Class Notes for BRICS Minicourse - Fall 2003

SEPARATION LOGIC

3. Specifications and Inference Rules

John C. Reynolds
Department of Computer Science
Carnegie Mellon University
and
BRICS
Aarhus University

Revised November 20, 2003

Specifications

$$\begin{array}{lll} \langle\text{spec}\rangle ::= \{\langle\text{assert}\rangle\}\ \langle\text{comm}\rangle\ \{\langle\text{assert}\rangle\} & & \text{(partial correctness)} \\ \qquad\quad \mid [\ \langle\text{assert}\rangle\ ]\ \langle\text{comm}\rangle\ [\ \langle\text{assert}\rangle\ ] & & \text{(total correctness)} \end{array}$$

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

Partial correctness:

$\{p\}\ c\ \{q\}$ *holds* iff $\forall(s,h) \in \mathrm{States}_V.\ s,h \vDash p$ implies

$\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}\ \mathbf{abort}$

$\{$i.e., c is safe at $(s,h)\}$

and $\big(\forall(s',h') \in \mathrm{States}_V.\ (s,h)\ [\![c]\!]_{\mathrm{comm}}\ (s',h')$ implies $s',h' \vDash q\big)$

Total correctness:

$[\ p\ ]\ c\ [\ q\ ]$ *holds* iff $\forall(s,h) \in \mathrm{States}_V.\ s,h \vDash p$ implies

$\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}\ \mathbf{abort}$ and $\neg(s,h)\ [\![c]\!]_{\mathrm{comm}}\ \bot$

$\{$i.e., c must terminate normally at $(s,h)\}$

and $\big(\forall(s',h') \in \mathrm{States}_V.\ (s,h)\ [\![c]\!]_{\mathrm{comm}}\ (s',h')$ implies $s',h' \vDash q\big)$

- Specifications are implicitly quantified over both stores and heaps — and over all executions.
- An execution giving a memory fault falsifies a specification.

Examples of Partial Correctness Specifications

$$\{\mathsf{x} - \mathsf{y} > 3\}\ \mathsf{x} := \mathsf{x} - \mathsf{y}\ \{\mathsf{x} > 3\}$$

$$\{\mathsf{x} + \mathsf{y} \geq 17\}\ \mathsf{x} := \mathsf{x} + 10\ \{\mathsf{x} + \mathsf{y} \geq 27\}$$

$$\{\mathsf{x} \leq 10\}\ \mathbf{while}\ \mathsf{x} \neq 10\ \mathbf{do}\ \mathsf{x} := \mathsf{x} + 1\ \{\mathsf{x} = 10\}$$

$$\{\mathbf{true}\}\ \mathbf{while}\ \mathsf{x} \neq 10\ \mathbf{do}\ \mathsf{x} := \mathsf{x} + 1\ \{\mathsf{x} = 10\}$$

$$\{\mathsf{x} > 10\}\ \mathbf{while}\ \mathsf{x} \neq 10\ \mathbf{do}\ \mathsf{x} := \mathsf{x} + 1\ \{\mathbf{false}\}$$

$$\{\mathbf{emp}\}\ \mathsf{x} := \mathbf{cons}(1, 2)\ \{\mathsf{x} \mapsto 1, 2\}$$

$$\{\mathsf{x} \mapsto 1, 2\}\ \mathsf{y} := [\mathsf{x}]\ ; [\mathsf{x} + 1] := 3\ \{\mathsf{x} \mapsto 1, 3 \wedge \mathsf{y} = 1\}$$

Examples of Total Correctness Specifications

$$[\,\mathsf{x} - \mathsf{y} > 3\,]\ \mathsf{x} := \mathsf{x} - \mathsf{y}\ [\,\mathsf{x} > 3\,]$$

$$[\,\mathsf{x} + \mathsf{y} \geq 17\,]\ \mathsf{x} := \mathsf{x} + 10\ [\,\mathsf{x} + \mathsf{y} \geq 27\,]$$

$$[\,\mathsf{x} \leq 10\,]\ \mathbf{while}\ \mathsf{x} \neq 10\ \mathbf{do}\ \mathsf{x} := \mathsf{x} + 1\ [\,\mathsf{x} = 10\,]$$

$$[\,\mathbf{emp}\,]\ \mathsf{x} := \mathbf{cons}(1, 2)\ [\,\mathsf{x} \mapsto 1, 2\,]$$

$$[\,\mathsf{x} \mapsto 1, 2\,]\ \mathsf{y} := [\mathsf{x}]\ ; [\mathsf{x} + 1] := 3\ [\,\mathsf{x} \mapsto 1, 3 \wedge \mathsf{y} = 1\,]$$

Annotations

When a specification is proved by using Hoare's inference rules, one cannot, in general, determine the proof from the specification. Many of the inference rules, however, can be modified, by adding *annotations* to their conclusions, in such a way that the resulting *annotated specifications* determine their proofs.

In the rules that follow, we will underline the annotations.

Hoare's Inference Rules for Specifications

Rules for total correctness are the same except for the **while** command. Underlining indicates annotations (in annnotated specifications).

Assignment (AS)

$$\frac{}{\{p/v \rightarrow e\}\ v := e\ \{p\}}$$

Sequential Composition (SQ)

$$\frac{\{p\}\ c_1\ \{q\} \qquad \{q\}\ c_2\ \{r\}}{\{p\}\ c_1\ ;\ \underline{\{q\}}\ c_2\ \{r\}}$$

Strengthening Precedent (SP)

$$\frac{p \Rightarrow q \qquad \{q\}\ c\ \{r\}}{\{p\}\ \underline{\{q\}}\ c\ \{r\}}$$

Weakening Consequent (WC)

$$\frac{\{p\}\ c\ \{q\} \qquad q \Rightarrow r}{\{p\}\ c\ \underline{\{q\}}\ \{r\}}$$

Hoare's Inference Rules (continued)

Partial Correctness of **while** (WHP)

$$\frac{\{i \wedge b\}\; c\; \{i\}}{\{i\}\; \textbf{while}\; b\; \textbf{do}\; c\; \{i \wedge \neg\, b\}}$$

Total Correctness of **while** (WHT)

$$\frac{[\, i \wedge b \wedge e = v_0\,]\; c\; [\, i \wedge e < v_0\,] \quad (i \wedge b) \Rightarrow e \geq 0}{[\, i\,]\; \textbf{while}\; \underline{[\textbf{vrnt}\colon e]}\; b\; \textbf{do}\; c\; [\, i \wedge \neg\, b\,]}$$

when v_0 does not occur free in i, b, c, or e.

skip (SK)

$$\frac{}{\{p\}\; \textbf{skip}\; \{p\}}$$

Conditional (CD)

$$\frac{\{p \wedge b\}\; c_1\; \{q\} \qquad \{p \wedge \neg\, b\}\; c_2\; \{q\}}{\{p\}\; \textbf{if}\; b\; \textbf{then}\; c_1\; \textbf{else}\; c_2\; \{q\}}$$

Variable Declaration (DC)

$$\frac{\{p\}\; c\; \{q\}}{\{p\}\; \textbf{newvar}\; v\; \textbf{in}\; c\; \{q\}}$$

when v does not occur free in p or q.

Some Derived Rules

To reduce the number of unnecessary annotations, it is useful to use the following derived inference rules:

Assignment (ASDR)

$$\frac{p_0 \Rightarrow (p/v \to e)}{\{p_0\}\ v := e\ \{p\}} \qquad \frac{\{p_0\}\ c\ \{p/v \to e\}}{\{p_0\}\ c\,;v := e\ \{p\}}$$

skip (SKDR)

$$\frac{p_0 \Rightarrow p}{\{p_0\}\ \mathbf{skip}\ \{p\}}$$

Partial Correctness of **while** (WHPDR)

$$\frac{\{i \wedge b\}\ c\ \{i\} \qquad (i \wedge \neg\, b) \Rightarrow p}{\{i\}\ \mathbf{while}\ b\ \mathbf{do}\ c\ \{p\}}$$

Total Correctness of **while** (WHTDR)

$$\frac{[\, i \wedge b \wedge e = v_0\,]\ c\ [\, i \wedge e < v_0\,] \quad (i \wedge b) \Rightarrow e \geq 0 \quad (i \wedge \neg\, b) \Rightarrow p}{[\, i\,]\ \mathbf{while}\ \underline{[\mathbf{vrnt}\colon e]}\ b\ \mathbf{do}\ c\ [\, p\,]}$$

when v_0 does not occur free in i, b, c, or e.

Hoare's Structural Inference Rules

A *structural* inference rule for specifications is one that is not specific to a particular form of command. In addition to the rules for Strengthening Precedent (SP) and Weakening Consequent (WC), the following structural rules are sound for both Hoare logic and Separation logic.

(These rules cannot be used with annotations.)

Renaming (RN)

$$\frac{\{p\}\ c\ \{q\}}{\{p'\}\ c'\ \{q'\},}$$

where p', c', and q' are obtained from p, c, and q by zero or more renamings of bound variables.

Substitution (SUB)

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

where $v_1, \ldots, v_n$ are the variables occurring free in p, c, or q, and, if $v_i \in \mathrm{MD}(c)$, then e_i is a variable that does not occur free in any other e_j.

Hoare's Structural Inference Rules (continued)

Disjunction (DISJ)

$$\frac{\{p_1\}\ c\ \{q_1\} \qquad \{p_2\}\ c\ \{q_2\}}{\{p_1 \vee p_2\}\ c\ \{q_1 \vee q_2\}}$$

Conjunction (CONJ)

$$\frac{\{p_1\}\ c\ \{q_1\} \qquad \{p_2\}\ c\ \{q_2\}}{\{p_1 \wedge p_2\}\ c\ \{q_1 \wedge q_2\}}$$

Existential Quantification (EQ)

(also called Auxilliary Variable Elimination)

$$\frac{\{p\}\ c\ \{q\}}{\{\exists v.\ p\}\ c\ \{\exists v.\ q\},}$$

where v is not free in c.

Universal Quantification (UQ)

$$\frac{\{p\}\ c\ \{q\}}{\{\forall v.\ p\}\ c\ \{\forall v.\ q\},}$$

where v is not free in c.

Hoare's Structural Inference Rules (continued)

The following structural rule is sound for Hoare logic, but not for separation logic:

The Unsound Rule of Constancy

$$\frac{\{p\}\ c\ \{q\}}{\{p \wedge r\}\ c\ \{q \wedge r\},}$$

when no variable occurs in both $\mathrm{FV}(r)$ and $\mathrm{MD}(c)$.

Counterexamples

$$\frac{\{\mathsf{x} = \mathsf{y} \wedge \mathsf{x} \mapsto 3\}\ [\mathsf{x}] := 4\ \{\mathsf{x} = \mathsf{y} \wedge \mathsf{x} \mapsto 4\}}{\{\mathsf{x} = \mathsf{y} \wedge \mathsf{x} \mapsto 3 \wedge \mathsf{y} \mapsto 3\}\ [\mathsf{x}] := 4\ \{\mathsf{x} = \mathsf{y} \wedge \mathsf{x} \mapsto 4 \wedge \mathsf{y} \mapsto 3\}}$$

$$\frac{\{\mathsf{x} \mapsto 3\}\ [\mathsf{x}] := 4\ \{\mathsf{x} \mapsto 4\}}{\{\mathsf{x} \mapsto 3 \wedge \mathsf{x} \mapsto 3\}\ [\mathsf{x}] := 4\ \{\mathsf{x} \mapsto 4 \wedge \mathsf{x} \mapsto 3\}}$$

Structural Inference Rules:

The Frame Rule (FR)

$$\frac{\{p\}\ c\ \{q\}}{\{p \ast r\}\ c\ \{q \ast r\},}$$

when no variable occurs in both $\mathrm{FV}(r)$ and $\mathrm{MD}(c)$.

Proposition 9 *The frame rule is sound.*

Proof

- Assume $\{p\}\ c\ \{q\}$ is valid, and $s, h \vDash p \ast r$. Then there is an $\hat{h} \subseteq h$ such that $s, h - \hat{h} \vDash p$ and $s, \hat{h} \vDash r$.

- From $\{p\}\ c\ \{q\}$, we know that c is safe at $(s, h - \hat{h})$ (and in the total-correctness case it must terminate normally). Then, by Proposition 6, we know that c is safe at (s, h) (and in the total-correctness case it must terminate normally).

- Suppose we have $(s, h)\, [\![c]\!]_{\mathrm{comm}}\, (s', h')$. Since c is safe at $(s, h - \hat{h})$, we know by Proposition 7 that $\hat{h} \subseteq h'$ and $(s, h-\hat{h})\, [\![c]\!]_{\mathrm{comm}}\, (s', h'-\hat{h})$. Then $\{p\}\ c\ \{q\}$ and $s, h-\hat{h} \vDash p$ imply that $s', h' - \hat{h} \vDash q$.

- Since $(s, h)\, [\![c]\!]_{\mathrm{comm}}\, (s', h')$, Proposition 3 gives $s\, v = s'v$ for all v that are not modified by c. Then, since these include the free variables of r, $s, \hat{h} \vDash r$ implies that $s', \hat{h} \vDash r$. Thus $s', h' \vDash q \ast r$.

END OF PROOF

Inference Rules for Mutation

The local form (MUL):

$$\overline{\{e \mapsto -\}\ [e] := e'\ \{e \mapsto e'\}.}$$

The global form (MUG):

$$\overline{\{(e \mapsto -) \ast r\}\ [e] := e'\ \{(e \mapsto e') \ast r\}.}$$

The backward-reasoning form (MUBR):

$$\overline{\{(e \mapsto -) \ast ((e \mapsto e') -\!\!\ast\, p)\}\ [e] := e'\ \{p\}.}$$

One can derive (MUG) from (MUL) by using the frame rule; one can go in the opposite direction by taking r to be **emp**.

One can derive (MUBR) from (MUG) by taking r to be $(e \mapsto e') -\!\!\ast\, p$ and using the law $q \ast (q -\!\!\ast\, p) \Rightarrow p$:

$$\begin{array}{l}
\{(e \mapsto -) \ast ((e \mapsto e') -\!\!\ast\, p)\} \\
[e] := e' \\
\{(e \mapsto e') \ast ((e \mapsto e') -\!\!\ast\, p)\} \\
\{p\}.
\end{array}$$

One can go in the opposite direction by taking p to be $(e \mapsto e') \ast r$ and using the law $(p \ast r) \Rightarrow (p \ast (q -\!\!\ast\, (q \ast r)))$:

$$\begin{array}{l}
\{(e \mapsto -) \ast r\} \\
\{(e \mapsto -) \ast ((e \mapsto e') -\!\!\ast\, ((e \mapsto e') \ast r))\} \\
[e] := e' \\
\{(e \mapsto e') \ast r\}.
\end{array}$$

Inference Rules for Deallocation

The local form (DISL):

$$\overline{\{e \mapsto -\}\ \textbf{dispose}\ e\ \{\textbf{emp}\}.}$$

The global (and backward-reasoning) form (DISG):

$$\overline{\{(e \mapsto -) \ast r\}\ \textbf{dispose}\ e\ \{r\}.}$$

One can derive (DISG) from (DISL) by using (FR); one can go in the opposite direction by taking r to be **emp**.

Inference Rules for Noninterfering Allocation

The local form (CONSNIL):

$$\frac{}{\{\mathbf{emp}\}\ v := \mathbf{cons}(e_0, \ldots, e_{n-1})\ \{v \mapsto e_0, \ldots, e_{n-1}\},}$$

where $v \notin \mathrm{FV}(e_0, \ldots, e_{n-1})$.

The global form (CONSNIG):

$$\frac{}{\{r\}\ v := \mathbf{cons}(e_0, \ldots, e_{n-1})\ \{(v \mapsto e_0, \ldots, e_{n-1})\ *\ r\},}$$

where $v \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, r)$.

One can derive (CONSNIG) from (CONSNIL) by using the frame rule (FR); one can go in the opposite direction by taking r to be **emp**.

Inference Rules for General Allocation

The local form (CONSL):

$$\frac{}{\{v = v' \wedge \mathbf{emp}\}\ v := \mathbf{cons}(e_0, \ldots, e_{n-1})\ \{v \mapsto e'_0, \ldots, e'_{n-1}\}.}$$

where v' is distinct from v, and e'_i denotes $e_i/v \to v'$.

The global form (CONSG):

$$\frac{}{\{r\}\ v := \mathbf{cons}(e_0, \ldots, e_{n-1})\ \{\exists v'.\ (v \mapsto e'_0, \ldots, e'_{n-1})\ *\ r'\},}$$

where v' is distinct from v, $v' \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, r)$, e'_i denotes $e_i/v \to v'$, and r' denotes $r/v \to v'$.

The backward-reasoning form (CONSBR):

$$\frac{}{\{\forall v''.\ (v'' \mapsto e_0, \ldots, e_{n-1}) \mathrel{-\!\!*} p''\}\ v := \mathbf{cons}(e_0, \ldots, e_{n-1})\ \{p\},}$$

where v'' is distinct from v, $v'' \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, p)$, and p'' denotes $p/v \to v''$.

Inference Rules for General Allocation (continued)

Let $\overline{e}$ abbreviate $e_0, \ldots, e_{n-1}$, $\overline{e}'$ abbreviate $e'_0, \ldots, e'_{n-1}$, and similarly for e''. One can derive (CONSG) from (CONSL) by using (FR) and (EQ):

$$\begin{array}{ll}
\{v = v' \wedge r'\} & \{r\} \\
\{v = v' \wedge (\mathbf{emp} \ * \ r')\} & \{\exists v'.\, v = v' \wedge r\} \\
\{(v = v' \wedge \mathbf{emp}) \ * \ r'\} & \{\exists v'.\, v = v' \wedge r'\} \\
v := \mathbf{cons}(\overline{e}) & v := \mathbf{cons}(\overline{e}) \\
\{(v \mapsto \overline{e}') \ * \ r'\}, & \{\exists v'.\, (v \mapsto \overline{e}') \ * \ r'\}.
\end{array}$$

To go in the other direction:

$$\begin{array}{l}
\{v = v' \wedge \mathbf{emp}\} \\
v := \mathbf{cons}(\overline{e}) \\
\{\exists v''.\, (v \mapsto \overline{e}'') \ * \ (v'' = v' \wedge \mathbf{emp})\} \\
\{\exists v''.\, ((v \mapsto \overline{e}'') \ * \ \mathbf{emp}) \wedge v'' = v'\} \\
\{\exists v''.\, (v \mapsto \overline{e}'') \wedge v'' = v'\} \\
\{\exists v''.\, (v \mapsto \overline{e}') \wedge v'' = v'\} \\
\{v \mapsto \overline{e}'\}.
\end{array}$$

Inference Rules for General Allocation (continued)

In order to derive (CONSBR) from (CONSG), we choose $v' \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, p)$ to be distinct from v and v'', take r to be $\forall v''.\ (v'' \mapsto \overline{e}) -\!\!* \ p''$, and use various laws about quantifiers, as well as $q \ * \ (q -\!\!* \ p) \Rightarrow p$:

$$
\begin{array}{l}
\{\forall v''.\ (v'' \mapsto \overline{e}) -\!\!* \ p''\} \\
v := \mathbf{cons}(\overline{e}) \\
\{\exists v'.\ (v \mapsto \overline{e}') \ * \ (\forall v''.\ (v'' \mapsto \overline{e}') -\!\!* \ p'')\} \\
\{\exists v'.\ (v \mapsto \overline{e}') \ * \ ((v \mapsto \overline{e}') -\!\!* \ p)\} \\
\{\exists v'.\ p\} \\
\{p\}.
\end{array}
$$

To go in the other direction, we choose $v'' \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, r)$ to be distinct from v and v', take p to be $\exists v'.\ (v \mapsto \overline{e}') \ * \ r'$, and use various laws about quantifiers, as well as $r \Rightarrow (q -\!\!* \ (q \ * \ r))$:

$$
\begin{array}{l}
\{r\} \\
\{\forall v''.\ r\} \\
\{\forall v''.\ (v'' \mapsto \overline{e}) -\!\!* \ ((v'' \mapsto \overline{e}) \ * \ r)\} \\
\{\forall v''.\ (v'' \mapsto \overline{e}) -\!\!* \ (((v'' \mapsto \overline{e}') \ * \ r')/v' \to v)\} \\
\{\forall v''.\ (v'' \mapsto \overline{e}) -\!\!* \ (\exists v'.\ (v'' \mapsto \overline{e}') \ * \ r')\} \\
v := \mathbf{cons}(\overline{e}) \\
\{\exists v'.\ (v \mapsto \overline{e}') \ * \ r'\}.
\end{array}
$$

Inference Rules for Lookup

The local form (LKL):

$$\frac{}{\{v = v' \wedge (e \mapsto v'')\}\ v := [e]\ \{v = v'' \wedge (e' \mapsto v'')\},}$$

where v, v', and v'' are distinct, and e' denotes $e/v \to v'$.

The global form (LKG):

$$\frac{}{\begin{array}{l}\{\exists v''.\ (e \mapsto v'')\ *\ (r/v' \to v)\}\ v := [e] \\ \qquad\qquad\qquad\qquad \{\exists v'.\ (e' \mapsto v)\ *\ (r/v'' \to v)\},\end{array}}$$

where v, v', and v'' are distinct, $v', v'' \notin \mathrm{FV}(e)$, $v \notin \mathrm{FV}(r)$, and e' denotes $e/v \to v'$.

The backward-reasoning form (LKBR):

$$\frac{}{\{\exists v''.\ (e \hookrightarrow v'') \wedge p''\}\ v := [e]\ \{p\},}$$

where $v'' \notin \mathrm{FV}(e) \cup (\mathrm{FV}(p) - \{v\})$, and p'' denotes $p/v \to v''$.

The global noninterfering form (LKNIG):

$$\frac{}{\{\exists v''.\ (e \mapsto v'')\ *\ p''\}\ v := [e]\ \{(e \mapsto v)\ *\ p\},}$$

where $v \notin \mathrm{FV}(e)$, $v'' \notin \mathrm{FV}(e) \cup (\mathrm{FV}(p) - \{v\})$, and p'' denotes $p/v \to v''$.

Inference Rules for Lookup (continued)

To derive (LKG) from (LKL), we use the frame rule (FR):

$$\{v = v' \wedge (e \mapsto v'') \ * \ (r/v' \to v)\}$$
$$\{v = v' \wedge (e \mapsto v'') \ * \ r\}$$
$$v := [e]$$
$$\{v = v'' \wedge (e' \mapsto v'') \ * \ r\}$$
$$\{v = v'' \wedge (e' \mapsto v) \ * \ (r/v'' \to v)\},$$

and the rule for existential quantification (EQ):

$$\{\exists v''.\ (e \mapsto v'') \ * \ (r/v' \to v)\}$$
$$\{\exists v', v''.\ v = v' \wedge (e \mapsto v'') \ * \ (r/v' \to v)\}$$
$$v := [e]$$
$$\{\exists v', v''.\ v = v'' \wedge (e' \mapsto v) \ * \ (r/v'' \to v)\}$$
$$\{\exists v'.\ (e' \mapsto v) \ * \ (r/v'' \to v)\}.$$

To derive (LKBR) from (LKG), we take r to be $(e' \mapsto v'') \mathrel{-\!\!*} p''$ and use the axiom schemata $(e \hookrightarrow e') \wedge p \Rightarrow (e \mapsto e') \ * \ ((e \mapsto e') \mathrel{-\!\!*} p)$ and $q \ * \ (q \mathrel{-\!\!*} p) \Rightarrow p$. (In the first line, we can rename v'' to be any variable not in $\mathrm{FV}(e) \cup (\mathrm{FV}(p) - \{v\})$.)

$$\{\exists v''.\ (e \hookrightarrow v'') \wedge p''\}$$
$$\{\exists v''.\ (e \mapsto v'') \ * \ ((e \mapsto v'') \mathrel{-\!\!*} p'')\}$$
$$v := [e]$$
$$\{\exists v'.\ (e' \mapsto v) \ * \ ((e' \mapsto v) \mathrel{-\!\!*} p)\}$$
$$\{\exists v'.\ p\}$$
$$\{p\}.$$

Inference Rules for Lookup (continued)

To derive (LKL) from (LKBR), we take p to be $v = v'' \wedge (e' \mapsto v'')$, after renaming v'' to $\hat{v}$:

$$
\begin{aligned}
&\{v = v' \wedge (e \mapsto v'')\}\\
&\{v = v' \wedge (e \hookrightarrow v'') \wedge (e' \mapsto v'')\}\\
&\{(e \hookrightarrow v'') \wedge (e' \mapsto v'')\}\\
&\{\exists \hat{v}.\ (e \hookrightarrow v'') \wedge \hat{v} = v'' \wedge (e' \mapsto v'')\}\\
&\{\exists \hat{v}.\ (e \hookrightarrow \hat{v}) \wedge \hat{v} = v'' \wedge (e' \mapsto v'')\}\\
&v := [e]\\
&\{v = v'' \wedge (e' \mapsto v'')\}.
\end{aligned}
$$

To derive (LKNIG) from (LKG), suppose v and v'' are distinct variables, $v \notin \mathrm{FV}(e)$, and $v'' \notin \mathrm{FV}(e) \cup \mathrm{FV}(p)$. We take v' to be a variable distinct from v and v'' that does not occur free in e or p, and r to be $p'' = p/v \to v''$. Then

$$
\begin{aligned}
&\{\exists v''.\ (e \mapsto v'') \ * \ p''\}\\
&\{\exists v''.\ (e \mapsto v'') \ * \ (p''/v' \to v)\}\\
&v := [e]\\
&\{\exists v'.\ (e' \mapsto v) \ * \ (p''/v'' \to v)\}\\
&\{\exists v'.\ (e \mapsto v) \ * \ p)\}\\
&\{(e \mapsto v) \ * \ p)\}.
\end{aligned}
$$

In the first line, we can rename v'' to be any variable not in $\mathrm{FV}(e) \cup (\mathrm{FV}(p) - \{v\})$.

Weakest Preconditions for the New Commands

The *weakest liberal precondition* of c and q, written $\mathrm{wlp}(c, q)$, is an assertion such that

$$\{\mathrm{wlp}(c, q)\}\ c\ \{q\} \text{ and } \forall p'.\ \{p'\}\ c\ \{q\} \text{ implies } p' \Rightarrow \mathrm{wlp}(c, q).$$

Similarly, the *weakest precondition* of c and q, written $\mathrm{wp}(c, q)$, is an assertion such that

$$[\ \mathrm{wp}(c, q)\ ]\ c\ [\ q\ ] \text{ and } \forall p'.\ [\ p'\ ]\ c\ [\ q\ ] \text{ implies } p' \Rightarrow \mathrm{wp}(c, q).$$

For each of the new heap-manipulating commands, the backward-reasoning form gives both the weakest liberal precondition and (since these commands always terminate) the weakest precondition.

Mutation:

$$\mathrm{wp}([e] := e', p) = (e \mapsto -) \ * \ ((e \mapsto e') \mathrel{-\!\!*} p).$$

Allocation:

$$\mathrm{wp}(v := \mathbf{cons}(e_0, \ldots, e_{n-1}), p) = \forall v''.\ (v'' \mapsto e_0, \ldots, e_{n-1}) \mathrel{-\!\!*} p'',$$

where v'' is distinct from v, $v'' \notin \mathrm{FV}(e_0, \ldots, e_{n-1}, p)$, and p'' denotes $p/v \to v''$.

Disposal:

$$\mathrm{wp}(\mathbf{dispose}\ e, r) = (e \mapsto -) \ * \ r.$$

Lookup:

$$\mathrm{wp}(v := [e], p) = \exists v''.\ (e \hookrightarrow v'') \wedge p'',$$

where $v'' \notin \mathrm{FV}(e) \cup (\mathrm{FV}(p) - \{v\})$, and p'' denotes $p/v \to v''$.

An Annotated Specification Revisited

$\{\textbf{emp}\}$

$\mathsf{x} := \textbf{cons}(\mathsf{a}, \mathsf{a})\ ;$ (CONSNIL)

$\{\mathsf{x} \mapsto \mathsf{a}, \mathsf{a}\}$ i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{a}\}$

$\mathsf{y} := \textbf{cons}(\mathsf{b}, \mathsf{b})\ ;$ (CONSNIG)

$\{(\mathsf{x} \mapsto \mathsf{a}, \mathsf{a}) \ast (\mathsf{y} \mapsto \mathsf{b}, \mathsf{b})\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{a} \ast \mathsf{y} \mapsto \mathsf{b} \ast \mathsf{y}+1 \mapsto \mathsf{b}\}$
$(p/v \to e \Rightarrow \exists v.\ p)$

$\{(\mathsf{x} \mapsto \mathsf{a}, -) \ast (\mathsf{y} \mapsto \mathsf{b}, \mathsf{b})\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast (\exists \mathsf{a}.\ \mathsf{x}+1 \mapsto \mathsf{a}) \ast \mathsf{y} \mapsto \mathsf{b} \ast \mathsf{y}+1 \mapsto \mathsf{b}\}$

$[\mathsf{x}+1] := \mathsf{y} - \mathsf{x}\ ;$ (MUG)

$\{(\mathsf{x} \mapsto \mathsf{a}, \mathsf{y}-\mathsf{x}) \ast (\mathsf{y} \mapsto \mathsf{b}, \mathsf{b})\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{y}-\mathsf{x} \ast \mathsf{y} \mapsto \mathsf{b} \ast \mathsf{y}+1 \mapsto \mathsf{b}\}$
$(p/v \to e \Rightarrow \exists v.\ p)$

$\{(\mathsf{x} \mapsto \mathsf{a}, \mathsf{y}-\mathsf{x}) \ast (\mathsf{y} \mapsto \mathsf{b}, -)\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{y}-\mathsf{x} \ast \mathsf{y} \mapsto \mathsf{b} \ast (\exists \mathsf{b}.\ \mathsf{y}+1 \mapsto \mathsf{b})\}$

$[\mathsf{y}+1] := \mathsf{x} - \mathsf{y}\ ;$ (MUG)

$\{(\mathsf{x} \mapsto \mathsf{a}, \mathsf{y}-\mathsf{x}) \ast (\mathsf{y} \mapsto \mathsf{b}, \mathsf{x}-\mathsf{y})\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{y}-\mathsf{x} \ast \mathsf{y} \mapsto \mathsf{b} \ast \mathsf{y}+1 \mapsto \mathsf{x}-\mathsf{y}\}$
$(p/v \to e \Rightarrow \exists v.\ p)$

$\{\exists \mathsf{o}.\ (\mathsf{x} \mapsto \mathsf{a}, \mathsf{o}) \ast (\mathsf{x}+\mathsf{o} \mapsto \mathsf{b}, -\mathsf{o})\}$
i.e., $\{\mathsf{x} \mapsto \mathsf{a} \ast \mathsf{x}+1 \mapsto \mathsf{o} \ast \mathsf{x}+\mathsf{o} \mapsto \mathsf{b} \ast \mathsf{x}+\mathsf{o}+1 \mapsto -\mathsf{o}\}$