Appendix to Diaframe: Automated Verification for Fine-Grained Concurrency in Iris

Ike Mulder[1], Robbert Krebbers[1], and Herman Geuvers[1]

[1]Radboud University Nijmegen

December 8, 2021

1 Definition of token counter

For various programs for which we prove functional correctness, it is useful to have a notion of a 'token-counting' object. Three kinds of resources are associated with this object:

- the token-requirement resource $P\,1$, representing full ownership of some resource
- a token $\texttt{token}\ P\ \gamma$, which represents fractional ownership of the above resource associated to ghost-name γ
- the token-counter resource $\texttt{counter}\ P\ \gamma\ p$, which states that there are exactly $p > 0$ tokens in existence that are associated to ghost-name γ. Additionally, this resource holds the remaining part of the ownership of the `req` resource.
- the no-tokens resource $\texttt{no\_tokens}\ P\ \gamma\ q$, stating the knowledge that no tokens currently exist associated to ghost-name γ, and that we own $0 < q \leq 1$ of that knowledge. The $\texttt{no\_tokens}\ P\ \gamma$ from the paper stood for $\texttt{no\_tokens}\ P\ \gamma\ \frac{1}{2}$.

The rules we need for token-counting object are shown in Figure 1. We will now show how we *define* such objects.

1.1 Instantiating token counters

In our foundational setting we have to prove there exist token-counting objects that satisfy the above rules. To do so, we can use an authoritative resource on an optional pair of positives and positive fractions that are ≤ 1, both with addition as operation. The definition is as follows:

$$\texttt{token}\ P\ \gamma \triangleq \exists q.\ \boxed{\circ\ \mathrm{Some}\ (1, q)}^{\gamma} * P\,q$$

$$\texttt{counter}\ P\ \gamma\ p \triangleq \exists q_1 q_2.\ \boxed{\bullet\ \mathrm{Some}\ (p, q_1)}^{\gamma} * P\,q_2 * \ulcorner q_1 + q_2 = 1 \urcorner$$

$$\texttt{no\_tokens}\ P\ \gamma q \triangleq \boxed{\bullet\{q\}\ \mathrm{None}}^{\gamma}$$

$$\vdash \dot{\Rightarrow} \exists \gamma.\, \mathsf{no\_tokens}\ P\ \gamma 1 \qquad P\,1 * \mathsf{no\_tokens}\ P\ \gamma 1 \vdash \dot{\Rightarrow} \mathsf{counter}\ P\ \gamma\ 1 * \mathsf{token}\ P\ \gamma$$

$$\mathsf{counter}\ P\ \gamma\ p \vdash \dot{\Rightarrow} \mathsf{counter}\ P\ \gamma\ (p+1) * \mathsf{token}\ P\ \gamma$$

$$\frac{p > 1}{\mathsf{counter}\ P\ \gamma\ p * \mathsf{token}\ P\ \gamma \vdash \dot{\Rightarrow} \mathsf{counter}\ P\ \gamma\ (p-1)}$$

$$\mathsf{counter}\ P\ \gamma\ 1 * \mathsf{token}\ P\ \gamma \vdash \dot{\Rightarrow} \mathsf{no\_tokens}\ P\ \gamma 1 * P\,1$$

$$\mathsf{token}\ P\ \gamma * \mathsf{no\_tokens}\ P\ \gamma\ q \vdash \mathsf{False} \qquad \mathsf{counter}\ P\ \gamma\ p * \mathsf{no\_tokens}\ P\ \gamma\ q \vdash \mathsf{False}$$

Figure 1: Required rules for token-counting objects.

When increasing the counter, we can always give away half our access to P. The ghost-reasoning when decreasing the counter is more interesting. It crucially relies on the following compatibility rule:

$$\boxed{\bullet\ \mathrm{Some}\ (p, q_1)}^{\gamma} * \boxed{\circ\ \mathrm{Some}\ (1, q_2)}^{\gamma} \vdash \ulcorner p = 1 \land q_1 = q_2 \urcorner \lor \ulcorner p > 1 \land q_1 > q_2 \urcorner$$

This causes a fraction to be left when decreasing the counter from $p > 1$ to $p - 1$, while ensuring all fractions are recovered when decreasing from 1 to noTokens.

2 Notes on proof search strategy

The actual proof search strategy employed by Diaframe is a tad more complicated than presented in the paper. For one thing, the symbolic execution rule is not baked into the strategy, and is actually an *abduction* hint. For another thing, we support more modalities than just ${}^{\mathcal{E}_1}\!\Rightarrow^{\mathcal{E}_2}$ – for example, $\dot{\Rightarrow}$ and id also work fine. The extended grammar on which Diaframe operates can be found in Figure 2. It also includes ε_0, the *first* hypothesis. It can be used to phrase simplification hints.

2.1 Definitions

We define an *abduction hint* as:

$$H * [N] \Vdash [M]\,; \vec{y}; A \quad := H * N \vdash M\,(\exists \vec{y}.\, A\,)$$

As in the paper, terms between square brackets [] are outputs, rest is input.

Abduction hints $H * [N] \Vdash [M]\,; \vec{t}; A$ do not have an acquired extra resource, but allow the new goals to be more complicated: N instead of L. We look for abduction hints when our goal is just a lone atom A, like wp $e\ \{\Phi\}$. For example, we can give the hint:

ABD-HOARE-SPEC

$$\frac{\mathrm{SPEC}\ \vec{x}, \{L\}\ e\ \{\vec{y},\ \mathrm{RET}\ v;\ H_R\}}{\varepsilon_0 * \left[{}^{\top}\!\Rightarrow^{\top} \exists \vec{x}.\, L * \triangleright(\forall \vec{y}.\, H_R \mathrel{-\!\!*} {}^{\top}\!\Rightarrow^{\top} \mathsf{wp}\ K v\ \{w.\, B\})\right] \Vdash [\mathrm{id}]\,; \_; \mathsf{wp}\ K e\ \{w.\, B\}}$$

$$
\begin{array}{rl}
\text{atoms} & A ::= \chi \mid \dots \\
\text{left-goals} & L ::= \ulcorner \phi \urcorner \mid A \mid L * L \mid \exists x.\, L \\
\text{unstructureds} & U ::= \ulcorner \phi \urcorner \mid A \mid U * U \mid \exists x.\, L \mid \forall x.\, U \mid L \mathrel{-\!\!*} U \mid {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} U \mid \triangleright U \\
\text{extended} & H ::= \varepsilon_0 \mid \varepsilon_1 \mid U \\
\text{extended atoms} & A_E ::= \varepsilon_0 \mid \varepsilon_1 \mid A \\
\text{clean hypotheses} & H_C ::= A \mid L \mathrel{-\!\!*} U \mid \forall x.\, U \mid {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} U \mid \triangleright H_C \\
\text{raw hypotheses} & H_R ::= \ulcorner \phi \urcorner \mid H_C \mid H_R * H_R \mid \exists x.\, H_R \\
\text{environments (1)} & \Gamma ::= \emptyset \mid \Gamma, x \mid \Gamma, \phi \qquad \Lambda ::= \emptyset \mid \Lambda, H \\
\text{environments (2)} & \Delta ::= \varepsilon_0, \Lambda, \varepsilon_1 \\
\text{modalities} & M ::= {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \mid \dot{\Rrightarrow} \mid \mathrm{id} \\
\text{introduction} & I ::= \forall x.\, G \mid H_R \mathrel{-\!\!*} G \mid \triangleright I \\
\text{raw/right goals} & R ::= \exists \vec{x}.\, \ulcorner \phi \urcorner \mid \exists \vec{x}.\, A \mid \exists \vec{x}.\, L * R \mid \exists \vec{x}.\, L * I \quad \text{if } \mathrm{FV}(L) = \vec{x} \\
\text{basic/both goals} & B ::= I \mid R \\
\text{natural goals} & N ::= B \mid M\, B \\
\text{goals} & G ::= N \mid M_1\, M_2\, B \mid \|M\|\, \exists \vec{x}.\, L * G \quad \text{if } \mathrm{FV}(L) = \vec{x}
\end{array}
$$

Figure 2: Extended grammar.

This abduction hint will not be applied on the goal $\Delta \vdash \mathsf{wp}\ K[e]\ \{\Phi\} * G_1$, since the new goal $N * G_1$ is not always in G.

For all modalities M, we require that they are *strong functors*, meaning:

MOD-MONO	MOD-STRONG
$\forall P.\ P \vdash M\,P$	$\forall\, P\, Q.\ P * MQ \vdash M(P * Q)$

We define $(\mathsf{split}(M_1) = [M_2, M_3]) \triangleq \forall P.\ M_2\, M_3\, P \vdash M_1\, P$ with M_2 and M_3 output. We also define the operation $(M_1 \ominus M_3 = M_2) \triangleq \forall P.\ M_2\, M_3\, P \vdash M_1\, P$. In this case, only M_2 is output. Let us see some examples:

$$\mathsf{split}({}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2}) = [{}^{\mathcal{E}_1}\!\Rrightarrow^{?\mathcal{E}_3}, {}^{?\mathcal{E}_3}\!\Rrightarrow^{\mathcal{E}_2}]$$
$$\mathsf{split}(\dot{\Rrightarrow}) = [\dot{\Rrightarrow}, \dot{\Rrightarrow}]$$
$$\mathsf{split}(\mathrm{id}) = [\mathrm{id}, \mathrm{id}]$$

$$ {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \ominus {}^{\mathcal{E}_3}\!\Rrightarrow^{\mathcal{E}_2} = {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_3}$$
$$ {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \ominus \dot{\Rrightarrow} = {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2}$$
$$ {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2} \ominus \mathrm{id} = {}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_2}$$
$$\dot{\Rrightarrow} \ominus \dot{\Rrightarrow} = \dot{\Rrightarrow}$$
$$\dot{\Rrightarrow} \ominus \mathrm{id} = \dot{\Rrightarrow}$$
$$\mathrm{id} \ominus \mathrm{id} = \mathrm{id}$$

2.2 Proof search for $\Gamma; \Delta \vdash G$

To prove $\Gamma; \Delta \vdash G$, we perform a case-distinction on G:

1. $G = R$. Continue with the equivalent goal $\Gamma; \Delta \vdash \mathrm{id}\, R$
2. $G = M_1\, B$ or $G = M_1\, M_2\, B$ and M_1 is *not* introducable. To continue we need to perform a view-shift/close an invariant. Continue with the equivalent goal

   $$\Gamma; \Delta \vdash \|{}^{\mathcal{E}_1}\!\Rrightarrow^{?\mathcal{E}_3}\| \exists\_.\ \chi * {}^{?\mathcal{E}_3}\!\Rrightarrow^{\mathcal{E}_2} N$$

   where $N = B$ or $N = M_2\, B$ respectively. Note that all cases below this can assume that M_1 is introducable.
3. $G = M_1\, M_2\, B$ and M_1 is introducable. Continue with goal $\Gamma; \Delta \vdash M'\, B$, where $M' = M_1 \circ M_2$ is obtained from the following table.

$\circ$	id	$\dot{\Rrightarrow}$	${}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_3}$
id	id	$\dot{\Rrightarrow}$	${}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_3}$
$\dot{\Rrightarrow}$	$\dot{\Rrightarrow}$	$\dot{\Rrightarrow}$	${}^{\mathcal{E}_2}\!\Rrightarrow^{\mathcal{E}_3}$
${}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_1}$	${}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_1}$	${}^{\mathcal{E}_1}\!\Rrightarrow^{\mathcal{E}_1}$	${}^{\mathcal{E}_2}\!\Rrightarrow^{E_3}$

4. $G = M\, I$. Continue with new goal $\Gamma; \Delta \vdash I$ (by introducing M)

5. $G = \forall x.\, G'$. Introduce x, continue with new goal $\Gamma, x; \Delta \vdash G'$.
6. $G = H_R \mathrel{-\!\!*} G'$. We perform a case-distinction on H_R:
   (a) $H_R = \ulcorner \phi \urcorner$. Introduce the pure proposition ϕ, continue with $\Gamma, \phi; \Delta \vdash G'$
   (b) $H_R = A$. Two cases:
      i. There is some $A_i \in \Delta$ which can be *merged* with A, *i.e.,* an H'_R for which $A * A_i \vdash H_R$. Continue with introducing H'_R: i.e. $\Gamma; \Delta \setminus A_i \vdash H'_R \mathrel{-\!\!*} G'$.
      ii. There is some $A_i \in \Delta$ which has a *compatibility* with A, *i.e.,* an ϕ for which $A * A_i \vdash \ulcorner \phi \urcorner$. Continue with introducing ϕ: i.e. $\Gamma; \Delta, A \vdash \ulcorner \phi \urcorner \mathrel{-\!\!*} G'$.
      iii. No such atoms are in Δ. Introduce the spatial resource A, continue with $\Gamma; \Delta, A \vdash G'$. Note that $A \in H_C$.
   (c) $H_R = H'_R * H''_R$. Continue with equivalent goal $\Gamma; \Delta \vdash H''_R \mathrel{-\!\!*} H'_R \mathrel{-\!\!*} G$
   (d) $H_R = \exists x.\, H'_R$. Continue with equivalent goal $\Gamma; \Delta \vdash \forall x.\, (H'_R \mathrel{-\!\!*} G)$
   (e) $H_R = H_C$. Add the cleaned hypothesis to the spatial context, continue with $\Gamma; \Delta, H_C \vdash G$.
7. $G = \triangleright I$. We want to introduce the later, but first strip off remaining laters in hypotheses:
   (a) If $\Delta = \Delta', \triangleright H_C$ for some H_C, continue with goal $\Gamma; \Delta' \vdash \triangleright(H_C \mathrel{-\!\!*} I)$. We put the hypotheses back in the goal to force searching for possibile merges.
   (b) If not, continue with goal $\Gamma; \Delta \vdash I$
8. $G = M\, R$. Case-distinction on R:
   (a) $R = \exists \vec{x}.\, \ulcorner \phi \urcorner$. Solve $\exists \vec{x}.\, \phi$ using pure context Γ to finish the proof.
   (b) $R = \exists \vec{x}.\, A$. Find $H \in \Delta$ for which both $H * [N] \Vdash [M_2]\,; \vec{x}; (A\, \vec{x})$ and $M \ominus M_2 = M_1$. Continue with $\Gamma; \Delta \setminus H \vdash M_1\, N$. Note that $M_1\, N \in G$.
   (c) $R = \exists \vec{x}.\, L * B$. Set $\vec{t} = \mathrm{FV}(L)$ and $\vec{s} = \vec{x} \setminus \vec{t}$. Find M_1 and M_2 such that $\mathrm{split}(M) = [M_1, M_2]$, then continue with $\Gamma; \Delta \vdash \|M_1\| \exists \vec{t}.\, L * (M_2\, \exists \vec{s}.\, B)$,
   Note that if $\vec{s}$ is non-empty, B must have been some R' (by the definition of R), so that $M_2\, \exists \vec{s}.\, R' \in G$. If instead we had $B = I$, then by the definition of R we have that $\vec{s}$ is empty, so that $M_2\, \exists \vec{s}.\, I = M_2 I \in G$.
9. $G = \|M\|\, \exists \vec{x}.\, L * G'$. Case-distinction on L:
   (a) $L = \ulcorner \phi \urcorner$. Solve $\exists \vec{x}.\, \phi$ using pure context Γ. Continue with the found existential x' in goal $\Gamma; \Delta \vdash G'$. We can introduce M since this case was created by 8c or one of the following cases, and these keep invariant that the modality between $\|$s is introducable. This relies on split always splitting introducable modalities into introducable modalities.
   (b) $L = A$. Find $H \in \Delta$ with $H * \left[\vec{t}_n; L\, \vec{t}_n\right] \Vdash [M_2]\,; \vec{x}; (A\, \vec{x}) * \left[H_R\, \vec{t}_n\, \vec{x}\right]$ and $M \ominus M_2 = M_1$. Continue with goal:
   $$\Gamma; \Delta \setminus H \vdash \|M_1\|\, \exists \vec{t}_n.\, L\, \vec{t}_n * \left(\forall \vec{t}.\, H_R\, \vec{t}_n\, \vec{t} \mathrel{-\!\!*} G'\, \vec{t}\right)$$
   (c) $L = L_1 * L_2$. Set $\vec{t} = \mathrm{FV}(L)$ and $\vec{s} = \vec{x} \setminus \vec{t}$. Find M_1 and M_2 such that $\mathrm{split}(M) = [M_1, M_2]$, then continue with $\Gamma; \Delta \vdash \|M_1\|\, \exists \vec{t}.\, L_1 * (\|M_2\|\, \exists \vec{s}.\, L_2 * G')$.
   (d) $L = \exists t.\, L'$. Continue with $\Gamma; \Delta \vdash \|M\|\, \exists (\vec{x}, t).\, L' * G'$

BIABD-SEP-L

$$\frac{U_1 * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [H_R]}{(U_1 * U_2) * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [U_2 * H_R]}$$

BIABD-WAND

$$\frac{U * [\vec{z}; L_2] \Vdash [M]\,; \vec{y}; A * [H_R]}{(L_1 \mathrel{-\!\!*} U) * [\vec{z}; L_2 * L_1] \Vdash [M]\,; \vec{y}; A * [H_R]}$$

BIABD-MOD-INTRO-L

$$\frac{U * [\vec{z}; L] \Vdash [M_2]\,; \vec{y}; A * [H_R]}{(M_1\, U) * [\vec{z}; L] \Vdash [M_1 \circ M_2]\,; \vec{y}; A * [H_R]}$$

BIABD-EXIST

$$\frac{\forall x.\ L * [\vec{z}; \ulcorner \phi \urcorner] \Vdash [M]\,; \vec{y}; A * [H_R]}{(\exists x.\, L) * [\_; \ulcorner \forall x.\, \exists \vec{z}.\, \phi \urcorner] \Vdash [M]\,; \vec{y}; A * [\exists x\, \vec{z}.\, H_R]}$$

BIABD-FORALL-POSTPONE

$$\frac{\forall t.\ U * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [H_R]}{(\forall t.\, U) * [(t, \vec{z}); L] \Vdash [M]\,; \vec{y}; A * [H_R]}$$

BIABD-FORALL-ONE

$$\frac{U[w/t] * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [H_R]}{(\forall t.\, U) * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [H_R]}$$

BIABD-WITNESS-POSTPONE

$$\frac{\forall t.\ A_E * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A * [H_R]}{A_E * [(t', \vec{z}); L[t'/t]] \Vdash [M]\,; (t, \vec{t}_2)\,; A * [\ulcorner t' = t \urcorner * H_R]}$$

BIABD-WITNESS

$$\frac{A_E * [\vec{z}; L] \Vdash [M]\,; \vec{y}; A[t'/t] * [H_R]}{A_E * [\vec{z}; L] \Vdash [M]\,; (t, \vec{y}); A * [\ulcorner t = t' \urcorner * H_R]}$$

Figure 3: Selection of rules for recursively finding hints.

3 Notes on hint search strategy

The algorithm described above can only make progress if suitable hints can be found. In this section we describe how we find such hints. Some of the general recursive rules can be found in Figure 3.

Backtracking on all possible orders of these hints does give rise to a hint search strategy, but this is not efficient enough. Instead, we try to first use all rules related to hypotheses (like BIABD-WAND), and only if this is no longer possible, use goal related rules like BIABD-WITNESS.

We thus divide hint search into two two phases: left phase, where left rules are applied eagerly, and right phase, where right rules are applied until a match with a user-provided hint is found. This algorithm *does* backtrack to select a hint, but the algorithm in section 2.2 does not backtrack on the found hint – it sticks with the first candidate.

3.1 Proof search for $H * \left[\vec{t}; L\right] \Vdash [M]\,; \vec{x}; A * [H_R]$

To support the search in different phases, we introduce three new versions of the hint:

$$H * \left[\vec{t}; L\right] \Vdash_l [M]\,; \vec{x}; A * [H_R]$$
$$A_E * \left[\vec{t}; L\right] \Vdash_r [M]\,; \vec{x}; A * [H_R]$$
$$A_E * \left[\vec{t}; L\right] \Vdash_b [M]\,; \vec{x}; A * [H_R]\,,$$

all of which have the same definition, but a different marker m after the $\Vdash_m$, and different allowed grammar for the first argument. The search is started by applying the following rule:

$$\textsc{biabd-from-left}\ \frac{H * \left[\vec{t}; L\right] \Vdash_l [M]\,; \vec{x}; A * [H_R]}{H * \left[\vec{t}; L\right] \Vdash [M]\,; \vec{x}; A * [H_R]}$$

Left rules When we are looking for a hint of the form $H * \left[\vec{t}; L\right] \Vdash_l [M]\,; \vec{x}; A * [H_R]$, we apply the following rules eagerly:

$$\textsc{biabd-later-mono}\ \frac{U * \left[\vec{t}_1; L\right] \Vdash_l [\mathrm{id}]\,; \vec{t}_2; A * [H_R]}{(\triangleright^n U) * \left[\vec{t}_1; \triangleright^n L\right] \Vdash_l [\mathrm{id}]\,; \vec{t}_2; (\triangleright^n A) * [\triangleright^n H_R]}$$

$$\textsc{biabd-sep-l}\ \frac{U_1 * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(U_1 * U_2) * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; (A) * [U_2 * H_R]}$$

$$\textsc{biabd-sep-r}\ \frac{U_2 * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(U_1 * U_2) * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [U_1 * H_R]}$$

$$\textsc{biabd-exist}\ \frac{\forall \vec{t}_p.\ L_1 * \left[\vec{t}_1; \ulcorner \phi \urcorner\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(\exists \vec{t}_p.\, L_1) * \left[\_; \ulcorner \forall \vec{t}_p.\, \exists \vec{t}_1.\, \phi \urcorner\right] \Vdash_l [M]\,; \vec{t}_2; A * \left[\exists \vec{t}_p\, \vec{t}_1.\, H_R\right]}$$

$$\textsc{biabd-forall-postpone}\ \frac{\forall t.\ U * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(\forall t.\, U\ t) * \left[(t, \vec{t}_1); L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]} \qquad \textsc{biabd-forall-one}\ \frac{U * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(\forall t.\, U\ t) * \left[\vec{t}_1; L\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}$$

$$\textsc{biabd-mod-intro-l}\ \frac{U * \left[\vec{t}_1; L\right] \Vdash_l [M_2]\,; \vec{t}_2; A * [H_R]}{(M_1\, U) * \left[\vec{t}_1; L\right] \Vdash_l [M_1 \circ M_2]\,; \vec{t}_2; A * [H_R]}$$

$$\textsc{biabd-wand}\ \frac{U * \left[\vec{t}_1; L_2\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}{(L_1 \mathrel{-\!\!*} U) * \left[\vec{t}_1; L_2 * L_1\right] \Vdash_l [M]\,; \vec{t}_2; A * [H_R]}$$

In the implementation we can merge forall-postpone and forall-one into one rule, but 'morally' they are for different cases. Forall-postpone is applicable when the argument of the for all does not appear in the goal, while forall-one applies when the argument of the forall *does* appear in the goal – so needs to be a specific value.

When none of the rules above applies, we know the first argument is an extended atom A_E, a $\triangleright U$ when the goal is not $\triangleright$, or a pure proposition $\ulcorner \phi \urcorner$. We then use the following rule:

BIABD-LEFT-FROM-RIGHT
$$\frac{A_E * \left[\vec{t}; L\right] \Vdash_r [M]\,; \vec{x}; A * [H_R]}{A_E * \left[\vec{t}; L\right] \Vdash_l [M]\,; \vec{x}; A * [H_R]}$$

Right rules When looking for a hint of the form $A_E * \left[\vec{t}; L\right] \Vdash_r [M]\,; \vec{x}; A * [H_R]$, the following rules are tried in their given order:

BIABD-FROM-BASE
$$\frac{A_E * \left[\vec{t}_1; L\right] \Vdash_b [M]\,; \vec{t}_2; A * [H_R]}{A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; \vec{t}_2; A * [H_R]}$$

BIABD-LATER-INTRO-R
$$\frac{A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; \vec{t}_2; A * [H_R]}{A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; \vec{t}_2; (\triangleright^n A) * [H_R]}$$

BIABD-WITNESS-POSTPONE
$$\frac{\forall t.\ A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; \vec{t}_2; A * [H_R]}{A_E * \left[(t', \vec{t}_1); L\right] \Vdash_r [M]\,; \left(t, \vec{t}_2\right); A * [\ulcorner t' = t \urcorner * H_R]}$$

BIABD-WITNESS
$$\frac{A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; \vec{t}_2; A * [H_R]}{A_E * \left[\vec{t}_1; L\right] \Vdash_r [M]\,; (t', \vec{t}_2); A * \left[\ulcorner t' = t \urcorner * H_R\, \vec{t}_1\, \vec{t}_2\right]}$$

3.2 Proof search for $H_C * [N] \Vdash [M]\,; \vec{x}; A\,\vec{x}$

Since the proof search for these hints is largely similar to that of biabduction, we only mention the differences.

As for bi-abduction, we define three new versions of these hints:

$$H * [N] \Vdash_l [M]\,; \vec{x}; A$$
$$A_E * [N] \Vdash_r [M]\,; \vec{x}; A$$
$$A_E * [N] \Vdash_b [M]\,; \vec{x}; A$$

The following left rules deserve some remarks:

ABD-EXIST
$$\frac{\forall \vec{t}_p.\ L * [N] \Vdash_l [M]\,; \vec{t}_2; A}{\left(\exists \vec{t}_p.\, L\right) * \left[\forall \vec{t}_p.\, N\right] \Vdash_l [M]\,; \vec{t}_2; A}$$

ABD-FORALL-POSTPONE
$$\frac{\forall t.\ U * [R] \Vdash_l [M]\,; \vec{t}_2; A}{(\forall t.\, U) * [\exists t.\, R] \Vdash_l [M]\,; \vec{t}_2; A}$$

ABD-WAND
$$\frac{U * [N] \Vdash_l [M]\,; \vec{t}_2; A}{(L \mathrel{-\!\!*} U) * [L * N] \Vdash_l [M]\,; \vec{t}_2; A}$$

ABD-SEP-L
$$\frac{U_1 * [N] \Vdash_l [M]\,; \vec{t}_2; A}{(U_1 * U_2) * [U_2 \mathrel{-\!\!*} N] \Vdash_l [M]\,; \vec{t}_2; A}$$

ABD-WITNESS-POSTPONE

$$\frac{\forall t.\ A_E * [R] \Vdash_r [M]\,;\vec{t}_2;A}{A_E * [\exists t.\, R] \Vdash_r [M]\,;\left(t,\vec{t}_2\right);A}$$

The found new goal of rules ABD-FORALL-POSTPONE and ABD-WITNESS-POSTPONE has shape $\exists t.\, Rt$. Note that if we had allowed N, the new goal would have shape $\exists t.\, Nt$ which is not necessarily in N. This problem does not occur for bi-abduction, since the found new goal must be in L, and $\exists x.\, Lx \in L$.

The found new goal of ABD-EXIST does not require changing the grammar of the found new goal, since $\forall t.\, Nt \in N$.

For the ABD-WAND rule, we prioritize solving the left-hand side of the wand. This is because $N * L \notin N$.

Note that for ABD-SEP-L we directly get access to the unused part of the separating conjunction.

3.3 Recursing inside atoms

Attentive readers may have noticed that we did not describe a recursive rule for invariants. This is because there is none. However, the recursive rules as stated in the previous sections are not directly applicable. How do we overcome this?

The crux is that some atoms, like for example invariants $\boxed{L}^{\mathcal{N}}$, may be interpreted both as an atom, and as a more complicated connective in the grammar. This is witnessed by the following rules:

INV-OPEN-FUPD

$$\overline{\boxed{L}^{\mathcal{N}} \vdash \ulcorner \mathcal{N} \subseteq \mathcal{E} \urcorner \mathrel{-\!\!*} {}^{\mathcal{E}}\!\Rrightarrow^{\mathcal{E}\setminus\mathcal{N}} \left(\triangleright L * \left(\triangleright L \mathrel{-\!\!*} {}^{\mathcal{E}\setminus\mathcal{N}}\!\Rrightarrow^{\mathcal{E}} \mathsf{True}\right)\right)}$$

TOKEN-ACCESS

$$\overline{\mathsf{token}\ P\ \gamma P \vdash \exists q.\, P\,q * (P\,q \mathrel{-\!\!*} \mathsf{token}\ P\ \gamma P)}$$

INV-OPEN-FUPD states that the hypothesis $\boxed{L}^{\mathcal{N}}$ should be seen by the recursive hint search not only as an atom, but also as a wand, and thus as a way to access L.

To accomodate this, the recursive hint search procedure does not just do a direct syntactic match on the hypothesis, but also looks for registered ways in which the focused hypothesis can be seen as a connective. This is not just applicable for invariants: for example, TOKEN-ACCESS also makes use of this.

3.4 Base Diaframe hints

The following two hints are always present, and are not Iris-specific:

BIABD-ASSUMPTION

$$A * [\_;\mathsf{True}] \Vdash [\mathsf{id}]\,;\_;A * [\mathsf{True}]$$

ABD-FROM-BIABD

$$\frac{A_1 * \left[\vec{t}_1; L\right] \Vdash [M]\,;\vec{t}_2;A_2 * [H_R]}{A_1 * \left[\exists \vec{t}_1.\, L\right] \Vdash [M]\,;\vec{t}_2;A_2}$$

If only BIABD-ASSUMPTION is present, the strategy only looks for syntactic matches. The ABD-FROM-BIABD rule lifts bi-abduction hints to abduction hints, but is only provable in affine separation logics.