# Verifying and Synthesizing Constant-Resource Implementations with Types: Full Language Semantics, Typing Rules, and Proofs

## 1. The language

The equivalent expressions in OCaml syntax of the language are given as follows.

$$
\begin{array}{rl}
e & ::= () \mid \mathsf{true} \mid \mathsf{false} \mid n \mid x \\
& \mid x_1 \diamond x_2 \\
& \mid f(x) \\
& \mid \text{let } x = e_1 \text{ in } e_2 \\
& \mid \text{if } x \text{ then } e_t \text{ else } e_f \\
& \mid (x_1, x_2) \\
& \mid \text{match } x \text{ with } (x_1, x_2) \to e \\
& \mid [] \\
& \mid x_1 :: x_2 \\
& \mid \text{match } x \text{ with } \mid [] \to e_1 \mid x_1 :: x_2 \to e_2 \\
& \mid \text{share } x = (x_1, x_2) \text{ in } e \\
\\
& \diamond \in \{+, -, *, \mathsf{div}, \mathsf{mod}, =, <>, >, <, <=, >=, \mathsf{and}, \mathsf{or}\}
\end{array}
$$

Fig. 1, Fig. 2, and Fig. 3 represent the typing rules for values, the base typing and the evaluation rules for the language, respectively.

## 2. Type systems for lower bounds and constant resource

The common typing rules except the relax and structural rules for all of three type systems; upper bounds, constant resource, and lower bounds are represented in Fig. 4. While the different rules are shown in Fig. 5, Fig. 6, and Fig. 7. We can see that the relax rules are consistent among these type systems in sense of satisfying the following.

$$
\begin{array}{l}
(q \geq p \wedge q - p \leq q' - p') \wedge (q \geq p \wedge q - p \geq q' - p') \quad \Leftrightarrow \\
(q \geq p \wedge q - p = q' - p')
\end{array}
$$

That means the constraints for upper bounds and lower bounds imply the constraints for constant resource and vice versa.

The type systems for upper bounds, constant resource, and lower bounds are *affine*, *linear*, and *relevant* substructural type systems, respectively. That means:

- Type system for upper bounds allows exchange and weakening, but not contraction properties.
- Type system for constant resource allows exchange but not weakening or contraction properties.
- Type system for lower bounds allows exchange and contraction, but not weakening properties.

**Lemma 1.** *If* $E_1 \approx_X E_2$ *then* $\Phi_{E_1}(X : \Gamma^r) = \Phi_{E_2}(X : \Gamma^r)$.

*Proof.* The claim is proved by induction on the definitions of potential and size-equivalence, in which $|E_1(x)| \approx |E_2(x)|$ implies $\Phi(E_1(x) : \Gamma^r(x)) = \Phi(E_2(x) : \Gamma^r(x))$. □

**Theorem 1.** *If* $\models E : \Gamma^r$, $E \vdash e \Downarrow v$, *and* $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : A$, *then for all* $p, r \in \mathbb{Q}_0^+$ *such that* $p = q + \Phi_E(\Gamma^r) + r$, *there exists* $p' \in \mathbb{Q}_0^+$ *satisfying* $E \vdash^{\frac{p}{p'}} e \Downarrow v$ *and* $p' = q' + \Phi(v : A) + r$.

*Proof.* The proof is done by induction on the length of the derivation of the evaluation judgment and the typing judgment with lexical order, in which the derivation of the evaluation judgment takes priority over the typing derivation. We need to do induction on the length of both evaluation and typing derivations since on one hand, an induction of only typing derivation would fail for the case of function application, which increases the length of the typing derivation, while the length of the evaluation derivation never increases. On the other hand, if the rule C:WEAKENING is final step in the derivation, then the length of typing derivation decreases, while the length of evaluation derivation is unchanged.

**A:SHARE** Assume that the typing derivation ends with an application of the rule A:SHARE, thus $\Sigma^r; \Gamma^r, x_1 : A_1, x_2 : A_2 \vdash^{\frac{q}{q'}} e : B$ and $\curlyvee(A|A_1, A_2)$.

Let $E_1 = E \setminus \{x\} \cup \{[x_1 \mapsto E(x), x_2 \mapsto E(x)]\}$. Since $\models E : \Gamma^r, x : A$ and following the property of the share relation we have $\models E_1 : \Gamma^r, x_1 : A_1, x_2 : A_2$. By the induction hypothesis for $e$, it holds that for all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_{E_1}(\Gamma^r, x_1 : A_1, x_2 : A_2) + r$, there exists $p' \in \mathbb{Q}_0^+$ satisfying $E_1 \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q' + \Phi(v : B) + r$.

Because $\Phi(E(x) : A) = \Phi(E_1(x_1) : A_1) + \Phi(E_1(x_2) : A_2)$ and $\Phi_E(\Gamma^r) = \Phi_{E_1}(\Gamma^r) = \Phi_{E\setminus\{x\}}(\Gamma^r)$, thus $p = q + \Phi_E(\Gamma^r, x : A) + r$ and there exists $p'$ satisfying $E \vdash^{\frac{p}{p'}} \text{share}(x, (x_1, x_2).e) \Downarrow v$.

**C:WEAKENING** Suppose that the typing derivation ends with an application of the rule C:WEAKENING. Thus we have $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : B$, in which the data type $A$ satisfies $\curlyvee(A \mid A, A)$.

Since $\models E : \Gamma^r, x : A$, it follows $\models E : \Gamma^r$. By the induction hypothesis for $e$, it holds that for all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma^r) + r$, there exists $p' \in \mathbb{Q}_0^+$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q' + \Phi(v : B) + r$. By the property of the share relation, $\Phi(a : A) = 0$, then we have $p = q + \Phi_E(\Gamma^r, x : A) + r$, $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q' + \Phi(v : B) + r$ as required.

**C:RELAX** Suppose that the typing derivation ends with an application of the rule C:RELAX, thus we have $\Sigma^r; \Gamma^r \vdash^{\frac{q_1}{q_1'}} e : A$, $q \geq q_1$, and $q - q_1 = q' - q_1'$.

For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma^r) + r = q_1 + \Phi_E(\Gamma^r) + (q - q_1) + r$, we have $\models E : \Gamma^r$. By the induction hypothesis for $e$ in the premise, there exists $p' \in \mathbb{Q}_0^+$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q_1' + \Phi(v : A) + (q - q_1) + r = q' + \Phi(v : A) + r$.

**A:VAR** Assume that $e$ is a variable $x$. If $\Sigma^r; x : A \vdash^{\frac{K^{\mathsf{var}}}{0}} x : A$. Thus for all $p, r \in \mathbb{Q}_0^+$ such that $p = K^{\mathsf{var}} + \Phi(v : A) + r$, there exists $p' = \Phi(v : A) + r$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$.

$$\text{(V:Unit)}\ \frac{v = ()}{\models v : \text{unit}} \qquad \text{(V:Bool)}\ \frac{v \in \{\text{true}, \text{false}\}}{\models v : \text{bool}} \qquad \text{(V:Int)}\ \frac{v \in \mathbb{Z}}{\models v : \text{int}}$$

$$\text{(V:Pair)}\ \frac{\models v_1 : T_1 \qquad \models v_2 : T_2}{\models (v_1, v_2) : T_1 * T_2} \qquad \text{(V:Nil)}\ \frac{v = \text{nil}}{\models v : L(T)}$$

$$\text{(V:List)}\ \frac{\models v_i : T \qquad \forall i = 1, ..., n}{\models [v_1, ..., v_n] : L(T)}$$

**Figure 1.** Typing rules: values

$$\text{(T:Unit)}\ \frac{}{\Sigma; \emptyset \vdash () : \text{unit}} \qquad \text{(T:Bool)}\ \frac{b \in \{\text{true}, \text{false}\}}{\Sigma; \emptyset \vdash b : \text{bool}} \qquad \text{(T:Int)}\ \frac{n \in \mathbb{Z}}{\Sigma; \emptyset \vdash n : \text{int}}$$

$$\text{(T:Var)}\ \frac{x \in \text{dom}(E)}{\Sigma; x : T \vdash x : T} \qquad \text{(T:B-Op)}\ \frac{\diamond \in \{\text{and}, \text{or}\}}{\Sigma; x_1 : \text{bool}, x_2 : \text{bool} \vdash \text{op}_\diamond(x_1, x_2) : \text{bool}}$$

$$\text{(T:IB-Op)}\ \frac{\diamond \in \{=, <>, >, <, <=, >=\}}{\Sigma; x_1 : \text{int}, x_2 : \text{int} \vdash \text{op}_\diamond(x_1, x_2) : \text{bool}} \qquad \text{(T:I-Op)}\ \frac{\diamond \in \{+, -, *, \text{div}, \text{mod}\}}{\Sigma; x_1 : \text{int}, x_2 : \text{int} \vdash \text{op}_\diamond(x_1, x_2) : \text{int}}$$

$$\text{(T:Fun)}\ \frac{\Sigma(g) = T_1 \to T_2}{\Sigma; x : T_1 \vdash \text{app}(g, x) : T_2}$$

$$\text{(T:Let)}\ \frac{\Sigma; \Gamma_1 \vdash e_1 : T_1 \qquad \Sigma; \Gamma_2, x : T_1 \vdash e_2 : T_2}{\Sigma; \Gamma_1, \Gamma_2 \vdash \text{let}(x, e_1, x.e_2) : T_2} \qquad \text{(T:If)}\ \frac{\Sigma; \Gamma \vdash e_t : T \qquad \Sigma; \Gamma \vdash e_f : T}{\Sigma; \Gamma, x : \text{bool} \vdash \text{if}(x, e_t, e_f) : T}$$

$$\text{(T:Pair)}\ \frac{}{\Sigma; x_1 : T_1, x_2 : T_2 \vdash \text{pair}(x_1, x_2) : T_1 * T_2}$$

$$\text{(T:Match-P)}\ \frac{\Sigma; \Gamma, x_1 : T_1, x_2 : T_2 \vdash e : T}{\Sigma; \Gamma, x : T_1 * T_2 \vdash \text{match}(x, (x_1, x_2).e) : T} \qquad \text{(T:Nil)}\ \frac{T \in \mathscr{T}}{\Sigma; \emptyset \vdash \text{nil} : L(T)}$$

$$\text{(T:Cons)}\ \frac{}{\Sigma; x_h : T, x_t : L(T) \vdash \text{cons}(x_h, x_t) : L(T)}$$

$$\text{(T:Match-L)}\ \frac{\Sigma; \Gamma \vdash e_1 : T_1 \qquad \Sigma; \Gamma, x_h : T, x_t : L(T) \vdash e_2 : T_1}{\Sigma; \Gamma, x : L(T) \vdash \text{match}(x, e_1, (x_h, x_t).e_2) : T_1}$$

$$\text{(T:Share)}\ \frac{\Sigma; \Gamma, x_1 : T, x_2 : T \vdash e : T_1}{\Sigma; \Gamma, x : T \vdash \text{share}(x, (x_1, x_2).e) : T_1} \qquad \text{(T:Weakening)}\ \frac{\Sigma; \Gamma \vdash e : T_1}{\Sigma; \Gamma, x : T \vdash e : T_1}$$

**Figure 2.** Base typing rules: language

$$\text{(E:Unit)}\ \frac{}{E \vdash^{\frac{q+K^{\text{unit}}}{q}} () \Downarrow ()} \qquad \text{(E:Bool)}\ \frac{b \in \{\text{true}, \text{false}\}}{E \vdash^{\frac{q+K^{\text{bool}}}{q}} b \Downarrow b} \qquad \text{(E:Int)}\ \frac{n \in \mathbb{Z}}{E \vdash^{\frac{q+K^{\text{int}}}{q}} n \Downarrow n}$$

$$\text{(E:Var)}\ \frac{x \in \text{dom}(E)}{E \vdash^{\frac{q+K^{\text{var}}}{q}} x \Downarrow E(x)} \qquad \text{(E:Bin)}\ \frac{v = E(x_1) \diamond E(x_2)}{E \vdash^{\frac{q+K^{\text{op}}}{q}} \text{op}_\diamond(x_1, x_2) \Downarrow v}$$

$$\text{(E:Fun)}\ \frac{\Sigma(g) = T_1 \to T_2 \qquad E[x^g \mapsto E(x)] \vdash^{\frac{q}{q'}} e_g \Downarrow v}{E \vdash^{\frac{q+K^{\text{app}}}{q'}} \text{app}(g, x) \Downarrow v}$$

$$\text{(E:Let)}\ \frac{E \vdash^{\frac{q-K^{\text{let}}}{q_1'}} e_1 \Downarrow v_1 \qquad E[x \mapsto v_1] \vdash^{\frac{q_1'}{q'}} e_2 \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{let}(x, e_1, x.e_2) \Downarrow v}$$

$$\text{(E:If-True)}\ \frac{E(x) = \text{true} \qquad E \vdash^{\frac{q-K^{\text{cond}}}{q'}} e_t \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{if}(x, e_t, e_f) \Downarrow v}$$

$$\text{(E:If-False)}\ \frac{E(x) = \text{false} \qquad E \vdash^{\frac{q-K^{\text{cond}}}{q'}} e_f \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{if}(x, e_t, e_f) \Downarrow v}$$

$$\text{(E:Pair)}\ \frac{x_1, x_2 \in \text{dom}(E) \qquad v = (E(x_1), E(x_2))}{E \vdash^{\frac{q+K^{\text{pair}}}{q}} \text{pair}(x_1, x_2) \Downarrow v} \qquad \text{(E:Nil)}\ \frac{}{E \vdash^{\frac{q+K^{\text{nil}}}{q}} \text{nil} \Downarrow \text{nil}}$$

$$\text{(E:Match-P)}\ \frac{E(x) = (v_1, v_2) \qquad E[x_1 \mapsto v_1, x_2 \mapsto v_2] \vdash^{\frac{q-K^{\text{matchP}}}{q'}} e \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{match}(x, (x_1, x_2).e) \Downarrow v}$$

$$\text{(E:Cons)}\ \frac{x_h, x_t \in \text{dom}(E) \qquad E(x_h) = v_1 \qquad E(x_t) = [v_2, .., v_n]}{E \vdash^{\frac{q+K^{\text{cons}}}{q}} \text{cons}(x_h, x_t) \Downarrow [v_1, ..., v_n]}$$

$$\text{(E:Match-N)}\ \frac{E(x) = \text{nil} \qquad E \vdash^{\frac{q-K^{\text{matchN}}}{q'}} e_1 \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{match}(x, e_1, (x_h, x_t).e_2) \Downarrow v}$$

$$\text{(E:Share)}\ \frac{E(x) = v_1 \qquad E[x_1 \mapsto v_1, x_2 \mapsto v_1] \setminus \{x\} \vdash^{\frac{q}{q'}} e \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{share}(x, (x_1, x_2).e) \Downarrow v}$$

$$\text{(E:Match-L)}\ \frac{E(x) = [v_1, ..., v_n] \qquad E[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]] \vdash^{\frac{q-K^{\text{matchL}}}{q'}} e_2 \Downarrow v}{E \vdash^{\frac{q}{q'}} \text{match}(x, e_1, (x_h, x_t).e_2) \Downarrow v}$$

**Figure 3.** Evaluation rules: language

$$\text{(A:Unit)}\ \frac{}{\Sigma^r;\emptyset \vdash^{\frac{K^{\mathsf{unit}}}{0}} () : \mathsf{unit}} \qquad \text{(A:Bool)}\ \frac{b \in \{\mathsf{true},\mathsf{false}\}}{\Sigma^r;\emptyset \vdash^{\frac{K^{\mathsf{bool}}}{0}} b : \mathsf{bool}} \qquad \text{(A:Int)}\ \frac{n \in \mathbb{Z}}{\Sigma^r;\emptyset \vdash^{\frac{K^{\mathsf{int}}}{0}} n : \mathsf{int}} \qquad \text{(A:Var)}\ \frac{}{\Sigma^r;x:A \vdash^{\frac{K^{\mathsf{var}}}{0}} x : A}$$

$$\text{(A:B-Op)}\ \frac{\diamond \in \{\mathsf{and},\mathsf{or}\}}{\Sigma^r;x_1:\mathsf{bool},x_2:\mathsf{bool} \vdash^{\frac{K^{\mathsf{op}}}{0}} \mathrm{op}_\diamond(x_1,x_2):\mathsf{bool}} \qquad \text{(A:IB-Op)}\ \frac{\diamond \in \{=,<>,>,<,<=,>=\}}{\Sigma^r;x_1:\mathsf{int},x_2:\mathsf{int} \vdash^{\frac{K^{\mathsf{op}}}{0}} \mathrm{op}_\diamond(x_1,x_2):\mathsf{bool}}$$

$$\text{(A:I-Op)}\ \frac{\diamond \in \{+,-,*,\mathsf{div},\mathsf{mod}\}}{\Sigma^r;x_1:\mathsf{int},x_2:\mathsf{int} \vdash^{\frac{K^{\mathsf{op}}}{0}} \mathrm{op}_\diamond(x_1,x_2):\mathsf{int}} \qquad \text{(A:Fun)}\ \frac{\Sigma^r(f) = A_1 \xrightarrow{q/q'} A_2}{\Sigma^r;x:A_1 \vdash^{\frac{q+K^{\mathsf{app}}}{q'}} \mathrm{app}(f,x):A_2}$$

$$\text{(A:Let)}\ \frac{\Sigma^r;\Gamma_1^r \vdash^{\frac{q-K^{\mathsf{let}}}{q_1'}} e_1 : A_1 \qquad \Sigma^r;\Gamma_2^r,x:A_1 \vdash^{\frac{q_1'}{q'}} e_2 : A_2}{\Sigma^r;\Gamma_1^r,\Gamma_2^r \vdash^{\frac{q}{q'}} \mathrm{let}(x,e_1,x.e_2) : A_2}$$

$$\text{(A:If)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{q-K^{\mathsf{cond}}}{q'}} e_t : A \qquad \Sigma^r;\Gamma^r \vdash^{\frac{q-K^{\mathsf{cond}}}{q'}} e_f : A}{\Sigma^r;\Gamma^r,x:\mathsf{bool} \vdash^{\frac{q}{q'}} \mathrm{if}(x,e_t,e_f) : A} \qquad \text{(A:Pair)}\ \frac{}{\Sigma^r;x_1:A_1,x_2:A_2 \vdash^{\frac{K^{\mathsf{pair}}}{0}} \mathrm{pair}(x_1,x_2) : A_1 * A_2}$$

$$\text{(A:Match-P)}\ \frac{\Sigma^r;\Gamma^r,x_1:A_1,x_2:A_2 \vdash^{\frac{q-K^{\mathsf{matchP}}}{q'}} e : A}{\Sigma^r;\Gamma^r,x:A_1*A_2 \vdash^{\frac{q}{q'}} \mathrm{match}(x,(x_1,x_2).e) : A} \qquad \text{(A:Nil)}\ \frac{A \in \mathscr{A}}{\Sigma^r;\emptyset \vdash^{\frac{K^{\mathsf{nil}}}{0}} \mathsf{nil} : L^p(A)} \qquad \text{(A:Cons)}\ \frac{}{\Sigma^r;x_h:A,x_t:L^p(A) \vdash^{\frac{p+K^{\mathsf{cons}}}{0}} \mathrm{cons}(x_h,x_t) : L^p(A)}$$

$$\text{(A:Match-L)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{q-K^{\mathsf{matchN}}}{q'}} e_1 : A_1 \qquad \Sigma^r;\Gamma^r,x_h:A,x_t:L^p(A) \vdash^{\frac{q+p-K^{\mathsf{matchL}}}{q'}} e_2 : A_1}{\Sigma^r;\Gamma^r,x:L^p(A) \vdash^{\frac{q}{q'}} \mathrm{match}(x,e_1,(x_h,x_t).e_2) : A_1} \qquad \text{(A:Share)}\ \frac{\Sigma^r;\Gamma^r,x_1:A_1,x_2:A_2 \vdash^{\frac{q}{q'}} e : B \qquad \curlyvee(A \mid A_1,A_2)}{\Sigma^r;\Gamma^r,x:A \vdash^{\frac{q}{q'}} \mathrm{share}(x,(x_1,x_2).e) : B}$$

**Figure 4.** Common typing rules: upper bounds, constant, and lower bounds

$$\text{(U:Relax)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{p}{p'}} e : A \qquad q \geq p \qquad q-p \geq q'-p'}{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A} \qquad \text{(U:Weakening)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : B}{\Sigma^r;\Gamma^r,x:A \vdash^{\frac{q}{q'}} e : B} \qquad \text{(U:Subtype)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A \qquad B <: A}{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : B}$$

$$\text{(U:Supertype)}\ \frac{\Sigma^r;\Gamma^r,x:B \vdash^{\frac{q}{q'}} e : C \qquad B <: A}{\Sigma^r;\Gamma^r,x:A \vdash^{\frac{q}{q'}} e : C}$$

**Figure 5.** Relax and structural typing rules: upper bounds

$$\text{(C:Relax)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{p}{p'}} e : A \qquad q \geq p \qquad q-p = q'-p'}{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A} \qquad \text{(C:Weakening)}\ \frac{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : B \qquad \curlyvee(A \mid A,A)}{\Sigma^r;\Gamma^r,x:A \vdash^{\frac{q}{q'}} e : B}$$

**Figure 6.** Relax and structural typing rules: constant

**A:Unit** It is similar to the case A:Var.

**A:Bool** It is similar to the case A:Var.

**A:Int** It is similar to the case A:Var.

**A:B-Op** Assume that $e$ is an expression of the form $\mathrm{op}_\diamond(x_1,x_2)$, where $\diamond = \{\mathsf{and},\mathsf{or}\}$. Thus $\Sigma^r;x_1:\mathsf{bool},x_2:\mathsf{bool} \vdash^{\frac{K^{\mathsf{op}}}{0}} e : \mathsf{bool}$ and $\models E : \{x_1:\mathsf{bool},x_2:\mathsf{bool}\}$. We have $E \vdash^{\frac{K^{\mathsf{op}}}{0}} e \Downarrow v$, thus for all $p,r \in \mathbb{Q}_0^+$ such that $p = K^{\mathsf{op}} + r = K^{\mathsf{op}} + \Phi_E(x_1:\mathsf{bool},x_2:\mathsf{bool}) + r$, there exists $p' = \Phi(v:\mathsf{bool}) + r = r$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$.

**A:I-Op** It is similar to the case A:B-Op.

**A:IB-Op** It is similar to the case A:B-Op.

**A:Cons** If $e$ is of the form $\mathrm{cons}(x_1,x_2)$, then the type derivation ends with an application of the rule A:Cons and the evaluation

(L:RELAX)
$$\frac{\Sigma^r;\Gamma^r \vdash^{\frac{p}{p'}} e : A \qquad q \geq p \qquad q - p \leq q' - p'}{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A}$$

(L:WEAKENING)
$$\frac{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : B \qquad \curlyvee(A \mid A, A)}{\Sigma^r;\Gamma^r, x : A \vdash^{\frac{q}{q'}} e : B}$$

(L:SUBTYPE)
$$\frac{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A \qquad A <: B}{\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : B}$$

(L:SUPERTYPE)
$$\frac{\Sigma^r;\Gamma^r, x : B \vdash^{\frac{q}{q'}} e : C \qquad A <: B}{\Sigma^r;\Gamma^r, x : A \vdash^{\frac{q}{q'}} e : C}$$

(L:CONTRACTION)
$$\frac{\Sigma^r;\Gamma^r, x_1 : A, x_2 : A \vdash^{\frac{q}{q'}} e : B}{\Sigma^r;\Gamma^r, x : A \vdash^{\frac{q}{q'}} \mathsf{share}(x, (x_1, x_2).e) : B}$$

**Figure 7.** Relax and structural typing rules: lower bounds

ends with the application of the rule E:CONS. Thus $\Sigma^r; x_1 : A, x_2 : L^{p_1}(A) \vdash^{\frac{p_1 + K^{\mathsf{cons}}}{0}} e : L^{p_1}(A)$ and $\models E : \{x_1 : A, x_2 : L^{p_1}(A)\}$.

We have $E \vdash^{\frac{K^{\mathsf{cons}}}{0}} e \Downarrow [v_1, ..., v_n]$, where $E(x_1) = v_1$ and $E(x_2) = [v_2, \cdots, v_n]$. Let $\Gamma^r = x_h : A, x_t : L^{p_1}(A)$, for all $p, r \in \mathbb{Q}_0^+$ such that $p = p_1 + K^{\mathsf{cons}} + \Phi_E(\Gamma^r) + r$, there exists $p' \in \mathbb{Q}_0^+$ satisfying $p' = \Phi([v_1, ..., v_n] : L^{p_1}(A)) + r = \Phi_E(\Gamma^r) + p_1 + r$ and $E \vdash^{\frac{p}{p'}} e \Downarrow [v_1, ..., v_n]$.

**A:PAIR** It is similar to the case A:CONS.

**A:NIL** It is similar to the case A:CONS.

**A:MATCH-P** Suppose that the typing derivation $\Sigma^r;\Gamma^r, x : A_1 * A_2 \vdash^{\frac{q}{q'}} \mathsf{match}(x, (x_1, x_2).e) : A$ ends with an application of the rule A:MATCH-P. Thus $\Sigma^r;\Gamma^r, x_1 : A_1, x_2 : A_2 \vdash^{\frac{q - K^{\mathsf{matchP}}}{q'}} e : A$ and $\models E : \Gamma^r, x : A_1 * A_2$.

Let $E_1 = E[x_1 \mapsto v_1, x_2 \mapsto v_2]$ and $\Gamma_1^r = \Gamma^r, x_1 : A_1, x_2 : A_2$. Since $\models v_1 : A_1$, $\models v_2 : A_2$, and $\models E : \Gamma^r$ it holds that $\models E_1 : \Gamma_1^r$. For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma^r, x : A_1 * A_2) + r$, thus $p - K^{\mathsf{matchP}} = q - K^{\mathsf{matchP}} + \Phi_{E_1}(\Gamma_1^r) + r$, by the induction hypothesis for $e$, there exists $p' \in \mathbb{Q}_0^+$ satisfying $p' = q' + \Phi(v : A) + r$ and $E_1 \vdash^{\frac{p - K^{\mathsf{matchP}}}{p'}} e \Downarrow v$. Hence, by the rule E:MATCH-P, there exists $p' = q' + \Phi(v : A) + r$ satisfying $E \vdash^{\frac{p}{p'}} \mathsf{match}(x, (x_1, x_2).e) \Downarrow v$.

**A:FUN** Assume that $e$ is a function application of the form $\mathsf{app}(f, x)$. Thus $\Sigma^r; x : A_1 \vdash^{\frac{q + K^{\mathsf{app}}}{q'}} e : A_2$ and $\Sigma^r(f) = A_1 \xrightarrow{q/q'} A_2$. Because the considering program is well-formed, there exists a well-typed expression $e_f$ under the typing context $\Gamma_1^r = y^{\widehat{f}} : A_1$ and the signature $\Sigma^r$, or $\Sigma^r;\Gamma_1^r \vdash^{\frac{q}{q'}} e_f : A_2$.

Let $\Gamma^r = x : A_1$, $E(x) = v_1$ and $E_1 = [y^{\widehat{f}} \mapsto v_1]$, since $\models E : \Gamma^r$, it follows that $\models E_1 : \Gamma_1^r$. For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + K^{\mathsf{app}} + \Phi_E(\Gamma^r) + r$, since $\Phi_{E_1}(\Gamma_1^r) = \Phi(E_1(y^{\widehat{f}}) : A_1) = \Phi_E(\Gamma^r) = \Phi(E(x) : A_1)$, it holds that $p - K^{\mathsf{app}} = q + \Phi_{E_1}(\Gamma_1^r) + r$. By the induction hypothesis for $e_f$, there exists $p' \in \mathbb{Q}_0^+$ satisfying $p' = q' + \Phi(v : A_2) + r$ and $E_1 \vdash^{\frac{p_1}{p_1'}} e_{\widehat{f}} \Downarrow v$. Hence, $E \vdash^{\frac{p}{p'}} e \Downarrow v$.

**A:IF** Suppose that $e$ is an expression of the form $\mathsf{if}(x, e_t, e_f)$. Then one of the rules E:IF-TRUE and E:IF-FALSE has been applied in the evaluation derivation depending on the value of $x$.

Assume that the variable $x$ is assigned the value $\mathsf{true}$ in $E$, or $E(x) = \mathsf{true}$. The typing rule for $e$ has been derived by an application of the rule A:IF using the premise on the left thus $\Sigma^r;\Gamma^r \vdash^{\frac{q - K^{\mathsf{cond}}}{q'}} e_t : A$.

Let $\Gamma_1^r = \Gamma^r, x : \mathsf{bool}$, since $\models E : \Gamma_1^r$, it follows that $\models E : \Gamma^r$. For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma_1^r) + r$, since $\Phi_E(\Gamma^r) = \Phi_E(\Gamma_1^r)$ thus $p_1 = p - K^{\mathsf{cond}} = q - K^{\mathsf{cond}} + \Phi_E(\Gamma^r) + r$. By the induction hypothesis for $e_t$, there exists $p_1' \in \mathbb{Q}_0^+$ satisfying $E \vdash^{\frac{p_1}{p_1'}} e_t \Downarrow v$ and $p_1' = q' + \Phi(v : A)$. Hence, by the rule E:IF-TRUE, there exists $p' = p_1'$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q' + \Phi(v : A)$. If $x$ is assigned the value $\mathsf{false}$ in $E$ then it is similar to the case $E(x) = \mathsf{true}$.

**A:MATCH-L** It is the same as the case of a conditional expression. The evaluation derivation applies one of the rules E:MATCH-N and E:MATCH-L depending on the value of $x$.

Assume that $x$ is assigned the value $[v_1, ...., v_n]$ under $E$, or $E(x) = [v_1, ..., v_n]$. Then, the evaluation derivation ends with an application of the rule E:MATCH-L. Let $E_1 = E[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]]$ and $\Gamma_1^r = \Gamma^r, x_h : A, x_t : L^{p_1}(A)$, the typing derivation ends with an application of the rule A:MATCH-L, thus $\Sigma^r;\Gamma_1^r \vdash^{\frac{q + p_1 - K^{\mathsf{matchL}}}{q'}} e_2 : A_1$.

Since $\models [v_1, ..., v_n] : L^{p_1}(A)$, we have $\models v_i : A, \forall i = 1, ..., n$. Hence, it holds that $\models v_1 : A$ and $\models [v_2, ..., v_n] : L^{p_1}(A)$. Finally, we have $\models E_1 : \Gamma_1^r$ (since $\models E : \Gamma^r$ implies $\models E_1 : \Gamma^r$).

For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma^r, x : L^{p_1}(A)) + r$, because $\Phi_E(\Gamma^r, x : L^{p_1}(A)) = \Phi_E(\Gamma^r) + n.p_1 + \Sigma_{i=1}^n \Phi(v_i : A)$, $\Phi_{E_1}(\Gamma_1^r) = \Phi_{E_1}(\Gamma^r) + (n-1).p_1 + \Sigma_{i=1}^n \Phi(v_i : A)$ and $\Phi_{E_1}(\Gamma^r) = \Phi_E(\Gamma^r)$, thus we have $\Phi_{E_1}(\Gamma_1^r) = \Phi_E(\Gamma^r, x : L^{p_1}(A)) - p_1$. Thus $p_2 = p - K^{\mathsf{matchL}} = q + p_1 - K^{\mathsf{matchL}} + \Phi_{E_1}(\Gamma_1^r) + r$. By the induction hypothesis for $e_2$, there exists $p_2' \in \mathbb{Q}_0^+$ satisfying $E_1 \vdash^{\frac{p_2}{p_2'}} e_2 \Downarrow v$ and $p_2' = q' + \Phi(v : A_1)$. Hence, there exists $p' = p_2'$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$. If $E(x) = \mathsf{nil}$ then it is similar to the case A:MATCH-P.

**A:LET** Assume that $e$ is an expression of the form $\mathsf{let}(x, e_1, x.e_2)$. Hence, the evaluation derivation ends with an application of the rule E:LET. Let $E_1 = E[x \mapsto v_1]$ and $\Gamma^r = \Gamma_1^r, \Gamma_2^r$. The typing derivation ends with an application of the rule A:LET, thus $\Sigma^r;\Gamma_1^r \vdash^{\frac{q - K^{\mathsf{let}}}{q_1'}} e_1 : A_1$ and $\Sigma^r;\Gamma_2^r, x : A_1 \vdash^{\frac{q_1'}{q'}} e_2 : A_2$.

For all $p, r \in \mathbb{Q}_0^+$ such that $p = q + \Phi_E(\Gamma^r) + r$, thus $p_1 = p - K^{\mathsf{let}} = q - K^{\mathsf{let}} + \Phi_E(\Gamma_1^r) + \Phi_E(\Gamma_2^r) + r$. Since $\models E : \Gamma^r$, we have $\models E : \Gamma_1^r$. By the induction hypothesis for $e_1$, there exists $p_1' \in \mathbb{Q}_0^+$ satisfying $E \vdash^{\frac{p_1}{p_1'}} e_1 \Downarrow v_1$ and $p_1' = q_1' + \Phi(v_1 : A_1) + \Phi_E(\Gamma_2^r) + r$.

We have $\models E : \Gamma_2^r$, thus $\models E_1 : \Gamma_2^r, x : A_1$. Again by the induction hypothesis for $e_2$, with $p_2 = p - K^{\mathsf{let}} - (p_1 - p_1') = p_1' = q_1' + \Phi_{E_1}(\Gamma_2^r, x : A_1) + r$, there exists $p_2' \in \mathbb{Q}_0^+$ satisfying $E_1 \vdash^{\frac{p_2}{p_2'}} e_2 \Downarrow v$ and $p_2' = q' + \Phi(v : A_2) + r$. Hence, by the rule E:LET, there exists $p' = p_2'$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' = q' + \Phi(v : A_2)$. □

**Theorem 2.** *If* $\models E : \Gamma^r$, $E \vdash e \Downarrow v$, $\Sigma^r;\Gamma^r \vdash^{\frac{q}{q'}} e : A$, $\curlyvee(A \mid A, A)$, *and* $\forall x \in dom(\Gamma^r) \setminus X. \curlyvee(\Gamma^r(x) \mid \Gamma^r(x), \Gamma^r(x))$ *then* $e$ *is constant resource w.r.t* $X \subseteq dom(\Gamma^r)$.

*Proof.* First, we prove that if $E \vdash^{\frac{p}{p'}} e \Downarrow v$ then $p - p' = q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A))$. Suppose $p - p' \neq q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A))$, there exists always some $r_1, r_2 \in \mathbb{Q}_0^+$ such that $p + r_1 = q + \Phi_E(\Gamma^r) + r_2$. Since $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $E \vdash^{\frac{p+r_1}{p'+r_1}} e \Downarrow v$. By Theorem 1, $p' + r_1 = q' + \Phi(v : A) + r_2$, thus the assumption is contradictory.

Consider any $E_1$ and $E_2$ such that $E_1 \approx_X E_2$, hence $E_1 \vdash e \Downarrow v_1$ and $E_2 \vdash e \Downarrow v_2$. For all $p_1, p_1' \in \mathbb{Q}_0^+$ such that $E_1 \vdash^{\frac{p_1}{p_1'}} e \Downarrow v_1$, we have $p_1 - p_1' = q + \Phi_{E_1}(\Gamma^r) - (q' + \Phi(v_1 : A))$. Similarly, for all $p_2, p_2' \in \mathbb{Q}_0^+$ such that $E_2 \vdash^{\frac{p_2}{p_2'}} e \Downarrow v_2$, $p_2 - p_2' = q + \Phi_{E_2}(\Gamma^r) - (q' + \Phi(v_2 : A))$. Since $\Phi_{E_1}(X) = \Phi_{E_2}(X)$ by Lemma 1, $\forall x \in \mathrm{dom}(\Gamma^r) \setminus X.\Phi(E_i(x) : \Gamma^r(x)) = 0$, and $\Phi(v_i : A) = 0$, $i = 1, 2$. Thus $p_1 - p_1' = p_2 - p_2'$. □

**Theorem 3.** *If $\models E : \Gamma^r$, $E \vdash e \Downarrow v$, and $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : A$, then for all $p, r \in \mathbb{Q}_0^+$ such that $p < q + \Phi_E(\Gamma^r) + r$, there exists no $p' \in \mathbb{Q}_0^+$ satisfying $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' \geq q' + \Phi(v : A) + r$.*

*Proof.* The proof is relied on Theorem 4. For all $p, r \in \mathbb{Q}_0^+$ such that $p < q + \Phi_E(\Gamma^r) + r$, assume that there exists some $p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and $p' \geq q' + \Phi(v : A) + r$. Thus we have $p - p' < q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A))$.

On the other hand, it holds that $q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A)) \leq p - p'$. The assumption is contradictory. □

**Theorem 4.** *If $\models E : \Gamma^r$, $E \vdash e \Downarrow v$, and $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : A$, then for all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, it satisfies $q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A)) \leq p - p'$.*

*Proof.* The proof is done by induction on the length of the derivation of the evaluation judgment $E \vdash^{\frac{p}{p'}} e \Downarrow v$ and the typing judgment $\Sigma; \Gamma \vdash^{\frac{q}{q'}} e : A$ with lexical order, in which the derivation of the evaluation judgment takes priority over the typing derivation. We need to do induction on the length of both evaluation and typing derivations since on one hand, an induction of only typing derivation would fail for the case of function application, which increases the length of the typing derivation, while the length of the evaluation derivation never increases. On the other hand, if the rules L:Weakening and A:Share are final step in the derivation, then the length of typing derivation decreases, while the length of evaluation derivation is unchanged.

**A:Share** Assume that the typing derivation ends with an application of the rule A:Share, thus $\Sigma^r; \Gamma^r, x_1 : A_1, x_2 : A_2 \vdash^{\frac{q}{q'}} e : B$ and $\curlyvee(A | A_1, A_2)$. Let $E_1 = E \setminus \{x\} \cup \{[x_1 \mapsto E(x), x_2 \mapsto E(x)]\}$. Since $\models E : \Gamma^r, x : A$ and following the property of the share relation we have $\models E_1 : \Gamma^r, x_1 : A_1, x_2 : A_2$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} \mathrm{share}(x, (x_1, x_2).e \Downarrow v$, by the rule E:Share we have $E_1 \vdash^{\frac{p}{p'}} e \Downarrow v$. Hence, by the induction hypothesis for $e$ in the premise, it holds that $q + \Phi_{E_1}(\Gamma^r, x_1 : A_1, x_2 : A_2) - (q' + \Phi(v : B)) \leq p - p'$.

Because $\Phi(E(x) : A) = \Phi(E_1(x_1) : A_1) + \Phi(E_1(x_2) : A_2)$ and $\Phi_E(\Gamma^r) = \Phi_{E_1}(\Gamma^r) = \Phi_{E \setminus \{x\}}(\Gamma^r)$, we have $q + \Phi_E(\Gamma^r, x : A) - (q' + \Phi(v : B)) \leq p - p'$.

**L:Weakening** Suppose that the typing derivation $\Sigma^r; \Gamma^r, x : A \vdash^{\frac{q}{q'}} e : B$ ends with an application of the rule L:Weakening. Thus we have $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : B$, in which the data type $A$ satisfies $\curlyvee(A \mid A, A)$. Since $\models E : \Gamma^r, x : A$, it follows that $\models E : \Gamma^r$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, by the induction hypothesis for $e$ in the premise, it holds that $q + \Phi_E(\Gamma^r) - (q' + \Phi(v : B)) \leq p - p'$. By the property of the share relation, $\Phi(a : A) = 0$, hence we have $q + \Phi_E(\Gamma^r, x : A) - (q' + \Phi(v : B)) \leq p - p'$.

**L:Relax** Suppose that the typing derivation ends with an application of the rule L:Relax, thus we have $\Sigma^r; \Gamma^r \vdash^{\frac{q_1}{q_1'}} e : A$, $q \geq q_1$, and $q - q_1 \leq q' - q_1'$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $\models E : \Gamma^r$, hence by the induction hypothesis for $e$ in the premise, it holds that $q_1 + \Phi_E(\Gamma^r) - (q_1' + \Phi(v : A)) \leq p - p'$. We have $q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A)) = q_1 + \Phi_E(\Gamma^r) - (q_1' + \Phi(v : A)) + ((q - q_1) - (q' - q_1'))$. Since $q - q_1 \leq q' - q_1'$, it holds that $q + \Phi_E(\Gamma^r) - (q' + \Phi(v : A)) \leq q_1 + \Phi_E(\Gamma^r) - (q_1' + \Phi(v : A)) \leq p - p'$.

**A:Var** Assume that $e$ is a variable $x$. If $\Sigma^r; x : A \vdash^{\frac{K^{\mathsf{var}}}{0}} x : A$. Thus for all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $p = p' + K^{\mathsf{var}}$, hence $K^{\mathsf{var}} + \Phi(E(x) : A) - \Phi(v : A) \leq p - p' = K^{\mathsf{var}}$.

**A:Unit** It is similar to the case A:Var.

**A:Bool** It is similar to the case A:Var.

**A:Int** It is similar to the case A:Var.

**A:B-Op** Assume that $e$ is an expression of the form $\mathrm{op}_\diamond(x_1, x_2)$, where $\diamond = \{\mathsf{and}, \mathsf{or}\}$. Thus $\Sigma^r; x_1 : \mathrm{bool}, x_2 : \mathrm{bool} \vdash^{\frac{K^{\mathsf{op}}}{0}} e : \mathrm{bool}$ and $\models E : \{x_1 : \mathrm{bool}, x_2 : \mathrm{bool}\}$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $K^{\mathsf{op}} + \Phi_E(x_1 : \mathrm{bool}, x_2 : \mathrm{bool}) - \Phi(v : \mathrm{bool}) = K^{\mathsf{op}} \leq p - p' = K^{\mathsf{op}}$.

**A:IB-Op** It is similar to the case A:B-Op.

**A:I-Op** It is similar to the case A:B-Op.

**A:Cons** If $e$ is of the form $\mathrm{cons}(x_1, x_2)$, then the typing derivation ends with an application of the rule A:Cons and the evaluation derivation ends with the application of the rule E:Cons. Thus $\Sigma^r; x_1 : A, x_2 : L^{p_1}(A) \vdash^{\frac{p_1 + K^{\mathsf{cons}}}{0}} e : L^{p_1}(A)$ and $\models E : \{x_1 : A, x_2 : L^{p_1}(A)\}$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $p - p' = K^{\mathsf{cons}}$, $E(x_1) = v_1$ and $E(x_2) = [v_2, \cdots, v_n]$. Let $\Gamma^r = x_h : A, x_t : L^{p_1}(A)$, it holds that $p_1 + K^{\mathsf{cons}} + \Phi_E(\Gamma^r) - (\Phi([v_1, ..., v_n] : L^{p_1}(A))) = K^{\mathsf{cons}} \leq p - p'$.

**A:Pair** It is similar to the case A:Cons.

**A:Nil** It is similar to the case A:Cons.

**A:Match-P** Suppose that the typing derivation $\Sigma^r; \Gamma^r, x : A_1 * A_2 \vdash^{\frac{q}{q'}} \mathrm{match}(x, (x_1, x_2).e) : A$ ends with an application of the rule A:Match-P. Thus $\Sigma^r; \Gamma^r, x_1 : A_1, x_2 : A_2 \vdash^{\frac{q - K^{\mathsf{matchP}}}{q'}} e : A$ and $\models E : \Gamma, x : A_1 * A_2$.

Let $E_1 = E[x_1 \mapsto v_1, x_2 \mapsto v_2]$ and $\Gamma_1^r = \Gamma^r, x_1 : A_1, x_2 : A_2$, since $\models v_1 : A_1$, $\models v_2 : A_2$, and $\models E : \Gamma^r$ it holds that $\models E_1 : \Gamma_1^r$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, by the rule E:Match-P we have $E_1 \vdash^{\frac{p - K^{\mathsf{matchP}}}{p'}} e \Downarrow v$. Hence, by the induction hypothesis for $e$ in the premise, it holds that $q - K^{\mathsf{matchP}} + \Phi_{E_1}(\Gamma_1^r) - (q' + \Phi(v : A)) \leq p - K^{\mathsf{matchP}} - p'$.

Since $\Phi_E(\Gamma^r, x : A_1 * A_2) = \Phi_{E_1}(\Gamma_1^r)$, it follows that $q + \Phi_E(\Gamma^r, x : A_1 * A_2) - (q' + \Phi(v : A)) \leq p - p'$.

**A:FUN** Assume that $e$ is a function application of the form $\mathrm{app}(f,x)$. Thus $\Sigma^r; x:A_1 \vdash^{\frac{q+K^{\mathsf{app}}}{q'}} e:A_2$ and $\Sigma^r(f) = A_1 \xrightarrow{q/q'} A_2$. Because the considering program is well-formed, there exists a well-typed expression $e_f$ under the typing context $\Gamma_1^r = y^{\widehat{f}} : A_1$ and the signature $\Sigma^r$, or $\Sigma^r; \Gamma_1^r \vdash^{\frac{q}{q'}} e_f : A_2$.

Let $\Gamma^r = x:A_1$, $E(x) = v_1$ and $E_1 = [y^{\widehat{f}} \mapsto v_1]$, since $\models E:\Gamma^r$, it follows that $\models E_1 : \Gamma_1^r$. For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, we have $E_1 \vdash^{\frac{p-K^{\mathsf{app}}}{p'}} e_{\widehat{f}} \Downarrow v$. Hence, by the induction hypothesis for $e_f$, it holds that $q + \Phi_{E_1}(\Gamma_1^r) - (q' + \Phi(v:A_2)) \le p - K^{\mathsf{app}} - p'$.

Since $\Phi_{E_1}(\Gamma_1^r) = \Phi(E_1(y^{\widehat{f}}) : A_1) = \Phi_E(\Gamma^r) = \Phi(E(x):A_1)$, it follows that $q + K^{\mathsf{app}} + \Phi(E(x):A_1) - (q' + \Phi(v:A_2)) \le p - p'$.

**A:IF** Suppose that $e$ is an expression of the form $\mathrm{if}(x, e_t, e_f)$. Then one of the rules E:IF-TRUE and E:IF-FALSE has been applied in the evaluation derivation depending on the value of $x$.

Assume that the variable $x$ is assigned the value $\mathsf{true}$ in $E$, or $E(x) = \mathsf{true}$. The typing rule for $e$ has been derived by an application of the rule A:IF using the premise on the left thus $\Sigma^r; \Gamma^r \vdash^{\frac{q-K^{\mathsf{cond}}}{q'}} e_t : A$. Let $\Gamma_1^r = \Gamma^r, x : \mathrm{bool}$, since $\models E : \Gamma_1^r$, it follows that $\models E : \Gamma^r$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, by the rule E:IF-TRUE we have $E \vdash^{\frac{p-K^{\mathsf{cond}}}{p'}} e_t \Downarrow v$. Hence, by the induction hypothesis for $e_t$, it holds that $q - K^{\mathsf{cond}} + \Phi_E(\Gamma^r) - (q' + \Phi(v:A)) \le p - K^{\mathsf{cond}} - p'$.

Because $\Phi_E(\Gamma^r) = \Phi_E(\Gamma_1^r)$, it follows $q + \Phi_E(\Gamma_1^r) - (q' + \Phi(v:A)) \le p - p'$. If $E(x) = \mathsf{false}$ then the proof is similar.

**A:MATCH-L** It is the same as the case of a conditional expression. The evaluation derivation applies one of the rules E:MATCH-N and E:MATCH-L depending on the value of $x$.

Assume that $x$ is assigned the value $[v_1, ...., v_n]$ under $E$, or $E(x) = [v_1, ..., v_n]$. Then, the evaluation derivation ends with an application of the rule E:MATCH-L. Let $E_1 = E[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]]$ and $\Gamma_1^r = \Gamma^r, x_h : A, x_t : L^{p_1}(A)$, the typing derivation ends with an application of the rule A:MATCH-L, thus $\Sigma^r; \Gamma_1^r \vdash^{\frac{q+p_1-K^{\mathsf{matchL}}}{q'}} e_2 : A_1$.

Since $\models [v_1, ..., v_n] : L^{p_1}(A)$, we have $\models v_i : A, \forall i = 1, ..., n$. Hence, it holds that $\models v_1 : A$ and $\models [v_2, ..., v_n] : L^{p_1}(A)$. Finally, we have $\models E_1 : \Gamma_1^r$ (since $\models E : \Gamma^r$ implies $\models E_1 : \Gamma^r$).

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, by the rule E:MATCH-L we have $E_1 \vdash^{\frac{p-K^{\mathsf{matchL}}}{p'}} e_2 \Downarrow v$. By the induction hypothesis for $e_2$, it holds that $q + p_1 - K^{\mathsf{matchL}} + \Phi_{E_1}(\Gamma_1^r) - (q' + \Phi(v:A_1)) \le p - K^{\mathsf{matchL}} - p'$.

Because $\Phi_E(\Gamma^r, x : L^p(A)) = \Phi_E(\Gamma^r) + n.p_1 + \Sigma_{i=1}^n \Phi(v_i : A)$, $\Phi_{E_1}(\Gamma_1^r) = \Phi_{E_1}(\Gamma^r) + (n-1).p_1 + \Sigma_{i=1}^n \Phi(v_i : A)$ and $\Phi_{E_1}(\Gamma^r) = \Phi_E(\Gamma^r)$, thus we have $\Phi_{E_1}(\Gamma_1^r) = \Phi_E(\Gamma^r, x : L^{p_1}(A)) - p_1$. Therefore, $q + \Phi_E(\Gamma^r, x : L^{p_1}(A)) - (q' + \Phi(v:A_1)) \le p - p'$. If $E(x) = \mathsf{nil}$ then it is similar to the case A:MATCH-P.

**A:LET** Assume that $e$ is an expression of the form $\mathrm{let}(x, e_1, x.e_2)$. Hence, the evaluation derivation ends with an application of the rule E:LET. Let $E_1 = E[x \mapsto v_1]$ and $\Gamma^r = \Gamma_1^r, \Gamma_2^r$. The typing derivation ends with an application of the rule A:LET, thus $\Sigma^r; \Gamma_1^r \vdash^{\frac{q-K^{\mathsf{let}}}{q_1'}} e_1 : A_1$ and $\Sigma^r; \Gamma_2^r, x : A_1 \vdash^{\frac{q_1'}{q'}} e_2 : A_2$.

For all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$, by the rule E:LET we have $E \vdash^{\frac{p-K^{\mathsf{let}}}{p_1'}} e_1 \Downarrow v_1$ and $E_1 \vdash^{\frac{p_1'}{p'}} e_2 \Downarrow v$. Since $\models E : \Gamma^r$, we have $\models E : \Gamma_1^r$. By the induction hypothesis for $e_1$, it holds that $q - K^{\mathsf{let}} + \Phi_E(\Gamma_1^r) - (q_1' + \Phi(v_1 : A_1)) \le p - K^{\mathsf{let}} - p_1'$.

We have $\models E : \Gamma_2^r$, thus $\models E_1 : \Gamma_2^r, x : A_1$. Again by the induction hypothesis for $e_2$, we derive that $q_1' + \Phi_{E_1}(\Gamma_2^r, x : A_1) - (q' + \Phi(v : A_2)) \le p_1' - p'$.

Sum two in-equations above, it follows $q + \Phi_E(\Gamma_1^r) - \Phi(v_1 : A_1) + \Phi_{E_1}(\Gamma_2^r, x : A_1) - (q' + \Phi(v : A_2)) = q + \Phi_E(\Gamma_1^r, \Gamma_2^r) - (q' + \Phi(v : A_2)) \le p - p'$.

**L:SUBTYPE** Assume that the typing derivation ends with an application of the rule L:SUBTYPE, thus $\Sigma^r; \Gamma^r \vdash^{\frac{q}{q'}} e : A$ and $A <: B$.

By the induction hypothesis for $e$ in the premise, for all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$ it holds that $q + \Phi_E(\Gamma^r) - (q' + \Phi(v:A)) \le p - p'$.

Because $\Phi(E(x):A) \le \Phi(E(x):B)$ we have $q + \Phi_E(\Gamma^r) - (q' + \Phi(v:B)) \le p - p'$.

**L:SUPERTYPE** Assume that the typing derivation ends with an application of the rule L:SUPERTYPE, thus $\Sigma^r; \Gamma^r, x : B \vdash^{\frac{q}{q'}} e : C$ and $A <: B$. Since $\models E : \Gamma^r, x : A$ and following the property of the subtyping relation we have $\models E : \Gamma^r, x : B$.

By the induction hypothesis for $e$ in the premise, for all $p, p' \in \mathbb{Q}_0^+$ such that $E \vdash^{\frac{p}{p'}} e \Downarrow v$ it holds that $q + \Phi_E(\Gamma^r, x:B) - (q' + \Phi(v : C)) \le p - p'$.

Because $\Phi(E(x):A) \le \Phi(E(x):B)$ we have $q + \Phi_E(\Gamma^r, x:A) - (q' + \Phi(v:C)) \le p - p'$. □

## 3. Type system for information flow and constant resource

The full typing rules of the type system for information flow and constant resource are presented in Fig. 9.

**Lemma 2.** *Suppose* $pc; \Sigma^s; \Gamma^s \vdash e : S$ *or* $pc; \Sigma^s; \Gamma^s \vdash^{const} e : S$. *For all variables* $x$ *in* $e$, *if* $S \blacktriangleleft k_1$ *then* $\Gamma^s(x) \blacktriangleleft k_1$.

*Proof.* By induction on the structure of the typing derivation.

**SR:UNIT** There is no variable thus it follows immediately.

**SR:BOOL** It is similar to the case SR:UNIT.

**SR:INT** It is similar to the case SR:UNIT.

**SR:VAR** Since $\Gamma^s(x) = S$, if $S \blacktriangleleft k_1$ then $\Gamma^s(x) \blacktriangleleft k_1$.

**SR:B-OP** If $(\mathrm{bool}, k_{x_1} \sqcup k_{x_2}) \blacktriangleleft k_1$ then $\Gamma^s(x_1) = (\mathrm{bool}, k_{x_1}) \blacktriangleleft k_1$ and $\Gamma^s(x_2) = (\mathrm{bool}, k_{x_2}) \blacktriangleleft k_1$.

**SR:IB-OP** It is similar to the case SR:B-OP.

**SR:I-OP** It is similar to the case SR:B-OP.

**SR:GEN - SR:C-GEN** By induction for $e$ in the premise, it follows.

**SR:FUN** Because $e$ is well-formed program, there exists a well-typed expression $e_f$ such that $\mathrm{pc}'; \Sigma^s; \Gamma^s \vdash e_f : S_2$. By induction for $e_f$, for all variables $x$ in $e$, if $S \blacktriangleleft k_1$ then $\Gamma^s(x) \blacktriangleleft k_1$. It is similar for SR:L-ARG and SR:C-FUN.

**S:LET** If $S_2 \blacktriangleleft k_1$ then by induction for $e_2$, $S_1 \blacktriangleleft k_1$. Thus for all variable $x$ in $e$, it is a variable in $e_1$ or $e_2$. By induction for $e_1$ and $e_2$, it follows. It is similar for SR:L-LET.

**SR:IF** If $S \blacktriangleleft k_1$ then by the hypothesis $(\mathrm{bool}, k_x) \blacktriangleleft k_1$. For all variable $y$ in $e$, it is a variable in $e_t$ or $e_f$. By induction for $e_t$ and $e_f$, it follows. It is similar for SR:L-IF.

$$\frac{k \sqsubseteq k' \qquad T \in \{\text{unit}, \text{int}, \text{bool}\}}{k \lhd (T, k')} \qquad \frac{k \sqsubseteq k' \qquad k \lhd S}{k \lhd (L(S), k')} \qquad \frac{k \lhd S_1 \qquad k \lhd S_2}{k \lhd S_1 * S_2} \qquad \frac{k' \sqsubseteq k \qquad T \in \{\text{unit}, \text{int}, \text{bool}\}}{(T, k') \blacktriangleleft k} \qquad \frac{k' \sqsubseteq k \qquad S \blacktriangleleft k}{(L(S), k') \blacktriangleleft k}$$

$$\frac{S_1 \blacktriangleleft k \qquad S_2 \blacktriangleleft k}{S_1 * S_2 \blacktriangleleft k} \qquad \frac{k \sqsubseteq k' \qquad T \in \{\text{unit}, \text{int}, \text{bool}\}}{(\mathrm{T}, k) \leq (\mathrm{T}, k')} \qquad \frac{k \sqsubseteq k' \qquad S \leq S'}{(L(S), k) \leq (L(S'), k')} \qquad \frac{S_1 \leq S_1' \qquad S_2 \leq S_2'}{S_1 * S_2 \leq S_1' * S_2'}$$

**Figure 8.** Guards, collecting, and subtyping relations

**SR:PAIR** If $S_1 * S_2 \blacktriangleleft k_1$ then $\Gamma^s(x_1) = S_1 \blacktriangleleft k_1$ and $\Gamma^s(x_2) = S_2 \blacktriangleleft k_1$.

**SR:MATCH-P** If $S \blacktriangleleft k_1$ then by induction for $e$, $\Gamma^s(x_1) \blacktriangleleft k_1$ and $\Gamma^s(x_2) \blacktriangleleft k_1$. Thus $\Gamma^s(x) \blacktriangleleft k_1$. For all other variables $y$ in $e$, again by induction for $e$, if $S \blacktriangleleft k_1$ then $\Gamma^s(y) \blacktriangleleft k_1$. It is similar for SR:C-MATCH-P.

**SR:NIL** It is similar to the case SR:UNIT.

**SR:CONS** If $(L(S), k_x) \blacktriangleleft k_1$ then $\Gamma^s(x_h) = S \blacktriangleleft k_1$ and $\Gamma^s(x_t) = (L(S), k_x) \blacktriangleleft k_1$.

**SR:MATCH-L** If $S_1 \blacktriangleleft k_1$ then by induction for $e_2$, $\Gamma^s(x_h) = S \blacktriangleleft k_1$ and $\Gamma^s(x_t) = (L(S), k_x) \blacktriangleleft k_1$. Thus $\Gamma^s(x) \blacktriangleleft k_1$. For all other variables $y$ in $e$, $y$ is a variable in $e_1$ or $e_2$. Again by induction for $e_1$ and $e_2$, if $S_1 \blacktriangleleft k_1$ then $\Gamma^s(y) \blacktriangleleft k_1$.

**SR:SUBTYPING** By the subtyping relation, if $S' \blacktriangleleft k_1$ then $S \blacktriangleleft k_1$. Thus by induction for $e$ in the premise, for all variables $x$ in $e$, if $S \blacktriangleleft k_1$ then $\Gamma^s(x) \blacktriangleleft k_1$. It is similar for SR:C-SUBTYPING. □

**Lemma 3.** *Suppose* $pc; \Sigma^s; \Gamma^s \vdash e : S$ *or* $pc; \Sigma^s; \Gamma^s \overset{const}{\nvdash} e : S$, $E_1 \vdash e \Downarrow v_1$, $E_2 \vdash e \Downarrow v_2$, *and* $E_1 \equiv_{k_1} E_2$. *Then* $v_1 = v_2$ *if* $S \blacktriangleleft k_1$.

*Proof.* The proof is done by induction on the structure of the evaluation derivation and the typing derivation.

**SR:UNIT** Suppose the evaluation derivation of $e$ ends with an application of the rule E:UNIT, thus $E_1 \vdash e \Downarrow ()$ and $E_2 \vdash e \Downarrow ()$. Hence, it follows.

**SR:BOOL** It is similar to the case SR:UNIT.

**SR:INT** It is similar to the case SR:UNIT.

**SR:VAR** Suppose the evaluation derivation ends with an application of the rule E:VAR, thus $E_1(x) = v_1$ and $E_2(x) = v_2$. The typing derivation ends with an application of the rule SR:VAR, thus $\Gamma^s(x) = S$. If $S \blacktriangleleft k_1$, by the hypothesis $E_1(x) = E_2(x)$ since $x \in \mathrm{dom}(E_i), i = \{1, 2\}$.

**SR:B-OP** Suppose the evaluation derivation ends with an application of the rule E:BIN, thus $E_1(x_1) \diamond E_1(x_2) = v_1$ and $E_2(x_1) \diamond E_2(x_2) = v_2$. The typing derivation ends with an application of the rules SR:B-OP or SR:GEN. We have $k_{x_1} \lhd S$ and $k_{x_2} \lhd S$. If $S \blacktriangleleft k_1$ then $k_{x_1} \sqsubseteq k_1$ and $k_{x_2} \sqsubseteq k_1$. By the hypothesis, we have $E_1(x_1) = E_2(x_1)$ and $E_1(x_2) = E_2(x_2)$, thus $v_1 = v_2$.

**SR:IB-OP** It is similar to the case SR:B-OP.

**SR:I-OP** It is similar to the case SR:B-OP.

**SR:GEN-SR:C-GEN** By induction for $e$ in the premise, it follows that if $S \blacktriangleleft k_1$ then $v_1 = v_2$.

**SR:FUN** Suppose the evaluation derivation ends with an application of the rule E:FUN, thus $\Sigma(g) = T_1 \to T_2$ and $[y^g \mapsto E_i(x)] \vdash e_g \Downarrow v_i$ for $i = \{1, 2\}$. The typing derivation ends with an application of the following rules.

- Case SR:FUN. Because $e$ is well-formed program, there exists a well-typed expression $e_f$ such that pc'; $\Sigma^s; \Gamma^s \vdash e_f : S_2$ and $\widehat{e_f} = e_g$. By induction for $e_f$, if $S_2 \blacktriangleleft k_1$ then $v_1 = v_2$.
- Case SR:L-ARG. It is similar to the case SR:FUN.
- Case SR:C-FUN. It is similar to the case SR:FUN.
- Case SR:GEN and SR:C-GEN. It follows.

**S:LET** Suppose the evaluation derivation ends with an application of the rule E:LET, thus $E_i \vdash e_1 \Downarrow v_1^i$ and $E_i[x \mapsto v_1^i] \vdash e_2 \Downarrow v_i$ for $i = \{1, 2\}$. The typing derivation ends with an application of the following rules.

- Case SR:L-LET. If $S_2 \blacktriangleleft k_1$, by the simple security lemma, it holds that $S_1 \blacktriangleleft k_1$. By induction for $e_1$, we have $v_1^1 = v_1^2$, so $E_1[x \mapsto v_1^1] \equiv_k E_2[x \mapsto v_1^2]$. Again by induction for $e_2$, we have $v_1 = v_2$.
- Case SR:LET. It is similar to the case SR:L-LET.
- Case SR:GEN and SR:C-GEN. It follows.

**SR:IF** Suppose $e$ is of the form $\mathrm{if}(x, e_t, e_f)$, the evaluation derivation ends with an application of the rule E:IF-TRUE or the rule E:IF-FALSE. The typing derivation ends with an application of the following rules.

- Case SR:L-IF. By the hypothesis we have $k_x \sqsubseteq k_1$, thus $E_1(x) = E_2(x)$. Assume that $E_1(x) = \mathsf{true}$, then $E_1 \vdash e_t \Downarrow v_1$ and $E_2 \vdash e_t \Downarrow v_2$. By induction for $e_t$ we have $v_1 = v_2$ if $S \blacktriangleleft k_1$. It is similar for $E_1(x) = \mathsf{false}$.
- Case SR:IF. If $k_x \sqsubseteq k_1$ the proof is similar to the case SR:L-IF. Otherwise, $k_x \not\sqsubseteq k_1$, thus by the simple security lemma we have $S \not\blacktriangleleft k_1$.
- Case SR:GEN and SR:C-GEN. It follows.

**SR:PAIR** Suppose the evaluation derivation ends with an application of the rule E:PAIR, thus $(E_i(x_1), E_i(x_2)) = v_i$ for $i = \{1, 2\}$. The typing derivation ends with an application of the rules SR:PAIR or SR:GEN.

If $S_1 * S_2 \blacktriangleleft k$, then by the simple security lemma we have $S_1 \blacktriangleleft k_1$ and $S_2 \blacktriangleleft k_1$. Hence it follows $v_1 = v_2$.

**SR:MATCH-P** Suppose the evaluation derivation ends with an application of the rule E:MATCH-P, thus $E_i(x) = (v_1^i, v_2^i)$ and $E_i[x_1 \mapsto v_1^i, x_2 \mapsto v_2^i] \vdash e \Downarrow v_i$ for $i = \{1, 2\}$. The typing derivation ends with an application of the following rules.

- Case SR:MATCH-P. If $S \blacktriangleleft k_1$, then by the simple security lemma we have $S_1 * S_2 \blacktriangleleft k_1$. By the hypothesis, $E_1(x) = E_2(x)$, thus $v_1^1 = v_1^2$ and $v_2^1 = v_2^2$. Hence, $E_1[x_1 \mapsto v_1^1, x_2 \mapsto v_2^1] \equiv_k E_2[x_1 \mapsto v_1^2, x_2 \mapsto v_2^2]$, by induction for $e$ in the premise, it holds that $v_1 = v_2$.
- Case SR:C-MATCH-P. It is similar to the case SR:MATCH-P.

(SR:UNIT)
$$\frac{}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} () : (\text{unit}, \mathrm{pc})}$$

(SR:BOOL)
$$\frac{b \in \{\mathsf{true}, \mathsf{false}\}}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} b : (\text{bool}, \mathrm{pc})}$$

(SR:INT)
$$\frac{n \in \mathbb{Z}}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} n : (\text{int}, \mathrm{pc})}$$

(SR:VAR)
$$\frac{x : S \in \Gamma^s \qquad \mathrm{pc} \lhd S}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} x : S}$$

(SR:B-OP)
$$\frac{x_1 : (\text{bool}, k_{x_1}) \in \Gamma^s \qquad x_2 : (\text{bool}, k_{x_2}) \in \Gamma^s \qquad \mathrm{pc} \sqsubseteq k_{x_1} \sqcup k_{x_2} \qquad \diamond \in \{\mathsf{and}, \mathsf{or}\}}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{op}_\diamond(x_1, x_2) : (\text{bool}, k_{x_1} \sqcup k_{x_2})}$$

(SR:C-GEN)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \vdash e : S \qquad \mathrm{const}_X(e)}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e : S}$$

(SR:GEN)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e : S}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash e : S}$$

(SR:IB-OP)
$$\frac{x_1 : (\text{int}, k_{x_1}) \in \Gamma^s \qquad x_2 : (\text{int}, k_{x_2}) \in \Gamma^s \qquad \mathrm{pc} \sqsubseteq k_{x_1} \sqcup k_{x_2} \qquad \diamond \in \{=, <>, >, <, <=, >=\}}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{op}_\diamond(x_1, x_2) : (\text{bool}, k_{x_1} \sqcup k_{x_2})}$$

(SR:PAIR)
$$\frac{x_1 : S_1 \in \Gamma^s \qquad x_2 : S_2 \in \Gamma^s \qquad \mathrm{pc} \lhd S_1 * S_2}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{pair}(x_1, x_2) : S_1 * S_2}$$

(SR:I-OP)
$$\frac{x_1 : (\text{int}, k_{x_1}) \in \Gamma^s \qquad x_2 : (\text{int}, k_{x_2}) \in \Gamma^s \qquad \mathrm{pc} \sqsubseteq k_{x_1} \sqcup k_{x_2} \qquad \diamond \in \{+, -, *, \mathsf{div}, \mathsf{mod}\}}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{op}_\diamond(x_1, x_2) : (\text{int}, k_{x_1} \sqcup k_{x_2})}$$

(SR:CONS)
$$\frac{x_h : S \in \Gamma^s \qquad x_t : (L(S), k_x) \in \Gamma^s \qquad \mathrm{pc} \lhd (L(S), k_x)}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{cons}(x_h, x_t) : (L(S), k_x)}$$

(SR:FUN)
$$\frac{x : S_1 \in \Gamma^s \qquad \Sigma^s(f) = S_1 \xrightarrow{\mathrm{pc}'} S_2 \qquad \mathrm{pc} \sqsubseteq \mathrm{pc}'}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash \mathrm{app}(f, x) : S_2}$$

(SR:L-ARG)
$$\frac{x : S_1 \in \Gamma^s \qquad \Sigma^s(f) = S_1 \xrightarrow{\mathrm{pc}'} S_2 \qquad \mathrm{pc} \sqsubseteq \mathrm{pc}' \qquad S_1 \blacktriangleleft k_1}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{app}(f, x) : S_2}$$

(SR:SUBTYPING)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \vdash e : S \qquad S \leq S'}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash e : S'}$$

(SR:C-FUN)
$$\frac{x : S_1 \in \Gamma^s \qquad \Sigma^s(f) = S_1 \xrightarrow{\mathrm{pc}'/\text{const}} S_2 \qquad \mathrm{pc} \sqsubseteq \mathrm{pc}'}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{app}(f, x) : S_2}$$

(SR:NIL)
$$\frac{S \in \mathscr{S} \qquad \mathrm{pc} \lhd S}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathsf{nil} : (L(S), \mathrm{pc})}$$

(SR:C-SUBTYPING)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e : S \qquad S \leq S'}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e : S'}$$

(SR:LET)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \vdash e_1 : S_1 \qquad \mathrm{pc};\Sigma^s;\Gamma^s, x : S_1 \vdash e_2 : S_2}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash \mathrm{let}(x, e_1, x.e_2) : S_2}$$

(SR:L-LET)
$$\frac{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e_1 : S_1 \qquad \mathrm{pc};\Sigma^s;\Gamma^s, x : S_1 \overset{\text{const}}{\vdash} e_2 : S_2 \qquad S_1 \blacktriangleleft k_1}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{let}(x, e_1, x.e_2) : S_2}$$

(SR:IF)
$$\frac{x : (\text{bool}, k_x) \in \Gamma^s \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \vdash e_t : S \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \vdash e_f : S \qquad \mathrm{pc} \sqcup k_x \lhd S}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash \mathrm{if}(x, e_t, e_f) : S}$$

(SR:L-IF)
$$\frac{x : (\text{bool}, k_x) \in \Gamma^s \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e_t : S \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e_f : S \qquad \mathrm{pc} \sqcup k_x \lhd S \qquad k_x \sqsubseteq k_1}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{if}(x, e_t, e_f) : S}$$

(SR:MATCH-P)
$$\frac{x : S_1 * S_2 \in \Gamma^s \qquad \mathrm{pc};\Sigma^s;\Gamma^s, x_1 : S_1, x_2 : S_2 \vdash e : S}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash \mathrm{match}(x, (x_1, x_2).e) : S}$$

(SR:C-MATCH-P)
$$\frac{x : S_1 * S_2 \in \Gamma^s \qquad \mathrm{pc};\Sigma^s;\Gamma^s, x_1 : S_1, x_2 : S_2 \overset{\text{const}}{\vdash} e : S}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{match}(x, (x_1, x_2).e) : S}$$

(SR:MATCH-L)
$$\frac{x : (L(S), k_x) \in \Gamma^s \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \vdash e_1 : S_1 \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s, x_h : S, x_t : (L(S), k_x) \vdash e_2 : S_1 \qquad \mathrm{pc} \sqcup k_x \lhd S_1}{\mathrm{pc};\Sigma^s;\Gamma^s \vdash \mathrm{match}(x, e_1, (x_h, x_t).e_2) : S_1}$$

(SR:C-MATCH-L)
$$\frac{x : (L(S), k_x) \in \Gamma^s \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} e_1 : S_1 \qquad \mathrm{pc} \sqcup k_x;\Sigma^s;\Gamma^s, x_h : S, x_t : (L(S), k_x) \overset{\text{const}}{\vdash} e_2 : S_1 \qquad \mathrm{pc} \sqcup k_x \lhd S_1}{\mathrm{pc};\Sigma^s;\Gamma^s \overset{\text{const}}{\vdash} \mathrm{match}(x, e_1, (x_h, x_t).e_2) : S_1}$$

**Figure 9.** Typing rules: security

- Case SR:GEN and SR:C-GEN. It follows.

**SR:NIL** It is similar to the case SR:UNIT.

**SR:CONS** Suppose the evaluation derivation ends with an application of the rule E:CONS, thus $E_i(x_h) = v_1^i$ and $E_i(x_t) = [v_2^i, \cdots, v_n^i]$ for $i = \{1,2\}$. The typing derivation ends with an application of the rules SR:CONS or SR:GEN. If $(L(S), k_x) \blacktriangleleft k_1$ then by the hypothesis we have $v_1^1 = v_1^2$ and $[v_2^1, \cdots, v_n^1] = [v_2^2, \cdots, v_n^2]$. Thus $E_1(\mathsf{cons}(x_h, x_t)) = E_2(\mathsf{cons}(x_h, x_t))$.

**SR:MATCH-L** Suppose $e$ is of the form $\mathrm{match}(x, e_1, (x_h, x_t).e_2)$, the evaluation derivation ends with an application of the rule E:MATCH-N or the rule E:MATCH-L. The typing derivation ends with an application of the following rules.

- Case SR:MATCH-L. If $S_1 \blacktriangleleft k_1$, then by the simple security lemma we have $(L(S), k_x) \blacktriangleleft k_1$. By the hypothesis we have $E_1(x) = E_2(x)$. Assume that $E_1(x) = E_2(x) = [v_1, \cdots, v_n]$, by the rule E:MATCH-L we have $E_i[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]] \vdash e_2 \Downarrow v_i$ for $i = \{1,2\}$. Since $E_1[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]] \equiv_k E_2[x_h \mapsto v_1, x_t \mapsto [v_2, ..., v_n]]$, by induction for $e_2$, it holds that $v_1 = v_2$ if $S_1 \blacktriangleleft k_1$. It is similar for $E_1(x) = E_2(x) = \mathsf{nil}$.
- Case SR:C-MATCH-L.It is similar to the case SR:MATCH-L.
- Case SR:GEN and SR:C-GEN. It follows.

**SR:SUBTYPING** Suppose the typing derivation ends with the rule SR:SUBTYPING. If $S' \blacktriangleleft k_1$ then $S \blacktriangleleft k_1$. Thus by induction for $e$ in the premise it follows. It is similar for SR:C-SUBTYPING. □

**Theorem 5.** *If* $\models E : \Gamma^s$, $E \vdash e \Downarrow v$, *and* $pc; \Sigma^s; \Gamma^s \vdash^{const} e : S$ *then* $e$ *is resource-aware noninterference expression at level* $k_1$.

*Proof.* The proof is done by induction on the structure of the typing derivation and the evaluation derivation. Let $X$ be the set of variables $[\Gamma^s]_{\blacktriangleleft k_1}$. For all environments $E_1$, $E_2$ such that $E_1 \approx_X E_2$ and $E_1 \equiv_{k_1} E_2$, if $E_1 \vdash^{p_1}_{p_1'} e \Downarrow v_1$ and $E_2 \vdash^{p_2}_{p_2'} e \Downarrow v_2$. We then show that $p_1 - p_1' = p_2 - p_2'$ and $v_1 = v_2$ if $S \blacktriangleleft k_1$. By Lemma 3, $e$ satisfies the noninterference property at security label $k_1$. Thus we need to prove that $p_1 - p_1' = p_2 - p_2'$.

**SR:UNIT** Suppose the evaluation derivation of $e$ ends with an application of the rule E:UNIT, thus $p_1 - p_1' = p_2 - p_2' = K^{\mathsf{unit}}$.

**SR:BOOL** It is similar to the case SR:UNIT.

**SR:INT** It is similar to the case SR:UNIT.

**SR:VAR** It is similar to the case SR:UNIT.

**SR:B-OP** Suppose the evaluation derivation ends with an application of the rule E:BIN, thus $E_1 \vdash^{p_1' + K^{\mathsf{op}}}_{p_1'} e \Downarrow v_1$ and $E_1 \vdash^{p_2' + K^{\mathsf{op}}}_{p_2'} e \Downarrow v_1$. We have $p_1 - p_1' = p_2 - p_2' = K^{\mathsf{op}}$.

**SR:IB-OP** It is similar to the case SR:B-OP.

**SR:I-OP** It is similar to the case SR:B-OP.

**SR:C-GEN** By the hypothesis we have $\mathrm{const}_X(e)$, thus it holds that $\Sigma^r; \Gamma^r \vdash^q_{q'} e : A$ and $\curlyvee(A \mid A, A)$. By the constant-resource theorem, for all $p_1, p_1', p_2, p_2' \in \mathbb{Q}_0^+$ such that $E_1 \vdash^{p_1}_{p_1'} e \Downarrow v_1$ and $E_2 \vdash^{p_2}_{p_2'} e \Downarrow v_2$, we have $p_1 - p_1' = q + \Phi_{E_1}(\Gamma^r) - (q' + \Phi(v_1 : A))$ and $p_2 - p_2' = q + \Phi_{E_2}(\Gamma^r) - (q' + \Phi(v_2 : A))$.

Since $E_1 \approx_X E_2$, $\Phi_{E_1}(X) = \Phi_{E_2}(X)$. For all $y \notin X$, $E_1(y) = E_2(y)$ since $E_1 \equiv_{k_1} E_2$, thus $\Phi(E_1(y)) = \Phi(E_2(y))$. Hence, $\Phi_{E_1}(\Gamma^r) = \Phi_{E_2}(\Gamma^r)$, it follows $p_1 - p_1' = p_2 - p_2'$.

**SR:FUN** Suppose $e$ is of the form $\mathrm{app}(f, x)$, thus the typing derivation ends with an application of either the rule SR:L-ARG, SR:C-FUN, or SR:C-GEN.

- Case SR:L-ARG. By the hypothesis we have $E_1(x) = E_2(x)$, it follows $p_1 - p_1' = p_2 - p_2'$.
- Case SR:C-FUN. Because $e$ is well-formed, there exists a well-typed expression $e_f$ such that $\mathrm{pc}'; \Sigma^s; \Gamma^s \vdash^{\mathrm{const}} e_f : S_2$. By induction for $e_f$ which is resource-aware noninterference w.r.t $X$, $p_1 - K^{\mathsf{app}} - p_1' = p_2 - K^{\mathsf{app}} - p_2'$, it follows.
- Case SR:C-GEN. By the case SR:C-GEN it follows.

**SR:LET** Suppose $e$ is of the form $\mathrm{let}(x, e_1, x.e_2)$, thus the typing derivation ends with an application of either the rule SR:L-LET or SR:C-GEN.

- Case SR:L-LET. Suppose the evaluations $E_1 \vdash^{p_1 - K^{\mathsf{let}}}_{p'} e_1 \Downarrow v_1^1$, $E_2 \vdash^{p_2 - K^{\mathsf{let}}}_{p"} e_1 \Downarrow v_1^2$, $E_1[x \mapsto v_1^1] \vdash^{p'}_{p_1'} e_2 \Downarrow v_1$, and $E_2[x \mapsto v_1^2] \vdash^{p"}_{p_2'} e_2 \Downarrow v_2$. By induction for $e_1$ that is resource-aware noninterference w.r.t $X$, $p_1 - K^{\mathsf{let}} - p' = p_2 - K^{\mathsf{let}} - p"$. By the hypothesis $v_1^1 = v_1^2$. Thus $E_1[x \mapsto v_1^1] \approx_X E_2[x \mapsto v_1^2]$ and $E_1[x \mapsto v_1^1] \equiv_{k_1} E_2[x \mapsto v_1^2]$, by induction for $e_2$ that is resource-aware noninterference w.r.t $X$, we have $p' - p_1' = p" - p_2'$. Hence, $p_1 - p_1' = p_2 - p_2'$.
- Case SR:C-GEN. By the case SR:C-GEN it follows.

**SR:IF** Suppose $e$ is of the form $\mathrm{if}(x, e_t, e_f)$, thus the typing derivation ends with an application of either the rule SR:L-IF or SR:C-GEN.

- Case SR:L-IF. By the hypothesis we have $E_1(x) = E_2(x)$. Assume that $E_1(x) = E_2(x) = \mathsf{true}$, by the evaluation rule E:IF-TRUE, $E_1 \vdash^{p_1 - K^{\mathsf{cond}}}_{p_1'} e_t \Downarrow v_1$ and $E_2 \vdash^{p_2 - K^{\mathsf{cond}}}_{p_2'} e_t \Downarrow v_2$. By induction for $e_t$ that is resource-aware noninterference w.r.t $X$, we have $p_1 - p_1' = p_2 - p_2'$. It is similar for $E_1(x) = E_2(x) = \mathsf{false}$.
- Case SR:C-GEN. Since $E_1 \approx_X E_2$ w.r.t $\Gamma^s$, we have $E_1 \approx_X E_2$ w.r.t $\Gamma^r$. By the hypothesis we have $\mathrm{const}_X(e)$. Thus by the soundness theorem of constant resource type system, it follows $p_1 - p_1' = p_2 - p_2'$.

**SR:PAIR** It is similar to the case SR:B-OP.

**SR:MATCH-P** Suppose $e$ is of the form $\mathrm{match}(x, (x_1, x_2).e)$, thus the typing derivation ends with an application of either the rule SR:C-MATCH-P or SR:C-GEN.

- Case SR:C-MATCH-P. Let $E_1' = E_1[x_1 \mapsto v_1^1, x_2 \mapsto v_2^1]$ and $E_2' = E_2[x_1 \mapsto v_1^2, x_2 \mapsto v_2^2]$. If $x \in X$ then $|E_1(x)| \approx |E_2(x)|$. Thus $|E_1'(x_1)| \approx |E_2'(x_1)|$ and $|E_1'(x_2)| \approx |E_2'(x_2)|$. Hence, $E_1' \approx_{X \cup \{x_1, x_2\}} E_2'$, by induction for $e$ in the premise which is resource-aware noninterference w.r.t $X \cup \{x_1, x_2\}$, $p_1 - K^{\mathsf{matchP}} - p_1' = p_2 - K^{\mathsf{matchP}} - p_2'$, it follows. If $x \notin X$ then $E_1(x) = E_2(x)$, it is similar.
- Case SR:C-GEN. By the case SR:C-GEN it follows.

**SR:NIL** It is similar to the case SR:UNIT.

**SR:CONS** It is similar to the case SR:B-OP.

**SR:MATCH-L** Suppose $e$ is of the form $\mathrm{match}(x, e_1, (x_h, x_t).e_2)$, thus the typing derivation ends with an application of either the rule SR:C-MATCH-L or SR:C-GEN.

- Case SR:C-MATCH-L. Let $E_1' = E_1[x_h \mapsto v_1^1, x_t \mapsto v_2^1]$ and $E_2' = E_2[x_h \mapsto v_1^2, x_t \mapsto v_2^2]$. If $x \in X$ then $|E_1(x)| \approx |E_2(x)|$. Suppose $E_1(x)$ and $E_2(x)$ are different from $\mathsf{nil}$, $|E_1'(x_h)| \approx |E_2'(x_h)|$ and $|E_1'(x_t)| \approx |E_2'(x_t)|$. Hence, $E_1' \approx_{X \cup \{x_t, x_h\}} E_2'$, by induction for $e_2$ which is resource-aware noninterference w.r.t $X \cup \{x_t, x_h\}$, we have $p_1 - K^{\mathsf{matchL}} - p_1' = p_2 - K^{\mathsf{matchL}} - p_2'$, thus $p_1 - p_1' = p_2 - p_2'$. If $E_1(x) = E_2(x) = \mathsf{nil}$ then by induction for $e_1$ that is resource-aware noninterference w.r.t $X$, it follows. If $x \notin X$ then $E_1(x) = E_2(x)$, it is similar.
- Case SR:C-GEN. By the case SR:C-GEN it follows.

**SR:SUBTYPING** The typing derivation ends with an application of either the rule SR:C-SHARE or SR:C-GEN.

- Case SR:C-SUBTYPING. By induction for $e$ in the premise, $p_1 - p_1' = p_2 - p_2'$.
- Case SR:C-GEN. By the case SR:C-GEN it follows.

□