TopHat: A formal foundation for task-oriented programming

Appendices

Tim Steenvoorden
Software Science
Radboud University
Nijmegen, The Netherlands
tim@cs.ru.nl

Nico Naus
Information and Computing Sciences
Utrecht University
Utrecht, The Netherlands
n.naus@uu.nl

Markus Klinik
Software Science
Radboud University
Nijmegen, The Netherlands
m.klinik@cs.ru.nl

ACM Reference Format:

Tim Steenvoorden, Nico Naus, and Markus Klinik. 2019. TopHat: A formal foundation for task-oriented programming: Appendices. In *Principles and Practice of Programming Languages 2019 (PPDP '19), October 7–9, 2019, Porto, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3354166.3354182

PPDP '19, October 7–9, 2019, Porto, Portugal

ACM ISBN 978-1-4503-7249-7/19/10…$15.00
https://doi.org/10.1145/3354166.3354182

A ADDITIONAL RULES

A.1 Evaluation rules

$$\boxed{e, \sigma \downarrow v, \sigma'}$$

$$\textsc{E-App}\ \frac{e_1, \sigma \downarrow \lambda x : \tau . e_1', \sigma' \quad e_2, \sigma' \downarrow v_2, \sigma'' \quad e_1'[x \mapsto v_2], \sigma'' \downarrow v_1, \sigma'''}{e_1 e_2, \sigma \downarrow v_1, \sigma'''}$$

$$\textsc{E-IfTrue}\ \frac{e_1, \sigma \downarrow \mathsf{True}, \sigma' \quad e_2, \sigma' \downarrow v, \sigma''}{\textbf{if}\ e_1\ \textbf{then}\ e_2\ \textbf{else}\ e_3, \sigma \downarrow v, \sigma''}$$

$$\textsc{E-Ref}\ \frac{e, \sigma \downarrow v, \sigma' \quad l \notin Dom(\sigma')}{\textbf{ref}\ e, \sigma \downarrow l, \sigma'[l \mapsto v]}$$

$$\textsc{E-IfFalse}\ \frac{e_1, \sigma \downarrow \mathsf{False}, \sigma' \quad e_3, \sigma' \downarrow v, \sigma''}{\textbf{if}\ e_1\ \textbf{then}\ e_2\ \textbf{else}\ e_3, \sigma \downarrow v, \sigma''}$$

$$\textsc{E-Deref}\ \frac{e, \sigma \downarrow l, \sigma'}{!e, \sigma \downarrow \sigma'(l), \sigma'}$$

$$\textsc{E-Value}\ \frac{}{v, \sigma \downarrow v, \sigma}$$

$$\textsc{E-Assign}\ \frac{e_1, \sigma \downarrow l, \sigma' \quad e_2, \sigma' \downarrow v_2, \sigma''}{e_1 := e_2, \sigma \downarrow \langle\rangle, \sigma''[l \mapsto v_2]}$$

$$\textsc{E-Pair}\ \frac{e_1, \sigma \downarrow v_1, \sigma' \quad e_2, \sigma' \downarrow v_2, \sigma''}{\langle e_1, e_2 \rangle, \sigma \downarrow \langle v_1, v_2 \rangle, \sigma''}$$

$$\textsc{E-Edit}\ \frac{e, \sigma \downarrow v, \sigma'}{\square\, e, \sigma \downarrow \square\, v, \sigma'}$$

$$\textsc{E-Enter}\ \frac{}{\boxtimes\, \tau, \sigma \downarrow \boxtimes\, \tau, \sigma}$$

$$\textsc{E-Update}\ \frac{e, \sigma \downarrow l, \sigma'}{\blacksquare\, e, \sigma \downarrow \blacksquare\, l, \sigma'}$$

$$\textsc{E-Then}\ \frac{e_1, \sigma \downarrow t_1, \sigma'}{e_1 \blacktriangleright e_2, \sigma \downarrow t_1 \blacktriangleright e_2, \sigma'}$$

$$\textsc{E-Next}\ \frac{e_1, \sigma \downarrow t_1, \sigma'}{e_1 \triangleright e_2, \sigma \downarrow t_1 \triangleright e_2, \sigma'}$$

$$\textsc{E-And}\ \frac{e_1, \sigma \downarrow t_1, \sigma' \quad e_2, \sigma' \downarrow t_2, \sigma''}{e_1 \bowtie e_2, \sigma \downarrow t_1 \bowtie t_2, \sigma''}$$

$$\textsc{E-Fail}\ \frac{}{\lightning, \sigma \downarrow \lightning, \sigma}$$

$$\textsc{E-Or}\ \frac{e_1, \sigma \downarrow t_1, \sigma' \quad e_2, \sigma' \downarrow t_2, \sigma''}{e_1 \blacklozenge e_2, \sigma \downarrow t_1 \blacklozenge t_2, \sigma''}$$

$$\textsc{E-Xor}\ \frac{}{e_1 \lozenge e_2, \sigma \downarrow e_1 \lozenge e_2, \sigma}$$

A.2 Typing rules

$$\boxed{\Gamma, \Sigma \vdash e : \tau}$$

$$\textsc{T-Var}\ \frac{x : \tau \in \Gamma}{\Gamma, \Sigma \vdash x : \tau}$$

$$\textsc{T-Loc}\ \frac{\Sigma(l) = \beta}{\Gamma, \Sigma \vdash l : \textsc{Ref}\ \beta}$$

$$\textsc{T-Pair}\ \frac{\Gamma, \Sigma \vdash e_1 : \tau_1 \quad \Gamma, \Sigma \vdash e_2 : \tau_2}{\Gamma, \Sigma \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2}$$

$$\textsc{T-Abs}\ \frac{\Gamma[x : \tau_1], \Sigma \vdash e : \tau_2}{\Gamma, \Sigma \vdash \lambda x : \tau_1 . e : \tau_1 \to \tau_2}$$

$$\textsc{T-App}\ \frac{\Gamma, \Sigma \vdash e_1 : \tau_1 \to \tau_2 \quad \Gamma, \Sigma \vdash e_2 : \tau_1}{\Gamma, \Sigma \vdash e_1 e_2 : \tau_2}$$

$$\textsc{T-If}\ \frac{\Gamma, \Sigma \vdash e_1 : \textsc{Bool} \quad \Gamma, \Sigma \vdash e_2 : \tau \quad \Gamma, \Sigma \vdash e_3 : \tau}{\Gamma, \Sigma \vdash \textbf{if}\ e_1\ \textbf{then}\ e_2\ \textbf{else}\ e_3 : \tau}$$

$$\textsc{T-Ref}\ \frac{\Gamma, \Sigma \vdash e : \beta}{\Gamma, \Sigma \vdash \textbf{ref}\ e : \textsc{Ref}\ \beta}$$

$$\textsc{T-Deref}\ \frac{\Gamma, \Sigma \vdash e : \textsc{Ref}\ \beta}{\Gamma, \Sigma \vdash\ !e : \beta}$$

$$\textsc{T-Assign}\ \frac{\Gamma, \Sigma \vdash e_1 : \textsc{Ref}\ \beta \quad \Gamma, \Sigma \vdash e_2 : \beta}{\Gamma, \Sigma \vdash e_1 := e_2 : \textsc{Unit}}$$

B PROOFS

B.1 ??

Proof. We prove **??** by induction on e:

Case $e = \lambda x : \tau . e, e_1 e_2, x, c, l, e_1 \star e_2, \textbf{if}\, e_1\, \textbf{then}\, e_2\, \textbf{else}\, e_3, \langle e_1, e_2 \rangle, \langle\rangle, \textbf{ref}\, e, !e, e_1 := e_2$

Preservation has been proven for these cases by Pierce [**?**].

Case $\dfrac{e, \sigma \downarrow v, \sigma'}{\square e, \sigma \downarrow \square v, \sigma'}$ E-Edit

Given that $\Gamma, \Sigma \vdash \square e : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s$,T-Edit gives us that $\Gamma, \Sigma \vdash e : \tau$. The induction hypothesis gives us that $e, s \downarrow v, s'$ also preserves, and thus $\Gamma, \Sigma \vdash v : \tau$ and $\Gamma, \Sigma \vdash s'$. Therefore $\Gamma, \Sigma \vdash \square v : \text{Task}\, \tau$.

Case $\dfrac{}{\boxtimes \tau, \sigma \downarrow \boxtimes \tau, \sigma}$ E-Enter

Evaluation does not alter e and s, therefore this case holds trivially.

Case $\dfrac{e, \sigma \downarrow l, \sigma'}{\blacksquare e, \sigma \downarrow \blacksquare l, \sigma'}$ E-Update

Given that $\Gamma, \Sigma \vdash \square e : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s$, T-Update gives us that $\Gamma, \Sigma \vdash e : \textbf{ref}\, \tau$. The induction hypothesis gives us that $e, s \downarrow l, s'$ also preserves, and thus $\Gamma, \Sigma \vdash l : \textbf{ref}\, \tau$ and $\Gamma, \Sigma \vdash s'$. Therefore $\Gamma, \Sigma \vdash \blacksquare l : \text{Task}\, \tau$.

Case $\dfrac{}{\lightning, \sigma \downarrow \lightning, \sigma}$ E-Fail

Evaluation does not alter e and s, therefore this case holds trivially.

Case $\dfrac{e_1, \sigma \downarrow t_1, \sigma'}{e_1 \blacktriangleright e_2, \sigma \downarrow t_1 \blacktriangleright e_2, \sigma'}$ E-Then

Given that $\Gamma, \Sigma \vdash e_1 \blacktriangleright e_2 : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s$, T-Then gives us that $\Gamma, \Sigma \vdash e_1 : \text{Task}\, \tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \text{Task}\, \tau$. By the induction hypothesis, we know that $e_1, s \downarrow t_1, s'$ preserves and thus $\Gamma, \Sigma \vdash t_1 : \text{Task}\, \tau_1$ and $\Gamma, \Sigma \vdash s'$. Therefore $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \text{Task}\, \tau$.

Case $\dfrac{e_1, \sigma \downarrow t_1, \sigma'}{e_1 \triangleright e_2, \sigma \downarrow t_1 \triangleright e_2, \sigma'}$ E-Next

Given that $\Gamma, \Sigma \vdash e_1 \triangleright e_2 : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s$, T-Next gives us that $\Gamma, \Sigma \vdash e_1 : \text{Task}\, \tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \text{Task}\, \tau$. By the induction hypothesis, we know that $e_1, s \downarrow t_1, s'$ preserves and thus $\Gamma, \Sigma \vdash t_1 : \text{Task}\, \tau_1$ and $\Gamma, \Sigma \vdash s'$. Therefore $\Gamma, \Sigma \vdash t_1 \triangleright e_2 : \text{Task}\, \tau$.

Case $\dfrac{e_1, \sigma \downarrow t_1, \sigma' \quad e_2, \sigma' \downarrow t_2, \sigma''}{e_1 \bowtie e_2, \sigma \downarrow t_1 \bowtie t_2, \sigma''}$ E-And

Given that $\Gamma, \Sigma \vdash e_1 \bowtie e_2 : \text{Task}(\tau_1 \times \tau_2)$ and $\Gamma, \Sigma \vdash s$, T-And gives us that $\Gamma, \Sigma \vdash e_1 : \text{Task}\, \tau_1$ and $\Gamma, \Sigma \vdash e_2 : \text{Task}\, \tau_2$. By the induction hypothesis, we know that both $e_1, s \downarrow t_1, s'$ and $e_2, s' \downarrow t_2, s''$ preserve and thus $\Gamma, \Sigma \vdash t_1 : \text{Task}\, \tau_1, \Gamma, \Sigma \vdash s', \Gamma, \Sigma \vdash t_2 : \text{Task}\, \tau_2$ and $\Gamma, \Sigma \vdash s''$. Therefore $\Gamma, \Sigma \vdash t_1 \bowtie t_2 : \text{Task}(\tau_1 \times \tau_2)$.

Case $\dfrac{e_1, \sigma \downarrow t_1, \sigma' \quad e_2, \sigma' \downarrow t_2, \sigma''}{e_1 \blacklozenge e_2, \sigma \downarrow t_1 \blacklozenge t_2, \sigma''}$ E-Or

Given that $\Gamma, \Sigma \vdash e_1 \blacklozenge e_2 : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s$, T-Or gives us that $\Gamma, \Sigma \vdash e_1 : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash e_2 : \text{Task}\, \tau$. By the induction hypothesis, we have that both $e_1, s \downarrow t_1, s'$ and $e_2, s' \downarrow t_2, s''$ preserve and thus $\Gamma, \Sigma \vdash t_1 : \text{Task}\, \tau, \Gamma, \Sigma \vdash s', \Gamma, \Sigma \vdash t_2 : \text{Task}\, \tau$ and $\Gamma, \Sigma \vdash s''$. Therefore $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \text{Task}\, \tau$.

Case $\dfrac{}{e_1 \lozenge e_2, \sigma \downarrow e_1 \lozenge e_2, \sigma}$ E-Xor

Evaluation does not alter e and s, therefore this case holds trivially.

□

B.2 Lemma B.1

Lemma B.1 (Task value preserves types). *For all expressions e and states σ such that $\Gamma, \Sigma \vdash e : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash \sigma$, if $\mathcal{V}(e, \sigma) = v$, then $v : \tau$.*

Proof. We prove Lemma B.1 by induction over e.

Case $\mathcal{V}(\square\, v, s) = v$

By T-Edit, if $\Gamma, \Sigma \vdash \square\, v : \textsc{Task}\,\tau$, then $\Gamma, \Sigma \vdash v : \tau$.

Case $\mathcal{V}(\boxtimes\, \tau, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(\blacksquare\, l, s) = s(l)$

Given that $\Gamma, \Sigma \vdash \blacksquare\, l : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, we know that $\Gamma, \Sigma \vdash s(l) : \tau$ by definiton.

Case $\mathcal{V}(\lightning, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(t_1 \blacktriangleright e_2, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(t_2 \triangleright e_2, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(t_1 \bowtie t_2, s) = \langle v_1, v_2 \rangle$ given that $\mathcal{V}(t_1, s) = v_1 \land \mathcal{V}(t_2, s) = v_2$

By T-And we have that $\Gamma, \Sigma \vdash t_1 \bowtie t_2 : \textsc{Task}(\tau_1 \times \tau_2)$ and $\Gamma, \Sigma \vdash t_1 : \tau_1$ and $\Gamma, \Sigma \vdash t_2 : \tau_2$. By the induction hypothesis, $\mathcal{V}(t_1, s) = v_1$ and $\mathcal{V}(t_2, s) = v_2$ preserve, and thus $\Gamma, \Sigma \vdash v_1 : \tau_1$ and $\Gamma, \Sigma \vdash v_2 : \tau_2$. This gives us that $\Gamma, \Sigma \vdash \langle v_1, v_2 \rangle : \textsc{Task}(\tau_1 \times \tau_2)$.

Case $\mathcal{V}(t_1 \bowtie t_2, s) = \bot$ given that $\neg(\mathcal{V}(t_1, s) = v_1 \land \mathcal{V}(t_2, s) = v_2)$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(t_1 \blacklozenge t_2, s) = v_1$ given that $\mathcal{V}(t_1, s) = v_1$

By T-Or we have that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$, and $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis, we have that $\Gamma, \Sigma \vdash v_1 : \tau$.

Case $\mathcal{V}(t_1 \blacklozenge t_2, s) = v_2$ given that $\mathcal{V}(t_1, s) = \bot \land \mathcal{V}(t_2, s) = v_2$

By T-Or we have that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$, and $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis, we have that $\Gamma, \Sigma \vdash v_2 : \tau$.

Case $\mathcal{V}(t_1 \blacklozenge t_2, s) = \bot$ given that $\mathcal{V}(t_1, s) = \bot \land \mathcal{V}(t_2, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

Case $\mathcal{V}(t_1 \diamond t_2, s) = \bot$

Since this case does not lead to a value, the lemma holds trivially.

$\square$

B.3 Lemma B.2

Lemma B.2 (Striding preserves types). *For all expressions e and states σ such that $\Gamma, \Sigma \vdash e : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash \sigma$, if $e, \sigma \rightsquigarrow e', \sigma'$, then $\Gamma, \Sigma \vdash e' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash \sigma'$.*

Proof. We prove Lemma B.2 by induction on e:

Case S-Fail
$$\frac{}{\lightning, \sigma \rightsquigarrow \lightning, \sigma}$$
Since this case does not alter the expression, the theorem holds trivially.

Case S-Xor
$$\frac{}{e_1 \Diamond e_2, \sigma \rightsquigarrow e_1 \Diamond e_2, \sigma}$$
Since this case does not alter the expression, the theorem holds trivially.

Case S-Update
$$\frac{}{\blacksquare\, l, \sigma \rightsquigarrow \blacksquare\, l, \sigma}$$
Since this case does not alter the expression, the theorem holds trivially.

Case S-Fill
$$\frac{}{\boxtimes\, \tau, \sigma \rightsquigarrow \boxtimes\, \tau, \sigma}$$
Since this case does not alter the expression, the theorem holds trivially.

Case S-Edit
$$\frac{}{\square\, v, \sigma \rightsquigarrow \square\, v, \sigma}$$
Since this case does not alter the expression, the theorem holds trivially.

Case S-And
$$\frac{t_1, \sigma \rightsquigarrow t_1', \sigma' \qquad t_2, \sigma' \rightsquigarrow t_2', \sigma''}{t_1 \bowtie t_2, \sigma \rightsquigarrow t_1' \bowtie t_2', \sigma''}$$
Given that $\Gamma, \Sigma \vdash t_1 \bowtie t_2 : \textsc{Task}(\tau_1 \times \tau_2)$, by T-And we have $\Gamma, \Sigma \vdash t_1 : \tau_1$ and $\Gamma, \Sigma \vdash t_2 : \tau_2$. By the induction hypothesis, we also have $\Gamma, \Sigma \vdash t_1' : \tau_1$ and $\Gamma, \Sigma \vdash t_2' : \tau_2$. This gives us that $\Gamma, \Sigma \vdash t_1' \bowtie t_2' : \textsc{Task}(\tau_1 \times \tau_2)$.

Case S-Next
$$\frac{t_1, \sigma \rightsquigarrow t_1', \sigma'}{t_1 \rhd e_2, \sigma \rightsquigarrow t_1' \rhd e_2, \sigma'}$$
Given that $\Gamma, \Sigma \vdash e_1 \rhd e_2 : \textsc{Task}\,\tau$, T-Then gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \rightarrow \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ preserves and thus $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau_1$. Therefore $\Gamma, \Sigma \vdash t_1' \rhd e_2 : \textsc{Task}\,\tau$.

Case S-OrLeft
$$\frac{t_1, \sigma \rightsquigarrow t_1', \sigma'}{t_1 \blacklozenge t_2, \sigma \rightsquigarrow t_1', \sigma'}\ \mathcal{V}(t_1', \sigma') = v_1$$
Given that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$, by T-Or we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ preserves and thus $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau$.

Case S-OrRight
$$\frac{t_1, \sigma \rightsquigarrow t_1', \sigma' \qquad t_2, \sigma' \rightsquigarrow t_2', \sigma''}{t_1 \blacklozenge t_2, \sigma \rightsquigarrow t_2', \sigma''}\ \mathcal{V}(t_1', \sigma') = \bot \wedge \mathcal{V}(t_2', \sigma'') = v_2$$
Given that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$, by T-Or we have $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_2 \rightsquigarrow t_2'$ preserves and thus $\Gamma, \Sigma \vdash t_2' : \textsc{Task}\,\tau$.

Case S-OrNone
$$\frac{t_1, \sigma \rightsquigarrow t_1', \sigma' \qquad t_2, \sigma' \rightsquigarrow t_2', \sigma''}{t_1 \blacklozenge t_2, \sigma \rightsquigarrow t_1' \blacklozenge t_2', \sigma''}\ \mathcal{V}(t_1', \sigma') = \bot \wedge \mathcal{V}(t_2', \sigma'') = \bot$$
Given that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$, by T-Or we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ and $t_2 \rightsquigarrow t_2'$ preserve, and thus $\Gamma, \Sigma \vdash t_1' \blacklozenge t_2' : \textsc{Task}\,\tau$.

Case $\dfrac{t_1, \sigma \rightsquigarrow t_1', \sigma'}{t_1 \blacktriangleright e_2, \sigma \rightsquigarrow t_1' \blacktriangleright e_2, \sigma'}\ \mathcal{V}(t_1', \sigma') = \bot$ S-ThenStay

Given that $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \textsc{Task}\,\tau$, by T-Then we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ preserves, and thus $\Gamma, \Sigma \vdash t_1' \blacktriangleright e_2 : \textsc{Task}\,\tau$.

Case $\dfrac{t_1, \sigma \rightsquigarrow t_1', \sigma' \quad e_2\ v_1, \sigma' \downarrow t_2, \sigma''}{t_1 \blacktriangleright e_2, \sigma \rightsquigarrow t_1' \blacktriangleright e_2, \sigma'}\ \mathcal{V}(t_1', \sigma') = v_1 \wedge \mathcal{F}(t_2, \sigma'')$ S-ThenFail

Given that $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \textsc{Task}\,\tau$, by T-Then we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $e_2 : \tau_1 \to \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ preserves, and thus $\Gamma, \Sigma \vdash t_1' \blacktriangleright e_2 : \textsc{Task}\,\tau$.

Case $\dfrac{t_1, \sigma \rightsquigarrow t_1', \sigma' \quad e_2\ v_1, \sigma' \downarrow t_2, \sigma''}{t_1 \blacktriangleright e_2, \sigma \rightsquigarrow t_2, \sigma''}\ \mathcal{V}(t_1', \sigma') = v_1 \wedge \neg\mathcal{F}(t_2, \sigma'')$ S-ThenCont

Given that $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \textsc{Task}\,\tau$, by T-Then we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1 \rightsquigarrow t_1'$ preserves. By Lemma B.1, we know that $\mathcal{V}(t_1') = v_1$ preserves. By **??** we know that $e_2 v_1 \downarrow t_2$ preserves. And finally by the induction hypothesis, we know that $t_2 \rightsquigarrow t_2'$ preserves. Therefore $\Gamma, \Sigma \vdash t_2' : \textsc{Task}\,\tau$.

□

B.4 ??

Proof. We prove **??** by induction on e:

Case $\dfrac{e, \sigma \downarrow t, \sigma' \quad t, \sigma' \rightsquigarrow t', \sigma''}{e, \sigma \Downarrow t, \sigma'}\ \sigma' = \sigma'' \wedge t = t'$ N-Done

Given that $\Gamma, \Sigma \vdash e : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, we know that $\Gamma, \Sigma \vdash t : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s'$ by **??**. Then by Lemma B.2, we have $\Gamma, \Sigma \vdash t' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s''$.

Case $\dfrac{e, \sigma \downarrow t, \sigma' \quad t, \sigma' \rightsquigarrow t', \sigma'' \quad t', \sigma'' \Downarrow t'', \sigma'''}{e, \sigma \Downarrow t'', \sigma'''}\ \sigma' \neq \sigma'' \vee t \neq t'$ N-Repeat

Given that $\Gamma, \Sigma \vdash e : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, we know that $\Gamma, \Sigma \vdash t : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s'$ by **??**. Then by Lemma B.2, we have $\Gamma, \Sigma \vdash t' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s''$. Then by the induction hypothesis, we finally obtain that $\Gamma, \Sigma \vdash t'' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s'''$.

□

B.5 ??

We require the following Lemma for this proof.

Lemma B.3. *Given that* $\Gamma, \Sigma \vdash s$, $\Sigma(l) = \tau$ *and* $\Gamma, \Sigma \vdash v : \tau$, *it holds that* $\Gamma, \Sigma \vdash s[l \mapsto v]$

This lemma follows immediately from definition.

Proof. We prove **??** by induction on e:

Case $\dfrac{}{\square\, v, \sigma \xrightarrow{v'} \square\, v', \sigma}\ v, v' : \tau$ H-Change

Given that $\Gamma, \Sigma \vdash \square\, v : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, the H-Change rule additionally gives us that $v, v' : \tau$. Therefore by T-Edit we have that $\Gamma, \Sigma \vdash \square\, v' : \textsc{Task}\,\tau$.

Case $\dfrac{}{\boxtimes\, \tau, \sigma \xrightarrow{v'} \square\, v', \sigma}\ v' : \tau$ H-Fill

Given that $\Gamma, \Sigma \vdash \boxtimes\, \tau$ and $\Gamma, \Sigma \vdash s$, the H-Fill rule additionally gives us that $v' : \tau$. Then by T-Enter we have $\Gamma, \Sigma \vdash \square\, v' : \textsc{Task}\,\tau$.

Case H-Update
$$\frac{}{\blacksquare\, l, \sigma \xrightarrow{v'} \blacksquare\, l, \sigma[l \mapsto v']}\ \sigma(l), v' : \tau$$

Given that $\Gamma, \Sigma \vdash \blacksquare\, l : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$. This gives us that $\Sigma(l) = \tau$, and we additionally obtain $s(l), v' : \tau$ by H-Update. By application of Lemma B.3 this case holds.

Case H-PickLeft
$$\frac{e_1, \sigma \Downarrow t_1, \sigma'}{e_1 \lozenge e_2, \sigma \xrightarrow{\mathsf{L}} t_1, \sigma'}\ \neg\mathcal{F}(t_1, \sigma')$$

Given that $\Gamma, \Sigma \vdash t_1 \lozenge t_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, then by T-Xor we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$.

Case H-PickRight
$$\frac{e_2, \sigma \Downarrow t_2, \sigma'}{e_1 \lozenge e_2, \sigma \xrightarrow{\mathsf{R}} t_2, \sigma'}\ \neg\mathcal{F}(t_2, \sigma')$$

Given that $\Gamma, \Sigma \vdash t_1 \lozenge t_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, then by T-Xor we have $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$.

Case H-Next
$$\frac{e_2\ v_1, \sigma \Downarrow t_2, \sigma'}{t_1 \rhd e_2, \sigma \xrightarrow{\mathsf{C}} t_2, \sigma'}\ \mathcal{V}(t_1, \sigma) = v_1 \land \neg\mathcal{F}(t_2, \sigma')$$

Given that $\Gamma, \Sigma \vdash t_1 \rhd e_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$. Then by T-Next, we have $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \textsc{Task}\,\tau$. Then by T-Then we obtain $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \textsc{Task}\,\tau$.

Case H-PassThen
$$\frac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \blacktriangleright e_2, \sigma \xrightarrow{i} t_1' \blacktriangleright e_2, \sigma'}$$

Given that $\Gamma, \Sigma \vdash t_1 \blacktriangleright e_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, T-Then gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1, s \xrightarrow{i} t_1', s'$ also preserves and thus $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau_1$ and $\Gamma, Sigma \vdash s'$. By T-Then we now obtain that $\Gamma, \Sigma \vdash t_1' \blacktriangleright e_2 : \textsc{Task}\,\tau$.

Case H-PassNext
$$\frac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \rhd e_2, \sigma \xrightarrow{i \neq \mathsf{C}} t_1' \rhd e_2, \sigma'}$$

Given that $\Gamma, \Sigma \vdash t_1 \rhd e_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, T-Next gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash e_2 : \tau_1 \to \textsc{Task}\,\tau$. By the induction hypothesis, we know that $t_1, s \xrightarrow{i} t_1', s'$ also preserves and thus $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau_1$ and $\Gamma, Sigma \vdash s'$. By T-Next we now obtain that $\Gamma, \Sigma \vdash t_1' \rhd e_2 : \textsc{Task}\,\tau$.

Case H-FirstAnd
$$\frac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \bowtie t_2, \sigma \xrightarrow{\mathsf{F}\, i} t_1' \bowtie t_2, \sigma'}$$

Given that $\Gamma, \Sigma \vdash t_1 \bowtie t_2 : \textsc{Task}(\tau_1 \times \tau_2)$ and $\Gamma, \Sigma \vdash s$, T-And gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau_2$. By the induction hypothesis, we know that $t_1, s \xrightarrow{i} t_1', s'$ also preserves and thus $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash s'$. Therefore by T-Next we obtain $\Gamma, \Sigma \vdash t_1' \bowtie t_2 : \textsc{Task}(\tau_1 \times \tau_2)$.

Case H-SecondAnd
$$\frac{t_2, \sigma \xrightarrow{i} t_2', \sigma'}{t_1 \bowtie t_2, \sigma \xrightarrow{\mathsf{S}\, i} t_1 \bowtie t_2', \sigma'}$$

Given that $\Gamma, \Sigma \vdash t_1 \bowtie t_2 : \textsc{Task}(\tau_1 \times \tau_2)$ and $\Gamma, \Sigma \vdash s$, T-And gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau_1$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau_2$. By the induction hypothesis, we know that $t_2, s \xrightarrow{i} t_2', s'$ also preserves and thus $\Gamma, \Sigma \vdash t_2' : \textsc{Task}\,\tau_2$ and $\Gamma, \Sigma \vdash s'$. Therefore by T-Next we obtain $\Gamma, \Sigma \vdash t_1 \bowtie t_2' : \textsc{Task}(\tau_1 \times \tau_2)$.

Case
$$\frac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \blacklozenge t_2, \sigma \xrightarrow{\mathsf{F}\,i} t_1' \blacklozenge t_2, \sigma'}\ \textsc{H-FirstOr}$$
Given that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, T-Or gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis we know that $t_1, s \xrightarrow{i} t_1', s'$ also preserves, and therefore $\Gamma, \Sigma \vdash t_1' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s'$. By T-Or we now obtain $\Gamma, \Sigma \vdash t_1' \blacklozenge t_2 : \textsc{Task}\,\tau$.

Case
$$\frac{t_2, \sigma \xrightarrow{i} t_2', \sigma'}{t_1 \blacklozenge t_2, \sigma \xrightarrow{\mathsf{S}\,i} t_1 \blacklozenge t_2', \sigma'}\ \textsc{H-SecondOr}$$
Given that $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s$, T-Or gives us that $\Gamma, \Sigma \vdash t_1 : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash t_2 : \textsc{Task}\,\tau$. By the induction hypothesis we know that $t_2, s \xrightarrow{i} t_2', s'$ also preserves, and therefore $\Gamma, \Sigma \vdash t_2' : \textsc{Task}\,\tau$ and $\Gamma, \Sigma \vdash s'$. By T-Or we now obtain $\Gamma, \Sigma \vdash t_1 \blacklozenge t_2' : \textsc{Task}\,\tau$.

□

B.6 ??

Proof. We prove **??** by induction on e'.

Case $e = \lightning$

$\mathcal{F}(\lightning, s) = \mathsf{True}$, and there is no handling rule that applies to fail.

Case $e = \square\, v$

$\Gamma, \Sigma \vdash \square\, v : \textsc{Task}\,\tau$, $\mathcal{F}(\square\, v, s) = \mathsf{False}$, and there exists an input i, namely $v' : \tau$.

Case $e = \boxtimes\, \tau$

$\mathcal{F}(\boxtimes\, \tau) = \mathsf{False}$, and there exists an in put i, namely $v : \tau$.

Case $e = \blacksquare\, l$

Given that $\Gamma, \Sigma \vdash \blacksquare\, l : \textsc{Task}\,\tau$, $\mathcal{F}(\blacksquare\, l, s) = \mathsf{False}$, and there exists an input i, namely $v : \tau$.

Case $e = t_1 \blacktriangleright e_2$

$\mathcal{F}(t_1 \blacktriangleright e_2, s) = \mathcal{F}(t_1, s)$. If there exists an i for t_1, then this i also applies to $t_1 \blacktriangleright e_2$. This case therefore holds by the induction hypothesis.

Case $e = t_1 \triangleright e_2$

$\mathcal{F}(t_1 \triangleright e_2, s) = \mathcal{F}(t_1, s)$. If there exists an i for t_1, then this i also applies to $t_1 \triangleright e_2$. This case therefore holds by the induction hypothesis.

Case $e = e_1 \lozenge e_2$

We normalise the two expressions first, $e_1, s \rightsquigarrow t_1, s', e_2, s \rightsquigarrow t_2, s'$ and we can then be in two situations. One, we can have that $\mathcal{F}(t_1, s')$ and $\mathcal{F}(t_2, s')$ are both true. If that is so, then by definition, we have both $\mathcal{F}(e_1 \lozenge e_2, s)$ and no rule of the handling semantics applies, and therefore there exists no input for this case.

Or we are in the situation where one or both of the two sub expressions does not fail. In that case, we know that $\mathcal{F}(e_1 \lozenge e_2, s)$ does not hold, and that at least one of the handling rules applies. Therefore, there must be an input i, namely L, R or both.

Case $e = t_1 \bowtie t_2$

We can again find ourselves in one of two situations. In the first case, both sub expressions fail, $\mathcal{F}(t_1, s)$ and $\mathcal{F}(t_2, s)$. In that case, we know that $\mathcal{F}(t_1 \bowtie t_2, s)$ also fails by definition. By the induction hypothesis, we know that for both t_1 and t_2 there is no input that can be handled. Since the only two rules for $\bowtie$ that handle input just pass this input on to one of the two expressions, we know that indeed no i applies.

In the case that one or both sub expressions do not fail, then by definition $t_1 \bowtie t_2$ not failing under s. Again by induction hypothesis, we know that for one or both of the expressions, there exits an i that can be handled. Then by H-FirstAnd and H-SecondAnd, we know that we can pass this i, by prefixing it with either F or S.

Case $e = t_1 \blacklozenge t_2$

We can again find ourselves in one of two situations. In the first case, both sub expressions fail, $\mathcal{F}(t_1, s)$ and $\mathcal{F}(t_2, s)$. In that case, we know that $\mathcal{F}(t_1 \blacklozenge t_2, s)$ also fails by definition. By the induction hypothesis, we know that for both t_1 and t_2 there is no input that can be handled. Since the only two rules for $\blacklozenge$ that handle input just pass this input on to one of the two expressions, we know that indeed no i applies.

In the case that one or both sub expressions do not fail, then by definition $t_1 \blacklozenge t_2$ not failing under s. Again by induction hypothesis, we know that for one or both of the expressions, there exits an i that can be handled. Then by H-FirstOr and H-SecondOr, we know that we can pass this i, by prefixing it with either F or S.

□

B.7 ??

Proof. **Case** $e = \square\, v : \textsc{Task}\, \tau, i = v' : \tau$

Given that $\dfrac{\text{H-Change}}{\square\, v, \sigma \xrightarrow{v'} \square\, v', \sigma}\; v, v' : \tau$, we have by definition that $\mathcal{I}(\square\, v : \textsc{Task}\, \tau, s) = \{v' : \tau, \mathsf{E}\}$, which includes $v' : \tau$.

Case $e = \boxtimes\, \tau, i = v' : \tau$

Given that $\dfrac{\text{H-Fill}}{\boxtimes\, \tau, \sigma \xrightarrow{v'} \square\, v', \sigma}\; v' : \tau$, we have by definition that $\mathcal{I}(\boxtimes\, \tau, s) = \{v' : \tau\}$, which includes $v' : \tau$.

Case $e = \blacksquare\, l : \textsc{Task}\, \tau, i = v' : \tau$

Given that $\dfrac{\text{H-Update}}{\blacksquare\, l, \sigma \xrightarrow{v'} \blacksquare\, l, \sigma[l \mapsto v']}\; \sigma(l), v' : \tau$, we have by definition that $\mathcal{I}(\blacksquare\, l : \textsc{Task}\, \tau, s) = \{v' : \tau\}$, which includes $v' : \tau$.

Case $e = t_1 \lozenge t_2, i = \mathsf{L}$

Given that H-PickLeft $\dfrac{e_1, \sigma \Downarrow t_1, \sigma'}{e_1 \lozenge e_2, \sigma \xrightarrow{\mathsf{L}} t_1, \sigma'}\; \neg\mathcal{F}(t_1, \sigma')$, we have by definition that $\mathcal{I}(t_1 \lozenge t_2, s) = \{\mathsf{L}, \mathsf{R}\}$, which includes L.

Case $e = t_1 \lozenge t_2, i = \mathsf{R}$

Given that H-PickRight $\dfrac{e_2, \sigma \Downarrow t_2, \sigma'}{e_1 \lozenge e_2, \sigma \xrightarrow{\mathsf{R}} t_2, \sigma'}\; \neg\mathcal{F}(t_2, \sigma')$, we have by definition that $\mathcal{I}(t_1 \lozenge t_2, s) = \{\mathsf{L}, \mathsf{R}\}$, which includes R.

Case $e = t_1 \rhd e_2, i = \mathsf{C}$

Given that H-Next $\dfrac{e_2\ v_1, \sigma \Downarrow t_2, \sigma'}{t_1 \rhd e_2, \sigma \xrightarrow{\mathsf{C}} t_2, \sigma'}\; \mathcal{V}(t_1, \sigma) = v_1 \land \neg\mathcal{F}(t_2, \sigma')$, we have by definition that $\mathcal{I}(t_1 \rhd e_2, s) = \mathcal{I}(t_1, s) \cup \{\mathsf{C} \mid \mathcal{V}(t_1, s) = v_1 \land \neg\mathcal{F}(e_2 v_1, s \ \rightsquigarrow)\}$. If the H-Next rule applies, this means that the conditions $\mathcal{V}(t_1, s) = v_1 \land \neg\mathcal{F}(e_2 v_1, s \ \rightsquigarrow)$ are fulfilled, and therefore C is contained.

Case $e = t_1 \rhd e_2, i \neq \mathsf{C}$

Given that H-PassNext $\dfrac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \rhd e_2, \sigma \xrightarrow{i \neq \mathsf{C}} t_1' \rhd e_2, \sigma'}$, we have by definition that $\mathcal{I}(t_1 \rhd e_2, s) = \mathcal{I}(t_1, s) \cup \{\mathsf{C} \mid \mathcal{V}(t_1, s) = v_1 \land \neg\mathcal{F}(e_2 v_1, s \ \rightsquigarrow)\}$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_1, s)$, and by definition of $\mathcal{I}$, i is therefore also included in this case.

Case $e = t_1 \blacktriangleright e_2, i$

Given that H-PassThen $\dfrac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \blacktriangleright e_2, \sigma \xrightarrow{i} t_1' \blacktriangleright e_2, \sigma'}$, we have by definition that $\mathcal{I}(t_1 \blacktriangleright e_2, s) = \mathcal{I}(t_1, s)$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_1, s)$, and by definition of $\mathcal{I}$, i is therefore also included in this case.

Case $e = t_1 \bowtie t_2, i = \mathsf{F}\, i$

Given that H-FirstAnd $\dfrac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \bowtie t_2, \sigma \xrightarrow{\mathsf{F}\, i} t_1' \bowtie t_2, \sigma'}$ we have by definition that $\mathcal{I}(t_1 \bowtie t_2, s) = \{\mathsf{F}\ i \mid i \in \mathcal{I}(t_1, s)\} \cup \{\mathsf{S}\ i \mid i \in \mathcal{I}(t_2, s)\}$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_1, s)$, and by definition of $\mathcal{I}$, $\mathsf{F}\, i$ is therefore als included in this case.

Case $e = t_1 \bowtie t_2, i = \mathsf{S}\, i$

Given that $\dfrac{t_2, \sigma \xrightarrow{i} t_2', \sigma'}{t_1 \bowtie t_2, \sigma \xrightarrow{\mathsf{S}\, i} t_1 \bowtie t_2', \sigma'}$ H-SecondAnd we have by definition that $\mathcal{I}(t_1 \bowtie t_2) = \{\mathsf{F}\ i \mid i \in \mathcal{I}(t_1, s)\} \cup \{\mathsf{S}\ i \mid i \in \mathcal{I}(t_2, s)\}$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_2, s)$, and by definition of $\mathcal{I}$, $\mathsf{S}\, i$ is therefore als included in this case.

Case $e = t_1 \blacklozenge t_2, i = \mathsf{F}\, i$

Given that $\dfrac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \blacklozenge t_2, \sigma \xrightarrow{\mathsf{F}\, i} t_1' \blacklozenge t_2, \sigma'}$ H-FirstOr we have by definition that $\mathcal{I}(t_1 \blacklozenge t_2, s) = \{\mathsf{F}\ i \mid i \in \mathcal{I}(t_1, s)\} \cup \{\mathsf{S}\ i \mid i \in \mathcal{I}(t_2, s)\}$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_1, s)$, and by definition of $\mathcal{I}$, $\mathsf{F}\, i$ is therefore als included in this case.

Case $e = t_1 \blacklozenge t_2, i = \mathsf{S}\, i$

Given that $\dfrac{t_1, \sigma \xrightarrow{i} t_1', \sigma'}{t_1 \blacklozenge t_2, \sigma \xrightarrow{\mathsf{F}\, i} t_1' \blacklozenge t_2, \sigma'}$ H-FirstOr we have by definition that $\mathcal{I}(t_1 \blacklozenge t_2, s) = \{\mathsf{F}\ i \mid i \in \mathcal{I}(t_1, s)\} \cup \{\mathsf{S}\ i \mid i \in \mathcal{I}(t_2, s)\}$. By the induction hypothesis, we have that $i \in \mathcal{I}(t_2, s)$, and by definition of $\mathcal{I}$, $\mathsf{S}\, i$ is therefore als included in this case.

□