mathsf{lseg}(t,t,[\,])$
$\Rightarrow \exists a,y.\, x \mapsto a,y * \mathsf{lseg}(y,t,[\,])$
$\Rightarrow \mathsf{lseg}(x,t,[a])$
$\Rightarrow \mathsf{lseg}(x,t,[\,]\cdot[a])$

Inductive Step: Assume property for all $|\alpha'| = n$. Consider arbitrary $\alpha = a : \alpha'$ with $|\alpha'| = n$.

$\models \mathsf{lseg}(x,h,\alpha) * h \mapsto a,t$
// Now unfold
$\Rightarrow \exists a',y,\alpha'.\, x \mapsto a',y * \mathsf{lseg}(y,h,\alpha') * \alpha \doteq a' : \alpha' * h \mapsto a,t$
// Now use IH
$\Rightarrow \exists a',y,\alpha'.\, x \mapsto a',y \;\; * \mathsf{lseg}(y,t,\alpha'\cdot[a]) * \alpha \doteq a' : \alpha'$
$\Rightarrow \exists a',y,\alpha'.\, x \mapsto a',y * \mathsf{lseg}(y,t,\alpha'\cdot[a]) * \alpha\cdot[a] \doteq (a' : \alpha')\cdot[a]$
$\Rightarrow \exists a',y,\alpha'.\, x \mapsto a',y * \mathsf{lseg}(y,t,\alpha'\cdot[a]) * \alpha\cdot[a] \doteq a' : (\alpha'\cdot[a])$
$\Rightarrow \exists a',y,\alpha'.\, x \mapsto a',y * \mathsf{lseg}(y,t,\alpha') * \alpha\cdot[a] \doteq a' : \alpha'$
//Now fold
$\Rightarrow \mathsf{lseg}(x,t,a' : \alpha') * \alpha\cdot[a] \doteq a' : \alpha'$
$\Rightarrow \mathsf{lseg}(x,t,\alpha\cdot[a])$

b The `CClist_addnode(x, v)` function adds a new value to the CC list:

$\Gamma_{\mathtt{CCadd}} \vdash \{\; P_{\mathtt{CCadd}} : \mathtt{x} \doteq x * \mathtt{v} \doteq v * \mathsf{CClist}(x,\alpha) \;\}$
`CClist_addnode(x, v)` {
$\{\; \mathtt{x} \doteq x * \mathtt{v} \doteq v * \mathsf{CClist}(x,\alpha) * \mathtt{n},\mathtt{l},\mathtt{t} \doteq \mathtt{null} \;\}$
`n := new(2);`
`[n] := v;`
`l := [x];`
`if (l = 0) {`
`[x] := 1; [x + 1] := n; [x + 2] := n;`
`} else {`
`t := [x + 2];`
`[t + 1] := n; [x + 2] := n;`
`[x] := l + 1;`
`};`
`return x`
}
$\{\; Q_{\mathtt{CCadd}} : \mathsf{CClist}(\mathtt{ret},\alpha\cdot[v]) \;\}$

where $\Gamma_{\mathtt{CCadd}} \triangleq \{\{P_{\mathtt{CCadd}}\}\mathtt{CClist\_addnode(x,v)}\{Q_{\mathtt{CCadd}}\}\}$.

Give a proof sketch to show that the `CClist_addnode(x, v)` function satisfies its specification, paying particular attention to the unfolding and folding of predicates and the use of the consequence rule.

Answer:

$\Gamma_{\texttt{CCadd}} \vdash \{\ P_{\texttt{CCadd}} : \texttt{x} \doteq x * \texttt{v} \doteq v * \mathsf{CClist}(x,\alpha)\ \}$

$\texttt{CCllist\_addnode}(\texttt{x},\texttt{v})\ \{$

$\{\ \texttt{x} \doteq x * \texttt{v} \doteq v * \mathsf{CClist}(x,\alpha) * \texttt{n},\texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$\{\ \texttt{v} \doteq v * \mathsf{CClist}(\texttt{x},\alpha) * \texttt{n},\texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$\texttt{n} := \texttt{new}(2);$

$\{\ \texttt{v} \doteq v * \mathsf{CClist}(\texttt{x},\alpha) * \texttt{n} \mapsto \texttt{null},\texttt{null} * \texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$[\texttt{n}] := \texttt{v};$

$\{\ \mathsf{CClist}(\texttt{x},\alpha) * \texttt{n} \mapsto v,\texttt{null} * \texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$\{\ \exists h,t.\ \mathsf{CClistht}(\texttt{x},h,t,\alpha) * \texttt{n} \mapsto v,\texttt{null} * \texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$\{\ \exists h,t,l.\ \texttt{x} \mapsto l,h,t * (P(l,h,t,\alpha) \vee Q(l,h,t,\alpha)) * \texttt{n} \mapsto v,\texttt{null} * \texttt{l},\texttt{t} \doteq \texttt{null}\ \}$

$\texttt{l} := [\texttt{x}];$

$\{\ \exists h,t.\ \texttt{x} \mapsto \texttt{l},h,t * (P(\texttt{l},h,t,\alpha) \vee Q(\texttt{l},h,t,\alpha)) * \texttt{n} \mapsto v,\texttt{null} * \texttt{t} \doteq \texttt{null}\ \}$

$\texttt{if}\ (\texttt{l} = 0)\ \{$

$\{\ \exists h,t.\ \texttt{x} \mapsto 0,h,t * \texttt{l} \doteq 0 * h \doteq \texttt{null} * t \doteq \texttt{null} * \alpha \doteq [\,] * \texttt{n} \mapsto v,\texttt{null} * \texttt{t} \doteq \texttt{null}\ \}$

$[\texttt{x}] := \texttt{l} + 1; [\texttt{x}+1] := \texttt{n};\ [\texttt{x}+2] := \texttt{n}$

$\{\ \texttt{x} \mapsto 1,\texttt{n},\texttt{n} * \alpha \doteq [\,] * \texttt{n} \mapsto v,\texttt{null}\ \}$

$\{\ \texttt{x} \mapsto 1,\texttt{n},\texttt{n} * \alpha \doteq [\,] * \mathsf{lseg}(\texttt{n},\texttt{n},[\,]) * \texttt{n} \mapsto v,\texttt{null}\ \}$ `lseg` from empty

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot [v])\ \}$

$\}\ \texttt{else}\ \{$

$\{\ \exists h,t.\ \exists \alpha', a.\ \texttt{x} \mapsto \texttt{l},h,t * \mathsf{lseg}(h,t,\alpha') * t \mapsto a,\texttt{null} * \alpha \doteq \alpha' \cdot [a] * \texttt{l} \doteq |\alpha'| + 1 * \texttt{n} \mapsto v,\texttt{null} * \texttt{t} \doteq \texttt{null}\ \}$

$\texttt{t} := [\texttt{x}+2];$

$\{\ \exists h.\ \exists \alpha', a.\ \texttt{x} \mapsto \texttt{l},h,\texttt{t} * \mathsf{lseg}(h,\texttt{t},\alpha') * \texttt{t} \mapsto a,\texttt{null} * \alpha \doteq \alpha' \cdot [a] * \texttt{l} \doteq |\alpha'| + 1 * \texttt{n} \mapsto v,\texttt{null}\ \}$

$[\texttt{t}+1] := \texttt{n};\ [\texttt{x}+2] := \texttt{n};\ [\texttt{x}] := \texttt{l} + 1;$

$\{\ \exists h.\ \exists \alpha', a.\ \texttt{x} \mapsto \texttt{l}+1,h,\texttt{n} * \mathsf{lseg}(h,\texttt{t},\alpha') * \texttt{t} \mapsto a,\texttt{n} * \alpha \doteq \alpha' \cdot [a] * \texttt{l} \doteq |\alpha'| + 1 * \texttt{n} \mapsto v,\texttt{null}\ \}$

$\{\ \exists h.\ \exists \alpha', a.\ \texttt{x} \mapsto \texttt{l}+1,h,\texttt{n} * \mathsf{lseg}(h,\texttt{n},\alpha' \cdot [a]) * \alpha \doteq \alpha' \cdot [a] * \texttt{l} \doteq |\alpha'| + 1 * \texttt{n} \mapsto v,\texttt{null}\ \}$ part a

$\{\ \exists h,t.\ \texttt{x} \mapsto |\alpha| + 1,h,t * \mathsf{lseg}(h,t,\alpha) * t \mapsto v,\texttt{null}\ \}$ renaming and expanding the exists

$\{\ \exists h,t.\mathsf{CClistht}(\texttt{x},h,t,\alpha \cdot [v])\ \}$

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot [v])\ \}$

$\};$

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot [v])\ \}$

$\texttt{return}\ \texttt{x}$

$\}$

$\{\ Q_{\texttt{CCadd}} : \mathsf{CClist}(\texttt{ret},\alpha \cdot [v])\ \}$

c The $\texttt{CClist\_concat}(\texttt{x},\texttt{y})$ function concatenates two CC lists:

$\Gamma_{\texttt{CCconc}} \vdash \{\ P_{\texttt{CCconc}} : \texttt{x} \doteq x * \texttt{y} \doteq y * \mathsf{CClist}(x,\alpha) * \mathsf{CClist}(y,\alpha')\ \}$
$\texttt{CClist\_concat}(\texttt{x},\texttt{y})\ \{$
$\quad \{\ \texttt{x} \doteq x * \texttt{y} \doteq y * \mathsf{CClist}(x,\alpha) * \mathsf{CClist}(y,\alpha') * \texttt{l1},\texttt{l2},\texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null}\ \}$
$\quad \texttt{l1} := [\texttt{x}];$
$\quad \texttt{l2} := [\texttt{y}];$
$\quad \texttt{if}\ (\texttt{l1} = 0)\ \{$
$\quad\quad \texttt{z} := \texttt{x};\ \texttt{x} := \texttt{y};$
$\quad\quad \texttt{dispose}(\texttt{z});\ \texttt{dispose}(\texttt{z}+1);\ \texttt{dispose}(\texttt{z}+2)$
$\quad \}\ \texttt{else}\ \{$
$\quad\quad \texttt{if}\ (\texttt{l2} = 0)\ \{$
$\quad\quad\quad \texttt{dispose}(\texttt{y});\ \texttt{dispose}(\texttt{y}+1);\ \texttt{dispose}(\texttt{y}+2)$
$\quad\quad \}\ \texttt{else}\ \{$
$\quad\quad\quad \left\{\ R : \quad \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha)) * \right.$
$\quad\quad\quad\quad \left.\ \texttt{y} \mapsto \texttt{l2},h_2,t_2 * Q(\texttt{l2},h_2,t_2,\alpha') * \texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null}\ \right\}$
$\quad\quad\quad \texttt{t1} := [\texttt{x}+2];\ \texttt{h2} := [\texttt{y}+1];\ \texttt{t2} := [\texttt{y}+2];$
$\quad\quad\quad [\texttt{t1}+1] := \texttt{h2};$
$\quad\quad\quad [\texttt{x}+2] := \texttt{t2};$
$\quad\quad\quad [\texttt{x}] := \texttt{l1} + \texttt{l2};$
$\quad\quad\quad \texttt{dispose}(\texttt{y});\ \texttt{dispose}(\texttt{y}+1);\ \texttt{dispose}(\texttt{y}+2)$
$\quad\quad \};$
$\quad \};$
$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot \alpha')\ \}$
$\quad \texttt{return}\ \texttt{x}$
$\}$
$\{\ Q_{\texttt{CCconc}} : \mathsf{CClist}(\texttt{ret},\alpha \cdot \alpha')\ \}$

where $\Gamma_{\texttt{CCconc}} \triangleq \{\{P_{\texttt{CCconc}}\}\texttt{CClist\_concat}(\texttt{x},\texttt{y})\{Q_{\texttt{CCconc}}\}$.

Give a proof sketch to show that the $\texttt{CClist\_concat}(\texttt{x},\texttt{y})$ function satisfies its specification, starting from the assertion R and paying particular attention to the unfolding and folding of predicates and the use of the consequence rule to establish the postcondition. State any lemmas about `lseg` you might use.

Answer:

The additional lemma is

$$\vDash \left(\mathsf{lseg}(x,h,\alpha) * \mathsf{lseg}(h,t,\alpha')\right) \Rightarrow \mathsf{lseg}(x,t,\alpha \cdot \alpha')$$

which they will recognise from the lectures.

$\Gamma_{\texttt{CCconc}} \vdash \{\ P_{\texttt{CCconc}} : \texttt{x} \doteq x * \texttt{y} \doteq y * \mathsf{CClist}(x,\alpha) * \mathsf{CClist}(y,\alpha')\ \}$

$\texttt{CClist\_concat}(\texttt{x},\texttt{y})\ \{$

$\{\ \texttt{x} \doteq x * \texttt{y} \doteq y * \mathsf{CClist}(x,\alpha) * \mathsf{CClist}(y,\alpha') * \texttt{l1},\texttt{l2},\texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null}\ \}$

$\{\ \mathsf{CClist}(\texttt{x},\alpha) * \mathsf{CClist}(\texttt{y},\alpha') * \texttt{l1},\texttt{l2},\texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null}\ \}$

$\{\ \exists h_1,t_1,h_2,t_2.\ \mathsf{CClistht}(\texttt{x},h_1,t_1,\alpha) * \mathsf{CClistht}(\texttt{y},h_2,t_2,\alpha') * \texttt{l1},\texttt{l2},\texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null}\ \}$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2,l_1,l_2.\ \texttt{x} \mapsto l_1,h_1,t_1 * (P(l_1,h_1,t_1,\alpha) \vee Q(l_1,h_1,t_1,\alpha)) * \\ \texttt{y} \mapsto l_2,h_2,t_2 * (P(l_2,h_2,t_2,\alpha') \vee Q(l_2,h_2,t_2,\alpha')) * \\ \texttt{l1},\texttt{l2},\texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null} \end{array} \right\}$

$\texttt{l1} := [\texttt{x}];\quad \texttt{l2} := [\texttt{y}];$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * (P(\texttt{l1},h_1,t_1,\alpha) \vee Q(\texttt{l1},h_1,t_1,\alpha)) * \\ \texttt{y} \mapsto \texttt{l2},h_2,t_2 * (P(\texttt{l2},h_2,t_2,\alpha') \vee Q(\texttt{l2},h_2,t_2,\alpha')) * \\ \texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null} \end{array} \right\}$

$\texttt{if}\ (\texttt{l1} = 0)\ \{$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto 0,h_1,t_1 * P(0,h_1,t_1,\alpha) * \\ \texttt{y} \mapsto l2,h_2,t_2 * (P(\texttt{l2},h_2,t_2,\alpha') \vee Q(\texttt{l2},h_2,t_2,\alpha')) * \texttt{z} \doteq \texttt{null} \end{array} \right\}$

$\texttt{z} := \texttt{x};\ \texttt{x} := \texttt{y};$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ \texttt{z} \mapsto 0,h_1,t_1 * P(0,h_1,t_1,\alpha) * \\ \texttt{x} \mapsto l2,h_2,t_2 * (P(\texttt{l2},h_2,t_2,\alpha') \vee Q(\texttt{l2},h_2,t_2,\alpha')) \end{array} \right\}$

$\texttt{dispose}(\texttt{z});\ \texttt{dispose}(\texttt{z}+1);\ \texttt{dispose}(\texttt{z}+2)$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ P(0,h_1,t_1,\alpha) * \\ \texttt{x} \mapsto l2,h_2,t_2 * (P(\texttt{l2},h_2,t_2,\alpha') \vee Q(\texttt{l2},h_2,t_2,\alpha')) \end{array} \right\}$

$\left\{ \begin{array}{l} (\exists h_1,t_1.\ P(0,h_1,t_1,\alpha)) * \\ (\exists h_2,t_2,l_2.\ \texttt{l2} \doteq l_2 * \texttt{x} \mapsto l_2,h_2,t_2 * (P(l_2,h_2,t_2,\alpha') \vee Q(l_2,h_2,t_2,\alpha'))) \end{array} \right\}$ expansion of exists

$\{\ \alpha \doteq [\,] * \exists h_2,t_2.\ \mathsf{CClistht}(\texttt{x},h_2,t_2,\alpha')\ \}$

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot \alpha')\ \}$

$\}\ \texttt{else}\ \{$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha)) * \\ \texttt{y} \mapsto \texttt{l2},h_2,t_2 * (P(\texttt{l2},h_2,t_2,\alpha') \vee Q(\texttt{l2},h_2,t_2,\alpha')) * \\ \texttt{z},\texttt{h1},\texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null} \end{array} \right\}$

$\texttt{if}\ (\texttt{l2} = 0)\ \{$

$\left\{ \begin{array}{l} \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha) * \\ \texttt{y} \mapsto 0,h_2,t_2 * P(0,h_2,t_2,\alpha') \end{array} \right\}$

$\texttt{dispose}(\texttt{y});\ \texttt{dispose}(\texttt{y}+1);\ \texttt{dispose}(\texttt{y}+2)$

$\{\ \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha) * P(0,h_2,t_2,\alpha')\ \}$

$\{\ \exists h_1,t_1.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha) * \alpha' \doteq [\,]\ \}$

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot \alpha')\ \}$

$\}\ \texttt{else}\ \{$

$\left\{ \begin{array}{ll} R: & \exists h_1,t_1,h_2,t_2.\ \texttt{x} \mapsto \texttt{l1},h_1,t_1 * Q(\texttt{l1},h_1,t_1,\alpha)) * \\ & \texttt{y} \mapsto \texttt{l2},h_2,t_2 * Q(\texttt{l2},h_2,t_2,\alpha') * \texttt{h2},\texttt{t1},\texttt{t2} \doteq \texttt{null} \end{array} \right\}$

$\texttt{t1} := [\texttt{x}+2];\ \texttt{h2} := [\texttt{y}+1];\ \texttt{t2} := [\texttt{y}+2];$

$\{\ \exists h_1.\ \texttt{x} \mapsto \texttt{l1},h_1,\texttt{t1} * Q(\texttt{l1},h_1,\texttt{t1},\alpha)) * \texttt{y} \mapsto \texttt{l2},\texttt{h2},\texttt{t2} * Q(\texttt{l2},\texttt{h2},\texttt{t2},\alpha')\ \}$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1},h_1,\texttt{t1} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t}_1,\alpha') * \texttt{t}_1 \mapsto a,\texttt{null} * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{y} \mapsto \texttt{l2},\texttt{h2},\texttt{t2} * \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$

$[\texttt{t1}+1] := \texttt{h2};$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1},h_1,\texttt{t1} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t1},\alpha') * \texttt{t1} \mapsto a,\texttt{h2} * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{y} \mapsto \texttt{l2},\texttt{h2},\texttt{t2} * \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$

$[\texttt{x}+2] := \texttt{t2};$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1},h_1,\texttt{t2} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t1},\alpha') * \texttt{t1} \mapsto a,\texttt{h2} * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{y} \mapsto \texttt{l2},\texttt{h2},\texttt{t2} * \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$

$[\texttt{x}] := \texttt{l1} + \texttt{l2};$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1}+\texttt{l2},h_1,\texttt{t2} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t1},\alpha') * \texttt{t1} \mapsto a,\texttt{h2} * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{y} \mapsto \texttt{l2},\texttt{h2},\texttt{t2} * \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$

$\texttt{dispose}(\texttt{y});\ \texttt{dispose}(\texttt{y}+1);\ \texttt{dispose}(\texttt{y}+2)$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1}+\texttt{l2},h_1,\texttt{t2} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t1},\alpha') * \texttt{t1} \mapsto a,\texttt{h2} * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1}+\texttt{l2},h_1,\texttt{t2} * \texttt{l1} \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{h2},\alpha' \cdot [a]) * \alpha \doteq \alpha' \cdot [a] * \\ \texttt{l2} \doteq |\alpha''|+1 * \mathsf{lseg}(\texttt{h2},\texttt{t2},\alpha'') * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$ using part a

$\left\{ \begin{array}{l} \exists h_1,\alpha',\alpha'',a,b.\ \texttt{x} \mapsto \texttt{l1}+\texttt{l2},h_1,\texttt{t2} * \texttt{l1}+\texttt{l2} \doteq |\alpha' \cdot [a] \cdot \alpha''|+1 * \mathsf{lseg}(h_1,\texttt{t2},\alpha' \cdot [a] \cdot \alpha'') * \\ \alpha \cdot \alpha'' \doteq \alpha' \cdot [a] \cdot \alpha'' * \texttt{t2} \mapsto b,\texttt{null} * \alpha' \doteq \alpha'' \cdot [b] \end{array} \right\}$ using add. lemma

$\left\{ \begin{array}{l} \exists h_1,l,\alpha',a.\ \texttt{x} \mapsto l,h_1,\texttt{t2} * l \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t2},\alpha') * \\ \texttt{t2} \mapsto a,\texttt{null} * \alpha \cdot \alpha' \doteq \alpha' \cdot [a] \end{array} \right\}$ renaming and expanding exists

$\{\ \exists h_1,l.\ \texttt{x} \mapsto l,h_1,\texttt{t2} * \exists \alpha',a.\ l \doteq |\alpha'|+1 * \mathsf{lseg}(h_1,\texttt{t2},\alpha') * \texttt{t2} \mapsto a,\texttt{null} * \alpha \cdot \alpha' \doteq \alpha' \cdot [a]\ \}$

$\{\ \exists h_1,l.\ \texttt{x} \mapsto l,h_1,\texttt{t2} * Q(l,h_1,\texttt{t2},\alpha \cdot \alpha')\ \}$

$\{\ \exists h,l,t.\ \texttt{x} \mapsto l,h,t * Q(l,h,t,\alpha \cdot \alpha')\ \}$ renaming expanding exists

$\{\ \exists h,t.\ \mathsf{CClistht}(\texttt{x},h,t,\alpha \cdot \alpha')\ \}$ renaming expanding exists

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot \alpha')\ \}$

$\};\quad \};$

$\{\ \mathsf{CClist}(\texttt{x},\alpha \cdot \alpha')\ \}$

$\texttt{return x}$

$\}\{\ Q_{\texttt{CCconc}} : \mathsf{CClist}(\texttt{ret},\alpha \cdot \alpha')\ \}$