GSep

John Wickerson

1 Preliminaries

1.1 Spatial closure operators

Suppose R and S are of type $\mathit{expr} \to \mathit{expr} \to \mathit{assertion}$. Define:

$$\begin{aligned}
R; S &\overset{\text{def}}{=} \lambda x\, z.\, \exists y.\, R\, x\, y \wedge S\, y\, z \\
R \vee S &\overset{\text{def}}{=} \lambda x\, y.\, R\, x\, y \vee S\, x\, y \\
\mathit{id} &\overset{\text{def}}{=} \lambda x\, y.\, x = y \wedge \mathit{emp}
\end{aligned}$$

Then let R^* be the least function satisfying $R^* = \mathit{id} \vee (R; R^*)$ and let $R^+ = R; R^*$.

2 GSep

GSep is RGSep without the R!

2.1 States and worlds

A state is a finite partial function from naturals to integers. A segmented state is a total mapping from segment names to states, of which a finite number are non-empty. A world comprises a 'local' state paired with a segmented state. The local state and all the shared states are pairwise disjoint.

$$\begin{aligned}
l, s, \sigma \in \mathsf{State} &\overset{\text{def}}{=} \mathbb{N} \rightharpoonup_{\text{fin}} \mathbb{Z} \\
w \in \mathsf{World} &\overset{\text{def}}{=} \{\langle l, S\rangle \in \mathsf{State} \times (\mathsf{Seg} \to_{\text{fin}} \mathsf{State}) \mid \text{defined}(l \cdot \odot S)\}
\end{aligned}$$

For states, $s \cdot s'$ is defined as $s \uplus s'$ when $\text{dom}\ s \cap \text{dom}\ s' = \emptyset$. We write $\odot$ for iterated $\cdot$. We write $\odot S$ as shorthand for $\odot_a\, S\, a$. For worlds, $\langle l, S\rangle \cdot \langle l', S'\rangle$ is defined as $\langle l \cdot l', S\rangle$ when $l \cdot l'$ is defined and $S = S'$. We lift $\cdot$ to $*$ in the usual way.

2.2 Assertions

The syntax of GSep assertions is as follows, where p is an ordinary separation logic assertion (a set of States):

$$P ::= \boxed{P}_a \mid \text{И} a{:}A.\, P \mid \text{И}(a{:}A, b{:}B).\, P \mid P * P \mid P \vee P \mid \exists x.\, P \mid p$$

Remark 1. Should include abstract predicates in the syntax too.

The semantics of an assertion is a set of worlds; that is:

$$[\![P]\!] : \mathcal{P}(\mathsf{World})$$

We define the semantics of assertions as follows.

$$\begin{aligned}
[\![p]\!] &\overset{\text{def}}{=} \{\langle l, S\rangle \mid l \models p\} \\
[\![P_0 * P_1]\!] &\overset{\text{def}}{=} [\![P_0]\!] * [\![P_1]\!] \\
[\![P_0 \vee P_1]\!] &\overset{\text{def}}{=} [\![P_0]\!] \cup [\![P_1]\!] \\
[\![\exists x.\, P]\!] &\overset{\text{def}}{=} \textstyle\bigcup_{v}.\, [\![P[v/x]]\!] \\
[\![\,\boxed{P}_a\,]\!] &\overset{\text{def}}{=} \{\langle \emptyset, S\rangle \mid \langle S\,a, S[a := \emptyset]\rangle \in [\![P]\!]\} \\
[\![\text{И} a{:}A.\, P]\!] &\overset{\text{def}}{=} \{\langle l \cdot S\,a, S[a := \emptyset]\rangle \mid \langle l, S\rangle \in [\![P]\!]\} \\
[\![\text{И}(a{:}A, b{:}B).\, P]\!] &\overset{\text{def}}{=} \{\langle l \cdot S\,a \cdot S\,b, S[a := \emptyset, b := \emptyset]\rangle \mid \langle l, S\rangle \in [\![P]\!]\}
\end{aligned}$$

2.3 Actions

Syntactically, an action is either a basic action '$G \mid P \rightsquigarrow Q \mid R$', or a finite union thereof. Semantically, an action denotes a set of pairs of worlds that can arise from transforming a given segment a in accordance with the action's specification. Notably, the action is free to transform the contents of other segments without restriction.

$$[\![a : A]\!] : \mathcal{P}(\mathsf{World}^2)$$

Consider firing the basic action '$G \mid P \rightsquigarrow Q \mid R$' on region a. The effect of the action is to allow any transformation to the world in which a part of segment a satisfying P has been replaced by a part satisf replaces a part of a given segment, say a, that satisfies P with a part satisfying Q, provided R also holds in a, and G holds of the local state. Segments other than a can be modified arbitrarily. We provide the following long-hand form:

$$\begin{array}{l}
\mathsf{action}\ A \\
\left\lfloor \begin{array}{ll}
\mathsf{pre} & P \\
\mathsf{post} & Q \\
\mathsf{context} & R \\
\mathsf{guard} & G
\end{array}\right.
\end{array}$$

This is its semantics:

$$[\![a : A]\!] \overset{\text{def}}{=} \left\{ \begin{matrix} (\langle l, S[a := \sigma \cdot \sigma'']\rangle\,, \\ \langle l', S'[a := \sigma' \cdot \sigma'']\rangle) \end{matrix} \;\middle|\; \begin{matrix} \langle \sigma, S[a := \emptyset]\rangle \in [\![P]\!] \wedge \langle \sigma', S'[a := \emptyset]\rangle \in [\![Q]\!] \wedge \\ \langle \sigma'', S[a := \emptyset]\rangle \in [\![R * \mathsf{tt}]\!] \wedge \langle l, S[a := \emptyset]\rangle \in [\![G * \mathsf{tt}]\!] \end{matrix} \right\}$$

We allow actions to be formed by the union of existing actions, like so:

$$[\![a : A_1 \cup A_2]\!] \overset{\text{def}}{=} [\![a : A_1]\!] \cup [\![a : A_2]\!]$$

2.4 Stability

Define '$\models P$ stable$_a$ A' to hold iff whenever $\langle l, S\rangle \in [\![P]\!]$ and $(\langle o, S\rangle, \langle o', S'\rangle) \in [\![a : A]\!]$ and defined$(l \cdot \odot S \cdot o)$ and defined$(l \cdot \odot S')$ then $\langle l, S'\rangle \in [\![P]\!]$. Define '$\models P$ stable Γ' to hold iff $\models P$ stable$_a$ A holds for all $(a : A) \in \Gamma$. Here are some (unchecked) proof rules for reasoning about stability.

$$\frac{\vdash P \text{ stable } \Gamma \qquad \vdash Q \text{ stable } \Gamma}{\vdash P * Q \text{ stable } \Gamma} \qquad \frac{\vdash P \text{ stable } \Gamma \qquad \vdash Q \text{ stable } \Gamma}{\vdash P \wedge Q \text{ stable } \Gamma}$$

$$\frac{\vdash P \text{ stable } \Gamma \qquad \vdash Q \text{ stable } \Gamma}{\vdash P \vee Q \text{ stable } \Gamma} \qquad \frac{(a : A) \in \Gamma \qquad \vdash P \text{ stable}_a\ A}{\vdash \boxed{P}_a \text{ stable } \Gamma} \qquad \frac{}{\vdash p \text{ stable } \Gamma}$$

$$\frac{\vdash P \text{ stable } \Gamma}{\vdash \text{И} a{:}A.\, P \text{ stable } \Gamma[a : A]}$$

2.5 GSep quadruples

First define $[\![\Gamma]\!] \overset{\text{def}}{=} \bigcap_{(a:A)\in\Gamma} [\![a : A]\!]$. Then the semantics of the GSep quadruple '$\Gamma \models \{P\}\, C\, \{Q\}$' is as follows:

$$\begin{aligned} &\forall l\, S\, \sigma\, \sigma'.\, (l \cdot \odot S) = \sigma \wedge \langle l, S\rangle \in [\![P]\!] \wedge (C, \sigma) \Downarrow \sigma' \\ &\qquad\qquad \Rightarrow \exists l'\, S'.\, \sigma' = (l' \cdot \odot S') \wedge \langle l', S'\rangle \in [\![Q]\!] \wedge (\langle l, S\rangle, \langle l', S'\rangle) \in [\![\Gamma]\!] \end{aligned}$$

2.6 Some GSep rules

Here is the frame rule:

$$\frac{\Gamma \vdash \{P\}\ C\ \{Q\} \qquad R \text{ stable } \Gamma}{\Gamma \vdash \{P * R\}\ C\ \{Q * R\}}\ \textsc{Frame}$$

Here is a rule for reading from a region (taken from Alias Logic).

$$\frac{}{\Gamma[a{:}A] \vdash \left\{\boxed{e \mapsto e' \ * \ P[e'/x]}_a\right\}\ x := [e]\ \left\{\boxed{e \mapsto e' \ * \ P}_a\right\}}\ \textsc{RegRead}$$

Here is a rule for hiding a region.

$$\frac{\Gamma[a{:}A] \vdash \{P\}\ C\ \{Q\} \qquad a \notin \operatorname{dom}\Gamma}{\Gamma \vdash \{\text{И}a{:}A.\, P\}\ C\ \{\text{И}a{:}A.\, Q\}}\ \textsc{Hide}$$

Here is a rule for weakening the guarantee.

$$\frac{\Gamma' \vdash \{P\}\ C\ \{Q\} \qquad [\![\Gamma']\!] \subseteq [\![\Gamma]\!]}{\Gamma \vdash \{P\}\ C\ \{Q\}}\ \textsc{GuarWeaken}$$

Here is a rule for hiding two regions simultaneously.

$$\frac{\Gamma[a{:}A, b{:}B] \vdash \{P\}\ C\ \{Q\} \qquad a, b \notin \operatorname{dom}\Gamma}{\Gamma \vdash \{\text{И}(a{:}A, b{:}B).\, P\}\ C\ \{\text{И}(a{:}A, b{:}B).\, Q\}}\ \textsc{TwoHide}$$

Here is a rule for updating a region. Note that we 'nullify' the value of a in Γ, rather than removing the mapping altogether, because P and Q might mention it.

$$\frac{\begin{array}{c}\Gamma[a{:}\emptyset] \vdash \{P' * P\}\ C\ \{Q' * Q\} \\ P, Q \text{ precise} \qquad [\![a : P' \mid P \rightsquigarrow Q \mid R]\!] \ \subseteq\ [\![a : A]\!]\end{array}}{\Gamma[a{:}A] \vdash \left\{P' * \boxed{P * R}_a\right\}\ C\ \left\{Q' * \boxed{Q * R}_a\right\}}\ \textsc{RegUpd}$$

Here is a rule for updating two regions simultaneously.

$$\frac{\begin{array}{c}\Gamma[a{:}\emptyset, b{:}\emptyset] \vdash \{P' * P_1 * P_2\}\ C\ \{Q' * Q_1 * Q_2\} \qquad P_1, Q_1, P_2, Q_2 \text{ precise} \\ [\![a : P' \mid P_1 \rightsquigarrow Q_1 \mid R_1]\!] \ \cap\ [\![b : P' \mid P_2 \rightsquigarrow Q_2 \mid R_2]\!] \ \subseteq\ [\![a : A]\!] \cap [\![b : B]\!]\end{array}}{\Gamma[a{:}A, b{:}B] \vdash \left\{P' \ * \ \boxed{P_1 * R_1}_a \ * \ \boxed{P_2 * R_2}_b\right\}\ C\ \left\{Q' \ * \ \boxed{Q_1 * R_1}_a \ * \ \boxed{Q_2 * R_2}_b\right\}}\ \textsc{TwoRegUpd}$$

Remark 2. In the RegUpd and TwoRegUpd rules, we may need to intersect with a relation that prohibits changes to other regions.

3 A singly-indexed list

Our first datastructure is a singly-indexed list. Every node has a value and a pointer to the next node. The final node's next pointer is set to 0. The first node is a sentinel, at a fixed location r. Our datastructure can be described by the following formulae:

$$\begin{array}{rcl} \mathit{list}\, r & \overset{\text{def}}{\iff} & \mathit{el}^+\, r\, 0 \\ x \in \mathit{list}\, r & \overset{\text{def}}{\iff} & \mathit{el}^+\, r\, x \ * \ \mathit{el}^+\, x\, 0 \end{array}$$

where:

$$\mathit{el}\, x\, y \ = \ x^1 \overset{.5}{\mapsto} \_ \ * \ x^2 \mapsto y$$

Our datastructure provides two methods: insertion and deletion. These are implemented as follows.

```
insert(x){
  int* p = r;
  while ([p+1]!=0 && ...) do p:=[p+1];
  [x+1]:=[p+1];
  [p+1]:=x;
}

remove(x){
  int* p = r;
  while ([p+1]!=x) do p:=[p+1];
  [p+1]:=[x+1];
}
```

We can specify these methods like so:

$$\begin{array}{l} c{:}C \vdash \left\{ \boxed{\mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \right\} \mathtt{insert(x)} \left\{ \boxed{\mathtt{x} \in \mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \right\} \\ c{:}C \vdash \left\{ \boxed{\mathtt{x} \in \mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \right\} \mathtt{remove(x)} \left\{ \boxed{\mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \right\} \end{array}$$

The region c denotes the module's internal state. Its interference, C, is as follows:

$$C \overset{\text{def}}{=} \bigcup_{x \in \mathbb{N}} \{\textsc{Add}\,x, \textsc{Rm}\,x\}$$

where:

$$\begin{array}{l} \mathsf{action}\ \textsc{Add}\,x \\ \left| \begin{array}{ll} \mathsf{pre} & p^2 \mapsto p' \\ \mathsf{post} & p^2 \mapsto x \;*\; x^1 \overset{.5}{\mapsto} v \;*\; x^2 \mapsto p' \\ \mathsf{context} & el^*\, r\, p \end{array} \right. \end{array}$$

$$\begin{array}{l} \mathsf{action}\ \textsc{Rm}\,x \\ \left| \begin{array}{ll} \mathsf{pre} & p^2 \mapsto x \;*\; x^1 \overset{.5}{\mapsto} v \;*\; x^2 \mapsto p' \\ \mathsf{post} & p^2 \mapsto p' \\ \mathsf{context} & el^*\, r\, p \\ \mathsf{guard} & x^1 \overset{.5}{\mapsto} \_ \end{array} \right. \end{array}$$

It is important to note that the pre- and post-conditions of `insert` and `remove` are stable under C.

3.1 Verification of the `insert` method

```
insert(x){
```

$$\left\{ \boxed{\mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \right\}$$

```
// begin frame
  $\left\{\boxed{list\,\mathtt{r}}_c\right\}$
  $\left\{\boxed{el^+\,\mathtt{r}\,0}_c\right\}$
  $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{r} \ast el^+\,\mathtt{r}\,0}_c\right\}$
  int* p = r; // using Hoare's assignment axiom
  $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast el^+\,\mathtt{p}\,0}_c\right\}$
  $\left\{\exists p'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast el\,\mathtt{p}\,p' \ast el^*\,p'\,0}_c\right\}$
  $\left\{\exists p'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto p' \ast el^*\,p'\,0}_c\right\}$
  // begin existential
    $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto p' \ast el^*\,p'\,0}_c\right\}$
    int* t = [p+1]; // using RegRead axiom
    $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
  // end existential
  $\left\{\exists p'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
  $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
  $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast el\,\mathtt{p}\,\mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
  while (t!=0 && ...) do {
    $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast el\,\mathtt{p}\,\mathtt{t} \ast el^*\,\mathtt{t}\,0}_c \ast \mathtt{t} \dot{\neq} 0\right\}$
    $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{t} \ast el^*\,\mathtt{t}\,0}_c \ast \mathtt{t} \dot{\neq} 0\right\}$
    $\left\{\exists t'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{t} \ast \mathtt{t}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{t}^2 \mapsto t' \ast el^*\,t'\,0}_c\right\}$
    p:=t; // using Hoare's assignment axiom
    $\left\{\exists t'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto t' \ast el^*\,t'\,0}_c\right\}$
    // begin existential
      $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto t' \ast el^*\,t'\,0}_c\right\}$
      t:=[p+1]; // using RegRead axiom
      $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
    // end existential
    $\left\{\exists t'.\,\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
    $\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \ast \mathtt{p}^1 \overset{.5}{\mapsto} \_ \ast \mathtt{p}^2 \mapsto \mathtt{t} \ast el^*\,\mathtt{t}\,0}_c\right\}$
  }
// end frame
```

```
$\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \;*\; \mathtt{p}^1 \overset{.5}{\mapsto} \_ \;*\; \mathtt{p}^2 \mapsto \mathtt{t} \;*\; el^*\,\mathtt{t}\,0}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_\right\}$
// begin null action on region $c$
  $\left\{\mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_\right\}$
  [x+1]:=t;
  $\left\{\mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \mathtt{t}\right\}$
// end null action
$\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \;*\; \mathtt{p}^1 \overset{.5}{\mapsto} \_ \;*\; \mathtt{p}^2 \mapsto \mathtt{t} \;*\; el^*\,\mathtt{t}\,0}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \mathtt{t}\right\}$
// begin action $\textsc{Add}\,\mathtt{x}$ on region $c$
  $\left\{\mathtt{p}^2 \mapsto \mathtt{t} \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \mathtt{t}\right\}$
  [p+1]:=x;
  $\left\{\mathtt{p}^2 \mapsto \mathtt{x} \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \;*\; \mathtt{x}^2 \mapsto \mathtt{t} \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v\right\}$
// end action
$\left\{\boxed{el^*\,\mathtt{r}\,\mathtt{p} \;*\; \mathtt{p}^1 \overset{.5}{\mapsto} \_ \;*\; \mathtt{p}^2 \mapsto \mathtt{x} \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \;*\; \mathtt{x}^2 \mapsto \mathtt{t} \;*\; el^*\,\mathtt{t}\,0}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v\right\}$
$\left\{\boxed{el^+\,\mathtt{r}\,\mathtt{x} \;*\; el^+\,\mathtt{x}\,0}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v\right\}$
$\left\{\boxed{\mathtt{x} \in \mathit{list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v\right\}$
}
```

4 Co-referring regions

We propose to describe the datastructure in a very different way. We shall see it as two *separate* lists (that is, we will use separating conjunction where previously we had ordinary conjunction). But in order to preserve the close relationship between the two lists (namely, that every node appearing in one list also appears in the other) we shall use 'ghost pointers', which map each element of one list to its counterpart in the other list. Here is our first attempt:

$$
\begin{aligned}
\mathit{2list}\,r &\overset{\text{def}}{\iff} \text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{+}_{b,r}\,r\,0}_a \;*\; \boxed{\widehat{el}^{+}_{a,r}\,r\,0}_b \\
x \in \mathit{2list}\,r &\overset{\text{def}}{\iff} \text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{+}_{b,r}\,r\,x \;*\; \widehat{el}^{+}_{b,r}\,x\,0}_a \;*\; \boxed{\widehat{el}^{+}_{a,r}\,r\,x \;*\; \widehat{el}^{+}_{a,r}\,x\,0}_b
\end{aligned}
$$

where:

$$
\begin{aligned}
el_0\,x\,y &\overset{\text{def}}{\iff} x^1 \overset{.25}{\mapsto} \_ \;*\; x^2 \mapsto y \\
el_1\,x\,y &\overset{\text{def}}{\iff} x^1 \overset{.25}{\mapsto} \_ \;*\; x^3 \mapsto y
\end{aligned}
$$

and:

$$
\begin{aligned}
in_{a,r}\,x &\overset{\text{def}}{\iff} \boxed{el_0^*\,r\,x \;*\; \mathsf{tt}}_a \\
in_{b,r}\,x &\overset{\text{def}}{\iff} \boxed{el_1^*\,r\,x \;*\; \mathsf{tt}}_b
\end{aligned}
$$

and:

$$\begin{aligned} \widehat{el}_{b,r}\,x\,y &\overset{\text{def}}{\iff} el_0\,x\,y \;*\; in_{b,r}\,x \\ \widehat{el}_{a,r}\,x\,y &\overset{\text{def}}{\iff} el_1\,x\,y \;*\; in_{a,r}\,x \end{aligned}$$

The predicate $\widehat{el}_{b,r}$ describes an element that appears in the first list. It uses the $in_{b,r}$ predicate to specify that the element appears in the second list too. From this and the symmetric fact about $\widehat{el}_{a,r}$, we can deduce that the two lists pass through exactly the same set of elements.

We specify the insert and remove methods in the same way as before (but now with the new implementation of the *2list* predicate):

$$\begin{aligned} c{:}C_{\mathtt{r}} \vdash \left\{ \boxed{\mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \;*\; \mathtt{x}^3 \mapsto \_ \right\} \mathtt{insert(x)} \left\{ \boxed{\mathtt{x} \in \mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \right\} \\ c{:}C_{\mathtt{r}} \vdash \left\{ \boxed{\mathtt{x} \in \mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \right\} \mathtt{remove(x)} \left\{ \boxed{\mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \;*\; \mathtt{x}^3 \mapsto \_ \right\} \end{aligned}$$

where

$$C_r \overset{\text{def}}{=} \bigcup_{x\in\mathbb{N}} \{\textsc{Add}_r\,x, \textsc{Rm}_r\,x\}$$

and

$$\begin{array}{l} \textsf{action } \textsc{Add}_r\,x \\ \left\lfloor \begin{array}{ll} \textsf{pre} & p^2 \mapsto p' \;*\; q^3 \mapsto q' \\ \textsf{post} & p^2 \mapsto x \;*\; q^3 \mapsto x \;*\; x^1 \overset{.5}{\mapsto} v \;*\; x^2 \mapsto p' \;*\; x^3 \mapsto q' \\ \textsf{context} & el_0^*\,r\,p \;*\; el_1^*\,r\,q \end{array} \right. \end{array}$$

$$\begin{array}{l} \textsf{action } \textsc{Rm}_r\,x \\ \left\lfloor \begin{array}{ll} \textsf{pre} & p^2 \mapsto x \;*\; q^3 \mapsto x \;*\; x^1 \overset{.5}{\mapsto} v \;*\; x^2 \mapsto p' \;*\; x^3 \mapsto q' \\ \textsf{post} & p^2 \mapsto p' \;*\; q^3 \mapsto q' \\ \textsf{context} & el_0^*\,r\,p \;*\; el_1^*\,r\,q \\ \textsf{guard} & x^1 \overset{.5}{\mapsto} v \end{array} \right. \end{array}$$

4.1 Verifying the insert method

```
$\left\{ \boxed{\mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \mapsto v \;*\; \mathtt{x}^2 \mapsto \_ \;*\; \mathtt{x}^3 \mapsto \_ \right\}$
insert(x){
  // begin frame
    $\left\{ \boxed{\mathit{2list}\,\mathtt{r}}_c \;*\; \mathtt{x}^1 \overset{.5}{\mapsto} v \;*\; \mathtt{x}^2 \mapsto \_ \;*\; \mathtt{x}^3 \mapsto \_ \right\}$
    // begin frame
      $\left\{ \boxed{\mathit{2list}\,\mathtt{r}}_c \right\}$
      $\left\{ \boxed{\text{И}a.\,\text{И}b.\, \boxed{\widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{r}\,0}_a \;*\; \boxed{\widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{r}\,0}_b}_c \right\}$
```

$$\left\{\boxed{\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{r} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{r}\,0}_a \ast \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{r} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{r}\,0}_b}_c\right\}$$

```
int* p = r;
int* q = r;
```

$$\left\{\boxed{\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{p}\,0}_a \ast \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{q}\,0}_b}_c\right\}$$

$$\left\{\exists p'\,q'.\,\boxed{\begin{array}{l}\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,p' \ast \widehat{el}^{+}_{b,\mathtt{r}}\,p'\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,q' \ast \widehat{el}^{+}_{a,\mathtt{r}}\,q'\,0}_b\end{array}}_c\right\}$$

```
int* t = [p+1];
int* u = [q+2];
```

$$\left\{\boxed{\begin{array}{l}\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}}_c\right\}$$

```
while (t!=0 && ...) do { p:=t; t:=[p+1]; }
while (u!=0 && ...) do { q:=u; u:=[q+2]; }
```

$$\left\{\boxed{\begin{array}{l}\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}}_c\right\}$$

```
// end frame
```

$$\left\{\begin{array}{l}\boxed{\begin{array}{l}\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}}_c\\ \ast\,\mathtt{x}^1 \overset{.5}{\mapsto} v \ast \mathtt{x}^2 \mapsto \_ \ast \mathtt{x}^3 \mapsto \_\end{array}\right\}$$

```
[x+1]:=t;
[x+2]:=u;
```

$$\left\{\begin{array}{l}\boxed{\begin{array}{l}\text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}}_c\\ \ast\,\mathtt{x}^1 \overset{.5}{\mapsto} v \ast \mathtt{x}^2 \mapsto \mathtt{t} \ast \mathtt{x}^3 \mapsto \mathtt{u}\end{array}\right\}$$

$$\left\{\begin{array}{l}\boxed{\begin{array}{l}el_0\,\mathtt{p}\,\mathtt{t} \ast el_1\,\mathtt{q}\,\mathtt{u} \ast\\ \left(\begin{array}{l}el_0\,\mathtt{p}\,\mathtt{t} \ast el_1\,\mathtt{q}\,\mathtt{u} \mathrel{-\!\!\ast}\\ \text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}\right)\end{array}}_c\\ \ast\,\mathtt{x}^1 \overset{.5}{\mapsto} v \ast \mathtt{x}^2 \mapsto \mathtt{t} \ast \mathtt{x}^3 \mapsto \mathtt{u}\end{array}\right\}$$

```
// tricky step (see below)
```

$$\left\{\begin{array}{l}\boxed{\begin{array}{l}el_0\,\mathtt{p}\,\mathtt{t} \ast el_1\,\mathtt{q}\,\mathtt{u} \ast\\ \left(\begin{array}{l}el_0\,\mathtt{p}\,\mathtt{x} \ast el_0\,\mathtt{x}\,\mathtt{t} \ast el_1\,\mathtt{q}\,\mathtt{x} \ast el_1\,\mathtt{x}\,\mathtt{u} \mathrel{-\!\!\ast}\\ \text{И}a.\,\text{И}b.\,\boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r}\,\mathtt{p} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{p}\,\mathtt{x} \ast \widehat{el}_{b,\mathtt{r}}\,\mathtt{x}\,\mathtt{t} \ast \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_a\\ \ast\,\boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r}\,\mathtt{q} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{q}\,\mathtt{x} \ast \widehat{el}_{a,\mathtt{r}}\,\mathtt{x}\,\mathtt{u} \ast \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_b\end{array}\right)\end{array}}_c\\ \ast\,\mathtt{x}^1 \overset{.5}{\mapsto} v \ast \mathtt{x}^2 \mapsto \mathtt{t} \ast \mathtt{x}^3 \mapsto \mathtt{u}\end{array}\right\}$$

// begin action $\textsc{Add}_{\mathtt{r}}\,\mathtt{x}$ on region c

$\left\{ el_0\,\mathtt{p\,t} * el_1\,\mathtt{q\,u} * \mathtt{x}^1 \overset{.5}{\mapsto} v * \mathtt{x}^2 \mapsto \mathtt{t} * \mathtt{x}^3 \mapsto \mathtt{u} \right\}$

```
[p+1]:=x;
[q+2]:=x;
```

$\left\{ el_0\,\mathtt{p\,x} * el_1\,\mathtt{q\,x} * \mathtt{x}^1 \overset{.5}{\mapsto} v * \mathtt{x}^2 \mapsto \mathtt{t} * \mathtt{x}^3 \mapsto \mathtt{u} \right\}$

$\left\{ el_0\,\mathtt{p\,x} * el_0\,\mathtt{x\,t} * el_1\,\mathtt{q\,x} * el_1\,\mathtt{x\,u} \right\}$

```
// end action
```

$$\left\{ \begin{array}{l} \boxed{\begin{array}{l} el_0\,\mathtt{p\,x} * el_0\,\mathtt{x\,t} * el_1\,\mathtt{q\,x} * el_1\,\mathtt{x\,u} * \\ \left(\begin{array}{l} el_0\,\mathtt{p\,x} * el_0\,\mathtt{x\,t} * el_1\,\mathtt{q\,x} * el_1\,\mathtt{x\,u} \mathrel{-\!\!*} \\ \text{И}a.\,\text{И}b.\, \boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r\,p} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{p\,x} * \widehat{el}^{n}_{b,\mathtt{r}}\,\mathtt{x\,t} * \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_{a} \\ \quad * \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r\,q} * \widehat{el}_{a,\mathtt{r}}\,\mathtt{q\,x} * \widehat{el}^{n}_{a,\mathtt{r}}\,\mathtt{x\,u} * \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_{b} \end{array} \right) \end{array}}_{c} \\ * \mathtt{x}^1 \overset{.5}{\mapsto} v * \mathtt{x}^2 \mapsto \mathtt{t} * \mathtt{x}^3 \mapsto \mathtt{u} \end{array} \right\}$$

$$\left\{ \boxed{\begin{array}{l} \text{И}a.\,\text{И}b.\, \boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r\,p} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{p\,x} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{x\,t} * \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_{a} \\ * \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r\,q} * \widehat{el}_{a,\mathtt{r}}\,\mathtt{q\,x} * \widehat{el}^{n}_{a,\mathtt{r}}\,\mathtt{x\,u} * \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_{b} \end{array}}_{c} \right\}$$

$\left\{ \boxed{\mathtt{x} \in \mathit{2list}\,\mathtt{r}}_{c} \right\}$

```
  // end frame
}
```

$\left\{ \boxed{\mathtt{x} \in \mathit{2list}\,\mathtt{r}}_{c} * \mathtt{x}^1 \overset{.5}{\mapsto} v \right\}$

The tricky step requires us to show that:

$$\begin{array}{l} el_0\,\mathtt{p\,t} * el_1\,\mathtt{q\,u} \mathrel{-\!\!*} \\ \text{И}a.\,\text{И}b.\, \boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r\,p} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{p\,t} * \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_{a} \\ \quad * \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r\,q} * \widehat{el}_{a,\mathtt{r}}\,\mathtt{q\,u} * \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_{b} \end{array}$$

implies

$$\begin{array}{l} el_0\,\mathtt{p\,x} * el_0\,\mathtt{x\,t} * el_1\,\mathtt{q\,x} * el_1\,\mathtt{x\,u} \mathrel{-\!\!*} \\ \text{И}a.\,\text{И}b.\, \boxed{\widehat{el}^{*}_{b,\mathtt{r}}\,\mathtt{r\,p} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{p\,x} * \widehat{el}_{b,\mathtt{r}}\,\mathtt{x\,t} * \widehat{el}^{+}_{b,\mathtt{r}}\,\mathtt{t}\,0}_{a} \\ * \boxed{\widehat{el}^{*}_{a,\mathtt{r}}\,\mathtt{r\,q} * \widehat{el}_{a,\mathtt{r}}\,\mathtt{q\,x} * \widehat{el}_{a,\mathtt{r}}\,\mathtt{x\,u} * \widehat{el}^{+}_{a,\mathtt{r}}\,\mathtt{u}\,0}_{b} \end{array}$$

That is, if we are obliged to provide length-1 chains from `p` to `t` and from `q` to `u`, then it suffices for us to provide length-2 chains from `p` to `x` to `t` and from `q` to `x` to `u`, and the resultant list will be duly extended by 1.