

Type Vigilance: Why Transient is Stronger Than You Thought (Technical Report)

CONTENTS

Contents	1
1 Common Definitions	2
1.1 Evaluation Language Definitions	2
1.2 Operational Semantics	4
1.3 Store-Based Evaluation Language Definitions	6
1.4 Store-Based Operational Semantics	8
1.5 Operational Semantics Simulation Result	11
2 Simple Typing	12
2.1 Simple Definitions	12
3 Tag Typing	14
3.1 Definition	14
3.2 Simple Typing Implies Tag Typing	15
4 Truer Transient Typing	16
4.1 Definition	16
4.2 Simple Typing Implies Truer Transient Typing	18
4.3 Tag Typing Implies Truer Transient Typing	22
5 Vigilance	23
5.1 Vigilance Logical Relation	23
5.2 Vigilance Theorem	25
5.3 Vigilance Fundamental Property for Natural with Simple Typing	26
5.4 Vigilance Fundamental Property for Transient with Truer Transient Typing	46
5.5 Vigilance Fundamental Property for Transient with Tag Typing	67
6 Contextual equivalence	68
6.1 Contextual Equivalence Logical Relation—No Store	68
6.2 Context typing	69
6.3 Contextual equivalence statement	71
6.4 Binary relation—Proofs	71
6.5 Context relation—Proofs	84
6.6 Check optimization	87
6.7 Check-elision—Proofs	89
7 GTL	92
7.1 Simple Translation	93
7.2 Tag Transient Translation	97
7.3 Truer Transient Translation	97
8 Vigilance Results for GTLs	104

Author's address:

2023-04-10 15:45. Page 1 of 1–104.

1

53	8.1	GTL Vigilance for Simple Typing with Natural Semantics	104
54	8.2	GTL Vigilance for Tag Typing with Transient Semantics	104
55	8.3	GTL Vigilance for Truer Transient Typing with Transient Semantics	104

1 Common Definitions

1.1 Evaluation Language Definitions

Evaluation Language

71	v	$::= n \mid i \mid \text{True} \mid \text{False} \mid \langle v, v \rangle \mid w$
72	w	$::= \lambda(x:\tau). e \mid \text{grd} \{ \tau \Leftarrow \tau \} w$
73	E	$::= [] \mid \langle E, e \rangle \mid \langle v, E \rangle \mid \text{fst} \{ \tau \} E \mid \text{snd} \{ \tau \} E \mid \text{app} \{ \tau \} E e \mid \text{app} \{ \tau \} v E \mid E e \mid v E \mid \text{binop} E e \mid \text{binop} v E$
74		$\mid \text{cast} \{ \tau \Leftarrow \tau' \} E \mid \text{if } E \text{ then } e \text{ else } e \mid \text{mon} \{ \tau \Leftarrow \tau \} E \mid \text{assert } \tau E$
75	Err°	$::= \text{Wrong}$
76	Err^\bullet	$::= \text{DivErr} \mid \text{TypeErr}(\tau, v)$
77	Err	$::= \text{Err}^\circ \mid \text{Err}^\bullet$
78	e	$::= \text{Err} \mid x \mid n \mid i \mid \lambda(x:\tau). e \mid \langle e, e \rangle \mid \text{app} \{ \tau \} e e \mid e e \mid \text{fst} \{ \tau \} e \mid \text{snd} \{ \tau \} e \mid \text{binop} e e \mid \text{cast} \{ \tau \Leftarrow \tau' \} e$
79		$\mid \text{if } e \text{ then } e \text{ else } e \mid \text{mon} \{ \tau \Leftarrow \tau \} e \mid \text{grd} \{ \tau \Leftarrow \tau \} e \mid \text{assert } \tau e$
80	K	$::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid * \times * \mid * \rightarrow * \mid *$
81	τ	$::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid \tau \times \tau \mid \tau \rightarrow \tau \mid *$
82	binop	$::= \text{sum} \mid \text{quotient}$
83	n	$::= \mathbb{N}$
84	i	$::= \mathbb{Z}$

$$\text{oc}: K \times v \longrightarrow \mathbb{B}$$

$$v_0 \text{ oc } K_0 = \begin{cases} \text{True} & \begin{cases} \text{if } K_0 = \text{Nat} \text{ and } v_0 \in \mathbb{N} \\ \text{or } K_0 = \text{Int} \text{ and } v_0 \in \mathbb{Z} \\ \text{or } K_0 = \text{Bool} \text{ and } v_0 \in \mathbb{B} \\ \text{or } K_0 = * \times * \text{ and } v_0 \in \langle v, v \rangle \\ \text{or } K_0 = * \rightarrow * \text{ and } v_0 \in w \\ \text{or } K_0 = * \end{cases} \\ \text{False} & \text{otherwise} \end{cases}$$

$$\delta : \text{binop} \times v \times v \longrightarrow e$$

$$\delta(\text{binop}, i_0, i_1) = \begin{cases} i_0 + i_1 & \text{if } \text{binop} = \text{sum}\{\tau\} \\ \text{DivErr} & \\ \text{if } \text{binop} = \text{quotient}\{\tau\} & \\ \text{and } i_1 = 0 & \\ \lfloor i_0 / i_1 \rfloor & \\ \text{if } \text{binop} = \text{quotient}\{\tau\} & \\ \text{and } i_1 \neq 0 & \end{cases}$$

105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156

$$\alpha_{pos}^L : \tau \times v \longrightarrow \mathbb{B}$$

L	$v \alpha_{bnd}^L \tau$	$v \alpha_{mon}^L \tau$	$v \alpha_{check}^L \tau$
N	$v \alpha [\tau]$	$v \alpha [\tau]$	True
T	$v \alpha [\tau]$	True	$v \alpha [\tau]$

1.2 Operational Semantics

157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208

\longrightarrow_L^* reflexive-transitive closure of \longrightarrow_L

\longrightarrow_L compatible closure of \hookrightarrow_L

$e \mapsto_L e$

$\text{fst}\{\tau_0\} v_0 \mapsto_L \text{Wrong}$
if $v_0 \neq \langle v_1, v_2 \rangle$

$\text{fst}\{\tau_0\} \langle v_0, v_1 \rangle \mapsto_L \text{assert } \tau_0 v_0$

$\text{snd}\{\tau_0\} v_0 \mapsto_L \text{Wrong}$
if $v_0 \neq \langle v_1, v_2 \rangle$

$\text{snd}\{\tau_0\} \langle v_0, v_1 \rangle \mapsto_L \text{assert } \tau_0 v_1$

$\text{binop} v_0 v_1 \mapsto_L \text{Wrong}$
if $\delta(\text{binop}, v_0, v_1)$ is undefined

$\text{binop} v_0 v_1 \mapsto_L \text{assert } \tau_0 \delta(\text{binop}, v_0, v_1)$
if $\delta(\text{binop}, v_0, v_1)$ is defined

$\text{app}\{\tau_0\} v_0 v_1 \mapsto_L \text{assert } \tau_0 (v_0 v_1)$

$v_0 v_1 \mapsto_L \text{Wrong}$
if $v_0 \neq w_0$

$(\lambda(x_0 : \tau_1). e_0) v_1 \mapsto_L e_0[x_0 \leftarrow v_1]$
if $v_1 \alpha_{check}^L \tau_1$

$(\lambda(x_0 : \tau_1). e_0) v_1 \mapsto_L \text{TypeErr}(\tau_1, v_1)$
if $\neg v_1 \alpha_{check}^L \tau_1$

$(\text{grd}\{\tau_1 \Leftarrow \tau_2\} w_0) v_1 \mapsto_L \text{mon}\{\text{cod}(\tau_1) \Leftarrow \text{cod}(\tau_2)\} (w_0 (\text{mon}\{\text{dom}(\tau_2) \Leftarrow \text{dom}(\tau_1)\} v_1))$

$\text{cast}\{\tau_1 \Leftarrow \tau_0\} v_0 \mapsto_L \text{mon}\{\tau_1 \Leftarrow \tau_0\} v_0$
if $v_0 \alpha_{bnd}^L \tau_1$
and $v_0 \alpha_{bnd}^L \tau_0$

209 $\text{cast } \{\tau_1 \Leftarrow \tau_0\} v_0 \quad \rightsquigarrow_L \text{TypeErr}(\tau_1, v_0)$
 210 $\quad \text{if } \neg v_0 \alpha_{\text{bind}}^L \tau_1$
 211
 212 $\text{cast } \{\tau_1 \Leftarrow \tau_0\} v_0 \quad \rightsquigarrow_L \text{TypeErr}(\tau_0, v_0)$
 213 $\quad \text{if } \neg v_0 \alpha_{\text{bind}}^L \tau_0$
 214
 215
 216 $\text{mon } \{\tau_1 \Leftarrow \tau_2\} i_0 \quad \rightsquigarrow_L i_0$
 217 $\quad \text{if } i_0 \alpha_{\text{mon}}^L \tau_1 \wedge i_0 \alpha_{\text{mon}}^L \tau_2$
 218
 219
 220 $\text{mon } \{\tau_1 \Leftarrow \tau_2\} \langle v_0, v_1 \rangle \rightsquigarrow_L \langle \text{mon } \{\text{fst}(\tau_1) \Leftarrow \text{fst}(\tau_2)\} v_0, \text{mon } \{\text{snd}(\tau_1) \Leftarrow \text{snd}(\tau_2)\} v_1 \rangle$
 221
 222
 223 $\text{mon } \{\tau_1 \Leftarrow \tau_2\} w \quad \rightsquigarrow_L \text{grd } \{\tau_1 \Leftarrow \tau_2\} w$
 224 $\quad \text{if } w \alpha_{\text{mon}}^L \tau_1 \wedge w \alpha_{\text{mon}}^L \tau_2$
 225
 226
 227 $\text{mon } \{\tau_0 \Leftarrow \tau_1\} v_0 \quad \rightsquigarrow_L \text{TypeErr}(\tau_0, v_0)$
 228 $\quad \text{if } \neg v_0 \alpha_{\text{mon}}^L \tau_0$
 229
 230
 231 $\text{mon } \{\tau_0 \Leftarrow \tau_1\} v_0 \quad \rightsquigarrow_L \text{TypeErr}(\tau_1, v_0)$
 232 $\quad \text{if } \neg v_0 \alpha_{\text{mon}}^L \tau_1$
 233
 234 $\text{if True then } e_1 \text{ else } e_2 \rightsquigarrow_L e_1$
 235
 236
 237 $\text{if False then } e_1 \text{ else } e_2 \rightsquigarrow_L e_2$
 238
 239 $\text{assert } \tau_0 v_0 \quad \rightsquigarrow_L v_0$
 240 $\quad \text{if } v_0 \alpha_{\text{check}}^L \tau_0$
 241
 242
 243 $\text{assert } \tau_0 v_0 \quad \rightsquigarrow_L \text{TypeErr}(\tau_0, v_0)$
 244 $\quad \text{if } \neg v_0 \alpha_{\text{check}}^L \tau_0$
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260

1.3 Store-Based Evaluation Language Definitions

Store-Based Evaluation Language

$v ::= \ell \mid n \mid i \mid \text{True} \mid \text{False} \mid \langle \ell, \ell \rangle \mid \lambda(x:\tau).e$
 $\text{Err}^\circ ::= \text{Wrong}$
 $\text{Err}^\bullet ::= \text{DivErr} \mid \text{TypeErr}(\tau, v)$
 $\text{Err} ::= \text{Err}^\circ \mid \text{Err}^\bullet$
 $e ::= \text{Err} \mid x \mid \ell \mid v \mid \langle e, e \rangle \mid \text{app}\{\tau\} e e \mid e e \mid \text{fst}\{\tau\} e \mid \text{snd}\{\tau\} e \mid \text{binop} e e \mid \text{cast}\{\tau \Leftarrow \tau'\} e$
 $\quad \mid \text{if } e \text{ then } e \text{ else } e \mid \text{mon}\{\tau \Leftarrow \tau\} e \mid \text{assert } \tau e$
 $K ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid * \times * \mid * \rightarrow * \mid *$
 $\tau ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid \tau \times \tau \mid \tau \rightarrow \tau \mid *$
 $\text{binop} ::= \text{sum} \mid \text{quotient}$
 $\Sigma \in \mathbb{L} \mapsto \mathbb{V} \times \text{option}(\mathbb{T} \times \mathbb{T})$
 $\ell \in \mathbb{L}$
 $n \in \mathbb{N}$
 $i \in \mathbb{Z}$
 $E ::= [] \mid \langle E, e \rangle \mid \langle \ell, E \rangle \mid \text{fst}\{\tau\} E \mid \text{snd}\{\tau\} E \mid \text{app}\{\tau\} E e \mid \text{app}\{\tau\} \ell E \mid E e \mid \ell E \mid \text{binop} E e \mid \text{binop} \ell E$
 $\quad \mid \text{cast}\{\tau \Leftarrow \tau'\} E \mid \text{if } E \text{ then } e \text{ else } e \mid \text{mon}\{\tau \Leftarrow \tau\} E \mid \text{assert } \tau E$

$\alpha: K \times \mathbb{V} \rightarrow \mathbb{B}$

$v_0 \alpha K_0 = \begin{cases} \text{True} & \text{if } K_0 = \text{Nat} \text{ and } v_0 \in \mathbb{N} \\ & \text{or } K_0 = \text{Int} \text{ and } v_0 \in \mathbb{Z} \\ & \text{or } K_0 = \text{Bool} \text{ and } v_0 \in \mathbb{B} \\ & \text{or } K_0 = * \times * \text{ and } v_0 \in \langle \ell, \ell \rangle \\ & \text{or } K_0 = * \rightarrow * \text{ and } v_0 \in \lambda(x:\tau).e \\ & \text{or } K_0 = * \\ \text{False} & \\ \text{otherwise} & \end{cases}$

$\delta: \text{binop} \times \mathbb{V} \times \mathbb{V} \rightarrow \mathbb{E}$

$\delta(\text{binop}, i_0, i_1) = \begin{cases} i_0 + i_1 & \text{if } \text{binop} = \text{sum}\{\tau\} \\ \text{DivErr} & \text{if } \text{binop} = \text{quotient}\{\tau\} \\ & \text{and } i_1 = 0 \\ \lfloor i_0 / i_1 \rfloor & \text{if } \text{binop} = \text{quotient}\{\tau\} \\ & \text{and } i_1 \neq 0 \end{cases}$

313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364

$$\alpha_{pos}^L: \mathbb{T} \times \mathbb{V} \longrightarrow \mathbb{B}$$

L	$v \alpha_{bnd}^L \tau$	$v \alpha_{mon}^L \tau$	$v \alpha_{check}^L \tau$
N	$v \alpha [\tau]$	$v \alpha [\tau]$	True
T	$v \alpha [\tau]$	True	$v \alpha [\tau]$

$$\text{pointsto}(\Sigma, \ell)$$

$$\text{pointsto}(\Sigma, \ell) = \begin{cases} fst(\Sigma(\ell)) & \text{if } fst(\Sigma(\ell)) \neq \ell' \\ \text{pointsto}(\Sigma, \ell') & \text{if } fst(\Sigma(\ell)) = \ell' \end{cases}$$

1.4 Store-Based Operational Semantics

365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416

\longrightarrow_L^* reflexive-transitive closure of \longrightarrow_L

\longrightarrow_L compatible closure of \hookrightarrow_L

$\Sigma, e \hookrightarrow_L \Sigma, e$

$\Sigma, v \hookrightarrow_L \Sigma[\ell \mapsto (v, \text{none})], \ell$
where $\text{loc} \notin \text{dom}(\Sigma)$

$\Sigma, \text{fst}\{\tau_0\} \ell_0 \hookrightarrow_L \Sigma, \text{Wrong}$
if $\Sigma(\ell_0) \neq (\langle \ell_1, \ell_2 \rangle, _)$

$\Sigma, \text{fst}\{\tau_0\} \ell_0 \hookrightarrow_L \Sigma, \text{assert } \tau_0 \ell_0$
if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, _)$

$\Sigma, \text{snd}\{\tau_0\} \ell_0 \hookrightarrow_L \Sigma, \text{Wrong}$
if $\Sigma(\ell_0) \neq (\langle \ell_1, \ell_2 \rangle, _)$

$\Sigma, \text{snd}\{\tau_0\} \ell_0 \hookrightarrow_L \Sigma, \text{assert } \tau_0 \ell_0$
if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, _)$

$\Sigma, \text{binop } \ell_0 \ell_1 \hookrightarrow_L \Sigma, \text{Wrong}$
if $\delta(\text{binop}, \text{pointsto}(\Sigma, \ell_0), \text{pointsto}(\Sigma, \ell_1))$ is undefined

$\Sigma, \text{binop } \ell_0 \ell_1 \hookrightarrow_L \Sigma, \text{assert } \tau_0 \delta(\text{binop}, \text{pointsto}(\Sigma, \ell_0), \text{pointsto}(\Sigma, \ell_1))$
if $\delta(\text{binop}, \text{pointsto}(\Sigma, \ell_0), \text{pointsto}(\Sigma, \ell_1))$ is defined

$\Sigma, \text{app}\{\tau_0\} \ell_0 \ell_1 \hookrightarrow_L \Sigma, \text{assert } \tau_0 (\ell_0 \ell_1)$

$\Sigma, \ell_0 \ell_1 \hookrightarrow_L \Sigma, \text{Wrong}$
if $\Sigma(\ell_0) = (v, _)$ and $v \notin \lambda(x:\tau). e \cup \ell$
or $\Sigma(\ell_0) = (\ell'_0, \text{none})$

$\Sigma, \ell_0 \ell_1 \hookrightarrow_L \Sigma, e_0[x_0 \leftarrow \ell_1]$
if $\Sigma(\ell_0) = (\lambda(x_0:\tau_1). e_0, _)$ and
 $\text{pointsto}(\Sigma, \ell_1) \alpha_{\text{check}}^L \tau_1$

417 $\Sigma, \ell_0 \ell_1 \quad \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_1, \ell_1)$
 418 if $\Sigma(\ell_0) = (\lambda(x_0 : \tau_1). e_0, _)$ and
 419 $\neg \text{pointsto}(\Sigma, \ell_1) \propto_{check}^L \tau_1$
 420
 421 $\Sigma, \ell_0 \ell_1 \quad \hookrightarrow_L \Sigma, \text{mon} \{ \text{cod}(\tau_1) \Leftarrow \text{cod}(\tau_2) \} (\ell_0 (\text{mon} \{ \text{dom}(\tau_2) \Leftarrow \text{dom}(\tau_1) \} \ell_1))$
 422 if $\Sigma(\ell_0) = (\ell_2, \text{some}(\tau_1, \tau_2))$
 423
 424
 425 $\Sigma, \text{cast} \{ \tau_1 \Leftarrow \tau_0 \} \ell_0 \quad \hookrightarrow_L \Sigma, \text{mon} \{ \tau_1 \Leftarrow \tau_0 \} \ell_0$
 426 if $\text{pointsto}(\Sigma, \ell_0) \propto_{bnd}^L \tau_1$
 427 and $\text{pointsto}(\Sigma, \ell_0) \propto_{bnd}^L \tau_0$
 428
 429
 430 $\Sigma, \text{cast} \{ \tau_1 \Leftarrow \tau_0 \} \ell_0 \quad \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_1, \ell_0)$
 431 if $\neg \text{pointsto}(\Sigma, \ell_0) \propto_{bnd}^L \tau_1$
 432
 433
 434 $\Sigma, \text{cast} \{ \tau_1 \Leftarrow \tau_0 \} \ell_0 \quad \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_0, \ell_0)$
 435 if $\neg \text{pointsto}(\Sigma, \ell_0) \propto_{bnd}^L \tau_0$
 436
 437
 438 $\Sigma, \text{mon} \{ \tau_1 \Leftarrow \tau_2 \} \ell_0 \quad \hookrightarrow_L \Sigma[\ell_1 \mapsto (\ell_0, \text{some}(\tau_1, \tau_2))], \ell_1$
 439 if $\ell_1 \notin \text{dom}(\Sigma)$
 440 and $\text{pointsto}(\Sigma, \ell_0) = v$ where $v = i$ or True or False
 441 and $v \propto_{mon}^L \tau_1 \wedge v \propto_{mon}^L \tau_2$
 442
 443
 444 $\Sigma, \text{mon} \{ \tau_1 \Leftarrow \tau_2 \} \ell_0 \quad \hookrightarrow_L \Sigma, \langle \text{mon} \{ \text{fst}(\tau_1) \Leftarrow \text{fst}(\tau_2) \} \ell_1, \text{mon} \{ \text{snd}(\tau_1) \Leftarrow \text{snd}(\tau_2) \} \ell_2 \rangle$
 445 if $\Sigma(\ell_0) = (\langle \ell_1, \ell_2 \rangle, _)$
 446
 447
 448 $\Sigma, \text{mon} \{ \tau_1 \Leftarrow \tau_2 \} \ell_0 \quad \hookrightarrow_L \Sigma[\ell_1 \mapsto (\ell_0, \text{some}(\tau_1, \tau_2))], \ell_1$
 449 if $\ell_1 \notin \text{dom}(\Sigma)$
 450 and $\text{pointsto}(\Sigma, \ell_0) = v$ and $v = \lambda(x_0 : \tau_1). e_0$
 451 and $v \propto_{mon}^L \tau_1 \wedge v \propto_{mon}^L \tau_2$
 452
 453
 454 $\Sigma, \text{mon} \{ \tau_0 \Leftarrow \tau_1 \} \ell_0 \quad \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_1, \ell_0)$
 455 if $\neg \text{pointsto}(\Sigma, \ell_0) \propto_{mon}^L \tau_1$
 456
 457
 458 $\Sigma, \text{mon} \{ \tau_0 \Leftarrow \tau_1 \} \ell_0 \quad \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_0, \ell_0)$
 459 if $\neg \text{pointsto}(\Sigma, \ell_0) \propto_{mon}^L \tau_0$
 460
 461
 462 $\Sigma, \text{if } \ell_0 \text{ then } e_1 \text{ else } e_2 \quad \hookrightarrow_L \Sigma, e_1$
 463 if $\text{pointsto}(\Sigma, \ell_0) = \text{True}$
 464
 465
 466 $\Sigma, \text{if } \ell_0 \text{ then } e_1 \text{ else } e_2 \quad \hookrightarrow_L \Sigma, e_2$
 467 if $\text{pointsto}(\Sigma, \ell_0) = \text{False}$
 468

469 $\Sigma, \text{if } \ell_0 \text{ then } e_1 \text{ else } e_2 \hookrightarrow_L \Sigma, \text{Wrong}$
 470 $\text{if } \text{pointsto}(\Sigma, \ell_0) \neq \ell \text{ or True or False}$
 471
 472
 473 $\Sigma, \text{assert } \tau_0 \ell_0 \hookrightarrow_L \Sigma, \ell_0$
 474 $\text{if } \text{pointsto}(\Sigma, \ell_0) \alpha_{check}^L \tau_0$
 475
 476 $\Sigma, \text{assert } \tau_0 \ell_0 \hookrightarrow_L \Sigma, \text{TypeErr}(\tau_0, \ell_0)$
 477 $\text{if } \neg \text{pointsto}(\Sigma, \ell_0) \alpha_{check}^L \tau_0$
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520

1.5 Operational Semantics Simulation Result

To compare the two semantics, we have to define a relation that compares values between the two languages. The store semantics will represent:

- (1) Guards as a linked list of pairs of types, ending at a lambda with no types.
- (2) Pairs as a pointer to the two subcomponents, with no types.
- (3) Base values as a linked list of pairs of types, ending at a base value with no types.

We capture this in the following value equivalence:

$$\boxed{(\Sigma, \ell) \equiv v}$$

$$\frac{\text{pointsto}(\Sigma, \ell) = v}{\ell \equiv v} \quad \frac{\begin{array}{c} \Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _) \\ (\Sigma, \ell_1) \equiv v_1 \\ (\Sigma, \ell_2) \equiv v_2 \end{array}}{(\Sigma, \ell) \equiv \langle v_1, v_2 \rangle} \quad \frac{\begin{array}{c} \Sigma(\ell) = (\ell', \text{some}(\tau', \tau)) \\ (\Sigma, \ell') \equiv v \end{array}}{(\Sigma, \ell) \equiv \text{grd} \{ \tau' \leftarrow \tau \} v} \quad \frac{\Sigma(\ell) = (\lambda x : \tau. e, _)}{(\Sigma, \ell) \equiv \lambda x : \tau. e}$$

THEOREM 1.1 (STORE AND NON STORE OPERATIONAL SEMANTICS ARE EQUIVALENT).

$e \longrightarrow_L^* e'$ and e' is irreducible iff $\forall \Sigma. \exists \Sigma', \ell. (\Sigma, e) \longrightarrow_L^* (\Sigma', \ell)$ and $(\Sigma', \ell) \equiv e'$

2 Simple Typing

2.1 Simple Definitions

Simple language

$e ::= x \mid n \mid i \mid \text{True} \mid \text{False} \mid \lambda(x:\tau). e \mid \langle e, e \rangle \mid \text{app}\{\tau\} e e \mid \text{fst}\{\tau\} e \mid \text{snd}\{\tau\} e \mid \text{binop } e e \mid \text{cast}\{\tau \Leftarrow \tau\} e \mid \text{if } e \text{ then } e \text{ else } e$

$\tau ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid \tau \times \tau \mid \tau \rightarrow \tau \mid *$

$\text{binop} ::= \text{sum} \mid \text{quotient}$

$\Gamma ::= \cdot \mid \Gamma, (x:\tau)$

$n ::= \mathbb{N}$

$i ::= \mathbb{Z}$

$\Gamma \vdash_{\text{sim}} e : \tau$ typing

$\frac{\text{T-VAR}}{(x_0:\tau_0) \in \Gamma_0}}{\Gamma_0 \vdash_{\text{sim}} x_0 : \tau_0}$	$\frac{\text{T-NAT}}{}{\Gamma_0 \vdash_{\text{sim}} n_0 : \text{Nat}}$	$\frac{\text{T-INT}}{}{\Gamma_0 \vdash_{\text{sim}} i_0 : \text{Int}}$	$\frac{\text{T-TRUE}}{}{\Gamma_0 \vdash_{\text{sim}} \text{True} : \text{Bool}}$	$\frac{\text{T-FALSE}}{}{\Gamma_0 \vdash_{\text{sim}} \text{False} : \text{Bool}}$
$\frac{\text{T-LAM}}{\Gamma_0, (x_0:\tau_0) \vdash_{\text{sim}} e_0 : \tau_1}}{\Gamma_0 \vdash_{\text{sim}} \lambda(x_0:\tau_0). e_0 : \tau_0 \rightarrow \tau_1}$	$\frac{\text{T-PAIR}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{sim}} e_1 : \tau_1}}{\Gamma_0 \vdash_{\text{sim}} \langle e_0, e_1 \rangle : \tau_0 \times \tau_1}$	$\frac{\text{T-CAST}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0}}{\Gamma_0 \vdash_{\text{sim}} \text{cast}\{\tau_1 \Leftarrow \tau_0\} e_0 : \tau_1}$		
$\frac{\text{T-APP}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \rightarrow \tau_1 \quad \Gamma_0 \vdash_{\text{sim}} e_1 : \tau_0}}{\Gamma_0 \vdash_{\text{sim}} \text{app}\{\tau_1\} e_0 e_1 : \tau_1}$	$\frac{\text{T-FST}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \times \tau_1}}{\Gamma_0 \vdash_{\text{sim}} \text{fst}\{\tau_0\} e_0 : \tau_0}$	$\frac{\text{T-SND}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \times \tau_1}}{\Gamma_0 \vdash_{\text{sim}} \text{snd}\{\tau_1\} e_0 : \tau_1}$	$\frac{\text{T-BINOP}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{sim}} e_1 : \tau_1 \quad \Delta(\text{binop}, \tau_0, \tau_1) = \tau_2}}{\Gamma_0 \vdash_{\text{sim}} \text{binop } e_0 e_1 : \tau_2}$	
$\frac{\text{T-IF}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \text{Bool} \quad \Gamma_0 \vdash_{\text{sim}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\text{sim}} e_2 : \tau_0}}{\Gamma_0 \vdash_{\text{sim}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \tau_0}$			$\frac{\text{T-SUB}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_0 \quad \tau_0 \leq \tau_1}}{\Gamma_0 \vdash_{\text{sim}} e_0 : \tau_1}$	

$\tau \leq \tau$

$\frac{}{\text{Nat} \leq \text{Int}}$	$\frac{\tau_0 \leq \tau_2 \quad \tau_1 \leq \tau_3}{\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3}$	$\frac{\tau_2 \leq \tau_0 \quad \tau_1 \leq \tau_3}{\tau_0 \rightarrow \tau_1 \leq \tau_2 \rightarrow \tau_3}$	$\frac{}{\tau_0 \leq \tau_0}$
---------------------------------------	--	--	-------------------------------

625 $\Delta : \text{binop} \times \tau \times \tau \longrightarrow \tau$
626 $\Delta(\text{sum}, \text{Nat}, \text{Nat}) = \text{Nat}$
627 $\Delta(\text{sum}, \text{Int}, \text{Int}) = \text{Int}$
628 $\Delta(\text{quotient}, \text{Nat}, \text{Nat}) = \text{Nat}$
629 $\Delta(\text{quotient}, \text{Int}, \text{Int}) = \text{Int}$

630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676

3 Tag Typing

3.1 Definition

Simple language

$e ::= x \mid n \mid i \mid \text{True} \mid \text{False} \mid \lambda(x:K).e \mid \langle e, e \rangle \mid \text{app}\{K\} e e \mid \text{fst}\{K\} e \mid \text{snd}\{K\} e \mid \text{binop} e e \mid \text{cast}\{K \Leftarrow K\} e \mid \text{if } e \text{ then } e \text{ else } e$

$K ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid ** \mid * \rightarrow * \mid *$

$\text{binop} ::= \text{sum} \mid \text{quotient}$

$\Gamma ::= \cdot \mid \Gamma, (x:K_0)$

$n ::= \mathbb{N}$

$i ::= \mathbb{Z}$

$\Gamma \vdash_{\text{tag}} e : \tau$ typing

$$\frac{\text{T-VAR} \quad (x_0:K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\text{tag}} x_0 : K_0}$$

$$\frac{\text{T-NAT}}{\Gamma_0 \vdash_{\text{tag}} n_0 : \text{Nat}}$$

$$\frac{\text{T-INT}}{\Gamma_0 \vdash_{\text{tag}} i_0 : \text{Int}}$$

$$\frac{\text{T-TRUE}}{\Gamma_0 \vdash_{\text{tag}} \text{True} : \text{Bool}}$$

$$\frac{\text{T-FALSE}}{\Gamma_0 \vdash_{\text{tag}} \text{False} : \text{Bool}}$$

$$\frac{\text{T-LAM} \quad \Gamma_0, (x_0:K_0) \vdash_{\text{tag}} e_0 : K_1}{\Gamma_0 \vdash_{\text{tag}} \lambda(x_0:K_0).e_0 : * \rightarrow *}$$

$$\frac{\text{T-PAIR} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_1}{\Gamma_0 \vdash_{\text{tag}} \langle e_0, e_1 \rangle : **}$$

$$\frac{\text{T-CAST} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : K_0}{\Gamma_0 \vdash_{\text{tag}} \text{cast}\{K_1 \Leftarrow K_0\} e_0 : K_1}$$

$$\frac{\text{T-APP} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : * \rightarrow * \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_0}{\Gamma_0 \vdash_{\text{tag}} \text{app}\{K_1\} e_0 e_1 : K_1}$$

$$\frac{\text{T-FST} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : **}{\Gamma_0 \vdash_{\text{tag}} \text{fst}\{K_0\} e_0 : K_0}$$

$$\frac{\text{T-SND} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : **}{\Gamma_0 \vdash_{\text{tag}} \text{snd}\{K_1\} e_0 : K_1}$$

$$\frac{\text{T-BINOP} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_1 \quad \Delta(\text{binop}, K_0, K_1) = K_2}{\Gamma_0 \vdash_{\text{tag}} \text{binop} e_0 e_1 : K_2}$$

$$\frac{\text{T-IF} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : \text{Bool} \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_2 : K_0}{\Gamma_0 \vdash_{\text{tag}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : K_0}$$

$$\frac{\text{T-SUB} \quad \Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad K_0 \leqslant K_1}{\Gamma_0 \vdash_{\text{tag}} e_0 : K_1}$$

3.2 Simple Typing Implies Tag Typing

$$e^+$$

$$\begin{aligned}
 i^+ &= i \\
 b^+ &= b \\
 \langle e_1, e_2 \rangle^+ &= \langle e_1^+, e_2^+ \rangle \\
 (\lambda x : \tau. e)^+ &= \lambda x : [\tau]. e^+ \\
 (\text{app}\{\tau\} e_1 e_2)^+ &= \text{app}\{[\tau]\} e_1^+ e_2^+ \\
 (\text{fst}\{\tau\} e)^+ &= \text{fst}\{[\tau]\} e^+ \\
 (\text{snd}\{\tau\} e)^+ &= \text{snd}\{[\tau]\} e^+ \\
 (\text{binop } e_1 e_2)^+ &= \text{binop } e_1^+ e_2^+ \\
 (\text{cast } \{\tau' \Leftarrow \tau\} e)^+ &= \text{cast } \{[\tau'] \Leftarrow [\tau]\} e^+ \\
 (\text{if } e_1 \text{ then } e_2 \text{ else } e_3)^+ &= \text{if } e_1^+ \text{ then } e_2^+ \text{ else } e_3^+
 \end{aligned}$$

$$\Gamma^+$$

$$\begin{aligned}
 (\Gamma, x : \tau)^+ &= \Gamma^+, x : [\tau] \\
 \cdot^+ &= \cdot
 \end{aligned}$$

THEOREM 3.1 (SIMPLE TYPING IMPLIES TAG TYPING). *If $\Gamma \vdash_{\text{sim}} e : \tau$ then $\Gamma^+ \vdash_{\text{tag}} e^+ : [\tau]$.*

PROOF. By induction over the typing derivation. The typing rules have a one to one correspondance, so each case follows by the induction hypothesis. \square

4 Truer Transient Typing

4.1 Definition

Simple language

$e ::= x \mid n \mid i \mid \text{True} \mid \text{False} \mid \lambda(x:K). e \mid \langle e, e \rangle \mid \text{app}\{K\} e e \mid \text{fst}\{K\} e \mid \text{snd}\{K\} e \mid \text{binop } e e \mid \text{cast } \{K \Leftarrow K\} e \mid \text{if } e \text{ then } e \text{ else } e$

$\tau ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid \tau \times \tau \mid * \rightarrow \tau \mid * \mid \perp$

$K ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid * \times * \mid * \rightarrow * \mid *$

$\text{binop} ::= \text{sum} \mid \text{quotient}$

$\Gamma ::= \cdot \mid \Gamma, (x:K_0)$

$n ::= \mathbb{N}$

$i ::= \mathbb{Z}$

$[\tau]$ tag of

$[\text{Int}] = \text{Int}$

$[\text{Nat}] = \text{Nat}$

$[\text{Bool}] = \text{Bool}$

$[\tau \times \tau'] = * \times *$

$[* \rightarrow \tau'] = * \rightarrow *$

$[*] = *$

$\sqcup, \sqcap : \tau \times \tau \rightarrow \tau$

$$\tau \sqcup \tau' = \begin{cases} * & \text{if } \tau = * \\ & \text{or } \tau' = * \\ & \text{or } [\tau] \neq [\tau'] \\ & \quad \text{and } \tau \neq \perp \text{ and } \tau' \neq \perp \\ \tau & \text{if } \tau' = \perp \\ \tau' & \text{if } \tau = \perp \\ \text{Int} & \text{if } \tau = \text{Nat and } \tau' = \text{Int} \\ & \text{or } \tau = \text{Int and } \tau' = \text{Nat} \\ \tau & \text{if } \tau = \tau' \\ \tau_1 \sqcup \tau'_1 \times \tau_2 \sqcup \tau'_2 & \text{if } \tau = \tau_1 \times \tau_2 \text{ and } \tau' = \tau'_1 \times \tau'_2 \\ * \rightarrow (\tau_2 \sqcup \tau'_2) & \text{if } \tau = * \rightarrow \tau_2 \text{ and } \tau' = * \rightarrow \tau'_2 \end{cases}$$

$$\tau \sqcap \tau' = \begin{cases} \perp & \text{if } \tau = \perp \\ & \text{or } \tau' = \perp \\ & \text{or } [\tau] \neq [\tau'] \\ & \quad \text{and } \tau \neq * \text{ and } \tau' \neq * \\ \tau & \text{if } \tau' = * \\ \tau' & \text{if } \tau = * \\ \text{Nat} & \text{if } \tau = \text{Nat and } \tau' = \text{Int} \\ & \text{or } \tau = \text{Int and } \tau' = \text{Nat} \\ \tau & \text{if } \tau = \tau' \\ \tau_1 \sqcap \tau'_1 \times \tau_2 \sqcap \tau'_2 & \text{if } \tau = \tau_1 \times \tau_2 \text{ and } \tau' = \tau'_1 \times \tau'_2 \\ * \rightarrow (\tau_2 \sqcap \tau'_2) & \text{if } \tau = * \rightarrow \tau_2 \text{ and } \tau' = * \rightarrow \tau'_2 \end{cases}$$

$\Gamma \vdash_{\text{tru}} e : \tau$ typing

T-VAR	T-NAT	T-INT	T-TRUE	T-FALSE
$\frac{(x_0 : K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\text{tru}} x_0 : K_0}$	$\frac{}{\Gamma_0 \vdash_{\text{tru}} n_0 : \text{Nat}}$	$\frac{}{\Gamma_0 \vdash_{\text{tru}} i_0 : \text{Int}}$	$\frac{}{\Gamma_0 \vdash_{\text{tru}} \text{True} : \text{Bool}}$	$\frac{}{\Gamma_0 \vdash_{\text{tru}} \text{False} : \text{Bool}}$
T-LAM	T-PAIR		T-CAST	
$\frac{\Gamma_0, (x_0 : K_0) \vdash_{\text{tru}} e_0 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \lambda(x_0 : K_0). e_0 : * \rightarrow \tau_1}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \langle e_0, e_1 \rangle : \tau_0 \times \tau_1}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0}{\Gamma_0 \vdash_{\text{tru}} \text{cast} \{K_1 \Leftarrow K_0\} e_0 : K_1 \sqcap K_0 \sqcap \tau_0}$		
T-APP	T-APPBOT	T-FST	T-FSTBOT	
$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : * \rightarrow \tau_1 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau'_0}{\Gamma_0 \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 : K_1 \sqcap \tau_1}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \perp \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau'_0}{\Gamma_0 \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 : \perp}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \times \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 : K_0 \sqcap \tau_0}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \perp \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 : \perp}$	
T-SND	T-SNDBOT	T-BINOP		
$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \times \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 : K_1 \sqcap \tau_1}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \perp}{\Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 : \perp}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{binop} e_0 e_1 : \Delta(\text{binop}, \tau_0, \tau_1)}$		
T-IF	T-IFBOT	T-SUB		
$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \text{Bool} \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_2 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \tau_0 \sqcup \tau_1}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \perp \quad \Gamma_0 \vdash_{\text{tru}} e_1 : \tau_0 \quad \Gamma_0 \vdash_{\text{tru}} e_2 : \tau_1}{\Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \perp}$	$\frac{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_0 \quad \tau_0 \leq \tau_1}{\Gamma_0 \vdash_{\text{tru}} e_0 : \tau_1}$		

$\Delta : \text{binop} \times \tau \times \tau \rightarrow \tau$

$\Delta(\text{sum}, \text{Nat}, \text{Nat})$	$= \text{Nat}$
$\Delta(\text{sum}, \text{Int}, \text{Int})$	$= \text{Int}$
$\Delta(\text{quotient}, \text{Nat}, \text{Nat})$	$= \text{Nat}$
$\Delta(\text{quotient}, \text{Int}, \text{Int})$	$= \text{Int}$
$\Delta(\text{binop}, \perp, \tau)$	$= \perp$ if $\tau = \text{Nat}$ or Int or \perp
$\Delta(\text{binop}, \tau, \perp)$	$= \perp$ if $\tau = \text{Nat}$ or Int or \perp

$\tau \leq \tau$

$\frac{\tau_0 \leq \tau_1}{\tau_0 \leq \tau_1}$	$\frac{\tau_0 \leq \tau_2 \quad \tau_1 \leq \tau_3}{\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3}$	$\frac{\tau_0 \leq \tau_1}{* \rightarrow \tau_0 \leq * \rightarrow \tau_1}$	$\frac{}{\perp \leq \tau}$	$\frac{}{\tau \leq *}$
---	--	---	----------------------------	------------------------

4.2 Simple Typing Implies Truer Transient Typing

e^+

$$\begin{aligned}
i^+ &= i \\
b^+ &= b \\
\langle e_1, e_2 \rangle^+ &= \langle e_1^+, e_2^+ \rangle \\
(\lambda x : \tau. e)^+ &= \lambda x : [\tau]. e^+ \\
(\text{app}\{\tau\} e_1 e_2)^+ &= \text{app}\{[\tau]\} e_1^+ e_2^+ \\
(\text{fst}\{\tau\} e)^+ &= \text{fst}\{[\tau]\} e^+ \\
(\text{snd}\{\tau\} e)^+ &= \text{snd}\{[\tau]\} e^+ \\
(\text{binop } e_1 e_2)^+ &= \text{binop } e_1^+ e_2^+ \\
(\text{cast } \{\tau' \Leftarrow \tau\} e)^+ &= \text{cast } \{[\tau'] \Leftarrow [\tau]\} e^+ \\
(\text{if } e_1 \text{ then } e_2 \text{ else } e_3)^+ &= \text{if } e_1^+ \text{ then } e_2^+ \text{ else } e_3^+
\end{aligned}$$

Γ^+

$$\begin{aligned}
(\Gamma, x : \tau)^+ &= \Gamma^+, x : [\tau] \\
\cdot^+ &= \cdot
\end{aligned}$$

The following proofs will use the fact honest transient types with \sqcup and \sqcap form a lattice ordered by \leq .

LEMMA 4.1 (LATTICE JOIN IDEMPOTENT). $\tau \sqcup \tau = \tau$

PROOF. By induction on the structure of τ , in each case following immediately from the definition of \sqcup . \square

LEMMA 4.2 (LATTICE JOIN ABSORPTION). $\tau_0 \sqcup (\tau_0 \sqcap \tau_1) = \tau_0$

PROOF. By induction on the structure of τ_0 ; in each case by induction on the structure of τ_1 , in each case following immediately from the definitions of \sqcup and \sqcap and the prior lemma. \square

LEMMA 4.3 (LATTICE MEET IDEMPOTENT). $\tau \sqcap \tau = \tau$

PROOF. By induction on the structure of τ , in each case following immediately from the definition of \sqcap . \square

LEMMA 4.4 (LATTICE MEET ABSORPTION). $\tau_0 \sqcap (\tau_0 \sqcup \tau_1) = \tau_0$

PROOF. By induction on the structure of τ_0 ; in each case by induction on the structure of τ_1 , in each case following immediately from the definitions of \sqcup and \sqcap and the prior lemma. \square

LEMMA 4.5 (LATTICE ORDERING IMPLIES \leq). *If $\tau = \tau \sqcap \tau'$, then $\tau \leq \tau'$.*

PROOF. We proceed by induction on the structure of the definition of $\tau \sqcap \tau'$:

937 \perp Since $\tau = \tau \sqcap \tau$, $\tau = \perp$; it is immediate that $\tau_0 \leq \tau_1$.
 938 τ This case occurs if $\tau' = *$; consequently it is immediate that $\tau \leq \tau'$.
 939 τ' In this case, the hypothesis ensures that $\tau = \tau'$, so $\tau \leq \tau'$ by reflexivity.
 940 **Nat** In this case, τ must be **Nat** and τ' must be **Int**. By definition, **Nat** \leq **Int**.
 941 τ In this case, $\tau = \tau'$; it is immediate that $\tau \leq \tau'$.
 942 $\tau_1 \sqcap \tau'_1 \times \tau_2 \sqcap \tau'_2$ In this case, by the hypothesis, $\tau_1 = \tau_1 \sqcap \tau'_1$ and $\tau_2 = \tau_2 \sqcap \tau'_2$, so by induction $\tau_1 \leq \tau'_1$ and $\tau_2 \leq \tau'_2$. Then
 943 it is immediate from the definition of the lattice ordering that $\tau_1 \times \tau_2 \leq \tau'_1 \times \tau'_2$.
 944 $* \rightarrow \tau_2 \sqcap \tau'_2$ In this case, $\tau_2 = \tau_2 \sqcap \tau'_2$ by the hypothesis, so $\tau_2 \leq \tau'_2$ by induction; hence it is immediate from the definition
 945 of the lattice ordering that $* \rightarrow \tau_2 \leq * \rightarrow \tau'_2$.
 946
 947
 948
 949 □

950 **LEMMA 4.6 (LATTICE ORDERING IS IMPLIED BY \leq).** *If $\tau \leq \tau'$, then $\tau = (\tau \sqcap \tau')$.*

951 **PROOF.** We proceed by induction on the structure of the definition of \leq , with the cases of \leq : inlined:

952 **Nat \leq : Int** This is immediate by the definition of \sqcap .
 953 $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ This is subsumed by the case $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ below.
 954 $\tau_0 \rightarrow \tau_1 \leq \tau_2 \rightarrow \tau_3$ Because we are considering the lattice of honest transient types, $\tau_0 = \tau_2 = *$, and this is subsumed
 955 by the case $* \rightarrow \tau_1 \leq * \rightarrow \tau_3$ below.
 956 $\tau_0 \leq \tau_0$ This is immediate by the definition of \sqcap .
 957 $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ This rule requires that $\tau_0 \leq \tau_2$ and $\tau_1 \leq \tau_3$; hence, by induction $\tau_0 = \tau_0 \sqcap \tau_2$ and $\tau_1 = \tau_1 \sqcap \tau_3$. This is
 958 then immediate by the definition of \sqcap .
 959 $* \rightarrow \tau_1 \leq * \rightarrow \tau_3$ This rule requires that $\tau_0 \leq \tau_1$, and so by induction $\tau_0 = \tau_0 \sqcap \tau_1$; this is then immediate by the definition
 960 of \sqcap .
 961 $\perp \leq \tau$ This is immediate by the definition of \sqcap .
 962 $\tau \leq *$ This is immediate by the definition of \sqcap .
 963
 964
 965
 966
 967 □

968
 969 **THEOREM 4.7 (SIMPLE TYPING IMPLIES TRUER TRANSIENT TYPING).**

970 *If $\Gamma \vdash_{\text{sim}} e : \tau$ then $\Gamma^+ \vdash_{\text{tru}} e^+ : \tau'$ where $\tau' \leq \lfloor \tau \rfloor$.*

971 **PROOF.** Proceed by induction on the simple typing derivation:

972
 973 **T-Var** By the definition of lowering, if $x : \tau \in \Gamma$, then $x : \lfloor \tau \rfloor \in \Gamma^+$, so **T-Var** applies and $\lfloor \tau \rfloor$ is precisely the τ' such that
 974 $\Gamma^+ \vdash e^+ : \tau'$ and $\tau' \leq \lfloor \tau \rfloor$.
 975
 976 **T-Nat, T-Int, T-True, T-False** For each base type literal, a corresponding rule exists in the honest transient type
 977 system, which ascribes the same time (which is also equal to, and hence below in the lattice, the original simple
 978 type).
 979
 980 **T-Lam** Consider arbitrary $\Gamma_0, x_0, \tau_0, e_0, \tau_1$, such that $\Gamma_0 \vdash \lambda(x_0 : \tau_0). e_0 : \tau_0 \rightarrow \tau_1$. Then by induction we know that
 981 $(\Gamma_0, (x_0) : \tau_0)^+ \vdash e_0^+ : \tau'_1$, for some $\tau'_1 \leq \lfloor \tau_1 \rfloor$. Note that $(\Gamma_0, (x_0) : \tau_0)^+ = \Gamma_0^+, x_0 : \lfloor \tau_0 \rfloor$ by definition, and similarly
 982 that $(\lambda x_0 : \tau_0. e_0)^+ = \lambda(x_0 : \lfloor \tau_0 \rfloor). e_0^+$ by definition. Then **T-Lam** applies s.t. $\Gamma_0^+ \vdash \lambda(x_0 : K_0). e_0^+ : * \rightarrow \tau'_1$. Note that
 983 $\lfloor \tau_0 \rightarrow \tau_1 \rfloor = * \rightarrow * \leq * \rightarrow *$ by the definition of lattice ordering, completing the proof.
 984
 985 **T-Pair** Consider arbitrary $\Gamma_0, e_0, e_1, \tau_0, \tau_1$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule **T-Pair** if $e = \langle e_0, e_1 \rangle$ and $\tau = \tau_0 \times \tau_1$. Then
 986 by induction, there exist some τ'_0 and τ'_1 , s.t. $\Gamma_0^+ \vdash e_0^+ : \tau'_0, \Gamma_0^+ \vdash e_1^+ : \tau'_1, \tau'_0 \leq \lfloor \tau_0 \rfloor$, and $\tau'_1 \leq \lfloor \tau_1 \rfloor$. Then instantiate
 987
 988

989 $\tau' = \tau_0 \times \tau_1$; it is clear that the honest transient typing rule T-Pair applies, since $(\langle e_0, e_1 \rangle)^+ = \langle e_0^+, e_1^+ \rangle$, and it is
990 immediate by the definition of \leq that $\tau' \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$.

991 **T-Cast** Consider arbitrary $\Gamma_0, e_0, \tau_0, \tau_1$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Cast if $e = \text{cast} \{ \tau_0 \Leftarrow \tau_1 \} e_0$ and $\tau = \tau_1$.
992 Then by induction, $\Gamma_0^+ \vdash e_0^+ : \tau'_0$ for some τ'_0 s.t. $\tau'_0 \leq \lfloor \tau_0 \rfloor$. Instantiate τ' by $\lfloor \tau_1 \rfloor \sqcap \lfloor \tau_0 \rfloor \sqcap \tau'_0$; then it is clear that
993 the honest transient typing rule T-Cast applies, since by definition $e^+ = \text{cast} \{ \lfloor \tau_0 \rfloor \Leftarrow \lfloor \tau_1 \rfloor \} e_0^+$. It remains to be
994 shown that $\lfloor \tau_1 \rfloor \sqcap \lfloor \tau_0 \rfloor \sqcap \tau'_0 \leq \lfloor \tau_1 \rfloor$; this follows immediately from the properties of the lattice meet operation.

995 **T-App** Consider arbitrary $\Gamma_0, e_0, \tau_0, \tau_1$ s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-App if $e = \text{app} \{ \tau_1 \} e_0 e_1$ and $\tau = \tau_1$. Then
996 by induction, $\Gamma_0^+ \vdash e_0^+ : \tau'_l$ for some $\tau'_l \leq \lfloor \tau_0 \rightarrow \tau_1 \rfloor = * \rightarrow *$, and $\Gamma_0^+ \vdash e_1^+ : \tau'_r$ for some $\tau'_r \leq \lfloor \tau_0 \rfloor$. By inspection
997 of \leq , note that τ'_l must be either \perp or $* \rightarrow \tau'_l$ for some τ'_l . Note that $e^+ = \text{app} \{ \lfloor \tau_1 \rfloor \} e_0^+ e_1^+$, and so in the former
998 case T-AppBot syntactically applies and in the latter T-App; consider each case:
999 $\tau'_l = \perp$: Instantiate $\tau' = \perp$; then it is clear that $\Gamma_0^+ \vdash e' : \tau'$ by T-AppBot. Then $\perp \leq \lfloor \tau_1 \rfloor$ is immediate by the
1000 definition of lattice ordering.
1001 $\tau'_l = * \rightarrow \tau'_l$: Instantiate $\tau' = \lfloor \tau_1 \rfloor \sqcap \tau'_l$; then it is clear that $\Gamma_0^+ \vdash e' : \tau'$ by T-App, so what remains to be shown is
1002 that $\lfloor \tau_1 \rfloor \sqcap \tau'_l \leq \lfloor \tau_1 \rfloor$; this is immediate by the definition of meet on a lattice.

1003 **T-Fst** Consider arbitrary $\Gamma_0, e_0, \tau_0, \tau_1$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Fst with premise $\Gamma_0 \vdash e_0 : \tau_0 \times \tau_1$ if
1004 $e = \text{fst} \{ \tau_0 \} e_0$ and $\tau = \tau_0$. Then, by induction, $\Gamma_0^+ \vdash e : \tau'_p$ s.t. $\tau'_p \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$. By inspection on \leq , note that
1005 τ'_p must be either \perp or $\tau_{p_0}' \times \tau_{p_1}'$ for some τ_{p_0}' and τ_{p_1}' . Since $e^+ = \text{fst} \{ \lfloor \tau_0 \rfloor \} e_0^+$, the rule T-FstBot applies in the
1006 former case, and similarly T-Fst applies in the latter. Consider each of these cases:
1007 $\tau'_p = \perp$: Instantiate $\tau' = \perp$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-FstBot, and $\perp \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1008 lattice ordering.
1009 $\tau'_p = \tau_{p_0}' \times \tau_{p_1}'$: Instantiate τ' with $\lfloor \tau_0 \rfloor \sqcap \tau_{p_0}'$. Then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-Fst, and $\tau' \leq \lfloor \tau_0 \rfloor$ by the the definition of
1010 meet on a lattice.

1011 **T-Snd** Consider arbitrary $\Gamma_0, e_0, \tau_0, \tau_1$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Snd with premise $\Gamma_0 \vdash e_0 : \tau_0 \times \tau_1$ if
1012 $e = \text{snd} \{ \tau_1 \} e_0$ and $\tau = \tau_1$. Then, by induction, $\Gamma_0^+ \vdash e : \tau'_p$ s.t. $\tau'_p \leq \lfloor \tau_0 \times \tau_1 \rfloor = * \times *$. By inspection on \leq , note
1013 that τ'_p must be either \perp or $\tau_{p_0}' \times \tau_{p_1}'$ for some τ_{p_0}' and τ_{p_1}' . Since $e^+ = \text{snd} \{ \lfloor \tau_1 \rfloor \} e_0^+$, the rule T-SndBot applies
1014 in the former case, and similarly T-Snd applies in the latter. Consider each of these cases:
1015 $\tau'_p = \perp$: Instantiate $\tau' = \perp$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-SndBot, and $\perp \leq \lfloor \tau_1 \rfloor$ follows immediately from the definition of
1016 lattice ordering.
1017 $\tau'_p = \tau_{p_0}' \times \tau_{p_1}'$: Instantiate τ' with $\lfloor \tau_1 \rfloor \sqcap \tau_{p_1}'$. Then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-Snd, and $\tau' \leq \lfloor \tau_1 \rfloor$ by the the definition of
1018 meet on a lattice.

1019 **T-Binop** Consider arbitrary $\Gamma_0, \text{binop}, e_0, e_1, \tau_0, \tau_1$, and τ_2 , s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Binop with premise
1020 $\Delta(\text{binop}, \tau_0, \tau_1) = \tau_2$ if $e = \text{binop} e_0 e_1$ and $\tau = \tau_2$. By induction, note that $\Gamma_0^+ \vdash e_0^+ : \tau'_0$ for some $\tau'_0 \leq \lfloor \tau_0 \rfloor$, and
1021 $\Gamma_0^+ \vdash e_1^+ : \tau'_1$ for some $\tau'_1 \leq \lfloor \tau_1 \rfloor$. Note that for the simple typing $\Delta(\text{binop}, \tau_0, \tau_1)$ to be defined, τ_0 and τ_1 must
1022 each be either Nat or Int; consequently, by inspection of the lattice order, τ'_0 and τ'_1 must each be Nat, Int, or \perp .
1023 Then by inspection, in any such case, $\Delta(\text{binop}, \tau'_0, \tau'_1)$ is defined and $\leq \Delta(\text{binop}, \tau_0, \tau_1) = \tau_2$. Then instantiate τ'
1024 with $\lfloor \tau_2 \rfloor \sqcap \Delta(\text{binop}, \tau'_0, \tau'_1)$; since $e^+ = \text{binop} e_0 e_1$, the rule S-Binop applies, and by the definition of meet on a
1025 lattice, $\lfloor \tau_2 \rfloor \leq \tau'$.

1026 **T-If** Consider arbitrary $\Gamma_0, e_0, e_1, e_2, \tau_0$, s.t. $\Gamma_0 \vdash$ if e_1 then e_2 else $e_3 : \tau_0$ by the T-If simple typing rule. Let
1027 $e = \text{if } e_1 \text{ then } e_2 \text{ else } e_3$ and $\tau = \tau_0$. Then by induction, there exist some $\tau'_b \leq \lfloor \text{Bool} \rfloor = \text{Bool}$, $\tau'_0 \leq \lfloor \tau_0 \rfloor$, and
1028 $\tau'_1 \leq \lfloor \tau_0 \rfloor$, s.t. $\Gamma_0^+ \vdash e_0^+ : \tau'_b$, $\Gamma_0^+ \vdash e_1^+ : \tau'_0$, and $\Gamma_0^+ \vdash e_2^+ : \tau'_1$. Notice that τ'_b may be only \perp or Bool, by the definition
1029 of \leq .
1030 $\tau'_b = \perp$: Instantiate $\tau' = \perp$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-IfBot, and $\perp \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1031 lattice ordering.
1032 $\tau'_b = \text{Bool}$: Instantiate $\tau' = \text{Bool} \sqcap \tau'_0$; then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-If, and $\tau' \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1033 meet on a lattice.
1034 $\tau'_b = \text{Bool}$: Instantiate $\tau' = \text{Bool} \sqcap \tau'_1$; then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-If, and $\tau' \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1035 meet on a lattice.

1036 **T-If** Consider arbitrary $\Gamma_0, e_0, e_1, e_2, \tau_0$, s.t. $\Gamma_0 \vdash$ if e_1 then e_2 else $e_3 : \tau_0$ by the T-If simple typing rule. Let
1037 $e = \text{if } e_1 \text{ then } e_2 \text{ else } e_3$ and $\tau = \tau_0$. Then by induction, there exist some $\tau'_b \leq \lfloor \text{Bool} \rfloor = \text{Bool}$, $\tau'_0 \leq \lfloor \tau_0 \rfloor$, and
1038 $\tau'_1 \leq \lfloor \tau_0 \rfloor$, s.t. $\Gamma_0^+ \vdash e_0^+ : \tau'_b$, $\Gamma_0^+ \vdash e_1^+ : \tau'_0$, and $\Gamma_0^+ \vdash e_2^+ : \tau'_1$. Notice that τ'_b may be only \perp or Bool, by the definition
1039 of \leq .
1040 $\tau'_b = \perp$: Instantiate $\tau' = \perp$; $\Gamma_0^+ \vdash e^+ : \tau'$ by T-IfBot, and $\perp \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1041 lattice ordering.
1042 $\tau'_b = \text{Bool}$: Instantiate $\tau' = \text{Bool} \sqcap \tau'_0$; then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-If, and $\tau' \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1043 meet on a lattice.
1044 $\tau'_b = \text{Bool}$: Instantiate $\tau' = \text{Bool} \sqcap \tau'_1$; then $\Gamma_0^+ \vdash e^+ : \tau'$ by T-If, and $\tau' \leq \lfloor \tau_0 \rfloor$ follows immediately from the definition of
1045 meet on a lattice.

of lattice ordering. Since $e^+ = \text{if } e_0^+ \text{ then } e_1^+ \text{ else } e_2^+$, in the former case the rule T-IfBot applies; in the latter the rule T-If applies. Consider each of these cases:

$\tau'_b = \perp$: By T-IfBot, $\Gamma_0^+ \vdash e^+ : \perp$, so instantiate $\tau' = \perp$. Notice then that $\perp \leq \lfloor \tau \rfloor$ by lattice ordering, so the proof is completed.

$\tau'_b = \text{Bool}$: By T-If, $\Gamma_0^+ \vdash e^+ : \tau'_0 \sqcup \tau'_1$. Instantiate τ' by $\tau'_0 \sqcup \tau'_1$; then we must show that $\tau' \leq \lfloor \tau \rfloor$. Since $\tau'_0 \leq \lfloor \tau_0 \rfloor$ and $\tau'_1 \leq \lfloor \tau_0 \rfloor$, $\lfloor \tau_0 \rfloor$ is an upper bound of τ'_0 and τ'_1 . By the definition of join on a lattice, $\tau'_0 \sqcup \tau'_1$ is less-than-or-equal-to any other upper bound of τ_0 and τ_1 , so this is shown.

T-Sub Consider arbitrary $\Gamma_0, e_0, \tau_1, \tau_0$, s.t. $\Gamma_0 \vdash e : \tau$ by simple typing rule T-Sub with premise $\tau_0 \leq \tau_1$ if $e = e_0$ and $\tau = \tau_1$. By induction, $\Gamma_0 \vdash e^+ : \tau'_0$ for some $\tau'_0 \leq \lfloor \tau_0 \rfloor$. Then instantiate $\tau' = \tau'_0$. It is immediate that $\Gamma_0 \vdash e^+ : \tau'$; it remains to be shown that $\tau' \leq \lfloor \tau_1 \rfloor$. Since $\tau_0 \leq \tau_1$, $\tau_0 \leq \tau_1$. By Lemma 4.8, $\lfloor \tau_0 \rfloor \leq \lfloor \tau_1 \rfloor$. Then by Lemma 4.9, $\tau' = \tau'_0 \leq \lfloor \tau_0 \rfloor \leq \lfloor \tau_1 \rfloor$ so $\tau' \leq \lfloor \tau_1 \rfloor$.

□

LEMMA 4.8 (LATTICE ORDERING IS PRESERVED BY TAG-OF). *If $\tau_0 \leq \tau_1$, then $\lfloor \tau_0 \rfloor \leq \lfloor \tau_1 \rfloor$.*

PROOF. By cases on the structure of the definition of \leq ; in each case the lemma is immediate. □

LEMMA 4.9 (LATTICE ORDERING IS TRANSITIVE). *If $\tau \leq \tau'$ and $\tau' \leq \tau''$, then $\tau \leq \tau''$.*

PROOF. By induction on the structure of the definition of $\tau \leq \tau'$ (generalized with respect to τ''), with the cases of \leq : inlined:

Nat \leq : Int: Since by assumption $\text{Int} \leq \tau''$, it is clear by inspection that τ'' must be either Int or $*$; in either case Nat $\leq \tau''$ is immediate.

$\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$: This is subsumed by the case $\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$ below.

$\tau_0 \rightarrow \tau_1 \leq \tau_2 \rightarrow \tau_3$: Because we are considering the lattice of honest transient types, $\tau_0 = \tau_2 = *$, and this is subsumed by the case $* \rightarrow \tau_1 \leq * \rightarrow \tau_3$ below.

$\tau \leq \tau$: Since by assumption $\tau' \leq \tau''$, $\tau = \tau' \leq \tau_2$.

$\tau_0 \times \tau_1 \leq \tau_2 \times \tau_3$: Since by assumption $\tau' = \tau_2 \times \tau_3 \leq \tau''$, it is clear that τ'' must be either $*$ or $\tau''_0 \times \tau''_1$ for some τ''_0, τ''_1 s.t. $\tau_2 \leq \tau''_0$ and $\tau_3 \leq \tau''_1$. If τ'' is $*$, the lemma follows immediately. Otherwise, note that this rule requires that $\tau_0 \leq \tau_2$ and $\tau_1 \leq \tau_3$; hence, by induction, $\tau_0 \leq \tau''_0$ and $\tau_1 \leq \tau''_1$, and therefore $\tau \leq \tau''$.

$* \rightarrow \tau_1 \leq * \rightarrow \tau_3$: Since by assumption $\tau' = * \rightarrow \tau_3 \leq \tau''$, it is clear that τ'' must be either $*$ or $* \rightarrow \tau''_1$ for some τ''_1 s.t. $\tau_3 \leq \tau''_1$. If τ'' is $*$, the lemma follows immediately. Otherwise, note that this rule requires that $\tau_1 \leq \tau_3$; hence, by induction, $\tau_1 \leq \tau''_1$, and therefore $\tau \leq \tau''$.

$\perp \leq \tau$ $\tau = \perp \leq \tau''$ is immediate by the definition of lattice ordering.

$\tau \leq *$ Since by assumption $\tau' = * \leq \tau''$, τ'' must be $*$, and so the lemma follows immediately.

□

4.3 Tag Typing Implies Truer Transient Typing

THEOREM 4.10 (TAG TYPING IMPLIES TRUER TRANSIENT TYPING). *If $\Gamma \vdash_{\text{tag}} e : K$ then $\exists \tau \leq K$ such that $\Gamma \vdash_{\text{tru}} e : \tau$.*

PROOF. By induction over the tag typing derivation.

$$\begin{array}{c}
\text{T-VAR} \\
\frac{(x_0 : K_0) \in \Gamma_0}{\Gamma_0 \vdash_{\text{tag}} x_0 : K_0}
\end{array}
\quad
\begin{array}{c}
\text{T-NAT} \\
\frac{}{\Gamma_0 \vdash_{\text{tag}} n_0 : \text{Nat}}
\end{array}
\quad
\begin{array}{c}
\text{T-INT} \\
\frac{}{\Gamma_0 \vdash_{\text{tag}} i_0 : \text{Int}}
\end{array}
\quad
\begin{array}{c}
\text{T-TRUE} \\
\frac{}{\Gamma_0 \vdash_{\text{tag}} \text{True} : \text{Bool}}
\end{array}
\quad
\begin{array}{c}
\text{T-FALSE} \\
\frac{}{\Gamma_0 \vdash_{\text{tag}} \text{False} : \text{Bool}}
\end{array}$$

These cases are immediate by applying the corresponding truer typing rule and from premises.

$$\begin{array}{c}
\text{T-LAM} \\
\frac{\Gamma_0, (x_0 : K_0) \vdash_{\text{tag}} e_0 : K_1}{\Gamma_0 \vdash_{\text{tag}} \lambda(x_0 : K_0). e_0 : * \rightarrow *}
\end{array}
\quad
\begin{array}{c}
\text{T-PAIR} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_1}{\Gamma_0 \vdash_{\text{tag}} \langle e_0, e_1 \rangle : * \times *}
\end{array}
\quad
\begin{array}{c}
\text{T-IF} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : \text{Bool} \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_2 : K_0}{\Gamma_0 \vdash_{\text{tag}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : K_0}
\end{array}
\quad
\begin{array}{c}
\text{T-SUB} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad K_0 \leq K_1}{\Gamma_0 \vdash_{\text{tag}} e_0 : K_1}
\end{array}$$

These cases follows by the induction hypothesis and the corresponding rule.

$$\begin{array}{c}
\text{T-APP} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : * \rightarrow * \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_0}{\Gamma_0 \vdash_{\text{tag}} \text{app}\{K_1\} e_0 e_1 : K_1}
\end{array}
\quad
\begin{array}{c}
\text{T-FST} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : * \times *}{\Gamma_0 \vdash_{\text{tag}} \text{fst}\{K_0\} e_0 : K_0}
\end{array}
\quad
\begin{array}{c}
\text{T-SND} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : * \times *}{\Gamma_0 \vdash_{\text{tag}} \text{snd}\{K_1\} e_0 : K_1}
\end{array}$$

These cases follow by induction and their corresponding typing rule, with the caveat that if the truer type of the premise is \perp , the corresponding bot rule must be used.

$$\begin{array}{c}
\text{T-CAST} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad K_0 \sim K_1}{\Gamma_0 \vdash_{\text{tag}} \text{cast}\{K_1 \Leftarrow K_0\} e_0 : K_1}
\end{array}$$

This case follows by induction and applying the bnd rule in truer, noting truer doesn't require any relationships between the type of what's underneath and the tags on the bnds.

$$\begin{array}{c}
\text{T-BINOP} \\
\frac{\Gamma_0 \vdash_{\text{tag}} e_0 : K_0 \quad \Gamma_0 \vdash_{\text{tag}} e_1 : K_1 \quad \Delta(\text{binop}, K_0, K_1) = K_2}{\Gamma_0 \vdash_{\text{tag}} \text{binop } e_0 e_1 : K_2}
\end{array}$$

This case follows by induction, noting that if either of the truer types corresponding to K_0 or K_1 are \perp , then the result type is \perp . If the truer types are different, ie one is Nat and the other Int, we apply subsumption to get both at Int, and then can apply the binop rule. Otherwise, we directly apply the binop rule.

□

1145 5 Vigilance

1146 5.1 Vigilance Logical Relation

1147 $\llbracket \Gamma \vdash_t e : \tau \rrbracket^L \triangleq \forall (k, \Psi, \Sigma, \gamma) \in \mathcal{G}^L \llbracket \Gamma \rrbracket$ where $\Sigma : (k, \Psi). (k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^L \llbracket \tau \rrbracket$

1148 $\mathcal{G}^L \llbracket \Gamma, x : \tau \rrbracket \triangleq \{(k, \Psi, \Sigma, \gamma[x \mapsto \ell]) \mid (k, \Psi, \Sigma, \gamma) \in \mathcal{G}^L \llbracket \Gamma \rrbracket$
 1149 $\wedge \ell \in \text{dom}(\Psi) \wedge \ell \notin \text{dom}(\gamma)$
 1150 $\wedge (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \tau \rrbracket_k\}$

1151 $\mathcal{G}^L \llbracket \bullet \rrbracket \triangleq \{(k, \Psi, \Sigma, \emptyset)\}$

1152 $\vdash \Sigma \triangleq \forall \ell \in \text{dom}(\Sigma). \Sigma(\ell) = ((\ell', \text{some}(\tau', \tau)) \wedge \tau' \propto \text{pointsto}(\Sigma, \ell) \wedge \tau \propto \text{pointsto}(\Sigma, \ell)$
 1153 $\wedge \neg * \times * \propto \text{pointsto}(\Sigma, \ell))$

1154 $\vee \Sigma(\ell) = (v, \text{none})$ where $v \notin \mathbb{L}$

1155 $\Sigma : (k, \Psi) \triangleq \text{dom}(\Sigma) = \text{dom}(\Psi) \wedge \vdash \Sigma \wedge \forall j < k, \ell \in \text{dom}(\Sigma). ((j, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \Psi(\ell) \rrbracket$
 1156 $\wedge (\Sigma(\ell) = (\ell', \text{some}(\tau, \tau')) \Rightarrow \Psi(\ell) = [\tau, \tau', \Psi(\ell')] \wedge \Psi(\ell') = [\tau'', \dots] \wedge \tau'' <: \tau')$
 1157 $\wedge (\Sigma(\ell) = (v, \text{none}) \wedge v \notin \mathbb{L} \Rightarrow \exists \tau. \Psi(\ell) = [\tau])$

1171 This is an unfolded version of the definition in the paper. We break up the definition there for ease of explanation, and
 1172 unfold here for ease of use.

1173 $(j, \Psi) \sqsupseteq (k, \Psi) \triangleq j \leq k \wedge \forall \ell \in \text{dom}(\Psi). \Psi'(\ell) = \Psi(\ell)$

1174 $\mathcal{E}^L \llbracket \bar{\tau} \rrbracket \triangleq \{(k, \Psi, \Sigma, e) \mid \forall j \leq k. \forall \Sigma' \supseteq \Sigma, e'. (\Sigma, e) \xrightarrow{J}_L^j (\Sigma', e') \wedge \text{irred}(e')$
 1175 $\Rightarrow (e' = \text{Err}^\bullet \vee (\exists (k-j, \Psi') \sqsupseteq (k, \Psi). \Sigma' : (k-j, \Psi') \wedge (k-j, \Psi', \Sigma', e') \in \mathcal{V}^L \llbracket \bar{\tau} \rrbracket))\}$

1176 $\mathcal{V}^L \llbracket [\text{Int}, \tau_2, \dots, \tau_n] \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\text{Int}, \tau_2, \dots, \tau_n]. (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \tau \rrbracket\}$

1177 $\mathcal{V}^L \llbracket [\text{Nat}, \tau_2, \dots, \tau_n] \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\text{Nat}, \tau_2, \dots, \tau_n]. (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \tau \rrbracket\}$

1178 $\mathcal{V}^L \llbracket [\text{Bool}, \tau_2, \dots, \tau_n] \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall \tau \in [\text{Bool}, \tau_2, \dots, \tau_n]. (k, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \tau \rrbracket\}$

1197

1198

1199

$$\mathcal{VH}^L \llbracket \tau_1' \times \tau_1'', \tau_2, \dots, \tau_n \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)\}$$

1200

$$\wedge (k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^L \llbracket \tau_1', fst(\tau_2), \dots, fst(\tau_n) \rrbracket$$

1201

$$\wedge (k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^L \llbracket \tau_1'', snd(\tau_2), \dots, snd(\tau_n) \rrbracket\}$$

1202

1203

1204

1205

1206

1207

$$\mathcal{VH}^L \llbracket \tau_1' \rightarrow \tau_1'', \tau_2, \dots, \tau_n \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi), \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi')\}$$

1208

$$\forall \tau_0 \text{ where } cod(\tau_1') \leq \tau_0. \forall \ell_0 \text{ where } (j, \Psi', \Sigma', \ell_0) \in \mathcal{V}^L \llbracket \tau_1' \rrbracket.$$

1209

$$(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_0) \in \mathcal{EH}^L \llbracket [\tau_0, cod(\tau_2), \dots, cod(\tau_n)] \rrbracket\}$$

1210

1211

1212

1213

$$\mathcal{VH}^L \llbracket *, \tau_2, \dots, \tau_n \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L \llbracket \text{Int}, \tau_2, \dots, \tau_n \rrbracket$$

1214

$$(k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L \llbracket \text{Bool}, \tau_2, \dots, \tau_n \rrbracket$$

1215

$$\vee (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L \llbracket * \times *, \tau_2, \dots, \tau_n \rrbracket$$

1216

$$\vee (k-1, \Psi, \Sigma, \ell) \in \mathcal{VH}^L \llbracket * \rightarrow *, \tau_2, \dots, \tau_n \rrbracket\}$$

1217

1218

1219

1220

1221

1222

1223

1224

$$\mathcal{E}^L \llbracket \tau \rrbracket \triangleq \{(k, \Psi, \Sigma, e) \mid \forall j \leq k. \forall \Sigma' \supseteq \Sigma, e'. (\Sigma, e) \xrightarrow{j} (\Sigma', e') \wedge \text{irred}(e')\}$$

1225

$$\Rightarrow (e' = \text{Err}^\bullet \vee (\exists (k-j, \Psi') \sqsupseteq (k, \Psi). \Sigma' : (k-j, \Psi') \wedge (k-j, \Psi', \Sigma', e') \in \mathcal{V}^L \llbracket \tau \rrbracket))\}$$

1226

1227

1228

1229

1230

$$\mathcal{V}^L \llbracket \text{Int} \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell \mid \text{pointsto}(\Sigma, \ell) \in \mathbb{Z}\}$$

1231

1232

1233

$$\mathcal{V}^L \llbracket \text{Nat} \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell \mid \text{pointsto}(\Sigma, \ell) \in \mathbb{N}\}$$

1234

1235

1236

$$\mathcal{V}^L \llbracket \text{Bool} \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell \mid \text{pointsto}(\Sigma, \ell) \in \mathbb{B}\}$$

1237

1238

1239

$$\mathcal{V}^L \llbracket \tau_1 \times \tau_2 \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _) \wedge (k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^L \llbracket \tau_1 \rrbracket \wedge (k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^L \llbracket \tau_2 \rrbracket\}$$

1240

1241

1242

1243

$$\mathcal{V}^L \llbracket \tau_1 \rightarrow \tau_2 \rrbracket \triangleq \{(k, \Psi, \Sigma, \ell) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi). \forall \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi')\}$$

1244

$$\forall \ell_0 \text{ where } (j, \Psi', \Sigma', \ell_0) \in \mathcal{V}^L \llbracket \tau_1 \rrbracket. \forall \tau_0. \text{ where } \tau_2 \leq \tau_0$$

1245

$$(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_0) \in \mathcal{E}^L \llbracket \tau_0 \rrbracket\}$$

1246

1247

1248

1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1298
 1299
 1300

$$\begin{aligned} \mathcal{V}^L[\![*\!]\!] \triangleq \{ & (k, \Psi, \Sigma, \ell) \mid (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\text{Int}]\!] \\ & (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![\text{Bool}]\!] \\ & \forall (k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^L[\![* \times *]\!] \\ & \forall (k-1, \Psi \ell \Sigma, \ell) \in \mathcal{V}^L[\![* \rightarrow *]\!] \} \end{aligned}$$

5.2 Vigilance Theorem

$\Gamma \vdash_t e : \tau$ then $\llbracket \Gamma \vdash_t e : \sigma \rrbracket^L$

1301 5.3 Vigilance Fundamental Property for Natural with Simple Typing

1302

1303

In this subsection, we use $\Gamma \vdash e : \tau$ to mean $\Gamma \vdash_{\text{sim}} e : \tau$.

1304

1305

5.3.1 Lemmas Used Without Mention

1306

1307

1308

LEMMA 5.1 (STEPPING TO ERROR IMPLIES EXPRESSION RELATION). *If $(\Sigma, e) \rightarrow_N^j (\Sigma', \text{Err}^\bullet)$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$*

1309

PROOF. If $k < j$, then we're done because the condition in the expression relation is vacuously true.

1310

1311

Otherwise, we can use j as our steps, Σ' as our ending value log, and Err^\bullet as our irreducible expression, and we satisfy the condition in the expression relation. \square

1313

1314

1315

LEMMA 5.2 (STEPPING TO ERROR IMPLIES EXPRESSION HISTORY RELATION). *If $(\Sigma, e) \rightarrow_N^j (\Sigma', \text{Err}^\bullet)$ then $(k, \Psi, \Sigma, e) \in \mathcal{EH}^N \llbracket \bar{\tau} \rrbracket$*

1316

1317

PROOF. Similar to the previous proof. \square

1318

1319

1320

LEMMA 5.3 (ANTI-REDUCTION - HEAD EXPANSION - EXPRESSION RELATION COMMUTES WITH STEPS). *If $(k, \Psi', \Sigma', e') \in \mathcal{E}^N \llbracket \tau \rrbracket$ and $(\Sigma, e) \rightarrow_N^j (\Sigma', e')$ and $\Sigma' : (k, \Psi')$ then $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$*

1321

1322

PROOF. Unfolding the expression relation in our hypothesis, there exists (Σ'', e'') , j' such that $(\Sigma', e') \rightarrow_N^{j'} (\Sigma'', e'')$ and (Σ'', e'') is irreducible.

1324

1325

1326

Either $e'' = \text{Err}^\bullet$, in which case $(\Sigma, e) \rightarrow_N^{j+j'} (\Sigma'', \text{Err}^\bullet)$, so we're done.

Otherwise, there is a $(k - j', \Psi'') \sqsupseteq (k, \Psi')$ such that $\Sigma'' : (k - j', \Psi'')$, and $(k - j', \Psi'', \Sigma'', e'') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

1327

Using this information, we can show $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ by noting $(\Sigma, e) \rightarrow_N^{j+j'} (\Sigma'', e'')$. \square

1328

1329

1330

LEMMA 5.4 (ANTI-REDUCTION - HEAD EXPANSION - EXPRESSION HISTORY COMMUTES WITH STEPS). *If $(k, \Psi', \Sigma', e') \in \mathcal{EH}^N \llbracket \bar{\tau} \rrbracket$ and $(\Sigma, e) \rightarrow_N^j (\Sigma', e')$ and $\Sigma' : (k, \Psi')$ then $(k + j, \Psi, \Sigma, e) \in \mathcal{EH}^N \llbracket \bar{\tau} \rrbracket$*

1331

1332

PROOF. Similar to the previous proof. \square

1333

1334

1335

LEMMA 5.5 (THE OPERATIONAL SEMANTICS PRESERVES WELL FORMED VALUE LOGS). *If $\vdash \Sigma$ and $(\Sigma, e) \rightarrow_N^* (\Sigma', e')$ then $\vdash \Sigma'$.*

1336

1337

PROOF. The proof is immediate by inspection of the Operational Semantics. \square

1338

1339

1340

LEMMA 5.6 (NOT ENOUGH STEPS IMPLIES ANY EXPRESSION RELATION). *If $(\Sigma, e) \rightarrow_N^k (\Sigma', e')$ and (Σ', e') is not irreducible, then $\forall j \leq k. (j, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ and $(j, \Psi, \Sigma, e) \in \mathcal{EH}^N \llbracket \bar{\tau} \rrbracket$.*

1341

1342

PROOF. Both conclusions are immediate, since the implications in the relations are vacuously true. \square

1343

1344

LEMMA 5.7 (THE OPERATIONAL SEMANTICS ONLY GROWS STORES). *If $(\Sigma, e) \rightarrow_N^* (\Sigma', e')$ then $\Sigma' \supseteq \Sigma$.*

1345

1346

PROOF. This is a corollary of Lemma 5.8. \square

1347

1348

5.3.2 Lemmas Used With Mention

1349

1350

1351

1352

LEMMA 5.8 (THE OPERATIONAL SEMANTICS PRODUCES VALUE LOG EXTENSIONS). *If $(\Sigma, e) \rightarrow_N^* (\Sigma', e')$, then $\exists \bar{\ell} \subseteq \text{dom}(\Sigma')$ such that $\bar{\ell} \notin \text{dom}(\Sigma)$ and $\Sigma' = \Sigma[\bar{\ell} \mapsto (v, _)]$.*

1353 **PROOF.** By inspection of the Operational Semantics, no steps modify the value stored in the value log, meaning
 1354 $\Sigma' \supseteq \Sigma$.

1355 And also by the inspection of the Operational Semantics, there is exactly one rule to allocate new entries in the value
 1356 log, meaning $\Sigma' \setminus \Sigma$ is a suitable choice for $\overline{[\ell \mapsto (v, _)]}$. \square
 1357

1358 **LEMMA 5.9 (STEPS ARE PRESERVED IN FUTURE VALUE LOGS).** *If $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$ and $\overline{\ell \notin \text{dom}(\Sigma')}$ then $(\Sigma[\overline{\ell \mapsto (v, _)}], e) \longrightarrow_N^j$
 1359 $(\Sigma'[\overline{\ell \mapsto (v, _)}], e')$.*
 1360

1361 **PROOF.** Since all of the added locations are not in Σ' , and therefore also not in Σ , no rule that will lookup a label in
 1362 the derivation tree for $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$ will find a different value or type.
 1363

1364 The only remaining notable reduction steps are those that allocate a new label and value entry, but since $\overline{\ell \notin \text{dom}(\Sigma')}$,
 1365 we can allocate the same entry unchanged. \square
 1366

1367 **LEMMA 5.10 (SUBTYPING PRESERVES LOGICAL RELATIONS).** $\forall \Sigma, k, \Psi, \tau, \tau'$. *where $\Sigma : (k, \Psi)$ and $\tau \leq \tau'$.*

- 1368 (1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau' \rrbracket$*
 1369 (2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau' \rrbracket$*
 1370 (3) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^N \llbracket \tau, \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^N \llbracket \tau', \bar{\tau} \rrbracket$*
 1371 (4) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau, \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau', \bar{\tau} \rrbracket$*
 1372
 1373
 1374

1375 **PROOF.** Proceed by mutual induction on k and τ :
 1376

- 1377 • $k = 0$: Both 1 and 3 are immediate if $e \neq \ell$.
 1378 If $e = \ell$ then 1 and 3 follow immediately from 2 and 4.
 1379 2 and 4 follow identically in the $k = 0$ case as they do in the $k > 0$ case, but the function case is vacuously true.
- 1380 • $k > 0$:

1381 (1) Unfolding our hypothesis, there is some (Σ', e') , j such that $(\Sigma, e) \longrightarrow_N^j (\Sigma', e')$.

1382 If $e' = \text{Err}^\bullet$ then we're done.

1383 Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi')$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

1384 We now have two obligations:

- 1385 a) $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N \llbracket \tau' \rrbracket$.
- 1386 b) $\Sigma' : (k - j, \Psi')$.

1387 For a) by IH 2) (not necessarily smaller by type or index), we have $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N \llbracket \tau' \rrbracket$, which is
 1388 what we wanted to show.
 1389
 1390
 1391

1392 For b), this is immediate from the premise.
 1393

1394 (2) Case split on $\tau \leq \tau'$:

- 1395 i) $\tau \leq \tau'$: immediate.
- 1396 ii) $\text{Nat} \leq \text{Int}$: immediate because $\mathbb{N} \subseteq \mathbb{Z}$.
- 1397 iii) $\tau_1 \times \tau_2 \leq \tau'_1 \times \tau'_2$, with $\tau_1 \leq \tau'_1$ and $\tau_2 \leq \tau'_2$:

1398 We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau' \rrbracket$.

1399 Unfolding our hypothesis, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

1400 We want to show $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N \llbracket \tau'_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N \llbracket \tau'_2 \rrbracket$.

1401 We can apply IH 2) (smaller by type) to both of these judgements to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N \llbracket \tau'_1 \rrbracket$ and
 1402
 1403
 1404

1405 $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N \llbracket \tau'_2 \rrbracket$.

1406 This is sufficient to show $(k, \Psi, \Sigma, \Sigma(\ell)) \in \mathcal{V}^N \llbracket \tau' \rrbracket$.

1407 iv) $\tau_1 \rightarrow \tau_2 \leq \tau'_1 \rightarrow \tau'_2$, with $\tau'_1 \leq \tau_1$ and $\tau_2 \leq \tau'_2$:

1408 We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau' \rrbracket$.

1409 Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.

1410 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau'_1 \rrbracket$.

1411 Let $\tau_0 \geq \tau'_2$.

1412 We want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

1413 From IH 2) (smaller by type) applied to the facts that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau'_1 \rrbracket$ and that $\tau'_1 \leq \tau_1$ gives

1414 us $(j + 1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

1415 Then, we can apply our hypothesis about $\Sigma(\ell)$ (noting that $\tau_0 \geq \tau'_2 \geq \tau_2$) to get $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in$

1416 $\mathcal{E}^N \llbracket \tau_0 \rrbracket$, which is what we wanted to prove.

1417 (3) Unfolding our hypothesis, we get that there are some (Σ', e') , j such that $(\Sigma, e) \xrightarrow{j}_N (\Sigma', e')$ and (Σ', e')

1418 are irreducible.

1419 If $e' = \text{Err}^\bullet$, then we're done.

1420 Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}\mathcal{H}^N \llbracket \tau, \bar{\tau} \rrbracket$,

1421 which means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.

1422 Then by IH 4) (not necessarily smaller by type or index) with $\tau \leq \tau'$, we get $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau', \bar{\tau} \rrbracket$,

1423 which is what we wanted to show.

1424 (4) We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau', \bar{\tau} \rrbracket$.

1425 We case split on $\tau \leq \tau'$:

1426 i) $\tau = \tau'$: immediate by premise.

1427 ii) $\text{Nat} \leq \text{Int}$:

1428 by our premise, we already get that $\forall \tau_0 \in \bar{\tau}$, $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau_0 \rrbracket$.

1429 Therefore, it suffices to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$ given $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \text{Nat} \rrbracket$ which is immediate since $\mathbb{N} \subset \mathbb{Z}$.

1430 iii) $\tau_1 \times \tau_2 \leq \tau'_1 \times \tau_2$ with $\tau_1 \leq \tau'_1$ and $\tau_2 \leq \tau'_2$:

1431 by our premise, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$ and $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}\mathcal{H}^N \llbracket \tau_1, \text{fst}(\bar{\tau}) \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in$

1432 $\mathcal{V}\mathcal{H}^N \llbracket \tau_2, \text{snd}(\bar{\tau}) \rrbracket$.

1433 We can apply IH 4) (smaller by type) to both to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}\mathcal{H}^N \llbracket \tau'_1, \text{fst}(\bar{\tau}) \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in$

1434 $\mathcal{V}\mathcal{H}^N \llbracket \tau'_2, \text{snd}(\bar{\tau}) \rrbracket$, which is what we wanted to show.

1435 iv) $\tau_1 \rightarrow \tau_2 \leq \tau'_1 \rightarrow \tau'_2$ with $\tau'_1 \leq \tau_1$ and $\tau_2 \leq \tau'_2$:

1436 unfolding what we want to show, let $\Sigma' \supseteq \Sigma$, $(j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (j, \Psi')$.

1437 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau'_1 \rrbracket$.

1438 Let $\tau_0 \leq \tau'_2$.

1439 We want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}\mathcal{H}^N \llbracket \tau_0, \text{cod}(\bar{\tau}) \rrbracket$.

1440 By IH 2) (smaller by type), we get that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

1441 We can then apply the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau, \bar{\tau} \rrbracket$ to get $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}\mathcal{H}^N \llbracket \tau_0, \text{cod}(\bar{\tau}) \rrbracket$,

1442 which is what we wanted to show.

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457 □

1458

1459 LEMMA 5.11 (RV-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$*
 1460 *and $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N \llbracket \bar{\tau} \rrbracket$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{VH}^N \llbracket \bar{\tau} \rrbracket$*

1461 PROOF. We want to show $(k - j, \Psi', \Sigma', \ell) \in \mathcal{VH}^N \llbracket \bar{\tau} \rrbracket$.

1462 Let τ be the head of $\bar{\tau}$ so that $\bar{\tau} = [\tau, \dots]$.

1463 We proceed by induction over k and τ :

1464 • $k = 0$: The function and dynamic cases are vacuously true, and the rest follow as in the other case.

1465 • $k > 0$:

1466 i) $\tau = \text{Int}$: immediate because $\Sigma(\ell) = \Sigma'(\ell)$.

1467 ii) $\tau = \text{Nat}$: same as previous case.

1468 iii) $\tau = \text{Bool}$: same as previous case.

1469 iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

1470 We want to show $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{VH}^L \llbracket \tau_1, \overline{\text{fst}(\tau)} \rrbracket$ and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{VH}^L \llbracket \tau_2, \overline{\text{snd}(\tau)} \rrbracket$.

1471 We have $(k, \Psi, \Sigma, \ell_1) \in \mathcal{VH}^L \llbracket \tau_1, \overline{\text{fst}(\tau)} \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{VH}^L \llbracket \tau_2, \overline{\text{snd}(\tau)} \rrbracket$.

1472 Both follow by IH (smaller by type).

1473 v) $\tau = \tau_1 \rightarrow \tau_2$:

1474 Let $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ such that $\Sigma'' : (j', \Psi')$.

1475 Let $\ell_0 \in \text{dom}(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_0) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

1476 Let $\tau_0 \supseteq \tau_2$.

1477 We want to show $(j', \Psi'', \Sigma'', \text{app}\{\tau_0\} \ell \ell_0) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

1478 Since $(j', \Psi'') \sqsupseteq (k, \Psi)$ and $\Sigma'' \supseteq \Sigma$, we can apply our premise to finish the case.

1479 vi) $\tau = *$: note by downward closure, $\Sigma' : (k - j - 1, \Psi')$.

1480 Then we want to show $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket * \times * \rrbracket$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket * \rightarrow * \rrbracket$.

1481 We know $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket * \times * \rrbracket$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket * \rightarrow * \rrbracket$.

1482 The case follows by the IH (smaller by index).

1483 □

1484

1485 LEMMA 5.12 (EXTENSIONS PRESERVE VALUE LOG TYPING). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$*
 1486 *and $\Sigma' : (k - j, \Psi')$ and $\ell \notin \text{dom}(\Sigma')$ and $\Sigma[\ell \mapsto (v, _)] : (k, \Psi[\ell \mapsto \bar{\tau}])$ then $\Sigma'[\ell \mapsto (v, _)] : (k - j, \Psi'[\ell \mapsto \bar{\tau}])$.*

1487 PROOF. Note that all of the conditions in $\Sigma'[\ell \mapsto (v, _)] : (k - j, \Psi'[\ell \mapsto \bar{\tau}])$ besides those concerning the history
 1488 relation are immediate from the hypotheses.

1489

1490 Let $\Sigma'' = \Sigma'[\ell \mapsto (v, _)]$ and let $\Psi'' = \Psi'[\ell \mapsto \bar{\tau}]$.

1491 We want to show $\forall j' < k - j$, and $\forall \ell \in \text{dom}(\Sigma'')$, $(j', \Psi'', \Sigma'', \ell) \in \mathcal{VH}^N \llbracket \Psi''(\ell) \rrbracket$.

1492 Note by downward closure, $\Sigma'' : (j', \Psi'')$. If $\ell \in \text{dom}(\Sigma')$, then we can apply Lemma 5.11 with the fact that
 1493 $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$.

1494 If $\ell \notin \text{dom}(\Sigma')$, then $\ell \in \bar{\ell}$.

1495 Then we can apply Lemma 5.11 with the fact that $(j', \Psi'') \sqsupseteq (k, \Psi[\ell \mapsto \bar{\tau}])$ and $\Sigma'' \supseteq \Sigma[\ell \mapsto (v, _)]$ to get $(j', \Psi'', \Sigma'', \ell) \in$
 1496 $\mathcal{VH}^N \llbracket \Psi''(\ell) \rrbracket$, which is what we wanted to show. □

1497

1509 LEMMA 5.13 (LATER THAN PRESERVED BY LOWER STEPS). *If $(j, \Psi') \sqsupseteq (k, \Psi)$ and $j' \leq j$ then $(j - j', \Psi') \sqsupseteq (k - j', \Psi)$.*

1510

1511 PROOF. Unfolding the world extension definition, we need to show $j - j' \leq k - j'$ and $\forall \ell \in \text{dom}(\Psi), \Psi'(\ell) = \Psi(\ell)$.

1512

For the first condition, since $j \leq k$ and $j' \leq j$, $j - j' \leq k - j'$.

1513

For the second condition, we can unfold the hypothesis to get the statement we need. \square

1514

1515 LEMMA 5.14 (RE-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$*

1516

and $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{E}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$.

1517

1518 PROOF. Unfolding the relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \xrightarrow{j'}_N (\Sigma'', e')$.

1519

If $e' = \text{Err}^\bullet$ then we're done.

1520

Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{V}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$.

1521

1522

1523

By Lemma 5.8, $\Sigma'' = \Sigma[\ell \mapsto (v, _)]$.

1524

By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi[\ell \mapsto \bar{\tau}]$.

1525

We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma[\ell' \mapsto (v', _)]$.

1526

And from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi[\ell' \mapsto \bar{\tau}']$.

1527

By alpha renaming, we can assume that $\ell' \notin \text{dom}(\Sigma'')$.

1528

Then by Lemma 5.9, we get that $(\Sigma', e) \xrightarrow{j'}_N (\Sigma''[\ell' \mapsto (v', _)], e')$.

1529

1530

1531

Now, unfolding the expression relation in what we want to show, we have two obligations:

1532

1533

1534

a) $\Sigma''[\ell' \mapsto (v', _)] : (k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'])$.

1535

b) $(k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'], \Sigma''[\ell' \mapsto (v', _)], e') \in \mathcal{V}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$.

1536

1537

For a) we can apply Lemma 5.12. We have a number of obligations:

1538

1539

1540

1541

1542

1543

1544

1545

1546

i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.

ii) $\Sigma'' \supseteq \Sigma$: immediate.

iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 5.13.

iv) $\Sigma'' : (k - j - j', \Psi'')$: immediate by downward closure.

v) $\ell' \notin \text{dom}(\Sigma'')$: assumed above by alpha renaming.

vi) $\Sigma[\ell' \mapsto (v', _)] : (k - j, \Psi[\ell' \mapsto \bar{\tau}'])$: this is exactly $\Sigma' : (k - j, \Psi')$.

1546

1547

For b), we can apply Lemma 5.11 with the fact proven in a). \square

1548

1549

1550

LEMMA 5.15 (E-V-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$*

1551

then

1552

(1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^N \llbracket \tau \rrbracket$*

1553

(2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$*

1554

1555

PROOF. Proceed by simultaneous induction on k and τ :

1556

1557

- $k = 0$: 1) follows immediately from 2).

1558

Proceeds similarly to the other case, but function and dynamic cases are vacuously true.

1559

- $k > 0$:

1560

1) Unfolding the expression relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \xrightarrow{j'}_N (\Sigma'', e')$.

If $e' = \text{Err}^\bullet$ then we're done.

Otherwise, there is some $(k-j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k-j', \Psi'')$ and $(k-j', \Psi'', \Sigma'', e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

By Lemma 5.8, $\Sigma'' = \overline{\Sigma[\ell \mapsto (v, _)]}$.

By the fact that $\Sigma'' : (k-j', \Psi'')$ this also means $\Psi'' = \overline{\Psi[\ell \mapsto \bar{\tau}]}$.

We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \overline{\Sigma[\ell' \mapsto (v', _)]}$, and from $\Sigma' : (k-j, \Psi')$ that $\Psi' = \overline{\Psi[\ell' \mapsto \bar{\tau}]}$.

By alpha renaming, we can assume that $\overline{\ell' \notin \text{dom}(\Sigma')}$.

Then by Lemma 5.9, we get that $(\Sigma', e) \xrightarrow{j'}_N (\Sigma''[\ell' \mapsto (v', _)], e')$.

Now, unfolding the expression relation in what we want to show, we have two obligations:

- a) $\Sigma''[\ell' \mapsto (v', _)] : (k-j-j', \Psi''[\ell' \mapsto \bar{\tau}'])$.
- b) $(k-j-j', \Psi''[\ell' \mapsto \bar{\tau}'], \Sigma''[\ell' \mapsto (v', _)], e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

For a) we can apply Lemma 5.12. We have a number of obligations:

- i) $\Sigma : (k-j, \Psi)$: immediate by downward closure.
- ii) $\Sigma'' \supseteq \Sigma$: immediate.
- iii) $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi)$: by Lemma 5.13.
- iv) $\Sigma'' : (k-j-j', \Psi'')$: immediate by downward closure.
- v) $\overline{\ell' \notin \text{dom}(\Sigma')}$: assumed above by alpha renaming.
- vi) $\Sigma[\ell' \mapsto (v', _)] : (k-j, \Psi[\ell' \mapsto \bar{\tau}'])$: this is exactly $\Sigma' : (k-j, \Psi')$.

For b), we can apply the IH 2) (not necessarily smaller by type or index) with the fact proven in a).

2) We want to show that $(k-j, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$.

We case split on τ :

- i) $\tau = \text{Nat}$: then $\Sigma(\ell) = (n, _)$ where $n \in \mathbb{N}$, so the case is immediate.
- ii) $\tau = \text{tint}$: same as above.
- iii) $\tau = \text{Bool}$: same as above.

iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

Unfolding our hypothesis gives us $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

Applying IH 2) (smaller by type) to both gives us $(k-j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and $(k-j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$, which is sufficient to complete the case.

v) $\tau = \tau_1 \rightarrow \tau_2$: Let $\Sigma'' \supseteq \Sigma'$ and $(j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (j', \Psi'')$.

Let $\ell_0 \in \text{dom}(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_0) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

Let $\tau_0 \supseteq \tau_2$.

We want to show $(j', \Psi'', \Sigma'', \text{app}\{\tau_0\} \ell \ell_0) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

Since \supseteq and \sqsupseteq are both transitive, we have $\Sigma'' \supseteq \Sigma$, and $(j', \Psi'') \sqsupseteq (k, \Psi)$.

Therefore we can apply the hypothesis to complete the case.

1613 vi) $\tau = *$: we want to show $(k-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$ or $\mathcal{V}^N \llbracket \text{Bool} \rrbracket$ or $\mathcal{V}^N \llbracket * \times * \rrbracket$ or $\mathcal{V}^N \llbracket * \rightarrow * \rrbracket$.
 1614 This follows from IH 2) (smaller by index).
 1615

1616 □

1617
 1618 LEMMA 5.16 (CHECK IS A NO OP IN NATURAL). (1) $(k+1, \Psi, \Sigma, \text{assert } \tau_0 e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ iff $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$.
 1619 (2) $(k+1, \Psi, \Sigma, \text{assert } \tau_0 e) \in \mathcal{E}^{\mathcal{H}^V} \llbracket \bar{\tau} \rrbracket$ iff $(k, \Psi, \Sigma, e) \in \mathcal{E}^{\mathcal{H}^V} \llbracket \bar{\tau} \rrbracket$.
 1620

1621 PROOF. By the operational semantics, $(\Sigma, \text{assert } \tau_0 e) \rightarrow_N (\Sigma, e)$, so the statement is immediate. □
 1622

1623 LEMMA 5.17 (APP ANNOTATIONS DON'T MATTER IN NATURAL). (1) $(k+1, \Psi, \Sigma, \text{app}\{\tau_0\} e_1 e_2) \in \mathcal{E}^N \llbracket \tau \rrbracket$ iff $(k, \Psi, \Sigma, e_1 e_2) \in$
 1624 $\mathcal{E}^N \llbracket \tau \rrbracket$.
 1625 (2) $(k+1, \Psi, \Sigma, \text{app}\{\tau_0\} e_1 e_2) \in \mathcal{E}^{\mathcal{H}^V} \llbracket \bar{\tau} \rrbracket$ iff $(k, \Psi, \Sigma, e_1 e_2) \in \mathcal{E}^{\mathcal{H}^V} \llbracket \bar{\tau} \rrbracket$.
 1626

1627 PROOF. By the operational semantics, $(\Sigma, \text{app}\{\tau_0\} e_1 e_2) \rightarrow_N (\Sigma, \text{assert } \tau_0 e_1 e_2)$.
 1628

1629 We can apply Lemma 5.16 to complete the proof. □
 1630

1631 LEMMA 5.18 (PAIRS OF SEMANTICALLY WELL TYPED TERMS ARE SEMANTICALLY WELL TYPED). If $(k, \Psi, \Sigma, e_1) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$
 1632 and $(k, \Psi, \Sigma, e_2) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$ then $(k, \Psi, \Sigma, \langle e_1, e_2 \rangle) \in \mathcal{E}^N \llbracket \tau_1 \times \tau_2 \rrbracket$.
 1633

1634 PROOF. Unfolding the expression relation in our hypothesis about e_1 , we get that there are $(\Sigma, e'_1), j$ such that
 1635 $(\Sigma, e_1) \rightarrow_N^j (\Sigma, e'_1)$ and (Σ', e'_1) is irreducible.
 1636

1637 If $e'_1 = \text{Err}^\bullet$, then were done because the entire application steps to an error.

1638 Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi)$ and $(k-j, \Psi', \Sigma', e'_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

1639 This means $e'_1 = \ell_1$ for some $\ell_1 \in \text{dom}(\Sigma')$.
 1640

1641 With this and by the OS, we get $(\Sigma, \langle e_1, e_2 \rangle) \rightarrow_N^j (\Sigma', \langle \text{loc}_1, e_2 \rangle)$.
 1642

1643
 1644 We can apply Lemma 5.15 to our hypothesis about e_2 to get $(k-j, \Psi', \Sigma', e_2) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

1645 Unfolding the expression relation, we get that there are $(\Sigma', e'_2), j'$ such that $(\Sigma', e_2) \rightarrow_N^{j'} (\Sigma', e'_2)$ and (Σ'', e'_2) is
 1646 irreducible.
 1647

1648 If $e'_2 = \text{Err}^\bullet$, then were done because the entire application steps to an error.

1649 Otherwise, there is a $(k-j-j', \Psi'') \sqsupseteq (k-j, \Psi')$ such that $\Sigma'' : (k-j-j', \Psi'')$ and $(k-j-j', \Psi'', \Sigma'', e'_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$,
 1650 which means $e'_2 = \ell_2$ for some $\ell_2 \in \text{dom}(\Sigma'')$.
 1651

1652
 1653 Putting everything together we get $(\Sigma, \langle e_1, e_2 \rangle) \rightarrow_N^{j'} (\Sigma'', \langle \ell_1, \ell_2 \rangle)$, with $\Sigma'' : (k-j-j', \Psi'')$.

1654 Note by OS, $(\Sigma'', \langle \ell_1, \ell_2 \rangle) \rightarrow_N (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)])$ where $\ell' \notin \text{dom}(\Sigma'')$.
 1655

1656 We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] : (k-j-j'-1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
 1657

1658 Note the only interesting part of this statement is that $\forall k' < k-j-j'-1. (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto$
 1659 $(\langle \ell_1, \ell_2 \rangle, _)] : \Psi''(\ell_1) \times \Psi''(\ell_2) \rrbracket$.
 1660

1661 This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in$
 1662 $\mathcal{V}^{\mathcal{H}^N} \llbracket \Psi''(\ell_1) \rrbracket$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^{\mathcal{H}^N} \llbracket \Psi''(\ell_2) \rrbracket$.
 1663
 1664

1665 We know that $(k - j, \Psi', \Sigma', \ell'_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$, and Lemma 5.15 with down-
 1666 ward closure and the store typing judgement above.

1667 From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)]], \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and
 1668 $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle], \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

1670 This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)]], \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N \llbracket \tau_1 \times \tau_2 \rrbracket$,
 1671 which is what we wanted to prove. \square

1673 LEMMA 5.19 (PAIRS OF HISTORY RELATED TERMS ARE HISTORY RELATED). *If $(k, \Psi, \Sigma, e_1) \in \mathcal{E}\mathcal{H}^N \llbracket fst(\bar{\tau}) \rrbracket$ and
 1674 $(k, \Psi, \Sigma, e_2) \in \mathcal{E}\mathcal{H}^N \llbracket snd(\bar{\tau}) \rrbracket$ then $(k, \Psi, \Sigma, \langle e_1, e_2 \rangle) \in \mathcal{E}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$.*

1676 PROOF. Unfolding the erroring expression relation in our hypothesis about e_1 , we get that there are $(\Sigma, e'_1), j$ such
 1677 that $(\Sigma, e_1) \xrightarrow{j}_N (\Sigma, e'_1)$ and (Σ', e'_1) is irreducible.

1678 If $e'_1 = \text{Err}^\bullet$, then were done because the entire application steps to an error.

1679 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi)$ and $(k - j, \Psi', \Sigma', e'_1) \in \mathcal{V}\mathcal{H}^N \llbracket fst(\bar{\tau}) \rrbracket$.

1681 This means $e'_1 = \ell_1$ for some $\ell_1 \in \text{dom}(\Sigma')$.

1683 With this and by the OS, we get $(\Sigma, \langle e_1, e_2 \rangle) \xrightarrow{j}_N (\Sigma', \langle loc_1, e_2 \rangle)$.

1686 We can apply Lemma 5.14 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{E}\mathcal{H}^N \llbracket snd(\bar{\tau}) \rrbracket$.

1687 Unfolding the erroring expression relation, we get that there are $(\Sigma', e'_2), j'$ such that $(\Sigma', e_2) \xrightarrow{j'}_N (\Sigma', e'_2)$ and (Σ'', e'_2)
 1688 is irreducible.

1689 If $e'_2 = \text{Err}^\bullet$, then were done because the entire application steps to an error.

1691 Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e'_2) \in$
 1692 $\mathcal{V}\mathcal{H}^N \llbracket snd(\bar{\tau}) \rrbracket$, which means $e'_2 = \ell_2$ for some $\ell_2 \in \text{dom}(\Sigma'')$.

1694 Putting everything together we get $(\Sigma, \langle e_1, e_2 \rangle) \xrightarrow{j'}_N (\Sigma'', \langle \ell_1, \ell_2 \rangle)$, with $\Sigma'' : (k - j - j', \Psi'')$.

1696 Note by OS, $(\Sigma'', \langle \ell_1, \ell_2 \rangle) \xrightarrow{N} (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)]])$ where $\ell' \notin \text{dom}(\Sigma'')$.

1698 We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] : (k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.

1699 Note the only interesting part of this statement is that $\forall k' < k - j - j' - 1. (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto$
 1700 $(\langle \ell_1, \ell_2 \rangle, _)]], \ell') \in \mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell_1) \times \Psi''(\ell_2) \rrbracket$.

1702 This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in$
 1703 $\mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell_1) \rrbracket$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell_2) \rrbracket$.

1705 We know that $(k - j, \Psi', \Sigma', \ell'_1) \in \mathcal{V}\mathcal{H}^N \llbracket fst(\bar{\tau}) \rrbracket$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}\mathcal{H}^N \llbracket snd(\bar{\tau}) \rrbracket$, and Lemma 5.11
 1706 with downward closure and the store typing judgement above.

1708 From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)]], \ell_1) \in \mathcal{V}\mathcal{H}^N \llbracket fst(\bar{\tau}) \rrbracket$
 1709 and $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle], \ell_2) \in \mathcal{V}\mathcal{H}^N \llbracket snd(\bar{\tau}) \rrbracket$.

1711 This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)]], \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$, which
 1712 is what we wanted to prove. \square

1714 LEMMA 5.20 (APPLICATIONS OF SEMANTICALLY WELL TYPED TERMS ARE SEMANTICALLY WELL TYPED). *If $(k, \Psi, \Sigma, e_f) \in$
 1715 $\mathcal{E}^N \llbracket \tau \rightarrow \tau' \rrbracket$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ then $\forall \tau_0 \geq \tau', (k, \Psi, \Sigma, \text{app}\{\tau_0\} e_f e) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.*

1716 2023-04-10 15:45. Page 33 of 1-104.

1717 PROOF. Unfolding the expression relation in our hypothesis about e_f , we get that there are $(\Sigma', e'_f), j$ such that
 1718 $(\Sigma, e_f) \longrightarrow_N^j (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.

1719 If $e'_f = \text{Err}^\bullet$, then we're done because the entire application steps to an error.

1720 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_f) \in \mathcal{V}^N \llbracket \tau \rightarrow \tau' \rrbracket$.

1721 This means $e'_f = \ell_f$ for some $\ell_f \in \text{dom}(\Sigma')$.

1722 Using this, we know from the OS that $(\Sigma, \text{app}\{\tau_0\} e_f e) \longrightarrow_N^j (\Sigma', \text{app}\{\tau_0\} \ell_f e)$.

1723 We can apply Lemma 5.15 with $\Sigma' : (k - j, \Psi')$ to our hypothesis about e to get $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^N \llbracket \tau \rrbracket$.

1724 Unfolding the expression relation, we get that there are $(\Sigma'', e'), j'$ such that $(\Sigma', e) \longrightarrow_N^{j'} (\Sigma'', e')$ where (Σ'', e') is
 1725 irreducible.

1726 If $e' = \text{Err}^\bullet$ then we're done, because the whole application errors.

1727 Otherwise, there exists $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

1728 This means $e' = \ell$ for some $\ell \in \text{dom}(\Sigma'')$.

1729 Putting what we have together, by the OS, $(\Sigma, \text{app}\{\tau_0\} e_f e) \longrightarrow_N^{j+j'} (\Sigma'', (\text{app}\{\tau_0\} \ell_f \ell))$.

1730 We have $(k - j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^N \llbracket \tau \rightarrow \tau' \rrbracket$ and $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \sqsupseteq \Sigma'$ and $\Sigma'' : (k - j - j', \Psi'')$
 1731 and $\tau_0 \geq \tau'$.

1732 We can combine these to get $(k - j - j', \Psi'', \Sigma'', \text{app}\{\tau_0\} \ell_f \ell) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

1733 This is sufficient to complete the proof. □

1734 COROLLARY 5.21. *If $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^N \llbracket * \rrbracket$ and $\Sigma(\ell) = w$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket * \rrbracket$ then $(k - 1, \Psi, \Sigma, \text{app}\{*\} w e) \in$
 1735 $\mathcal{E}^N \llbracket * \rrbracket$.*

1736 LEMMA 5.22 (APPLICATIONS OF HISTORY RELATED TERMS ARE HISTORY RELATED). *If $(k, \Psi, \Sigma, e_f) \in \mathcal{E}^N \llbracket \tau, \bar{\tau} \rrbracket$ and
 1737 $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \text{dom}(\tau) \rrbracket$ then $\forall \tau_0 \geq \text{cod}(\tau)$, $(k, \Psi, \Sigma, \text{app}\{\tau_0\} e_f e) \in \mathcal{E}^N \llbracket \tau_0, \text{cod}(\bar{\tau}) \rrbracket$.*

1738 PROOF. Unfolding the erroring expression relation in our hypothesis about e_f , we get that there are $(\Sigma', e'_f), j$ such
 1739 that $(\Sigma, e_f) \longrightarrow_N^j (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.

1740 If $e'_f = \text{Err}^\bullet$, then we're done because the entire application steps to an error.

1741 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_f) \in \mathcal{V}^N \llbracket \tau, \bar{\tau} \rrbracket$.

1742 This means $e'_f = \ell_f$ for some $\ell_f \in \text{dom}(\Sigma')$.

1743 Using this, we know from the OS that $(\Sigma, \text{app}\{\tau_0\} e_f e) \longrightarrow_N^j (\Sigma', \text{app}\{\tau_0\} \ell_f e)$.

1744 We can apply Lemma 5.15 with $\Sigma' : (k - j, \Psi')$ to our hypothesis about e to get $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^N \llbracket \text{dom}(\tau) \rrbracket$.

1745 Unfolding the expression relation, we get that there are $(\Sigma'', e'), j'$ such that $(\Sigma', e) \longrightarrow_N^{j'} (\Sigma'', e')$ where (Σ'', e') is
 1746 irreducible.

1747 If $e' = \text{Err}^\bullet$ then we're done, because the whole application errors.

1748 Otherwise, there exists $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

1749 This means $e' = \ell$ for some $\ell \in \text{dom}(\Sigma'')$.

1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768

Putting what we have together, by the OS, $(\Sigma, \text{app}\{\tau_0\} e_f e) \xrightarrow{N}^{j+j'} (\Sigma'', (\text{app}\{\tau_0\} \ell_f \ell))$.

We have $(k - j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^N \llbracket \tau \rightarrow \tau' \rrbracket$ and $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \sqsupseteq \Sigma'$ and $\Sigma'' : (k - j - j', \Psi'')$ and $\tau_0 \geq \tau'$.

We can combine these to get $(k - j - j', \Psi'', \Sigma'', \text{app}\{\tau_0\} \ell_f \ell) \in \mathcal{E}\mathcal{H}^N \llbracket \tau_0, \text{cod}(\bar{\tau}) \rrbracket$.

This is sufficient to complete the proof. \square

COROLLARY 5.23. *If $(k, \Psi, \Sigma, e_f) \in \mathcal{E}\mathcal{H}^N \llbracket *, \bar{\tau} \rrbracket$ and $(k - 1, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket * \rrbracket$ then $(k - 1, \Psi, \Sigma, \text{app}\{\tau_0\} e_f e) \in \mathcal{E}\mathcal{H}^N \llbracket *, \text{cod}(\bar{\tau}) \rrbracket$.*

LEMMA 5.24 (EXPRESSION RELATION IMPLIES EXPRESSION HISTORY RELATION). (1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ then*

$(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^N \llbracket \tau \rrbracket$.

(2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \tau \rrbracket$.*

PROOF. Proceed by induction on k and τ :

- $k = 0$: 1) is immediate from 2).

- $\tau = \text{Int}$: immediate.

- $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

The case follows from the IH on ℓ_1 and ℓ_2 .

- $\tau = \tau_1 \rightarrow \tau_2$: vacuously true.

- $\tau = *$: vacuously true.

- $k > 0$: 1) is immediate from 2).

- $\tau = \text{Int}$: immediate.

- $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

The case follows from the IH on ℓ_1 and ℓ_2 .

- $\tau = \tau_1 \rightarrow \tau_2$: Follows from 1) from the IH (smaller by index).

- $\tau = *$: Follows from 2) from the IH (smaller by index), using $* \times *$, $* \rightarrow *$, or Int.

\square

LEMMA 5.25 (MONITOR COMPATIBILITY). *If $\Sigma : (k, \Psi)$, then*

(1) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$ and $\Sigma(\ell') = (\ell, \text{some}(\tau', \tau))$, then $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket \tau' \rrbracket$*

(2) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^N \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \text{mon}\{\tau' \leftarrow \tau\} e) \in \mathcal{E}^N \llbracket \tau' \rrbracket$.*

(3) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \Psi(\ell) \rrbracket$ and $\Psi(\ell) = [\tau_s, \dots]$ and $\tau \geq \tau_s$ and $\Sigma' = \Sigma[\ell' \mapsto (\ell, \text{some}(\tau', \tau))]$ and $\Psi' = [\ell' \mapsto \tau', \tau, \Psi(\ell)]\Psi$ and $\ell' \notin \text{dom}(\Sigma)$ and $\vdash \Sigma'$ then $(k, \Psi', \Sigma', \ell') \in \mathcal{V}\mathcal{H}^N \llbracket \tau', \tau, \Psi(\ell) \rrbracket$*

(4) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$ and $\bar{\tau} = [\tau, \dots]$ then $(k, \Psi, \Sigma, \text{mon}\{\tau' \leftarrow \tau\} e) \in \mathcal{E}\mathcal{H}^N \llbracket \tau', \tau, \bar{\tau} \rrbracket$*

PROOF. Proceed by simultaneous induction on k and τ .

- $k = 0$: 2) and 4) follow from 1) and 3) respectively.

The proofs follow similarly to the other case, but any function or dynamic cases are vacuously true.

- $k > 0$:

1) Unfolding the relation in the statement we want to prove, note from our hypothesis about Σ , we get that $\vdash \Sigma$.

Proceed by case analysis on τ' :

a) $\tau' = \text{Nat}$: Since $\vdash \Sigma$, we have $\text{pointsto}(\Sigma, \ell') \propto \text{Nat}$.

Therefore, we have $\text{pointsto}(\Sigma, \ell') \in \mathbb{N}$, which is sufficient to complete the case.

b) $\tau' = \text{Int}$: same reasoning as Nat.

c) $\tau' = \text{Bool}$: same reasoning as Nat.

d) $\tau' = \tau'_1 \times \tau'_2$: By the fact that $\vdash \Sigma$, this case is a contradiction.

e) $\tau' = \tau'_1 \rightarrow \tau'_2$: Unfolding the value relation, let $\Sigma' \supseteq \Sigma$, and $(j, \Psi') \sqsupseteq (k, \Psi)$, such that $\Sigma' : (j, \Psi')$.

Let ℓ_v such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \text{dom}(\tau') \rrbracket$.

Let $\tau_0 \leqslant \text{cod}(\tau')$.

We want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell' \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

Note by the operational semantics, $(\Sigma', \text{app}\{\tau_0\} \ell' \ell_v) \xrightarrow{2}_N$

$(\Sigma', \text{assert } \tau_0 (\text{mon } \{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v))))$.

Note by downward closure we have $\Sigma' : (j-2, \Psi')$.

Therefore it suffices to show $(j-2, \Psi', \Sigma', \text{assert } \tau_0 (\text{mon } \{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v)))) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

Note that $\tau_0 \geqslant \text{cod}(\tau')$.

By Lemma 5.10, it suffices to show $(j-2, \Psi', \Sigma', \text{assert } \tau_0 (\text{mon } \{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v)))) \in \mathcal{E}^N \llbracket \text{cod}(\tau') \rrbracket$.

By Lemma 5.16, it suffices to show $(j-3, \Psi', \Sigma', \text{mon } \{\text{cod}(\tau') \leftarrow \text{cod}(\tau')\} (\ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v))) \in \mathcal{E}^N \llbracket \text{cod}(\tau') \rrbracket$.

By IH 2) (smaller by type), it suffices to show $(j-3, \Psi', \Sigma', \ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v)) \in \mathcal{E}^N \llbracket \text{cod}(\tau') \rrbracket$.

By Lemma 5.17, it suffices to show $(j-2, \Psi', \Sigma', \text{app}\{\text{cod}(\tau')\} \ell (\text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v)) \in \mathcal{E}^N \llbracket \text{cod}(\tau') \rrbracket$.

We now have two cases:

i) $\tau = *$:

Then by Lemma 5.21 it suffices to show $(j-1, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket * \rrbracket$ and $(j-1, \Psi', \Sigma', \text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v) \in \mathcal{E}^N \llbracket \text{dom}(\tau') \rrbracket$.

Both follow by Lemma 5.15, and IH 2) (smaller by index) in the second case.

ii) $\tau = \tau_1 \rightarrow \tau_2$:

Then by Lemma 5.20 it suffices to show $(j-2, \Psi', \Sigma', \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$ and $(j-2, \Psi', \Sigma', \text{mon } \{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v) \in \mathcal{E}^N \llbracket \text{dom}(\tau') \rrbracket$.

Both follow by Lemma 5.15, and IH 2) (smaller by index) in the second case.

f) $\tau' = *$: Unfolding the relation in what we want to show, we want to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$ or $\mathcal{V}^N \llbracket \text{Bool} \rrbracket$ or $\mathcal{V}^N \llbracket * \times * \rrbracket$ or $\mathcal{V}^N \llbracket * \rightarrow * \rrbracket$.

In each case, we can apply IH 1) (smaller by index) to complete the case.

2) Unfolding the expression relation in our hypothesis, we have that there are (e', Σ') , j such that $(e, \Sigma) \xrightarrow{j}_N (e', \Sigma')$ with (e', Σ') irreducible.

If $e' = \text{Err}^\bullet$ then we're done, because the monitor will step to an error as well.

Otherwise, there is $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

This means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.

If $\neg \text{pointsto}(\Sigma', \ell) \propto \tau'$, then $(\Sigma, \text{mon} \{ \tau' \Leftarrow \tau \} e) \xrightarrow{j}_N (\Sigma', \text{mon} \{ \tau' \Leftarrow \tau \} \ell) \xrightarrow{N} (\Sigma', \text{TypeErr}(\tau', \ell))$, so we're done.

Otherwise, we have $\text{pointsto}(\Sigma', \ell) \propto \tau'$, and since $\text{pointsto}(\Sigma', \ell) \propto \tau$, we also have $\tau \propto \tau'$.

We have 5 cases:

(a) $\tau' = \text{Nat}$:

Then $(\Sigma', \text{mon} \{ \text{Nat} \Leftarrow \tau \} \ell) \xrightarrow{N} (\Sigma'[\ell' \mapsto (\ell, \text{some}(\text{Nat}, \tau))], \ell')$.

It suffices to show $(k - j - 1, \Psi'[\ell' \mapsto \text{Nat}, \tau, \Psi(\ell)], \Sigma'[\ell' \mapsto (\ell, \text{some}(\text{Nat}, \tau))], \ell) \in \mathcal{V}^N \llbracket \text{Nat} \rrbracket$, and that $\Sigma'[\ell' \mapsto (\ell, \text{some}(\text{Nat}, \tau))]$: $(k - j - 1, \Psi'[\ell' \mapsto \text{Nat}, \tau, \Psi(\ell)])$.

The first follows from downward closure, and the fact that $\Sigma'(\ell) \propto \text{Nat}$ means $\Sigma'(\ell) = n$.

The second follows from IH 3) (smaller by index).

(b) $\tau' = \text{Int}$: Essentially the same as Nat.

(c) $\tau' = \text{Bool}$: Essentially the same as Nat.

(d) $\tau' = \tau'_1 \times \tau'_2$:

By the fact that $\text{fst}(\Sigma'(\ell)) \propto \tau'_1 \times \tau'_2$, we have that $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

Then by the OS we have that $(\Sigma', \text{mon} \{ \tau' \Leftarrow \tau \} \ell) \xrightarrow{N} (\Sigma', \langle \text{mon} \{ \tau'_1 \Leftarrow \text{fst}(\tau) \} \ell_1, \text{mon} \{ \tau'_2 \Leftarrow \text{snd}(\tau) \} \ell_2 \rangle)$.

By downward closure, we get $\Sigma' : (k - j - 1, \Psi')$.

By Lemma 5.18, it suffices to show $(k - j - 1, \Psi', \Sigma', \text{mon} \{ \tau'_1 \Leftarrow \text{fst}(\tau) \} \ell_1) \in \mathcal{E}^N \llbracket \tau'_1 \rrbracket$ and $(k - j - 1, \Psi', \Sigma', \text{mon} \{ \tau'_2 \Leftarrow \text{snd}(\tau) \} \ell_2) \in \mathcal{E}^N \llbracket \tau'_2 \rrbracket$.

If $\tau = \tau_1 \times \tau_2$, then we have $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$, and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

Then we just need to apply IH 2) (smaller by type) and Lemma 5.15.

If $\tau = *$, then we have $(k - j, \Psi', \Sigma', \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N \llbracket * \rrbracket$.

This means $(k - j - 1, \Psi', \Sigma', \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N \llbracket * \times * \rrbracket$.

Therefore $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket * \rrbracket$, and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N \llbracket * \rrbracket$.

Then we just need to apply IH 2) (smaller by index).

(e) $\tau' = \tau'_1 \rightarrow \tau'_2$:

By the fact that $\tau \propto \tau'$, and by the OS, we have $(\Sigma', \text{mon} \{ \tau' \Leftarrow \tau \} \ell) \xrightarrow{N} (\Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))])$ for $\ell' \notin \text{dom}(\Sigma')$.

Let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))]$, and $\Psi'' = \Psi'[\ell' \mapsto [\tau', \tau, \Psi'(\ell)]]$.

We want to show $\Sigma'' : (k - j - 2, \Psi'')$.

1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976

To start, the condition on entries in the value log is immediate.

Otherwise the only interesting case is the value history relation.

Let $k' < k - j - 2$.

Then by downward closure, we get $\Sigma' : (k', \Psi')$.

By IH 3) (smaller by index), we get $(k', \Psi'', \Sigma'', \ell') \in \mathcal{VH}^N \llbracket \tau', \tau, \Psi(\ell) \rrbracket$, which is sufficient.

Then we just need to apply IH 1) (smaller by index).

(f) $\tau' = *$: case split on the shape of $\text{pointsto}(\Sigma', \ell)$:

i) $\text{pointsto}(\Sigma', \ell) = i$: the proof follows identically to the Nat case.

ii) $\text{pointsto}(\Sigma', \ell) = b$: the proof follows identically to the Bool case.

iii) $\text{pointsto}(\Sigma', \ell) = \lambda x : _ . e$: then by the operational semantics, $(\Sigma', \text{mon} \{ * \leftarrow \tau \} \ell) \rightarrow_N (\Sigma'[\ell' \mapsto (\ell, \text{some}(*, \tau))], \ell')$.

Therefore we want to show:

– $\Sigma'[\ell' \mapsto (\ell, \text{some}(*, \tau))] : (k - j - 2, \Psi'[\ell' \mapsto [* , \tau, \Psi'(\ell)]])$

– $(k - j - 2, \Psi'[\ell' \mapsto [* , \tau, \Psi'(\ell)]], \Sigma'[\ell' \mapsto (\ell, \text{some}(*, \tau))], \ell') \in \mathcal{V}^N \llbracket * \rrbracket$

The first condition follows from applications of IH 3) (smaller by index).

The second condition follows from an application of IH 1) (smaller by index).

iv) $\text{pointsto}(\Sigma', \ell) = \langle \ell_1, \ell_2 \rangle$:

By the operational semantics, either:

– $(\Sigma', \text{mon} \{ * \leftarrow \tau \} \ell) \rightarrow_N (\Sigma', \langle \text{mon} \{ * \leftarrow \text{fst}(\tau) \} \ell_1, \text{mon} \{ * \leftarrow \text{snd}(\tau) \} \ell_2 \rangle)$ or

– $(\Sigma', \text{mon} \{ * \leftarrow \tau \} \ell) \rightarrow_N (\Sigma', \text{TypeErr}(\tau, \ell))$

In the case it errors, we're done.

Otherwise, it suffices to show $(k - j - 1, \Psi', \Sigma', \langle \text{mon} \{ * \leftarrow \text{fst}(\tau) \} \ell_1, \text{mon} \{ * \leftarrow \text{snd}(\tau) \} \ell_2 \rangle) \in \mathcal{E}^N \llbracket * \rrbracket$.

By Lemma 5.18, it suffices to show:

– $(k - j - 1, \Psi', \Sigma', \text{mon} \{ * \leftarrow \text{fst}(\tau) \} \ell_1) \in \mathcal{E}^N \llbracket * \rrbracket$

– $(k - j - 1, \Psi', \Sigma', \text{mon} \{ * \leftarrow \text{snd}(\tau) \} \ell_2) \in \mathcal{E}^N \llbracket * \rrbracket$

We can unfold our hypothesis that $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^N \llbracket \tau \rrbracket$ to get $(k, \Psi, \Sigma, \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^N \llbracket \tau \rrbracket$.

We now have two cases depending on whether $\tau = *$ or $\tau_1 \times \tau_2$:

– If $\tau = *$, then $(k - 1, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N \llbracket * \rrbracket$ and $(k - 1, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N \llbracket * \rrbracket$.

By Lemma 5.15, $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket * \rrbracket$ and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N \llbracket * \rrbracket$.

Then we can apply IH 2) (smaller by index) to get what we need.

– If $\tau = \tau_1 \times \tau_2$, then $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

By Lemma 5.15, $(k - j - 1, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ and $(k - j - 1, \Psi', \Sigma', \ell_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

Then we can apply IH 2) (smaller by index) to get what we need.

3) We proceed by case analysis on τ' :

(a) $\tau' = \text{Nat}$: Since we already know $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^V \llbracket N \rrbracket \Psi(\ell)$, it suffices to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket \tau' \rrbracket$ and $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket \tau \rrbracket$.

This is immediate from $\vdash \Sigma'$, which implies $\tau' \propto \text{pointsto}(\Sigma', \ell')$ and $\tau \propto \text{pointsto}(\Sigma', \ell')$.

(b) $\tau' = \text{Int}$: same as the Nat case.

(c) $\tau' = \text{Bool}$: same as the Nat case.

(d) $\tau' = \tau'_1 \times \tau'_2$: this case is a contradiction by the fact that $\vdash \Sigma$.

(e) $\tau' = \tau'_1 \rightarrow \tau'_2$: Unfolding the relation in what we want to prove, let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.

Let τ_0 such that $\text{cod}(\tau') \leq \tau_0$.

Let ℓ_v such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \text{dom}(\tau') \rrbracket$.

We want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell' \ell_v) \in \mathcal{E}\mathcal{H}^N \llbracket \tau_0, \text{cod}(\tau), \text{cod}(\Psi'(\ell)) \rrbracket$.

We know by the OS that $(\Sigma', \text{app}\{\tau_0\} \ell' \ell_v) \rightarrow_N (\Sigma', \text{assert } \tau_0 (\ell' \ell_v)) \rightarrow_N (\Sigma', \text{assert } \tau_0 (\text{mon}\{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell' (\text{mon}\{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v))))$.

Note by downward closure, $\Sigma' : (j-2, \Psi')$.

By Lemma 5.10, it suffices to show $(j-2, \Psi', \Sigma', \text{assert } \tau_0 (\text{mon}\{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell' (\text{mon}\{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v)))) \in \mathcal{E}\mathcal{H}^N \llbracket \text{cod}(\tau'), \text{cod}(\tau), \text{cod}(\Psi'(\ell)) \rrbracket$

By Lemma 5.16, it suffices to show $(j-1, \Psi', \Sigma', \text{mon}\{\text{cod}(\tau') \leftarrow \text{cod}(\tau)\} (\ell' (\text{mon}\{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v))) \in \mathcal{E}\mathcal{H}^N \llbracket \text{cod}(\tau'), \text{cod}(\tau), \text{cod}(\Psi'(\ell)) \rrbracket$.

By IH 4) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', (\ell' (\text{mon}\{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v))) \in \mathcal{E}\mathcal{H}^N \llbracket \text{cod}(\Psi'(\ell)) \rrbracket$.

We now have two cases:

i) $\tau = *$: By Lemma 5.23, it suffices to show $(j, \Psi', \Sigma', \ell) \in \mathcal{E}\mathcal{H}^N \llbracket \Psi'(\ell) \rrbracket$ and $(j-1, \Psi', \Sigma', \text{mon}\{*\} \leftarrow \text{dom}(\tau')\} \ell_v) \in \mathcal{E}^N \llbracket * \rrbracket$ (since $\Psi'(\ell) = [\tau, \dots]$).

The first follows from the fact that $(j, \Psi', \Sigma', \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \Psi'(\ell) \rrbracket$ by Lemma 5.11.

For the second, by IH 2) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{E}^N \llbracket \text{dom}(\tau') \rrbracket$.

This follows by Lemma 5.15 applied to the fact that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \text{dom}(\tau') \rrbracket$.

ii) $\tau = \tau_1 \rightarrow \tau_2$:

By Lemma 5.22, it suffices to show $(j-1, \Psi', \Sigma', \ell) \in \mathcal{E}\mathcal{H}^N \llbracket \Psi'(\ell) \rrbracket$ and $(j-1, \Psi', \Sigma', \text{mon}\{\text{dom}(\tau) \leftarrow \text{dom}(\tau')\} \ell_v) \in \mathcal{E}^N \llbracket \text{dom}(\tau) \rrbracket$ (since $\Psi'(\ell) = [\tau, \dots]$).

The first follows from the fact that $(j-1, \Psi', \Sigma', \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \Psi'(\ell) \rrbracket$ by Lemma 5.11.

For the second, by IH 2) (smaller by index), it suffices to show $(j-1, \Psi', \Sigma', \ell_v) \in \mathcal{E}^N \llbracket \text{dom}(\tau') \rrbracket$.

This follows by Lemma 5.15 applied to the fact that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \text{dom}(\tau') \rrbracket$.

(f) $\tau' = *$: unfolding the relation in what we want to show, the proof follows by IH 3) (smaller by index).

4) Unfolding the expression relation in our hypothesis, we have that there are $(e', \Sigma'), j$ such that $(e, \Sigma) \xrightarrow{J}_N^j (e', \Sigma')$ with (e', Σ') irreducible.

If $e' = \text{Err}^\bullet$ then we're done, because the monitor will step to an error as well.

Otherwise, there is $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{V}\mathcal{H}^N \llbracket \bar{\tau} \rrbracket$.

This means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$, and $\Psi'(\ell) = \bar{\tau}$.

2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080

If $\neg \text{pointsto}(\Sigma', \ell) \propto \tau'$, then $(\Sigma, \text{mon } \{\tau' \Leftarrow \tau\} \ell) \longrightarrow_N^j (\Sigma', \text{mon } \{\tau' \Leftarrow \tau\} \ell) \longrightarrow_N (\Sigma', \text{TypeErr}(\tau', \ell))$, so we're done.

Otherwise, we have $\text{pointsto}(\Sigma', \ell) \propto \tau'$, and since $\text{pointsto}(\Sigma', \ell) \propto \tau$, we also have $\tau \propto \tau'$.

We want to show $(k - j, \Psi', \Sigma', \text{mon } \{\tau' \Leftarrow \tau\} \ell) \in \mathcal{E}\mathcal{H}^N \llbracket \tau', \tau, \Psi'(\ell) \rrbracket$.

We have three cases:

a) $\text{pointsto}(\Sigma', \ell) = i$: By OS, $(\Sigma', \text{mon } \{\tau' \Leftarrow \tau\} \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))], \ell')$.

Let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))]$ and $\Psi'' = \text{Psi}'[\ell' \mapsto \tau', \tau, \Psi(\ell)]$.

Unfolding the relation in what we want to show, it suffices to show $\forall \tau_z \in \Psi''(\ell), (k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N \llbracket \tau_z \rrbracket$ and $\Sigma'' : (k - j - 1, \Psi'')$.

For the second, we can apply IH 3) (smaller by index).

For the first, by downward closure, by Lemma 5.11, $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell) \rrbracket$.

Then we already know $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N \llbracket \tau_z \rrbracket$ when $\tau_z \in \Psi'(\ell)$.

So it suffices to show $(k - j - 1, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N \llbracket \tau' \rrbracket$.

If $\tau' = \text{Int}$, then we're done.

Otherwise, $\tau' = *$, in which case we need to show $(k - j - 2, \Psi'', \Sigma'', \ell) \in \mathcal{V}^N \llbracket \text{Int} \rrbracket$, which is also immediate.

b) $\text{pointsto}(\Sigma', \ell) = b$: essentially the same as the previous case.

c) $\Sigma'(\ell) = \langle \ell_1, \ell_2 \rangle$:

By OS, $(\Sigma', \text{mon } \{\tau' \Leftarrow \tau\} \ell) \longrightarrow_N (\Sigma', \langle \text{mon } \{\text{fst}(\tau') \Leftarrow \text{fst}(\tau)\} \ell_1, \text{mon } \{\text{snd}(\tau') \Leftarrow \text{snd}(\tau)\} \ell_2 \rangle)$.

Note by downward closure, $\Sigma' : (k - j - 2, \Psi')$.

By Lemma 5.19, it suffices to show $(k - j - 2, \Psi', \Sigma', \text{mon } \{\text{fst}(\tau') \Leftarrow \text{fst}(\tau)\} \ell_1) \in \mathcal{E}\mathcal{H}^N \llbracket \text{fst}(\tau'), \text{fst}(\tau), \text{fst}(\Psi'(\ell)) \rrbracket$

and $(k - j - 2, \Psi', \Sigma', \text{mon } \{\text{snd}(\tau') \Leftarrow \text{snd}(\tau)\} \ell_2) \in \mathcal{E}\mathcal{H}^N \llbracket \text{snd}(\tau'), \text{snd}(\tau), \text{snd}(\Psi'(\ell)) \rrbracket$.

Both of these follow by unfolding the relation in the hypothesis about ℓ , applying Lemma 5.14, and applying IH 4) (smaller by index).

d) $\text{pointsto}(\Sigma', \ell) = \lambda x : _ . e$:

By OS, $(\Sigma', \text{mon } \{\tau' \Leftarrow \tau\} \ell) \longrightarrow_N (\Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))], \ell')$, where $\ell' \notin \text{dom}(\Sigma')$.

Then let $\Sigma'' = \Sigma'[\ell' \mapsto (\ell, \text{some}(\tau', \tau))]$ and let $\Psi'' = \Psi'[\ell' \mapsto \tau', \tau, \Psi'(\ell)]$.

By IH 3) (smaller by index) we get $(k - j - 2, \Psi'', \Sigma'', \ell') \in \mathcal{V}\mathcal{H}^N \llbracket \tau', \tau, \Psi'(\ell) \rrbracket$, so all that's left is to show is $\Sigma'' : (k - j - 2, \Psi'')$.

Let $k' < k - j - 2$.

Note by downward closure, $\Sigma' : (k', \Psi')$, so $\forall \ell'' \in \text{dom}(\Sigma')$, by Lemma 5.11, $(k', \Psi'', \Sigma'', \ell'') \in \mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell'') \rrbracket$ (note $\Psi'(\ell'') = \Psi''(\ell'')$).

So the final condition is $(k', \Psi'', \Sigma'', \ell') \in \mathcal{V}\mathcal{H}^N \llbracket \Psi''(\ell') \rrbracket$, which follows from IH 3) (smaller by index).

□

5.3.3 Compatability Lemmas

LEMMA 5.26 (**T-VAR COMPATIBILITY**).
$$\frac{\llbracket (x:\tau) \in \Gamma \rrbracket}{\llbracket \Gamma \vdash x : \tau \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(x)) \in \mathcal{E}^N \llbracket \tau \rrbracket$.

Since $x : \tau \in \Gamma$, we get that $\gamma(x) = \ell$.

Since $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$, we get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \tau \rrbracket$.

Then we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^N \llbracket \tau \rrbracket$ immediately since ℓ is already a value and we have as a premise that $\Sigma : (k, \Psi)$. \square

LEMMA 5.27 (**T-NAT COMPATIBILITY**).
$$\frac{}{\llbracket \Gamma \vdash n : \text{Nat} \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(n)) \in \mathcal{E}^N \llbracket \text{Nat} \rrbracket$.

Note $\gamma(n) = n$.

By the OS, we have $(\Sigma, n) \rightarrow_N (\Sigma[\ell \mapsto (n, _)], \ell)$.

We get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^N \llbracket \text{Nat} \rrbracket$ immediately because $n \in \mathbb{N}$.

Since $\mathcal{V}^N \llbracket \text{Nat} \rrbracket$ does not rely on Ψ or Σ , we have that $(k, \Psi[\ell \mapsto [\text{Nat}]], \Sigma[\ell \mapsto (n, _)], \ell) \in \mathcal{V}^N \llbracket \text{Nat} \rrbracket$. \square

LEMMA 5.28 (**T-INT COMPATIBILITY**).
$$\frac{}{\llbracket \Gamma \vdash i : \text{Int} \rrbracket}$$

PROOF. Not meaningfully different from **T-Int**. \square

LEMMA 5.29 (**T-TRUE COMPATIBILITY**).
$$\frac{}{\llbracket \Gamma_1 \vdash \text{True} : \text{Bool} \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{True})) \in \mathcal{E}^N \llbracket \text{Bool} \rrbracket$.

Note $\gamma(\text{True}) = \text{True}$.

By the OS, we have $(\Sigma, \text{True}) \rightarrow_N (\Sigma[\ell \mapsto (\text{True}, _)], \ell)$.

We get $(k, \Psi, \Sigma, \text{True}) \in \mathcal{V}^N \llbracket \text{Bool} \rrbracket$ immediately.

Since $\mathcal{V}^N \llbracket \text{Bool} \rrbracket$ does not rely on Ψ or Σ , we have that $(k, \Psi[\ell \mapsto [\text{Bool}]], \Sigma[\ell \mapsto (\text{True}, _)], \ell) \in \mathcal{V}^N \llbracket \text{Bool} \rrbracket$. \square

LEMMA 5.30 (**T-FALSE COMPATIBILITY**).
$$\frac{}{\llbracket \Gamma_1 \vdash \text{False} : \text{Bool} \rrbracket}$$

PROOF. Not meaningfully different from the previous case. \square

LEMMA 5.31 (**T-LAM COMPATIBILITY**).
$$\frac{\llbracket \Gamma_1, (x_1:\tau_1) \vdash e_1 : \tau_2 \rrbracket}{\llbracket \Gamma_1 \vdash \lambda(x_1:\tau_1). e_1 : \tau_1 \rightarrow \tau_2 \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\lambda x_1 : \tau_1. e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rightarrow \tau_2 \rrbracket$.

2133 Note that $\gamma(\lambda x_1 : \tau_1. e_1) = \lambda x_1 : \tau_1. \gamma(e_1)$.

2134 Since $\lambda x_1 : \tau_1. \gamma(e_1)$ is a value, by the OS we have $(\Sigma, \lambda x_1 : \tau_1. \gamma(e_1)) \longrightarrow_N (\Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})])$, where
2135 $\ell \notin \text{dom}(\Sigma)$.

2136 We choose our later Ψ' to be $\Psi[\ell \mapsto \tau_1 \rightarrow \tau_2]$.

2137 We now have two obligations:

- 2138
- 2139
- 2140 (1) $(k - 1, \Psi[\ell \mapsto \tau_1 \rightarrow \tau_2], \Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})], \ell) \in \mathcal{V}^N \llbracket \tau_1 \rightarrow \tau_2 \rrbracket$
- 2141 (2) $\Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})] : (k - 1, \Psi[\ell \mapsto \tau_1 \rightarrow \tau_2])$
- 2142

2143 For 1), unfolding the value relation:

2144 Let $(j, \Psi') \sqsupseteq (k - 1, \Psi[\ell \mapsto \tau_1 \rightarrow \tau_2])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})]$ such that $\Sigma' : (j, \Psi')$.

2145 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

2146 Let $\tau_0 \geq \tau_2$.

2147 We want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

2148 By Lemma 5.17, it suffices to show $(j - 1, \Psi', \Sigma', \ell \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

2149 By the OS, $(\Sigma', \ell \ell_v) \longrightarrow_N (\Sigma', \gamma(e_1)[\ell_v/x])$.

2150 By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.

2151 Note that $(j - 1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^N \llbracket \Gamma, x : \tau_1 \rrbracket$:

- 2152
- 2153 i) $(j - 1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ by Lemma 5.15.
- 2154 ii) $\forall y \in \text{dom}(\gamma), (j - 1, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^N \llbracket \Gamma(y) \rrbracket$ by the premise about γ and Lemma 5.15.
- 2155

2156 Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma'$, and e_1 at $j-1$ to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2157 Finally, we can apply Lemma 5.10 to get $(j - 1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$ which is what we wanted to show.

2162 For 2), first note the domains are equal, since $\text{dom}(\Sigma) = \text{dom}(\Psi)$.

2163 Then note $\vdash \Sigma[\ell \mapsto \lambda x_1 : \tau_1. \gamma(e_1)]$ since $\vdash \Sigma$.

2164 Then let $j < k - 1$ and let $\ell' \in \text{dom}(\Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})])$.

2165 If $\ell' \neq \ell$, then we get the remaining conditions from $\Sigma : (k, \Psi)$ and Lemma 5.11.

2166 If $\ell' = \ell$, then note the structural obligation on $\Psi[\ell \mapsto [\tau_1 \rightarrow \tau_2]]$ is immediate.

2167 We want to show $(j, \Psi[\ell \mapsto \tau_1 \rightarrow \tau_2], \Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})], \ell) \in \mathcal{V}^N \llbracket \tau_1 \rightarrow \tau_2 \rrbracket$.

2168 Let $(j, \Psi') \sqsupseteq (k - 1, \Psi[\ell \mapsto \tau_1 \rightarrow \tau_2])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : \tau_1. \gamma(e_1), \text{none})]$ such that $\Sigma' : (j, \Psi')$.

2169 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

2170 Let $\tau_0 \geq \tau_2$.

2171 By inspection of the value relation, we get immediately that $\Sigma'(\ell_v) \propto \tau_1$, so we want to show $(j, \Psi', \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

2172 By Lemma 5.17, it suffices to show $(j - 1, \Psi', \Sigma', \ell \ell_v) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$.

2173 By the OS, $(\Sigma', \ell \ell_v) \longrightarrow_N (\Sigma', \gamma(e_1)[\ell_v/x])$.

2174 By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.

2175 Note that $(j - 1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^N \llbracket \Gamma, x : \tau_1 \rrbracket$:

- 2176
- 2177 i) $(j - 1, \Psi', \Sigma', \ell_v) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ by Lemma 5.15.
- 2178 ii) $\forall y \in \text{dom}(\gamma), (j - 1, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^N \llbracket \Gamma(y) \rrbracket$ by the premise about γ and Lemma 5.15.
- 2179

2184

2185 Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma',$ and e_1 at $j-1$ to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2186 Then we can apply Lemma 5.24 to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2187 Finally, we can apply Lemma 5.10 to get $(j-1, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^N \llbracket \tau_0 \rrbracket$ which is what we wanted to show.

2188 \square

2189

2190

2191

$$\text{LEMMA 5.32 (T-PAIR COMPATIBILITY). } \frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \rrbracket \quad \llbracket \Gamma_1 \vdash e_2 : \tau_2 \rrbracket}{\llbracket \Gamma_1 \vdash \langle e_1, e_2 \rangle : \tau_1 \times \tau_2 \rrbracket}$$

2192

2193

2194

2195 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

2196 We want to show $(k, \Psi, \Sigma, \gamma(\langle e_1, e_2 \rangle)) \in \mathcal{E}^N \llbracket \tau_1 \times \tau_2 \rrbracket$.

2197 Note $\gamma(\langle e_1, e_2 \rangle) = \langle \gamma(e_1), \gamma(e_2) \rangle$.

2198 We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

2199 We can apply the second hypothesis to get $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2200 Then by Lemma 5.19, $(k, \Psi, \Sigma, \langle \gamma(e_1), \gamma(e_2) \rangle) \in \mathcal{E}^N \llbracket \tau_1 \times \tau_2 \rrbracket$, which is what we wanted to show. \square

2201

2202

$$\text{LEMMA 5.33 (T-APP COMPATIBILITY). } \frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \rightarrow \tau_2 \rrbracket \quad \llbracket \Gamma_1 \vdash e_2 : \tau_1 \rrbracket}{\llbracket \Gamma_1 \vdash \text{app}\{\tau_2\} e_1 e_2 : \tau_2 \rrbracket}$$

2203

2204

2205

2206 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

2207 We want to show $(k, \Psi, \Sigma, \gamma(\text{app}\{\tau_2\} e_1 e_2)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2208 Note $\gamma(\text{app}\{\tau_2\} e_1 e_2) = \text{app}\{\tau_2\} \gamma(e_1) \gamma(e_2)$.

2209 By the first hypothesis we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rightarrow \tau_2 \rrbracket$.

2210 By the second hypothesis we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

2211 Then we can apply Lemma 5.20 to get $(k, \Psi, \Sigma, \text{app}\{\tau_2\} \gamma(e_1) \gamma(e_2)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$ which is what we wanted to show. \square

2212

2213

$$\text{LEMMA 5.34 (T-FST COMPATIBILITY). } \frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \times \tau_2 \rrbracket}{\llbracket \Gamma_1 \vdash \text{fst}\{\tau_1\} e_1 : \tau_1 \rrbracket}$$

2214

2215

2216

2217 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma_1 \rrbracket$ such that $\Sigma : (k, \Psi)$.

2218 We want to show $(k, \Psi, \Sigma, \gamma(\text{fst}\{\tau_1\} e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

2219 Note $\gamma(\text{fst}\{\tau_1\} e_1) = \text{fst}\{\tau_1\} \gamma(e_1)$.

2220 From the first hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \times \tau_2 \rrbracket$.

2221 Unfolding the expression relation, there are j, Σ', e'_1 such that $(\Sigma, \gamma(e_1)) \rightarrow_N^j (\Sigma', e'_1)$ and e'_1 is irreducible.

2222 If $e'_1 = \text{Err}^\bullet$ then we're done because the projection also steps to an error.

2223 Otherwise, there is a $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e'_1) \in \mathcal{V}^N \llbracket \tau_1 \times \tau_2 \rrbracket$.

2224 Unfolding the location and value relations, we get that $\Sigma'(e'_1) = \langle \ell_1, \ell_2 \rangle$.

2225 By the OS, $(\Sigma, \text{fst}\{\tau_1\} e_1) \rightarrow_N^j (\Sigma', \text{fst}\{\tau_1\} e'_1) \rightarrow_N (\Sigma', \text{assert } \tau_1 \ell_1) \rightarrow_N (\Sigma', \ell_1)$.

2226 We can apply Lemma 5.15 to the premise that $(k-j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$ to get $(k-j-2, \Psi', \Sigma', \ell_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

2227 Finally, we can apply Lemma 5.11 to get that $\Sigma' : (k-j-2, \Psi')$, which is sufficient to complete the proof. \square

2228

2229

2230

2231

$$\text{LEMMA 5.35 (T-SND COMPATIBILITY). } \frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \times \tau_2 \rrbracket}{\llbracket \Gamma_1 \vdash \text{snd}\{\tau_2\} e_1 : \tau_2 \rrbracket}$$

2232

2233

2234

2235 **PROOF.** Not meaningfully different from the previous lemma. \square

$$\begin{array}{l} 2237 \\ 2238 \\ 2239 \\ 2240 \\ 2241 \end{array} \quad \frac{\begin{array}{c} \llbracket \Gamma_1 \vdash e_1 : \tau_1 \rrbracket \quad \llbracket \Gamma_1 \vdash e_2 : \tau_2 \rrbracket \\ \Delta(\text{binop}, \tau_1, \tau_2) = \tau_3 \end{array}}{\llbracket \Gamma_1 \vdash \text{binop } e_1 e_2 : \tau_3 \rrbracket}$$

2242 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

2243 We want to show $(k, \Psi, \Sigma, \gamma(\text{binop } e_1 e_2)) \in \mathcal{E}^N \llbracket \tau_3 \rrbracket$.

2244 Note $\gamma(\text{binop } e_1 e_2) = \text{binop } \gamma(e_1) \gamma(e_2)$.

2245 By the first hypothesis applied to γ we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

2246 Unfolding we get there are j, Σ', e'_1 such that $(\Sigma, \gamma(e_1)) \rightarrow_N^j (\Sigma', e'_1)$ and e'_1 is irreducible.

2247 If $e'_1 = \text{Err}^\bullet$ then we're done, because the whole operation errors.

2248 Otherwise there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_1) \in \mathcal{V}^N \llbracket \tau_1 \rrbracket$.

2251 Note by Lemma 5.15 and Lemma 5.11, we have $(k - j, \Psi', \Sigma', \gamma) \in \mathcal{G}^N \llbracket \Gamma_1 \rrbracket$ and $\Sigma' : (k - j, \Psi')$.

2252 By the second hypothesis applied to γ we have $(k - j, \Psi', \Sigma', \gamma(e_2)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

2253 Unfolding we get there are j', Σ'', e'_2 such that $(\Sigma', \gamma(e_2)) \rightarrow_N^{j'} (\Sigma'', e'_2)$ and e'_2 is irreducible.

2254 If $e'_2 = \text{Err}^\bullet$ then we're done, because the whole operation errors.

2255 Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e'_2) \in \mathcal{V}^N \llbracket \tau_2 \rrbracket$.

2256 From the definition of Δ , $\tau_3 = \text{Int}$ or Nat the cases proceed identically, so without loss of generality assume $\tau_3 = \text{Int}$.

2257 $\tau_1 = \tau_2 = \text{Int}$, and therefore $\Sigma''(e'_1) = i_1$ and $\Sigma''(e'_2) = i_2$.

2258 If $\text{binop} = \text{quotient}$ and $i_2 = 0$ then $(\Sigma'', \text{binop } e'_1 e'_2) \rightarrow_N (\Sigma'', \text{DivErr})$, so we're done.

2259 If $\text{binop} = \text{quotient}$ and $i_2 \neq 0$, then $(\Sigma'', \text{binop } e'_1 e'_2) \rightarrow_N (\Sigma'', i_1/i_2) \rightarrow_N (\Sigma''[\ell \mapsto (i_1/i_2, \text{none})], \ell)$.

2260 Since $i_1/i_2 \in \mathbb{Z}$, we're done.

2261 If $\text{binop} = \text{sum}$ then $(\Sigma'', \text{binop } e'_1 e'_2) \rightarrow_N (\Sigma'', i_1 + i_2) \rightarrow_N (\Sigma''[\ell \mapsto (i_1 + i_2, \text{none})], \ell)$.

2262 Since $i_1 + i_2 \in \mathbb{Z}$, we're done. □

$$\begin{array}{l} 2270 \\ 2271 \\ 2272 \\ 2273 \\ 2274 \\ 2275 \\ 2276 \end{array} \quad \frac{\begin{array}{c} \llbracket \Gamma_1 \vdash e_1 : \text{Bool} \rrbracket \\ \llbracket \Gamma_1 \vdash e_2 : \tau \rrbracket \\ \llbracket \Gamma_1 \vdash e_3 : \tau \rrbracket \end{array}}{\llbracket \Gamma_1 \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : \tau \rrbracket}$$

2277 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

2278 We want to show $(k, \Psi, \Sigma, \gamma(\text{if } e_1 \text{ then } e_2 \text{ else } e_3)) \in \mathcal{E}^N \llbracket \tau \rrbracket$.

2279 Note $\gamma(\text{if } e_1 \text{ then } e_2 \text{ else } e_3) = \text{if } \gamma(e_1) \text{ then } \gamma(e_2) \text{ else } \gamma(e_3)$.

2280 From the first hypothesis applied to γ , we know $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \text{Bool} \rrbracket$.

2281 Unfolding, we have that there is Σ', e'_1, j such that $(\Sigma, \gamma(e_1)) \rightarrow_N^j (\Sigma', e'_1)$ where e'_1 is irreducible.

2282 If $e'_1 = \text{Err}^\bullet$ then we're done, because the entire if statement errors.

2283 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_1) \in \mathcal{V}^N \llbracket \text{Bool} \rrbracket$.

2284 Unfolding the location and then the value relation, we get that $\text{pointsto}(\Sigma', e'_1) = \text{True}$ or $\text{pointsto}(\Sigma', e'_1) = \text{False}$.

- $\text{pointsto}(\Sigma', e'_1) = \text{True}$: Note by OS, $(\Sigma, \text{if } \gamma(e_1) \text{ then } \gamma(e_2) \text{ else } \gamma(e_3)) \longrightarrow_N^j (\Sigma', \text{if } e'_1 \text{ then } \gamma(e_2) \text{ else } \gamma(e_3)) \longrightarrow_N (\Sigma', \gamma(e_2))$.

By Lemma 5.15 and Lemma 5.11, we have $(k - j - 1, \Psi', \Sigma', \gamma) \in \mathcal{G}^N \llbracket \Gamma_1 \rrbracket$ and $\Sigma' : (k - j - 1, \Psi')$.

From the second hypothesis, we get $(k - j - 1, \Psi', \Sigma', \gamma(e_2)) \in \mathcal{E}^N \llbracket \tau \rrbracket$, which is sufficient to complete the proof.

- $\text{pointsto}(\Sigma', e'_1) = \text{False}$: same as other case except replace e_2 with e_3 .

□

LEMMA 5.38 (T-CAST COMPATIBILITY).
$$\frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \rrbracket}{\llbracket \Gamma_1 \vdash \text{cast } \{ \tau_2 \Leftarrow \tau_1 \} e_1 : \tau_2 \rrbracket}}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{cast } \{ \tau_2 \Leftarrow \tau_1 \} e_1)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

Note $\gamma(\text{cast } \{ \tau_2 \Leftarrow \tau_1 \} e_1) = \text{cast } \{ \tau_2 \Leftarrow \tau_1 \} \gamma(e_1)$.

By the operational semantics, $(\Sigma, \text{cast } \{ \tau_2 \Leftarrow \tau_1 \} \gamma(e_1)) \longrightarrow_N (\Sigma, \text{mon } \{ \tau_2 \Leftarrow \tau_1 \} e_1)$.

By Lemma 5.11 and Lemma 5.15, $(k - 1, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ and $\Sigma : (k - 1, \Psi)$.

By the hypothesis, $(k - 1, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

By Lemma 5.25, $(k - 1, \Psi, \Sigma, \text{mon } \{ \tau_2 \Leftarrow \tau_1 \} e_1) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$, which is sufficient to complete the proof.

□

LEMMA 5.39 (T-SUB COMPATIBILITY).
$$\frac{\llbracket \Gamma_1 \vdash e_1 : \tau_1 \rrbracket \quad \tau_1 \leq \tau_2}{\llbracket \Gamma_1 \vdash e_1 : \tau_2 \rrbracket}}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^N \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_2 \rrbracket$.

From our hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^N \llbracket \tau_1 \rrbracket$.

We can apply Lemma 5.10 to finish the case.

□

5.3.4 Fundamental Property / Vigilance

THEOREM 5.40 (VIGILANCE). *If $\Gamma \vdash e : \tau$ then $\llbracket \Gamma \vdash e : \tau \rrbracket^N$*

PROOF. By induction over the typing derivation, using the compatibility lemmas.

□

5.4 Vigilance Fundamental Property for Transient with Truer Transient Typing

In this subsection, we use $\Gamma \vdash e : \tau$ to mean $\Gamma \vdash_{\text{tru}} e : \tau$.

The relation needs to be extended with a case to handle \perp :

$$\mathcal{V}^L \llbracket \perp \rrbracket = \emptyset$$

We also edit the function cases of the relation to insert a tag into the annotation of the app, and produce a value in the meet of the tag and the result type:

$$\mathcal{V}^L \llbracket * \rightarrow \tau_1', \tau_2, \dots, \tau_n \rrbracket = \{(k, \Psi, \Sigma, \ell) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi), \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi'). \forall K.$$

$$\forall \ell_v \text{ where } (j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^L \llbracket * \rrbracket.$$

$$(j, \Psi' \Sigma', \text{app}\{\tau_0\} \ell \ell_v) \in \mathcal{E}^L \llbracket [\tau_1' \sqcap K, \text{cod}(\tau_2), \dots, \text{cod}(\tau_n)] \rrbracket \}$$

$$\mathcal{V}^L \llbracket * \rightarrow \tau_2 \rrbracket = \{(k, \Psi, \Sigma, w) \mid \forall (j, \Psi') \sqsupseteq (k, \Psi). \forall \Sigma' \supseteq \Sigma \text{ where } \Sigma' : (j, \Psi').$$

$$\forall \ell \text{ where } (j, \Psi', \Sigma', \ell) \in \mathcal{V}^L \llbracket * \rrbracket. \forall K.$$

$$(j + 1, \Psi', \Sigma', \text{app}\{K\} w \ell) \in \mathcal{E}^L \llbracket \tau_2 \sqcap K \rrbracket \}$$

We also need to edit the $\Sigma : (k, \Psi)$ judgement because we no longer have or need a correspondance between the from type of a guard and the type underneath the guard:

$$\Sigma : (k, \Psi) \triangleq \text{dom}(\Sigma) = \text{dom}(\Psi) \wedge \vdash \Sigma \wedge \forall j < k, \ell \in \text{dom}(\Sigma). ((j, \Psi, \Sigma, \ell) \in \mathcal{V}^L \llbracket \Psi(\ell) \rrbracket$$

$$\wedge (\Sigma(\ell) = (\ell', \text{some}(\tau, \tau')) \Rightarrow \Psi(\ell) = [\tau, \tau', \Psi(\ell')]) \wedge$$

$$\wedge (\Sigma(\ell) = (v, \text{none}) \wedge v \notin \mathbb{L} \Rightarrow \exists \tau. \Psi(\ell) = [\tau])$$

5.4.1 Lemmas Used Without Mention

LEMMA 5.41 (STEPPING TO ERROR IMPLIES EXPRESSION RELATION). *If $(\Sigma, e) \xrightarrow{j}_T (\Sigma', \text{Err}^\bullet)$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$*

PROOF. If $k < j$, then we're done because the condition in the expression relation is vacuously true.

Otherwise, we can use j as our steps, Σ' as our ending value log, and Err^\bullet as our irreducible expression, and we satisfy the condition in the expression relation. \square

LEMMA 5.42 (STEPPING TO ERROR IMPLIES EXPRESSION HISTORY). *If $(\Sigma, e) \xrightarrow{j}_T (\Sigma', \text{Err}^\bullet)$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \bar{\tau} \rrbracket$*

PROOF. Similar to the previous proof. \square

LEMMA 5.43 (ANTI-REDUCTION - HEAD EXPANSION - EXPRESSION RELATION COMMUTES WITH STEPS). *If $(k, \Psi', \Sigma', e') \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $(\Sigma, e) \xrightarrow{j}_T (\Sigma', e')$ and $\Sigma' : (k, \Psi')$ then $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$*

PROOF. Unfolding the expression relation in our hypothesis, there exists $(\Sigma'', e''), j'$ such that $(\Sigma', e') \xrightarrow{j'}_T (\Sigma'', e'')$ and (Σ'', e'') is irreducible.

Either $e'' = \text{Err}^\bullet$, in which case $(\Sigma, e) \xrightarrow{j+j'}_T (\Sigma'', \text{Err}^\bullet)$, so we're done.

Otherwise, there is a $(k - j', \Psi'') \sqsupseteq (k, \Psi')$ such that $\Sigma'' : (k - j', \Psi'')$, and $(k - j', \Psi'', \Sigma'', e'') \in \mathcal{V}^T \llbracket \tau \rrbracket$.

Using this information, we can show $(k + j, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ by noting $(\Sigma, e) \xrightarrow{j+j'}_T (\Sigma'', e'')$. \square

2393 LEMMA 5.44 (ANTI-REDUCTION - HEAD EXPANSION - EXPRESSION HISTORY COMMUTES WITH STEPS). *If $(k, \Psi', \Sigma', e') \in$*
 2394 *$\mathcal{E}\mathcal{H}^T \llbracket \bar{\tau} \rrbracket$ and $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and $\Sigma' : (k, \Psi')$ then $(k + j, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \bar{\tau} \rrbracket$*
 2395

2396 PROOF. Similar to the previous proof. □

2398 LEMMA 5.45 (THE OPERATIONAL SEMANTICS PRESERVES WELL FORMED VALUE LOGS). *If $\vdash \Sigma$ and $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$*
 2399 *then $\vdash \Sigma'$.*
 2400

2401 PROOF. The proof is immediate by inspection of the Operational Semantics. □

2403 LEMMA 5.46 (NOT ENOUGH STEPS IMPLIES ANY EXPRESSION RELATION). *If $(\Sigma, e) \longrightarrow_T^k (\Sigma', e')$ and (Σ', e') is not*
 2404 *irreducible, then $\forall j \leq k. (j, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $(j, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \tau \rrbracket$.*
 2405

2406 PROOF. Both conclusions are immediate, since the implications in the relations are vacuously true. □

2408 LEMMA 5.47 (THE OPERATIONAL SEMANTICS ONLY GROWS STORES). *If $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$ then $\Sigma' \supseteq \Sigma$.*

2410 PROOF. This is a corollary of Lemma 5.48. □

2412 5.4.2 Lemmas Used With Mention

2414 LEMMA 5.48 (THE OPERATIONAL SEMANTICS PRODUCES VALUE LOG EXTENSIONS). *If $(\Sigma, e) \longrightarrow_T^* (\Sigma', e')$, then $\exists \bar{\ell} \subseteq$*
 2415 *$\text{dom}(\Sigma')$ such that $\bar{\ell} \notin \text{dom}(\Sigma)$ and $\Sigma' = \Sigma[\bar{\ell} \mapsto (v, _)]$.*
 2416

2417 PROOF. By inspection of the Operational Semantics, no steps modify the value stored in the value log, meaning
 2418 $\Sigma' \supseteq \Sigma$.

2420 And also by the inspection of the Operational Semantics, there is exactly one rule to allocate new entries in the value
 2421 log, meaning $\Sigma' \setminus \Sigma$ is a suitable choice for $[\bar{\ell} \mapsto (v, _)]$. □

2423 LEMMA 5.49 (STEPS ARE PRESERVED IN FUTURE VALUE LOGS). *If $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ and $\bar{\ell} \notin \text{dom}(\Sigma')$ then $(\Sigma[\bar{\ell} \mapsto (v, _)], e) \longrightarrow_T^j$*
 2424 *$(\Sigma'[\bar{\ell} \mapsto (v, _)], e')$.*
 2425

2426 PROOF. Since all of the added locations are not in Σ' , and therefore also not in Σ , no rule that will lookup a label in
 2427 the derivation tree for $(\Sigma, e) \longrightarrow_T^j (\Sigma', e')$ will find a different value or type.

2428 The only remaining notable reduction steps are those that allocate a new label and value entry, but since $\bar{\ell} \notin \text{dom}(\Sigma')$,
 2429 we can allocate the same entry unchanged. □

2431 LEMMA 5.50 (SUBTYPING PRESERVES LOGICAL RELATIONS). $\forall \Sigma, k, \Psi, \tau, \tau'$. *where $\Sigma : (k, \Psi)$ and $\tau \leq \tau'$.*

2433 (1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau' \rrbracket$*

2434 (2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau' \rrbracket$*

2435 (3) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \tau', \bar{\tau} \rrbracket$*

2436 (4) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \tau', \bar{\tau} \rrbracket$*
 2437
 2438

2439 PROOF. Proceed by mutual induction on k and τ :

2440 • $k = 0$: Both 1 and 3 are immediate if $e \neq \ell$.

2441 If $e = \ell$ then 1 and 3 follow immediately from 2 and 4.

2442 2 and 4 follow identically in the $k = 0$ case as they do in the $k > 0$ case, but the function case is vacuously true.

- 2445 • $k > 0$:
- 2446 (1) Unfolding our hypothesis, there is some (Σ', e') , j such that $(\Sigma, e) \xrightarrow{j}_T (\Sigma', e')$.
- 2447 If $e' = \text{Err}^\bullet$ then we're done.
- 2448 Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi')$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau \rrbracket$.
- 2449 We now have two obligations:
- 2450 a) $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau' \rrbracket$.
- 2451 b) $\Sigma' : (k - j, \Psi')$.
- 2452 For a) by IH 2) (not necessarily smaller by type or index), we have $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau' \rrbracket$, which is
- 2453 what we wanted to show.
- 2454 For b), this is immediate from the premise.
- 2455 (2) Case split on $\tau \leqslant \tau'$:
- 2456 i) $\tau \leqslant \tau$: immediate.
- 2457 ii) $\text{Nat} \leqslant \text{Int}$: immediate because $\mathbb{T} \subseteq \mathbb{Z}$.
- 2458 iii) $\tau_1 \times \tau_2 \leqslant \tau'_1 \times \tau'_2$, with $\tau_1 \leqslant \tau'_1$ and $\tau_2 \leqslant \tau'_2$:
- 2459 We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.
- 2460 Unfolding our hypothesis, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.
- 2461 We want to show $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau'_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau'_2 \rrbracket$.
- 2462 We can apply IH 2) (smaller by type) to both of these judgements to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau'_1 \rrbracket$ and
- 2463 $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau'_2 \rrbracket$.
- 2464 This is sufficient to show $(k, \Psi, \Sigma, \Sigma(\ell)) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.
- 2465 iv) $* \rightarrow \tau_2 \leqslant * \rightarrow \tau'_2$, with $\tau_2 \leqslant \tau'_2$:
- 2466 We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.
- 2467 Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
- 2468 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket * \rrbracket$.
- 2469 Let K .
- 2470 We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket \tau'_2 \sqcap K \rrbracket$.
- 2471 Then, we can apply our hypothesis about ℓ to get $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket \tau_2 \sqcap K \rrbracket$.
- 2472 Finally, we can apply IH 1) (smaller by type) to get $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket \tau'_2 \sqcap K \rrbracket$ which is
- 2473 what we wanted to show.
- 2474 (3) Unfolding our hypothesis, we get that there are some (Σ', e') , j such that $(\Sigma, e) \xrightarrow{j}_T (\Sigma', e')$ and (Σ', e')
- 2475 are irreducible.
- 2476 If $e' = \text{Err}^\bullet$, then we're done.
- 2477 Otherwise, there is some $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$,
- 2478 which means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.
- 2479 Then by IH 4) (not necessarily smaller by type or index) with $\tau \leqslant \tau'$, we get $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \tau', \bar{\tau} \rrbracket$,
- 2480 which is what we wanted to show.
- 2481 (4) Unfolding the history relation, we want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \tau', \bar{\tau} \rrbracket$.
- 2482 We case split on $\tau \leqslant \tau'$:
- 2483 i) $\tau = \tau'$: immediate by premise.
- 2484
- 2485
- 2486
- 2487
- 2488
- 2489
- 2490
- 2491
- 2492
- 2493
- 2494
- 2495
- 2496

2497 ii) $\text{Nat} \leq \text{Int}$:

2498 by our premise, we already get that $\forall \tau_o \in \bar{\tau}, (k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau_o \rrbracket$.

2499 Therefore, it suffices to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \text{Int} \rrbracket$ given $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \text{Nat} \rrbracket$ which is immedi-
2500 ate since $\mathbb{T} \subset \mathbb{Z}$.

2501
2502 iii) $\tau_1 \times \tau_2 \leq \tau'_1 \times \tau_2$ with $\tau_1 \leq \tau'_1$ and $\tau_2 \leq \tau'_2$:

2503 by our premise, we get that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$ and $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_1, \text{fst}(\bar{\tau}) \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in$
2504 $\mathcal{V}^T \llbracket \tau_2, \text{snd}(\bar{\tau}) \rrbracket$.

2505 We can apply IH 4) (smaller by type) to both to get $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau'_1, \text{fst}(\bar{\tau}) \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in$
2506 $\mathcal{V}^T \llbracket \tau'_2, \text{snd}(\bar{\tau}) \rrbracket$, which is what we wanted to show.

2507
2508 iv) $* \rightarrow \tau_2 \leq * \rightarrow \tau'_2$ with $\tau_2 \leq \tau'_2$:

2509 unfolding what we want to show, let $\Sigma' \supseteq \Sigma, (j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (j, \Psi')$.

2510 Let $\ell_o \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_o) \in \mathcal{V}^T \llbracket * \rrbracket$.

2511 Let K .

2512 We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau' \sqcap K, \text{cod}(\bar{\tau}) \rrbracket$.

2513
2514 We can then apply the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau, \bar{\tau} \rrbracket$ to get $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau \sqcap K, \text{cod}(\bar{\tau}) \rrbracket$.

2515 Then we can apply IH 3) (smaller by type) to get $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau' \sqcap K, \text{cod}(\bar{\tau}) \rrbracket$,
2516 which is what we wanted to show.

2517 \square

2518
2519 LEMMA 5.51 (RV-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$
2520 and $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \bar{\tau} \rrbracket$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^T \llbracket \bar{\tau} \rrbracket$*

2521
2522 PROOF. We want to show $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^T \llbracket \bar{\tau} \rrbracket$.

2523 Let τ be the head of $\bar{\tau}$ so that $\bar{\tau} = [\tau, \dots]$.

2524 We proceed by induction over k and τ :

2525 • $k = 0$: The function and dynamic cases are vacuously true, and the rest follow as in the other case.

2526 • $k > 0$:

2527 i) $\tau = \text{Int}$: immediate because $\Sigma(\ell) = \Sigma'(\ell)$.

2528 ii) $\tau = \text{Nat}$: same as previous case.

2529 iii) $\tau = \text{Bool}$: same as previous case.

2530 iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

2531 We want to show $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^T \llbracket \tau_1, \overline{\text{fst}(\bar{\tau})} \rrbracket$ and $(k - j, \Psi', \Sigma', \ell_2) \in \mathcal{V}^T \llbracket \tau_2, \overline{\text{snd}(\bar{\tau})} \rrbracket$.

2532 We have $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_1, \overline{\text{fst}(\bar{\tau})} \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau_2, \overline{\text{snd}(\bar{\tau})} \rrbracket$.

2533 Both follow by IH (smaller by type).

2534 v) $\tau = * \rightarrow \tau_2$:

2535 Let $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$ such that $\Sigma''(j', \Psi')$.

2536 Let $\ell_o \in \text{dom}(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_o) \in \mathcal{V}^T \llbracket * \rrbracket$.

2537 Let K .

2538 We want to show $(j', \Psi'', \Sigma'', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau_2 \sqcap K \rrbracket$.

2539 Since $(j', \Psi'') \sqsupseteq (k, \Psi)$ and $\Sigma'' \supseteq \Sigma$, we can apply our premise to finish the case.

vi) $\tau = *$: note by downward closure, $\Sigma' : (k - j - 1, \Psi')$.

Then we want to show $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\text{Int}]$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[** \times **]$ or $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T[** \rightarrow **]$.

We know $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\text{Int}]$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[** \times **]$ or $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T[** \rightarrow **]$.

The case follows by the IH (smaller by index). □

LEMMA 5.52 (EXTENSIONS PRESERVE VALUE LOG TYPING). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ and $\ell \notin \text{dom}(\Sigma')$ and $\Sigma[\ell \mapsto (v, _)] : (k, \Psi[\ell \mapsto \bar{\tau}])$ then $\Sigma'[\ell \mapsto (v, _)] : (k - j, \Psi'[\ell \mapsto \bar{\tau}])$.*

PROOF. Note that all of the conditions in $\Sigma'[\ell \mapsto (v, _)] : (k - j, \Psi'[\ell \mapsto \bar{\tau}])$ besides those concerning the history relation are immediate from the hypotheses.

Let $\Sigma'' = \Sigma'[\ell \mapsto (v, _)]$ and let $\Psi'' = \Psi'[\ell \mapsto \bar{\tau}]$.

We want to show $\forall j' < k - j$, and $\forall \ell \in \text{dom}(\Sigma'')$, $(j', \Psi'', \Sigma'', \ell) \in \mathcal{V}\mathcal{H}^T[\Psi''(\ell)]$.

Note by downward closure, $\Sigma'' : (j', \Psi'')$. If $\ell \in \text{dom}(\Sigma')$, then we can apply Lemma 5.51 with the fact that $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \supseteq \Sigma'$.

If $\ell \notin \text{dom}(\Sigma')$, then $\ell \in \bar{\ell}$.

Then we can apply Lemma 5.51 with the fact that $(j', \Psi'') \sqsupseteq (k, \Psi[\ell \mapsto \bar{\tau}])$ and $\Sigma'' \supseteq \Sigma[\ell \mapsto (v, _)]$ to get $(j', \Psi'', \Sigma'', \ell) \in \mathcal{V}\mathcal{H}^T[\Psi''(\ell)]$, which is what we wanted to show. □

LEMMA 5.53 (LATER THAN PRESERVED BY LOWER STEPS). *If $(j, \Psi') \sqsupseteq (k, \Psi)$ and $j' \leq j$ then $(j - j', \Psi') \sqsupseteq (k - j', \Psi)$.*

PROOF. Unfolding the world extension definition, we need to show $j - j' \leq k - j'$ and $\forall \ell \in \text{dom}(\Psi)$, $\Psi'(\ell) = \Psi(\ell)$. For the first condition, since $j \leq k$ and $j' \leq j$, $j - j' \leq k - j'$.

For the second condition, we can unfold the hypothesis to get the statement we need. □

LEMMA 5.54 (RE-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T[\bar{\tau}]$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{E}\mathcal{H}^T[\bar{\tau}]$.*

PROOF. Unfolding the relation in our hypothesis, we get that there is some (Σ'', e') , j' such that $(\Sigma, e) \xrightarrow{j'}_T (\Sigma'', e')$. If $e' = \text{Err}^\bullet$ then we're done.

Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{V}\mathcal{H}^T[\bar{\tau}]$.

By Lemma 5.48, $\Sigma'' = \Sigma[\ell' \mapsto (v, _)]$.

By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi[\ell' \mapsto \bar{\tau}]$.

We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma[\ell' \mapsto (v', _)]$.

And from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi[\ell' \mapsto \bar{\tau}']$.

By alpha renaming, we can assume that $\ell' \notin \text{dom}(\Sigma'')$.

Then by Lemma 5.49, we get that $(\Sigma', e) \xrightarrow{j'}_T (\Sigma''[\ell' \mapsto (v', _)], e')$.

Now, unfolding the expression relation in what we want to show, we have two obligations:

- a) $\Sigma''[\ell' \mapsto (v', _)] : (k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'])$.
- b) $(k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'], \Sigma''[\ell' \mapsto (v', _)], e') \in \mathcal{V}\mathcal{H}^T[\bar{\tau}]$.

For a) we can apply Lemma 5.52. We have a number of obligations:

- i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
- ii) $\Sigma'' \supseteq \Sigma$: immediate.
- iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 5.53.
- iv) $\Sigma'' : (k - j - j', \Psi'')$: immediate by downward closure.
- v) $\ell' \notin \text{dom}(\Sigma'')$: assumed above by alpha renaming.
- vi) $\Sigma[\ell' \mapsto (v', _)] : (k - j, \Psi[\ell' \mapsto \bar{\tau}'])$: this is exactly $\Sigma' : (k - j, \Psi')$.

For b), we can apply Lemma 5.51 with the fact proven in a). □

LEMMA 5.55 (E-V-MONOTONICITY). *If $\Sigma : (k, \Psi)$ and $0 \leq j \leq k$ and $\Sigma' \supseteq \Sigma$ and $(k - j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' : (k - j, \Psi')$ then*

- (1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T[\tau]$ then $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^T[\tau]$*
- (2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T[\tau]$ then $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^T[\tau]$*

PROOF. Proceed by simultaneous induction on k and τ :

- $k = 0$: 1) follows immediately from 2).

Proceeds similarly to the other case, but function and dynamic cases are vacuously true.

- $k > 0$:

- 1) Unfolding the expression relation in our hypothesis, we get that there is some $(\Sigma'', e'), j'$ such that $(\Sigma, e) \longrightarrow_T^{j'} (\Sigma'', e')$.

If $e' = \text{Err}^\bullet$ then we're done.

Otherwise, there is some $(k - j', \Psi'') \sqsupseteq (k, \Psi)$ such that $\Sigma'' : (k - j', \Psi'')$ and $(k - j', \Psi'', \Sigma'', e') \in \mathcal{V}^T[\tau]$.

By Lemma 5.48, $\Sigma'' = \Sigma[\ell' \mapsto (v', _)]$.

By the fact that $\Sigma'' : (k - j', \Psi'')$ this also means $\Psi'' = \Psi[\ell' \mapsto \bar{\tau}']$.

We also know from $\Sigma' \supseteq \Sigma$ that $\Sigma' = \Sigma[\ell' \mapsto (v', _)]$, and from $\Sigma' : (k - j, \Psi')$ that $\Psi' = \Psi[\ell' \mapsto \bar{\tau}']$.

By alpha renaming, we can assume that $\ell' \notin \text{dom}(\Sigma'')$.

Then by Lemma 5.49, we get that $(\Sigma'', e) \longrightarrow_T^{j'} (\Sigma''[\ell' \mapsto (v', _)], e')$.

Now, unfolding the expression relation in what we want to show, we have two obligations:

- a) $\Sigma''[\ell' \mapsto (v', _)] : (k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'])$.
- b) $(k - j - j', \Psi''[\ell' \mapsto \bar{\tau}'], \Sigma''[\ell' \mapsto (v', _)], e') \in \mathcal{V}^T[\tau]$.

For a) we can apply Lemma 5.52. We have a number of obligations:

- i) $\Sigma : (k - j, \Psi)$: immediate by downward closure.
- ii) $\Sigma'' \supseteq \Sigma$: immediate.
- iii) $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$: by Lemma 5.53.
- iv) $\Sigma'' : (k - j - j', \Psi'')$: immediate by downward closure.
- v) $\ell' \notin \text{dom}(\Sigma'')$: assumed above by alpha renaming.
- vi) $\Sigma[\ell' \mapsto (v', _)] : (k - j, \Psi[\ell' \mapsto \bar{\tau}'])$: this is exactly $\Sigma' : (k - j, \Psi')$.

For b), we can apply the IH 2) (not necessarily smaller by type or index) with the fact proven in a).

- 2653 2) We want to show that $(k - j, \Psi', \Sigma', \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$.
- 2654 We case split on τ :
- 2655 i) $\tau = \text{Nat}$: then $\Sigma(\ell) = (n, _)$ where $n \in \mathbb{T}$, so the case is immediate.
- 2656
- 2657 ii) $\tau = \text{tint}$: same as above.
- 2658
- 2659
- 2660 iii) $\tau = \text{Bool}$: same as above.
- 2661
- 2662
- 2663 iv) $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.
- 2664 Unfolding our hypothesis gives us $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$.
- 2665 Applying IH 2) (smaller by type) to both gives us $(k - j, \Psi', \Sigma', \ell_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k - j, \Psi', \Sigma', \ell_2) \in$
- 2666 $\mathcal{V}^T \llbracket \tau_2 \rrbracket$, which is sufficient to complete the case.
- 2667
- 2668 v) $\tau = * \rightarrow \tau_2$: Let $\Sigma'' \supseteq \Sigma'$ and $(j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (j', \Psi'')$.
- 2669 Let $\ell_v \in \text{dom}(\Sigma'')$ such that $(j', \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^T \llbracket * \rrbracket$.
- 2670 Let K .
- 2671 We want to show $(j', \Psi'', \Sigma'', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket K \sqcap \tau_2 \rrbracket$.
- 2672 Since \supseteq and \sqsupseteq are both transitive, we have $\Sigma'' \supseteq \Sigma$, and $(j', \Psi'') \sqsupseteq (k, \Psi)$.
- 2673 Therefore we can apply the hypothesis to complete the case.
- 2674
- 2675 vi) $\tau = *$: we want to show $(k - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T \llbracket \text{Int} \rrbracket$ or $\mathcal{V}^T \llbracket \text{Bool} \rrbracket$ or $\mathcal{V}^T \llbracket * \times * \rrbracket$ or $\mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.
- 2676 This follows from IH 2) (smaller by index).
- 2677
- 2678
- 2679 □

2680 LEMMA 5.56 (BOT RELATION IF AND ONLY IF ERROR). $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \perp \rrbracket$ and $(\Sigma, e) \xrightarrow{j}_T (\Sigma', e')$ where (Σ', e') is

2681 irreducible and $j \leq k$, iff $e' = \text{Err}^\bullet$.

2682

2683 PROOF. $\bullet \Rightarrow$: Unfolding our hypothesis about e in the expression relation, we get that either:

- 2684 - $e' = \text{Err}^\bullet$ or
- 2685 - $\exists (k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \perp \rrbracket$
- 2686

2687 Assume for sake of contradiction the second case holds.

2688 $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \perp \rrbracket$ implies $(k - j, \Psi', \Sigma', \Sigma'(e')) \in \mathcal{V}^T \llbracket \perp \rrbracket$, which is a contradiction.

2689 Therefore, $e' = \text{Err}^\bullet$.

- 2690 $\bullet \Leftarrow$: immediate.
- 2691
- 2692 □

2693 LEMMA 5.57 (TAGMATCH MAKES VALUES IN RELATION AT MEET). If $K \propto \text{pointsto}(\Sigma, \ell)$ and $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$ then

2694 $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket K \sqcap \tau \rrbracket$

2695

2696 PROOF. There are three cases to consider:

- 2697 (1) $K \sqcap \tau = \perp$: a contradiction.
- 2698
- 2699
- 2700 (2) $K \sqcap \tau = \tau$: immediate by Lemma 5.55.
- 2701
- 2702 (3) $K \sqcap \tau = K$ and $\tau = *$: immediate by unfolding the value relation in our hypothesis, and noting that whichever
- 2703 type of $\text{Int}, * \times *$ or $* \rightarrow *$ we satisfy must be K .
- 2704

2705 □

2706

2707 LEMMA 5.58 (CHECK MAKES TERMS IN RELATION AT MEET). *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \text{assert } K e) \in$*
 2708 *$\mathcal{E}^T \llbracket \tau \sqcap K \rrbracket$.*

2709
 2710 PROOF. Unfolding the expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \xrightarrow{T}^j (\Sigma', e')$
 2711 and (Σ', e') is irreducible.

2712 If $e' = \text{Err}^\bullet$ then we're done.

2713 Otherwise $\exists (k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau \rrbracket$.

2714 It suffices to show $(k - j, \Psi', \Sigma', \text{assert } K e') \in \mathcal{E}^T \llbracket \tau \sqcap K \rrbracket$.

2715 By the OS, if $\neg K \propto \text{pointsto}(\Sigma', e')$ then $(\Sigma', \text{assert } K e') \xrightarrow{T} (\Sigma', \text{Err}^\bullet)$ and we're done.

2716 Otherwise, $(\Sigma', \text{assert } K e') \xrightarrow{T} (\Sigma', e')$ and $K \propto \text{pointsto}(\Sigma', e')$.

2717 By Lemma 5.57, we therefore get $(k - j - 1, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau \sqcap K \rrbracket$, which is sufficient to complete the proof. □

2718

2719 LEMMA 5.59 (TAGMATCH MAKES VALUES IN HISTORY RELATION AT MEET). *If $K \propto \text{pointsto}(\Sigma, \ell)$ and $(k, \Psi, \Sigma, \ell) \in$*
 2720 *$\mathcal{V}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$ then $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket K \sqcap \tau, \bar{\tau} \rrbracket$*

2721

2722 PROOF. There are three cases to consider:

2723 (1) $K \sqcap \tau = \perp$: a contradiction because $K \propto \Sigma(\ell)$ and $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

2724

2725 (2) $K \sqcap \tau = \tau$: immediate by Lemma 5.51.

2726 (3) $K \sqcap \tau = K$ and $\tau = *$: immediate by unfolding the erroring value relation in our hypothesis, and noting that
 2727 whichever type of $\text{Int}, * \times * \text{ or } * \rightarrow *$ we satisfy must be K .

2728 □

2729

2730 LEMMA 5.60 (CHECK MAKES TERMS IN HISTORY RELATION AT MEET). *If $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, \text{assert } K e) \in$*
 2731 *$\mathcal{E}\mathcal{H}^T \llbracket \tau \sqcap K, \bar{\tau} \rrbracket$.*

2732

2733 PROOF. Unfolding the erroring expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \xrightarrow{T}^j$
 2734 (Σ', e') and (Σ', e') is irreducible.

2735 If $e' = \text{Err}^\bullet$ then we're done.

2736 Otherwise $\exists (k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}\mathcal{H}^V \llbracket T \rrbracket \tau, \bar{\tau}$.

2737 It suffices to show $(k - j, \Psi', \Sigma', \text{assert } K e') \in \mathcal{E}\mathcal{H}^T \llbracket \tau \sqcap K, \bar{\tau} \rrbracket$.

2738 By the OS, if $\neg K \propto \text{pointsto}(\Sigma', e')$ then $(\Sigma', \text{assert } K e') \xrightarrow{T} (\Sigma', \text{Err}^\bullet)$ and we're done.

2739 Otherwise, $(\Sigma', \text{assert } K e') \xrightarrow{T} (\Sigma', e')$ and $K \propto \text{pointsto}(\Sigma', e')$.

2740 By Lemma 5.59, we therefore get $(k - j - 1, \Psi', \Sigma', e') \in \mathcal{V}\mathcal{H}^V \llbracket T \rrbracket \tau \sqcap K, \bar{\tau}$, which is sufficient to complete the proof. □

2741

2742 LEMMA 5.61 (LATTICE ORDERING PRESERVES RELATION). *If $\tau \leq \tau'$ then*

2743 (1) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau' \rrbracket$*

2744 (2) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.*

2745

2746 PROOF. (1) Unfolding the expression relation in our hypothesis, we have that $\exists e', \Sigma', j$ such that $(\Sigma, e) \xrightarrow{T}^j$
 2747 (Σ', e') and (Σ', e') is irreducible.

2748 If $e' = \text{Err}^\bullet$ then we're done.

2749

2757 Otherwise $\exists(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau \rrbracket$.
 2758 It suffices to show $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau' \rrbracket$, which follows by IH 2).
 2759 (2) Proceed by induction over the lattice ordering:
 2760 (a) $\tau \leq \tau'$: follows from Lemma 5.50.
 2761 (b) $\tau = \tau_1 \times \tau_2$, $\tau' = \tau'_1 \times \tau'_2$, $\tau_1 \leq \tau'_1$, and $\tau_2 \leq \tau'_2$:
 2762 Then unfolding the location relation in our hypothesis, we have that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.
 2763 We also have that $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$.
 2764 Unfolding the relation in what we want to show, we want to show $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in$
 2765 $\mathcal{V}^T \llbracket \tau'_2 \rrbracket$, which follows by IH 2).
 2766 (c) $\tau = * \rightarrow \tau_o$, $\tau' = * \rightarrow \tau'_o$, and $\tau_o \leq \tau'_o$:
 2767 We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket * \rightarrow \tau'_o \rrbracket$.
 2768 Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
 2769 Let $\ell_o \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_o) \in \mathcal{V}^T \llbracket * \rrbracket$.
 2770 Let K .
 2771 We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau'_o \sqcap K \rrbracket$.
 2772 From our hypothesis, we get that $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau_o \sqcap K \rrbracket$.
 2773 The proof follows from IH 1).
 2774 (d) $\tau' = *$: Proceed by case analysis on τ :
 2775 (i) $\tau = \text{Nat}$: Immediate.
 2776 (ii) $\tau = \text{Int}$: Immediate.
 2777 (iii) $\tau = \text{Bool}$: Immediate.
 2778 (iv) $\tau = \tau_1 \times \tau_2$: Then unfolding the location relation in our hypothesis, we have that $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.
 2779 We also have that $(k, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$.
 2780 Unfolding the relation in what we want to show, we want to show $(k - 1, \Psi, \Sigma, \ell_1) \in \mathcal{V}^T \llbracket * \rrbracket$ and
 2781 $(k - 1, \Psi, \Sigma, \ell_2) \in \mathcal{V}^T \llbracket * \rrbracket$, which follows by IH 2) and Lemma 5.55.
 2782 (v) $\tau = * \rightarrow \tau'$: We want to show $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.
 2783 Let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \supseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.
 2784 Let $\ell_o \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_o) \in \mathcal{V}^T \llbracket * \rrbracket$.
 2785 Let K .
 2786 We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket K \rrbracket$.
 2787 From our hypothesis, we get that $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket \tau' \sqcap K \rrbracket$.
 2788 By the IH 1), we get that $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_o) \in \mathcal{E}^T \llbracket K \rrbracket$ which is what we wanted to show.

□

2799 **LEMMA 5.62 (PAIRS OF SEMANTICALLY WELL TYPED TERMS ARE SEMANTICALLY WELL TYPED).** *If $(k, \Psi, \Sigma, e_1) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$*
 2800 *and $(k, \Psi, \Sigma, e_2) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$ then $(k, \Psi, \Sigma, \langle e_1, e_2 \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.*
 2801

2802 **PROOF.** Unfolding the expression relation in our hypothesis about e_1 , we get that there are $(\Sigma, e'_1), j$ such that
 2803 $(\Sigma, e_1) \xrightarrow{j}_T (\Sigma, e'_1)$ and (Σ', e'_1) is irreducible.
 2804

2805 If $e'_1 = \text{Err}^\bullet$, then were done because the entire application steps to an error.

2806 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$.

2807 This means $e'_1 = \ell_1$ for some $\ell_1 \in \text{dom}(\Sigma')$.
 2808

2809
2810 With this and by the OS, we get $(\Sigma, \langle e_1, e_2 \rangle) \rightarrow_T^j (\Sigma', \langle loc_1, e_2 \rangle)$.
2811

2812 We can apply Lemma 5.55 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.
2813

2814 Unfolding the expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \rightarrow_T^{j'} (\Sigma', e_2')$ and (Σ', e_2') is
2815 irreducible.

2816 If $e_2' = \text{Err}^\bullet$, then were done because the entire application steps to an error.
2817

2818 Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$,
2819 which means $e_2' = \ell_2$ for some $\ell_2 \in \text{dom}(\Sigma'')$.
2820

2821 Putting everything together we get $(\Sigma, \langle e_1, e_2 \rangle) \rightarrow_T^{j'} (\Sigma'', \langle \ell_1, \ell_2 \rangle)$, with $\Sigma'' : (k - j - j', \Psi'')$.
2822

2823 Note by OS, $(\Sigma'', \langle \ell_1, \ell_2 \rangle) \rightarrow_T (\Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle])$ where $\ell' \notin \text{dom}(\Sigma'')$.
2824

2825 We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] : (k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.
2826

2827 Note the only interesting part of this statement is that $\forall k' < k - j - j' - 1$. $(k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto$
2828 $(\langle \ell_1, \ell_2 \rangle, _)] , \ell') \in \mathcal{V}^T \llbracket \Psi''(\ell_1) \times \Psi''(\ell_2) \rrbracket$.

2829 This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in$
2830 $\mathcal{V}^T \llbracket \Psi''(\ell_1) \rrbracket$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^T \llbracket \Psi''(\ell_2) \rrbracket$.
2831

2832 We know that $(k - j, \Psi', \Sigma', \ell_1') \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$, and Lemma 5.55 with down-
2833 ward closure and the store typing judgement above.
2834

2835 From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] , \ell_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and
2836 $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle] , \ell_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$.
2837

2838 This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] , \langle \ell_1, \ell_2 \rangle) \in \mathcal{V}^T \llbracket \tau_1 \times \tau_2 \rrbracket$,
2839 which is what we wanted to prove. \square
2840

2841 LEMMA 5.63 (PAIRS OF RELATED TERMS ARE RELATED). *If $(k, \Psi, \Sigma, e_1) \in \mathcal{E}^T \llbracket fst(\bar{\tau}) \rrbracket$ and $(k, \Psi, \Sigma, e_2) \in \mathcal{E}^T \llbracket snd(\bar{\tau}) \rrbracket$*
2842 *then $(k, \Psi, \Sigma, \langle e_1, e_2 \rangle) \in \mathcal{E}^T \llbracket \bar{\tau} \rrbracket$.*
2843

2844 PROOF. Unfolding the erroring expression relation in our hypothesis about e_1 , we get that there are $(\Sigma, e_1'), j$ such
2845 that $(\Sigma, e_1) \rightarrow_T^j (\Sigma, e_1')$ and (Σ, e_1') is irreducible.
2846

2847 If $e_1' = \text{Err}^\bullet$, then were done because the entire application steps to an error.
2848

2849 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e_1') \in \mathcal{V}^T \llbracket fst(\bar{\tau}) \rrbracket$.
2850

2851 This means $e_1' = \ell_1$ for some $\ell_1 \in \text{dom}(\Sigma')$.
2852

2853 With this and by the OS, we get $(\Sigma, \langle e_1, e_2 \rangle) \rightarrow_T^j (\Sigma', \langle loc_1, e_2 \rangle)$.
2854

2855 We can apply Lemma 5.54 to our hypothesis about e_2 to get $(k - j, \Psi', \Sigma', e_2) \in \mathcal{E}^T \llbracket snd(\bar{\tau}) \rrbracket$.
2856

2857 Unfolding the erroring expression relation, we get that there are $(\Sigma', e_2'), j'$ such that $(\Sigma', e_2) \rightarrow_T^{j'} (\Sigma', e_2')$ and (Σ', e_2')
2858 is irreducible.

2859 If $e_2' = \text{Err}^\bullet$, then were done because the entire application steps to an error.
2860

2861 Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e_2') \in$
2862 $\mathcal{V}^T \llbracket snd(\bar{\tau}) \rrbracket$.

2861 $\mathcal{VH}^T \llbracket \text{snd}(\bar{\tau}) \rrbracket$, which means $e'_2 = \ell_2$ for some $\ell_2 \in \text{dom}(\Sigma'')$.

2862

2863

2864

Putting everything together we get $(\Sigma, \langle e_1, e_2 \rangle) \xrightarrow{j'}_T (\Sigma'', \langle \ell_1, \ell_2 \rangle)$, with $\Sigma'' : (k - j - j', \Psi'')$.

2865

Note by OS, $(\Sigma'', \langle \ell_1, \ell_2 \rangle) \xrightarrow{T} (\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)])$ where $\ell' \notin \text{dom}(\Sigma'')$.

2866

2867

We firstly need $\Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)] : (k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)])$.

2868

2869

Note the only interesting part of this statement is that $\forall k' < k - j - j' - 1. (k', \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)], \ell') \in \mathcal{VH}^T \llbracket \Psi''(\ell_1) \times \Psi''(\ell_2) \rrbracket$.

2870

2871

This is immediate from the fact that $\Sigma'' : (k', \Psi'')$ from downward closure, and therefore that $(k', \Psi'', \Sigma'', \ell_1) \in \mathcal{VH}^T \llbracket \Psi''(\ell_1) \rrbracket$ and $(k', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^T \llbracket \Psi''(\ell_2) \rrbracket$.

2872

2873

2874

We know that $(k - j, \Psi', \Sigma', \ell'_1) \in \mathcal{VH}^T \llbracket \text{fst}(\bar{\tau}) \rrbracket$ and $(k - j - j', \Psi'', \Sigma'', \ell_2) \in \mathcal{VH}^T \llbracket \text{snd}(\bar{\tau}) \rrbracket$, and Lemma 5.51 with downward closure and the store typing judgement above.

2875

2876

From these facts we get that $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)], \ell_1) \in \mathcal{VH}^T \llbracket \text{fst}(\bar{\tau}) \rrbracket$ and $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto \langle \ell_1, \ell_2 \rangle], \ell_2) \in \mathcal{VH}^T \llbracket \text{snd}(\bar{\tau}) \rrbracket$.

2877

2878

This is sufficient to show $(k - j - j' - 1, \Psi''[\ell' \mapsto \Psi''(\ell_1) \times \Psi''(\ell_2)], \Sigma''[\ell' \mapsto (\langle \ell_1, \ell_2 \rangle, _)], \langle \ell_1, \ell_2 \rangle) \in \mathcal{VH}^T \llbracket \bar{\tau} \rrbracket$, which is what we wanted to prove. \square

2881

2882

2883

2884

2885

LEMMA 5.64 (APPLICATIONS OF SEMANTICALLY WELL TYPED TERMS ARE SEMANTICALLY WELL TYPED). *If $(k, \Psi, \Sigma, e_f) \in \mathcal{E}^T \llbracket * \rightarrow \tau \rrbracket$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket * \rrbracket$ then $\forall K, (k, \Psi, \Sigma, \text{app}\{K\} e_f e) \in \mathcal{E}^T \llbracket \tau \sqcap K \rrbracket$.*

2886

2887

PROOF. Unfolding the expression relation in our hypothesis about e_f , we get that there are $(\Sigma', e'_f), j$ such that $(\Sigma, e_f) \xrightarrow{j}_T (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.

2888

2889

If $e'_f = \text{Err}^\bullet$, then we're done because the entire application steps to an error.

2890

2891

Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_f) \in \mathcal{V}^T \llbracket * \rightarrow \tau \rrbracket$.

2892

2893

This means $e'_f = \ell_f$ for some $\ell_f \in \text{dom}(\Sigma')$.

2894

2895

Using this, we know from the OS that $(\Sigma, \text{app}\{K\} e_f e) \xrightarrow{j}_T (\Sigma', \text{app}\{K\} \ell_f e)$.

2896

2897

We can apply Lemma 5.55 with $\Sigma' : (k - j, \Psi')$ to our hypothesis about e to get $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^T \llbracket * \rrbracket$.

2898

2899

Unfolding the expression relation, we get that there are (Σ'', e') , j' such that $(\Sigma', e) \xrightarrow{j'}_T (\Sigma'', e')$ where (Σ'', e') is irreducible.

2900

2901

If $e' = \text{Err}^\bullet$ than we're done, because the whole application errors.

2902

2903

Otherwise, there exists $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e') \in \mathcal{V}^T \llbracket * \rrbracket$.

2904

2905

This means $e' = \ell$ for some $\ell \in \text{dom}(\Sigma'')$.

2906

2907

Putting what we have together, by the OS, $(\Sigma, \text{app}\{K\} e_f e) \xrightarrow{j+j'}_T (\Sigma'', (\text{app}\{K\} \ell_f \ell))$.

2908

2909

We have $(k - j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^T \llbracket * \rightarrow \tau \rrbracket$ and $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \sqsupseteq \Sigma'$ and $\Sigma'' : (k - j - j', \Psi'')$.

2910

We can combine these to get $(k - j - j', \Psi'', \Sigma'', \text{app}\{K\} \ell_f \ell) \in \mathcal{E}^T \llbracket \tau \sqcap K \rrbracket$.

2911

2912

This is sufficient to complete the proof. \square

2913 COROLLARY 5.65. *If $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^T \llbracket * \rrbracket$ and $\Sigma(\ell) = w$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket * \rrbracket$ then $(k - 1, \Psi, \Sigma, \text{app}\{*\} w e) \in$*
 2914 *$\mathcal{E}^T \llbracket * \rrbracket$.*
 2915

2916 LEMMA 5.66 (APPLICATIONS OF RELATED TERMS ARE RELATED). *If $(k, \Psi, \Sigma, e_f) \in \mathcal{E}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$ and $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket * \rrbracket$*
 2917 *then $\forall K, (k, \Psi, \Sigma, \text{app}\{K\} e_f e) \in \mathcal{E}\mathcal{H}^T \llbracket \text{cod}(\tau) \sqcap K, \text{cod}(\bar{\tau}) \rrbracket$.*
 2918

2919 PROOF. Unfolding the erroring expression relation in our hypothesis about e_f , we get that there are $(\Sigma', e'_f), j$ such
 2920 that $(\Sigma, e_f) \xrightarrow{j}_T (\Sigma', e'_f)$ and (Σ', e'_f) is irreducible.
 2921

2922 If $e'_f = \text{Err}^\bullet$, then we're done because the entire application steps to an error.

2923 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_f) \in \mathcal{V}\mathcal{H}^T \llbracket \tau, \bar{\tau} \rrbracket$.

2924 This means $e'_f = \ell_f$ for some $\ell_f \in \text{dom}(\Sigma')$.
 2925

2926 Using this, we know from the OS that $(\Sigma, \text{app}\{K\} e_f e) \xrightarrow{j}_T (\Sigma', \text{app}\{K\} \ell_f e)$.
 2927

2928 We can apply Lemma 5.55 with $\Sigma' : (k - j, \Psi')$ to our hypothesis about e to get $(k - j, \Psi', \Sigma', e) \in \mathcal{E}^T \llbracket * \rrbracket$.
 2929

2930 Unfolding the expression relation, we get that there are (Σ'', e') , j' such that $(\Sigma', e) \xrightarrow{j'}_T (\Sigma'', e')$ where (Σ'', e') is
 2931 irreducible.
 2932

2933 If $e' = \text{Err}^\bullet$ then we're done, because the whole application errors.

2934 Otherwise, there exists $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e') \in \mathcal{V}^T \llbracket * \rrbracket$.
 2935

2936 This means $e' = \ell$ for some $\ell \in \text{dom}(\Sigma'')$.
 2937

2938 Putting what we have together, by the OS, $(\Sigma, \text{app}\{K\} e_f e) \xrightarrow{j+j'}_T (\Sigma'', (\text{app}\{K\} \ell_f \ell))$.
 2939

2940 We have $(k - j, \Psi', \Sigma', \ell_f) \in \mathcal{V}^T \llbracket * \rightarrow \tau \rrbracket$ and $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi')$ and $\Sigma'' \sqsupseteq \Sigma'$ and $\Sigma'' : (k - j - j', \Psi'')$.
 2941

2942 We can combine these to get $(k - j - j', \Psi'', \Sigma'', \text{app}\{K\} \ell_f \ell) \in \mathcal{E}\mathcal{H}^T \llbracket \text{cod}(\tau) \sqcap K, \text{cod}(\bar{\tau}) \rrbracket$.
 2943

2944 This is sufficient to complete the proof. □

2945 COROLLARY 5.67. *If $(k, \Psi, \Sigma, e_f) \in \mathcal{E}\mathcal{H}^T \llbracket *, \bar{\tau} \rrbracket$ and $(k - 1, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket * \rrbracket$ then $(k - 1, \Psi, \Sigma, \text{app}\{\tau_0\} e_f e) \in$*
 2946 *$\mathcal{E}\mathcal{H}^T \llbracket *, \text{cod}(\bar{\tau}) \rrbracket$.*
 2947

2948 LEMMA 5.68 (DYNAMIC CHECKS ARE NOOPS). (1) *If $(k + 1, \Psi, \Sigma, \text{assert } * e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$.*
 2949

(2) *If $(k + 1, \Psi, \Sigma, \text{assert } * e) \in \mathcal{E}\mathcal{H}^T \llbracket \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \bar{\tau} \rrbracket$.*
 2950

2951 PROOF. (1) assume there is Σ', e', j such that $(\Sigma, e) \xrightarrow{j}_T (\Sigma', e')$ where (Σ', e') is irreducible.

2952 By the OS, we get that $(\Sigma, \text{assert } * e) \xrightarrow{j}_T (\Sigma', \text{assert } * e')$.

2953 Then by OS, we have $(\Sigma', \text{assert } * e') \xrightarrow{j}_T (\Sigma', e')$.

2954 Therefore, we can apply our hypothesis to complete the proof.

2955 (2) Same as previous case, just using the history relation.
 2956 □

2957 LEMMA 5.69 (MONITOR COMPATIBILITY). *If $\Sigma : (k, \Psi)$, then*

2958 (1) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$ and $\Sigma(\ell') = (\ell, \text{some}(K', K))$, then $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^T \llbracket K' \sqcap K \sqcap \tau \rrbracket$*
 2959

2960 (2) *If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \sqcap K \sqcap K' \rrbracket$ then $(k, \Psi, \Sigma, \text{mon } \{K' \leftarrow K\} e) \in \mathcal{E}^T \llbracket \tau \sqcap K \sqcap K' \rrbracket$.*
 2961

2962 (3) *If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \Psi(\ell) \rrbracket$ and $\Sigma' = \Sigma[\ell' \mapsto (\ell, \text{some}(K', K))]$ and $\Psi' = [\ell' \mapsto K', K, \Psi(\ell)]\Psi$ and $\ell' \notin$
 2963 *$\text{dom}(\Sigma)$ and $\vdash \Sigma'$ then $(k, \Psi', \Sigma', \ell') \in \mathcal{V}\mathcal{H}^T \llbracket K', K, \Psi(\ell) \rrbracket$*
 2964*

2965 (4) If $(k, \Psi, \Sigma, e) \in \mathcal{E}\mathcal{H}^T \llbracket \bar{\tau} \rrbracket$ then $(k, \Psi, \Sigma, \text{mon} \{ * \Leftarrow * \} e) \in \mathcal{E}\mathcal{H}^T \llbracket *, *, \bar{\tau} \rrbracket$

2966

2967

PROOF. Proceed by simultaneous induction on k and τ .

2968

- $k = 0$: 2) and 4) follow from 1) and 3) respectively.

2969

The proofs follow similarly to the other case, but any function or dynamic cases are vacuously true.

2970

- $k > 0$:

2971

- 1) Unfolding the relation in the statement we want to prove, note from our hypothesis about Σ , we get that

2972

$\vdash \Sigma$.

2973

Proceed by case analysis on $\tau \sqcap K \sqcap K'$:

2974

- i) $\tau = \tau \sqcap K \sqcap K'$: Immediate.

2975

- ii) $\tau \sqcap K \sqcap K' = \perp$: then either K or K' is \perp , which is a contradiction since they both tagmatch $\text{pointsto}(\Sigma, \ell)$.

2976

- iii) $\tau \sqcap K \sqcap K' \leq \tau$: then $\tau = \text{Int}$ and K or $K' = \text{Nat}$.

2977

Immediate because by $\vdash \Sigma$, $\text{Nat} \propto \text{pointsto}(\Sigma, \ell)$.

2978

- iv) $\tau \sqcap K \sqcap K' \neq \tau$: then it must be the case that $\tau = *$ and K or $K' = * \rightarrow *$.

2979

Note K or K' cannot be $* \times *$, by $\vdash \Sigma$.

2980

Unfolding the relation in our hypothesis, we have that $(k-1, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.

2981

We want to show that $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.

2982

Unfolding the relation, let $(j, \Psi') \sqsupseteq (k, \Psi)$ and $\Sigma' \sqsupseteq \Sigma$ such that $\Sigma' : (j, \Psi')$.

2983

Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket * \rrbracket$.

2984

Let K .

2985

We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell' \ell_v) \in \mathcal{E}^T \llbracket K \rrbracket$.

2986

By the OS, $(\Sigma', \text{app}\{K\} \ell' \ell_v) \xrightarrow{2}_T (\Sigma', \text{assert } K (\text{mon} \{ * \Leftarrow * \} (\ell (\text{mon} \{ * \Leftarrow * \} \ell_v))))$.

2987

By IH 2), we have $(j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow * \} \ell_v) \in \mathcal{E}^T \llbracket * \rrbracket$.

2988

By Lemma 5.64, we have that $(j, \Psi', \Sigma', \text{app}\{K\} \ell (\text{mon} \{ * \Leftarrow * \} \ell_v)) \in \mathcal{E}^T \llbracket K \rrbracket$.

2989

Then by IH 2), we have $(j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow * \} (\text{app}\{K\} \ell (\text{mon} \{ * \Leftarrow * \} \ell_v))) \in \mathcal{E}^T \llbracket K \rrbracket$.

2990

Note that $(j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow * \} (\text{app}\{K\} \ell (\text{mon} \{ * \Leftarrow * \} \ell_v))) \in \mathcal{E}^T \llbracket K \rrbracket$ iff $(j, \Psi', \Sigma', \text{assert } K (\text{mon} \{ * \Leftarrow * \} (\ell (\text{mon} \{ * \Leftarrow * \} \ell_v)))) \in \mathcal{E}^T \llbracket K \rrbracket$.

2991

Therefore, this is sufficient to complete the case.

2992

- 2) Unfolding the expression relation in our hypothesis, we have that there are $(e', \Sigma'), j$ such that $(e, \Sigma) \xrightarrow{j}_T (e', \Sigma')$ with (e', Σ') irreducible.

2993

If $e' = \text{Err}^\bullet$ then we're done, because the monitor will step to an error as well.

2994

Otherwise, there is $(k-j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k-j, \Psi')$ and $(k-j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau \sqcap K \sqcap K' \rrbracket$.

2995

This means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.

2996

2997

2998

2999

3000

3001

3002

3003

3004

3005

3006

3007

3008

3009

3010

3011

We want to show $(k-j, \Psi', \Sigma', \text{mon} \{ K \Leftarrow \ell \}) \in \mathcal{E}^T \llbracket \tau \sqcap K \sqcap K' \rrbracket$.

3012

We destruct on whether $\Sigma'(\ell)$ is a pair.

3013

If $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$, then by the OS, $(\Sigma', \text{mon} \{ K \Leftarrow \ell \}) \xrightarrow{T} (\Sigma', \langle \text{mon} \{ * \Leftarrow * \} \ell_1, \text{mon} \{ * \Leftarrow * \} \ell_2 \rangle)$.

3014

Then by Lemma 5.62, it suffices to show $(k-j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow \ell_1 \}) \in \mathcal{E}^T \llbracket \text{fst}(\tau) \rrbracket$ and $(k-j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow \ell_2 \}) \in \mathcal{E}^T \llbracket \text{snd}(\tau) \rrbracket$

3015

3016

These both follow from IH 2) (smaller by index).

Otherwise, by the OS, $(\Sigma', \text{mon } \{K \leftarrow \ell\} \rightarrow_T (\Sigma'[\ell' \mapsto (\ell, \text{some}(K', K))], \ell'))$.

Then by IH 3), we get $\Sigma'[\ell' \mapsto (\ell, \text{some}(K', K))] : (k - j - 1, \Psi'[\ell' \mapsto K', K, \Psi'(\ell)])$.

And by IH 1), we get $(k - j - 1, \Psi'[\ell' \mapsto K', K, \Psi'(\ell)], \Sigma'[\ell' \mapsto (\ell, \text{some}(K', K))], \ell') \in \mathcal{V}^T \llbracket \tau \sqcap K \sqcap K' \rrbracket$.

These two facts are sufficient to complete the case.

3) We proceed by case analysis on K' (note by the fact that $\vdash \Sigma', K \propto K'$):

- (a) $K' = \text{Nat}$: Since we already know $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^V \llbracket N \rrbracket \Psi(\ell)$, it suffices to show $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket K' \rrbracket$ and $(k, \Psi, \Sigma, \ell') \in \mathcal{V}^N \llbracket K \rrbracket$.

This is immediate from $\vdash \Sigma'$, which implies $K' \propto \text{pointsto}(\Sigma', \ell')$ and $K \propto \text{pointsto}(\Sigma', \ell')$.

- (b) $K' = \text{Int}$: same as the Nat case.

- (c) $K' = \text{Bool}$: same as the Nat case.

- (d) $K' = * \times *$: this case is a contradiction by the fact that $\vdash \Sigma$.

- (e) $K' = * \rightarrow *$: Since $\text{pointsto}(\Sigma, \ell) \propto K'$ and $\text{pointsto}(\Sigma, \ell) \propto K$, $K = * \text{ or } * \rightarrow *$.

Also, since $\vdash \Sigma'$, we get that $\Psi(\ell) = [* , \bar{\tau}']$ or $[* \rightarrow *, \bar{\tau}']$.

From the fact that $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket \Psi(\ell) \rrbracket$, we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket [* , \bar{\tau}'] \rrbracket$ or $(k, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket [* \rightarrow *, \bar{\tau}'] \rrbracket$.

In the case of $*$, we can unfold and get $(k - 1, \Psi, \Sigma, \ell) \in \mathcal{V}\mathcal{H}^T \llbracket [* \rightarrow *, \bar{\tau}'] \rrbracket$.

Otherwise we can get the same using Lemma 5.51.

Similarly, we want to show that $(k, \Psi', \Sigma', \ell') \in \mathcal{V}\mathcal{H}^T \llbracket K', K, \Psi(\ell) \rrbracket$.

By Lemma 5.51, in the $K' = *$ case, it suffices to show $(k, \Psi', \Sigma', \ell') \in \mathcal{V}\mathcal{H}^T \llbracket [* \rightarrow *, K, \Psi(\ell)] \rrbracket$.

So let $(j, \Psi'') \sqsupseteq (k, \Psi')$, and let $\Sigma'' \supseteq \Sigma'$ such that $\Sigma'' : (j, \Psi'')$.

Let $\ell_v \in \text{dom}(\Sigma'')$ such that $(j, \Psi'', \Sigma'', \ell_v) \in \mathcal{V}^T \llbracket [*] \rrbracket$.

Let K'' .

We want to show $(j, \Psi'', \Sigma'', \text{app}\{K''\} \ell' \ell_v) \in \mathcal{E}\mathcal{H}^T \llbracket K'', *, \text{cod}(\Psi(\ell)) \rrbracket$.

By the OS, $(\Sigma'', \text{app}\{K''\} \ell' \ell_v) \rightarrow_T (\Sigma'', \text{assert } K'' (\ell' \ell_v))$.

By Lemma 5.60, it suffices to show $(j - 1, \Psi'', \Sigma'', \ell' \ell_v) \in \mathcal{E}\mathcal{H}^T \llbracket [* , *, \text{cod}(\Psi(\ell))] \rrbracket$.

By the OS, $(\Sigma'', \ell' \ell_v) \rightarrow_T (\Sigma'', \text{mon } \{* \leftarrow *\} (\ell (\text{mon } \{* \leftarrow *\} \ell_v)))$.

By IH 2) (smaller by index), it suffices to show $(j - 2, \Psi'', \Sigma'', \ell (\text{mon } \{* \leftarrow *\} \ell_v)) \in \mathcal{E}\mathcal{H}^T \llbracket [* , *, \text{cod}(\Psi(\ell))] \rrbracket$.

By Lemma 5.68, it suffices to show $(j - 1, \Psi'', \Sigma'', \text{assert } * \ell (\text{mon } \{* \leftarrow *\} \ell_v)) \in \mathcal{E}\mathcal{H}^T \llbracket [* , *, \text{cod}(\Psi(\ell))] \rrbracket$.

Then by the OS, it suffices to show $(j, \Psi'', \Sigma'', \text{app}\{*\} \ell (\text{mon } \{* \leftarrow *\} \ell_v)) \in \mathcal{E}\mathcal{H}^T \llbracket [* , *, \text{cod}(\Psi(\ell))] \rrbracket$.

By IH 2), $(j, \Psi'', \Sigma'', \text{mon } \{* \leftarrow *\} \ell_v) \in \mathcal{V}^T \llbracket [*] \rrbracket$.

Unfolding, we get that there exists some j', e'', Σ''' such that $(\Sigma'', \text{mon } \{* \leftarrow *\}) \rightarrow_T^{j'} (\Sigma''', e')$.

If $e' = \text{Err}^\bullet$, then we're done because the entire application errors.

Otherwise, we get that there exists a $(j - j', \Psi''') \sqsupseteq (j, \Psi'')$ such that $\Sigma''' : (j - j', \Psi''')$ and $(j - j', \Psi''', \Sigma''', e'') \in \mathcal{V}^T \llbracket [*] \rrbracket$.

Note by the operational semantics, $j' \geq 1$.

By Lemma 5.51, we get $(j - j', \Psi''', \Sigma''', \ell) \in \mathcal{V}\mathcal{H}^T \llbracket [* \rightarrow *, \bar{\tau}'] \rrbracket$.

Finally we can apply this hypothesis to the fact about e'' to get that $(j - j', \Psi''', \Sigma''', \text{app}\{*\} \ell e'') \in \mathcal{E}\mathcal{H}^T \llbracket [* , *, \text{cod}(\Psi(\ell))] \rrbracket$, which is sufficient to complete the case.

- (f) $K' = *$: unfolding the relation in what we want to show, the proof follows by IH 3) (smaller by index).

4) Unfolding the expression relation in our hypothesis, we have that there are (e', Σ') , j such that $(e, \Sigma) \xrightarrow{T}^j (e', \Sigma')$ with (e', Σ') irreducible.

If $e' = \text{Err}^\bullet$ then we're done, because the monitor will step to an error as well.

Otherwise, there is $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{VH}^T \llbracket \bar{\tau} \rrbracket$.

This means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.

We want to show $(k - j, \Psi', \Sigma', \text{mon} \{ * \Leftarrow * \} \ell) \in \mathcal{EH}^T \llbracket *, *, \Psi'(\ell) \rrbracket$.

For ii), by OS, if $\Sigma'(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$, then $(\Sigma', \text{mon} \{ * \Leftarrow * \} \ell) \xrightarrow{T} (\Sigma', \langle \text{mon} \{ * \Leftarrow * \} \ell_1, \text{mon} \{ * \Leftarrow * \} \ell_2 \rangle)$.

Then by Lemma 5.63, it suffices to show $(k - j - j' - 1, \Psi, \Sigma, \text{mon} \{ * \Leftarrow * \} \ell_1) \in \mathcal{VH}^T \llbracket *, *, \tau \rrbracket$ and $(k - j - j' - 1, \Psi, \Sigma, \text{mon} \{ * \Leftarrow * \} \ell_2) \in \mathcal{VH}^T \llbracket *, *, \tau \rrbracket$.

Both of these follow from (4) (smaller by index).

Otherwise, by the OS, $(\Sigma', \text{mon} \{ * \Leftarrow * \} \ell) \xrightarrow{T} (\Sigma'[\ell' \mapsto (\ell, \text{some}(*, *)), \ell'])$.

We can finish the proof by applying IH 3) (smaller by index).

□

LEMMA 5.70 (EXPRESSION RELATION IMPLIES ERRORING EXPRESSION RELATION). (1) If $(k, \Psi, \Sigma, e) \in \mathcal{E}^T \llbracket \tau \rrbracket$ then

$(k, \Psi, \Sigma, e) \in \mathcal{EH}^T \llbracket \tau \rrbracket$.

(2) If $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$ then $(k, \Psi, \Sigma, \ell) \in \mathcal{VH}^T \llbracket \tau \rrbracket$.

PROOF. Proceed by induction on k and τ :

- $k = 0$: 1) is immediate from 2).

- $\tau = \text{Int}$: immediate.

- $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

The case follows from the IH on ℓ_1 and ℓ_2 .

- $\tau = \tau_1 \rightarrow \tau_2$: vacuously true.

- $\tau = *$: vacuously true.

- $k > 0$: 1) is immediate from 2).

- $\tau = \text{Int}$: immediate.

- $\tau = \tau_1 \times \tau_2$: then $\Sigma(\ell) = (\langle \ell_1, \ell_2 \rangle, _)$.

The case follows from the IH on ℓ_1 and ℓ_2 .

- $\tau = \tau_1 \rightarrow \tau_2$: Follows from 1) from the IH (smaller by index).

- $\tau = *$: Follows from 2) from the IH (smaller by index), using $* \times *, * \rightarrow *,$ or Int .

□

5.4.3 Compatability Lemmas

LEMMA 5.71 (T-VAR COMPATIBILITY).
$$\frac{(x_0 : K_0) \in \Gamma}{\Gamma \vdash x_0 : K_0}$$

3121 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3122 We want to show $(k, \Psi, \Sigma, \gamma(x)) \in \mathcal{E}^T \llbracket \tau \rrbracket$.

3123 Since $x : \tau \in \Gamma$, we get that $\gamma(x) = \ell$.

3124 Since $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$, we get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

3126 Then we get that $(k, \Psi, \Sigma, \ell) \in \mathcal{E}^T \llbracket \tau \rrbracket$ immediately since ℓ is already a value and we have as a premise that $\Sigma : (k, \Psi)$. \square

3128 **LEMMA 5.72 (T-NAT COMPATIBILITY).** $\frac{}{\llbracket \Gamma \vdash n_0 : \text{Nat} \rrbracket}$

3131 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3132 We want to show $(k, \Psi, \Sigma, \gamma(n)) \in \mathcal{E}^T \llbracket \text{Nat} \rrbracket$.

3133 Note $\gamma(n) = n$.

3134 By the OS, we have $(\Sigma, n) \longrightarrow_T (\Sigma[\ell \mapsto (n, \text{none})], \ell)$.

3136 We get $(k, \Psi, \Sigma, \ell) \in \mathcal{V}^T \llbracket \text{Nat} \rrbracket$ immediately because $n \in \mathbb{T}$.

3137 Since $\mathcal{V}^T \llbracket \text{Nat} \rrbracket$ does not rely on Ψ or Σ , we have that $(k, \Psi[\ell \mapsto [\text{Nat}]], \Sigma[\ell \mapsto (n, \text{none})], \ell) \in \mathcal{V}^T \llbracket \text{Nat} \rrbracket$.

3138 Since $\ell \mapsto \text{Nat}$, we have that $(k, \Psi[\ell \mapsto [\text{Nat}]], \Sigma[\ell \mapsto (n, \text{none})], \ell) \in \mathcal{V}^T \llbracket \text{Nat} \rrbracket$.

3140 Similarly we have $(k, \Psi[\ell \mapsto [\text{Nat}]], \Sigma[\ell \mapsto (n, \text{none})], \ell) \in \mathcal{V}^{\mathcal{H}^V} \llbracket T \rrbracket \text{Nat}$.

3141 Therefore, given we know $\Sigma : (k, \Psi)$, we know $\Sigma[\ell \mapsto (n, \text{none})] : (k, \Psi[\ell \mapsto [\text{Nat}]])$. \square

3143 **LEMMA 5.73 (T-INT COMPATIBILITY).** $\frac{}{\llbracket \Gamma \vdash i_0 : \text{Int} \rrbracket}$

3146 **PROOF.** Not meaningfully different from **T-Nat** \square

3148 **LEMMA 5.74 (T-TRUE COMPATIBILITY).** $\frac{}{\llbracket \Gamma \vdash \text{True} : \text{Bool} \rrbracket}$

3150 **PROOF.** Not meaningfully different from **T-Nat** \square

3152 **LEMMA 5.75 (T-FALSE COMPATIBILITY).** $\frac{}{\llbracket \Gamma \vdash \text{False} : \text{Bool} \rrbracket}$

3155 **PROOF.** Not meaningfully different from **T-Nat** \square

3157 **LEMMA 5.76 (T-LAM COMPATIBILITY).** $\frac{\llbracket \Gamma_0, (x_0 : K_0) \vdash e_0 : \tau_1 \rrbracket}{\llbracket \Gamma_0 \vdash \lambda(x_0 : K_0). e_0 : * \rightarrow \tau_1 \rrbracket}$

3160 **PROOF.** Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3161 We want to show $(k, \Psi, \Sigma, \gamma(\lambda x_1 : K. e_1)) \in \mathcal{E}^T \llbracket * \rightarrow \tau_1 \rrbracket$.

3163 Note that $\gamma(\lambda x_1 : K. e_1) = \lambda x_1 : K. \gamma(e_1)$.

3164 Since $\lambda x_1 : K. \gamma(e_1)$ is a value, by the OS we have $(\Sigma, \lambda x_1 : K. \gamma(e_1)) \longrightarrow_T (\Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})], \ell)$, where $\ell \notin \text{dom}(\Sigma)$.

3167 We choose our later Ψ' to be $\Psi[\ell \mapsto * \rightarrow *]$.

3168 We now have two obligations:

3170 (1) $(k - 1, \Psi[\ell \mapsto * \rightarrow *], \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})], \ell) \in \mathcal{V}^T \llbracket * \rightarrow \tau_1 \rrbracket$

3171 (2) $\Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})] : (k - 1, \Psi[\ell \mapsto * \rightarrow *])$

3172 2023-04-10 15:45. Page 61 of 1-104.

3173 For 1), we want to show $(k-1, \Psi[\ell \mapsto * \rightarrow *], \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})], \lambda x_1 : K. \gamma(e_1)) \in \mathcal{V}^T \llbracket * \rightarrow \tau_1 \rrbracket$.

3174 Unfolding the value relation:

3175 Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto * \rightarrow *])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})]$ such that $\Sigma' : (j, \Psi')$.

3176 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket * \rrbracket$.

3177 Let K .

3178 We want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket \tau_1 \sqcap K \rrbracket$.

3179 By the OS, if $\neg K \propto \Sigma(\ell_v)$ then the application steps to an error and we're done.

3180 Otherwise, $(\Sigma', \text{app}\{K\} \ell \ell_v) \longrightarrow_T (\Sigma', \text{assert } K \gamma(e_1)[\ell_v/x])$.

3181 By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.

3182 Note that $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^T \llbracket \Gamma, x : K \rrbracket$:

- 3183 i) $(j-2, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket K \rrbracket$ by Lemma 5.55 and Lemma 5.57.
- 3184 ii) $\forall y \in \text{dom}(\gamma), (j-2, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^T \llbracket \Gamma(y) \rrbracket$ by the premise about γ and Lemma 5.55.

3185 Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma'$, and e_1 at $j-2$ to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

3186 Finally, we can apply Lemma 5.58 to get $(j-1, \Psi', \Sigma', \text{assert } K \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket \tau_1 \sqcap K \rrbracket$ which is what we wanted to show.

3187 For 2), first note the domains are equal, since $\text{dom}(\Sigma) = \text{dom}(\Psi)$.

3188 Then note $\vdash \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})]$ since $\vdash \Sigma$.

3189 Then let $j < k-1$ and let $\ell' \in \text{dom}(\Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})])$.

3190 If $\ell' \neq \ell$, then we get the remaining conditions from $\Sigma : (k, \Psi)$ and Lemma 5.51.

3191 If $\ell' = \ell$, then note the structural obligation on $\Psi[\ell \mapsto [* \rightarrow *]]$ is immediate.

3192 We want to show $(j, \Psi[\ell \mapsto * \rightarrow *], \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), \text{none})], \ell) \in \mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.

3193 Let $(j, \Psi') \sqsupseteq (k-1, \Psi[\ell \mapsto * \rightarrow *])$ and $\Sigma' \supseteq \Sigma[\ell \mapsto (\lambda x_1 : K. \gamma(e_1), _)]$ such that $\Sigma' : (j, \Psi')$.

3194 Let $\ell_v \in \text{dom}(\Sigma')$ such that $(j, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket * \rrbracket$.

3195 Let K .

3196 We get immediately that $\text{pointsto}(\Sigma', \ell_v) \propto *$, so we want to show $(j, \Psi', \Sigma', \text{app}\{K\} \ell \ell_v) \in \mathcal{E}^T \llbracket * \sqcap K \rrbracket$.

3197 By the OS, if $\neg K \propto \Sigma(\ell_v)$, then the application errors and we're done. Otherwise, $(\Sigma', \text{app}\{K\} \ell \ell_v) \longrightarrow_T (\Sigma', \text{assert } K \gamma(e_1)[\ell_v/x])$.

3198 By the definition of substitution, $\gamma(e_1)[\ell_v/x] = \gamma[x \mapsto \ell_v](e_1)$.

3199 Note that $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{G}^T \llbracket \Gamma, x : * \rrbracket$:

- 3200 i) $(j-2, \Psi', \Sigma', \ell_v) \in \mathcal{V}^T \llbracket K \rrbracket$ by Lemma 5.55 and Lemma 5.57.
- 3201 ii) $\forall y \in \text{dom}(\gamma), (j-2, \Psi', \Sigma', \gamma(y)) \in \mathcal{V}^T \llbracket \Gamma(y) \rrbracket$ by the premise about γ and Lemma 5.55.

3202 Therefore, we can apply the hypothesis to $\gamma[x \mapsto \ell_v], \Psi', \Sigma'$, and e_1 at $j-2$ to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

3203 Then we can apply Lemma 5.70 to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

3204 We can then apply Lemma 5.61 to get $(j-2, \Psi', \Sigma', \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket * \rrbracket$.

3205 Finally, we can apply Lemma 5.58 to get $(j-1, \Psi', \Sigma', \text{assert } K \gamma[x \mapsto \ell_v](e_1)) \in \mathcal{E}^T \llbracket * \sqcap K \rrbracket$ which is what we wanted to show.

□

$$\text{LEMMA 5.77 (T-PAIR COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket \\ \llbracket \Gamma \vdash e_1 : \tau_1 \rrbracket \end{array}}{\llbracket \Gamma \vdash \langle e_0, e_1 \rangle : \tau_0 \times \tau_1 \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\langle e_1, e_2 \rangle)) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Note $\gamma(\langle e_1, e_2 \rangle) = \langle \gamma(e_1), \gamma(e_2) \rangle$.

We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

We can apply the second hypothesis to get $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

Then by Lemma 5.63, $(k, \Psi, \Sigma, \langle \gamma(e_1), \gamma(e_2) \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$, which is what we wanted to show. \square

$$\text{LEMMA 5.78 (T-CAST COMPATIBILITY). } \frac{\llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket}{\llbracket \Gamma \vdash \text{cast } \{K_1 \leftarrow K_0\} e_0 : K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{cast } \{K_1 \leftarrow K_0\} e_0)) \in \mathcal{E}^T \llbracket K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket$.

Note $\gamma(\text{cast } \{K_1 \leftarrow K_0\} e_0) = \text{cast } \{K_1 \leftarrow K_0\} \gamma(e_0)$.

We can apply the first hypothesis to get $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T \llbracket \tau_0 \rrbracket$.

Unfolding the expression relation, there are j, Σ', e' such that $(\Sigma, \gamma(e_0)) \rightarrow_T^j (\Sigma', e')$ where (Σ', e') is irreducible.

If $e' = \text{Err}^\bullet$ then we're done, because the entire boundary expression errors.

Otherwise, we know there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e') \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$.

This means $\exists \ell \in \text{dom}(\Sigma')$ such that $e' = \ell$.

By the OS, $(\Sigma, \text{cast } \{K_1 \leftarrow K_0\} \gamma(e_0)) \rightarrow_T^j (\Sigma', \text{cast } \{K_1 \leftarrow K_0\} \ell) \rightarrow_T (\Sigma', \text{mon } \{K_1 \leftarrow K_0\} \ell)$.

By Lemma 5.55, $(k - j - 1, \Psi', \Sigma', \ell) \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$.

By Lemma 5.69, $(k - j - 1, \Psi', \Sigma', \text{mon } \{K_1 \leftarrow K_0\} \ell) \in \mathcal{E}^T \llbracket K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket$, which is what we wanted to show. \square

$$\text{LEMMA 5.79 (T-APP COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma \vdash e_0 : * \rightarrow \tau_1 \rrbracket \\ \llbracket \Gamma \vdash e_1 : \tau'_0 \rrbracket \end{array}}{\llbracket \Gamma \vdash \text{app}\{K_1\} e_0 e_1 : K_1 \sqcap \tau_1 \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{app}\{K_1\} e_1 e_2)) \in \mathcal{E}^T \llbracket K_1 \sqcap \tau_1 \rrbracket$.

Note $\gamma(\text{app}\{K_1\} e_1 e_2) = \text{app}\{K_1\} \gamma(e_1) \gamma(e_2)$.

By the first hypothesis we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T \llbracket * \rightarrow \tau_1 \rrbracket$.

By the second hypothesis we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T \llbracket \tau'_0 \rrbracket$.

By Lemma 5.61, we have $(k, \Psi, \Sigma, \gamma(e_2)) \in \mathcal{E}^T \llbracket * \rrbracket$.

Then we can apply Lemma 5.64 to get $(k, \Psi, \Sigma, \text{app}\{K_1\} \gamma(e_1) \gamma(e_2)) \in \mathcal{E}^T \llbracket \tau_1 \sqcap K_1 \rrbracket$ which is what we wanted to show. \square

$$\text{LEMMA 5.80 (T-APPBOT COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma \vdash e_0 : \perp \rrbracket \\ \llbracket \Gamma \vdash e_1 : \tau'_0 \rrbracket \end{array}}{\llbracket \Gamma \vdash \text{app}\{K_1\} e_0 e_1 : \perp \rrbracket}$$

3277 PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3278 We want to show $(k, \Psi, \Sigma, \gamma(\text{app}\{K_1\} e_0 e_1)) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

3279 By Lemma 5.56, we have that $(\Sigma, e_0) \xrightarrow{T}^* (\Sigma', e'_0)$ where $e'_0 = \text{Err}^\bullet$, which is sufficient to complete the case. \square

3281
3282 LEMMA 5.81 (**T-Fst** COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \times \tau_1 \rrbracket}{\llbracket \Gamma \vdash \text{fst}\{K_0\} e_0 : K_0 \sqcap \tau_0 \rrbracket}$$

3285 PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma_1 \rrbracket$ such that $\Sigma : (k, \Psi)$.

3286 We want to show $(k, \Psi, \Sigma, \gamma(\text{fst}\{K_0\} e_0)) \in \mathcal{E}^T \llbracket \tau_0 \sqcap K_0 \rrbracket$.

3287 Note $\gamma(\text{fst}\{K_0\} e_1) = \text{fst}\{K_0\} \gamma(e_0)$.

3288 From the first hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T \llbracket \tau_0 \times \tau_1 \rrbracket$.

3290 Unfolding the expression relation, there are j, Σ', e'_0 such that $(\Sigma, \gamma(e_0)) \xrightarrow{T}^j (\Sigma', e'_0)$ and e'_0 is irreducible.

3291 If $e'_0 = \text{Err}^\bullet$ then we're done because the projection also steps to an error.

3292 Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_0) \in \mathcal{V}^T \llbracket \tau_0 \times \tau_1 \rrbracket$.

3293 Unfolding the location and value relations, we get that $\Sigma'(e'_0) = (\langle \ell_0, \ell_1 \rangle, _)$.

3294 By the OS, $(\Sigma, \text{fst}\{K_0\} e_0) \xrightarrow{N}^j (\Sigma' \text{fst}\{K_0\} e'_0) \xrightarrow{T} (\Sigma', \text{assert } K_0 \ell_0)$.

3296 We can apply Lemma 5.55 to the premise that $(k - j, \Psi', \Sigma', \ell_0) \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$ to get $(k - j - 1, \Psi', \Sigma', \ell_0) \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$.

3297 Then we can apply Lemma 5.58 to get $(k - j - 1, \Psi', \Sigma', \text{assert } K_0 \ell_0) \in \mathcal{E}^T \llbracket \tau_0 \sqcap K_0 \rrbracket$.

3299 Finally, we can apply Lemma 5.51 to get that $\Sigma' : (k - j - 1, \Psi')$, which is sufficient to complete the proof. \square

3300
3301 LEMMA 5.82 (**T-FstBot** COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash e_0 : \perp \rrbracket}{\llbracket \Gamma \vdash \text{fst}\{K_0\} e_0 : \perp \rrbracket}$$

3304 PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3305 We want to show $(k, \Psi, \Sigma, \gamma(\text{fst}\{K_0\} e_0)) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

3306 By Lemma 5.56, we have that $(\Sigma, e_0) \xrightarrow{T}^* (\Sigma', e'_0)$ where $e'_0 = \text{Err}^\bullet$, which is sufficient to complete the case. \square

3308
3309 LEMMA 5.83 (**T-Snd** COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \times \tau_1 \rrbracket}{\llbracket \Gamma \vdash \text{snd}\{K_1\} e_0 : K_1 \sqcap \tau_1 \rrbracket}$$

3311 PROOF. Not meaningfully different from the **T-Fst** case. \square

3314
3315 LEMMA 5.84 (**T-SndBot** COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash e_0 : \perp \rrbracket}{\llbracket \Gamma \vdash \text{snd}\{K_1\} e_0 : \perp \rrbracket}$$

3317 PROOF. Not meaningfully different from the **T-FstBot** case. \square

3318
3319
3320 LEMMA 5.85 (**T-Binop** COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash e_0 : \tau_0 \rrbracket \quad \llbracket \Gamma \vdash e_1 : \tau_1 \rrbracket}{\llbracket \Gamma \vdash \text{binop } e_0 e_1 : \Delta(\text{binop}, \tau_0, \tau_1) \rrbracket}$$

3323 PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

3324 We want to show $(k, \Psi, \Sigma, \gamma(\text{binop } e_0 e_1)) \in \mathcal{E}^T \llbracket K_2 \rrbracket$.

3325 Note $\gamma(\text{binop } e_0 e_1) = \text{binop } \gamma(e_0) \gamma(e_1)$.

3327 By the first hypothesis applied to γ we have $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T \llbracket \tau_0 \rrbracket$.

3328

3329 Unfolding we get there are j, Σ', e'_0 such that $(\Sigma, \gamma(e_0)) \rightarrow_T^j (\Sigma', e'_0)$ and e'_0 is irreducible.

3330 If $e'_0 = \text{Err}^\bullet$ then we're done, because the whole operation errors.

3331 Otherwise there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_0) \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$.

3332

3333

3334 Note by Lemma 5.55 and Lemma 5.51, we have $(k - j, \Psi', \Sigma', \gamma) \in \mathcal{G}^T \llbracket \Gamma_1 \rrbracket$ and $\Sigma' : (k - j, \Psi')$.

3335 By the second hypothesis applied to γ we have $(k - j, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

3336 Unfolding we get there are j', Σ'', e'_1 such that $(\Sigma', \gamma(e_1)) \rightarrow_T^{j'} (\Sigma'', e'_1)$ and e'_1 is irreducible.

3337 If $e'_1 = \text{Err}^\bullet$ then we're done, because the whole operation errors.

3338 Otherwise, there is a $(k - j - j', \Psi'') \sqsupseteq (k - j, \Psi)$ such that $\Sigma'' : (k - j - j', \Psi'')$ and $(k - j - j', \Psi'', \Sigma'', e'_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$.

3339

3340

3341 From the definition of Δ , $K_2 = \text{Int}$ or Nat or \perp .

3342 In the case of \perp , we're done because either τ_0 or τ_1 is a \perp , which is a contradiction.

3343 Otherwise, the cases proceed identically, so without loss of generality assume $K_2 = \text{Int}$.

3344 $\tau_0 = \tau_1 = \text{Int}$, and therefore $\text{pointsto}(\Sigma'', ()e'_0) = i_0$ and $\text{pointsto}(\Sigma'', e'_1) = i_1$.

3345 If $\text{binop} = \text{quotient}$ and $i_1 = 0$ then $(\Sigma'', \text{binop } e'_0 e'_1) \rightarrow_T (\Sigma'', \text{DivErr})$, so we're done.

3346 If $\text{binop} = \text{quotient}$ and $i_1 \neq 0$, then $(\Sigma'', \text{binop } e'_0 e'_1) \rightarrow_T (\Sigma'', i_0/i_1) \rightarrow_T (\Sigma''[\ell \mapsto (i_0/i_1, \text{none})], \ell)$.

3347 Since $i_0/i_1 \in \mathbb{Z}$, we're done.

3348 If $\text{binop} = \text{sum}$ then $(\Sigma'', \text{binop } e'_0 e'_1) \rightarrow_T (\Sigma'', i_0 + i_1) \rightarrow_T (\Sigma''[\ell \mapsto (i_0 + i_1, \text{none})], \ell)$.

3349 Since $i_0 + i_1 \in \mathbb{Z}$, we're done. □

3350

3351

3352

3353

3354

3355

3356

3357

3358

3359

3360

3361

3362

3363

3364

3365

3366

3367

3368

3369

3370

3371

3372

3373

3374

3375

3376

3377

3378

3379

3380

$$\text{LEMMA 5.86 (T-IF COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma \vdash e_0 : \text{Bool} \rrbracket \\ \llbracket \Gamma \vdash e_1 : \tau_0 \rrbracket \\ \llbracket \Gamma \vdash e_2 : \tau_1 \rrbracket \end{array}}{\llbracket \Gamma \vdash \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \tau_0 \sqcup \tau_1 \rrbracket}$$

Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{if } e_0 \text{ then } e_1 \text{ else } e_2)) \in \mathcal{E}^T \llbracket \tau_0 \sqcup \tau_1 \rrbracket$.

Note $\gamma(\text{if } e_0 \text{ then } e_1 \text{ else } e_2) = \text{if } \gamma(e_0) \text{ then } \gamma(e_1) \text{ else } \gamma(e_2)$.

From the first hypothesis applied to γ , we know $(k, \Psi, \Sigma, \gamma(e_0)) \in \mathcal{E}^T \llbracket \text{Bool} \rrbracket$.

Unfolding, we have that there is Σ', e'_0, j such that $(\Sigma, e_0) \rightarrow_T^j (\Sigma', e'_0)$ where e'_0 is irreducible.

If $e'_0 = \text{Err}^\bullet$ then we're done, because the entire if statement errors.

Otherwise, there is a $(k - j, \Psi') \sqsupseteq (k, \Psi)$ such that $\Sigma' : (k - j, \Psi')$ and $(k - j, \Psi', \Sigma', e'_0) \in \mathcal{V}^T \llbracket \text{Bool} \rrbracket$.

Unfolding the location and then the value relation, we get that $\text{pointsto}(\Sigma', e'_0) = \text{True}$ or $\text{pointsto}(\Sigma', e'_0) = \text{False}$.

- $\text{pointsto}(\Sigma', e'_0) = \text{True}$: Note by OS, $(\Sigma, \text{if } \gamma(e_0) \text{ then } \gamma(e_1) \text{ else } \gamma(e_2)) \rightarrow_T^j (\Sigma', \text{if } e'_0 \text{ then } \gamma(e_1) \text{ else } \gamma(e_2)) \rightarrow_T (\Sigma', \gamma(e_1))$.

By Lemma 5.55 and Lemma 5.51, we have $(k - j - 1, \Psi', \Sigma', \gamma) \in \mathcal{G}^T \llbracket \Gamma_1 \rrbracket$ and $\Sigma' : (k - j - 1, \Psi')$.

From the second hypothesis, we get $(k - j - 1, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_0 \rrbracket$.

Finally, by Lemma 5.61, we get $(k - j - 1, \Psi', \Sigma', \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_0 \sqcup \tau_1 \rrbracket$ which is sufficient to complete the proof.

- $\text{pointsto}(\Sigma', e'_0) = \text{False}$: same as other case except replace e_1 with e_2 .

3381 PROOF. □

3382

3383

3384

3385

3386

3387

3388

3389

3390

3391

3392

3393

3394

3395

3396

3397

3398

3399

3400

3401

3402

3403

3404

3405

3406

3407

3408

3409

3410

3411

3412

3413

3414

3415

3416

3417

3418

3419

3420

3421

3422

3423

3424

3425

3426

3427

3428

3429

3430

3431

3432

$$\llbracket \Gamma \vdash e_0 : \perp \rrbracket$$

$$\llbracket \Gamma \vdash e_1 : \tau_0 \rrbracket$$

$$\llbracket \Gamma \vdash e_2 : \tau_1 \rrbracket$$

$$\text{LEMMA 5.87 (T-IFBOT COMPATIBILITY). } \frac{}{\llbracket \Gamma \vdash \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \perp \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(\text{if } e_0 \text{ then } e_1 \text{ else } e_2)) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

By Lemma 5.56, we have that $(\Sigma, e_0) \xrightarrow{T}^* (\Sigma', e'_0)$ where $e'_0 = \text{Err}^\bullet$, which is sufficient to complete the case. □

$$\llbracket \Gamma \vdash e_1 : \tau_1 \rrbracket$$

$$\tau_1 \leq \tau_2$$

$$\text{LEMMA 5.88 (T-SUB COMPATIBILITY). } \frac{}{\llbracket \Gamma \vdash e_1 : \tau_2 \rrbracket}$$

PROOF. Let $(k, \Psi, \Sigma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$ such that $\Sigma : (k, \Psi)$.

We want to show $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

From our hypothesis, we have $(k, \Psi, \Sigma, \gamma(e_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

We can apply Lemma 5.61 to finish the case. □

5.4.4 Transient with Truer Transient Typing is Vigilant

THEOREM 5.89 (TRANSIENT WITH TRUER TRANSIENT TYPING IS VIGILANT). *If $\Gamma \vdash e : \tau$ then $\llbracket \Gamma \vdash e : \tau \rrbracket^T$*

PROOF. By induction over the typing derivation, using the compatibility lemmas. □

3433 5.5 Vigilance Fundamental Property for Transient with Tag Typing

3434
3435 THEOREM 5.90 (TRANSIENT IS TAG VIGILANT). *If $\Gamma \vdash_{\text{tag}} e : K$ then $\llbracket \Gamma \vdash_{\text{tag}} e : K \rrbracket^T$*

3436 PROOF. By Theorem 4.10, we have that there exists some $\tau \leq K$ such that $\Gamma \vdash_{\text{tru}} e : \tau$.

3437 By Theorem 5.89, we have that $\llbracket \Gamma \vdash_{\text{tru}} e : \tau \rrbracket^T$.

3439 Unfolding this result and what we want to prove, we note the only distinction is that in what we have, we get
3440 $(k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^T \llbracket \tau \rrbracket$, and what we want to prove is $(k, \Psi, \Sigma, \gamma(e)) \in \mathcal{E}^T \llbracket K \rrbracket$.

3441 This follows directly from Lemma 5.61. □

3443

3444

3445

3446

3447

3448

3449

3450

3451

3452

3453

3454

3455

3456

3457

3458

3459

3460

3461

3462

3463

3464

3465

3466

3467

3468

3469

3470

3471

3472

3473

3474

3475

3476

3477

3478

3479

3480

3481

3482

3483

3484

6 Contextual equivalence

6.1 Contextual Equivalence Logical Relation—No Store

DivErr \approx DivErr

TypeErr(τ, v) \approx TypeErr(τ', v')

$\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \tau \rrbracket_C^{\mathcal{L}} \triangleq \forall (k, \gamma_1, \gamma_2) \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]. (k, \gamma_1(e_1), \gamma_2(e_2)) \in \mathcal{E}^{\mathcal{L}}[\llbracket \tau \rrbracket]$

$\llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e_2 : \tau \rrbracket_C^{\mathcal{L}} \triangleq \llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \tau \rrbracket_C^{\mathcal{L}} \wedge \llbracket \Gamma \vdash_{\text{tru}} e_2 \leq e_1 : \tau \rrbracket_C^{\mathcal{L}}$

$\mathcal{G}^{\mathcal{L}}[\llbracket \Gamma, x : \tau \rrbracket] \triangleq \{(k, \gamma_1[x \mapsto v_1], \gamma_2[x \mapsto v_2]) \mid (k, \gamma_1, \gamma_2) \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]$
 $\wedge (k, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\llbracket \tau \rrbracket]_k\}$

$\mathcal{G}^{\mathcal{L}}[\bullet] \triangleq \{(k, \emptyset, \emptyset)\}$

$\mathcal{E}^{\mathcal{L}}[\llbracket \tau \rrbracket] \triangleq \{(k, e_1, e_2) \mid \forall j \leq k, e'_1. e_1 \xrightarrow{j}_L e'_1 \wedge \text{irred}_L(e'_1)$
 $\Rightarrow \exists e'_2. e_2 \xrightarrow{*}_L e'_2$
 $\wedge (e'_1 \approx e'_2 \in \text{Err}^\bullet \vee (k - j, e'_1, e'_2) \in \mathcal{V}^{\mathcal{L}}[\llbracket \tau \rrbracket])\}$

$\mathcal{V}^{\mathcal{L}}[\llbracket \text{Int} \rrbracket] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{Z})\}$

$\mathcal{V}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{N})\}$

$\mathcal{V}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket] \triangleq \{(k, v_1, v_2 \mid v_1 = v_2 \in \mathbb{B})\}$

$\mathcal{V}^{\mathcal{L}}[\llbracket \tau_1 \times \tau_2 \rrbracket] \triangleq \{(k, \langle v_{1,1}, v_{1,2} \rangle, \langle v_{2,1}, v_{2,2} \rangle) \mid (k, v_{1,1}, v_{2,1}) \in \mathcal{V}^{\mathcal{L}}[\llbracket \tau_1 \rrbracket] \wedge (k, v_{2,1}, v_{2,2}) \in \mathcal{V}^{\mathcal{L}}[\llbracket \tau_2 \rrbracket]\}$

$\mathcal{V}^{\mathcal{L}}[\llbracket \tau_1 \rightarrow \tau_2 \rrbracket] \triangleq \{(k, v_1, v_2) \mid \forall j \leq k,$

$\forall v'_1, v'_2 \text{ where } (j, v'_1, v'_2) \in \mathcal{V}^{\mathcal{L}}[\llbracket \tau_1 \rrbracket].$

$\forall K, K' \text{ where } K \sqcap \tau_2 = K' \sqcap \tau_2.$

$(j, \text{app}\{K\} v_1 v'_1, \text{app}\{K'\} v_2 v'_2) \in \mathcal{E}^{\mathcal{L}}[\llbracket K \sqcap \tau_2 \rrbracket]\}$

3537
 3538
 3539 $\mathcal{V}^{\mathcal{L}}[\![*]\!] \triangleq \{(k, \Sigma_1, \Sigma_2, \ell_1, \ell_2) \mid (k-1, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\![\text{Int}]\!]\}$
 3540 $(k-1, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\![\text{Bool}]\!]$
 3541 $\forall (k-1, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\![* \times *]\!]$
 3542 $\forall (k-1, v_1, v_2) \in \mathcal{V}^{\mathcal{L}}[\![* \rightarrow *]\!]\}$
 3543
 3544
 3545
 3546
 3547
 3548
 3549

3550 $\mathcal{V}^{\mathcal{L}}[\![\perp]\!] \triangleq \emptyset$
 3551
 3552
 3553
 3554
 3555
 3556
 3557
 3558
 3559
 3560
 3561
 3562
 3563
 3564
 3565
 3566
 3567
 3568
 3569
 3570
 3571
 3572
 3573

3574 6.2 Context typing

3575 Truer transient contexts:

3576 $E ::= [] \mid \lambda(x:K).E \mid \langle e, E \rangle \mid \langle E, e \rangle \mid \text{app}\{K\} e E \mid \text{app}\{K\} E e \mid \text{fst}\{K\} E \mid \text{snd}\{K\} E$
 3577 $\mid \text{binop} e E \mid \text{binop} E e \mid \text{cast}\{K \leftarrow K\} E \mid \text{if } E \text{ then } e \text{ else } e \mid \text{if } e \text{ then } E \text{ else } e \mid \text{if } e \text{ then } e \text{ else } E$
 3578
 3579

3580
 3581
 3582
 3583
 3584
 3585
 3586
 3587
 3588

3589

3590

3591

3592

3593

3594

3595

3596

3597

3598

3599

3600

3601

3602

3603

3604

3605

3606

3607

3608

3609

3610

3611

3612

3613

3614

3615

3616

3617

3618

3619

3620

3621

3622

3623

3624

3625

3626

3627

3628

3629

3630

3631

3632

3633

3634

3635

3636

3637

3638

3639

3640

T-CTX-HOLE

$$\frac{\Gamma' \subseteq \Gamma}{\Gamma \vdash_{\text{tru}} [] : (\Gamma' \triangleright \tau) \rightsquigarrow \tau}$$

T-CTX-LAM

$$\frac{\Gamma, (x:K) \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \tau'}{\Gamma \vdash_{\text{tru}} \lambda(x:K). E : (\Gamma', (x:K) \triangleright \tau) \rightsquigarrow * \rightarrow \tau'}$$

T-CTX-PAIR-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} e : \tau_2}{\Gamma \vdash_{\text{tru}} \langle E, e \rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2}$$

T-CTX-PAIR-2

$$\frac{\Gamma \vdash_{\text{tru}} e : \tau_1 \quad \Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \langle e, E \rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2}$$

T-CTX-APP-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow * \rightarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} e : \tau_2}{\Gamma \vdash_{\text{tru}} \text{app}\{K\} E e : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1}$$

T-CTX-APPBOT-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \perp \quad \Gamma \vdash_{\text{tru}} e : \tau_2}{\Gamma \vdash_{\text{tru}} \text{app}\{K\} E e : (\Gamma' \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-APP-2

$$\frac{\Gamma \vdash_{\text{tru}} e : * \rightarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \text{app}\{K\} e E : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1}$$

T-CTX-APPBOT-2

$$\frac{\Gamma \vdash_{\text{tru}} e : \perp \quad \Gamma \vdash_{\text{tru}} E : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \text{app}\{K\} e E : (\Gamma' \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-FST

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \text{fst}\{K\} E : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_1}$$

T-CTX-FSTBOT

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \perp}{\Gamma \vdash_{\text{tru}} \text{fst}\{K\} E : (\Gamma \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-SND

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \text{snd}\{K\} E : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_2}$$

T-CTX-SNDBOT

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \perp}{\Gamma \vdash_{\text{tru}} \text{snd}\{K\} E : (\Gamma \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-BINOP-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} e : \tau_2}{\Gamma \vdash_{\text{tru}} \text{binop} E e : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\text{binop}, \tau_1, \tau_2)}$$

T-CTX-BINOP-2

$$\frac{\Gamma \vdash_{\text{tru}} e : \tau_1 \quad \Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \text{binop} E e : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\text{binop}, \tau_1, \tau_2)}$$

T-CTX-BND-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau'}{\Gamma \vdash_{\text{tru}} \text{cast}\{K_2 \Leftarrow K_1\} E : (\Gamma \triangleright \tau) \rightsquigarrow K_2 \sqcap K_1 \sqcap \tau'}$$

T-CTX-IF-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \text{Bool} \quad \Gamma \vdash_{\text{tru}} e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} e_2 : \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } E \text{ then } e_1 \text{ else } e_2 : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2}$$

T-CTX-IFBOT-1

$$\frac{\Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \perp \quad \Gamma \vdash_{\text{tru}} e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} e_2 : \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } E \text{ then } e_1 \text{ else } e_2 : (\Gamma \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-IF-2

$$\frac{\Gamma \vdash_{\text{tru}} e_b : \text{Bool} \quad \Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} e_2 : \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } E \text{ else } e_2 : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2}$$

T-CTX-IFBOT-2

$$\frac{\Gamma \vdash_{\text{tru}} e_b : \perp \quad \Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \quad \Gamma \vdash_{\text{tru}} e_2 : \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } E \text{ else } e_2 : (\Gamma \triangleright \tau) \rightsquigarrow \perp}$$

T-CTX-IF-3

$$\frac{\Gamma \vdash_{\text{tru}} e_b : \text{Bool} \quad \Gamma \vdash_{\text{tru}} e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } e_1 \text{ else } E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2}$$

T-CTX-IFBOT-3

$$\frac{\Gamma \vdash_{\text{tru}} e_b : \perp \quad \Gamma \vdash_{\text{tru}} e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} E : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } e_1 \text{ else } E : (\Gamma \triangleright \tau) \rightsquigarrow \perp}$$

6.3 Contextual equivalence statement

We define a logical relation for contexts:

$$\llbracket \Gamma \vdash_{\text{tru}} C_1 \approx C_2 : (\Gamma' \triangleright \tau) \rightsquigarrow \tau' \rrbracket \triangleq \forall e_1, e_2. \llbracket \Gamma' \vdash_{\text{tru}} e_1 \approx e_2 : \tau \rrbracket \Rightarrow \llbracket \Gamma \vdash_{\text{tru}} C_1[e_1] \approx C_2[e_2] : \tau' \rrbracket$$

We define an abbreviation for the notion that an expression reduces to an eventual value without encountering an error: $e \Downarrow \triangleq \exists e'. e \longrightarrow_L^* e' \wedge (\text{val}(e'))$

THEOREM 6.1 (EXPRESSION RELATION IMPLIES REDUCTION EQUIVALENCE). *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e_2 : \tau \rrbracket$, then $e_1 \Downarrow \Leftrightarrow e_2 \Downarrow$.*

PROOF. By applying Lemm 6.2 in both directions. \square

LEMMA 6.2 (EXPRESSION RELATION IMPLIES REDUCTION EQUIVALENCE). *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \tau \rrbracket$, then $e_1 \Downarrow \Rightarrow e_2 \Downarrow$.*

PROOF. Since $e_1 \Downarrow$, then there exists some e'_1, k s.t. $e_1 \longrightarrow_L^k e'_1$ and e'_1 is a value and hence irreducible.

We want to show that $e'_2 \Downarrow$. Instantiate the premise with $(k, \emptyset, \emptyset)$, obtaining that $(k, e_1, e_2) \in \mathcal{E}^{\mathcal{L}} \llbracket \tau \rrbracket$. Instantiate j with k and e'_1 with e'_1 , observing that e'_1 being a value entails it is irreducible. Then e'_2 from this relation is just what we need, since e_2 reduces to it, and it is syntactically a value. \square

The usual definition of contextual equivalence is then:

$$\Gamma \vdash_{\text{tru}} e_1 \approx^{\text{ctx}} e_2 : \tau \triangleq \forall C, \bullet \vdash_{\text{tru}} C : (\Gamma \triangleright \tau) \rightsquigarrow \tau' \Rightarrow (C[e_1] \Downarrow \Leftrightarrow C[e_2] \Downarrow)$$

THEOREM 6.3 (BINARY RELATION IS SOUND FOR CONTEXTUAL EQUIVALENCE). *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e_2 : \tau \rrbracket$, then $\Gamma \vdash_{\text{tru}} e_1 \approx^{\text{ctx}} e_2 : \tau$.*

PROOF. Consider an arbitrary type τ' and context C s.t. $\bullet \vdash_{\text{tru}} C : (\Gamma \triangleright \tau) \rightsquigarrow \tau'$. Then we must show that $C[e_1] \Downarrow \Leftrightarrow C[e_2] \Downarrow$. By Theorem 6.1, it is sufficient to show that $\llbracket \bullet \vdash_{\text{tru}} C[e_1] \approx C[e_2] : \tau' \rrbracket$.

By Theorem 6.71, $\llbracket \bullet \vdash_{\text{tru}} C \approx C : (\Gamma \triangleright \tau) \rightsquigarrow \tau' \rrbracket$. Unfolding this definition and instantiating it with e_1, e_2 , and our hypothesis about them, we obtain precisely the required conclusion. \square

6.4 Binary relation—Proofs

6.4.1 Lemmas Used Without Mention

LEMMA 6.4 (VALUES ARE IN THE \mathcal{E} -RELATION). *If $(k, v, v') \in \mathcal{V}^{\mathcal{L}} \llbracket \tau \rrbracket$, then $(k, v, v') \in \mathcal{E}^{\mathcal{L}} \llbracket \tau \rrbracket$.*

PROOF. Consider arbitrary j s.t. $v \longrightarrow^j v_f \wedge \text{irred}_{\mathcal{L}}(v_f)$. Note that j must be equal to 0 since values do not reduce. Then choose v' as the e'_2 of the expression relation; it is easy to see that v' reduces to v in some number (0) of steps. By our assumption, $(k - 0, v, v') \in \mathcal{V}^{\mathcal{L}} \llbracket \tau \rrbracket$, so we are done. \square

LEMMA 6.5 (ANTI-REDUCTION - HEAD EXPANSION - EXPRESSION RELATION COMMUTES WITH STEPS). *If $(k, e'_1, e'_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $e_1 \longrightarrow_T^j e'_1$ and $e_2 \longrightarrow_T^{j'} e'_2$, then $(k + j, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$*

PROOF. Consider arbitrary j', e''_1 s.t. $e_1 \longrightarrow_T^{j'} e''_1$. If $j' \leq j$, by determinism of the operational semantics, e''_1 must not be irreducible and so we are trivially done. Otherwise, assume $\text{irred}_T(e''_1)$ and $j' \leq k + j$; we must show that $\exists e''_2. e_2 \longrightarrow_T^* e''_2 \wedge (e''_1 \approx e''_2 \in \text{Err}^\bullet \vee (k + j - j', e''_1, e''_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$.

Instantiate the hypothesis with $(k + j' - j, e''_1)$. Since $k + j' - j \leq k$ and the operational semantics are deterministic, this gives us that $\exists e''_2. e'_2 \longrightarrow_T^* e''_2 \wedge (e''_1 \approx e''_2 \in \text{Err}^\bullet \vee (k + j - j', e''_1, e''_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$, from which our conclusion follows immediately. \square

3693 LEMMA 6.6 (ANTI-REDUCTION - HEAD EXPANSION - STEPS COMMUTE WITH EXPRESSION RELATION). *If $(k + j, e_1, e_2) \in$*
 3694 *$\mathcal{E}^T \llbracket \tau \rrbracket$ and $e_1 \xrightarrow{T^j} e'_1$ and $e_2 \xrightarrow{T^{j'}} e'_2$, then $(k, e'_1, e'_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$*
 3695

3696 PROOF. Consider arbitrary j', e'_1 s.t. $j' \leq k \wedge \text{irred}_T(e'_1) \wedge e'_1 \xrightarrow{T^{j'}} e''_1$.

3697 We must show that $\exists e''_2. e'_2 \xrightarrow{T^*} e''_2 \wedge (e'_1 \approx e''_2 \in \text{Err}^\bullet \vee (k - j', e'_1, e''_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$.

3698 Instantiate the hypothesis with $j + j', e'_1$. Since $j' \leq k$, $j + j' \leq k + j$. Since the operational semantics are deterministic
 3699 and transitive, the other conditions apply. Then the hypothesis provides precisely the appropriate e''_2 and conditions on
 3700 it and e'_1 . □
 3701
 3702

3703 We define a notion of tags extended with bottom that are compatible with the usual lattice:

$$K^\perp = K \mid \perp$$

$$\lfloor K^\perp \rfloor^\perp = \begin{cases} \perp & \text{if } K^\perp = \perp \\ \lfloor K^\perp \rfloor & \text{otherwise} \end{cases}$$

$$\alpha^\perp(K^\perp, v) = \begin{cases} \text{False} & \text{if } K^\perp = \perp \\ v \alpha K^\perp & \text{otherwise} \end{cases}$$

3713 LEMMA 6.7 (TAGOF-BOT IS COMPATIBLE WITH MEET). $\lfloor K_1^\perp \sqcap K_2^\perp \rfloor^\perp = \lfloor K_1^\perp \rfloor^\perp \sqcap \lfloor K_2^\perp \rfloor^\perp$.
 3714

3715 PROOF. Immediate, by unfolding definitions and case analysis. □
 3716

3717 LEMMA 6.8 (RELATION IMPLIES TAGMATCH). *If $(k, v, v') \in \mathcal{V}^{\mathcal{L}} \llbracket \tau \rrbracket$ and $K^\perp \leq \lfloor \tau \rfloor^\perp$, then $\alpha^\perp(K^\perp, v)$.*
 3718

3719 PROOF. By case analysis on τ and K^\perp ; in each case this follows immediately from unfolding the definitions of \mathcal{V}
 3720 and tagmatch. □
 3721

3722 6.4.2 Lemmas Used With Mention

3723 LEMMA 6.9 (RELATED VALUES HAVE MATCHING CONSTRUCTORS). *If $(k, v, v') \in \mathcal{V}^{\mathcal{L}} \llbracket \tau \rrbracket$, then either*
 3724

- 3725 • $v = v'$
- 3726 • *There exist some v_1, v_2, v'_1, v'_2 s.t. $v = \langle v_1, v_2 \rangle$ and $v' = \langle v'_1, v'_2 \rangle$*
- 3727 • *There exist some w, w' s.t. $v = w$ and $v' = w'$.*

3728
 3729
 3730 PROOF. By induction on τ , unfolding the definition of \mathcal{V} in each case. □
 3731

3732 LEMMA 6.10 (TAGMATCH IS UP TO APPROXIMATION). *If $(k, v, v') \in \mathcal{V}^T \llbracket \tau \rrbracket$, then $\alpha^\perp(K^\perp, v) \Leftrightarrow \alpha^\perp(K^\perp, v')$.*
 3733

3734 PROOF. By Lemma 6.9 and inspection of the definition of $\alpha^\perp(K^\perp, v)$. □
 3735

3736 LEMMA 6.11 (TAGMATCH RESPECTS MEETS). $\alpha^\perp(K_1^\perp \sqcap K_2^\perp, v) \Leftrightarrow \alpha^\perp(K_1^\perp, v) \wedge \alpha^\perp(K_2^\perp, v)$.
 3737

3738 PROOF. By case analysis on K_1^\perp, K_2^\perp ; in each case the conclusion follows immediately by unfolding. □
 3739

3740 LEMMA 6.12 (TAGMATCH IMPLIES VALUES IN RELATION AT MEET). *If $(k, v, v') \in \mathcal{V}^T \llbracket \tau \rrbracket$ and $\alpha^\perp(K^\perp, v)$, then $(k -$*
 3741 *$1, v, v') \in \mathcal{V}^T \llbracket K^\perp \sqcap \tau \rrbracket$.*
 3742

3743 PROOF. Proceed by case analysis on K^\perp :
 3744

3745 * By lattice properties, $K^\perp \sqcap \tau = \tau$, so this is trivial by Lemma ??.

3746 Nat By the definition of tagmatch, v must be a natural number. By inspection, this is possible only when τ is *, Int, or

3747 Nat; in each case, $K^\perp \sqcap \tau = \text{Nat}$. By inspection on the relation, v always satisfied what is needed.

3748

3749 Int Analogous to the Nat case above.

3750 * \times * By the definition of tagmatch, v must be a pair; by inspection this is possible only if τ is * or some pair type. If

3751 the latter, $K^\perp \sqcap \tau = \tau$, and so the conclusion is immediate; otherwise, $K^\perp \sqcap \tau = * \times *$, and the conclusion is

3752 immediate from the definition of the * case of the relation.

3753

3754 * \rightarrow * By the definition of tagmatch, v must be a w ; by inspection this is possible only if τ is * or some function type. If

3755 the latter, $K^\perp \sqcap \tau = \tau$, and so the conclusion is immediate; otherwise, $K^\perp \sqcap \tau = * \rightarrow *$, and the conclusion is

3756 immediate from the definition of the * case of the relation.

3757

3758 \perp Contradiction

3759 □

3760

3761 LEMMA 6.13 (\mathcal{E} - \mathcal{V} -MONOTONICITY). (1) If $(k, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $j \leq k$, then $(j, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$.

3762 (2) If $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$ and $j \leq k$, then $(j, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

3763

3764 PROOF. Proceed by simultaneous induction on k and τ :

3765

- 3766 • $k = 0$: 1) follows immediately from 2).

3767 Proceeds similarly to the other case, but function and dynamic cases are vacuously true.

- 3768 • $k > 0$:

- 3769 1) Unfolding the expression relation in our hypothesis, we get that there is some e'_1, j' such that $e_1 \xrightarrow{T}^{j'} e'_1$,
- 3770 and some e'_2 such that $e_2 \xrightarrow{T}^* e'_2$.

3771 If $e'_1 = \text{Err}^\bullet$ then we're done.

3772 Otherwise, $(k - j', e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

3773

3774

3775

3776 Now, unfolding the expression relation, we want to show $(k - j - j', e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

3777 We can apply the IH 2) with the fact proven in a).

- 3778 2) We want to show that $(k - j, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

3779 We case split on τ :

- 3780 i) $\tau = \text{Nat}$: then where $n \in \mathbb{N}$, so the case is immediate.

- 3781 ii) $\tau = \text{tint}$: same as above.

- 3782 iii) $\tau = \text{Bool}$: same as above.

- 3783 iv) $\tau = \tau_1 \times \tau_2$: Then unfolding our hypothesis gives us $v_1 = \langle v'_1, v''_1 \rangle$ and $v_2 = \langle v'_1, v''_1 \rangle$ with $(k, v'_1, v''_1) \in$
- 3784 $\mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k, v'_1, v''_1) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$.

3785 The case follows by applying the IH 2) to both premises.

- 3786 v) $\tau = * \rightarrow \tau_2$: Let $j' \leq k - j$.

3787 Let $(j', v'_1, v'_2) \in \mathcal{V}^T \llbracket * \rrbracket$.

3788 Let K, K' .

3789

3790

3791

3792

3793

3794

3795

3796

3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848

We want to show $(j', \text{app}\{K\} v_1 v'_1, \text{app}\{K'\} v_2 v'_2) \in \mathcal{E}^T \llbracket K \sqcap \tau_2 \rrbracket$.

Since $j' \leq k - j \leq k$, we can apply the hypothesis to complete the case.

vi) $\tau = *$: we want to show $(k - 1, v_1, v_2) \in \mathcal{V}^T \llbracket \text{Int} \rrbracket$ or $\mathcal{V}^T \llbracket \text{Bool} \rrbracket$ or $\mathcal{V}^T \llbracket * \times * \rrbracket$ or $\mathcal{V}^T \llbracket * \rightarrow * \rrbracket$.

This follows from IH 2) (smaller by index). □

LEMMA 6.14 (MONADIC BIND). *Suppose that E_1, E_2 are any evaluation contexts (n.b. not a general context, as used elsewhere in these proofs), $(k, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$, and for all k', v_1, v_2 , if $k' \leq k \wedge (k', v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$ then $(k', E_1[v_1], E_2[v_2]) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.*

Then $(k, E_1[e_1], E_2[e_2]) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

PROOF. Consider arbitrary j, e'_1 s.t. $j \leq k \wedge E_1[e_1] \xrightarrow{j}_T e'_1 \wedge \text{irred}_T(e'_1)$. Then we must show that must show that $\exists e'_2. E_2[e_2] \xrightarrow{*}_T e'_2 \wedge (e'_1 \approx e'_2 \in \text{Err}^\bullet \vee (k - j, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$.

Because $E_1[e_1]$ reaches an irreducible term in at most j steps, by our operational semantics e_1 must itself reduce to some irreducible term e_3 in some smaller number of steps $j' \leq j$. Then since $j' \leq j \wedge e_1 \xrightarrow{j'}_T e_3 \wedge \text{irred}_T(e_3)$, we can instantiate our first assumption, obtaining that there similarly exists e_4 s.t. $e_2 \xrightarrow{*}_T e_4 \wedge (e_3 \approx e_4 \in \text{Err}^\bullet \vee (k - j', e_3, e_4) \in \mathcal{V}^T \llbracket \tau \rrbracket)$.

Suppose that $e_3 \approx e_4 \in \text{Err}^\bullet$. Then by the operational semantics, $E_1[e_1]$ and $E_2[e_2]$ reduce to the same errors, so instantiating e'_1 and e'_2 with them proves our goal.

Otherwise, we know that $(k - j', e_3, e_4) \in \mathcal{V}^T \llbracket \tau \rrbracket$. We may therefore instantiate our other assumption with $k - j', e_3, e_4$ and this fact, obtaining that $(k - j', E_1[e_3], E_2[e_4]) \in \mathcal{E}^T \llbracket \tau \rrbracket$. We still must show that $\exists e'_2. E_2[e_2] \xrightarrow{*}_T e'_2 \wedge (e'_1 \approx e_2 \in \text{Err}^\bullet \vee (k - j, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$.

Instantiate the result of our assumption with step index $j - j' \leq k - j'$ and e'_1 . By determinism of the operational semantics, $E_1[e_3] \xrightarrow{j - j'}_T e'_1$, so we obtain that $\exists e'_2. E_2[e_4] \xrightarrow{*}_T e'_2 \wedge (e'_1 \approx e'_2 \in \text{Err}^\bullet \vee (k - j' - (j - j'), e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket)$. Note that $k - j' - (j - j') = k - j$, and that since $E_2[e_4] \xrightarrow{*}_T e'_2$ and $e_2 \xrightarrow{*}_T e_4$, then $E_2[e_2] \xrightarrow{*}_T e'_2$, so this is precisely the e'_2 that we needed to show the existence of. □

LEMMA 6.15 (CHECK COMPATIBILITY). *If $(k, v, v') \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $\tau' = K \sqcap \tau = K' \sqcap \tau$, then $(k, \text{assert } K v, \text{assert } K' v') \in \mathcal{E}^T \llbracket \tau' \rrbracket$.*

PROOF. Proceed by case analysis on $K \sqcap \tau$:

$K \sqcap \tau = \tau$ Then it must be the case that $K \propto v$ and $K' \propto v'$, meaning $\text{assert } K v \xrightarrow{T} v$ and $\text{assert } K' v' \xrightarrow{T} v'$, which is sufficient to complete the case.

$K \sqcap \tau = \text{Nat}$ **and** $\tau = \text{Int}$ Unfolding our hypothesis, we get that $v = v'$ and $v \in \mathbb{Z}$.

If $v \in \mathbb{N}$, then $\text{assert } K v \xrightarrow{T} v$ and $\text{assert } K' v' \xrightarrow{T} v'$, which is sufficient to complete the case.

Otherwise, $\text{assert } K v \xrightarrow{T} \text{TypeErr}(\text{Nat}, v)$ and $\text{assert } K' v' \xrightarrow{T} \text{TypeErr}(\text{Nat}, v')$, which is sufficient to complete the case.

$K \sqcap \tau = \perp$ Then $\text{assert } K v \xrightarrow{T} \text{TypeErr}(\text{Nat}, v)$ and $\text{assert } K v' \xrightarrow{T} \text{TypeErr}(\text{Nat}, v')$, which is sufficient to complete the case.

$K \sqcap \tau = K$ **and** $\tau \neq K$ Then $\tau = *$ and $K = K'$.

We can unfold our hypothesis to get that $(k - 1, v, v') \in \mathcal{V}^T \llbracket K'' \rrbracket$ for some K'' , which implies $v' \propto v$.

By the OS, either $\text{assert } K v \xrightarrow{T} v$ and $v \propto K$, or $\text{assert } K v \xrightarrow{T} \text{TypeErr}(K, v)$ and $\neg v \propto K$.

In either case, we have the corresponding property needed to complete the case.

3849 □

3850
3851 LEMMA 6.16 (DYNAMIC CHECKS ARE NO-OPS). *If $(k + 1, \text{assert } * v, \text{assert } * v') \in \mathcal{E}^T \llbracket \tau \rrbracket$, then $(k, v, v') \in \mathcal{E}^T \llbracket \tau \rrbracket$*

3852 PROOF. By the OS, $\text{assert } * v \longrightarrow v$ and $\text{assert } * v' \longrightarrow v'$.

3853 Then by our hypothesis, $(k, v, v') \in \mathcal{V}^T \llbracket \tau \rrbracket$, which is sufficient to complete the proof. □

3854 LEMMA 6.17 (SUBTYPING COMPATIBILITY). (1) *If $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$ and $\tau \leq \tau'$ then $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau' \rrbracket$*

3855 (2) *If $(k, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$ and $\tau \leq \tau'$ then $(k, e_1, e_2) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.*

3856

3857 PROOF. Proceed by mutual induction on k and τ :

3858

3859

3860

3861

3862

3863

3864

3865

3866

3867

3868

3869

3870

3871

3872

3873

3874

3875

3876

3877

3878

3879

3880

3881

3882

3883

3884

3885

3886

3887

3888

3889

3890

3891

3892

3893

3894

3895

3896

3897

3898

3899

3900

- $k = 0$: 2 is immediate if $e \neq v$.

If $e = v$ then 2 follows immediately from 1.

1 follows identically in the $k = 0$ case as it does in the $k > 0$ case, but the function case is vacuously true.

- $k > 0$:

(1) Case split on $\tau \leq \tau'$:

i) $\tau \leq \tau$: immediate.

ii) $\text{Nat} \leq \text{Int}$: immediate because $\mathbb{T} \subseteq \mathbb{Z}$.

iii) $\tau_1 \times \tau_2 \leq \tau'_1 \times \tau'_2$, with $\tau_1 \leq \tau'_1$ and $\tau_2 \leq \tau'_2$:

We want to show $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.

Unfolding our hypothesis, we get that $v_1 = \langle v'_1, v''_1 \rangle$ and similarly for v_2 .

We want to show $(k, v'_1, v'_2) \in \mathcal{V}^T \llbracket \tau'_1 \rrbracket$ and $(k, v''_1, v''_2) \in \mathcal{V}^T \llbracket \tau'_2 \rrbracket$.

We can apply IH 1) to both of judgements in our hypothesis to get $(k, v'_1, v'_2) \in \mathcal{V}^T \llbracket \tau'_1 \rrbracket$ and

$(k, v''_1, v''_2) \in \mathcal{V}^T \llbracket \tau'_2 \rrbracket$.

This is sufficient to show $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.

iv) $* \rightarrow \tau_2 \leq * \rightarrow \tau'_2$, with $\tau_2 \leq \tau'_2$:

We want to show $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau' \rrbracket$.

Let $j \leq k$ and $(j, v'_1, v'_2) \in \mathcal{V}^T \llbracket * \rrbracket$.

Let K .

We want to show $(j, \text{app}\{K\} v_1 v'_1, \text{app}\{K\} v_2 v'_2) \in \mathcal{E}^T \llbracket \tau'_2 \sqcap K \rrbracket$.

Then, we can apply our hypothesis about v_1, v_2 to get $(j, \text{app}\{K\} v_1 v'_1, \text{app}\{K\} v_2 v'_2) \in \mathcal{E}^T \llbracket \tau_2 \sqcap K \rrbracket$.

Finally, we can apply IH 1) to get $(j, \text{app}\{K\} v_1 v'_1, \text{app}\{K\} v_2 v'_2) \in \mathcal{E}^T \llbracket \tau'_2 \sqcap K \rrbracket$ which is what we

wanted to show.

(2) Unfolding our hypothesis, there is some $j \leq k$ and irreducible e'_1, e'_2 such that $e_1 \longrightarrow_T^j e'_1$ and $e_2 \longrightarrow_T^* e'_2$.

If $e'_1, e'_2 \in \text{Err}^\bullet$ then we're done.

Otherwise, $(k - j, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.

By IH 1), we have $(k - j, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau' \rrbracket$, which is what we wanted to show. □

LEMMA 6.18 (MONITOR COMPATIBILITY). *If $(k, v, v') \in \mathcal{V}^T \llbracket \tau \rrbracket$, then $(k + 1, \text{mon}\{K'_1 \Leftarrow K_1\}, \text{mon}\{K'_2 \Leftarrow K_2\} v') \in \mathcal{E}^T \llbracket \tau \rrbracket$.*

PROOF. By induction on k and v :

3901 $k = 0$ By case analysis on v, v' :

3902 i, i' By OS, $\text{mon} \{K'_1 \Leftarrow K_1\} i \longrightarrow i$ and $\text{mon} \{K'_2 \Leftarrow K_2\} i' \longrightarrow i'e$, so this is immediate.

3903 True, True As in case i above.

3904 False, False As in case True above.

3905 $\langle v_1, v_2 \rangle, \langle v'_1, v'_2 \rangle$ Since $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$, by inspection τ must be either $\tau_1 \times \tau_2$ or $*$:

3906 $\tau_1 \times \tau_2$ Note that $\text{mon} \{K'_1 \Leftarrow K_1\} \langle v_1, v_2 \rangle \longrightarrow \langle \text{mon} \{fst(K'_1) \Leftarrow fst(K_1)\} v_1, \text{mon} \{snd(K'_1) \Leftarrow snd(K_1)\} v_2 \rangle$,
3907 and similarly $\text{mon} \{K'_2 \Leftarrow K_2\} \langle v'_1, v'_2 \rangle \longrightarrow \langle \text{mon} \{fst(K'_2) \Leftarrow fst(K_2)\} v'_1, \text{mon} \{snd(K'_2) \Leftarrow snd(K_2)\} v'_2 \rangle$
3908 It is therefore sufficient to show that

3909 $(k, \langle \text{mon} \{fst(K'_1) \Leftarrow fst(K_1)\} v_1, \text{mon} \{snd(K'_1) \Leftarrow snd(K_1)\} v_2 \rangle, \langle \text{mon} \{fst(K'_2) \Leftarrow fst(K_2)\} v'_1, \text{mon} \{snd(K'_2) \Leftarrow snd(K_2)\} v'_2 \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$

3910

3911

3912 By unfolding, this is the same as showing $(k, \text{mon} \{fst(K'_1) \Leftarrow fst(K_1)\} v_1, \text{mon} \{fst(K'_2) \Leftarrow fst(K_2)\} v'_1) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$ and $(k, \text{mon} \{snd(K'_1) \Leftarrow snd(K_1)\} v_2, \text{mon} \{snd(K'_2) \Leftarrow snd(K_2)\} v'_2) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

3913 By Lemma 6.13, it suffices to show $(k+1, \text{mon} \{fst(K'_1) \Leftarrow fst(K_1)\} v_1, \text{mon} \{fst(K'_2) \Leftarrow fst(K_2)\} v'_1) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$ and $(k+1, \text{mon} \{snd(K'_1) \Leftarrow snd(K_1)\} v_2, \text{mon} \{snd(K'_2) \Leftarrow snd(K_2)\} v'_2) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

3914 In both cases, IH applies and hence it suffices to show $(k, v_1, v'_1) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$ and $(k, v_2, v'_2) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

3915 These are both obtained by unfolding our assumption.

3916 * Impossible, since $k = 0$.

3917 w, w' Since $(k, w, w') \in \mathcal{V}^T \llbracket \tau \rrbracket$, by inspection τ must be either $* \rightarrow \tau'$ or $*$:

3918 $* \rightarrow \tau'$ Note that $\text{mon} \{K'_1 \Leftarrow K_1\} w \longrightarrow \text{grd} \{K'_1 \Leftarrow K_1\} w$, and similarly $\text{mon} \{K'_2 \Leftarrow K_2\} w' \longrightarrow \text{grd} \{K'_2 \Leftarrow K_2\} w'$.

3919 Consequently, it is sufficient to show that $(k, \text{grd} \{K'_1 \Leftarrow K_1\} w, \text{grd} \{K'_2 \Leftarrow K_2\} w') \in \mathcal{E}^T \llbracket * \rightarrow \tau' \rrbracket$.

3920 Consider arbitrary $j \leq k, v, v'$ s.t. $(j, v, v') \in \mathcal{V}^T \llbracket * \rrbracket, K, K'$. Then we must show that

3921 $(j, \text{app}\{K\} (\text{grd} \{K'_1 \Leftarrow K_1\} w) v, \text{app}\{K'\} (\text{grd} \{K'_2 \Leftarrow K_2\} w') v') \in \mathcal{E}^T \llbracket K \sqcap \tau' \rrbracket$.

3922 By assumption, $k = 0$, so $j = 0$. Therefore, this is vacuously true.

3923 * Impossible, since $k = 0$.

3924 **otherwise** Impossible by Lemma 6.9.

3925 $k > 0$ By case analysis on v, v' :

3926 i, i' As in $k = 0$ case.

3927 True, True As in $k = 0$ case.

3928 False, False As in $k = 0$ case.

3929 $\langle v_1, v_2 \rangle, \langle v'_1, v'_2 \rangle$ Since $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$, by inspection τ must be either $\tau_1 \times \tau_2$ or $*$:

3930 $\tau_1 \times \tau_2$ As in $k = 0$ case.

3931 * By unfolding, $(k-1, w, w') \in \mathcal{V}^T \llbracket * \times * \rrbracket$. By an argument essentially identical to the previous case, merely reducing one application of monotonicity by one is sufficient to show what is needed.

3932 w, w' Since $(k, w, w') \in \mathcal{V}^T \llbracket \tau \rrbracket$, by inspection τ must be either $* \rightarrow \tau'$ or $*$:

3933 $* \rightarrow \tau'$ Note that $\text{mon} \{K'_1 \Leftarrow K_1\} w \longrightarrow \text{grd} \{K'_1 \Leftarrow K_1\} w$, and similarly $\text{mon} \{K'_2 \Leftarrow K_2\} w' \longrightarrow \text{grd} \{K'_2 \Leftarrow K_2\} w'$.

3934 Consequently, it is sufficient to show that $(k, \text{grd} \{K'_1 \Leftarrow K_1\} w, \text{grd} \{K'_2 \Leftarrow K_2\} w') \in \mathcal{E}^T \llbracket * \rightarrow \tau' \rrbracket$.

3935 Consider arbitrary $j \leq k, v, v'$ s.t. $(j, v, v') \in \mathcal{V}^T \llbracket * \rrbracket, K, K'$ s.t. $K \sqcap \tau' = K' \sqcap \tau'$. Then we must show that

3936 $(j, \text{app}\{K\} (\text{grd} \{K'_1 \Leftarrow K_1\} w) v, \text{app}\{K'\} (\text{grd} \{K'_2 \Leftarrow K_2\} w') v') \in \mathcal{E}^T \llbracket K \sqcap \tau' \rrbracket$.

3937 By OS, it suffices to show that

3938 $(j-1, \text{assert } K ((\text{grd} \{K'_1 \Leftarrow K_1\} w) v), \text{assert } K' ((\text{grd} \{K'_2 \Leftarrow K_2\} w') v')) \in \mathcal{E}^T \llbracket K \sqcap \tau' \rrbracket$.

3939

3953 By Lemma 6.15, it suffices to show that $(j - 1, (\text{grd} \{K'_1 \Leftarrow K_1\} w) v, (\text{grd} \{K'_2 \Leftarrow K_2\} w') v') \in$
3954 $\mathcal{E}^T \llbracket \tau' \rrbracket$.
3955 By OS, it suffices to show that
3956 $(j - 2, \text{mon} \{ \text{cod}(K'_1) \Leftarrow \text{cod}(K_1) \} w \text{ mon} \{ \text{dom}(K_1) \Leftarrow \text{dom}(K'_1) \} v,$
3957 $\text{mon} \{ \text{cod}(K'_2) \Leftarrow \text{cod}(K_2) \} w' \text{ mon} \{ \text{dom}(K_2) \Leftarrow \text{dom}(K'_2) \} v')$
3958 $\in \mathcal{E}^T \llbracket \tau' \rrbracket$.
3959 By IH, it suffices to show that $(j-3, w \text{ mon} \{ \text{dom}(K_1) \Leftarrow \text{dom}(K'_1) \} v, w' \text{ mon} \{ \text{dom}(K_2) \Leftarrow \text{dom}(K'_2) \} v') \in$
3960 $\mathcal{E}^T \llbracket \tau' \rrbracket$.
3961 By Lemma 6.16, it suffices to show that
3962 $(j - 2, \text{assert} * w \text{ mon} \{ \text{dom}(K_1) \Leftarrow \text{dom}(K'_1) \} v, \text{assert} * w' \text{ mon} \{ \text{dom}(K_2) \Leftarrow \text{dom}(K'_2) \} v') \in$
3963 $\mathcal{E}^T \llbracket \tau' \rrbracket$.
3964 By the definition of meet and OS, this is equivalent to
3965 $(j-1, \text{app}\{*\} w \text{ mon} \{ \text{dom}(K_1) \Leftarrow \text{dom}(K'_1) \} v, \text{app}\{*\} w' \text{ mon} \{ \text{dom}(K_2) \Leftarrow \text{dom}(K'_2) \} v') \in \mathcal{E}^T \llbracket * \sqcap$
3966 $\tau' \rrbracket$.
3967 By unfolding the assumption that $(k, w, w') \in \mathcal{E}^T \llbracket * \rightarrow \tau' \rrbracket$, it suffices to show that
3968 $(j - 1, \text{mon} \{ \text{dom}(K_1) \Leftarrow \text{dom}(K'_1) \} v, \text{mon} \{ \text{dom}(K_2) \Leftarrow \text{dom}(K'_2) \} v') \in \mathcal{E}^T \llbracket * \rrbracket$.
3969 By IH, it suffices to show that $(j - 2, v, v') \in \mathcal{E}^T \llbracket * \rrbracket$.
3970 By Lemma 6.13, it suffices to show that $(j, v, v') \in \mathcal{E}^T \llbracket * \rrbracket$.
3971 This is immediate from the assumption that $(j, v, v') \in \mathcal{V}^T \llbracket * \rrbracket$.
3972 * By unfolding, $(k - 1, w, w') \in \mathcal{V}^T \llbracket * \rightarrow * \rrbracket$. By an argument essentially identical to the previous case,
3973 merely reducing one application of monotonicity by one is sufficient to show what is needed.
3974 **otherwise** Impossible by Lemma 6.9.

□

3975 COROLLARY 6.19. *If $(k, e_1, e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$, then $(k + 1, \text{mon} \{K'_1 \Leftarrow K_1\}, \text{mon} \{K'_2 \Leftarrow K_2\} e_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$.*

3976 PROOF. Unfolding the expression relation in our hypothesis, we get that there is a j and e'_1 such that $e_1 \xrightarrow{j}_T e'_1$
3977 such that e'_1 is irreducible, and an e'_2 such that $e_2 \xrightarrow{*}_T e'_2$ and either they're errors, or $(k - j, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$.
3978 If they're errors, then we're done because the monitors will also step to errors.
3979 Otherwise, we have $\text{mon} \{K'_1 \Leftarrow K_1\} \xrightarrow{j}_T \text{mon} \{K'_1 \Leftarrow K_1\}$ and $\text{mon} \{K'_2 \Leftarrow K_2\} \xrightarrow{j}_T \text{mon} \{K'_2 \Leftarrow K_2\}$.
3980 By Lemma 6.18, we have that $(k - j, \text{mon} \{K'_1 \Leftarrow K_1\}, \text{mon} \{K'_2 \Leftarrow K_2\}) \in \mathcal{E}^T \llbracket \tau \rrbracket$, which is sufficient to complete the
3981 proof. □

3982 LEMMA 6.20 (BOUNDARY COMPATIBILITY). *If $(k, v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$ and $\tau' = K'_1 \sqcap K_1 \sqcap \tau = K'_2 \sqcap K_2 \sqcap \tau$, then $(k +$
3983 $1, \text{cast} \{K'_1 \Leftarrow K_1\} v_1, \text{cast} \{K'_2 \Leftarrow K_2\} v_2) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.*

3984 PROOF. By Lemma 6.10, notice that $\alpha^\perp (\lfloor \tau' \rfloor^\perp, v_1) \Leftrightarrow \alpha^\perp (\lfloor \tau' \rfloor^\perp, v_2)$. By Lemma 6.11 and our assumption, therefore,
3985 $\alpha^\perp (K'_1, v_1) \wedge \alpha^\perp (K_1, v_1) \wedge \alpha^\perp (\lfloor \tau \rfloor^\perp, v_1) \Leftrightarrow \alpha^\perp (K'_2, v_2) \wedge \alpha^\perp (K_2, v_2) \wedge \alpha^\perp (\lfloor \tau \rfloor^\perp, v_2)$. By Lemma 6.10, α^\perp
3986 $(\lfloor \tau \rfloor^\perp, v_1) \Leftrightarrow \alpha^\perp (\lfloor \tau \rfloor^\perp, v_2)$. Consequently, $\alpha^\perp (K'_1, v_1) \wedge \alpha^\perp (K_1, v_1) \Leftrightarrow \alpha^\perp (K'_2, v_2) \wedge \alpha^\perp (K_2, v_2)$ —which is to
3987 say, either both of the values match both of their annotated tags, or both of them do not match at least one of their
3988 annotated tags.

3989 Consider then each case:

3990 2023-04-10 15:45. Page 77 of 1–104.

4005 **Tags match** By the operational semantics, it is sufficient to show that $(k, \text{mon}\{K'_1 \Leftarrow K_1\} v_1, \text{mon}\{K'_2 \Leftarrow K_2\} v_2) \in$
 4006 $\mathcal{E}^T \llbracket \tau' \rrbracket$.

4007 By Lemma 6.18, it is sufficient to show that $(k - 1, v_1, v_2) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

4008 By Lemma 6.12, it is sufficient to show that $(k, v_1, v_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$, which is our assumption.

4009 **Tags do not match** Inspection of the operational semantics shows that both terms step to a boundary error, and so
 4010 are trivially in the relation.
 4011
 4012

4013 □

4014 **LEMMA 6.21 (BOUNDARY COMPATIBILITY—OPEN RELATION).** *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \tau \rrbracket_C^T$ and $\tau' = K'_1 \sqcap K_1 \sqcap \tau = K'_2 \sqcap K_2 \sqcap \tau$,*
 4015 *then $\llbracket \Gamma \vdash_{\text{tru}} \text{cast}\{K'_1 \Leftarrow K_1\} e_1 \leq \text{cast}\{K'_2 \Leftarrow K_2\} e_1 : \tau' \rrbracket$.*

4016 **PROOF.** Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

4017 We must show that $(k, \gamma(\text{cast}\{K'_1 \Leftarrow K_1\} e_1), \gamma'(\text{cast}\{K'_2 \Leftarrow K_2\} e_2)) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

4018 By the definition of substitution, it suffices to show that $(k, \text{cast}\{K'_1 \Leftarrow K_1\} \gamma(e_1), \text{cast}\{K'_2 \Leftarrow K_2\} \gamma'(e_2)) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

4019 Instantiate the hypothesis with (k, γ, γ') , providing that $(k, \gamma(e_1), \gamma'(e_2)) \in \mathcal{E}^T \llbracket \tau \rrbracket$.

4020 Then Lemma 6.14 applies. Consider arbitrary (k', v_1, v_2) s.t. $(k', v_1, v_2) \in \mathcal{V}^T \llbracket \tau \rrbracket$; we must show that $(k', \text{cast}\{K'_1 \Leftarrow$
 4021 $K_1\} v_1, \text{cast}\{K'_2 \Leftarrow K_2\} v_2) \in \mathcal{E}^T \llbracket \tau \rrbracket$. This is immediate by Lemma 6.20 and Lemma 6.13. □

4022

4023 **LEMMA 6.22 (APPLICATION COMPATIBILITY).** *If $(k, v_f, v'_f) \in \mathcal{V}^T \llbracket * \rightarrow \tau_2 \rrbracket$ and $(k, v_a, v'_a) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $\tau' = K \sqcap \tau_2 =$
 4024 $K' \sqcap \tau_2$, then $(k, \text{app}\{K\} v_f v_a, \text{app}\{K'\} v'_f v'_a) \in \mathcal{E}^T \llbracket \tau' \rrbracket$*

4025 **PROOF.** Unfolding the \mathcal{V} relation on our first assumption and instantiating with $j = k, v'_1 = v_a, v'_2 = v'_a, K = K,$
 4026 $K' = K'$ gives precisely what is to be shown. □

4027

4028 **LEMMA 6.23 (APPLICATION COMPATIBILITY—OPEN RELATION).** *If $\llbracket \Gamma \vdash_{\text{tru}} e_{f1} \leq e_{f2} : * \rightarrow \tau_2 \rrbracket_C^T$ and $\tau' = K_1 \sqcap \tau_2 = K_2 \sqcap \tau_2$
 4029 and $\llbracket \Gamma \vdash_{\text{tru}} e_{a1} \leq e_{a2} : \tau_1 \rrbracket_C^T$, then $\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K_1\} e_{f1} e_{a1} \leq \text{app}\{K_2\} e_{f2} e_{a2} : \tau' \rrbracket_C^T$.*

4030 **PROOF.** Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

4031 We must show that $(k, \gamma(\text{app}\{K_1\} e_{f1} e_{a1}), \gamma'(\text{app}\{K_2\} e_{f2} e_{a2})) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

4032 By the definition of substitution, it suffices to show that $(k, \text{app}\{K_1\} \gamma(e_{f1}) \gamma(e_{a1}), \text{app}\{K_2\} \gamma'(e_{f2}) \gamma'(e_{a2})) \in$
 4033 $\mathcal{E}^T \llbracket \tau' \rrbracket$.

4034 Instantiate the first hypothesis with (k, γ, γ') , providing $(k, \gamma(e_{f1}), \gamma'(e_{f2})) \in \mathcal{E}^T \llbracket * \rightarrow \tau_2 \rrbracket$. Similarly, the second
 4035 provides $(k, \gamma(e_{a1}), \gamma'(e_{a2})) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

4036 Then Lemma 6.14 applies. Consider arbitrary $(k', v_{f1}, v_{f2}) \in \mathcal{V}^T \llbracket * \rightarrow \tau_2 \rrbracket$ with $k' \leq k$. Then by Lemma 6.13,
 4037 $(k', \gamma(e_{a1}), \gamma'(e_{a2})) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$, Lemma 6.14 again applies. Consider arbitrary $(k'', v_{a1}, v_{a2}) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ with $k'' \leq k'$.
 4038 We must show that $(k'', \text{app}\{K_1\} v_{f1} v_{a1}, \text{app}\{K_2\} v_{f2} v_{a2}) \in \mathcal{E}^T \llbracket \tau' \rrbracket$; this is immediate by Lemma 6.22. □

4039

4040 **LEMMA 6.24 (APPLICATION COMPATIBILITY—FUNCTION IS BOTTOM).** *If $\llbracket \Gamma \vdash_{\text{tru}} e_{f1} \leq e_{f2} : \perp \rrbracket_C^T$ then $\llbracket \Gamma \vdash_{\text{tru}}$
 4041 $\text{app}\{K_1\} e_{f1} e_{a1} \leq \text{app}\{K_2\} e_{f2} e_{a2} : \perp \rrbracket_C^T$.*

4042 **PROOF.** Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

4043 We must show that $(k, \gamma(\text{app}\{K_1\} e_{f1} e_{a1}), \gamma'(\text{app}\{K_2\} e_{f2} e_{a2})) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

4044 By the definition of substitution, it suffices to show that $(k, \text{app}\{K_1\} \gamma(e_{f1}) \gamma(e_{a1}), \text{app}\{K_2\} \gamma'(e_{f2}) \gamma'(e_{a2})) \in$
 4045 $\mathcal{E}^T \llbracket \tau' \rrbracket$.

4046 Instantiate the first hypothesis with (k, γ, γ') , providing $(k, \gamma(e_{f1}), \gamma'(e_{f2})) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

4047

Then Lemma 6.14 applies. Consider arbitrary $(k', v_{f_1}, v_{f_2}) \in \mathcal{V}^T \llbracket \perp \rrbracket$ with $k' \leq k$. By unfolding of \mathcal{V} no such values can exist, so we are done. \square

LEMMA 6.25 (FST COMPATIBILITY). *If $(k, v, v') \in \mathcal{V}^T \llbracket \tau_1 \times \tau_2 \rrbracket$ and $\tau' = K \sqcap \tau_1 = K' \sqcap \tau_1$, then $(k, \text{fst}\{K\} v, \text{fst}\{K'\} v') \in \mathcal{E}^T \llbracket \tau' \rrbracket$.*

PROOF. Unfolding the definition of \mathcal{V} tells us that there must be some v_1, v_2, v'_1, v'_2 s.t. $v = \langle v_1, v_2 \rangle$, $v' = \langle v'_1, v'_2 \rangle$, $(k, v_1, v'_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$, and $(k, v_2, v'_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$. We must show that $(k, \text{fst}\{K\} \langle v_1, v_2 \rangle, \text{fst}\{K'\} \langle v'_1, v'_2 \rangle) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

By the OS, it suffices to show that $(k - 1, \text{assert } K v_1, \text{assert } K' v'_1) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

By Lemma 6.15, it suffices to show that $(k - 1, v_1, v'_1) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$. This is immediate by Lemma 6.13. \square

LEMMA 6.26 (FST COMPATIBILITY—OPEN RELATION). *If $\llbracket \Gamma \vdash_{\text{tru}} e \leq e' : \tau_1 \times \tau_2 \rrbracket_C^T$ and $\tau' = K \sqcap \tau_1 = K' \sqcap \tau_1$, then $\llbracket \Gamma \vdash_{\text{tru}} \text{fst}\{K\} e \leq \text{fst}\{K'\} e' : \tau' \rrbracket_C^T$.*

PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

We must show that $(k, \gamma(\text{fst}\{K\} e), \gamma'(\text{fst}\{K'\} e')) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

By the definition of substitution, it suffices to show that $(k, \text{fst}\{K\} \gamma(e), \text{fst}\{K'\} \gamma'(e')) \in \mathcal{E}^T \llbracket \tau' \rrbracket$.

Instantiate the hypothesis with (k, γ, γ') , providing $(k, \gamma(e), \gamma'(e')) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Then Lemma 6.14 applies. Consider arbitrary $(k', v, v') \in \mathcal{V}^T \llbracket \tau_1 \times \tau_2 \rrbracket$. We must show that $(k', \text{fst}\{K\} v, \text{fst}\{K'\} v') \in \mathcal{E}^T \llbracket \tau' \rrbracket$; this is immediate by Lemma 6.25. \square

LEMMA 6.27 (FST COMPATIBILITY—PAIR IS BOTTOM). *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \perp \rrbracket_C^T$ then $\llbracket \Gamma \vdash_{\text{tru}} \text{fst}\{K_1\} e_1 \leq \text{fst}\{K_2\} e_2 : \perp \rrbracket_C^T$.*

PROOF. By the same reasoning as Lemma 6.24. \square

LEMMA 6.28 (SND COMPATIBILITY).

PROOF. Nearly identical to that of Lemma 6.25. \square

LEMMA 6.29 (FST COMPATIBILITY—OPEN RELATION). *If $\llbracket \Gamma \vdash_{\text{tru}} e \leq e' : \tau_1 \times \tau_2 \rrbracket_C^T$ and $\tau' = K \sqcap \tau_2 = K' \sqcap \tau_2$, then $\llbracket \Gamma \vdash_{\text{tru}} \text{snd}\{K\} e \leq \text{snd}\{K'\} e' : \tau' \rrbracket_C^T$.*

PROOF. Nearly identical to that of Lemma 6.26, using Lemma 6.28. \square

LEMMA 6.30 (SND COMPATIBILITY—PAIR IS BOTTOM). *If $\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e_2 : \perp \rrbracket_C^T$ then $\llbracket \Gamma \vdash_{\text{tru}} \text{snd}\{K_1\} e_1 \leq \text{snd}\{K_2\} e_2 : \perp \rrbracket_C^T$.*

PROOF. By the same reasoning as Lemma 6.24. \square

6.4.3 Binary relation: Compatibility Lemmata

LEMMA 6.31 (T-VAR COMPATIBILITY).
$$\frac{(x : K) \in \Gamma}{\llbracket \Gamma \vdash_{\text{tru}} x \leq x : K \rrbracket_C^{\mathcal{L}}}$$

PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}} \llbracket \Gamma \rrbracket$.

We must show that $(k, \gamma(x), \gamma'(x)) \in \mathcal{E}^{\mathcal{L}} \llbracket K \rrbracket$.

Since $x : K \in \Gamma$, we know that there exist some values v, v' s.t. $\gamma(x) = v$ and $\gamma'(x) = v'$. Since $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}} \llbracket \Gamma \rrbracket$, we know that $(k, v, v') \in \mathcal{V}^{\mathcal{L}} \llbracket K \rrbracket$. Then we get $(k, v, v') \in \mathcal{E}^{\mathcal{L}} \llbracket \Gamma \rrbracket$ immediately since v, v' are already values. \square

4109 LEMMA 6.32 (T-NAT COMPATIBILITY). $\frac{}{\llbracket \Gamma \vdash_{\text{tru}} n \leq n : \text{Nat} \rrbracket_C^{\mathcal{L}}}$
 4110

4111 PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]$.

4112 We must show $(k, \gamma(n), \gamma'(n)) \in \mathcal{E}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket]$.

4113 Note that $\gamma(n) = n$.

4114 Since n is already a value, it suffices to show that $(k, n, n) \in \mathcal{V}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket]$.

4115 Unfolding the definition of $\mathcal{V}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket]$, this is true. □

4116
 4117
 4118 LEMMA 6.33 (T-INT COMPATIBILITY). $\frac{}{\llbracket \Gamma \vdash_{\text{tru}} i \leq i : \text{Int} \rrbracket_C^{\mathcal{L}}}$
 4119

4120 PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]$.

4121 We must show $(k, \gamma(i), \gamma'(i)) \in \mathcal{E}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket]$.

4122 Note that $\gamma(i) = i$.

4123 Since i is already a value, it suffices to show that $(k, i, i) \in \mathcal{V}^{\mathcal{L}}[\llbracket \text{Int} \rrbracket]$.

4124 Unfolding the definition of $\mathcal{V}^{\mathcal{L}}[\llbracket \text{Nat} \rrbracket]$, this is true. □

4125
 4126
 4127
 4128 LEMMA 6.34 (T-TRUE COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{True} \leq \text{True} : \text{Bool} \rrbracket_C^{\mathcal{L}}}$
 4129

4130 PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]$.

4131 We must show $(k, \gamma(\text{True}), \gamma'(\text{True})) \in \mathcal{E}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$.

4132 Note that $\gamma(\text{True}) = \text{True}$.

4133 Since True is already a value, it suffices to show that $(k, \text{True}, \text{True}) \in \mathcal{V}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$.

4134 Unfolding the definition of $\mathcal{V}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$, this is true. □

4135
 4136
 4137
 4138 LEMMA 6.35 (T-FALSE COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{False} \leq \text{False} : \text{Bool} \rrbracket_C^{\mathcal{L}}}$
 4139

4140 PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^{\mathcal{L}}[\llbracket \Gamma \rrbracket]$.

4141 We must show $(k, \gamma(\text{False}), \gamma'(\text{False})) \in \mathcal{E}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$.

4142 Note that $\gamma(\text{False}) = \text{False}$.

4143 Since False is already a value, it suffices to show that $(k, \text{False}, \text{False}) \in \mathcal{V}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$.

4144 Unfolding the definition of $\mathcal{V}^{\mathcal{L}}[\llbracket \text{Bool} \rrbracket]$, this is true. □

4145
 4146
 4147
 4148 LEMMA 6.36 (T-LAM COMPATIBILITY). $\frac{\llbracket \Gamma_0, (x_0 : K_0) \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_1 \rrbracket_C^{\mathcal{L}}}{\llbracket \Gamma_0 \vdash_{\text{tru}} \lambda(x_0 : K_0). e_0 \leq \lambda(x_0 : K_0). e'_0 : * \rightarrow \tau_1 \rrbracket_C^{\mathcal{L}}}$
 4149

4150 PROOF. Let $(k, \gamma, \gamma') \in \mathcal{G}^T[\llbracket \Gamma_0 \rrbracket]$.

4151 We want to show $(k, \gamma(\lambda x_0 : K_0. e_0), \gamma'(\lambda x_0. K_0 e'_0)) \in \mathcal{E}^T[\llbracket * \rightarrow \tau_1 \rrbracket]$.

4152 Note that $\gamma(\lambda x_0 : K_0. e_0) = \lambda x_0 : K_0. \gamma(e_0)$ and similarly for the other.

4153 We want to show $(k - 1, \lambda x_0 : K_0. \gamma(e_0), \lambda x_0 : K_0. \gamma(e'_0)) \in \mathcal{V}^T[\llbracket * \rightarrow \tau_1 \rrbracket]$.

4154 Unfolding the value relation:

4155 Let $j \leq k$.

4156 Let $(j, v, v') \in \mathcal{V}^T[\llbracket * \rrbracket]$.

4157 Let K .

4158
 4159
 4160

4161 We want to show $(j, \text{app}\{K\} (\lambda x_0 : K_0. \gamma(e_0)) v, \text{app}\{K\} (\lambda x_0 : K_0. \gamma(e'_0)) v') \in \mathcal{E}^T \llbracket \tau_1 \sqcap K \rrbracket$.

4162 By the OS, if $\neg K \propto v$ then the application steps to an error and we're done.

4163 Otherwise, $\text{app}\{K\} (\lambda x_0 : K_0. \gamma(e_0)) v \longrightarrow_T \text{assert } K ((\lambda x_0 : K_0. \gamma(e_0)) v) \longrightarrow \text{assert } K \gamma(e_0)[v/x]$.

4164 By the definition of substitution, $\gamma(e_0)[v/x] = \gamma[x \mapsto v](e_0)$.

4165 Note that $(j-2, \gamma[x \mapsto v](e_0), \gamma'[x \mapsto v](e'_0)) \in \mathcal{G}^T \llbracket \Gamma, x : K \rrbracket$ by Lemma 5.55 and Lemma 5.57.

4166 Therefore, we can apply the hypothesis to $\gamma[x \mapsto v]$, $\gamma'[x \mapsto v]$, and e_0, e'_0 at $j-2$ to get $(j-2, \gamma[x \mapsto v](e_0), \gamma'[x \mapsto v](e'_0)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.

4167 Finally, we can apply Lemma 5.58 to get $(j-1, \text{assert } K \gamma[x \mapsto v](e_0), \text{assert } K \gamma'[x \mapsto v](e'_0)) \in \mathcal{E}^T \llbracket \tau_1 \sqcap K \rrbracket$ which is what we wanted to show. \square

4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212

LEMMA 6.37 (T-PAIR COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e_1 \leq e'_1 : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_2 \leq e'_2 : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \langle e_1, e_2 \rangle \leq \langle e'_1, e'_2 \rangle : \tau_1 \times \tau_2 \rrbracket_C^T}$$

PROOF. Consider arbitrary $(k, \gamma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

We must show $(k, \gamma(\langle e_1, e_2 \rangle), \gamma'(\langle e'_1, e'_2 \rangle)) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Note that $\gamma(\langle e_1, e_2 \rangle) = \langle \gamma(e_1), \gamma(e_2) \rangle$, and similarly for γ', e'_1, e'_2 . We want to show that $(k, \langle \gamma(e_1), \gamma(e_2) \rangle, \langle \gamma'(e'_1), \gamma'(e'_2) \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Notice that by instantiating our hypothesis with (k, γ, γ') , we know that $(k, \gamma(e_1), \gamma'(e'_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$ and $(k, \gamma(e_2), \gamma'(e'_2)) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$.

By Lemma 6.14, it suffices to show that for any $(k', v_1, v'_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ where $k' \leq k$, $(k', \langle v_1, e_2 \rangle, \langle v'_1, e'_2 \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

By Lemma 6.13, we know that $(k', \gamma(e_2), \gamma'(e'_2)) \in \mathcal{E}^T \llbracket \tau_2 \rrbracket$. Again by Lemma 6.14, therefore, it suffices to show that for any $k'' \leq k'$ and v_2, v'_2 s.t. $(k'', v_2, v'_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$, $(k'', \langle v_1, v_2 \rangle, \langle v'_1, v'_2 \rangle) \in \mathcal{E}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Since these terms are values, it suffices to show that $(k'', \langle v_1, v_2 \rangle, \langle v'_1, v'_2 \rangle) \in \mathcal{V}^T \llbracket \tau_1 \times \tau_2 \rrbracket$.

Unfolding the definition of \mathcal{V} , it suffices to show that $(k'', v_1, v'_1) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$ and $(k'', v_2, v'_2) \in \mathcal{V}^T \llbracket \tau_2 \rrbracket$; both of these are immediate by Lemma 6.13 from our assumptions. \square

LEMMA 6.38 (T-CAST COMPATIBILITY).
$$\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{cast} \{K_1 \leftarrow K_0\} e_0 \leq \text{cast} \{K_1 \leftarrow K_0\} e'_0 : K_1 \sqcap K_0 \sqcap \tau_0 \rrbracket_C^T}$$

PROOF. Follows immediately from Lemma 6.21. \square

LEMMA 6.39 (T-APP COMPATIBILITY).
$$\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : * \rightarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \leq e'_1 : \tau'_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 \leq \text{app}\{K_1\} e'_0 e'_1 : K_1 \sqcap \tau_1 \rrbracket_C^T}$$

PROOF. Follows immediately from Lemma 6.23. \square

LEMMA 6.40 (T-APPBOT COMPATIBILITY).
$$\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \perp \rrbracket_C^T \quad \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \leq e'_1 : \tau'_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 \leq \text{app}\{K_1\} e'_0 e'_1 : \perp \rrbracket_C^T}$$

PROOF. Consider arbitrary $(k, \gamma, \gamma) \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

We must show $(k, \gamma(\text{app}\{K_1\} e_0 e_1), \gamma'(\text{app}\{K_1\} e'_0 e'_1)) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

Apply the first hypothesis to get $(k, \gamma(e_0), \gamma'(e'_0)) \in \mathcal{E}^T \llbracket \perp \rrbracket$.

4213 Unfolding, there exists some $j \leq k$, e_2, e_3 such that $\gamma(e_0) \rightarrow_T^j e_2$ and $\gamma'(e'_0) \rightarrow_T^j e_3$ where e_2 and e_3 are irreducible.
 4214 Either $e_2 = e_3 \in \text{Err}^\bullet$, or $(j, e_2, e_3) \in \mathcal{V}^T \llbracket \perp \rrbracket$.

4215 By inversion, it must be the case that $e_2 = e_3 \in \text{Err}^\bullet$, which means that by the OS, $\gamma(\text{app}\{K_1\} e_0 e_1 \rightarrow_T^{j+1} e_2$ and
 4216 $\gamma'(\text{app}\{K_1\} e'_0 e'_1) \rightarrow_T^{j+1} e_3$.

4218 Then either, $j + 1 > k$, in which case we're done, and otherwise both applications step to the same error within k
 4219 steps, in which case we're done. \square

4221 LEMMA 6.41 (T-FST COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 \leq \text{fst}\{K_0\} e'_0 : K_0 \sqcap \tau_0 \rrbracket_C^T}$

4224 PROOF. Consider arbitrary $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

4225 We must show $(k, \gamma(\text{fst}\{K_0\} e_0), \gamma'(\text{fst}\{K_1\} e'_0)) \in \mathcal{E}^T \llbracket K_0 \sqcap \tau_0 \rrbracket$.

4226 Note that $\gamma(\text{fst}\{K_0\} e_0) = \text{fst}\{K_0\} \gamma(e_0)$ and similarly for e'_0 .

4228 Assume that there are $j \leq k$, e_1 such that $\text{fst}\{K_0\} e_0 \rightarrow_T^j e_1$ and e_1 is irreducible.

4229 By the OS, it must be the case that there are irreducible e'_1, e''_1 such that $\text{fst}\{K_0\} e_0 \rightarrow^{j-2} \text{fst}\{K_0\} e'_1 \rightarrow \text{assert } K_0 e''_1 \rightarrow$
 4230 e_1 .

4231 Unfolding our hypothesis and applying it to the reduction $e_0 \rightarrow^{j-2} e'_1$, we get that there is an irreducible e'_2 such
 4232 that $e'_0 \rightarrow_T^* e'_2$ and $(k - j + 2, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau_0 \times \tau_1 \rrbracket$.

4233 Unfolding the value relation, we get that both e'_1 and e'_2 are pairs.

4234 Therefore, we have by the OS that there exists e''_2, e_2 such that $\text{fst}\{K_0\} e'_0 \rightarrow_T^* \text{fst}\{K_0\} e'_2 \rightarrow_T \text{assert } K_0 e''_2 \rightarrow_T e_2$.

4235 Unfolding the fact that $(k - j + 2, e'_1, e'_2) \in \mathcal{V}^T \llbracket \tau_0 \times \tau_1 \rrbracket$ gives us that $(k - j + 2, e''_1, e''_2) \in \hat{\mathcal{V}}^T \llbracket \tau_0 \rrbracket$.

4236 Finally, by Lemma 6.15, we get that $(k - j + 2, \text{assert } K_0 e''_1, \text{assert } K_0 e''_2) \in \mathcal{E}^T \llbracket \tau_0 \sqcap K_0 \rrbracket$, which is sufficient to
 4237 complete the proof. \square

4241 LEMMA 6.42 (T-FSTBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 \leq \text{fst}\{K_0\} e'_0 : \perp \rrbracket_C^T}$

4244 PROOF. Similar reasoning to T-APPBOT. \square

4246 LEMMA 6.43 (T-SND COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 \leq \text{snd}\{K_1\} e'_0 : K_1 \sqcap \tau_1 \rrbracket_C^T}$

4249 PROOF. Almost identical to T-FST. \square

4251 LEMMA 6.44 (T-SNDBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 \leq \text{snd}\{K_1\} e'_0 : \perp \rrbracket_C^T}$

4254 PROOF. Similar reasoning to T-APPBOT. \square

4256 LEMMA 6.45 (T-BINOP COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_0 \rrbracket_C^T \quad \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \leq e'_1 : \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{binop } e_0 e_1 \leq \text{binop } e'_0 e'_1 : \Delta(\text{binop}, \tau_0, \tau_1) \rrbracket_C^T}$

4260 PROOF. Let $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

4261 We want to show $(k, \gamma(\text{binop } e_0 e_1), \gamma(\text{binop } e'_0 e'_1)) \in \mathcal{E}^T \llbracket \Delta(\text{binop}, \tau_0, \tau_1) \rrbracket$.

4263 Note $\gamma(\text{binop } e_0 e_1) = \text{binop } \gamma(e_0) \gamma(e_1)$, and similarly for e'_0, e'_1 .

4264

4265 By the first hypothesis applied to γ, γ' we have $(k, \gamma(e_0), \gamma'(e'_0)) \in \mathcal{E}^T \llbracket \tau_0 \rrbracket$.
 4266 Unfolding we get there is a $j \leq k$, and irreducible e_2, e'_2 such that $\gamma(e_0) \xrightarrow{T^j} e_2$ and $\gamma'(e'_0) \xrightarrow{T^*} e'_2$.
 4267 If $e_2 = e'_2 = \text{Err}^\bullet$ then we're done, because the whole operation errors.
 4268 Otherwise $(k - j, e_2, e'_2) \in \mathcal{V}^T \llbracket \tau_0 \rrbracket$.

4270
 4271 Note by Lemma 6.13 $(k - j, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma_1 \rrbracket$.
 4272 By the second hypothesis applied to γ, γ' and $k - j$, we have $(k - j, \gamma(e_1), \gamma'(e'_1)) \in \mathcal{E}^T \llbracket \tau_1 \rrbracket$.
 4273 Unfolding we get there are j' , and irreducible e_3, e'_3 such that $\gamma(e_1) \xrightarrow{T^{j'}} e_3$ and $\gamma'(e'_1) \xrightarrow{T^*} e'_3$.
 4274 If $e_3 = e'_3 = \text{Err}^\bullet$ then we're done, because the whole operation errors.
 4275 Otherwise $(k - j - j', e_3, e'_3) \in \mathcal{V}^T \llbracket \tau_1 \rrbracket$.

4276 From the definition of Δ , $K_2 = \text{Int}$ or Nat or \perp .
 4277 In the case of \perp , we're done because either τ_0 or τ_1 is a \perp , which is a contradiction.
 4278 Otherwise, the cases proceed identically, so without loss of generality assume $K_2 = \text{Int}$.
 4279 $\tau_0 = \tau_1 = \text{Int}$, and therefore $e_2 = e'_2 = i_0$ and $e_3 = e'_3 = i_1$.
 4280 If $\text{binop} = \text{quotient}$ and $i_1 = 0$ then $\text{binop } i_0 \ i_1 \xrightarrow{T} \text{DivErr}$, so we're done.
 4281 If $\text{binop} = \text{quotient}$ and $i_1 \neq 0$, then $\text{binop } i_0 \ i_1 \xrightarrow{T} (i_0 / i_1)$.
 4282 Since $i_0 / i_1 \in \mathbb{Z}$, we're done.
 4283 If $\text{binop} = \text{sum}$ then $\text{binop } i_0 \ i_1 \xrightarrow{T} i_0 + i_1$.
 4284 Since $i_0 + i_1 \in \mathbb{Z}$, we're done. □

4285
 4286
 4287
 4288
 4289
 4290
 4291
 4292
 4293
 4294
 4295
 4296
 4297
 4298
 4299
 4300
 4301
 4302
 4303
 4304
 4305
 4306
 4307
 4308
 4309
 4310
 4311
 4312
 4313
 4314
 4315
 4316

$$\text{LEMMA 6.46 (T-IF COMPATIBILITY). } \frac{\begin{array}{l} \llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \text{Bool} \rrbracket_C^T \\ \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \leq e'_1 : \tau_0 \rrbracket_C^T \\ \llbracket \Gamma_0 \vdash_{\text{tru}} e_2 \leq e'_2 : \tau_1 \rrbracket_C^T \end{array}}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \leq \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2 : \tau_0 \sqcup \tau_1 \rrbracket_C^T}$$

PROOF. Let $(k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket$.

We want to show $(k, \gamma(\text{if } e_0 \text{ then } e_1 \text{ else } e_2), \gamma'(\text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2)) \in \mathcal{E}^T \llbracket \tau_0 \sqcup \tau_1 \rrbracket$.

Note $\gamma(\text{if } e_0 \text{ then } e_1 \text{ else } e_2) = \text{if } \gamma(e_0) \text{ then } \gamma(e_1) \text{ else } \gamma(e_2)$ and similarly for e'_0, e'_1, e_2 .

From the first hypothesis applied to γ, γ' , we know $(k, \gamma(e_0), \gamma'(e'_0)) \in \mathcal{E}^T \llbracket \text{Bool} \rrbracket$.

Unfolding, we have that there is a $j \leq k$ and irreducible e_4, e'_4 such that $e_0 \xrightarrow{T^j} e_4$ and $e'_0 \xrightarrow{T^*} e'_4$.

If $e_4, e'_4 \in \text{Err}^\bullet$ then we're done, because the entire if statement errors.

Otherwise, $(k - j, e_4, e'_4) \in \mathcal{V}^T \llbracket \text{Bool} \rrbracket$.

Unfolding the location and then the value relation, we get that $e_4 = e'_4 = \text{True}$ or $e_4 = e'_4 = \text{False}$.

- $e_4 = e'_4 = \text{True}$: Note by OS, if $\gamma(e_0)$ then $\gamma(e_1)$ else $\gamma(e_2) \xrightarrow{T^j}$ if e_4 then $\gamma(e_1)$ else $\gamma(e_2) \xrightarrow{T} \gamma(e_1)$, and similarly for if $\gamma'(e'_0)$ then $\gamma'(e'_1)$ else $\gamma'(e'_2)$.

By Lemma 6.13, we have $(k - j - 1, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma_1 \rrbracket$.

From the second hypothesis, we get $(k - j - 1, \gamma(e_1), \gamma'(e'_1)) \in \mathcal{E}^T \llbracket \tau_0 \rrbracket$.

Finally, by Lemma 5.61, we get $(k - j - 1, \gamma(e_1), \gamma'(e'_1)) \in \mathcal{E}^T \llbracket \tau_0 \sqcup \tau_1 \rrbracket$ which is sufficient to complete the proof.

- $e_4 = e'_4 = \text{False}$: same as other case except replace e_1 with e_2 .

4317

4318

4319

4320

4321

4322

4323

4324

4325

4326

4327

4328

4329

4330

4331

4332

4333

4334

4335

4336

4337

4338

4339

4340

4341

4342

4343

4344

4345

4346

4347

4348

4349

4350

4351

4352

4353

4354

4355

4356

4357

4358

4359

4360

4361

4362

4363

4364

4365

4366

4367

4368

□

$$\text{LEMMA 6.47 (T-IFBOT COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \perp \rrbracket_C^T \\ \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \leq e'_1 : \tau_0 \rrbracket_C^T \\ \llbracket \Gamma_0 \vdash_{\text{tru}} e_2 \leq e'_2 : \tau_1 \rrbracket_C^T \end{array}}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \leq \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2 : \perp \rrbracket_C^T}$$

PROOF. Similar reasoning to T-APPBOT. □

$$\text{LEMMA 6.48 (T-SUB COMPATIBILITY). } \frac{\begin{array}{c} \llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_0 \rrbracket_C^T \\ \tau_0 \leq \tau_1 \end{array}}{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \leq e'_0 : \tau_1 \rrbracket_C^T}$$

PROOF. Follows directly from Lemma 6.17. □

6.4.4 Binary relation: Fundamental Property

THEOREM 6.49 (BINARY RELATION IS REFLEXIVE). *If* $\Gamma \vdash_{\text{tru}} e : \tau$ *then* $\llbracket \Gamma \vdash_{\text{tru}} e \approx e : \tau \rrbracket_C^T$

PROOF. By induction over the typing derivation, using the compatibility lemmata. □

6.5 Context relation—Proofs

6.5.1 Context relation: Compatibility Lemmata

$$\text{LEMMA 6.50 (T-CTX-HOLE COMPATIBILITY). } \frac{\Gamma' \subseteq \Gamma}{\llbracket \Gamma \vdash_{\text{tru}} [] \approx [] : (\Gamma' \triangleright \tau) \rightsquigarrow \tau \rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket \Gamma' \vdash_{\text{tru}} e \approx e' : \tau \rrbracket$.

We want to show $\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau \rrbracket$.

Note $\forall (k, \gamma, \gamma') \in \mathcal{G}^T \llbracket \Gamma \rrbracket, (k, \gamma|_{\text{dom}(\Gamma')}, \gamma'|_{\text{dom}(\Gamma')}) \in \mathcal{G}^T \llbracket \Gamma' \rrbracket$.

And note $\gamma(e) = \gamma|_{\text{dom}(\Gamma')}(e)$ and similarly for e' .

Then given such k, γ, γ' , we can apply the hypothesis to get that $(k, \gamma(e), \gamma'(e')) \in \mathcal{E}^T \llbracket \tau \rrbracket$, which is sufficient to complete the proof. □

$$\text{LEMMA 6.51 (T-CTX-LAM COMPATIBILITY). } \frac{\llbracket \Gamma, (x:K) \vdash_{\text{tru}} E \approx E' : (\Gamma', (x:K) \triangleright \tau) \rightsquigarrow \tau' \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \lambda(x:K). E \approx \lambda(x:K). E' : (\Gamma', (x:K) \triangleright \tau) \rightsquigarrow * \rightarrow \tau' \rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket \Gamma', (x:K) \vdash_{\text{tru}} e \approx e' : \tau \rrbracket$.

We want to show $\llbracket \Gamma \vdash_{\text{tru}} \lambda(x:K). e \approx \lambda(x:K). e' : * \rightarrow \tau' \rrbracket$.

From our hypothesis we get $\llbracket \Gamma', (x:K) \vdash_{\text{tru}} E[e] \approx E[e'] : \tau' \rrbracket$.

Then the case follows from Lemma 6.36. □

$$\text{LEMMA 6.52 (T-CTX-PAIR-1 COMPATIBILITY). } \frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \langle E, e \rangle \approx \langle E', e' \rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2 \rrbracket_C^T}$$

PROOF. Let e, e' such that $\llbracket \Gamma' \vdash_{\text{tru}} e_1 \approx e'_1 : \tau \rrbracket$.

4369 We want to show $\llbracket \Gamma' \vdash_{\text{tru}} \langle E[e_1], e \rangle \approx \langle E'[e'_1], e \rangle : \tau_1 \times \tau_2 \rrbracket$.

4370 From our first hypothesis, we have $\llbracket \Gamma' \vdash_{\text{tru}} E[e_1] \approx E'[e'_1] : \tau_1 \rrbracket$.

4371 Then the case follows by Lemma 6.37. □

4372
4373
4374 LEMMA 6.53 (T-CTX-PAIR-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \langle e, E \rangle \approx \langle e', E' \rangle : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2 \rrbracket_C^T}$$

4375
4376 PROOF. Analogous to T-CTX-PAIR-1. □

4377
4378 LEMMA 6.54 (T-CTX-APP-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow * \rightarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K\} E e \approx \text{app}\{K\} E' e' : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1 \rrbracket_C^T}$$

4379
4380 PROOF. Let e, e' such that $\llbracket \Gamma' \vdash_{\text{tru}} e_1 \approx e'_1 : * \rightarrow \tau_1 \rrbracket$.

4381 We want to show $\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K\} E[e_1] e \approx \text{app}\{K\} E'[e'_1] e' : K \sqcap \tau_1 \rrbracket$.

4382 By the first hypothesis, we have $\llbracket \Gamma \vdash_{\text{tru}} E[e_1] \approx E'[e'_1] : * \rightarrow \tau_1 \rrbracket$.

4383 Then the case follows by Lemma 6.22. □

4384
4385 LEMMA 6.55 (T-CTX-APPBOT-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K\} E e \approx \text{app}\{K\} E' e' : (\Gamma' \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4386
4387 PROOF. Analogous to T-CTX-APP-1. □

4388
4389 LEMMA 6.56 (T-CTX-APP-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : * \rightarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K\} e E \approx \text{app}\{K\} e' E' : (\Gamma' \triangleright \tau) \rightsquigarrow K \sqcap \tau_1 \rrbracket_C^T}$$

4390
4391 PROOF. Analogous to T-CTX-APP-1. □

4392
4393 LEMMA 6.57 (T-CTX-APPBOT-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \perp \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma' \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{app}\{K\} e E \approx \text{app}\{K\} e' E' : (\Gamma' \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4394
4395 PROOF. Analogous to T-CTX-APP-1. □

4396
4397 LEMMA 6.58 (T-CTX-FST COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{fst}\{K\} E \approx \text{fst}\{K\} E' : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_1 \rrbracket_C^T}$$

4398
4399 PROOF. Let e, e' such that $\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_1 \times \tau_2 \rrbracket$.

4400 We want to show $\llbracket \Gamma \vdash_{\text{tru}} \text{fst}\{K\} E[e] \approx \text{fst}\{K\} E'[e'] : K \sqcap \tau_1 \rrbracket$.

4401 By the hypothesis, we get $\llbracket \Gamma \vdash_{\text{tru}} E[e] \approx E'[e'] : \tau_1 \times \tau_2 \rrbracket$.

4402 Then the case follows by Lemma 6.25. □

4403
4404 LEMMA 6.59 (T-CTX-FSTBOT COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{fst}\{K\} E \approx \text{fst}\{K\} E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4405
4406 PROOF. Analogous to T-CTX-FST. □

4407
4408 LEMMA 6.60 (T-CTX-SND COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \times \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{snd}\{K\} E \approx \text{snd}\{K\} E' : (\Gamma \triangleright \tau) \rightsquigarrow K \sqcap \tau_2 \rrbracket_C^T}$$

4409
4410 PROOF. Analogous to T-CTX-FST. □

4421

LEMMA 6.61 (T-CTX-SNDBOT COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{snd}\{K\} E \approx \text{snd}\{K\} E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4423

PROOF. Analogous to T-CTX-FST. □

4425

LEMMA 6.62 (T-CTX-BINOP-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{binop} E e \approx \text{binop} E' e' : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\text{binop}, \tau_1, \tau_2) \rrbracket_C^T}$$

4427

4428

PROOF. Let e_1, e'_1 such that $\llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e'_1 : \tau \rrbracket$.

4429

4430

We want to show $\llbracket \Gamma \vdash_{\text{tru}} \text{binop} E[e_1] e \approx \text{binop} E'[e'_1] e' : \Delta(\text{binop}, \tau_1, \tau_2) \rrbracket$.

4431

By the first hypothesis, $\llbracket \Gamma \vdash_{\text{tru}} E[e_1] \approx E'[e'_1] : \tau_1 \rrbracket$.

4432

Then the case follows by Lemma 6.45. □

4433

4434

4435

LEMMA 6.63 (T-CTX-BINOP-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{binop} E e \approx \text{binop} E' e' : (\Gamma \triangleright \tau) \rightsquigarrow \Delta(\text{binop}, \tau_1, \tau_2) \rrbracket_C^T}$$

4436

4437

4438

PROOF. Analogous to T-CTX-BINOP-1. □

4439

4440

LEMMA 6.64 (T-CTX-BND-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau' \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{cast}\{K_2 \leftarrow K_1\} E \approx \text{cast}\{K_2 \leftarrow K_1\} E' : (\Gamma \triangleright \tau) \rightsquigarrow K_2 \sqcap K_1 \sqcap \tau' \rrbracket_C^T}$$

4441

4442

4443

PROOF. □

4444

4445

LEMMA 6.65 (T-CTX-IF-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \text{Bool} \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } E \text{ then } e_1 \text{ else } e_2 \approx \text{if } E' \text{ then } e'_1 \text{ else } e'_2 : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket_C^T}$$

4446

4447

4448

PROOF. Let e_0, e'_0 such that $\llbracket \Gamma \vdash_{\text{tru}} e_0 \approx e'_0 : \tau \rrbracket$.

4449

We want to show $\llbracket \Gamma \vdash_{\text{tru}} \text{if } E[e_0] \text{ then } e_1 \text{ else } e_2 \approx \text{if } E'[e'_0] \text{ then } e'_1 \text{ else } e'_2 : \tau_1 \sqcup \tau_2 \rrbracket$.

4450

By the first hypothesis, $\llbracket \Gamma \vdash_{\text{tru}} E[e_0] \approx E'[e'_0] : \text{Bool} \rrbracket$.

4451

The case follows by Lemma 6.46. □

4452

4453

4454

LEMMA 6.66 (T-CTX-IFBOT-1 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } E \text{ then } e_1 \text{ else } e_2 \approx \text{if } E' \text{ then } e'_1 \text{ else } e'_2 : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4455

4456

4457

PROOF. Analogous to T-CTX-IF-1 □

4458

4459

LEMMA 6.67 (T-CTX-IF-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e_b \approx e'_b : \text{Bool} \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } E \text{ else } e_2 \approx \text{if } e'_b \text{ then } E' \text{ else } e'_2 : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket_C^T}$$

4460

4461

4462

PROOF. Analogous to T-CTX-IF-1 □

4463

4464

LEMMA 6.68 (T-CTX-IFBOT-2 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e_b \approx e'_b : \perp \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } E \text{ else } e_2 \approx \text{if } e'_b \text{ then } E' \text{ else } e'_2 : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4465

4466

4467

PROOF. Analogous to T-CTX-IF-1 □

4468

4469

LEMMA 6.69 (T-CTX-IF-3 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e_b \approx e'_b : \text{Bool} \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } e_1 \text{ else } E \approx \text{if } e'_b \text{ then } e'_1 \text{ else } E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_1 \sqcup \tau_2 \rrbracket_C^T}$$

4470

4471

4472

4473 PROOF. Analogous to T-CTX-IF-1

□

4474
4475
4476
4477
4478
4479
4480
4481
4482
4483 LEMMA 6.70 (T-CTX-IFBOT-3 COMPATIBILITY).
$$\frac{\llbracket \Gamma \vdash_{\text{tru}} e_b \approx e'_b : \perp \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T \quad \llbracket \Gamma \vdash_{\text{tru}} E \approx E' : (\Gamma \triangleright \tau) \rightsquigarrow \tau_2 \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } e_1 \text{ else } E \approx \text{if } e'_b \text{ then } e'_1 \text{ else } E' : (\Gamma \triangleright \tau) \rightsquigarrow \perp \rrbracket_C^T}$$

4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494 PROOF. Analogous to T-CTX-IF-1

□

4496 6.5.2 Context relation: Fundamental Property

4497 THEOREM 6.71 (CONTEXT RELATION IS REFLEXIVE). *If $\Gamma \vdash_{\text{tru}} C : (\Gamma' \triangleright \tau) \rightsquigarrow \tau'$, then $\llbracket \Gamma \vdash_{\text{tru}} C \approx C : (\Gamma' \vdash_{\text{tru}} \tau) \rightsquigarrow \tau' \rrbracket$.*

4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508 PROOF. By induction over the typing derivation, using the compatibility lemmata.

□

4516 6.6 Check optimization

4517
4518
4519
$$K \setminus \tau = \begin{cases} * & \text{if } \tau \leq K \\ K & \text{otherwise} \end{cases}$$

4525 $\boxed{\Gamma \vdash_{\text{tru}} e : \tau \rightsquigarrow e}$ optimization

4526

4527

4528

4529

4530

4531

4532

4533

4534

4535

4536

4537

4538

4539

4540

4541

4542

4543

4544

4545

4546

4547

4548

4549

4550

4551

4552

4553

4554

4555

4556

4557

4558

4559

4560

4561

4562

4563

4564

4565

4566

4567

4568

4569

4570

4571

4572

4573

4574

4575

4576

$\frac{\text{T-VAR}}{\Gamma \vdash_{\text{tru}} x_0 : K_0 \rightsquigarrow x_0}$	$\frac{\text{T-NAT}}{\Gamma \vdash_{\text{tru}} n_0 : \text{Nat} \rightsquigarrow n_0}$	$\frac{\text{T-INT}}{\Gamma \vdash_{\text{tru}} i_0 : \text{Int} \rightsquigarrow i_0}$	$\frac{\text{T-TRUE}}{\Gamma \vdash_{\text{tru}} \text{True} : \text{Bool} \rightsquigarrow \text{True}}$
$\frac{\text{T-FALSE}}{\Gamma \vdash_{\text{tru}} \text{False} : \text{Bool} \rightsquigarrow \text{False}}$	$\frac{\text{T-LAM}}{\Gamma \vdash_{\text{tru}} \lambda(x_0 : K_0). e_0 : * \rightarrow \tau_1 \rightsquigarrow \lambda(x_0 : K_0). e'_0}$	$\frac{\text{T-PAIR}}{\Gamma \vdash_{\text{tru}} \langle e_0, e_1 \rangle : \tau_0 \times \tau_1 \rightsquigarrow \langle e'_0, e'_1 \rangle}$	
$\frac{\text{T-CAST}}{\Gamma \vdash_{\text{tru}} \text{cast} \{K_1 \Leftarrow K_0\} e_0 : K_1 \sqcap K_0 \sqcap \tau_0 \rightsquigarrow \text{cast} \{K_1 \setminus (K_0 \sqcap \tau_0) \Leftarrow K_0 \setminus \tau_0\} e'_0}$			
$\frac{\text{T-APP}}{\Gamma \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 : K_1 \sqcap \tau_1 \rightsquigarrow \text{app}\{K_1 \setminus \tau_1\} e'_0 e'_1}$	$\frac{\text{T-APPBOT}}{\Gamma \vdash_{\text{tru}} \text{app}\{K_1\} e_0 e_1 : \perp \rightsquigarrow \text{app}\{K_1 \setminus \perp\} e'_0 e'_1}$		
$\frac{\text{T-FST}}{\Gamma \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 : K_0 \sqcap \tau_0 \rightsquigarrow \text{app}\{K_0 \setminus \tau_0\} e'_0}$	$\frac{\text{T-FSTBOT}}{\Gamma \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 : \perp \rightsquigarrow \text{fst}\{K_0 \setminus \perp\} e'_0}$		
$\frac{\text{T-SND}}{\Gamma \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 : K_1 \sqcap \tau_1 \rightsquigarrow \text{snd}\{K_1 \setminus \tau_1\} e'_0}$	$\frac{\text{T-SNDBOT}}{\Gamma \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 : \perp \rightsquigarrow \text{snd}\{K_1 \setminus \perp\} e'_0}$		
$\frac{\text{T-BINOP}}{\Gamma \vdash_{\text{tru}} \text{binop} e_0 e_1 : \Delta(\text{binop}, \tau_0, \tau_1) \rightsquigarrow \text{binop} e'_0 e'_1}$		$\frac{\text{T-IF}}{\Gamma \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \tau_0 \sqcup \tau_1 \rightsquigarrow \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2}$	
$\frac{\text{T-IFBOT}}{\Gamma \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 : \perp \rightsquigarrow \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2}$			
$\frac{\text{T-SUB}}{\Gamma \vdash_{\text{tru}} e_0 : \tau_0 \rightsquigarrow e'_0}$			$\frac{\tau_0 \leq \tau_1}{\Gamma \vdash_{\text{tru}} e_0 : \tau_1 \rightsquigarrow e'_0}$

4571 **THEOREM 6.72 (CHECK-ELISION CORRECTNESS).** *If $\Gamma \vdash_{\text{tru}} e : \tau \rightsquigarrow e'$, then $\Gamma \vdash_{\text{tru}} e \approx^{\text{ctx}} e' : \tau$.*

4572 **PROOF.** Consider arbitrary Γ, e, τ, e' s.t. $\Gamma \vdash_{\text{tru}} e : \tau \rightsquigarrow e'$. By Lemma 6.92, $\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau \rrbracket_C^T$. By Theorem 6.3,
 4573 $\Gamma \vdash_{\text{tru}} e \approx^{\text{ctx}} e' : \tau$, which is what was to be shown. \square

6.7 Check-elision—Proofs

LEMMA 6.73 ($K \setminus \tau$ PRESERVES MEETS). $K \sqcap \tau = (K \setminus \tau) \sqcap \tau$.

PROOF. Immediate by unfolding and lattice properties. \square

6.7.1 Check-elision: Compatibility Lemmata

LEMMA 6.74 (T-VAR COMPATIBILITY). $\frac{(x_0 : K_0) \in \Gamma_0}{\llbracket \Gamma_0 \vdash_{\text{tru}} x_0 \approx x_0 : K_0 \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.31. \square

LEMMA 6.75 (T-NAT COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} n_0 \approx n_0 : \text{Nat} \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.32. \square

LEMMA 6.76 (T-INT COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} i_0 \approx i_0 : \text{Int} \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.32. \square

LEMMA 6.77 (T-TRUE COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{True} \approx \text{True} : \text{Bool} \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.34. \square

LEMMA 6.78 (T-FALSE COMPATIBILITY). $\frac{}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{False} \approx \text{False} : \text{Bool} \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.35. \square

LEMMA 6.79 (T-LAM COMPATIBILITY). $\frac{\llbracket \Gamma_0, (x_0 : K_0) \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \lambda(x_0 : K_0). e_0 \approx \lambda(x_0 : K_0). e'_0 : * \rightarrow \tau_1 \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.36. \square

LEMMA 6.80 (T-PAIR COMPATIBILITY). $\frac{\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \rrbracket_C^T \quad \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \langle e_0, e_1 \rangle \approx \langle e'_0, e'_1 \rangle : \tau_0 \times \tau_1 \rrbracket_C^T}}{\llbracket \Gamma_0 \vdash_{\text{tru}} \langle e_0, e_1 \rangle \approx \langle e'_0, e'_1 \rangle : \tau_0 \times \tau_1 \rrbracket_C^T}$

PROOF. By unfolding and Lemma 6.37. \square

LEMMA 6.81 (T-CAST COMPATIBILITY). $\frac{\llbracket \Gamma \vdash_{\text{tru}} e_1 \approx e_2 : \tau \rrbracket_C^T}{\llbracket \Gamma \vdash_{\text{tru}} \text{cast} \{K' \leftarrow K\} e_1 \approx \text{cast} \{K' \setminus (K \sqcap \tau) \leftarrow K \setminus \tau\} e_2 : K' \sqcap K \sqcap \tau \rrbracket_C^T}$

PROOF. Follows immediately from lattice properties and Lemma 6.21. \square

LEMMA 6.82 (T-APP COMPATIBILITY). $\frac{\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : * \rightarrow \tau_1 \rrbracket_C^T \quad \llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau'_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{app} \{K_1\} e_0 e_1 \approx \text{app} \{K_1 \setminus \tau_1\} e'_0 e'_1 : K_1 \sqcap \tau_1 \rrbracket_C^T}}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{app} \{K_1\} e_0 e_1 \approx \text{app} \{K_1 \setminus \tau_1\} e'_0 e'_1 : K_1 \sqcap \tau_1 \rrbracket_C^T}$

4629 PROOF. Follows immediately from lattice properties and Lemma 6.23. □

4630

4631

$$\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau'_0 \rrbracket_C^T}$$

4632 LEMMA 6.83 (T-APPBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{app}\{K_1\} e_0 \approx \text{app}\{K_1 \setminus \perp\} e'_0 e'_1 : \perp \rrbracket_C^T}$

4633

4634

4635 PROOF. Follows immediately from Lemma 6.24. □

4636

4637

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T$$

4638 LEMMA 6.84 (T-FST COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 \approx \text{fst}\{K_0 \setminus \tau_0\} e'_0 : K_0 \sqcap \tau_0 \rrbracket_C^T}$

4639

4640

4641 PROOF. Follows immediately from lattice properties and Lemma 6.26. □

4642

4643

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T$$

4644 LEMMA 6.85 (T-FSTBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{fst}\{K_0\} e_0 \approx \text{fst}\{K_0 \setminus \perp\} e'_0 : \perp \rrbracket_C^T}$

4645

4646

4647 PROOF. Follows immediately from Lemma 6.27. □

4648

4649

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T$$

4650 LEMMA 6.86 (T-SND COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \times \tau_1 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 \approx \text{snd}\{K_1 \setminus \tau_1\} e'_0 : K_1 \sqcap \tau_1 \rrbracket_C^T}$

4651

4652

4653 PROOF. Follows immediately from lattice properties and Lemma 6.29. □

4654

4655

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T$$

4656 LEMMA 6.87 (T-SNDBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{snd}\{K_1\} e_0 \approx \text{snd}\{K_1 \setminus \perp\} e'_0 : \perp \rrbracket_C^T}$

4657

4658

4659 PROOF. Follows immediately from Lemma 6.30. □

4660

4661

$$\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_1 \rrbracket_C^T}$$

4662 LEMMA 6.88 (T-BINOP COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{binop} e_0 e_1 \approx \text{binop} e'_0 e'_1 : \Delta(\text{binop}, \tau_0, \tau_1) \rrbracket_C^T}$

4663

4664

4665 PROOF. By unfolding and Lemma 6.45. □

4666

4667

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \text{Bool} \rrbracket_C^T$$

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_0 \rrbracket_C^T$$

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_1 \rrbracket_C^T$$

4668 LEMMA 6.89 (T-IF COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \text{Bool} \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \approx \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2 : \tau_0 \sqcup \tau_1 \rrbracket_C^T}$

4669

4670

4671 PROOF. By unfolding and Lemma 6.46. □

4672

4673

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T$$

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_1 \approx e'_1 : \tau_0 \rrbracket_C^T$$

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_2 \approx e'_2 : \tau_1 \rrbracket_C^T$$

4674 LEMMA 6.90 (T-IFBOT COMPATIBILITY). $\frac{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \perp \rrbracket_C^T}{\llbracket \Gamma_0 \vdash_{\text{tru}} \text{if } e_0 \text{ then } e_1 \text{ else } e_2 \approx \text{if } e'_0 \text{ then } e'_1 \text{ else } e'_2 : \perp \rrbracket_C^T}$

4675

4676

4677 PROOF. By unfolding and Lemma 6.47. □

4678

4679

$$\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_0 \rrbracket_C^T$$

4680 LEMMA 6.91 (T-SUB COMPATIBILITY). $\frac{\tau_0 \leq \tau_1}{\llbracket \Gamma_0 \vdash_{\text{tru}} e_0 \approx e'_0 : \tau_1 \rrbracket_C^T}$

4681

4682

4683 PROOF. By unfolding and Lemma 6.48. □

4681 6.7.2 Check-elision: Fundamental Property

4682

4683 THEOREM 6.92 (CHECK-ELISION IS CORRECT FOR BINARY LR). *If* $\Gamma \vdash_{\text{tru}} e : \tau \rightsquigarrow e'$, *then* $\llbracket \Gamma \vdash_{\text{tru}} e \approx e' : \tau \rrbracket_C^T$.

4684

4685 PROOF. By induction over the check-elision judgment derivation, using the compatibility lemmata. □

4686

4687

4688

4689

4690

4691

4692

4693

4694

4695

4696

4697

4698

4699

4700

4701

4702

4703

4704

4705

4706

4707

4708

4709

4710

4711

4712

4713

4714

4715

4716

4717

4718

4719

4720

4721

4722

4723

4724

4725

4726

4727

4728

4729

4730

4731

4732

7 GTL

Surface language

4733
4734
4735
4736 $t ::= x \mid n \mid i \mid \text{True} \mid \text{False} \mid \lambda(x:K) \rightarrow \tau. t \mid \langle t, t \rangle \mid t t \mid \text{fst } t \mid \text{snd } t \mid \text{binop } t t \mid \text{if } t \text{ then } t \text{ else } t$
4737
4738 $\tau ::= \text{Nat} \mid \text{Int} \mid \text{Bool} \mid \tau \times \tau \mid * \rightarrow \tau \mid *$
4739 $\text{binop} ::= \text{sum} \mid \text{quotient}$
4740 $\Gamma ::= \cdot \mid \Gamma, (x:\tau)$
4741 $n ::= \mathbb{N}$
4742
4743 $i ::= \mathbb{Z}$
4744 $\Delta^{-1}(\text{binop}, \tau) = \begin{cases} \text{Int, Int} & \text{if } \tau = \text{Int} \\ \text{Nat, Nat} & \text{if } \tau = \text{Nat} \end{cases}$
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784

7.1 Simple Translation

$$\boxed{[\tau \swarrow \tau']e}$$

$$[\tau \swarrow \tau']e = \begin{cases} e & \text{if } \tau \geq \tau' \\ \text{cast } \{\tau \leftarrow \tau'\} e & \text{if } \tau \not\geq \tau' \wedge \tau \sim \tau' \end{cases}$$

$$\boxed{\tau \sim \tau'}$$

$$\frac{}{\tau \sim *} \quad \frac{}{\text{Nat} \sim \text{Int}} \quad \frac{\tau_0 \sim \tau_2 \quad \tau_1 \sim \tau_3}{\tau_0 \times \tau_1 \sim \tau_2 \times \tau_3} \quad \frac{\tau_0 \sim \tau_2 \quad \tau_1 \sim \tau_3}{\tau_0 \rightarrow \tau_1 \sim \tau_2 \rightarrow \tau_3} \quad \frac{}{\tau \sim \tau} \quad \frac{\tau \sim \tau'}{\tau' \sim \tau}$$

$$\boxed{\tilde{\tau}, \tilde{\tau}': \tau \times \tau \rightarrow \tau}$$

$$\text{Nat } \tilde{\tau} \text{ Int} = \text{Int}$$

$$\tau_0 \rightarrow \tau_1 \tilde{\tau} \tau_2 \rightarrow \tau_3 = \tau_0 \tilde{\tau} \tau_2 \rightarrow \tau_1 \tilde{\tau} \tau_3$$

$$\tau_0 \times \tau_1 \tilde{\tau} \tau_2 \times \tau_3 = \tau_0 \tilde{\tau} \tau_2 \times \tau_1 \tilde{\tau} \tau_3$$

$$\tau \tilde{\tau} * = \tau$$

$$\tau \tilde{\tau} \tau' = \tau' \tilde{\tau} \tau$$

$$\tau \tilde{\tau} \tau = \tau$$

$$\tau \tilde{\tau} \tau' \text{ undefined otherwise}$$

$$\text{Nat } \tilde{\tau} \text{ Int} = \text{Nat}$$

$$\tau_0 \rightarrow \tau_1 \tilde{\tau} \tau_2 \rightarrow \tau_3 = \tau_0 \tilde{\tau} \tau_2 \rightarrow \tau_1 \tilde{\tau} \tau_3$$

$$\tau_0 \times \tau_1 \tilde{\tau} \tau_2 \times \tau_3 = \tau_0 \tilde{\tau} \tau_2 \times \tau_1 \tilde{\tau} \tau_3$$

$$\tau \tilde{\tau} * = \tau$$

$$\tau \tilde{\tau} \tau' = \tau' \tilde{\tau} \tau$$

$$\tau \tilde{\tau} \tau = \tau$$

$$\tau \tilde{\tau} \tau' \text{ undefined otherwise}$$

4837	$\Gamma \vdash_{\text{sim}} t : \tau \rightsquigarrow e$
4838	
4839	
4840	$\frac{(x:\tau) \in \Gamma}{\Gamma \vdash_{\text{sim}} x : \tau \rightsquigarrow x}$
4841	$\frac{}{\Gamma \vdash_{\text{sim}} n : \text{Nat} \rightsquigarrow n}$
4842	$\frac{}{\Gamma \vdash_{\text{sim}} i : \text{Int} \rightsquigarrow i}$
4843	$\frac{\Gamma, (x:\tau) \vdash_{\text{sim}} t : \tau'' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \lambda(x:\tau) \rightarrow \tau'. t : \tau \rightarrow \tau' \rightsquigarrow \lambda(x:\tau). ([\tau' \swarrow \tau'']e)}$
4844	$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \langle t_1, t_2 \rangle : \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle}$
4845	
4846	
4847	$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau \rightarrow \tau' \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau'' \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} t_1 t_2 : \tau' \rightsquigarrow \text{app}\{\tau'\} e_1 ([\tau \swarrow \tau'']e_2)}$
4848	$\frac{\Gamma \vdash_{\text{sim}} t_1 : * \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau'$
4849	$\frac{\Gamma \vdash_{\text{sim}} t_1 t_2 : * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow* \leftarrow *\} e_1) [* \swarrow \tau']e_2}{\Gamma \vdash_{\text{sim}} t_1 t_2 : * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow* \leftarrow *\} e_1) [* \swarrow \tau']e_2}$
4850	
4851	$\frac{\Gamma \vdash_{\text{sim}} t : \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{fst } t : \tau \rightsquigarrow \text{fst}\{\tau\} e}$
4852	$\frac{\Gamma \vdash_{\text{sim}} t : * \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{fst } t : * \rightsquigarrow \text{fst}\{*\} (\text{cast}\{*\times* \leftarrow *\} e)}$
4853	$\frac{\Gamma \vdash_{\text{sim}} t : \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{snd } t : \tau' \rightsquigarrow \text{snd}\{\tau'\} e}$
4854	
4855	$\frac{\Gamma \vdash_{\text{sim}} t : * \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{snd } t : * \rightsquigarrow \text{snd}\{*\} (\text{cast}\{*\times* \leftarrow *\} e)}$
4856	
4857	
4858	$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2 \quad \Delta(\text{binop}, \tau_1 \sqsupset \tau_2, \tau_1 \sqsupset \tau_2) = \tau' \quad \tau_1 \leq \text{Int} \wedge \tau_2 \leq \text{Int}}{\Gamma \vdash_{\text{sim}} \text{binop } t_1 t_2 : \tau' \rightsquigarrow \text{binop } e_1 e_2}$
4859	
4860	
4861	
4862	$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \text{binop } t_1 t_2 : \tau' \rightsquigarrow \text{binop}([\text{Int} \swarrow \tau_1]e_1) ([\text{Int} \swarrow \tau_2]e_2)}$
4863	
4864	
4865	
4866	$\frac{\Gamma \vdash_{\text{sim}} t_b : \text{Bool} \rightsquigarrow e_b \quad \Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \text{if } t_b \text{ then } t_1 \text{ else } t_2 : \tau_1 \sqsupset \tau_2 \rightsquigarrow \text{if } e_b \text{ then } ([\tau_1 \sqsupset \tau_2 \swarrow \tau_1]e_1) \text{ else } ([\tau_1 \sqsupset \tau_2 \swarrow \tau_2]e_2)}$
4867	
4868	
4869	
4870	
4871	
4872	
4873	
4874	
4875	
4876	
4877	
4878	
4879	
4880	
4881	
4882	
4883	
4884	
4885	
4886	
4887	
4888	

THEOREM 7.1 (TYPED TRANSLATION IMPLIES SIMPLE TYPING).

If $\Gamma \vdash_{\text{sim}} t : \tau \rightsquigarrow e$ then $\Gamma \vdash_{\text{sim}} e : \tau$.

PROOF. Proceed by induction on the typed translation.

$$\frac{(x:\tau) \in \Gamma}{\Gamma \vdash_{\text{sim}} x : \tau \rightsquigarrow x} \quad \frac{}{\Gamma \vdash_{\text{sim}} n : \text{Nat} \rightsquigarrow n} \quad \frac{}{\Gamma \vdash_{\text{sim}} i : \text{Int} \rightsquigarrow i}$$

These cases are all immediate.

$$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \langle t_1, t_2 \rangle : \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle} \quad \frac{\Gamma \vdash_{\text{sim}} t : \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{fst } t : \tau \rightsquigarrow \text{fst}\{\tau\} e} \quad \frac{\Gamma \vdash_{\text{sim}} t : \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{snd } t : \tau' \rightsquigarrow \text{snd}\{\tau'\} e}$$

These cases are all immediate by the IH applied to their premises and their corresponding typing rule in sim.

$$\frac{\Gamma, (x:\tau) \vdash_{\text{sim}} t : \tau'' \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \lambda(x:\tau) \rightarrow \tau'. t : \tau \rightarrow \tau' \rightsquigarrow \lambda(x:\tau). ([\tau' \checkmark \tau'']e)} \quad \frac{\Gamma \vdash_{\text{sim}} t_1 : \tau \rightarrow \tau' \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau'' \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} t_1 t_2 : \tau' \rightsquigarrow \text{app}\{\tau'\} e_1 ([\tau \checkmark \tau'']e_2)}$$

These cases proceed similarly.

First we apply the IH to all premises.

Then we either use subsumption to typecheck the body or argument respectively if the types are subtype related, or use T-CAST if they're instead compatible subtypes.

Finally, we use the corresponding typing rule to typecheck the elimination form.

$$\frac{\Gamma \vdash_{\text{sim}} t_1 : * \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau'}{\Gamma \vdash_{\text{sim}} t_1 t_2 : * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow* \leftarrow *\} e_1) [* \checkmark \tau'] e_2} \quad \frac{\Gamma \vdash_{\text{sim}} t : * \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{fst } t : * \rightsquigarrow \text{fst}\{*\} (\text{cast}\{*\times* \leftarrow *\} e)}$$

$$\frac{\Gamma \vdash_{\text{sim}} t : * \rightsquigarrow e}{\Gamma \vdash_{\text{sim}} \text{snd } t : * \rightsquigarrow \text{snd}\{*\} (\text{cast}\{*\times* \leftarrow *\} e)}$$

All of these cases proceed similarly.

First, we apply the IH to all premises.

Then we typecheck the casts with T-CAST.

Finally we use the corresponding typing rule to typecheck the elimination form.

$$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2 \quad \Delta(\text{binop}, \tau_1 \sqsupseteq \tau_2, \tau_1 \sqsupseteq \tau_2) = \tau' \quad \tau_1 \leq \text{Int} \wedge \tau_2 \leq \text{Int}}{\Gamma \vdash_{\text{sim}} \text{binop } t_1 t_2 : \tau' \rightsquigarrow \text{binop } e_1 e_2}$$

By the IH, we have $\Gamma \vdash_{\text{sim}} e_1 : \tau_1$.

By the IH, we have $\Gamma \vdash_{\text{sim}} e_2 : \tau_2$.

Then we can use subsumption to get both $\Gamma \vdash_{\text{sim}} e_1 : \tau_1 \sqsupseteq \tau_2$ and $\Gamma \vdash_{\text{sim}} e_2 : \tau_1 \sqsupseteq \tau_2$.

Finally we can typecheck with T-BINOP.

4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992

$$\frac{\Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \text{binop } t_1 t_2 : \tau' \rightsquigarrow \text{binop}([\text{Int} \swarrow \tau_1]e_1)([\text{Int} \swarrow \tau_2]e_2)}$$

By the IH, we have $\Gamma \vdash_{\text{sim}} e_1 : \tau_1$.

By the IH, we have $\Gamma \vdash_{\text{sim}} e_2 : \tau_2$.

If $\tau_1 \leq \text{Int}$, then $[\text{Int} \swarrow \tau_1]e_1 = \text{cast} \{ \text{Int} \leftarrow \tau_1 \} e_1$, and by the IH we have $\Gamma \vdash_{\text{sim}} \text{cast} \{ \text{Int} \leftarrow \tau_1 \} e_1 : \text{Int}$.

Otherwise, $[\text{Int} \swarrow \tau_1]e_1 = e_1$.

If $\tau_2 \leq \text{Int}$, then $[\text{Int} \swarrow \tau_2]e_2 = \text{cast} \{ \text{Int} \leftarrow \tau_2 \} e_2$, and by the IH we have $\Gamma \vdash_{\text{sim}} \text{cast} \{ \text{Int} \leftarrow \tau_2 \} e_2 : \text{Int}$.

Otherwise, $[\text{Int} \swarrow \tau_2]e_2 = e_2$.

Finally we can typecheck with T-BINOP and potentially T-SUBSUMPTION.

$$\frac{\Gamma \vdash_{\text{sim}} t_b : \text{Bool} \rightsquigarrow e_b \quad \Gamma \vdash_{\text{sim}} t_1 : \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash_{\text{sim}} t_2 : \tau_2 \rightsquigarrow e_2}{\Gamma \vdash_{\text{sim}} \text{if } t_b \text{ then } t_1 \text{ else } t_2 : \tau_1 \sqsupset \tau_2 \rightsquigarrow \text{if } e_b \text{ then } ([\tau_1 \sqsupset \tau_2 \swarrow \tau_1]e_1) \text{ else } ([\tau_1 \sqsupset \tau_2 \swarrow \tau_2]e_2)}$$

By the IH, we have $\Gamma \vdash_{\text{sim}} e_b : \text{Bool}$.

By the IH, we have $\Gamma \vdash_{\text{sim}} e_1 : \tau_1$.

By the IH, we have $\Gamma \vdash_{\text{sim}} e_2 : \tau_2$.

If $\tau_1 \leq \tau_1 \sqsupset \tau_2$, then by subsumption, we have $\Gamma \vdash_{\text{sim}} e_1 : \tau_1 \sqsupset \tau_2$.

Otherwise, by T-CAST, we have $\Gamma \vdash_{\text{sim}} \text{cast} \{ \tau_1 \sqsupset \tau_2 \leftarrow \tau_1 \} e_1 : \tau_1 \sqsupset \tau_2$.

If $\tau_2 \leq \tau_1 \sqsupset \tau_2$, then by subsumption, we have $\Gamma \vdash_{\text{sim}} e_2 : \tau_1 \sqsupset \tau_2$.

Otherwise, by T-CAST, we have $\Gamma \vdash_{\text{sim}} \text{cast} \{ \tau_1 \sqsupset \tau_2 \leftarrow \tau_2 \} e_2 : \tau_1 \sqsupset \tau_2$.

Finally, we can typecheck with T-IF. □

7.2 Tag Transient Translation

$$\Gamma \vdash_{\text{tag}} t : K \rightsquigarrow e$$

$$\Gamma \vdash_{\text{tag}} t : K \rightsquigarrow e \text{ iff } \exists \tau. \lfloor \tau \rfloor = K \wedge \Gamma \vdash_{\text{sim}} t : \tau \rightsquigarrow e' \wedge e = (e')^+$$

THEOREM 7.2 (TYPED TRANSLATION IMPLIES SIMPLE TYPING). *If $\Gamma \vdash_{\text{tag}} t : K \rightsquigarrow e$ then $\Gamma \vdash_{\text{tag}} e : K$.*

PROOF. By Theorem 7.1 and Theorem 3.1. □

7.3 Truer Transient Translation

$$\tau \setminus K = \begin{cases} * & \text{if } K \leq \tau \\ \tau & \text{otherwise} \end{cases}$$

5045	$\Gamma \vdash_{\text{tru}} t \Rightarrow \tau \rightsquigarrow e : \tau'$
5046	
5047	
5048	$(x:K) \in \Gamma$
5049	$\frac{}{\Gamma \vdash_{\text{tru}} x \Rightarrow K \rightsquigarrow x : K} \quad \frac{}{\Gamma \vdash_{\text{tru}} n \Rightarrow \text{Nat} \rightsquigarrow n : \text{Nat}} \quad \frac{}{\Gamma \vdash_{\text{tru}} i \Rightarrow \text{Int} \rightsquigarrow i : \text{Int}}$
5050	
5051	$\frac{\Gamma, (x:K) \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e : \tau'}{\Gamma \vdash_{\text{tru}} \lambda(x:K) \rightarrow \tau. t \Rightarrow * \rightarrow \tau \rightsquigarrow \lambda(x:K). e : * \rightarrow \tau'} \quad \frac{\Gamma \vdash_{\text{tru}} t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} \langle t_1, t_2 \rangle \Rightarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle : \tau'_1 \times \tau'_2}$
5052	
5053	
5054	
5055	$\frac{\Gamma \vdash_{\text{tru}} t_1 \Rightarrow * \rightarrow \tau \rightsquigarrow e_1 : * \rightarrow \tau' \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} t_1 t_2 \Rightarrow \tau \rightsquigarrow \text{app}\{*\} e_1 e_2 : \tau'}$
5056	
5057	
5058	
5059	$\frac{\Gamma \vdash_{\text{tru}} t_1 \Rightarrow * \rightsquigarrow e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau' \rightsquigarrow e_2 : \tau_2 \quad \tau_1 \sqcap * \rightarrow * = * \rightarrow \tau'_1}{\Gamma \vdash_{\text{tru}} t_1 t_2 \Rightarrow * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow* \leftarrow *\} e_1) e_2 : \tau'_1}$
5060	
5061	
5062	
5063	$\frac{\Gamma \vdash_{\text{tru}} t_1 \Rightarrow * \rightsquigarrow e_1 : \tau_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau' \rightsquigarrow e_2 : \tau_2 \quad \tau_1 \sqcap * \rightarrow * = \perp \quad \Gamma \vdash_{\text{tru}} t \Rightarrow \tau \times \tau' \rightsquigarrow e : \tau''}{\Gamma \vdash_{\text{tru}} t_1 t_2 \Rightarrow * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow* \leftarrow *\} e_1) e_2 : \perp \quad \Gamma \vdash_{\text{tru}} \text{fst } t \Rightarrow \tau \rightsquigarrow \text{fst}\{*\} e : \text{fst}(\tau'')}$
5064	
5065	
5066	
5067	$\frac{\Gamma \vdash_{\text{tru}} t \Rightarrow * \rightsquigarrow e : \tau \quad \tau \sqcap * \times * = \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \text{fst } t \Rightarrow * \rightsquigarrow \text{fst}\{*\} (\text{cast}\{*\times* \leftarrow *\} e) : \tau_1} \quad \frac{\Gamma \vdash_{\text{tru}} t \Rightarrow * \rightsquigarrow e : \tau \quad \tau \sqcap * \times * = \perp}{\Gamma \vdash_{\text{tru}} \text{fst } t \Rightarrow * \rightsquigarrow \text{fst}\{*\} (\text{cast}\{*\times* \leftarrow *\} e) : \perp}$
5068	
5069	
5070	$\frac{\Gamma \vdash_{\text{tru}} t \Rightarrow \tau \times \tau' \rightsquigarrow e : \tau''}{\Gamma \vdash_{\text{tru}} \text{snd } t \Rightarrow \tau \rightsquigarrow \text{snd}\{*\} e : \text{snd}(\tau'')} \quad \frac{\Gamma \vdash_{\text{tru}} t \Rightarrow * \rightsquigarrow e : \tau \quad \tau \sqcap * \times * = \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \text{snd } t \Rightarrow * \rightsquigarrow \text{snd}\{*\} (\text{cast}\{*\times* \leftarrow *\} e) : \tau_2}$
5071	
5072	
5073	
5074	$\frac{\Gamma \vdash_{\text{tru}} t \Rightarrow * \rightsquigarrow e : \tau \quad \tau \sqcap * \times * = \perp}{\Gamma \vdash_{\text{tru}} \text{snd } t \Rightarrow * \rightsquigarrow \text{snd}\{*\} (\text{cast}\{*\times* \leftarrow *\} e) : \perp}$
5075	
5076	
5077	
5078	$\frac{\Gamma \vdash_{\text{tru}} t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 : \tau'_2 \quad \Delta(\text{binop}, \tau_1, \tau_2) = \tau' \quad \Delta(\text{binop}, \tau'_1, \tau'_2) = \tau''}{\Gamma \vdash_{\text{tru}} \text{binop } t_1 t_2 \Rightarrow \tau' \rightsquigarrow \text{binop } e_1 e_2 : \tau''}$
5079	
5080	
5081	
5082	$\frac{\Gamma \vdash_{\text{tru}} t_b \Rightarrow \text{Bool} \rightsquigarrow e_b : \text{Bool} \quad \Gamma \vdash_{\text{tru}} t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Rightarrow \tau_1 \sqcup \tau_2 \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \tau'_1 \sqcup \tau'_2}$
5083	
5084	
5085	
5086	$\frac{\Gamma \vdash_{\text{tru}} t_b \Rightarrow \text{Bool} \rightsquigarrow e_b : \perp \quad \Gamma \vdash_{\text{tru}} t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Rightarrow \tau_1 \sqcup \tau_2 \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \perp}$
5087	
5088	$\Gamma \vdash_{\text{tru}} t \Leftarrow \Rightarrow \tau \rightsquigarrow e : \tau'$
5089	
5090	
5091	$\frac{\Gamma \vdash_{\text{tru}} t \Rightarrow \tau' \rightsquigarrow e : \tau'' \quad \tau' \leq \tau}{\Gamma \vdash_{\text{tru}} t \Leftarrow \Rightarrow \tau \rightsquigarrow e : \tau''} \quad \frac{\Gamma \vdash_{\text{tru}} t \Rightarrow \tau' \rightsquigarrow e : \tau'' \quad \tau' \not\leq K}{\Gamma \vdash_{\text{tru}} t \Leftarrow \Rightarrow K \rightsquigarrow \text{cast}\{K \leftarrow [\tau']\} e : K \sqcap [\tau'] \sqcap \tau''}$
5092	
5093	
5094	
5095	
5096	

$$\boxed{\Gamma \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e : \tau'}$$

5097

5098

5099

5100

5101

5102

5103

5104

5105

5106

5107

5108

5109

5110

5111

5112

5113

5114

5115

5116

5117

5118

5119

5120

5121

5122

$$\boxed{\Gamma \vdash_{\text{tru}} t \Rightarrow \tau \rightsquigarrow e}$$

5123

5124

5125

5126

$$\boxed{\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e}$$

5127

5128

5129

5130

$$\boxed{\Gamma \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e}$$

5131

5132

5133

5134

$$\boxed{\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e}$$

5135

5136

5137

5138

5139

5140

5141

5142

5143

5144

5145

5146

5147

5148

$$\frac{\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : \tau' \quad \neg(\exists e, \tau'. \Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : \tau') \quad \Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : \tau'}{\Gamma \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e : \tau'}$$

$$\boxed{\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : \tau'}$$

$$\frac{\Gamma \vdash_{\text{tru}} t_1 \Leftarrow^+ \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Leftarrow^+ \tau_2 \rightsquigarrow e_2 : \tau'_2 \quad \Gamma \vdash_{\text{tru}} t \Leftarrow^+ (\tau \setminus [\tau]) \times * \rightsquigarrow e : \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \langle t_1, t_2 \rangle \Leftarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle : \tau'_1 \times \tau'_2 \quad \Gamma \vdash_{\text{tru}} \text{fst } t \Leftarrow \tau \rightsquigarrow \text{fst}\{[\tau]\} e : \tau_1 \sqcap [\tau]}$$

$$\frac{\Gamma \vdash_{\text{tru}} t \Leftarrow^+ * \times (\tau \setminus [\tau]) \rightsquigarrow e : \tau_1 \times \tau_2}{\Gamma \vdash_{\text{tru}} \text{snd } t \Leftarrow \tau \rightsquigarrow \text{snd}\{[\tau]\} e : \tau_2 \sqcap [\tau]}$$

$$\frac{\Gamma \vdash_{\text{tru}} t_b \Leftarrow^+ \text{Bool} \rightsquigarrow e_b : \text{Bool} \quad \Gamma \vdash_{\text{tru}} t_1 \Leftarrow^+ \tau \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Leftarrow^+ \tau \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Leftarrow \tau \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \tau'_1 \sqcup \tau'_2}$$

$$\frac{\Gamma \vdash_{\text{tru}} t_b \Leftarrow^+ \text{Bool} \rightsquigarrow e_b : \perp \quad \Gamma \vdash_{\text{tru}} t_1 \Leftarrow^+ \tau \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Leftarrow^+ \tau \rightsquigarrow e_2 : \tau'_2}{\Gamma \vdash_{\text{tru}} \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Leftarrow \tau \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \perp}$$

$$\frac{\Gamma \vdash_{\text{tru}} t_1 \Leftarrow^+ \tau_1 \rightsquigarrow e_1 : \tau'_1 \quad \Gamma \vdash_{\text{tru}} t_2 \Leftarrow^+ \tau_2 \rightsquigarrow e_2 : \tau'_2 \quad \Delta^{-1}(\text{binop}, \tau') = \tau_1, \tau_2 \quad \Delta(\text{binop}, \tau'_1, \tau'_2) = \tau''}{\Gamma \vdash_{\text{tru}} \text{binop } t_1 t_2 \Leftarrow \tau' \rightsquigarrow \text{binop } e_1 e_2 : \tau''}$$

$$\Gamma \vdash_{\text{tru}} t \Rightarrow \tau \rightsquigarrow e \text{ iff } \Gamma \vdash_{\text{tru}} t \Rightarrow \tau \rightsquigarrow e : _$$

$$\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e \text{ iff } \Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : _$$

$$\Gamma \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e \text{ iff } \Gamma \vdash_{\text{tru}} t \Leftarrow^+ \tau \rightsquigarrow e : _$$

$$\Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e \text{ iff } \Gamma \vdash_{\text{tru}} t \Leftarrow \tau \rightsquigarrow e : _$$

For the purpose of the following proof, assume the tru rules are used in each judgement.

LEMMA 7.3 (TYPED TRANSLATIONS IMPLY TRUER TRANSIENT TYPING).

- (1) If $\Gamma \vdash t \Rightarrow \tau \rightsquigarrow e : \tau'$ then $\Gamma \vdash e : \tau'$ with $\tau' \leq \tau$.
- (2) If $\Gamma \vdash t \Leftarrow \tau \rightsquigarrow e : \tau'$ then $\Gamma \vdash e : \tau'$ with $\tau' \leq \tau$.
- (3) If $\Gamma \vdash t \Leftarrow^+ \tau \rightsquigarrow e : \tau'$ then $\Gamma \vdash e : \tau'$ with $\tau' \leq \tau$.
- (4) If $\Gamma \vdash t \Leftarrow \tau \rightsquigarrow e : \tau'$ then $\Gamma \vdash e : \tau'$ with $\tau' \leq \tau$.

5149 **PROOF.** All cases proceed by induction over their respective judgement derivations.

5150 This is well founded by the size of the term e , with the caveat that (2) will call into (1) with the same term, but (1) will
5151 then reduce the size before calling back into (2) (in the lambda case, through (3)).
5152

5153 Similarly, (3) will call into (2), but by the time it gets back to (3), the term will have been reduced in size in (1) (in the
5154 lambda case).

5155 And similarly, (3) will call into (4), but by the time it gets back to (3), the term will have reduced in size.
5156

$$5157 \frac{(x:K) \in \Gamma}{\Gamma \vdash x \Rightarrow K \rightsquigarrow x} \qquad \frac{}{\Gamma \vdash n \Rightarrow \text{Nat} \rightsquigarrow n} \qquad \frac{}{\Gamma \vdash i \Rightarrow \text{Int} \rightsquigarrow i}$$

5161 All of the above cases follow immediately.
5162

$$5163 \frac{\Gamma \vdash t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Rightarrow \tau_2 \rightsquigarrow e_2}{\Gamma \vdash \langle t_1, t_2 \rangle \Rightarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle}$$

5167 Follows immediately by the induction hypotheses.
5168

$$5169 \frac{\Gamma \vdash t_1 \Rightarrow * \rightarrow \tau \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Rightarrow \tau'}{\Gamma \vdash t_1 t_2 \Rightarrow \tau \rightsquigarrow \text{app}\{*\} t_1 t_2} \qquad \frac{\Gamma \vdash t \Rightarrow \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash \text{fst } t \Rightarrow \tau \rightsquigarrow \text{fst}\{*\} e} \qquad \frac{\Gamma \vdash t \Rightarrow \tau \times \tau' \rightsquigarrow e}{\Gamma \vdash \text{snd } t \Rightarrow \tau \rightsquigarrow \text{snd}\{*\} e}$$

5173 All of the above cases follow similar reasoning.
5174

5174 We apply the induction hypothesis to each premise.

5175 If the term being eliminated is at type \perp , then we use the corresponding \perp rule.
5176

5177 Otherwise we use the corresponding elimination rule with check $*$.
5178

$$5179 \frac{\Gamma \vdash t_1 \Rightarrow * \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Rightarrow \tau'}{\Gamma \vdash t_1 t_2 \Rightarrow * \rightsquigarrow \text{app}\{*\} (\text{cast}\{*\rightarrow*\} \leftarrow *) t_1 t_2} \qquad \frac{\Gamma \vdash t \Rightarrow * \rightsquigarrow e}{\Gamma \vdash \text{fst } t \Rightarrow * \rightsquigarrow \text{fst}\{*\} (\text{cast}\{*\times*\} \leftarrow *) e}$$

$$5181 \frac{\Gamma \vdash t \Rightarrow * \rightsquigarrow e}{\Gamma \vdash \text{snd } t \Rightarrow * \rightsquigarrow \text{snd}\{*\} (\text{cast}\{*\times*\} \leftarrow *) e}$$

5186 All of the above cases follow similar reasoning.
5187

5188 The reasoning is identical to the previous case, with the note that the boundary term also sends the type below the tag
5189 corresponding to the kind of elimination form.
5190

$$5191 \frac{\Gamma \vdash t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Rightarrow \tau_2 \rightsquigarrow e_2 \quad \Delta(\text{binop}, \tau_1, \tau_2) = \tau}{\Gamma \vdash \text{binop } t_1 t_2 \Rightarrow \tau' \rightsquigarrow \text{binop } e_1 e_2}$$

5196 From (1) we get that there is a $\tau'_1 \leq \tau_1$ such that $\Gamma \vdash e_1 : \tau'_1$.

5197 From (1) we get that there is a $\tau'_2 \leq \tau_2$ such that $\Gamma \vdash e_2 : \tau'_2$.

5198 If $\tau'_1 = \perp$ or $\tau'_2 = \perp$ then were done, because $\Delta(\text{binop}, \tau'_1, \tau'_2) = \perp$.
5199

5200

5201 Otherwise, $\tau'_1 = \text{Int}$ or Nat and $\tau'_2 = \text{Int}$ or Nat . If $\tau'_1 \neq \tau'_2$, we can use subsumption to get both e_1 and e_2 at Int to
 5202 complete the case.

5203 Otherwise they're both at Nat or Int , which is sufficient to complete the case.
 5204
 5205

$$\frac{\Gamma \vdash t_b \Rightarrow \text{Bool} \rightsquigarrow e_b \quad \Gamma \vdash t_1 \Rightarrow \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Rightarrow \tau_2 \rightsquigarrow e_2}{\Gamma \vdash \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Rightarrow \tau_1 \sqcup \tau_2 \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2}$$

5209 By (1) we have $\exists \tau_b \leq \text{Bool}$ such that $\Gamma \vdash e_b : \tau_b$.

5210 By (1) we have $\exists \tau_1 \leq \tau$ such that $\Gamma \vdash e_1 : \tau_1$.

5211 By (1) we have $\exists \tau_2 \leq \tau$ such that $\Gamma \vdash e_2 : \tau_2$.

5212 If $\tau_b = \perp$, then were done by the if bot rule.

5213 Otherwise, we get by the if rule that $\Gamma \vdash \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \tau_1 \sqcup \tau_2$, and that $\tau_1 \sqcup \tau_2 \leq \tau$ by the fact that \sqcup is a
 5214 greatest lower bound.
 5215
 5216

$$\frac{\Gamma, (x:K) \vdash t \Leftarrow^+ \tau \rightsquigarrow e}{\Gamma \vdash \lambda(x:K) \rightarrow \tau. t \Rightarrow * \rightarrow \tau \rightsquigarrow \lambda(x:K). e}$$

5217
 5218 By the lambda typing rule for truer typing, we want to show there is a $\tau' \leq \tau$ such that $\Gamma, (x:K) \vdash e : \tau'$.

5219 This is immediate from (3) applied to the premise.
 5220
 5221

$$\frac{\Gamma \vdash t \Rightarrow \tau' \rightsquigarrow e \quad \tau' \leq \tau}{\Gamma \vdash t \Leftarrow \tau \rightsquigarrow e}$$

5222 By (1), we have there is a $\tau'' \leq \tau'$ such that $\Gamma \vdash t : \tau''$.

5223 Since \leq is transitive, this completes the case.
 5224
 5225

$$\frac{\Gamma \vdash t \Rightarrow \tau' \rightsquigarrow e \quad \tau' \not\leq K}{\Gamma \vdash t \Leftarrow K \rightsquigarrow \text{cast } \{K \Leftarrow \lfloor \tau' \rfloor\} e}$$

5226
 5227 From (1) we have $\tau'' \leq \tau'$ such that $\Gamma \vdash e : \tau''$.

5228 We want to show there is a $\tau''' \leq K$ such that $\Gamma \vdash \text{cast } \{K \Leftarrow \lfloor \tau' \rfloor\} e : \tau'''$.

5229 Set $\tau''' \sqcap \lfloor \tau' \rfloor \sqcap K$ to be τ''' .

5230 By the boundary typing rule of truer typing, this typechecks.

5231 The last condition is that $\tau''' \leq K$, which is immediate by the fact that \sqcap is the greatest lower bound.
 5232
 5233

$$\frac{\neg(\exists e. \Gamma \vdash t \Leftarrow \tau \rightsquigarrow e) \quad \Gamma \vdash t \Leftarrow \tau \rightsquigarrow e}{\Gamma \vdash t \Leftarrow^+ \tau \rightsquigarrow e}$$

5234
 5235 Immediate by (2).
 5236
 5237
 5238
 5239
 5240
 5241
 5242
 5243
 5244
 5245
 5246
 5247
 5248
 5249
 5250
 5251
 5252

5253

5254

5255

$$\frac{\Gamma \vdash t \Leftarrow \tau \rightsquigarrow e}{\Gamma \vdash t \Leftarrow^+ \tau \rightsquigarrow e}$$

5256

5257

Immediate by (4).

5258

5259

5260

5261

$$\frac{\Gamma \vdash t_1 \Leftarrow^+ \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Leftarrow^+ \tau_2 \rightsquigarrow e_2}{\Gamma \vdash \langle t_1, t_2 \rangle \Leftarrow \tau_1 \times \tau_2 \rightsquigarrow \langle e_1, e_2 \rangle}$$

5262

5263

5264

Immediate by (3) and induction.

5265

5266

5267

$$\frac{\Gamma \vdash t \Leftarrow^+ (\tau \setminus \lfloor \tau \rfloor) \times * \rightsquigarrow e}{\Gamma \vdash \text{fst } t \Leftarrow \tau \rightsquigarrow \text{fst}\{\lfloor \tau \rfloor\} e}$$

5268

5269

5270

By our induction hypothesis, we have that there is some $\tau' \leq (\tau \setminus \lfloor \tau \rfloor) \times *$ such that $\Gamma \vdash e : \tau'$.

5271

If $\tau' = \perp$, then were done by the fst bot rule.

5272

Otherwise, $\tau' = \tau'_1 \times \tau'_2$, and $\tau'_1 \leq \tau \setminus \lfloor \tau \rfloor$.

5273

5274

By the fst projection typing rule, we have that $\Gamma \vdash \text{fst}\{\lfloor \tau \rfloor\} e : \tau'_1 \sqcap \lfloor \tau \rfloor$.

5275

It suffices to show that $\tau'_1 \sqcap \lfloor \tau \rfloor \leq \tau$.

5276

If $\tau \setminus \lfloor \tau \rfloor = *$, then $\lfloor \tau \rfloor \leq \tau$, which means $\tau'_1 \sqcap \lfloor \tau \rfloor \leq \lfloor \tau \rfloor \leq \tau$.

5277

Otherwise, $\tau \setminus \lfloor \tau \rfloor = \tau$, which means $\tau'_1 \leq \tau$ and therefore $\tau'_1 \sqcap \lfloor \tau \rfloor \leq \tau$.

5278

5279

5280

$$\frac{\Gamma \vdash t \Leftarrow^+ * \times (\tau \setminus \lfloor \tau \rfloor) \rightsquigarrow e}{\Gamma \vdash \text{snd } t \Leftarrow \tau \rightsquigarrow \text{snd}\{\lfloor \tau \rfloor\} e}$$

5281

5282

Not meaningfully different from the previous case regarding fst .

5283

5284

5285

5286

$$\frac{\Gamma \vdash t_b \Leftarrow^+ \text{Bool} \rightsquigarrow e_b \quad \Gamma \vdash t_1 \Leftarrow^+ \tau \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Leftarrow^+ \tau \rightsquigarrow e_2}{\Gamma \vdash \text{if } e_b \text{ then } t_1 \text{ else } t_2 \Leftarrow \tau \rightsquigarrow \text{if } e_b \text{ then } e_1 \text{ else } e_2}$$

5287

5288

By (3) we have $\exists \tau_b \leq \text{Bool}$ such that $\Gamma \vdash e_b : \tau_b$.

5289

By (3) we have $\exists \tau_1 \leq \tau$ such that $\Gamma \vdash e_1 : \tau_1$.

5290

By (3) we have $\exists \tau_2 \leq \tau$ such that $\Gamma \vdash e_2 : \tau_2$.

5291

If $\tau_b = \perp$, then were done by the if bot rule.

5292

Otherwise, we get by the if rule that $\Gamma \vdash \text{if } e_b \text{ then } e_1 \text{ else } e_2 : \tau_1 \sqcup \tau_2$, and that $\tau_1 \sqcup \tau_2 \leq \tau$ by the fact that \sqcup is a greatest lower bound.

5293

5294

5295

5296

5297

5298

$$\frac{\Gamma \vdash t_1 \Leftarrow^+ \tau_1 \rightsquigarrow e_1 \quad \Gamma \vdash t_2 \Leftarrow^+ \tau_2 \rightsquigarrow e_2 \quad \Delta^{-1}(\text{binop}, \tau') = \tau_1, \tau_2}{\Gamma \vdash \text{binop } t_1 t_2 \Leftarrow \tau' \rightsquigarrow \text{binop } e_1 e_2}$$

5299

5300

By (3) we have $\exists \tau'_1 \leq \tau_1$ such that $\Gamma \vdash e_1 : \tau'_1$.

5301

By (3) we have $\exists \tau'_2 \leq \tau_2$ such that $\Gamma \vdash e_2 : \tau'_2$.

5302

5303

5304

5305 By the definition of Δ^{-1} , either $\tau_1 = \tau_2 = \text{Int}$ or $\tau_1 = \tau_2 = \text{Nat}$.

5306 If $\tau'_1 = \perp$ or $\tau'_2 = \perp$, then were done because $\Delta(\text{binop}, \tau'_1, \tau'_2) = \perp$.

5307 Otherwise, we have $\tau'_1 = \text{Int}$ or Nat and similarly for τ'_2 .

5308 If $\tau'_1 \neq \tau'_2$, then we can use subsumption to get both at Int and complete the case.

5309 Otherwise, we get that both are Int or Nat , which is sufficient to complete the case. □

5310

5311 **THEOREM 7.4 (TYPED TRANSLATION IMPLIES TRUER TRANSIENT TYPING).**

5312 *If $\Gamma \vdash t \Rightarrow \tau \rightsquigarrow e$ then $\Gamma \vdash e : \tau$.*

5313

5314 **PROOF.** Follows from Lemma 7.3 and T-SUB □

5315

5316

5317

5318

5319

5320

5321

5322

5323

5324

5325

5326

5327

5328

5329

5330

5331

5332

5333

5334

5335

5336

5337

5338

5339

5340

5341

5342

5343

5344

5345

5346

5347

5348

5349

5350

5351

5352

5353

5354

5355

5356

5357 8 Vigilance Results for GTLs

5358

5359 8.1 GTL Vigilance for Simple Typing with Natural Semantics

5360

5361 THEOREM 8.1 (VIGILANCE FOR SIMPLE TYPING WITH NATURAL SEMANTICS). *If $\Gamma \vdash_{\text{sim}} t : \tau \rightsquigarrow e$ then $\llbracket \Gamma \vdash e : \tau \rrbracket^N$*

5362

5363 PROOF. By Theorem 7.1 and Theorem 5.40. □

5364

5365 8.2 GTL Vigilance for Tag Typing with Transient Semantics

5366

5367 THEOREM 8.2 (VIGILANCE FOR TAG TYPING WITH TRANSIENT SEMANTICS). *If $\Gamma \vdash_{\text{tag}} t : K \rightsquigarrow e$ then $\llbracket \Gamma \vdash e : K \rrbracket^N$*

5368

5369 PROOF. By Theorem 7.2 and Theorem 5.90. □

5370

5371 8.3 GTL Vigilance for Truer Transient Typing with Transient Semantics

5372

5373 THEOREM 8.3 (VIGILANCE FOR SIMPLE TYPING WITH NATURAL SEMANTICS). *If $\Gamma \vdash_{\text{tru}} t : \tau \rightsquigarrow e$ then $\llbracket \Gamma \vdash e : \tau \rrbracket^N$*

5374

5375 PROOF. By Theorem 7.4 and Theorem 5.89. □

5376

5377

5378

5379

5380

5381

5382

5383

5384

5385

5386

5387

5388

5389

5390

5391

5392

5393

5394

5395

5396

5397

5398

5399

5400

5401

5402

5403

5404

5405

5406

5407

5408