Stripping multiple laters, one step at a time
Introducing the “step-taking modality”

Jonas Kastberg Hinrichsen, Aarhus University

3. March 2023
Iris Seminar, Aarhus University

Outline

The presentation addresses the following:

- A brief coverage of what the later modality (the "later") is
- The ongoing story about adding and stripping more and more laters
- The challenges regarding laters with respect to **expressivity** and **presentation**
- A proposal to solve both of these problems: The step-taking modality $\mid\!\rightsquigarrow P$

Please do: ask questions, add missing clarifications, provide running feedback

What are laters?

Structural recursion through program steps (step-indexing):

$$\textsc{Later-intro}\ \frac{P}{\triangleright P} \qquad \textsc{Later-mono}\ \frac{P \vdash Q}{\triangleright P \vdash \triangleright Q} \qquad \textsc{Ht-frame-later}\ \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \triangleright R\}\, e\, \{w.\, Q * R\}}$$

What are laters?

Structural recursion through program steps (step-indexing):

$$\textsc{Later-intro}\quad \frac{P}{\triangleright P} \qquad \textsc{Later-mono}\quad \frac{P \vdash Q}{\triangleright P \vdash \triangleright Q} \qquad \textsc{Ht-frame-later}\quad \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \triangleright R\}\, e\, \{w.\, Q * R\}}$$

Higher-order ghost state (such as Invariants and propositional agreement):

$$\textsc{Ht-inv-open}\quad \frac{e \text{ is atomic} \qquad \{\triangleright P * Q\}\, e\, \{w.\, \triangleright P * R\}}{\left\{\boxed{P} * Q\right\}\, e\, \{w.\, R\}} \qquad \textsc{HO-agree}\quad \frac{\boxed{\mathrm{ag}(P)}^{\gamma} \qquad \boxed{\mathrm{ag}(Q)}^{\gamma}}{\triangleright(P = Q)}$$

What are laters?

Structural recursion through program steps (step-indexing):

$$\textsc{Later-intro}\quad \frac{P}{\triangleright P} \qquad \textsc{Later-mono}\quad \frac{P \vdash Q}{\triangleright P \vdash \triangleright Q} \qquad \textsc{Ht-frame-later}\quad \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \triangleright R\}\, e\, \{w.\, Q * R\}}$$

Higher-order ghost state (such as Invariants and propositional agreement):

$$\textsc{Ht-inv-open}\quad \frac{e \text{ is atomic} \qquad \{\triangleright P * Q\}\, e\, \{w.\, \triangleright P * R\}}{\left\{\boxed{P} * Q\right\}\, e\, \{w.\, R\}} \qquad \textsc{HO-agree}\quad \frac{\boxed{\mathrm{ag}(P)}^{\gamma} \qquad \boxed{\mathrm{ag}(Q)}^{\gamma}}{\triangleright(P = Q)}$$

"Co-inductive" definitions:

$$\texttt{is\_stream} \triangleq \mu X\ \ell.\, (\ell = \texttt{inl}()) \vee (\exists v, \ell'.\, \ell = \texttt{inr}(v, \ell') * \triangleright X\ \ell')$$

What are the challenges with laters?

People started putting them in various (sound) places, and they started to crop up in presented abstract specifications (ghost theories):

$$\frac{\text{prot\_ctx}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \text{prot\_own}_{\text{r}}\ \chi\ (\mathbf{?}\vec{x}:\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \text{prot\_ctx}\ \chi\ \vec{v_1}\ \ \vec{v_2} * \text{prot\_own}_{\text{r}}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}\ \textsc{proto-recv-r}$$

What are the challenges with laters?

People started putting them in various (sound) places, and they started to crop up in presented abstract specifications (ghost theories):

$$\dfrac{\text{prot\_ctx}\ \chi\ ([w] \cdot \vec{v_1})\ \vec{v_2} * \text{prot\_own}_{\mathsf{r}}\ \chi\ (\mathbf{?}\vec{x}\!:\!\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \text{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \text{prot\_own}_{\mathsf{r}}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}\ \textsc{proto-recv-r}$$

Expressivity: Fine! We do take a step whenever we use the ghost theory

What are the challenges with laters?

People started putting them in various (sound) places, and they started to crop up in presented abstract specifications (ghost theories):

$$\frac{\textsc{proto-recv-r}\qquad \mathsf{prot\_ctx}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Expressivity: Fine! We do take a step whenever we use the ghost theory
Presentation: How do we explain this specification to newcomers (and reviewers)?

“Dont think about it”

“Dont think about it”

“We can just get rid of it, when we take a step”

But then the problem multiplied.

Multiple laters

People started putting even more laters in (sound) places.

$$\dfrac{\textsc{proto-alloc}}{\Rrightarrow \exists \chi.\, \mathsf{prot\_ctx}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own_l}\ \chi\ \mathit{prot} * \mathsf{prot\_own_r}\ \chi\ \overline{\mathit{prot}}}$$

$$\textsc{proto-send-l}\quad \dfrac{\mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathbf{!}\,\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\Rrightarrow \triangleright^{|\vec{v_2}|}\,\mathsf{prot\_ctx}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

$$\textsc{proto-recv-r}\quad \dfrac{\mathsf{prot\_ctx}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright\, \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Multiple laters

People started putting even more laters in (sound) places.

$$\textsc{proto-alloc}\quad \Rrightarrow \exists \chi.\, \mathsf{prot\_ctx}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own_l}\ \chi\ \mathit{prot} * \mathsf{prot\_own_r}\ \chi\ \overline{\mathit{prot}}$$

$$\textsc{proto-send-l}\quad \frac{\mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathbf{!}\,\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\Rrightarrow \triangleright^{|\vec{v_2}|}\, \mathsf{prot\_ctx}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

$$\textsc{proto-recv-r}\quad \frac{\mathsf{prot\_ctx}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright\, \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Expressivity: We can just synchronise using a physical lock and take many more steps!

Multiple laters

People started putting even more laters in (sound) places.

PROTO-ALLOC

$$\Rrightarrow \exists \chi.\, \mathsf{prot\_ctx}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own_l}\ \chi\ \mathit{prot} * \mathsf{prot\_own_r}\ \chi\ \overline{\mathit{prot}}$$

PROTO-SEND-L

$$\frac{\mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathbf{!}\,\vec{x}\!:\!\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\Rrightarrow \triangleright^{|\vec{v_2}|}\ \mathsf{prot\_ctx}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

PROTO-RECV-R

$$\frac{\mathsf{prot\_ctx}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}\!:\!\vec{\tau}\,\langle v\rangle\{P\}.\,\mathit{prot})}{\Rrightarrow \triangleright \exists \vec{y}.\,(w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Expressivity: We can just synchronise using a physical lock and take many more steps!
Presentation: How do we present this to newcomers (and reviewers)???

You roll up your sleeves..

The need for skip instructions. The rules PROTO-SEND-L and PROTO-SEND-R from Figure 20 contain a number of later modalities ($\triangleright$) proportional to the other endpoint's buffer. As explained in § 9.3 these later modalities are the consequence of having to perform a number of case analyses on the subprotocol relation, which is defined using guarded recursion, and thus contains a later modality for each recursive unfolding.

To eliminate these later modalities, we instrument the code of the **send** function with the **skipN** (**llength** r) instruction, which performs a number of skips equal to the size of the other endpoint's buffer r. The **skipN** instruction has the following specification:

$$\{\triangleright^n P\}\, \texttt{skipN}\ n\, \{P\}$$

Instrumentation with skip instructions is used often in work on step-indexing, see *e.g.*, [SSB16; GST+20]. Instrumentation is needed because current step-indexed logics like Iris unify physical/program steps and logical steps, *i.e.*, for each physical/program step at most one later can be eliminated from the hypotheses. In recent work by Svendsen *et al.* [SSB16], Matsushita and Jourdan [MJ20], and Spies *et al.* [SGG+21] more liberal versions of step-indexing have been proposed, but none of these versions of step-indexing have been integrated into the main Coq development of Iris and HeapLang.

But what if you can only take one step?

e.g. if you have to put your ghost state in an invariant

Stripping multiple laters at every step

People came up with clever solutions to strip multiple laters during one step!

$$\textsc{Ht-step-get}\quad \frac{\{P * \text{⧖}\ 0\}\, e\, \{\Phi\}}{\{P\}\, e\, \{\Phi\}}$$

$$\textsc{Ht-step-incr}\quad \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \text{⧖}\ n\}\, e\, \{w.\, Q * \text{⧖}\ n+1\}}$$

$$\textsc{Ht-step-frame}\quad \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \text{⧖}\ n * \triangleright^{n} R\}\, e\, \{w.\, Q * R\}}$$

Stripping multiple laters at every step

People came up with clever solutions to strip multiple laters during one step!

$$\textsc{Ht-step-get}\ \frac{\{P * ⧖\, 0\}\; e\; \{\Phi\}}{\{P\}\; e\; \{\Phi\}} \qquad \textsc{Ht-step-incr}\ \frac{\{P\}\; e\; \{w.\, Q\}}{\{P * ⧖\, n\}\; e\; \{w.\, Q * ⧖\, n+1\}} \qquad \textsc{Ht-step-frame}\ \frac{\{P\}\; e\; \{w.\, Q\}}{\{P * ⧖\, n * \triangleright^{n} R\}\; e\; \{w.\, Q * R\}}$$

Expressivity: We can strip all the laters we need, by tracking a step lower bound inside our invariant that grows in tandem with our ghost state, so we can guarantee that we can always strip the appropriate laters using Ht-step-frame, while updating the lower bound at every step using Ht-step-incr!

Stripping multiple laters at every step

People came up with clever solutions to strip multiple laters during one step!

$$\textsc{Ht-step-get}\ \frac{\{P * ⧖\ 0\}\, e\, \{\Phi\}}{\{P\}\, e\, \{\Phi\}} \qquad \textsc{Ht-step-incr}\ \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * ⧖\ n\}\, e\, \{w.\, Q * ⧖\ n+1\}} \qquad \textsc{Ht-step-frame}\ \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * ⧖\ n * \triangleright^n R\}\, e\, \{w.\, Q * R\}}$$

Expressivity: We can strip all the laters we need, by tracking a step lower bound inside our invariant that grows in tandem with our ghost state, so we can guarantee that we can always strip the appropriate laters using Ht-step-frame, while updating the lower bound at every step using Ht-step-incr!
Presentation: ...?

Well..

$$\frac{\{P * \text{⧗}\, 0\}\ \langle ip;\ e\rangle\ \{\Phi\}}{\{P\}\ \langle ip;\ e\rangle\ \{\Phi\}}\ \textsc{Ht-step-get} \qquad \frac{\{P\}\ \langle ip;\ e\rangle\ \{w.\, Q\}}{\{P * \text{⧗}\, n\}\ \langle ip;\ e\rangle\ \{w.\, Q * \text{⧗}\, n+1\}}\ \textsc{Ht-step-incr}$$

$$\frac{\{P\}\ \langle ip;\ e\rangle\ \{w.\, Q\}}{\{P * \text{⧗}\, n * \triangleright^n R\}\ \langle ip;\ e\rangle\ \{w.\, Q * R\}}\ \textsc{Ht-step-frame} \qquad \frac{\text{⧗}\, n}{\text{⧗}\, n * \text{⧗}\, n}\ \textsc{step-dup}$$

Fig. 12. The mechanism for stripping multiple laters. We require e to be an atomic expression.

The shared logical context can then be captured as the following Iris invariant:

$$\begin{aligned}\exists Tl, Tr, Rl, Rr.\ &\mathsf{auth\_list}\ \chi_{\mathsf{Tl}}\ Tl * \mathsf{auth\_list}\ \chi_{\mathsf{Tr}}\ Tr * \mathsf{auth\_list}\ \chi_{\mathsf{Rl}}\ Rl * \mathsf{auth\_list}\ \chi_{\mathsf{Rr}}\ Rr * \\ &\mathsf{prot\_ctx}\ \chi_{\mathsf{chan}}\ (Tl - Rr)\ (Tr - Rl) * Rr \leq_{\mathsf{p}} Tl * Rl \leq_{\mathsf{p}} Tr * \text{⧗}\, |Tl| * \text{⧗}\, |Tr|\end{aligned}$$

Stripping multiple laters. In Iris, and thus Aneris, one can strip a later whenever a step of computation is taken. Conventionally the intuition is that one step equates stripping one later. However, recent discoveries [Matsushita et al. 2022; Mével et al. 2019; Spies et al. 2022] uncovered various methods for stripping *mutliple* laters per step. Based on these discoveries we extended Aneris with a similar, albeit more simplistic, mechanism as presented in Figure 12. The mechanism lets us strip multiple laters during one physical step, based on the amount of steps that has been taken thus far. The rule Ht-step-get lets us track a new lower bound of steps taken thus far $\text{⧗}\, 0$, and Ht-step-incr allows us to increase it by one, every time a step is taken. Crucially, the rule Ht-step-frame lets us frame resources under an amount of laters corresponding to the lower bound

What is the problem?

There is currently no way to present specification patterns without the context of the later-stripping mechanism.

What is the problem?

There is currently no way to present specification patterns without the context of the later-stripping mechanism.

Ideally we would like to *precisely* present the transitions in the ghost state, while *abstracting* over the later-stripping mechanism.

What is the problem?

There is currently no way to present specification patterns without the context of the later-stripping mechanism.
Ideally we would like to *precisely* present the transitions in the ghost state, while *abstracting* over the later-stripping mechanism.

WISHFUL-PROTO-ALLOC

$$\exists \chi.\, \mathsf{prot\_ctx}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ \mathit{prot} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ \overline{\mathit{prot}}$$

WISHFUL-PROTO-SEND-L

$$\frac{\mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ (\mathbf{!}\, \vec{x} : \vec{\tau}\ \langle v \rangle \{P\}.\, \mathit{prot}) * P[\vec{t}/\vec{x}]}{\mathsf{prot\_ctx}\ \chi\ (\vec{v_1} \cdot [v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

WISHFUL-PROTO-RECV-R

$$\frac{\mathsf{prot\_ctx}\ \chi\ ([w] \cdot \vec{v_1})\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ (\mathbf{?} \vec{x} : \vec{\tau}\ \langle v \rangle \{P\}.\, \mathit{prot})}{\exists \vec{y}.\, (w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Solution: Introducing the step-taking modality!

$$\models\!\!\rightsquigarrow P$$

Solution: Introducing the step-taking modality!

$$\mid\!\leadsto P$$

Recovers the intuition that we can get P *after taking a step*:

$$\dfrac{\{P\}\, e\, \{w.\, Q\}}{\{P * \mid\!\leadsto R\}\, e\, \{w.\, Q * R\}}\ \textsc{Ht-step-modality}$$

Solution: Introducing the step-taking modality!

$$|\!\!\rightsquigarrow P$$

Recovers the intuition that we can get P *after taking a step*:

$$\frac{\textsc{Ht-step-modality}\qquad \{P\}\, e\, \{w.\, Q\}}{\{P * |\!\!\rightsquigarrow R\}\, e\, \{w.\, Q * R\}}$$

Intentionally looks like the original later-stripping rule:

$$\frac{\textsc{Ht-later-frame}\qquad \{P\}\, e\, \{w.\, Q\}}{\{P * \triangleright R\}\, e\, \{w.\, Q * R\}}$$

Abstract specification of later-stripping mechanism

The step-taking modality can be used to express the later-stripping mechanism as an abstract specification pattern (rather than via Hoare triples):

$$\textsc{Step-step-get}\quad \mathrel{|\!\!\rightsquigarrow} \text{⧗}\, 0 \qquad\qquad \textsc{Step-step-incr}\quad \frac{\text{⧗}\, n}{\mathrel{|\!\!\rightsquigarrow} \text{⧗}\, n+1} \qquad\qquad \textsc{Step-step-frame}\quad \frac{\text{⧗}\, n * \triangleright^{n} P}{\mathrel{|\!\!\rightsquigarrow} P}$$

$$\textsc{Ht-step-modality}\quad \frac{\{P\}\, e\, \{w.\, Q\}}{\{P * \mathrel{|\!\!\rightsquigarrow} R\}\, e\, \{w.\, Q * R\}}$$

These rules supercede the former Hoare triple rules for the later-stripping mechanism

Abstract specifications of ghost theory

We can derive abstractions with the step-taking modality on top of each other:

$$\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} \triangleq \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * ⧖\, |\vec{v_1}| * ⧖\, |\vec{v_2}|$$

Step-proto-alloc

$$\models\!\!\leadsto \exists \chi.\, \texttt{prot\_ctx\_step}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own_l}\ \chi\ \mathit{prot} * \mathsf{prot\_own_r}\ \chi\ \overline{\mathit{prot}}$$

Step-proto-send-l

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathbf{!}\,\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\, \mathit{prot}) * P[\vec{t}/\vec{x}]}{\models\!\!\leadsto \texttt{prot\_ctx\_step}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

Step-proto-recv-r

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\, \mathit{prot})}{\models\!\!\leadsto \exists \vec{y}.\, (w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Abstract specifications of ghost theory

We can derive abstractions with the step-taking modality on top of each other:

$$\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} \triangleq \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * ⧖\ |\vec{v_1}| * ⧖\ |\vec{v_2}|$$

Step-proto-alloc

$$\vDash\!\!\leadsto \exists \chi.\, \texttt{prot\_ctx\_step}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ \mathit{prot} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ \overline{\mathit{prot}}$$

Step-proto-send-l

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ (\mathbf{!}\,\vec{x}\!:\!\vec{\tau}\ \langle v\rangle\{P\}.\, \mathit{prot}) * P[\vec{t}/\vec{x}]}{\vDash\!\!\leadsto \texttt{prot\_ctx\_step}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{l}}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

Step-proto-recv-r

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ (\mathbf{?}\vec{x}\!:\!\vec{\tau}\ \langle v\rangle\{P\}.\, \mathit{prot})}{\vDash\!\!\leadsto \exists \vec{y}.\, (w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own}_{\mathsf{r}}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Expressivity: Fine! We do take a step whenever we use the ghost theory

Abstract specifications of ghost theory

We can derive abstractions with the step-taking modality on top of each other:

$$\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} \triangleq \mathsf{prot\_ctx}\ \chi\ \vec{v_1}\ \vec{v_2} * ⧖\ |\vec{v_1}| * ⧖\ |\vec{v_2}|$$

Step-proto-alloc

$$\models\!\!\rightsquigarrow \exists \chi.\, \texttt{prot\_ctx\_step}\ \chi\ \epsilon\ \epsilon * \mathsf{prot\_own_l}\ \chi\ \mathit{prot} * \mathsf{prot\_own_r}\ \chi\ \overline{\mathit{prot}}$$

Step-proto-send-l

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathbf{!}\,\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\, \mathit{prot}) * P[\vec{t}/\vec{x}]}{\models\!\!\rightsquigarrow \texttt{prot\_ctx\_step}\ \chi\ (\vec{v_1}\cdot[v[\vec{t}/\vec{x}]])\ \vec{v_2} * \mathsf{prot\_own_l}\ \chi\ (\mathit{prot}[\vec{t}/\vec{x}])}$$

Step-proto-recv-r

$$\frac{\texttt{prot\_ctx\_step}\ \chi\ ([w]\cdot\vec{v_1})\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ (\mathbf{?}\vec{x}{:}\vec{\tau}\,\langle v\rangle\{P\}.\, \mathit{prot})}{\models\!\!\rightsquigarrow \exists \vec{y}.\, (w = v[\vec{y}/\vec{x}]) * P[\vec{y}/\vec{x}] * \texttt{prot\_ctx\_step}\ \chi\ \vec{v_1}\ \vec{v_2} * \mathsf{prot\_own_r}\ \chi\ \mathit{prot}[\vec{y}/\vec{x}]}$$

Expressivity: Fine! We do take a step whenever we use the ghost theory
Presentation: How do we explain this specification to newcomers (and reviewers)?

"Dont think about it"

“Dont think about it”

“We can just get rid of it, when we take a step”

Hidden benefit of the step-taking modality abstracting over laters

Derived abstractions may hide the number of laters that is needed to be stripped

Hidden benefit of the step-taking modality abstracting over laters

Derived abstractions may hide the number of laters that is needed to be stripped:

$$\textsc{SesCrow-Init}\quad \Rrightarrow \exists \chi.\, \texttt{ses\_own}\ \chi\ \texttt{left}\ 0\ 0\ \mathit{prot} * \texttt{ses\_own}\ \chi\ \texttt{right}\ 0\ 0\ \overline{\mathit{prot}}$$

$$\textsc{SesCrow-Send}\quad \frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{!}\,(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\Rrightarrow \triangleright\, \texttt{ses\_own}\ \chi\ s\ (n+1)\ m\ (\mathit{prot}[\vec{t}/\vec{x}]) * \texttt{ses\_idx}\ \chi\ s\ n\ (v[\vec{t}/\vec{x}])}$$

$$\textsc{SesCrow-Recv}\quad \frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{?}(\vec{x}:\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * \texttt{ses\_idx}\ \chi\ \overline{s}\ m\ w}{\Rrightarrow \triangleright^{???}\, \exists(\vec{y}:\vec{\tau}).\, \texttt{ses\_own}\ \chi\ s\ n\ (m+1)\ (\mathit{prot}[\vec{y}/\vec{x}]) * w = v[\vec{y}/\vec{x}] * P[\vec{y}/\vec{x}]}$$

Hidden benefit of the step-taking modality abstracting over laters

Derived abstractions may hide the number of laters that is needed to be stripped:

SESCROW-INIT

$$\mathrel{|\!\!\leadsto} \exists \chi.\, \texttt{ses\_own}\ \chi\ \texttt{left}\ 0\ 0\ \mathit{prot} * \texttt{ses\_own}\ \chi\ \texttt{right}\ 0\ 0\ \overline{\mathit{prot}}$$

SESCROW-SEND

$$\frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{!}\,(\vec{x}\!:\!\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\mathrel{|\!\!\leadsto} \texttt{ses\_own}\ \chi\ s\ (n+1)\ m\ (\mathit{prot}[\vec{t}/\vec{x}]) * \texttt{ses\_idx}\ \chi\ s\ n\ (v[\vec{t}/\vec{x}])}$$

SESCROW-RECV

$$\frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{?}(\vec{x}\!:\!\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * \texttt{ses\_idx}\ \chi\ \overline{s}\ m\ w}{\mathrel{|\!\!\leadsto} \exists(\vec{y} : \vec{\tau}).\, \texttt{ses\_own}\ \chi\ s\ n\ (m+1)\ (\mathit{prot}[\vec{y}/\vec{x}]) * w = v[\vec{y}/\vec{x}] * P[\vec{y}/\vec{x}]}$$

Hidden benefit of the step-taking modality abstracting over laters

Derived abstractions may hide the number of laters that is needed to be stripped:

SESCROW-INIT

$$\models\!\leadsto \exists \chi.\, \texttt{ses\_own}\ \chi\ \texttt{left}\ 0\ 0\ \mathit{prot} * \texttt{ses\_own}\ \chi\ \texttt{right}\ 0\ 0\ \overline{\mathit{prot}}$$

SESCROW-SEND

$$\frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{!}\,(\vec{x}{:}\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * P[\vec{t}/\vec{x}]}{\models\!\leadsto \texttt{ses\_own}\ \chi\ s\ (n+1)\ m\ (\mathit{prot}[\vec{t}/\vec{x}]) * \texttt{ses\_idx}\ \chi\ s\ n\ (v[\vec{t}/\vec{x}])}$$

SESCROW-RECV

$$\frac{\texttt{ses\_own}\ \chi\ s\ n\ m\ (\mathbf{?}(\vec{x}{:}\vec{\tau})\,\langle v\rangle\{P\}.\,\mathit{prot}) * \texttt{ses\_idx}\ \chi\ \overline{s}\ m\ w}{\models\!\leadsto \exists(\vec{y} : \vec{\tau}).\, \texttt{ses\_own}\ \chi\ s\ n\ (m+1)\ (\mathit{prot}[\vec{y}/\vec{x}]) * w = v[\vec{y}/\vec{x}] * P[\vec{y}/\vec{x}]}$$

This abstract specificiation pattern is *undefinable* without the step-taking modality.

The step-taking modality definition

$$|\!\!\leadsto P \triangleq \forall n.\, ⧗_{\bullet}\, n \Rrightarrow (⧗_{\bullet}\, n * (\triangleright^{n+1} ⧗_{\bullet}\, n+1 \Rrightarrow ⧗_{\bullet}\, n+1 * P))$$

The step-taking modality definition

$$\models\!\rightsquigarrow P \triangleq \forall n.\, ⧖_{\bullet}\, n \Rrightarrow (⧖_{\bullet}\, n * (\triangleright^{n+1} ⧖_{\bullet}\, n+1 \Rrightarrow ⧖_{\bullet}\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧖_{\bullet}\, n$)

The step-taking modality definition

$$\models\!\!\rightsquigarrow P \triangleq \forall n.\, ⧗_\bullet\, n \Rrightarrow (⧗_\bullet\, n * (\triangleright^{n+1} ⧗_\bullet\, n+1 \Rrightarrow ⧗_\bullet\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧗_\bullet\, n$)
- Take a ghost step (in which n may be approximated)

The step-taking modality definition

$$\models\!\!\rightsquigarrow P \triangleq \forall n.\, ⌛_{\bullet}\, n \Rrightarrow (⌛_{\bullet}\, n * (\triangleright^{n+1}\, ⌛_{\bullet}\, n+1 \Rrightarrow ⌛_{\bullet}\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⌛_{\bullet}\, n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($⌛_{\bullet}\, n$)

The step-taking modality definition

$$\models_{\rightsquigarrow} P \triangleq \forall n.\ \text{⧖}_{\bullet}\ n \Rrightarrow (\text{⧖}_{\bullet}\ n * (\triangleright^{n+1}\ \text{⧖}_{\bullet}\ n+1 \Rrightarrow \text{⧖}_{\bullet}\ n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($\text{⧖}_{\bullet}\ n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($\text{⧖}_{\bullet}\ n$)
- Strip laters according to the authorty ($\triangleright^{n}$)

The step-taking modality definition

$$\mid\!\rightsquigarrow P \triangleq \forall n.\, ⧖_\bullet\, n \Rrightarrow (⧖_\bullet\, n * (\triangleright^{n+1} ⧖_\bullet\, n+1 \Rrightarrow ⧖_\bullet\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧖_\bullet\, n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($⧖_\bullet\, n$)
- Strip laters according to the authorty ($\triangleright^n$)
- Get the updated step-taking authority, as a step has been taken ($⧖_\bullet\, n$)

The step-taking modality definition

$$\mid\!\!\rightsquigarrow P \triangleq \forall n.\, ⧖_{\bullet}\; n \Rrightarrow (⧖_{\bullet}\; n * (\triangleright^{n+1} ⧖_{\bullet}\; n+1 \Rrightarrow ⧖_{\bullet}\; n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧖_{\bullet}\; n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($⧖_{\bullet}\; n$)
- Strip laters according to the authorty ($\triangleright^{n}$)
- Get the updated step-taking authority, as a step has been taken ($⧖_{\bullet}\; n$)
- Take a ghost step (in which local step lower bounds can be updated)

The step-taking modality definition

$$|\!\leadsto P \triangleq \forall n.\, \text{⧖}_{\bullet}\, n \Rrightarrow (\text{⧖}_{\bullet}\, n * (\triangleright^{n+1}\, \text{⧖}_{\bullet}\, n+1 \Rrightarrow \text{⧖}_{\bullet}\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($\text{⧖}_{\bullet}\, n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($\text{⧖}_{\bullet}\, n$)
- Strip laters according to the authorty ($\triangleright^{n}$)
- Get the updated step-taking authority, as a step has been taken ($\text{⧖}_{\bullet}\, n$)
- Take a ghost step (in which local step lower bounds can be updated)
- Give back the step-taking authority ($\text{⧖}_{\bullet}\, n$)

The step-taking modality definition

$$\models\!\!\rightsquigarrow P \triangleq \forall n.\, ⧖_{\bullet}\; n \Rrightarrow (⧖_{\bullet}\; n * (\triangleright^{n+1}\, ⧖_{\bullet}\; n+1 \Rrightarrow ⧖_{\bullet}\; n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧖_{\bullet}\; n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($⧖_{\bullet}\; n$)
- Strip laters according to the authorty ($\triangleright^{n}$)
- Get the updated step-taking authority, as a step has been taken ($⧖_{\bullet}\; n$)
- Take a ghost step (in which local step lower bounds can be updated)
- Give back the step-taking authority ($⧖_{\bullet}\; n$)
- Show P

The step-taking modality definition

$$\models\!\!\leadsto P \triangleq \forall n.\, ⧖_{\bullet}\, n \Rrightarrow (⧖_{\bullet}\, n * (\triangleright^{n+1} ⧖_{\bullet}\, n+1 \Rrightarrow ⧖_{\bullet}\, n+1 * P))$$

It reads as follows:

- Get the step-taking authority ($⧖_{\bullet}\, n$)
- Take a ghost step (in which n may be approximated)
- Give back the step-taking authority ($⧖_{\bullet}\, n$)
- Strip laters according to the authorty ($\triangleright^{n}$)
- Get the updated step-taking authority, as a step has been taken ($⧖_{\bullet}\, n$)
- Take a ghost step (in which local step lower bounds can be updated)
- Give back the step-taking authority ($⧖_{\bullet}\, n$)
- Show P

OBS: details about masks are omitted for brevity sake

The step-taking modality properties

The step-taking modality enjoys a mix of the rules for the later modality and the ghost update modality:

$$\textsc{Step-intro}\quad \frac{P}{\mid\!\leadsto P} \qquad \textsc{Step-mono}\quad \frac{P \vdash Q}{\mid\!\leadsto P \vdash \mid\!\leadsto Q} \qquad \textsc{Step-upd}\quad \frac{\mid\!\Rrightarrow \mid\!\leadsto \mid\!\Rrightarrow P}{\mid\!\leadsto P} \qquad \textsc{Step-sep-comm}\quad \frac{\mid\!\leadsto P * \mid\!\leadsto Q}{\mid\!\leadsto P * Q}$$

These lets us derive abstract specification patterns on top of each other, without breaking abstraction

Further motivation for the step-taking modality

A valid concern is that the step-taking modality will become the new later modality

$$\mid\!\rightsquigarrow^{n}$$

The intention is that this wont happen.
Regardless of how later-stripping mechanisms evolved, the step-taking modality should always capture the notion of being able to strip however many laters is available during *one* step!
If multiple step-taking modalities are iterated, that should semantically mean that multiple steps are *intended* to be taken.

Questions?