Appendix A

For lock-free dynamic hash tables

A.1 Invariants

Some abbreviations.

$$Find(\mathtt{r},\ \mathtt{a}) \triangleq \mathtt{r} = \mathbf{null} \vee \mathtt{a} = ADR(\mathtt{r})$$
$$LeastFind(a,\ n) \triangleq (\forall m < n : \neg Find(\mathtt{Y}[key(a,\ curSize,\ m)],\ a)) \wedge Find(\mathtt{Y}[key(a,\ curSize,\ n)],\ a))$$
$$LeastFind(h,\ a,\ n) \triangleq (\forall m < n : \neg Find(h.\mathtt{table}[key(a,\ h.\mathtt{size},\ m)],\ a)) \wedge Find(h.\mathtt{table}[key(a,\ h.\mathtt{size},\ n)],\ a))$$

Axioms on functions *key* and *ADR*.

Ax1: $v = \mathbf{null} \ \equiv\ ADR(v) = \mathbf{0}$

Ax2: $0 \leq key(a,\ l,\ k) < l$

Ax3: $0 \leq k < m < l \ \Rightarrow\ key(a,\ l,\ k) \neq key(a,\ l,\ m)$

Main correctness properties

Co1: $pc = 14 \ \Rightarrow\ val(r_{fi}) = rS_{fi}$

Co2: $pc \in \{25, 26\} \ \Rightarrow\ suc_{del} = sucS_{del}$

Co3: $pc \in \{41, 42\} \ \Rightarrow\ suc_{ins} = sucS_{ins}$

Cn1: $pc = 14 \ \Rightarrow\ cnt_{fi} = 1$

Cn2: $pc \in \{25, 26\} \ \Rightarrow\ cnt_{del} = 1$

Cn3: $pc \in \{41, 42\} \ \Rightarrow\ cnt_{ins} = 1$

Cn4: $pc = 57 \ \Rightarrow\ cnt_{ass} = 1$

The absence of memory loss is shown by

No1: $\sharp(nbSet1) \leq 2 * P$

No2: $\sharp(nbSet1) = \sharp(nbSet2)$

where $nbSet1$ and $nbSet2$ are sets of integers, characterized by

$$
\begin{aligned}
\mathit{nbSet1} &= \{k \mid k < \texttt{H\_index} \wedge \texttt{Heap}(k) \neq \bot\} \\
\mathit{nbSet2} &= \{i \mid \texttt{H}(i) \neq 0 \vee (\exists r : \mathit{pc}.r = 71 \wedge i_{rA}.r = i)\}
\end{aligned}
$$

Further, we have the following definitions of sets of integers:

$$
\begin{aligned}
\mathit{deSet1} &= \{k \mid k < \mathit{curSize} \wedge \texttt{Y}[k] = \mathbf{del}\} \\
\mathit{deSet2} &= \{r \mid \mathit{index}.r = \texttt{currInd} \wedge \mathit{pc}.r = 25 \wedge \mathit{suc}_{del}.r\} \\
\mathit{deSet3} &= \{k \mid k < \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{size} \wedge \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{table}[k] = \mathbf{del}\} \\
\mathit{ocSet1} &= \{r \mid \mathit{index}.r \neq \texttt{currInd} \\
&\quad \vee \mathit{pc}.r \in [30, 41] \vee \mathit{pc}.r \in [46, 57] \vee \mathit{pc}.r \in [59, 65] \wedge \mathit{return}_{gA}.r \geq 30 \\
&\quad \vee \mathit{pc}.r \in [67, 72] \wedge (\mathit{return}_{rA}.r = 59 \wedge \mathit{return}_{gA}.r \geq 30 \\
&\qquad \vee \mathit{return}_{rA}.r = 90 \wedge \mathit{return}_{ref}.r \geq 30) \\
&\quad \vee (\mathit{pc}.r = 90 \vee \mathit{pc}.r \in [104, 105]) \wedge \mathit{return}_{ref}.r \geq 30\} \\
\mathit{ocSet2} &= \{r \mid \mathit{pc}.r \geq 125 \wedge b_{mE}.r \wedge \mathit{to}.r = \texttt{H}(\texttt{currInd})\} \\
\mathit{ocSet3} &= \{r \mid \mathit{index}.r = \texttt{currInd} \wedge \mathit{pc}.r = 41 \wedge \mathit{suc}_{ins}.r \\
&\quad \vee \mathit{index}.r = \texttt{currInd} \wedge \mathit{pc}.r = 57 \wedge r_{ass}.r = \mathbf{null}\} \\
\mathit{ocSet4} &= \{k \mid k < \mathit{curSize} \wedge \mathit{val}(\texttt{Y}[k]) \neq \mathbf{null}\} \\
\mathit{ocSet5} &= \{k \mid k < \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{size} \wedge \mathit{val}(\texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{table}[k]) \neq \mathbf{null}\} \\
\mathit{ocSet6} &= \{k \mid k < \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{size} \wedge \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{table}[k] \neq \mathbf{null}\} \\
\mathit{ocSet7} &= \{r \mid \mathit{pc}.r \geq 125 \wedge b_{mE}.r \wedge \mathit{to}.r = \texttt{H}(\texttt{next}(\texttt{currInd}))\} \\
\mathit{prSet1}(i) &= \{r \mid \mathit{index}.r = i \wedge \mathit{pc}.r \notin \{0, 59, 60\}\} \\
\mathit{prSet2}(i) &= \{r \mid \mathit{index}.r = i \wedge \mathit{pc}.r \in \{104, 105\} \ \vee\ i_{rA}.r = i \wedge \mathit{index}.r \neq i \wedge \mathit{pc}.r \in [67, 72] \\
&\quad \vee\ i_{nT}.r = i \wedge \mathit{pc}.r \in [81, 84] \ \vee\ i_{mig}.r = i \wedge \mathit{pc}.r \geq 97\} \\
\mathit{prSet3}(i) &= \{r \mid \mathit{index}.r = i \wedge \mathit{pc}.r \in [61, 65] \cup [104, 105] \ \vee\ i_{rA}.r = i \wedge \mathit{pc}.r = 72 \\
&\quad \vee\ i_{nT}.r = i \wedge \mathit{pc}.r \in [81, 82] \ \vee\ i_{mig}.r = i \wedge \mathit{pc}.r \in [97, 98]\} \\
\mathit{prSet4}(i) &= \{r \mid \mathit{index}.r = i \wedge \mathit{pc}.r \in [61, 65] \ \vee\ i_{mig}.r = i \wedge \mathit{pc}.r \in [97, 98]\} \\
\mathit{buSet1}(i) &= \{r \mid \mathit{index}.r = i \ \wedge\ (\mathit{pc}.r \in [1, 58] \cup (62, 68] \wedge \mathit{pc}.r \neq 65 \\
&\qquad \vee\ \mathit{pc}.r \in [69, 72] \wedge \mathit{return}_{rA}.r > 59 \ \vee\ \mathit{pc}.r > 72)\} \\
\mathit{buSet2}(i) &= \{r \mid \mathit{index}.r = i \wedge \mathit{pc}.r = 104 \ \vee\ i_{rA}.r = i \wedge \mathit{index}.r \neq i \wedge \mathit{pc}.r \in [67, 68] \\
&\quad \vee\ i_{nT}.r = i \wedge \mathit{pc}.r \in [82, 84] \ \vee\ i_{mig}.r = i \wedge \mathit{pc}.r \geq 100\}
\end{aligned}
$$

We have the following invariants concerning the `Heap`

He1: $\texttt{Heap}(0) = \bot$

He2: $\texttt{H}(i) \neq 0 \equiv \texttt{Heap}(\texttt{H}(i)) \neq \bot$

He3: $\texttt{Heap}(\texttt{H}(\texttt{currInd})) \neq \bot$

He4: $pc \in [1,58] \vee pc > 65 \wedge \neg(pc \in [67,72] \wedge i_{rA} = index) \;\Rightarrow\; \texttt{Heap}(\texttt{H}(index)) \neq \bot$

He5: $\texttt{Heap}(\texttt{H}(i)) \neq \bot \;\Rightarrow\; \texttt{H}(i).\texttt{size} \geq P$

He6: $\texttt{next}(\texttt{currInd}) \neq 0 \;\Rightarrow\; \texttt{Heap}(\texttt{H}(\texttt{next}(\texttt{currInd}))) \neq \bot$

Invariants concerning hash table pointers

Ha1: $\texttt{H\_index} > 0$

Ha2: $\texttt{H}(i) < \texttt{H\_index}$

Ha3: $i \neq j \wedge \texttt{Heap}(\texttt{H}(i)) \neq \bot \;\Rightarrow\; \texttt{H}(i) \neq \texttt{H}(j)$

Ha4: $index \neq \texttt{currInd} \;\Rightarrow\; \texttt{H}(index) \neq \texttt{H}(\texttt{currInd})$

Invariants about counters for calling the specification.

Cn5: $pc \in [6,7] \;\Rightarrow\; cnt_{fi} = 0$

Cn6: $pc \in [8,13] \;\vee\; pc \in [59,65] \wedge return_{gA} = 10$
$\vee\; pc \in [67,72] \wedge (return_{rA} = 59 \wedge return_{gA} = 10 \;\vee\; return_{rA} = 90 \wedge return_{ref} = 10)$
$\vee\; pc \geq 90 \wedge return_{ref} = 10$
$\Rightarrow\; cnt_{fi} = \sharp(r_{fi} = \mathbf{null} \vee a_{fi} = ADR(r_{fi}))$

Cn7: $pc \in [16,21] \wedge pc \neq 18 \;\vee\; pc \in [59,65] \wedge return_{gA} = 20$
$\vee\; pc \in [67,72] \wedge (return_{rA} = 59 \wedge return_{gA} = 20 \;\vee\; return_{rA} = 90 \wedge return_{ref} = 20)$
$\vee\; pc \geq 90 \wedge return_{ref} = 20$
$\Rightarrow\; cnt_{del} = 0$

Cn8: $pc = 18 \;\Rightarrow\; cnt_{del} = \sharp(r_{del} = \mathbf{null})$

Cn9: $pc \in [28,33] \;\vee\; pc \in [59,65] \wedge return_{gA} = 30$
$\vee\; pc \in [67,72] \wedge (return_{rA} = 59 \wedge return_{gA} = 30 \;\vee\; return_{rA} = 77 \wedge return_{nT} = 30)$
$\vee\; return_{rA} = 90 \wedge return_{ref} = 30$
$\vee\; pc \in [77,84] \wedge return_{nT} = 30 \;\vee\; pc \geq 90 \wedge return_{ref} = 30$
$\Rightarrow\; cnt_{ins} = 0$

Cn10: $pc \in [35,37] \;\vee\; pc \in [59,65] \wedge return_{gA} = 36$
$\vee\; pc \in [67,72] \wedge (return_{rA} = 59 \wedge return_{gA} = 36 \;\vee\; return_{rA} = 90 \wedge return_{ref} = 36)$
$\vee\; pc \geq 90 \wedge return_{ref} = 36$
$\Rightarrow\; cnt_{ins} = \sharp(a_{ins} = ADR(r_{ins}) \vee suc_{ins})$

Cn11: $pc \in [44,52] \vee pc \in [59,65] \wedge return_{gA} \in \{46,51\}$
$\vee\; pc \in [67,72] \wedge (return_{rA} = 59 \wedge return_{gA} \in \{46,51\}$
$\vee\; return_{rA} = 77 \wedge return_{nT} = 46 \;\vee\; return_{rA} = 90 \wedge return_{ref} \in \{46,51\})$
$\vee\; pc \in [77,84] \wedge return_{nT} = 46 \;\vee\; pc \geq 90 \wedge return_{ref} \in \{46,51\}$
$\Rightarrow\; cnt_{ass} = 0$

Invariants about old hash tables, current hash table and the auxiliary hash table Y. Here, we universally quantify over all non-negative integers $n < curSize$.

Cu1: $\mathtt{H}(index) \neq \mathtt{H}(\mathtt{currInd}) \wedge k < \mathtt{H}(index).\mathtt{size}$
$\wedge\ (pc \in [1,58] \vee pc > 65 \wedge \neg(pc \in [67,72] \wedge i_{rA} = index)$
$\Rightarrow\ \mathtt{H}(index).\mathtt{table}[k] = \mathbf{done}$

Cu2: $\sharp(\{k \mid k < curSize \wedge \mathtt{Y}[k] \neq \mathbf{null}\}) < curSize$

Cu3: $\mathtt{H}(\mathtt{currInd}).\mathtt{bound} + 2 * P < curSize$

Cu4: $\mathtt{H}(\mathtt{currInd}).\mathtt{dels} + \sharp(deSet2) = \sharp(deSet1)$

Cu5: *Cu5* has been eliminated, but the numbering has been kept.

Cu6: $\mathtt{H}(\mathtt{currInd}).\mathtt{occ} + \sharp(ocSet1) + \sharp(ocSet2) \leq \mathtt{H}(\mathtt{currInd}).\mathtt{bound} + 2 * P$

Cu7: $\sharp(\{k \mid k < curSize \wedge \mathtt{Y}[k] \neq \mathbf{null}\} = \mathtt{H}(\mathtt{currInd}).\mathtt{occ} + \sharp(ocSet2) + \sharp(ocSet3)$

Cu8: $\mathtt{next}(\mathtt{currInd}) = 0\ \Rightarrow\ \neg\, oldp(\mathtt{H}(\mathtt{currInd}).\mathtt{table}[n])$

Cu9: $\neg(oldp(\mathtt{H}(\mathtt{currInd}).\mathtt{table}[n]))\ \Rightarrow\ \mathtt{H}(\mathtt{currInd}).\mathtt{table}[n] = \mathtt{Y}[n]$

Cu10: $oldp(\mathtt{H}(\mathtt{currInd}).\mathtt{table}[n]) \wedge val(\mathtt{H}(\mathtt{currInd}).\mathtt{table}[n]) \neq \mathbf{null}$
$\Rightarrow\ val(\mathtt{H}(\mathtt{currInd}).\mathtt{table}[n]) = val(\mathtt{Y}[n])$

Cu11: $LeastFind(a,n)\ \Rightarrow\ \mathtt{X}(a) = val(\mathtt{Y}[key(a,\ curSize,\ n)])$

Cu12: $\mathtt{X}(a) = val(\mathtt{Y}[key(a,\ curSize\ ,n)]) \neq \mathbf{null}\ \Rightarrow\ LeastFind(a,\ n)$

Cu13: $\mathtt{X}(a) = val(\mathtt{Y}[key(a,\ curSize,\ n)]) \neq \mathbf{null} \wedge n \neq m < curSize$
$\Rightarrow\ ADR(\mathtt{Y}[key(a,\ curSize,\ m)]) \neq a$

Cu14: $\mathtt{X}(a) = \mathbf{null} \wedge val(\mathtt{Y}[key(a,\ curSize,\ n)]) \neq \mathbf{null}\ \Rightarrow\ ADR(\mathtt{Y}[key(a,\ curSize,\ n)]) \neq a$

Cu15: $\mathtt{X}(a) \neq \mathbf{null}\ \Rightarrow\ \exists m < curSize : \mathtt{X}(a) = val(\mathtt{Y}[key(a,\ curSize,\ m)])$

Cu16: $\exists(f : [\{m : 0 \leq m < curSize) \wedge val(\mathtt{Y}[m]) \neq \mathbf{null}\} \rightarrow$
$\{v : v \neq \mathbf{null} \wedge (\exists k < curSize : v = val(\mathtt{Y}[k]))\}]) :\ f\ \mathtt{is\ bijective}$

Invariants about `next` and `next(currInd)`:

Ne1: $\mathtt{currInd} \neq \mathtt{next}(\mathtt{currInd})$

Ne2: $\mathtt{next}(\mathtt{currInd}) \neq 0\ \Rightarrow\ \mathtt{next}(\mathtt{next}(\mathtt{currInd})) = 0$

Ne3: $pc \in [1,59] \vee pc \geq 62 \wedge pc \neq 65\ \Rightarrow\ index \neq \mathtt{next}(\mathtt{currInd})$

Ne4: $pc \in [1,58] \vee pc \geq 62 \wedge pc \neq 65\ \Rightarrow\ index \neq \mathtt{next}(index)$

Ne5: $pc \in [1,58] \vee pc \geq 62 \wedge pc \neq 65 \wedge \mathtt{next}(index) = 0\ \Rightarrow\ index = \mathtt{currInd}$

Ne6: $\mathtt{next}(\mathtt{currInd}) \neq 0$
$\Rightarrow\ \sharp(ocSet6) \leq \sharp(\{k \mid k < curSize \wedge \mathtt{Y}[k] \neq \mathbf{null}\} - \mathtt{H}(\mathtt{currInd}).\mathtt{dels} - \sharp(deSet2)$

Ne7: $\mathtt{next}(\mathtt{currInd}) \neq 0$
$\Rightarrow\ \mathtt{H}(\mathtt{currInd}).\mathtt{bound} - \mathtt{H}(\mathtt{currInd}).\mathtt{dels} + 2 * P \leq \mathtt{H}(\mathtt{next}(\mathtt{currInd})).\mathtt{bound}$

Ne8: $\mathtt{next}(\mathtt{currInd}) \neq 0$
$\Rightarrow\ \mathtt{H}(\mathtt{next}(\mathtt{currInd})).\mathtt{bound} + 2 * P < \mathtt{H}(\mathtt{next}(\mathtt{currInd})).\mathtt{size}$

Ne9: $\mathtt{next}(\mathtt{currInd}) \neq 0\ \Rightarrow\ \mathtt{H}(\mathtt{next}(\mathtt{currInd})).\mathtt{dels} = \sharp(deSet3)$

Ne9a: $\mathtt{next}(\mathtt{currInd}) \neq 0\ \Rightarrow\ \mathtt{H}(\mathtt{next}(\mathtt{currInd})).\mathtt{dels} = 0$

Ne10: $\mathtt{next}(\mathtt{currInd}) \neq 0 \wedge k < h.\mathtt{size}\ \Rightarrow\ h.\mathtt{table}[k] \notin \{\mathbf{del}, \mathbf{done}\}$,
where $h = \mathtt{H}(\mathtt{next}(\mathtt{currInd}))$

Ne11: $\texttt{next}(\texttt{currInd}) \neq 0 \wedge k < \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{size}$
$\Rightarrow \neg oldp(\texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{table}[k])$

Ne12: $k < curSize \wedge \texttt{H}(\texttt{currInd}).\texttt{table}[k] = \textbf{done} \wedge m < h.\texttt{size} \wedge LeastFind(h,\ a,\ m)$
$\Rightarrow \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)])$,
where $a = ADR(\texttt{Y}[k])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd})))$

Ne13: $k < curSize \wedge \texttt{H}(\texttt{currInd}).\texttt{table}[k] = \textbf{done} \wedge m < h.\texttt{size}$
$\wedge\, \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]) \neq \textbf{null}$
$\Rightarrow LeastFind(h,\ a,\ m)$,
where $a = ADR(\texttt{Y}[k])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne14: $\texttt{next}(\texttt{currInd}) \neq 0 \wedge a \neq \mathbf{0} \wedge k < h.\texttt{size} \wedge \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ k)]) \neq \textbf{null}$
$\Rightarrow LeastFind(h,\ a,\ k)$,
where $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne15: $k < curSize \wedge \texttt{H}(\texttt{currInd}).\texttt{table}[k] = \textbf{done} \wedge \texttt{X}(a) \neq \textbf{null}$
$\wedge\, m < h.\texttt{size} \wedge \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]) \wedge n < h.\texttt{size} \wedge m \neq n$
$\Rightarrow ADR(h.\texttt{table}.[key(a,\ h.\texttt{size},\ n)]) \neq a$,
where $a = ADR(\texttt{Y}[k])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne16: $k < curSize \wedge \texttt{H}(\texttt{currInd}).\texttt{table}[k] = \textbf{done} \wedge \texttt{X}(a) = \textbf{null} \wedge m < h.\texttt{size}$
$\Rightarrow val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]) = \textbf{null}$
$\vee\, ADR(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]) \neq a$,
where $a = ADR(\texttt{Y}[k])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne17: $\texttt{next}(\texttt{currInd}) \neq 0 \wedge m < h.\texttt{size} \wedge a = ADR(h.\texttt{table}[m]) \neq 0$
$\Rightarrow \texttt{X}(a) = val(h.\texttt{table}[m]) \neq \textbf{null}$,
where $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne18: $\texttt{next}(\texttt{currInd}) \neq 0 \wedge m < h.\texttt{size} \wedge a = ADR(h.\texttt{table}[m]) \neq 0$
$\Rightarrow \exists n < curSize : val(\texttt{Y}[n]) = val(h.\texttt{table}[m]) \wedge oldp(\texttt{H}(\texttt{currInd}).\texttt{table}[n])$,
where $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne19: $\texttt{next}(\texttt{currInd}) \neq 0 \wedge m < h.\texttt{size} \wedge m \neq n < h.\texttt{size}$
$\wedge\, a = ADR(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]) \neq 0$
$\Rightarrow ADR(h.\texttt{table}[key(a,\ h.\texttt{size},\ n)]) \neq a$,
where $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne20: $k < curSize \wedge \texttt{H}(\texttt{currInd}).\texttt{table}[k] = \textbf{done} \wedge \texttt{X}(a) \neq \textbf{null}$
$\Rightarrow \exists m < h.\texttt{size} : \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)])$,
where $a = ADR(\texttt{Y}[k])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Ne21: *Ne21* has been eliminated.

Ne22: $\texttt{next}(\texttt{currInd}) \neq 0 \ \Rightarrow\ \sharp(ocSet6) = \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{occ} + \sharp(ocSet7)$

Ne23: $\texttt{next}(\texttt{currInd}) \neq 0$
$\Rightarrow \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{occ} \leq \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{bound}$

Ne24: $\mathtt{next}(\mathtt{currInd}) \neq 0 \;\Rightarrow\; \sharp(ocSet5) \leq \sharp(ocSet4)$

Ne25: $\mathtt{next}(\mathtt{currInd}) \neq 0$
$\Rightarrow\; \exists(f : [\{m : 0 \leq m < h.\mathtt{size} \wedge val(h.\mathtt{table}[m]) \neq \mathbf{null}\} \rightarrow$
$\{v : v \neq \mathbf{null} \wedge (\exists k < h.\mathtt{size} : v = val(h.\mathtt{table}[k]))\}]) : \; f \texttt{ is bijective},$
where $h = \mathtt{H}(\mathtt{next}(\mathtt{currInd}))$

Ne26: $\mathtt{next}(\mathtt{currInd}) \neq 0$
$\Rightarrow\; \exists(f : [\{v : v \neq \mathbf{null} \wedge (\exists m < h.\mathtt{size} : v = val(h.\mathtt{table}[m]))\} \rightarrow$
$\{v : v \neq \mathbf{null} \wedge (\exists k :< curSize : v = val(\mathtt{Y}[k]))\}]) : \; f \texttt{ is injective},$
where $h = \mathtt{H}(\mathtt{next}(\mathtt{currInd}))$

Ne27: $\mathtt{next}(\mathtt{currInd}) \neq 0 \wedge (\exists n < h.\mathtt{size} : val(h.\mathtt{table}[n]) \neq \mathbf{null})$
$\Rightarrow\; \exists(f : [\{m : 0 \leq m < h.\mathtt{size} \wedge val(h.\mathtt{table}[m]) \neq \mathbf{null}\} \rightarrow$
$\{k : 0 \leq k < curSize \wedge val(\mathtt{Y}[k]) \neq \mathbf{null}\}]) : \; f \texttt{ is injective},$
where $h = \mathtt{H}(\mathtt{next}(\mathtt{currInd}))$

Invariants concerning procedure *find* (5. . . 14)

fi1: $a_{fi} \neq \mathbf{0}$

fi2: $pc \in \{6, 11\} \;\Rightarrow\; n_{fi} = 0$

fi3: $pc \in \{7, 8, 13\} \;\Rightarrow\; l_{fi} = h_{fi}.\mathtt{size}$

fi4: $pc \in [6, 13] \wedge pc \neq 10 \;\Rightarrow\; h_{fi} = \mathtt{H}(index)$

fi5: $pc = 7 \wedge h_{fi} = \mathtt{H}(\mathtt{currInd}) \;\Rightarrow\; n_{fi} < curSize$

fi6: $pc = 8 \wedge h_{fi} = \mathtt{H}(\mathtt{currInd}) \wedge \neg Find(r_{fi},\ a_{fi}) \wedge r_{fi} \neq \mathbf{done}$
$\Rightarrow\; \neg\ Find(\mathtt{Y}[key(a_{fi},\ curSize,\ n_{fi})],\ a_{fi})$

fi7: $pc = 13 \wedge h_{fi} = \mathtt{H}(\mathtt{currInd}) \wedge \neg Find(r_{fi},\ a_{fi}) \wedge m < n_{fi}$
$\Rightarrow\; \neg Find(\mathtt{Y}[key(a_{fi},\ curSize,\ m)],\ a_{fi})$

fi8: $pc \in \{7, 8\} \wedge h_{fi} = \mathtt{H}(\mathtt{currInd}) \wedge m < n_{fi} \;\Rightarrow\; \neg Find(\mathtt{Y}[key(a_{fi},\ curSize,\ m)],\ a_{fi})$

fi9: $pc = 7 \wedge Find(t,\ a_{fi}) \;\Rightarrow\; \mathtt{X}(a_{fi}) = val(t),$
where $t = h_{fi}.\mathtt{table}[key(a_{fi},\ l_{fi},\ n_{fi})]$

fi10: $pc \notin (1, 7] \wedge Find(r_{fi},\ a_{fi}) \;\Rightarrow\; val(r_{fi}) = rS_{fi}$

fi11: $pc = 8 \wedge oldp(r_{fi}) \wedge index = \mathtt{currInd} \;\Rightarrow\; \mathtt{next}(\mathtt{currInd}) \neq 0$

Invariants concerning procedure *delete* (15. . . 26)

de1: $a_{del} \neq \mathbf{0}$

de2: $pc \in \{17, 18\} \;\Rightarrow\; l_{del} = h_{del}.\mathtt{size}$

de3: $pc \in [16, 25] \wedge pc \neq 20 \;\Rightarrow\; h_{del} = \mathtt{H}(index)$

de4: $pc = 18 \;\Rightarrow\; k_{del} = key(a_{del},\ l_{del},\ n_{del})$

de5: $pc \in \{16, 17\} \vee Deleting \;\Rightarrow\; \neg suc_{del}$

de6: $Deleting \wedge sucS_{del} \;\Rightarrow\; r_{del} \neq \mathbf{null}$

de7: $pc = 18 \wedge \neg\ oldp(h_{del}.\mathtt{table}[k_{del}]) \;\Rightarrow\; h_{del} = \mathtt{H}(\mathtt{currInd})$

de8: $pc \in \{17, 18\} \wedge h_{del} = \mathtt{H}(\mathtt{currInd}) \;\Rightarrow\; n_{del} < curSize$

de9: $pc = 18 \wedge h_{del} = \mathtt{H}(\mathtt{currInd}) \wedge (val(r_{del}) \neq \mathbf{null} \vee r_{del} = \mathbf{del})$
$\Rightarrow\; r \neq \mathbf{null} \wedge (r = \mathbf{del} \vee ADR(r) = ADR(r_{del}))$,
where $r = \mathtt{Y}[key(a_{del},\ h_{del}.size,\ n_{del})]$

de10: $pc \in \{17, 18\} \wedge h_{del} = \mathtt{H}(\mathtt{currInd}) \wedge m < n_{del}) \;\Rightarrow\; \neg Find(\mathtt{Y}[key(a_{del},\ curSize,\ m)],\ a_{del})$

de11: $pc \in \{17, 18\} \wedge Find(t,\ a_{del}) \;\Rightarrow\; \mathtt{X}(a_{del}) = val(t)$,
where $t = h_{del}.\mathtt{table}[key(a_{del},\ l_{del},\ n_{del})]$

de12: $pc = 18 \wedge oldp(r_{del}) \wedge index = \mathtt{currInd} \;\Rightarrow\; \mathtt{next}(\mathtt{currInd}) \neq 0$

de13: $pc = 18 \;\Rightarrow\; k_{del} < \mathtt{H}(index).\mathtt{size}$

Deleting is characterized by

$Deleting \;\equiv\; pc \in [18, 21] \vee pc \in [59, 65] \wedge return_{gA} = 20$
$\vee\; pc \in [67, 72] \wedge (return_{rA} = 59 \wedge return_{gA} = 20 \vee return_{rA} = 90 \wedge return_{ref} = 20)$
$\vee\; pc \geq 90 \wedge return_{ref} = 20$

Invariants concerning procedure *insert* (27. . . 52)

in1: $a_{ins} = ADR(v_{ins}) \wedge v_{ins} \neq \mathbf{null}$

in2: $pc \in [32, 35] \;\Rightarrow\; l_{ins} = h_{ins}.\mathtt{size}$

in3: $pc \in [28, 41] \wedge pc \notin \{30, 36\} \;\Rightarrow\; h_{ins} = \mathtt{H}(index)$

in4: $pc \in \{33, 35\} \;\Rightarrow\; k_{ins} = key(a_{ins},\ l_{ins},\ n_{ins})$

in5: $pc \in [32, 33] \vee Inserting \;\Rightarrow\; \neg suc_{ins}$

in6: $Inserting \wedge sucS_{ins} \;\Rightarrow\; ADR(r_{ins}) \neq a_{ins}$

in7: $pc = 35 \wedge \neg\, oldp(h_{ins}.\mathtt{table}[k_{ins}]) \;\Rightarrow\; h_{ins} = \mathtt{H}(\mathtt{currInd})$

in8: $pc \in \{33, 35\} \wedge h_{ins} = \mathtt{H}(\mathtt{currInd}) \;\Rightarrow\; n_{ins} < curSize$

in9: $pc = 35 \wedge h_{ins} = \mathtt{H}(\mathtt{currInd}) \wedge (val(r_{ins}) \neq \mathbf{null} \vee r_{ins} = \mathbf{del})$
$\Rightarrow\; r \neq \mathbf{null} \wedge (r = \mathbf{del} \vee ADR(r) = ADR(r_{ins}))$,
where $r = \mathtt{Y}[key(a_{ins},\ h_{ins}.size,\ n_{ins})]$

in10: $pc \in \{32, 33, 35\} \wedge h_{ins} = \mathtt{H}(\mathtt{currInd}) \wedge m < n_{ins}$
$\Rightarrow\; \neg Find(\mathtt{Y}[key(a_{ins},\ curSize,\ m)],\ a_{ins})$

in11: $pc \in \{33, 35\} \wedge Find(t,\ a_{ins}) \;\Rightarrow\; \mathtt{X}(a_{ins}) = val(t)$,
where $t = h_{ins}.\mathtt{table}[key(a_{ins},\ l_{ins},\ n_{ins})]$

in12: $pc = 35 \wedge oldp(r_{ins}) \wedge index = \mathtt{currInd} \;\Rightarrow\; \mathtt{next}(\mathtt{currInd}) \neq 0$

in13: $pc = 35 \;\Rightarrow\; k_{ins} < \mathtt{H}(index).\mathtt{size}$

Inserting is characterized by

$Inserting \;\equiv\; pc \in [35, 37] \vee pc \in [59, 65] \wedge return_{gA} = 36$
$\vee\; pc \in [67, 72] \wedge (return_{rA} = 59 \wedge return_{gA} = 36 \;\vee\; return_{rA} = 90 \wedge return_{ref} = 36)$
$\vee\; pc \geq 90 \wedge return_{ref} = 36$

Invariants concerning procedure *assign* (43. . . 57)

as1: $a_{ass} = ADR(v_{ass}) \wedge v_{ass} \neq \mathbf{null}$

as2: $pc \in [48, 50] \;\Rightarrow\; l_{ass} = h_{ass}.\mathtt{size}$

as3: $pc \in [44, 57] \wedge pc \notin \{46, 51\} \;\Rightarrow\; h_{ass} = \mathtt{H}(index)$

as4: $pc \in \{49, 50\} \;\Rightarrow\; k_{ass} = key(a_{ass},\ l_{ass},\ n_{ass})$

as5: $pc = 50 \wedge \neg\ oldp(h_{ass}.\mathtt{table}[k_{ass}]) \;\Rightarrow\; h_{ass} = \mathtt{H}(\mathtt{currInd})$

as6: $pc = 50 \wedge h_{ass} = \mathtt{H}(\mathtt{currInd}) \;\Rightarrow\; n_{ass} < curSize$

as7: $pc = 50 \wedge h_{ass} = \mathtt{H}(\mathtt{currInd}) \wedge (val(r_{ass}) \neq \mathbf{null} \vee r_{ass} = \mathbf{del})$
$\Rightarrow\; r \neq \mathbf{null} \wedge (r = \mathbf{del} \vee ADR(r) = ADR(r_{ass}))$,
where $r = \mathtt{Y}[key(a_{ass},\ h_{ass}.size,\ n_{ass})]$

as8: $pc \in \{48, 49, 50\} \wedge h_{ass} = \mathtt{H}(\mathtt{currInd}) \wedge m < n_{ass}$
$\Rightarrow\; \neg Find(\mathtt{Y}[key(a_{ass},\ curSize,\ m)],\ a_{ass})$

as9: $pc = 50 \wedge Find(t,\ a_{ass}) \;\Rightarrow\; \mathtt{X}(a_{ass}) = val(t)$,
where $t = h_{ass}.\mathtt{table}[key(a_{ass},\ l_{ass},\ n_{ass})]$

as10: $pc = 50 \wedge oldp(r_{ass} sign) \wedge index = \mathtt{currInd} \;\Rightarrow\; \mathtt{next}(\mathtt{currInd}) \neq 0$

as11: $pc = 50 \;\Rightarrow\; k_{ass} < \mathtt{H}(index).\mathtt{size}$

Invariants concerning procedure *releaseAccess* (67. . . 72)

rA1: $h_{rA} < \mathtt{H\_index}$

rA2: $pc \in [70, 71] \;\Rightarrow\; h_{rA} \neq 0$

rA3: $pc = 71 \;\Rightarrow\; \mathtt{Heap}(h_{rA}) \neq \perp$

rA4: $pc = 71 \;\Rightarrow\; \mathtt{H}(i_{rA}) = 0$

rA5: $pc = 71 \;\Rightarrow\; h_{rA} \neq \mathtt{H}(i)$

rA6: $pc = 70 \;\Rightarrow\; \mathtt{H}(i_{rA}) \neq \mathtt{H}(\mathtt{currInd})$

rA7: $pc = 70 \wedge (pc.r \in [1, 58] \vee pc.r > 65 \wedge \neg(pc.r \in [67, 72] \wedge i_{rA}.r = index.r))$
$\Rightarrow\; \mathtt{H}(i_{rA}) \neq \mathtt{H}(index.r)$

rA8: $pc = 70 \;\Rightarrow\; i_{rA} \neq \mathtt{next}(\mathtt{currInd})$

rA9: $pc \in [68, 72] \wedge\ (h_{rA} = 0 \vee h_{rA} \neq \mathtt{H}(i_{rA})) \;\Rightarrow\; \mathtt{H}(i_{rA}) = 0$

rA10: $pc \in [67, 72] \wedge return_{rA} \in \{0, 59\} \;\Rightarrow\; i_{rA} = index$

rA11: $pc \in [67, 72] \wedge return_{rA} \in \{77, 90\} \;\Rightarrow\; i_{rA} \neq index$

rA12: $pc \in [67, 72] \wedge return_{rA} = 77 \;\Rightarrow\; \mathtt{next}(index) \neq 0$

rA13: $pc = 71 \wedge pc.r = 71 \wedge p \neq r \;\Rightarrow\; h_{rA} \neq h_{rA}.r$

rA14: $pc = 71 \wedge pc.r = 71 \wedge p \neq r \;\Rightarrow\; i_{rA} \neq i_{rA}.r$

Invariants concerning procedure *newTable* (77. . . 84)

nT1: $pc \in [81, 82] \;\Rightarrow\; \mathtt{Heap}(\mathtt{H}(i_{nT})) = \perp$

nT2: $pc \in [83, 84] \;\Rightarrow\; \mathtt{Heap}(\mathtt{H}(i_{nT})) \neq \perp$

nT3: $pc = 84 \;\Rightarrow\; \texttt{next}(i_{nT}) = 0$

nT4: $pc \in [83, 84] \;\Rightarrow\; \texttt{H}(i_{nT}).\texttt{dels} = 0$

nT5: $pc \in [83, 84] \;\Rightarrow\; \texttt{H}(i_{nT}).\texttt{occ} = 0$

nT6: $pc \in [83, 84] \;\Rightarrow\; \texttt{H}(i_{nT}).\texttt{bound} + 2 * P < \texttt{H}(i_{nT}).\texttt{size}$

nT7: $pc \in [83, 84] \land index = \texttt{currInd}$
$\Rightarrow\; \texttt{H}(\texttt{currInd}).\texttt{bound} - \texttt{H}(\texttt{currInd}).\texttt{dels} + 2 * P < \texttt{H}(i_{nT}).\texttt{bound}$

nT8: $pc \in [83, 84] \land k < \texttt{H}(i_{nT}).\texttt{size} \;\Rightarrow\; \texttt{H}(i_{nT}).\texttt{table}[k] = \textbf{null}$

nT9: $pc \in [81, 84] \;\Rightarrow\; i_{nT} \neq \texttt{currInd}$

nT10: $pc \in [81, 84] \land (pc.r \in [1, 58] \lor pc.r \geq 62 \land pc.r \neq 65)$ *Implies* $i_{nT} \neq index.r$

nT11: $pc \in [81, 84] \;\Rightarrow\; i_{nT} \neq \texttt{next}(\texttt{currInd})$

nT12: $pc \in [81, 84] \;\Rightarrow\; \texttt{H}(i_{nT}) \neq \texttt{H}(\texttt{currInd})$

nT13: $pc \in [81, 84] \land (pc.r \in [1, 58] \lor pc.r > 65 \land \neg(pc.r \in [67, 72] \land i_{rA}.r = index.r))$
$\Rightarrow\; \texttt{H}(i_{nT}) \neq \texttt{H}(index.r)$

nT14: $pc \in [81, 84] \land pc.r \in [67, 72] \;\Rightarrow\; i_{nT} \neq i_{rA}.r$

nT15: $pc \in [83, 84] \land pc.r \in [67, 72] \;\Rightarrow\; \texttt{H}(i_{nT}) \neq \texttt{H}(i_{rA}.r)$

nT16: $pc \in [81, 84] \land pc.r \in [81, 84] \land p \neq r \;\Rightarrow\; i_{nT} \neq i_{nT}.r$

nT17: $pc \in [81, 84] \land pc.r \in [95, 99] \land index.r = \texttt{currInd} \;\Rightarrow\; i_{nT} \neq i_{mig}.r$

nT18: $pc \in [81, 84] \land pc.r \geq 99 \;\Rightarrow\; i_{nT} \neq i_{mig}.r$

Invariants concerning procedure *migrate* (94. . . 105)

mi1: $pc = 98 \lor pc \in \{104, 105\} \;\Rightarrow\; index \neq \texttt{currInd}$

mi2: $pc \geq 95 \;\Rightarrow\; i_{mig} \neq index$

mi3: $pc = 94 \;\Rightarrow\; \texttt{next}(index) > 0$

mi4: $pc \geq 95 \;\Rightarrow\; i_{mig} \neq 0$

mi5: $pc \geq 95 \;\Rightarrow\; i_{mig} = \texttt{next}(index)$

mi6: $pc.r = 70 \land (pc \in [95, 102) \land index = \texttt{currInd} \lor pc \in [102, 103] \lor pc \geq 110)$
$\Rightarrow\; i_{rA}.r \neq i_{mig}$

mi7: $pc \in [95, 97] \land index = \texttt{currInd} \lor pc \geq 99 \;\Rightarrow\; i_{mig} \neq \texttt{next}(i_{mig})$

mi8: $(pc \in [95, 97] \lor pc \in [99, 103] \lor pc \geq 110) \land index = \texttt{currInd}$
$\Rightarrow\; \texttt{next}(i_{mig}) = 0$

mi9: $(pc \in [95, 103] \lor pc \geq 110) \land index = \texttt{currInd} \;\Rightarrow\; \texttt{H}(i_{mig}) \neq \texttt{H}(\texttt{currInd})$

mi10: $(pc \in [95, 103] \lor pc \geq 110) \land index = \texttt{currInd} \land (pc.r \in [1, 58] \lor pc.r \geq 62 \land pc.r \neq 65)$
$\Rightarrow\; \texttt{H}(i_{mig}) \neq \texttt{H}(index.r)$

mi11: $pc = 101 \land index = \texttt{currInd} \lor pc = 102 \;\Rightarrow\; h_{mig} = \texttt{H}(i_{mig})$

mi12: $pc \geq 95 \land index = \texttt{currInd} \lor pc \in \{102, 103\} \lor pc \geq 110 \;\Rightarrow\; \texttt{Heap}(\texttt{H}(i_{mig})) \neq \perp$

mi13: $pc = 103 \land index = \texttt{currInd} \land k < curSize \;\Rightarrow\; \texttt{H}(index).\texttt{table}[k] = \textbf{done}$

mi14: $pc = 103 \land index = \texttt{currInd} \land n < \texttt{H}(i_{mig}).\texttt{size} \land LeastFind(\texttt{H}(i_{mig}),\ a,\ n)$
$\Rightarrow\; \texttt{X}(a) = val(\texttt{H}(i_{mig})[key(a,\ \texttt{H}(i_{mig}).\texttt{size},\ n)])$

mi15: $pc = 103 \wedge index = \texttt{currInd} \wedge n < \texttt{H}(i_{mig}).\texttt{size}$
$\wedge\ \texttt{X}(a) = val(\texttt{H}(i_{mig}).\texttt{table}[key(a,\ \texttt{H}(i_{mig}).\texttt{size},\ n)] \neq \textbf{null}$
$\Rightarrow\ LeastFind(\texttt{H}(i_{mig}),\ a,\ n)$

mi16: $pc = 103 \wedge index = \texttt{currInd} \wedge k < \texttt{H}(i_{mig}).\texttt{size}$
$\Rightarrow\ \neg oldp(\texttt{H}(i_{mig}).\texttt{table}[k])$

mi17: $pc = 103 \wedge index = \texttt{currInd} \wedge \texttt{X}(a) \neq \textbf{null} \wedge k < h.\texttt{size}$
$\wedge\ \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ k)]) \wedge k \neq n < h.\texttt{size}$
$\Rightarrow\ ADR(h.\texttt{table}.[key(a,\ h.\texttt{size},\ n)]) \neq a,$
where $h = \texttt{H}(i_{mig})$

mi18: $pc = 103 \wedge index = \texttt{currInd} \wedge \texttt{X}(a) = \textbf{null} \wedge k < h.\texttt{size}$
$\Rightarrow\ val(h.\texttt{table}[key(a,\ h.\texttt{size},\ k)]) = \textbf{null} \vee ADR(h.\texttt{table}[key(a,\ h.\texttt{size},\ k)]) \neq a,$
where $h = \texttt{H}(i_{mig})$

mi19: $pc = 103 \wedge index = \texttt{currInd} \wedge \texttt{X}(a) \neq \textbf{null}$
$\Rightarrow\ \exists m < h.\texttt{size} : \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)],$
where $h = \texttt{H}(i_{mig})$

mi20: $pc = 117 \wedge \texttt{X}(a) \neq \textbf{null} \wedge val(\texttt{H}(index).\texttt{table}[i_{mC}]) \neq \textbf{null}$
$\vee\ pc \geq 126 \wedge \texttt{X}(a) \neq \textbf{null} \wedge index = \texttt{currInd}$
$\vee\ pc = 125 \wedge \texttt{X}(a) \neq \textbf{null} \wedge index = \texttt{currInd}$
$\wedge\ (b_{mE} \vee val(w_{mE}) \neq \textbf{null} \wedge a_{mE} = ADR(w_{mE}))$
$\Rightarrow\ \exists m < h.\texttt{size} : \texttt{X}(a) = val(h.\texttt{table}[key(a,\ h.\texttt{size},\ m)]),$
where $a = ADR(\texttt{Y}[i_{mC}])$ and $h = \texttt{H}(\texttt{next}(\texttt{currInd}))$

Invariants concerning procedure *moveContents* (110. . . 118):

mC1: $pc = 103 \vee pc \geq 110\ \Rightarrow\ to = \texttt{H}(i_{mig})$

mC2: $pc \geq 110\ \Rightarrow\ from = \texttt{H}(index)$

mC3: $pc > 102 \wedge m \in toBeMoved\ \Rightarrow\ m < \texttt{H}(index).\texttt{size}$

mC4: $pc = 111\ \Rightarrow\ \exists m < from.\texttt{size} : m \in toBeMoved$

mC5: $pc \geq 114 \wedge pc \neq 118\ \Rightarrow\ v_{mC} \neq \textbf{done}$

mC6: $pc \geq 114\ \Rightarrow\ i_{mC} < \texttt{H}(index).\texttt{size}$

mC7: $pc = 118\ \Rightarrow\ \texttt{H}(index).\texttt{table}[i_{mC}] = \textbf{done}$

mC8: $pc \geq 110 \wedge k < \texttt{H}(index).\texttt{size} \wedge k \notin toBeMoved\ \Rightarrow\ \texttt{H}(index).\texttt{table}[k] = \textbf{done}$

mC9: $pc \geq 110 \wedge index = \texttt{currInd} \wedge toBeMoved = \emptyset \wedge k < \texttt{H}(index).\texttt{size}$
$\Rightarrow\ \texttt{H}(index).\texttt{table}[k] = \textbf{done}$

mC10: $pc \geq 116 \wedge val(v_{mC}) \neq \textbf{null} \wedge \texttt{H}(index).\texttt{table}[i_{mC}] = \textbf{done}$
$\Rightarrow\ \texttt{H}(i_{mig}).\texttt{table}[key(a,\ \texttt{H}(i_{mig}).\texttt{size},\ 0)] \neq \textbf{null},$
where $a = ADR(v_{mC})$

mC11: $pc \geq 116 \wedge \texttt{H}(index).\texttt{table}[i_{mC}] \neq \textbf{done}$
$\Rightarrow\ val(v_{mC}) = val(\texttt{H}(index).\texttt{table}[i_{mC}]) \wedge oldp(\texttt{H}(index).\texttt{table}[i_{mC}])$

mC12: $pc \geq 116 \wedge index = \texttt{currInd} \wedge val(v_{mC}) \neq \textbf{null}$
$\Rightarrow\ val(v_{mC}) = val(\texttt{Y}[i_{mC}])$

Invariants concerning procedure *moveElement* (120...126):

mE1: $pc \geq 120 \ \Rightarrow\ val(v_{mC}) = v_{mE}$
mE2: $pc \geq 120 \ \Rightarrow\ v_{mE} \neq \textbf{null}$
mE3: $pc \geq 120 \ \Rightarrow\ to = \texttt{H}(i_{mig})$
mE4: $pc \geq 121 \ \Rightarrow\ a_{mE} = ADR(v_{mC})$
mE5: $pc \geq 121 \ \Rightarrow\ m_{mE} = to.\texttt{size}$
mE6: $pc \in \{121, 123\} \ \Rightarrow\ \neg b_{mE}$
mE7: $pc = 123 \ \Rightarrow\ k_{mE} = key(a_{mE},\ to.\texttt{size},\ n_{mE})$
mE8: $pc \geq 123 \ \Rightarrow\ k_{mE} < \texttt{H}(i_{mig}).\texttt{size}$
mE9: $pc = 120 \wedge to.\texttt{table}[key(ADR(v_{mE}),\ to.\texttt{size},\ 0)] = \textbf{null}$
$\Rightarrow\ index = \texttt{currInd}$
mE10: $pc \in \{121, 123\} \wedge to.\texttt{table}[key(a_{mE},\ to.\texttt{size},\ n_{mE})] = \textbf{null}$
$\Rightarrow\ index = \texttt{currInd}$
mE11: $pc \in \{121, 123\} \wedge pc.r = 103 \wedge to.\texttt{table}[key(a_{mE},\ to.\texttt{size},\ n_{mE})] = \textbf{null}$
$\Rightarrow\ index.r \neq \texttt{currInd}$
mE12: $pc \in \{121, 123\} \wedge \texttt{next}(\texttt{currInd}) \neq 0 \wedge to = \texttt{H}(\texttt{next}(\texttt{currInd}))$
$\Rightarrow\ n_{mE} < \texttt{H}(\texttt{next}(\texttt{currInd})).\texttt{size}$
mE13: $pc \in \{123, 125\} \wedge w_{mE} \neq \textbf{null}$
$\Rightarrow\ ADR(w_{mE}) = ADR(to.\texttt{table}[k_{mE}]) \vee to.\texttt{table}[k_{mE}] \in \{\textbf{del},\ \textbf{done}\}$
mE14: $pc \geq 123 \wedge w_{mE} \neq \textbf{null} \ \Rightarrow\ \texttt{H}(i_{mig}).\texttt{table}[k_{mE}] \neq \textbf{null}$
mE15: $pc = 117 \wedge val(v_{mC}) \neq \textbf{null} \ \vee\ pc \in \{121, 123\} \wedge n_{mE} > 0 \ \vee\ pc = 125$
$\Rightarrow\ h.\texttt{table}[key(ADR(v_{mC}),\ h.\texttt{size},\ 0)] \neq \textbf{null}$,
where $h = \texttt{H}(i_{mig})$
mE16: $pc \in \{121, 123\}$
$\vee\ (pc = 125 \wedge \neg b_{mE} \wedge (val(w_{mE}) = \textbf{null} \vee a_{mE} \neq ADR(w_{mE})))$
$\Rightarrow\ \forall m < n_{mE} :\ \neg Find(to.\texttt{table}[key(a_{mE},\ to.\texttt{size},\ m)],\ a_{mE})$

Invariants about the integer array `prot`.

pr1: $\texttt{prot}[i] = \sharp(prSet1(i)) + \sharp(prSet2(i)) + \sharp(\texttt{currInd} = i) + \sharp(\texttt{next}(\texttt{currInd}) = i)$
pr2: $\texttt{prot}[\texttt{currInd}] > 0$
pr3: $pc \in [1, 58] \vee pc \geq 62 \wedge pc \neq 65 \ \Rightarrow\ \texttt{prot}[index] > 0$
pr4: $\texttt{next}(\texttt{currInd}) \neq 0 \ \Rightarrow\ \texttt{prot}[\texttt{next}(\texttt{currInd})] > 0$
pr5: $\texttt{prot}[i] = 0 \ \Rightarrow\ \texttt{Heap}(\texttt{H}[i]) = \bot$
pr6: $\texttt{prot}[i] \leq \sharp(prSet3(i)) \wedge \texttt{busy}[i] = 0 \ \Rightarrow\ \texttt{Heap}(\texttt{H}[i]) = \bot$
pr7: $pc \in [67, 72] \ \Rightarrow\ \texttt{prot}[i_{rA}] > 0$

pr8: $pc \in [81, 84] \Rightarrow \texttt{prot}[i_{nT}] > 0$
pr9: $pc \geq 97 \Rightarrow \texttt{prot}[i_{mig}] > 0$
pr10: $pc \in [81, 82] \Rightarrow \texttt{prot}[i_{nT}] = \sharp(prSet4(i_{nT})) + 1$

Invariants about the integer array `busy`.

bu1: $\texttt{busy}[i] = \sharp(buSet1(i)) + \sharp(buSet2(i)) + \sharp(\texttt{currInd} = i) + \sharp(\texttt{next}(\texttt{currInd}) = i)$
bu2: $\texttt{busy}[\texttt{currInd}] > 0$
bu3: $pc \in [1, 58] \lor pc > 65 \land \neg(i_{rA} = index \land pc \in [67, 72])$
$\Rightarrow \texttt{busy}[index] > 0$
bu4: $\texttt{next}(\texttt{currInd}) \neq 0 \Rightarrow \texttt{busy}[\texttt{next}(\texttt{currInd})] > 0$
bu5: $pc = 81 \Rightarrow \texttt{busy}[i_{nT}] = 0$
bu6: $pc \geq 100 \Rightarrow \texttt{busy}[i_{mig}] > 0$

Some other invariants we have postulated:

Ot1: $\texttt{X}(\mathbf{0}) = \textbf{null}$
Ot2: $\texttt{X}(a) \neq \textbf{null} \Rightarrow ADR(\texttt{X}(a)) = a$

The motivation of invariant (Ot1) is that we never store a value for the address 0. The motivation of invariant (Ot2) is that the address in the hash table is unique.

Ot3: $return_{gA} = \{1, 10, 20, 30, 36, 46, 51\} \land return_{rA} = \{0, 59, 77, 90\}$
$\land\ return_{ref} = \{10, 20, 30, 36, 46, 51\} \land return_{nT} = \{30, 46\}$
Ot4: $pc \in \{0, 1, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16, 17, 18, 20, 21, 25, 26, 27, 28, 30, 31, 32, 33, 35, 36, 37, 41, 42, 43, 44, 46, 47, 48, 49, 50, 51, 52, 57, 59, 60, 61, 62, 63, 65, 67, 68, 69, 70, 71, 72, 77, 78, 81, 82, 83, 84, 90, 94, 95, 97, 98, 99, 100, 101, 102, 103, 104, 105, 110, 111, 114, 116, 117, 118, 120, 121, 123, 125, 126\}$

A.2 Dependencies between invariants

Let us write "φ **from** $\psi_1, \cdots, \psi_n$" to denote that φ is proved to be an invariant using that $\psi_1, \ldots, \psi_n$ hold. We write "$\varphi \Leftarrow \psi_1, \cdots, \psi_n$" to denote that predicate φ is implied by the conjunction of $\psi_1, \ldots, \psi_n$. We have verified the following "**from**" and "$\Leftarrow$" relations mechanically:

Co1 **from** fi10, Ot3, fi1
Co2 **from** de5, Ot3, de6, del, de11
Co3 **from** in5, Ot3, in6, in1, in11

Cn1 **from** Cn6, Ot3
Cn2 **from** Cn8, Ot3, del
Cn3 **from** Cn10, Ot3, in1, in5
Cn4 **from** Cn11, Ot3
No1 ⇐ No2
No2 **from** nT1, He2, rA2, Ot3, Ha2, Ha1, rA1, rA14, rA3, nT14, rA4
He1 **from** Ha1
He2 **from** Ha3, rA5, Ha1, He1, rA2
He3, He4 **from** Ot3, rA6, rA7, mi12, rA11, rA5
He5 **from** He1
He6 **from** rA8, Ha3, mi8, nT2, rA5
Ha1 **from** true
Ha2 **from** Ha1
Ha3 **from** Ha2, Ha1, He2, He1
Ha4 ⇐ Ha3, He3, He4
Cn5 **from** Cn6, Ot3
Cn6 **from** Cn5, Ot3
Cn7 **from** Cn8, Ot3, del
Cn8 **from** Cn7, Ot3
Cn9 **from** Cn10, Ot3, in1, in5
Cn10 **from** Cn9, Ot3, in5
Cn11 **from** Cn11, Ot3
Cu1 **from** Ot3, Ha4, rA6, rA7, nT13, nT12, Ha2, He3, He4, rA11, nT9, nT10, mi13, rA5
Cu2 ⇐ Cu6, cu7, Cu3, He3, He4
Cu3 **from** rA6, rA7, nT13, nT12, mi5, mi4, Ne8, rA5
Cu4 **from** del, in1, as1, rA6, rA7, Ha2, nT13, nT12, Ne9, Cu9, Cu10, de7, in7, as5, He3, He4, mi5, mi4, Ot3, Ha4, de3, mi9, mi10, de5, rA5
Cu6 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, Ha3, in3, as3, Ne23, mi5, mE6, mE7, mE10, mE3, Ne3, mi1, mi4, rA5
Cu7 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, Ha3, in3, as3, in5, mi5, mE6, mE7, mE10, mi4, mE3, Ne3, de7, in7, as5, Ne22, mi9, mi10, rA5, He3, mi12, mi1, Cu9, de1, in1, as1
Cu8 **from** Cu8, Ha2, nT9, nT10, rA6, rA7, mi5, mi4, mC2, mC5, He3, He4, Cu1, Ha4, mC6, mi16, rA5
Cu9, Cu10 **from** rA6, rA7, nT13, nT12, Ha2, He3, He4, Cu1, Ha4, de3, in3, as3, mE3, mi9, mi10, mE10, mE7, rA5
Cu11, Cu12 **from** Cu9, Cu10, Cu13, Cu14, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, He3, He4, Cu1, Ha4, in3, as3, mi14, mi15, de3, in10, as8, mi12, Ot2, fi5, de8, in8,

as6, Cu15, de11, in11, rA5

Cu13, Cu14 **from** He3, He4, Ot2, del, in1, as1, Ot1, rA6, rA7, nT13, nT12, Ha2, Cu9, Cu10, Cu1, Ha4, de3, in3, as3, Cu11, Cu12, in10, as8, fi5, de8, in8, as6, Cu15, mi17, mi18, mi12, mi4, de11, rA5

Cu15 **from** He3, He4, rA6, rA7, nT13, nT12, Ha2, Cu1, Ha4, del, in1, as1, de3, in3, as3, fi5, de8, in8, as6, mi12, mi19, mi4, Ot2, Cu9, Cu10, Cu11, Cu12, Cu13, Cu14, rA5

Cu16 ⇐ Cu13, Cu14, Cu15, He3, He4, Ot1

Ne1 **from** nT9, nT10, mi7

Ne2 **from** Ne5, nT3, mi8, nT9, nT10

Ne3 **from** Ne1, nT9, nT10, mi8

Ne4 **from** Ne1, nT9, nT10

Ne5 **from** Ot3, nT9, nT10, mi5

Ne6 ⇐ Ne10, Ne24, He6, He3, He4, Cu4

Ne7 **from** Ha3, rA6, rA7, rA8, nT13, nT12, nT11, He3, He4, mi8, nT7, Ne5, Ha2, He6, rA5

Ne8 **from** Ha3, rA8, nT11, mi8, nT6, Ne5, rA5

Ne9 **from** Ha3, Ha2, Ne3, Ne5, de3, as3, rA8, rA6, rA7, nT8, nT11, mC2, nT4, mi8, rA5

Ne9a **from** Ha3, Ne3, rA5, de3, rA8, nT4, mi8

Ne10 **from** Ha3, Ha2, de3, rA8, nT11, Ne3, He6, mi8, nT8, mC2, nT2, Ne5, rA5

Ne11 **from** Ha3, Ha2, He6, nT2, nT8, rA8, nT11, mi8, Ne3, mC2, rA5

Ne12, Ne13 **from** Ha3, Ha2, Cu8, He6, He3, He4, Cu1, de3, in3, as3, rA8, rA6, rA7, nT11, nT13, nT12, mi12, mi16, mi5, mi4, de7, in7, as5, Ot2, del,in1, as1, Cu9, Cu10, Cu13, Cu14, Cu15, as9, fi5, de8, in8, as6, mC2, Ne3, Ot1, Ne14, Ne20, mE16, mE7, mE4, mE1, mE12, mE2, Ne15, Ne16, Ne17, Ne18, mi20, de11, in11, rA5

Ne14 **from** Ha3, Ha2, He6, He3, He4, nT2, nT8, de3, in3, as3, rA8, nT11, Ot2, del, in1, as1, Cu9, Cu10, mi8, Ne3, mC2, mE7, mE16, mE1, mE4, mE12, Ne17, Ne18, Cu1, rA5

Ne15, Ne16 **from** Ha3, Ha2, Cu8, He6, He3, He4, Cu1, de3, in3, as3, rA8, rA6, rA7, nT11, nT13, nT12, mi12, mi16, mi5, mi4, de7, in7, as5, Ot2, del, in1, as1, Cu9, Cu10, Cu13, Cu14, Cu15, as9, fi5, de8, in8, as6, mC2, Ne3, Ot1, Ne19, Ne20, Ne12, Ne13, mE16, mE7, mE4, mE1, mE12, mE10, mE2, in11, de11, rA5

Ne17, Ne18 **from** Ha3, Ha2, mi8, He6, He3, He4, Cu1, nT2, de3, in3, as3, rA8, rA6, rA7, nT11, nT13, nT12, de7, in7, as5, Ot2, del, in1, as1, Cu9, Cu10, nT8, mE2, fi5, de8, in8, as6, mC2, Ne3, mC11, mC6, mC12, mE7, mE10, mE1, Cu8, Cu15, Cu13, Cu14, Cu11, Cu12, as8, de11, rA5

Ne19 **from** Ha3, Ha2, He6, nT2, nT8, de3, in3, as3, rA8, nT11, mi8, Ne3, mE7, Ne14, mE16, Ot1, mE1, mE4, mE12, Ne17, Ne18, rA5

Ne20 **from** Ha3, Ha2, Cu8, He6, He3, He4, Cu1, Ha4, de3, in3, as3, rA8, rA6, rA7, nT11, nT13, nT12, mi12, mi16, mi5, mi4, Ne1, de7, in7, as5, del, in1, as1, Cu9, Cu10, Cu13, Cu14, Cu15, as9, fi5, de8, in8, as6, mC2, Ne3, Ot1, mi20, in11, rA5

Ne22 **from** Ot3, rA8, Ha2, nT11, Ha3, de3, in3, as3, mi5, mi4, Ne3, nT18, mE3, mi8, mE10, mE7, mE6, Ne5, nT5, nT2, rA5, nT8, nT12, mC2, mE2

Ne23 ⇐ Cu6, cu7, Ne6, Ne7, He3, He4, Ne22, He6

Ne24 ⇐ Ne27, He6

Ne25 ⇐ Ne19, Ne17, Ne18, He6

Ne26 ⇐ Ne17, Ne18, He6

Ne27 ⇐ Cu16, Ne25, Ne26, Ne17, Ne18, He6

fi1, del, in1, as1 **from**

fi2 **from** fi2, Ot3

fi3 **from** fi4, Ot3, rA6, rA7, Ha2, rA5

fi4 **from** Ot3, rA6, rA7, nT13, nT12

fi5, de8, in8, as6 ⇐ Cu2, de10, in10, as8, fi8, He3, He4

fi6 **from** Ot3, fi1, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, Cu1, Ha4, fi4, in3, as3, rA5

fi7 **from** fi8, fi6, fi2, Ot3, fi1, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, Cu1, Ha4, fi4, in3, as3, rA5

fi8 **from** fi4, fi7, fi2, Ot3, fi1, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, Cu1, Ha4, in3, as3, rA5

fi9 ⇐ Cu1, Ha4, Cu9, Cu10, Cu11, Cu12, fi8, fi3, fi4, fi5, de8, in8, as6, He3, He4

fi10 **from** fi9, Ot3

fi11, de12, in12, as10 **from** Ot3, nT9, nT10, mi9, mi10, Cu8, fi4, de3, in3, as3, fi3, de2, in2, as2

de2 **from** de3, Ot3, rA6, rA7, Ha2, rA5

de3 **from** Ot3, rA6, rA7, nT13, nT12

de4, in4, as4 **from** Ot3

de5 **from** Ot3

de6 **from** Ot3, de1, de11

de7, in7, as5 ⇐ de3, in3, as3, Cu1, Ha4, de13, in13, as11

de9 **from** Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, de3, de7, in7, as5, rA5

de10 **from** de3, de9, Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, de7, in7, as5, He3, He4, rA5

de11 ⇐ de10, de2, de3, He3, He4, Cu1, Ha4, Cu9, Cu10, Cu11, Cu12, fi5, de8, in8, as6

de13, in13, as11 ⇐ Ax2, de2, de3, de4, in2, in3, in4, as2, as3, as4

in2 **from** in3, Ot3, rA6, rA7, Ha2, rA5

in3 **from** Ot3, rA6, rA7, nT13, nT12
in5 **from** Ot3
in6 **from** Ot3, in1, in11
in9 **from** Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, in3, de7, in7, as5, rA5
in10 **from** in9, fi2, Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, in3, de7, in7, as5, rA5
in11 $\Leftarrow$ in10, in2, in3, Cu1, Ha4, Cu9, Cu10, Cu11, Cu12, fi5, de8, in8, as6
as2 **from** as3, He3, He4, Ot3, rA6, rA7, Ha2, rA5
as3 **from** Ot3, rA6, rA7, nT13, nT12
as7 **from** Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, as3, de7, in7, as5, rA5
as8 **from** as7, Ot3, del, in1, as1, rA6, rA7, Ha2, nT13, nT12, mi9, mi10, Cu9, Cu10, He3, He4, as3, de7, in7, as5, rA5
as9 $\Leftarrow$ as8, as2, as3, He3, He4, Cu1, Ha4, Cu9, Cu10, Cu11, Cu12, fi5, de8, in8, as6
rA1 **from** Ha2
rA2 **from** Ot3
rA3 **from** Ot3, rA9, He2, He1, rA2, rA13
rA4 **from** Ot3, nT14
rA5 **from** Ot3, rA1, rA2, Ha3, He2
rA6, rA7 **from** Ot3, nT13, nT12, nT14, rA11, mi4, bu2, bu3, Ha3, mi6, Ha2, He3, He4, He2, rA2
rA8 **from** Ot3, bu4, nT14, mi6, Ne2, mi5
rA9 **from** Ot3, Ha2, nT14, He1, He2
rA10 **from** Ot3
rA11 **from** Ot3, nT13, nT12, mi2
rA12 **from** Ot3, nT9, nT10
rA13 **from** Ot3, rA5
rA14 **from** Ot3, rA4, He1, rA2
nT1 **from** Ot3, pr5, Ha3, nT14, nT16, Ha2
nT2 **from** Ot3, nT14, Ha3, rA5
nT3 **from** Ot3, nT9, nT10
nT4 **from** Ot3, Ha3, de3, nT13, nT12, nT15, rA5
nT5 **from** Ot3, Ha3, in3, as3, nT13, nT12, nT15, nT18, mE3, mi4, rA5
nT6 **from** Ot3, nT13, nT12, nT14, Ha3, rA5
nT7 **from** Ot3, nT13, nT12, nT15, rA6, rA7, Ha2, mi9, mi10, nT14, Ha3, nT16, rA5
nT8 **from** Ot3, de3, in3, as3, nT13, nT12, nT15, nT18, mE3, mi4, Ha3, mC2, nT16, nT2,

Ha2, rA5

nT9, nT10 **from** Ot3, pr2, pr3, nT18

nT11 **from** Ot3, pr4, nT16, mi8

nT13, nT12 ⇐ nT9, nT10, Ha3, He3, He4

nT14 **from** Ot3, nT9, nT10, nT18, nT16, pr7

nT15 ⇐ nT14, Ha3, nT2

nT16 **from** Ot3, pr8

nT17 **from** Ot3, mi5, pr4, nT11, mi10

nT18 **from** Ot3, pr9, mi5, nT11

mi1 **from** Ot3, mi9, mi10, mi10

mi2 **from** Ot3, Ne4

mi3 **from** Ot3, fi11, de12, in12, as10, nT9, nT10, Ne5

mi4 **from** Ot3, mi9, mi10, mi3

mi5 **from** Ot3, nT9, nT10, Ne5, mi10, mi4

mi6 **from** Ot3, mi5, bu6, rA8, mi9, mi10, bu4, mi4

mi7 **from** Ot3, mi2, mi7, mi4, nT18, Ne2, mi10, nT17, mi3

mi8 **from** Ot3, mi10, Ne2, mi3

mi9, mi10 **from** Ot3, He3, He4, nT9, nT10, nT18, Ne3, Ha3, mi3, nT17, mi10, He2, mi4, mi12, mi6, He6

mi11 **from** Ot3, nT18, mi9, mi6, mi6

mi12 **from** Ot3, rA8, nT2, He6, mi9, mi5, mi3, Ha3, mi4, rA5

mi12 **from** Ot3, mi12, nT18, mi6, Ha3, mi4, rA5

mi13 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, He3, He4, mi9, mi10, mC9, rA5

mi14, mi15 ⇐ Ne12, Ne13, mi5, Cu15, mi13, Ot2, He3, He4, Ne17, Ne18, Cu8, He6, He5, mi4, Ot1

mi16 ⇐ Ne11, mi5, mi4

mi17, mi18 ⇐ Ne15, Ne16, mi5, Cu15, mi13, Ot2, He3, He4, Ne17, Ne18, Cu8, He6, He5, mi4

mi19 ⇐ Ne20, mi5, Cu15, mi13, Ot2, He3, He4

mi20 **from** Ha3, Ha2, Cu8, He6, He3, He4, Cu1, Ha4, de3, in3, as3, rA8, rA6, rA7, nT11, nT13, nT12, mi5, mi4, de7, in7, as5, Ot2, del, in1, as1, Cu9, Cu10, Cu13, Cu14, Cu15, as9, fi5, de8, in8, as6, mC6, Ne3, Ot3, mC11, mi13, mi9, mi10, mC2, mE3, mE10, mE7, mC12, mE1, mE13, Ne17, Ne18, mE2, mE4, Ot1, mE6, Ne10, in11, rA5

mC1 **from** Ot3, mi6, mi11, nT18

mC2 **from** Ot3, rA6, rA7, nT13, nT12, mC2

mC3 **from** Ot3, mC3, nT13, nT12, rA6, rA7, Ha2, rA5

mC4 **from** Ot3, mC4, mC2, mC3, He3, He4, rA6, rA7, Ha2, rA5 mC5 **from** Ot3

mC6 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, mC2, rA5

mC7 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, mC2, rA5

mC8 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, He3, He4, mC7, rA5

mC9 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, He3, He4, mi9, mi10, He5, mC7, mC8, rA5

mC10 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, mC2, del, in1, as1, mi6, Ha3, mi4, nT18, mE15, mC11, mi5, rA5

mC11 **from** Ot3, rA6, rA7, Ha2, nT13, nT12, mC2, rA5

mC12 **from** Ot3, rA6, rA7, mC2, mC11, Cu9, Cu10, de7, in7, as5, mi9, mC6

mE1 **from** Ot3

mE2 **from** Ot3

mE3 **from** mC1, Ot3, mi6, nT18

mE4 **from** Ot3, mE1

mE5 **from** Ot3, mE3, Ha3, mi6, mi4, nT18, Ha2, rA5

mE6 **from** Ot3

mE7 **from** Ot3, Ha2, Ha3, mi6, mi4, mE3, rA5

mE8 **from** Ot3, Ha3, mi6, mi4, nT18, Ha2, mE3, rA5

mE9 **from** Cu1, Ha4, Ot3, Ha2, Ha3, mi6, mi4, mE3, mC2, mC10, mE1, mC1, del, in1, as1, mi13, mi12, mC6, mE2, rA5

mE10 **from** del, in1, as1, mE3, mi6, Ot3, Ha2, Ha3, mi4, mE11, mE9, mE7, rA5

mE11 $\Leftarrow$ mE10, mi13, mE16, mi16, mi5, mE3, Ne12, Ne13, mC12, mE2, mE1, mE4, mC6, mE12, mi12, Cu13, Cu14, He3, He4, mi4

mE12 $\Leftarrow$ Ne23, Ne22, mE16, He6, Ne8

mE13 **from** Ot3, Ha2, mE14, del, in1, as1, Ha3, mi6, mi4, mE3, rA5

mE14 **from** Ot3, Ha2, del, in1, as1, Ha3, mi6, mi4, nT18, mE3, mE2, rA5

mE15 **from** Ot3, mE1, Ha2, del, in1, as1, Ha3, mi6, mi4, nT18, mE3, mE2, mE7, mE14, mE4, rA5

mE16 **from** Ha3, Ha2, mE3, del, in1, as1, mi6, mE2, mE4, mE1, mE7, mi4, Ot3, mE14, mE13, rA5

pr1 **from** Ot3, rA11, rA10, nT9, nT10, Ne5, mi2, mi4, mi8, mi5

pr2, pr3 **from** pr1, Ot3, rA11, mi1

pr4 $\Leftarrow$ pr1

pr5 $\Leftarrow$ pr6, pr1, bu1

pr6 **from** Ot3, Ha2, nT9, nT10, nT14, nT16, He2, rA2, pr1, bu1, pr10, rA9, He1, rA4

pr7, pr8, pr9 $\Leftarrow$ pr1, mi4

pr10 **from** Ot3, pr1, nT9, nT10, nT14, nT17

bu1 **from** Ot3, rA11, rA10, nT9, nT10, Ne5, mi2, mi8, mi5, bu5 bu2, bu3 $\Leftarrow$ bu1, Ot3, rA10

bu4 $\Leftarrow$ bu1

bu5 **from** Ot3, nT9, nT10, nT16, nT18, pr1, bu1

bu6 $\Leftarrow$ bu1, mi4

Ot1 **from** del, in1, as1

Ot2 **from** del, in1, as1

Ot3 **from** true

Ot4 **from** Ot3

Appendix B

For lock-free parallel GC

B.1 Invariants

In the invariants and the lemmas given below, we use the relations $R(x)$, $R(p,x)$, $R1(p,x)$, $\rhd_q$ defined in sections 5.2, 5.2, 5.3.1 and 5.4.2, respectively. The relation $\xrightarrow{*}$ is defined in section 5.2. The relation $\xrightarrow{M*}$ is the reflexive transitive closure of relation $\xrightarrow{M}$ on nodes defined by:

$$\begin{aligned} z \xrightarrow{M} x \equiv\ & (\texttt{color}[z] = \mathit{black} \wedge \mathit{aux}[z] \wedge \exists k\colon 1 \ldots \texttt{arity}[z]\colon \texttt{child}[z,\ k] = x) \\ \vee\ & (\texttt{color}[z] = \mathit{grey} \wedge \exists k\colon 1 \ldots \texttt{ari}[z]\colon \texttt{child}[z,\ k] = x) \end{aligned}$$

We define the j-th ancestor of a node x by the recursive function:

$$\mathit{anc}(x,\ j) \equiv ((j = 0\ \vee\ \texttt{father}[x] \leq 0)\ ?\ x : \mathit{anc}(\texttt{father}[x],\ j-1))$$

Main invariants:

I1: $\texttt{color}[x] = \mathit{white}\ \Rightarrow\ \neg R(x)$

I2: $\texttt{color}[x] = \mathit{white}\ \equiv\ x \in \texttt{free}$

I3: $554 \leq pc_p \leq 559\ \Rightarrow\ x_p \in \mathit{roots}_p$

I4: $\neg(\exists p\colon \mathit{rnd}_p = \texttt{shRnd})\ \Rightarrow\ \texttt{color}[x] \neq \mathit{grey}$

I5: $\texttt{srcnt}[x] - \texttt{freecnt}[x] = \sharp(\{p \mid x \in \mathit{roots}_p\}) + \sharp(\{(p,\ q) \mid (\texttt{Mbox}(p,\ q) = x \wedge \neg(pc_q = 559 \wedge p = r_q)) \vee (pc_p = 508 \wedge x_p = x \wedge q = r_p)\})$

Invariants about the stability of the preconditions in the offered procedures:

I6: $250 \leq pc_p \leq 258\ \Rightarrow\ R(p,\ x_p) \wedge R(p,\ y_p)$

I7: $pc_p = 280 \;\Rightarrow\; R(p,\ x_p)$

I8: $300 \le pc_p \le 308 \lor (100 \le pc_p \le 180 \land return_p = 300) \;\Rightarrow\; \forall k{:}\ 1 \ldots n_p{:}\ R(p,\ c_p[k])$

I9: $pc_p = 400 \lor (500 \le pc_p \le 508) \;\Rightarrow\; R(p,\ x_p)$

I10: $500 \le pc_p \le 508 \;\Rightarrow\; \mathtt{Mbox}[p,\ r_p] = 0$

I11: $550 \le pc_p \le 559 \;\Rightarrow\; \mathtt{Mbox}[r_p,\ p] \ne 0$

Invariants that hold globally:

I12: $rnd_p \le \mathtt{shRnd}$

I13: $\mathtt{shRnd} \le \mathtt{round}[x] \le \mathtt{shRnd} + 1$

I14: $\neg aux[x] \;\Rightarrow\; \mathtt{round}[x] = \mathtt{shRnd} + 1$

I15: $\neg(\exists p{:}\ rnd_p = \mathtt{shRnd}) \;\Rightarrow\; \mathtt{round}[x] \le \mathtt{shRnd}$

I16: $\mathtt{round}[x] \le \mathtt{shRnd} \;\Rightarrow\; \mathtt{color}[x] \ne grey$

I17: $\mathtt{color}[x] = grey \;\Rightarrow\; \neg aux[x]$

I18: $\mathtt{color}[x] = white \;\Rightarrow\; \neg R1(x)$

I19: $\mathtt{color}[x] = white \;\Rightarrow\; \mathtt{father}[x] \le -1$

I20: $\mathtt{color}[x] = grey \lor \mathtt{father}[x] \ge 0 \;\Rightarrow\; \mathtt{ari}[x] \le \mathtt{arity}[x]$

I21: $\mathtt{color}[x] = grey \land \mathtt{father}[x] > 0$
$\Rightarrow \exists k{:}\ 1 \ldots \mathtt{ari}[\mathtt{father}[x]]{:}\ \mathtt{child}[\mathtt{father}[x], k] = x$

I22: $x \ne \mathtt{father}[x]$

I23: $\mathtt{father}[x] > 0 \;\Rightarrow\; \neg(\exists j{:}\ \mathbb{N}{:}\ x = anc(x,\ j))$

I24: $\mathtt{father}[x] = 0 \land \mathtt{color}[x] = grey \;\Rightarrow\; \mathtt{srcnt}[x] > 0$

I25: $(\exists p{:}\ x \in roots_p) \;\Rightarrow\; \mathtt{srcnt}[x] - \mathtt{freecnt}[x] > 0$

I26: $\neg R(x) \;\Rightarrow\; \mathtt{srcnt}[x] - \mathtt{freecnt}[x] = 0$

I27: $R1(x) \land (\mathtt{color}[x] = grey \lor (\mathtt{color}[x] = black \land aux[x]))$
$\Rightarrow \exists w{:}\ \mathtt{srcnt}[w] > 0 \land (\mathtt{father}[w] = 0 \lor aux[w]) \land w \xrightarrow{M*} x$

I28: $return_p = 200 \lor return_p = 300 \lor return_p = 450$

Invariants about the first phase of GC:

I29: $101 \le pc_p \le 110 \land rnd_p = \mathtt{shRnd} \land \neg(x \in toBeC_p \land aux[x])$
$\Rightarrow \mathtt{round}[x] = rnd_p + 1$

I30: $101 \le pc_p \le 110 \land rnd_p = \mathtt{shRnd} \land \neg(\exists r{:}\ \neg(101 \le pc_r \le 110) \land rnd_r = \mathtt{shRnd})$
$\Rightarrow \mathtt{father}[x] \ne 0 \lor aux[x] \lor \mathtt{color}[x] \ne black$

I31: $101 \le pc_p \le 110 \land rnd_p = \mathtt{shRnd} \land \neg(\exists r{:}\ \neg(101 \le pc_r \le 110) \land rnd_r = \mathtt{shRnd})$
$\land\ (x \notin toBeC_p \lor \mathtt{round}[x] = rnd_p + 1)$
$\Rightarrow \neg aux[x]$

I32: $101 \le pc_p \le 110 \land rnd_p = \mathtt{shRnd} \land \neg(\exists r{:}\ \neg(101 \le pc_r \le 110) \land rnd_r = \mathtt{shRnd})$

$\wedge \neg(x \in toBeC_p \wedge aux[x])$
$\Rightarrow \texttt{father}[x] = 0 \vee \texttt{father}[x] = -1$

I33: $101 \leq pc_p \leq 110 \wedge rnd_p = \texttt{shRnd} \wedge \neg(\exists r\colon \neg(101 \leq pc_r \leq 110) \wedge rnd_r = \texttt{shRnd})$
$\wedge (x \notin toBeC_p \vee \texttt{round}[x] = rnd_p + 1) \wedge \texttt{srcnt}[x] = 0$
$\Rightarrow \texttt{father}[x] = -1$

Invariants about the second phase of GC:

I34: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \;\Rightarrow\; \texttt{round}[x] = rnd_p + 1$

I35: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \;\Rightarrow\; \neg aux[x]$

I36: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \wedge \texttt{father}[w] \leq -1 \;\Rightarrow\; \neg(\exists x\colon \texttt{father}[x] = w)$

I37: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \wedge \texttt{color}[x] = grey \wedge \texttt{father}[x] > 0$
$\Rightarrow \texttt{color}[\texttt{father}[x]] = grey \wedge \texttt{father}[\texttt{father}[x]] \geq 0$

I38: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \wedge \texttt{father}[x] > 0 \wedge \texttt{color}[x] = grey$
$\Rightarrow \exists j\colon 1 \ldots N\colon \texttt{father}[anc(x,\ j)] = 0 \wedge \texttt{color}[anc(x,\ j)] = grey$

I39: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \wedge \neg R1(x) \wedge \texttt{color}[x] = grey$
$\Rightarrow \texttt{father}[x] = -1$

I40: $\neg(101 \leq pc_p \leq 110) \wedge rnd_p = \texttt{shRnd} \wedge \texttt{color}[w] = black \wedge \texttt{father}[w] \geq 0$
$\Rightarrow \forall k\colon 1 \ldots \texttt{ari}[w]\colon (\texttt{father}[\texttt{child}[w,\ k]] = w \vee \texttt{father}[\texttt{child}[w,\ k]] < 0$
$\quad \Rightarrow \texttt{color}[\texttt{child}[w,\ k]] = black)$

I41: $pc_p = 121 \;\Rightarrow\; rnd_p \neq \texttt{shRnd} \vee toBeC_p = \emptyset$

I42: $\neg(101 \leq pc_p \leq 121) \wedge rnd_p = \texttt{shRnd} \wedge \texttt{srcnt}[x] > 0 \wedge \texttt{father}[x] = 0$
$\wedge \neg(x \in toBeD_p \vee (150 \leq pc_p \leq 180 \wedge x = x_p))$
$\Rightarrow \texttt{color}[x] = black$

I43: $(122 \leq pc_p \leq 127 \vee 150 \leq pc_p \leq 180) \wedge rnd_p = \texttt{shRnd} \wedge x \notin toBeC_p$
$\Rightarrow \texttt{father}[x] \geq 0$

I44: $((122 \leq pc_p \leq 127 \vee 150 \leq pc_p \leq 180) \wedge rnd_p = \texttt{shRnd} \wedge x \in toBeD_p$
$\wedge \texttt{father}[x] = 0) \vee (pc_p = 150 \wedge rnd_p = \texttt{shRnd} \wedge x = x_p)$
$\Rightarrow x \in toBeC_p$

I45: $(122 \leq pc_p \leq 127 \vee 150 \leq pc_p \leq 180) \wedge rnd_p = \texttt{shRnd} \wedge x \notin toBeC_p$
$\Rightarrow \neg(\exists w\colon \texttt{father}[x] = w \wedge (w \in set_p \vee w \in toBeC_p))$

I46: $122 \leq pc_p \leq 180 \wedge rnd_p = \texttt{shRnd}$
$\wedge \neg(x \in toBeC_p \vee (pc_p \geq 151 \wedge (x \in set_p \vee \exists i\colon 1 \ldots head_p\colon x = stack_p[i])))$
$\Rightarrow \texttt{color}[x] \neq grey$

Invariants about procedure *Mark_stack*:

I47: $pc_p = 150 \;\Rightarrow\; x_p \notin toBeD_p$

I48: $150 \le pc_p \le 180 \wedge rnd_p = \texttt{shRnd}$
$\Rightarrow (\texttt{color}[x_p] = grey \wedge \texttt{father}[x_p] = 0 \wedge \texttt{srcnt}[x_p] > 0)$
$\vee\ (\texttt{color}[x_p] = black \wedge \texttt{father}[x_p] = 0)$

I49: $151 \le pc_p \le 180 \wedge x \in toBeC_p$
$\Rightarrow \neg(x \in set_p \vee \exists i\colon 1 \ldots head_p\colon x = stack_p[i])$

I50: $151 \le pc_p \le 180 \wedge (\exists i\colon 1 \ldots head_p\colon x = stack_p[i])$
$\Rightarrow x \notin set_p \wedge x \notin toBeC_p$

I51: $151 \le pc_p \le 180 \wedge rnd_p = \texttt{shRnd}$
$\Rightarrow x_p \in set_p \vee \texttt{color}[x_p] = black \vee (\exists i\colon 1 \ldots head_p\colon x_p = stack_p[i])$

I52: $151 \le pc_p \le 180 \wedge rnd_p = \texttt{shRnd} \wedge (x \in set_p \vee \exists i\colon 1 \ldots head_p\colon x = stack_p[i])$
$\Rightarrow (\texttt{color}[x] = grey \vee \texttt{color}[x] = black) \wedge \texttt{father}[x] \ge 0$

I53: $151 \le pc_p \le 180 \wedge rnd_p = \texttt{shRnd} \wedge \texttt{color}[x] = grey$
$\wedge (x \in set_p \vee \exists i\colon 1 \ldots head_p\colon x = stack_p[i])$
$\Rightarrow R1(x) \wedge \texttt{father}[x] \ge 0$

I54: $151 \le pc_p \le 180 \wedge rnd_p = \texttt{shRnd} \wedge (\exists i\colon 1 \ldots head_p\colon w = stack_p[i])$
$\Rightarrow \forall k\colon 1 \ldots \texttt{ari}[w]\colon (\texttt{father}[\texttt{child}[w,\ k]] \ge 0 \vee \texttt{color}[\texttt{child}[w,\ k]] = black)$
$\wedge (father[\texttt{child}[w,\ k]] = w \Rightarrow \texttt{child}[w,\ k] \in set_p \vee \texttt{color}[\texttt{child}[w,\ k]] = black$
$\vee ((\exists j\colon 1 \ldots head_p\colon \texttt{child}[w,\ k] = stack_p[j]) \wedge (\forall m,\ n\colon 1 \ldots head_p\colon$
$w = stack_p[m] \wedge \texttt{child}[w,\ k] = stack_p[n] \Rightarrow m < n)))$
$\vee (158 \le pc_p \le 164 \wedge w_p = w \wedge k \ge j_p)$

I55: $pc_p = 158 \wedge rnd_p = \texttt{shRnd} \;\Rightarrow\; j_p = 1 \vee 1 < j_p \le \texttt{ari}[w_p] + 1$

I56: $158 \le pc_p \le 164 \wedge rnd_p = \texttt{shRnd}$
$\Rightarrow k_p = \texttt{ari}[w_p] \wedge \forall j\colon 1 \ldots k_p\colon ch_p[j] = \texttt{child}[w_p,\ j]$

I57: $158 \le pc_p \le 164 \wedge rnd_p = \texttt{shRnd}$
$\wedge \neg(x \in toBeC_p \vee \exists j\colon 1 \ldots j_p - 1\colon x = \texttt{child}[w_p,\ j])$
$\Rightarrow \texttt{father}[x] \ne w_p$

I58: $158 \le pc_p \le 164 \wedge rnd_p = \texttt{shRnd}$
$\Rightarrow \forall k\colon 1 \ldots j_p - 1\colon (\texttt{color}[\texttt{child}[w_p,\ k]] = grey \Rightarrow \texttt{father}[\texttt{child}[w_p,\ k]] \ge 0$
$\wedge (\texttt{father}[\texttt{child}[w_p,\ k]] = w_p \Rightarrow \texttt{child}[w_p,\ k] \in set_p))$

I59: $158 \le pc_p \le 165 \;\Rightarrow\; \exists i\colon 1 \ldots head_p\colon w_p = stack_p[i]$

I60: $159 \le pc_p \le 164 \wedge rnd_p = \texttt{shRnd} \;\Rightarrow\; 1 \le j_p \le \texttt{ari}[w_p] \wedge y_p = \texttt{child}[w_p,\ j_p]$

I61: $168 \le pc_p \le 180 \;\Rightarrow\; rnd_p \ne \texttt{shRnd} \vee set_p = \emptyset$

I62: $170 \le pc_p \le 176 \;\Rightarrow\; head_p \ne 0$

I63: $pc_p = 180 \;\Rightarrow\; rnd_p \ne \texttt{shRnd} \vee head_p = 0$

Invariants about the third phase of GC:

I64: $129 \le pc_p \le 137 \wedge rnd_p = \texttt{shRnd} \wedge \texttt{color}[x] = grey \;\Rightarrow\; \neg R1(x)$

I65: $pc_p = 134 \wedge \texttt{round}[x] = rnd_p + 1 \wedge \texttt{color}[x] = grey$
$\Rightarrow \neg R(x) \wedge x \notin \texttt{free}$
I66: $\neg(101 \leq pc_p \leq 134 \vee 150 \leq pc_p \leq 180) \wedge rnd_p = \texttt{shRnd}$
$\Rightarrow \texttt{color}[x] \neq grey$
I67: $pc_p = 135 \ \Rightarrow\ rnd_p \neq \texttt{shRnd} \vee toBeC_p = \emptyset$
I68: $\neg(101 \leq pc_p \leq 135 \vee 150 \leq pc_p \leq 180) \ \Rightarrow\ rnd_p \neq \texttt{shRnd}$

Invariants outside GC:

I69: $pc_p = 450 \vee (100 \leq pc_p \leq 180 \wedge return_p = 450) \ \Rightarrow\ R(p,\ z_p)$
I70: $500 \leq pc_p \leq 508 \ \Rightarrow\ \texttt{Mbox}[p,\ r_p] = 0$
I71: $552 \leq pc_p \leq 559 \ \Rightarrow\ x_p = \texttt{Mbox}[r_p,\ p] \wedge x_p \neq 0$
I72: $pc_p = 558 \ \Rightarrow\ \texttt{srcnt}[x_p] > 1$

Main lemmas:

V1: $p \neq q \wedge R(p,\ x) \wedge I18 \wedge I25 \quad \triangleright_q \quad R(p,\ x)$
V2: $\texttt{color}[x] \neq white \wedge \neg R1(x) \wedge I6 \wedge I8 \wedge I9 \wedge I25 \wedge I63 \wedge I66 \quad \triangleright \quad \neg R1(x)$

B.2 Dependencies between invariants

Let us write "φ **from** $\psi_1, \cdots, \psi_n$" to denote that φ can be proved to be an invariant using that $\psi_1, \cdots, \psi_n$ hold in the precondition of every step. We write "$\varphi \ \Leftarrow\ \psi_1, \cdots, \psi_n$" to denote that φ can be directly derived from $\psi_1, \cdots, \psi_n$. We have verified the following "**from**" and "$\Leftarrow$" relations mechanically:

I1 $\Leftarrow$ I3, I5, I18, I71
I2 **from** : *true*
I3 **from** : I28
I4 $\Leftarrow$ I12, I15, I16
I5 **from** : I18, I25, I28, I70, I71
I6 **from** : I18, I25, I28
I7 **from** : I18, I25, I28
I8 **from** : I18, I25, I28
I9 **from** : I18, I25, I28
I10 **from** : I28
I11 **from** : I28
I12 **from** : *true*

I13 **from** : I12, I34
I14 **from** : I12, I13
I15 **from** : I12, I13, I34
I16 **from** : I12, I13, I66
I17 **from** : I66
I18 **from** : I6, I8, I9, I12, I16, I25, I64, I69
I19 **from** : I12, I16, I18, I39, I64
I20 **from** : I19
I21 **from** : I12, I13, I14, I15, I17, I20, I32, I34, I37, I60, I66
I22 $\Leftarrow$ I23
I23 **from** : I13, I16, I36, I50, I59
I24 **from** : *true*
I25 $\Leftarrow$ I5
I26 $\Leftarrow$ I3, I5, I9
I27 **from** : I6, I8, I9, I12, I13, I14, I16, I17, I18, I20, I21, I24, I25, I35, I37, I38, I54, I61, I64, I66, I69
I28 **from** : *true*
I29 **from** : I12, I13, I14, I28
I30 **from** : I12, I13, I14, I15, I16, I19, I28, I68
I31 **from** : I12, I13, I15, I28, I35, I68
I32 **from** : I12, I13, I14, I15, I28, I29, I31, I34, I68
I33 **from** : I12, I13, I15, I28, I29, I34, I68
I34 **from** : I12, I13, I29, I34
I35 **from** : I12, I31, I34
I36 **from** : I12, I32, I34, I52, I59
I37 **from** : I12, I21, I28, I32, I34, I39, I40, I52, I54, I59, I60, I61, I64
I38 $\Leftarrow$ I23, I32, I35, I37
I39 **from** : I12, I18, I19, I20, I21, I24, I27, I28, I33, I34, I35, I37, I38, I40, I52, I53, I54, I59, I60, I61
I40 **from** : I12, I19, I20, I28, I30, I31, I32, I34, I35, I54, I61, I62
I41 **from** : I12, I28
I42 **from** : I9, I12, I18, I24, I25, I34, I42, I51, I61, I63
I43 **from** : I12, I28, I34, I43, I48
I44 **from** : I12, I28, I34, I35, I47
I45 **from** : I12, I28, I43, I44, I48, I50, I59
I46 **from** : I12, I34, I61, I63
I47 **from** : I28

I48 **from** : I12, I18, I19, I24, I28, I34, I39, I64
I49 **from** : I28
I50 **from** : I28, I49
I51 **from** : I12, I28, I34, I52
I52 **from** : I12, I28, I34, I48, I53, I64
I53 **from** : I12, I18, I19, I20, I21, I24, I27, I28, I34, I35, I37, I38, I40, I48, I52, I54, I59, I60, I61
I54 **from** : I12, I18, I19, I20, I22, I28, I34, I35, I40, I43, I50, I52, I53, I55, I56, I57, I58, I60, I62
I55 **from** : I12, I28, I34, I56, I60
I56 **from** : I12, I20, I28, I34, I52, I59
I57 **from** : I12, I19, I20, I28, I34, I43, I45, I52, I55, I59, I60
I58 **from** : I12, I20, I28, I34, I35, I43, I52, I55, I56, I57, I59, I60
I59 **from** : I28
I60 **from** : I12, I20, I28, I34, I52, I56, I59
I61 **from** : I12, I28
I62 **from** : I28
I63 **from** : I12, I28
I64 **from** : I6, I8, I9, I12, I25, I27, I34, I35, I42
I65 $\Leftarrow$ I2, I3, I5, I12, I16, I64, I71
I66 **from** : I12, I34, I46
I67 **from** : I12, I28
I68 **from** : I12
I69 **from** : I18, I25
I70 **from** : I28
I71 **from** : I28, I70
I72 $\Leftarrow$ I3, I5, I71

B.3 The low-level lock-free algorithm

B.3.1 Data Structure

Constant

$P =$ `number of processes`;
$N =$ `number of nodes`;
$C =$ `upper bound of number of children`;

Type

`colorType`: $\{$*white, black, grey*$\}$;

nodeType: **record** =
  arity: $\mathbb{N}$;
  child: **array** $[1 \ldots C]$ **of** $1 \ldots N$;
  color: colorType;
  srcnt, freecnt, ari: $\mathbb{N}$;
  father: $\mathbb{N} \cup \{-1\}$;
  round: $\mathbb{N}$;
end

Shared variables
  node: **array** $[1 \ldots N+P]$ **of** nodeType;
  indir: **array** $[1 \ldots N]$ **of** $1 \ldots N+P$;
  Mbox: **array** $[1 \ldots P,\ 1 \ldots P]$ **of** $0 \ldots N$;
  shRnd: $\mathbb{N}$;

Private variables
  roots: **a subset of** $1 \ldots N$;
  rnd: $\mathbb{N}$;
  toBeC: **a subset of** $1 \ldots N$;
  mp: $1 \ldots N+P$;

Initialization:
  shRnd $= 1 \wedge \forall x: 1 \ldots N$: (indir$[x] = x \wedge$ round[indir$[x]] = 1$);
  $\forall p: 1 \ldots P$: $mp_p = N + p$;
  all other variables are equal to be the minimal values in their respective domains.

B.3.2 Algorithm

proc *GCollect*() =
  local $m : 1 \ldots N+P$; x: $1 \ldots N$; *toBeD*: **a subset of** $1 \ldots N$;
% first phase
100: *rnd* := shRnd; *toBeC* := $\{1, \ldots, N\}$;
101: **while** shRnd $= rnd \wedge toBeC \neq \emptyset$ **do**
    choose $x \in toBeC$;
    while *true* **do**
102:     $m :=$ *LL*(indir$[x]$);
103:     node$[mp]$:= node$[m]$;
104:     **if** round$[mp] = rnd$ **then**
105:       round$[mp] := rnd + 1$; ari$[mp]$:= arity$[mp]$;
       if color$[mp] = black$ **then** color$[mp]$:= *grey*; **fi**;

```
            if srcnt[mp] > 0 then father[mp] := 0; else father[mp] := −1; fi;
            if SC(indir[x], mp) then toBeC := toBeC − {x}; mp := m; break; fi;
         elseif VL(indir[x]) then toBeC := toBeC − {x}; break; fi;
      od;
   od;
% second phase
   toBeC := {1, ..., N}; toBeD := {1, ..., N};
   while shRnd = rnd ∧ toBeD ≠ ∅ do
      choose x ∈ toBedone;
      while true do
         m := LL(indir[x]);
         node[mp] := node[m];
         if father[mp] = 0 then
            if VL(indir[x]) then
               toBeD := toBeD − {x};
               Mark_stack(x); break; fi;
         elseif VL(indir[x]) then toBeD := toBeD − {x}; break; fi;
      od;
   od;
% last phase
   while shRnd = rnd ∧ toBeC ≠ ∅ do
      choose x ∈ toBeC;
      while true do
         m := LL(indir[x]);
         node[mp] := node[m];
         if round[mp] = rnd + 1 ∧ color[mp] = grey then
            color[mp] := white;
            if SC(indir[x], mp) then toBeC := toBeC − {x}; mp := m; break; fi;
         elseif VL(indir[x]) then toBeC := toBeC − {x}; break; fi;
      od;
   od;
   CAS(shRnd, rnd, rnd + 1);
   return;
end GCollect.

proc Mark_stack(x: 1 ... N) =
   local w, y: 1 ... N; suc: Bool; j, k: ℕ;
```

```
     stack: Stack; head: ℕ; set: a subset of 1...N;
     ch: [1...C] of 1...N; m, n: 1...N + P;
150: toBeC := toBeC − {x}; set := {x}; head := 0;
151: while shRnd = rnd ∧ set ≠ ∅ do
         choose w ∈ set;
         while true do
152:         m := LL(indir[w]);
153:         node[mp] := node[m];
154:         if color[mp] = grey ∧ round[mp] = rnd + 1 then
155:             k := ari[mp];
                 for j := 1 to k do ch[j] := child[mp, j]; od;
156:             if VL(indir[w]) then
                     set := set − {w}; head++; stack[head] := w; j := 1;
157:                 while shRnd = rnd ∧ j ≤ k do
                         y := ch[j];
                         if y ∈ toBeC then
                             while true do
158:                             n := LL(indir[y]);
159:                             node[mp] := node[n];
160:                             if (father[mp] = −1 ∨ father[mp] = w)
                                             ∧round[mp] = rnd + 1 then
161:                                 if father[mp] = −1 then father[mp] := w; fi;
162:                                 if SC(indir[y], mp) then
                                         toBeC := toBeC − {y}; mp := n;
                                         set := set + {y}; break; fi;
163:                             elseif VL(indir[y]) then break; fi;
                             od; fi;
                         j := j + 1;
                     od;
                     break; fi;
164:         elseif VL(indir[w]) then set := set − {w}; break; fi;
         od;
     od;
170: while shRnd = rnd ∧ head ≠ 0 do
         y := stack[head];
         while true do
171:         m := LL(indir[y]);
```

```
172:          node[mp] := node[m];
173:          if color[mp] = grey ∧ round[mp] = rnd + 1 then
174:             color[mp] := black;
                 srcnt[mp] := srcnt[mp] − freecnt[mp]; freecnt[mp] := 0;
175:             if SC(indir[y], mp) then mp := m; head--; break; fi ;
176:          elseif VL(indir[y]) then head--; break; fi;
           od;
       od;
180:   return;
end Mark_stack.

proc Create(): 1 . . . N =
       local m: 1 . . . N + P; x: 1 . . . N;
       while true do
200:       choose x ∈ 1 . . . N;
201:       m := LL(indir[x]);
202:       node[mp] = node[m];
203:       if color[mp] = white then
204:          color[mp] := black; srcnt[mp] := 1; arity[mp] := 0;
205:          if SC(indir[x], mp) then
                 roots := roots + {x};
                 mp := m; break; fi;
206:       elseif time to do GC then
                GCollect(); fi;
       od;
207:   return x
end Create.

proc AddChild(x, y: 1 . . . N): Bool =
{R(self, x) ∧ R(self, y)}
       local m: 1 . . . N + P; suc: Bool;
250:   suc := false;
       while true do
251:       m := LL(indir[x]);
252:       node[mp] := node[m];
253:       if arity[mp] < C then
254:          arity[mp]++;
```

```
              child[mp, arity[mp]] := y;
255:          if SC(indir[x], mp) then
                 mp := m; suc := true; break; fi;
256:      elseif VL(indir[x]) then break; fi;
       od;
257:   return suc
end AddChild.

proc GetChild(x: 1...N, rth: 1...N): 0...N =
{R(self, x)}
       local m: 1...N + P; y: 1...N;
       while true do
280:      m := LL(indir[x]);
281:      node[mp] := node[m];
282:      if 1 ≤ rth ≤ arity[mp] then y := child[mp, rth]; else y := 0; fi;
283:      if VL(indir[x]) then break; fi;
       od;
284:   return y
end GetChild.

proc Make(c: array [1...C] of 1...N, n: 1...C): 1...N =
{∀ j: 1...n: R(self, c[j])}
       local m: 1...N + P; x: 1...N; j: ℕ;
       while true do
300:      choose x ∈ [1...N];
301:      m := LL(indir(x));
302:      node[mp] := node[m];
303:      if color[mp] = white then
304:         color[mp] := black;
             srcnt[mp] := 1; arity[mp] := n;
             for j := 1 to n do child[mp, j] := c[j] od;
305:         if SC(indir(x), mp) then
                roots := roots + {x};
                mp := m; break; fi;
306:      elseif time to do GC then
               GCollect(); fi;
       od;
```

```
307:  return x
end Make.

proc Protect(x: 1...N) =
{R(self, x) ∧ x ∉ roots}
      local m: 1...N + P;
      while true do
400:     m := LL(indir[x]);
401:     node[mp] := node[m];
402:     srcnt[mp]++;
403:     if SC(indir[x], mp) then
            roots := roots + {x};
            mp := m; break; fi;
      od;
404:  return
end Protect.

proc UnProtect(z: 1...N) =
{z ∈ roots}
      local m: 1...N + P;
      while true do
450:     m := LL(indir[z]);
451:     node[mp] := node[m];
452:     freecnt[mp]++;
453:     if SC(indir[x], mp) then
            roots := roots \ {z};
            mp := m; break; fi;
      od;
454:  return
end UnProtect.

proc Send(x: 1...N, r: 1...P) =
{R(self, x) ∧ Mbox[self, r] = 0}
      local m: 1...N + P;
      while true do
500:     m := LL(indir[x]);
501:     node[mp] := node[m];
```

```
502:     srcnt[mp]++;
503:     if SC(indir[x], mp) then
            mp := m;
504:        Mbox[self, r] := x; break; fi;
     od;
505: return
end Send.

proc Receive(r: 1...P): 0...N =
{Mbox[r, self] ≠ 0}
     local x: 1...N;
550: x := Mbox[r, self];
551: if x ∉ roots then
        roots := roots ∪ {x};
        Mbox[r, self] := 0;
     else
        while true do
552:        m := LL(indir[x]);
553:        node[mp] := node[m];
554:        srcnt[mp]--;
555:        if SC(indir[x], mp) then
               mp := m;
556:           Mbox[r, self] := 0;
               break; fi;
        od; fi;
557: return
end Receive.

proc Check(r, q: 1...P): Bool
     local suc : Bool;
600: suc := (Mbox[r, q] = 0);
601: return suc;
end Receive.
```