Semantic Type Qualifiers

Brian Chin Shane Markstrum Todd Millstein
Computer Science Department
University of California, Los Angeles
{naerbnic,smarkstr,todd}@cs.ucla.edu

Technical Report CSD-TR-040045
November 2004

This technical report provides the formal details of our framework for semantic type qualifiers, which is described in a PLDI 2005 paper of the same name. We defer to that paper for the high-level description of the framework. This report formalizes the case when all user-defined type qualifiers are *value* qualifiers. Sections 1 and 2 respectively formalize the syntax and semantics of our formal language, and section 3 presents a proof of "semantic" soundness of the language's type system.

1 Syntax

The language is a fairly standard simply-typed lambda calculus, augmented with references and user-defined type qualifiers. For convenience, we separate side-effect-free expressions (called *expressions*) from potentially side-effecting expressions (called *statements*). This separation causes no loss of expressiveness.

$$
\begin{array}{llll}
\textit{Stmts} & s & ::= & e \mid s_1\ s_2 \mid \texttt{let}\ x = s_1\ \texttt{in}\ s_2 \mid \texttt{ref}\ s \mid s_1 := s_2 \\
\textit{Exprs} & e & ::= & c \mid () \mid x \mid \lambda x.s \mid !e \\
\textit{Consts} & c & ::= & \text{integer constants} \\
\textit{Vars} & x & ::= & \text{variable names} \\
\textit{Types} & \tau & ::= & \texttt{unit} \mid \texttt{int} \mid \tau_1 \rightarrow \tau_2 \mid \texttt{ref}\ \tau \mid q\tau \\
\textit{Qualifiers} & q & ::= & \text{user-defined value qualifiers}
\end{array}
$$

We restrict the above syntax of types slightly: for any type containing a component of the form $q\tau$, we require that τ not be of the form $\tau_1 \rightarrow \tau_2$. This restriction is consistent with our implementation of semantic type qualifiers for C, and it makes the soundness proof cleaner. Note that types such as $(q\tau_1) \rightarrow (q'\tau_2)$ and $q(\texttt{ref}\ (\tau_1 \rightarrow \tau_2))$ are still supported.

We also need a notion of *values*, which are the legal results of expressions:

$$
\begin{array}{llll}
\textit{Values} & v & ::= & c \mid () \mid \lambda x.s \mid l \\
\textit{Locations} & l & ::= & \text{location constants (i.e., addresses)}
\end{array}
$$

Note that locations are not directly available at source level.

2 Semantics

2.1 Static Semantics

The base type system is standard and is defined by the following rules. As usual, metavariable Γ ranges over *type environments*, which are finite functions from variable names to types. Also, we assume that bound variables are α-renamed as necessary.

$\boxed{\Gamma \vdash s : \tau}$

$$\frac{\Gamma \vdash s_1 : \tau_2 \rightarrow \tau \qquad \Gamma \vdash s_2 : \tau_2}{\Gamma \vdash s_1\ s_2 : \tau}\ \text{T-App} \qquad \frac{\Gamma \vdash s_1 : \tau_1 \qquad \Gamma, x : \tau_1 \vdash s_2 : \tau}{\Gamma \vdash \texttt{let}\ x = s_1\ \texttt{in}\ s_2 : \tau}\ \text{T-Let}$$

$$\frac{\Gamma \vdash s : \tau}{\Gamma \vdash \texttt{ref}\ s : \texttt{ref}\ \tau}\ \text{T-Ref} \qquad \frac{\Gamma \vdash s_1 : \texttt{ref}\ \tau \qquad \Gamma \vdash s_2 : \tau}{\Gamma \vdash s_1 := s_2 : \texttt{unit}}\ \text{T-Assgn}$$

$$\frac{}{\Gamma \vdash c : \texttt{int}}\ \text{T-Int} \qquad \frac{}{\Gamma \vdash () : \texttt{unit}}\ \text{T-Unit} \qquad \frac{\Gamma(x) = \tau}{\Gamma \vdash x : \tau}\ \text{T-Var}$$

$$\frac{\Gamma, x : \tau_1 \vdash s : \tau_2}{\Gamma \vdash \lambda x.s : \tau_1 \rightarrow \tau_2}\ \text{T-Fun} \qquad \frac{\Gamma \vdash e : \texttt{ref}\ \tau}{\Gamma \vdash !e : \tau}\ \text{T-Deref}$$

In addition to these rules, users can provide a set of *introduction* rules for value-qualified types. Each rule is assumed to match the following template:

$$\frac{\begin{array}{c}\Gamma \vdash e : \tau \\ \Gamma \vdash e_1 : q_1\tau_1 \qquad \ldots \qquad \Gamma \vdash e_n : q_n\tau_n \\ \text{each } e_i \text{ is a subexpression of } e\end{array}}{\Gamma \vdash e : q\tau}\ \text{T-QualCase}$$

This template formalizes the `case` rules in our implementation for C. For example, here is an introduction rule indicating that the product of two positive expressions is also positive.

$$\frac{\Gamma \vdash e_1 * e_2 : \overline{q}(\texttt{int}) \qquad \Gamma \vdash e_1 : \texttt{pos int} \qquad \Gamma \vdash e_2 : \texttt{pos int}}{\Gamma \vdash e_1 * e_2 : \texttt{pos}\ \overline{q}(\texttt{int})}\ \text{PosMult}$$

In addition to the user-defined rules of the form specified by T-QualCase, we provide a "base case" introduction rule for all qualified types. We assume that the definition of each qualifier q includes a unary predicate $[[q]]$ on values, which is used below to formalize a qualifier's semantics. The base case says that any value satisfying $[[q]]$ may have a type qualified with q:

$$\frac{[[q]](v) \qquad \Gamma \vdash v : \tau}{\Gamma \vdash v : q\tau}\ \text{T-QualVal}$$

This natural rule facilitates the proof of our soundness theorem, described below.

Finally, we include a subsumption rule and an associated subtyping relation for types, defined by the following rules:

$$\frac{\Gamma \vdash s : \tau' \qquad \tau' \leq \tau}{\Gamma \vdash s : \tau}\ \text{T-Sub}$$

$$\boxed{\tau \leq \tau'}$$

$$\frac{}{q\tau \leq \tau}\ \textsc{SubValQual} \qquad \frac{}{q_1 q_2 \tau \leq q_2 q_1 \tau}\ \textsc{SubQualReorder}$$

$$\frac{}{\tau \leq \tau}\ \textsc{SubRef} \qquad \frac{\tau \leq \tau'' \qquad \tau'' \leq \tau'}{\tau \leq \tau'}\ \textsc{SubTrans}$$

$$\frac{\tau_1' \leq \tau_1 \qquad \tau_2 \leq \tau_2'}{\tau_1 \rightarrow \tau_2 \leq \tau_1' \rightarrow \tau_2'}\ \textsc{SubFun}$$

2.2 Dynamic Semantics

The dynamic semantics describes how to evaluate programs. Metavariable σ ranges over *stores*, which are finite functions from locations to values. We define an abstract machine for our language. A machine configuration $< \sigma, s >$ is a pair of a store and a statement to be evaluated. The steps of the machine are defined by the following inference rules:

$$\boxed{< \sigma, s > \rightarrow < \sigma', v >}$$

$$\frac{< \sigma, s_1 > \rightarrow < \sigma_1, \lambda x.s > \qquad < \sigma_1, s_2 > \rightarrow < \sigma_2, v_2 > \qquad < \sigma_2, s[x \mapsto v_2] > \rightarrow < \sigma', v >}{< \sigma, s_1\ s_2 > \rightarrow < \sigma', v >}\ \textsc{E-App}$$

$$\frac{< \sigma, s_1 > \rightarrow < \sigma_1, v_1 > \qquad < \sigma_1, s_2[x \mapsto v_1] > \rightarrow < \sigma_2, v_2 >}{< \sigma, \texttt{let}\ x = s_1\ \texttt{in}\ s_2 > \rightarrow < \sigma_2, v_2 >}\ \textsc{E-Let}$$

$$\frac{< \sigma, s > \rightarrow < \sigma', v > \qquad l \text{ fresh in } \sigma'}{< \sigma, \texttt{ref}\ s > \rightarrow < \sigma'[l \mapsto v], l >}\ \textsc{E-Ref}$$

$$\frac{< \sigma, s_1 > \rightarrow < \sigma_1, l > \qquad < \sigma_1, s_2 > \rightarrow < \sigma_2, v > \qquad l \in \mathrm{dom}(\sigma_1)}{< \sigma, s_1 := s_2 > \rightarrow < \sigma_2[l \mapsto v], () >}\ \textsc{E-Assgn}$$

$$\frac{< \sigma, e > \rightarrow v}{< \sigma, e > \rightarrow < \sigma, v >}\ \textsc{E-Expr}$$

$$\boxed{< \sigma, e > \rightarrow v}$$

$$\frac{}{< \sigma, v > \rightarrow v}\ \textsc{E-Val} \qquad \frac{< \sigma, e > \rightarrow l \qquad \sigma(l) = v}{< \sigma, !e > \rightarrow v}\ \textsc{E-Deref}$$

3 Soundness

We use a qualifier q's associated predicate $[[q]]$ to formalize a local soundness condition on user-defined type rules. This formalization makes use of an overloading of the $[[q]]$ notation that lifts these predicates from values to arbitrary expressions:

$$[[q]](\sigma, e, v) \equiv (< \sigma, e > \rightarrow v \Rightarrow [[q]](v))$$

Definition 3.1 A type rule matching the template T-QualCase is *locally sound* if the following proof obligation is true:

$\forall \sigma, v_1, \ldots, v_n, v.([[q_1]](\sigma, e_1, v_1) \wedge \ldots \wedge [[q_n]](\sigma, e_n, v_n)) \Rightarrow [[q]](\sigma, e, v)$

Intuitively, (global) soundness means that, if all user-defined type rules are locally sound, then any well-typed program fragment will satisfy its qualifiers' invariants at run time. We formalize this notion of type soundness via a few auxiliary definitions.

$\boxed{\Gamma; \tau \vdash < \sigma, v >}$

$$\frac{}{\Gamma; \texttt{int} \vdash < \sigma, c >}\text{ Q-INT} \qquad \frac{}{\Gamma; \texttt{unit} \vdash < \sigma, () >}\text{ Q-UNIT} \qquad \frac{\Gamma \vdash \lambda x.s : \tau_1 \rightarrow \tau_2}{\Gamma; \tau_1 \rightarrow \tau_2 \vdash < \sigma, \lambda x.s >}\text{ Q-FUN}$$

$$\frac{\Gamma \vdash l : \texttt{ref}\ \tau \qquad \Gamma; \tau \vdash < \sigma, \sigma(l) > \qquad l \in \text{dom}(\sigma)}{\Gamma; \texttt{ref}\ \tau \vdash < \sigma, l >}\text{ Q-REF}$$

$$\frac{[[q]](v) \qquad \Gamma; \tau \vdash < \sigma, v >}{\Gamma; q\tau \vdash < \sigma, v >}\text{ Q-QUAL}$$

The relation $\Gamma; \tau \vdash < \sigma, v >$ represents *semantic conformance* of a value to a type. Intuitively, $\Gamma; \tau \vdash < \sigma, v >$ holds if $\Gamma \vdash v : \tau$ and v additionally satisfies all of the associated invariants for qualifiers in τ. The first three rules are the standard typechecking rules for integers, unit, and functions, respectively. Rule Q-Qual checks that a value of qualified type satisfies the qualifier's invariant. Rule Q-Ref checks that a location l is well-typed and recursively checks semantic conformance of the value that l points to in the given store. For purposes of the static semantics we treat locations as variables.

Next we lift this notion of semantic conformance to a relation between a store and a type environment:

Definition 3.2 We say that $\Gamma \sim \sigma$ if both of the following conditions hold:

1. $\text{dom}(\Gamma) = \text{dom}(\sigma)$
2. $\forall l \in \text{dom}(\Gamma).(\Gamma; \Gamma(l) \vdash < \sigma, l >)$

In other words, $\Gamma \sim \sigma$ if every memory location is well typed and satisfies its qualifiers' invariants.

Finally we can state our type soundness theorem, which is a variant of the standard type preservation theorem:

Theorem 3.1 If $\Gamma \sim \sigma$ and $\Gamma \vdash s : \tau$ and $< \sigma, s > \rightarrow < \sigma', v >$ and all user-defined type rules are locally sound, then there exists some $\Gamma' \supseteq \Gamma$ such that $\Gamma' \sim \sigma'$ and $\Gamma'; \tau \vdash < \sigma', v >$.

To prove this theorem, it is helpful to make use of (un)reachability properties of well-formed stores. The following judgment and associated inference rules formalize when a value cannot reach a location through a given store.

$\boxed{\sigma \vdash v \not\hookrightarrow l}$

$$\frac{}{\sigma \vdash c \not\hookrightarrow l}\text{ UNREACHINT} \qquad \frac{}{\sigma \vdash () \not\hookrightarrow l}\text{ UNREACHUNIT} \qquad \frac{}{\sigma \vdash \lambda x.s \not\hookrightarrow l}\text{ UNREACHFUN}$$

$$\frac{l' \neq l \qquad l' \in \mathrm{dom}(\sigma) \Rightarrow \sigma \vdash \sigma(l') \not\leadsto l}{\sigma \vdash l' \not\leadsto l} \textsc{UnreachLoc}$$

Finally we prove Theorem 3.1:

Proof By induction on the depth of the derivation of $< \sigma, s > \rightarrow < \sigma', v >$. Case analysis of the last rule used in the derivation.

- E-App: Then $s = s_1\ s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, \lambda x.s' >$ and $< \sigma_1, s_2 > \rightarrow < \sigma_2, v_2 >$ and $< \sigma_2, s'[x \mapsto v_2] > \rightarrow < \sigma', v >$. We prove this case by induction on the depth of the derivation of $\Gamma \vdash s : \tau$. Case analysis of the last rule used in the derivation.
  - T-App: Then $\Gamma \vdash s_1 : \tau_2 \rightarrow \tau$ and $\Gamma \vdash s_2 : \tau_2$. By (outer) induction there exists $\Gamma_1 \supseteq \Gamma$ such that $\Gamma_1 \sim \sigma_1$ and $\Gamma_1; \tau_2 \rightarrow \tau \vdash < \sigma_1, \lambda x.s' >$. Since $\Gamma \vdash s_2 : \tau_2$, by Lemma 3.6 also $\Gamma_1 \vdash s_2 : \tau_2$. Then by (outer) induction there exists $\Gamma_2 \supseteq \Gamma_1$ such that $\Gamma_2 \sim \sigma_2$ and $\Gamma_2; \tau_2 \vdash < \sigma_2, v_2 >$.

    Since $\Gamma_1; \tau_2 \rightarrow \tau \vdash < \sigma_1, \lambda x.s' >$, by Q-Fun also $\Gamma_1 \vdash \lambda x.s' : \tau_2 \rightarrow \tau$, so by Lemma 3.6 we have $\Gamma_2 \vdash \lambda x.s' : \tau_2 \rightarrow \tau$. Then by Lemma 3.12 we have $\Gamma_2, x : \tau_2' \vdash s' : \tau'$, where $\tau_2 \leq \tau_2'$ and $\tau' \leq \tau$. Then by T-Sub also $\Gamma_2, x : \tau_2' \vdash s' : \tau$. Since $\Gamma_2; \tau_2 \vdash < \sigma_2, v_2 >$ and $\tau_2 \leq \tau_2'$, by Lemma 3.4 also $\Gamma_2; \tau_2' \vdash < \sigma_2, v_2 >$. So we have $\Gamma_2, x : \tau_2' \vdash s' : \tau$ and $\Gamma_2; \tau_2' \vdash < \sigma_2, v_2 >$, and by Lemma 3.2 also $\Gamma_2 \vdash s'[x \mapsto v_2] : \tau$. Since also $\Gamma_2 \sim \sigma_2$ and $< \sigma_2, s'[x \mapsto v_2] > \rightarrow < \sigma', v >$, by (outer) induction there exists some $\Gamma' \supseteq \Gamma_2$ such that $\Gamma' \sim \sigma'$ and $\Gamma'; \tau \vdash < \sigma', v >$.
  - T-Sub: Then $\Gamma \vdash s : \tau'$ and $\tau' \leq \tau$. By inner induction, there exists some $\Gamma' \supseteq \Gamma$ such that $\Gamma' \sim \sigma'$ and $\Gamma'; \tau' \vdash < \sigma', v >$. Then by Lemma 3.4 also $\Gamma'; \tau \vdash < \sigma', v >$.
- E-Let: Then $s = \texttt{let}\ x = s_1\ \texttt{in}\ s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, v_1 >$ and $< \sigma_1, s_2[x \mapsto v_1] > \rightarrow < \sigma', v >$. We prove this case by induction on the depth of the derivation of $\Gamma \vdash s : \tau$. Case analysis of the last rule used in the derivation.
  - T-Let: Then $\Gamma \vdash s_1 : \tau_1$ and $\Gamma, x : \tau_1 \vdash s_2 : \tau$. By (outer) induction there exists $\Gamma_1 \supseteq \Gamma$ such that $\Gamma_1 \sim \sigma_1$ and $\Gamma_1; \tau_1 \vdash < \sigma_1, v_1 >$. Since $\Gamma, x : \tau_1 \vdash s_2 : \tau$, by Lemma 3.6 also $\Gamma_1, x : \tau_1 \vdash s_2 : \tau$. Then since $\Gamma_1; \tau_1 \vdash < \sigma_1, v_1 >$, by Lemma 3.2 also $\Gamma_1 \vdash s_2[x \mapsto v_1] : \tau$. Finally, since $< \sigma_1, s_2[x \mapsto v_1] > \rightarrow < \sigma', v >$, by (outer) induction there exists some $\Gamma' \supseteq \Gamma_1$ such that $\Gamma' \sim \sigma'$ and $\Gamma'; \tau \vdash < \sigma', v >$.
  - T-Sub: See the proof of the T-Sub case within the case for E-App.
- E-Ref: Then $s = \texttt{ref}\ s_0$ and $< \sigma, s_0 > \rightarrow < \sigma_0, v_0 >$ and l fresh in σ_0 and $\sigma' = \sigma_0[l \mapsto v_0]$ and $v = l$. We prove this case by induction on the depth of the derivation of $\Gamma \vdash s : \tau$. Case analysis of the last rule used in the derivation.
  - T-Ref: Then $\tau = \texttt{ref}\ \tau_0$ and $\Gamma \vdash s_0 : \tau_0$. By (outer) induction there exists some $\Gamma_0 \supseteq \Gamma$ such that $\Gamma_0 \sim \sigma_0$ and $\Gamma_0; \tau_0 \vdash < \sigma_0, v_0 >$. Let $\Gamma' = \Gamma_0[l \mapsto \texttt{ref}\ \tau_0]$. Since l fresh in σ_0 and $\Gamma_0 \sim \sigma_0$, also $l \notin \mathrm{dom}(\Gamma_0)$, so $\Gamma' \supseteq \Gamma_0$. To complete this case we show that $\Gamma' \sim \sigma'$ and $\Gamma'; \texttt{ref}\ \tau_0 \vdash < \sigma', l >$.

    First we prove $\Gamma'; \texttt{ref}\ \tau_0 \vdash < \sigma', l >$. We're given $\Gamma_0; \tau_0 \vdash < \sigma_0, v_0 >$. Since l fresh in σ_0, also $l \notin \mathrm{dom}(\sigma_0)$, so $\sigma' \supseteq \sigma_0$. We also saw above that $\Gamma' \supseteq \Gamma_0$. Then by Lemma 3.5 we have $\Gamma'; \tau_0 \vdash < \sigma', v_0 >$. By T-Var and the definition of Γ' we have $\Gamma' \vdash l : \texttt{ref}\ \tau_0$. Finally, by definition of σ' we have that $l \in \mathrm{dom}(\sigma')$ and $\sigma'(l) = v_0$. Therefore by Q-Ref we have $\Gamma'; \texttt{ref}\ \tau_0 \vdash < \sigma', l >$.

Finally we prove $\Gamma' \sim \sigma'$. Since $\text{dom}(\Gamma_0) = \text{dom}(\sigma_0)$, also $\text{dom}(\Gamma') = \text{dom}(\sigma')$, so part 1 is proven. Now consider some $l' \in \text{dom}(\Gamma')$. To prove part 2 we must show that $\Gamma'; \Gamma'(l') \vdash< \sigma', l' >$. If $l' = l$, then we must show that $\Gamma'; \texttt{ref}\ \tau_0 \vdash< \sigma', l >$, which was proven above. Otherwise $l' \neq l$. Then $l' \in \text{dom}(\Gamma_0)$ and since $\Gamma_0 \sim \sigma_0$, we have $\Gamma_0; \Gamma_0(l') \vdash< \sigma_0, l' >$. Since $l' \neq l$, we have $\Gamma_0(l') = \Gamma'(l')$, so $\Gamma_0; \Gamma'(l') \vdash< \sigma_0, l' >$. Then by Lemma 3.5 we have $\Gamma'; \Gamma'(l') \vdash< \sigma', l' >$.

  - T-Sub: See the proof of the T-Sub case within the case for E-App.

- E-Assgn: Then $s = s_1 := s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, l_1 >$ and $< \sigma_1, s_2 > \rightarrow < \sigma_2, v_2 >$ and $l_1 \in \text{dom}(\sigma_1)$ and $\sigma' = \sigma_2[l_1 \mapsto v_2]$ and $v = ()$. We prove this case by induction on the depth of the derivation of $\Gamma \vdash s : \tau$. Case analysis of the last rule used in the derivation.

  - T-Assgn: Then $\tau = \texttt{unit}$ and $\Gamma \vdash s_1 : \texttt{ref}\ \tau'$ and $\Gamma \vdash s_2 : \tau'$. By (outer) induction there exists some $\Gamma_1 \supseteq \Gamma$ such that $\Gamma_1 \sim \sigma_1$ and $\Gamma_1; \texttt{ref}\ \tau' \vdash< \sigma_1, l_1 >$. Since $\Gamma \vdash s_2 : \tau'$ and $\Gamma_1 \supseteq \Gamma$, by Lemma 3.6 also $\Gamma_1 \vdash s_2 : \tau'$. Then by (outer) induction there exists some $\Gamma_2 \supseteq \Gamma_1$ such that $\Gamma_2 \sim \sigma_2$ and $\Gamma_2; \tau' \vdash< \sigma_2, v_2 >$.

    To prove this case, we must show that there exists $\Gamma' \supseteq \Gamma$ such that $\Gamma' \sim \sigma'$ and $\Gamma'; \texttt{unit} \vdash< \sigma', () >$. We will show that $\Gamma_2 \sim \sigma'$ and $\Gamma_2; \texttt{unit} \vdash< \sigma', () >$. $\Gamma_2; \texttt{unit} \vdash< \sigma', () >$ follows from Q-Unit, so it remains to prove $\Gamma_2 \sim \sigma'$.

    First we show that $\text{dom}(\Gamma_2) = \text{dom}(\sigma')$. Since $\Gamma_2 \sim \sigma_2$, we know that $\text{dom}(\Gamma_2) = \text{dom}(\sigma_2)$. Since $l_1 \in \text{dom}(\sigma_1)$ and $< \sigma_1, s_2 > \rightarrow < \sigma_2, v_2 >$, by Lemma 3.7 also $l_1 \in \text{dom}(\sigma_2)$. Therefore, $\text{dom}(\sigma_2) = \text{dom}(\sigma_2[l_1 \mapsto v_2]) = \text{dom}(\sigma')$. Therefore $\text{dom}(\Gamma_2) = \text{dom}(\sigma')$.

    Second, we must show that for each $l \in \text{dom}(\Gamma_2)$ we have $\Gamma_2; \Gamma_2(l) \vdash< \sigma', l >$. Since $\Gamma_2 \sim \sigma_2$ we have $\Gamma_2; \Gamma_2(l) \vdash< \sigma_2, l >$. Suppose $\sigma_2 \vdash l \not\twoheadrightarrow l_1$. Then by Lemma 3.15 we have $\Gamma_2; \Gamma_2(l) \vdash< \sigma', l >$ as desired. Suppose instead that it is not the case that $\sigma_2 \vdash l \not\twoheadrightarrow l_1$. Then since $\Gamma_2 \sim \sigma_2$, by Lemma 3.16 there exists a nonnegative integer k such that $\sigma_2^k(l) = l_1$. Since $\Gamma_1; \texttt{ref}\ \tau' \vdash< \sigma_1, l_1 >$, by Q-Ref we have $\Gamma_1 \vdash l_1 : \texttt{ref}\ \tau'$, and by Lemma 3.6 also $\Gamma_2 \vdash l_1 : \texttt{ref}\ \tau'$. Then the result follows by Lemma 3.19.

  - T-Sub: See the proof of the T-Sub case within the case for E-App.

- E-Expr: Then $s = e$ and $< \sigma, e > \rightarrow v$ and $\sigma' = \sigma$. We're given that $\Gamma \sim \sigma$, and by Lemma 3.1 we have $\Gamma; \tau \vdash< \sigma, v >$, so the result follows by taking $\Gamma' = \Gamma$.

$\square$

Lemma 3.1 If $\Gamma \sim \sigma$ and $\Gamma \vdash e : \tau$ and $< \sigma, e > \rightarrow v$, then $\Gamma; \tau \vdash< \sigma, v >$.
Proof By induction on the depth of the derivation of $\Gamma \vdash e : \tau$. Case analysis of the last rule used in the derivation.

- T-Int: Then $e = c$ and $\tau = \texttt{int}$. Since $< \sigma, e > \rightarrow v$, by E-Val we have $v = c$. Then by Q-Int we have $\Gamma; \tau \vdash< \sigma, v >$.

- T-Unit: Then $e = ()$ and $\tau = \texttt{unit}$. Since $< \sigma, e > \rightarrow v$, by E-Val we have $v = ()$. Then by Q-Unit we have $\Gamma; \tau \vdash< \sigma, v >$.

- T-Var: Then $e = x$ and $\Gamma(x) = \tau$. Since $\Gamma \sim \sigma$, we have that $\text{dom}(\Gamma) = \text{dom}(\sigma)$, so x must be a location l. Since $< \sigma, e > \rightarrow v$, by E-Val we have $v = l$. Then by $\Gamma \sim \sigma$ we have $\Gamma; \tau \vdash< \sigma, l >$.

- T-DEREF: Then $e = !e'$ and $\Gamma \vdash e' : \texttt{ref}\ \tau$. Since $<\sigma, e> \rightarrow v$, by E-DEREF we have $<\sigma, e'> \rightarrow l$ and $\sigma(l) = v$. By induction we have $\Gamma; \texttt{ref}\ \tau \vdash <\sigma, l>$, so by Q-REF also $\Gamma; \tau \vdash <\sigma, v>$.

- T-QUALCASE: Then $\tau = q\tau'$ and $\Gamma \vdash e : \tau'$ and $\Gamma \vdash e_1 : q_1\tau_1 \ldots \Gamma \vdash e_n : q_n\tau_n$. By induction we have $\Gamma; \tau' \vdash <\sigma, v>$. Therefore, $\Gamma; \tau \vdash <\sigma, v>$ follows from Q-QUAL if we can show $[[q]](v)$.

  Consider one of the e_i subexpressions of e, and let v_i be some value. If it is not the case that $<\sigma, e_i> \rightarrow v_i$, then $[[q_i]](\sigma, e_i, v_i)$ holds trivially. Otherwise, if $<\sigma, e_i> \rightarrow v_i$, then by induction we have $\Gamma; q_i\tau_i \vdash <\sigma, v_i>$. Then by Q-QUAL we have $[[q_i]](v_i)$, so also $[[q_i]](\sigma, e_i, v_i)$ holds. Since we assume that T-QUALCASE is locally sound, and since we can find a v_i for each e_i such that $[[q_i]](\sigma, e_i, v_i)$ holds, by Definition 3.1 we have $[[q]](\sigma, e, v)$. Then since $<\sigma, e> \rightarrow v$, we have $[[q]](v)$.

- T-QUALVAL: Then $e = v'$ and $\tau = q\tau'$ and $[[q]](v')$ and $\Gamma \vdash v' : \tau'$. Since $<\sigma, e> \rightarrow v$, by E-VAL we have $v = v'$. So we have $[[q]](v)$ and $\Gamma \vdash v : \tau'$. By induction $\Gamma; \tau' \vdash <\sigma, v>$, and by Q-QUAL also $\Gamma; q\tau' \vdash <\sigma, v>$.

- T-SUB: Then $\Gamma \vdash e : \tau'$ and $\tau' \leq \tau$. By induction we have $\Gamma; \tau' \vdash <\sigma, v>$, and the result follows from Lemma 3.4.

$\square$

Lemma 3.2 If $\Gamma, x_0 : \tau_0 \vdash s : \tau$ and $\Gamma; \tau_0 \vdash <\sigma, v_0>$, then $\Gamma \vdash s[x_0 \mapsto v_0] : \tau$.
Proof By induction on the depth of the derivation of $\Gamma, x_0 : \tau_0 \vdash s : \tau$. Case analysis of the last rule used in the derivation.

- T-APP: Then $s = s_1\ s_2$ and $\Gamma, x_0 : \tau_0 \vdash s_1 : \tau_2 \rightarrow \tau$ and $\Gamma, x_0 : \tau_0 \vdash s_2 : \tau_2$. By induction we have $\Gamma \vdash s_1[x_0 \mapsto v_0] : \tau_2 \rightarrow \tau$ and $\Gamma \vdash s_2[x_0 \mapsto v_0] : \tau_2$, and the result follows by T-APP.

- T-LET: Then $s = \texttt{let}\ x = s_1\ \texttt{in}\ s_2$ and $\Gamma, x_0 : \tau_0 \vdash s_1 : \tau_1$ and $\Gamma, x_0 : \tau_0, x : \tau_1 \vdash s_2 : \tau$. By induction we have $\Gamma \vdash s_1[x_0 \mapsto v_0] : \tau_1$ and $\Gamma, x : \tau_1 \vdash s_2[x_0 \mapsto v_0] : \tau$, and the result follows by T-LET.

- T-REF: Then $s = \texttt{ref}\ s'$ and $\tau = \texttt{ref}\ \tau'$ and $\Gamma, x_0 : \tau_0 \vdash s' : \tau'$. By induction we have $\Gamma \vdash s'[x_0 \mapsto v_0] : \tau'$, and the result follows by T-REF.

- T-ASSGN: Then $s = s_1 := s_2$ and $\tau = \texttt{unit}$ and $\Gamma, x_0 : \tau_0 \vdash s_1 : \texttt{ref}\ \tau'$ and $\Gamma, x_0 : \tau_0 \vdash s_2 : \tau'$. By induction we have $\Gamma \vdash s_1[x_0 \mapsto v_0] : \texttt{ref}\ \tau'$ and $\Gamma \vdash s_2[x_0 \mapsto v_0] : \tau'$, and the result follows by T-ASSGN.

- T-INT: Then $s = c$ and $\tau = \texttt{int}$, and the result follows by T-INT.

- T-UNIT: Then $s = ()$ and $\tau = \texttt{unit}$, and the result follows by T-UNIT.

- T-VAR: Then $s = x$ and $(\Gamma, x_0 : \tau_0)(x) = \tau$. Suppose $x_0 = x$. Then $\tau_0 = \tau$ and $x[x_0 \mapsto v_0] = v_0$, so we must prove $\Gamma \vdash v_0 : \tau_0$. Since $\Gamma; \tau_0 \vdash <\sigma, v_0>$, the result follows by Lemma 3.3. Otherwise, suppose $x_0 \neq x$. Then $x[x_0 \mapsto v_0] = x$, so we must prove $\Gamma \vdash x : \tau$. Since $(\Gamma, x_0 : \tau_0)(x) = \tau$ and $x_0 \neq x$, also $\Gamma(x) = \tau$, so the result follows by T-VAR.

- T-FUN: Then $s = \lambda x.s'$ and $\tau = \tau_1 \rightarrow \tau_2$ and $\Gamma, x_0 : \tau_0, x : \tau_1 \vdash s' : \tau_2$. By induction we have $\Gamma, x : \tau_1 \vdash s'[x_0 \mapsto v_0] : \tau_2$, and the result follows by T-FUN.

- T-Deref: Then $s = !e$ and $\Gamma, x_0 : \tau_0 \vdash e : \texttt{ref}\ \tau$. By induction we have $\Gamma \vdash e[x_0 \mapsto v_0] : \texttt{ref}\ \tau$, and the result follows by T-Deref.

- T-QualCase: Then $\tau = q\tau'$ and $\Gamma, x_0 : \tau_0 \vdash e : \tau'$ and $\Gamma, x_0 : \tau_0 \vdash e_1 : q_1\tau_1 \ldots \Gamma, x_0 : \tau_0 \vdash e_n : q_n\tau_n$. By induction we have $\Gamma \vdash e[x_0 \mapsto v_0] : \tau'$ and $\Gamma \vdash e_1[x_0 \mapsto v_0] : q_1\tau_1 \ldots \Gamma \vdash e_n[x_0 \mapsto v_0] : q_n\tau_n$, and the result follows by T-QualCase.

- T-QualVal: Then $s = v$ and $\tau = q\tau'$ and $[[q]](v)$ and $\Gamma, x_0 : \tau_0 \vdash v : \tau'$. By induction also $\Gamma \vdash v[x_0 \mapsto v_0] : \tau'$. Since arrow types may not be qualified, τ' is not of the form $\overline{q}(\tau_1 \to \tau_2)$. Then by Lemma 3.10 v is not of the form $\lambda x.s'$. Therefore $v[x_0 \mapsto v_0] = v$, so $[[q]](v[x_0 \mapsto v_0])$. Then the result follows by T-QualVal.

- T-Sub: Then $\Gamma, x_0 : \tau_0 \vdash e : \tau'$ and $\tau' \leq \tau$. By induction we have $\Gamma \vdash e[x_0 \mapsto v_0] : \tau'$, and the result follows by T-Sub.

$\square$

Lemma 3.3 If $\Gamma; \tau \vdash < \sigma, v >$, then $\Gamma \vdash v : \tau$.
Proof By induction on the depth of the derivation of $\Gamma; \tau \vdash < \sigma, v >$. Case analysis of the last rule used in the derivation.

- Q-Int: Then $\tau = \texttt{int}$ and $v = c$, and the result follows by T-Int.

- Q-Unit: Then $\tau = \texttt{unit}$ and $v = ()$, and the result follows by T-Unit.

- Q-Fun: Then $v = \lambda x.s$ and $\Gamma \vdash \lambda x.s : \tau$, which is what we wanted to prove.

- Q-Ref: Then $v = l$ and $\Gamma \vdash l : \tau$, which is what we wanted to prove.

- Q-Qual: Then $\tau = q\tau'$ and $[[q]](v)$ and $\Gamma; \tau' \vdash < \sigma, v >$. By induction we have $\Gamma \vdash v : \tau'$. Then since $[[q]](v)$, by T-QualVal also $\Gamma \vdash v : \tau$.

$\square$

Lemma 3.4 If $\Gamma; \tau' \vdash < \sigma, v >$ and $\tau' \leq \tau$, then $\Gamma; \tau \vdash < \sigma, v >$.
Proof By induction on the derivation of $\tau' \leq \tau$. Case analysis of the last rule used in the derivation.

- SubValQual: Then $\tau' = q\tau$. Since $\Gamma; \tau' \vdash < \sigma, v >$, by Q-Qual we have $\Gamma; \tau \vdash < \sigma, v >$.

- SubQualReorder: Then $\tau' = q_1q_2\tau_0$ and $\tau = q_2q_1\tau_0$. Since $\Gamma; \tau' \vdash < \sigma, v >$, by Q-Qual we have $[[q_1]](v)$ and $\Gamma; q_2\tau_0 \vdash < \sigma, v >$. Then again by Q-Qual we have $[[q_2]](v)$ and $\Gamma; \tau_0 \vdash < \sigma, v >$. Therefore, by Q-Qual we have $\Gamma; q_1\tau_0 \vdash < \sigma, v >$, and again by Q-Qual we have $\Gamma; q_2q_1\tau_0 \vdash < \sigma, v >$.

- SubRef: Then $\tau' = \tau$ and the result follows.

- SubTrans: Then $\tau' \leq \tau''$ and $\tau'' \leq \tau$. By induction $\Gamma; \tau'' \vdash < \sigma, v >$, and by induction again $\Gamma; \tau \vdash < \sigma, v >$.

- SubFun: Then $\tau' = \tau_1' \to \tau_2'$ and $\tau = \tau_1 \to \tau_2$. Since $\Gamma; \tau' \vdash < \sigma, v >$, by Q-Fun we have that $v = \lambda x.s$ and $\Gamma \vdash \lambda x.s : \tau_1' \to \tau_2'$. Since $\tau' \leq \tau$, by T-Sub we have $\Gamma \vdash \lambda x.s : \tau_1 \to \tau_2$, so by Q-Fun $\Gamma; \tau_1 \to \tau_2 \vdash < \sigma, \lambda x.s >$.

$\square$

Lemma 3.5 If $\Gamma; \tau \vdash < \sigma, v >$ and $\Gamma' \supseteq \Gamma$ and $\sigma' \supseteq \sigma$, then $\Gamma'; \tau \vdash < \sigma', v >$.
Proof By induction on the depth of the derivation of $\Gamma; \tau \vdash < \sigma, v >$. Case analysis of the last rule used in the derivation.

- Q-Int: Then $\tau = \texttt{int}$ and $v = c$, and the result follows by Q-Int.
- Q-Unit: Then $\tau = \texttt{unit}$ and $v = ()$, and the result follows by Q-Unit.
- Q-Fun: Then $\tau = \tau_1 \rightarrow \tau_2$ and $v = \lambda x.s$ and $\Gamma \vdash \lambda x.s : \tau_1 \rightarrow \tau_2$. By Lemma 3.6 also $\Gamma' \vdash \lambda x.s : \tau_1 \rightarrow \tau_2$, and the result follows by Q-Fun.
- Q-Ref: Then $\tau = \texttt{ref}\ \tau'$ and $v = l$ and $\Gamma \vdash l : \texttt{ref}\ \tau'$ and $\Gamma; \tau' \vdash < \sigma, \sigma(l) >$ and $l \in \mathrm{dom}(\sigma)$. By Lemma 3.6 we have $\Gamma' \vdash l : \texttt{ref}\ \tau'$. Since $l \in \mathrm{dom}(\sigma)$ and $\sigma' \supseteq \sigma$, also $l \in \mathrm{dom}(\sigma')$ and $\sigma(l) = \sigma'(l)$. Finally, by induction $\Gamma'; \tau' \vdash < \sigma', \sigma'(l) >$, and the result follows by Q-Ref.
- Q-Qual: Then $\tau = q\tau'$ and $[[q]](v)$ and $\Gamma; \tau' \vdash < \sigma, v >$. By induction we have $\Gamma'; \tau' \vdash < \sigma', v >$, and the result follows by Q-Qual.

$\square$

Lemma 3.6 If $\Gamma \vdash s : \tau$ and $\Gamma' \supseteq \Gamma$, then $\Gamma' \vdash s : \tau$.
Proof By induction on the depth of the derivation of $\Gamma \vdash s : \tau$. Case analysis of the last rule used in the derivation.

- T-App: Then $s = s_1\ s_2$ and $\Gamma \vdash s_1 : \tau_2 \rightarrow \tau$ and $\Gamma \vdash s_2 : \tau_2$. By induction we have $\Gamma' \vdash s_1 : \tau_2 \rightarrow \tau$ and $\Gamma' \vdash s_2 : \tau_2$, and the result follows by T-App.
- T-Let: Then $s = \texttt{let}\ x = s_1\ \texttt{in}\ s_2$ and $\Gamma \vdash s_1 : \tau_1$ and $\Gamma, x : \tau_1 \vdash s_2 : \tau$. By induction we have $\Gamma' \vdash s_1 : \tau_1$ and $\Gamma', x : \tau_1 \vdash s_2 : \tau$, and the result follows by T-Let.
- T-Ref: Then $s = \texttt{ref}\ s'$ and $\tau = \texttt{ref}\ \tau'$ and $\Gamma \vdash s' : \tau'$. By induction we have $\Gamma' \vdash s' : \tau'$, and the result follows by T-Ref.
- T-Assgn: Then $s = s_1 := s_2$ and $\tau = \texttt{unit}$ and $\Gamma \vdash s_1 : \texttt{ref}\ \tau'$ and $\Gamma \vdash s_2 : \tau'$. By induction we have $\Gamma' \vdash s_1 : \texttt{ref}\ \tau'$ and $\Gamma' \vdash s_2 : \tau'$, and the result follows by T-Assgn.
- T-Int: Then $s = c$ and $\tau = \texttt{int}$, and the result follows by T-Int.
- T-Unit: Then $s = ()$ and $\tau = \texttt{unit}$, and the result follows by T-Unit.
- T-Var: Then $s = x$ and $\Gamma(x) = \tau$. Since $\Gamma' \supseteq \Gamma$, also $\Gamma'(x) = \tau$, and the result follows by T-Var.
- T-Fun: Then $s = \lambda x.s'$ and $\tau = \tau_1 \rightarrow \tau_2$ and $\Gamma, x : \tau_1 \vdash s' : \tau_2$. By induction we have $\Gamma', x : \tau_1 \vdash s' : \tau_2$, and the result follows by T-Fun.
- T-Deref: Then $s = !e$ and $\Gamma \vdash e : \texttt{ref}\ \tau$. By induction we have $\Gamma' \vdash e : \texttt{ref}\ \tau$, and the result follows by T-Deref.
- T-QualVal: Then $s = v$ and $\tau = q\tau'$ and $[[q]](v)$ and $\Gamma \vdash v : \tau'$. By induction also $\Gamma' \vdash v : \tau'$, and the result follows by T-QualVal.
- T-Sub: Then $\Gamma \vdash s : \tau'$ and $\tau' \leq \tau$. By induction we have $\Gamma' \vdash s : \tau'$, and the result follows by T-Sub.

- T-QualCase: Then $\tau = q\tau'$ and $\Gamma \vdash e : \tau'$ and $\Gamma \vdash e_1 : q_1\tau_1 \ldots \Gamma \vdash e_n : q_n\tau_n$. By induction we have $\Gamma' \vdash s : \tau'$ and $\Gamma' \vdash e_1 : q_1\tau_1 \ldots \Gamma' \vdash e_n : q_n\tau_n$, and the result follows by T-QualCase.

$\square$

Lemma 3.7 If $l \in \text{dom}(\sigma)$ and $< \sigma, s > \rightarrow < \sigma', v >$, then $l \in \text{dom}(\sigma')$.
Proof By induction on the depth of the derivation of $< \sigma, s > \rightarrow < \sigma', v >$. Case analysis of the last rule used in the derivation.

- E-App: Then $s = s_1\ s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, \lambda x.s' >$ and $< \sigma_1, s_2 > \rightarrow < \sigma_2, v_2 >$ and $< \sigma_2, s'[x \mapsto v_2] > \rightarrow < \sigma', v >$. By induction $l \in \text{dom}(\sigma_1)$. By induction again, $l \in \text{dom}(\sigma_2)$. By induction again, $l \in \text{dom}(\sigma')$.

- E-Let: Then $s = \texttt{let}\ x = s_1\ \texttt{in}\ s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, v_1 >$ and $< \sigma_1, s_2[x \mapsto v_1] > \rightarrow < \sigma', v >$. By induction $l \in \text{dom}(\sigma_1)$ and by induction again, $l \in \text{dom}(\sigma')$.

- E-Ref: Then $s = \texttt{ref}\ s'$ and $< \sigma, s' > \rightarrow < \sigma_1, v' >$ and $\sigma' = \sigma_1[l' \mapsto v']$. By induction $l \in \text{dom}(\sigma_1)$, so also $l \in \text{dom}(\sigma_1[l' \mapsto v'])$.

- E-Assgn: Then $s = s_1 := s_2$ and $< \sigma, s_1 > \rightarrow < \sigma_1, l' >$ $< \sigma_1, s_2 > \rightarrow < \sigma_2, v' >$ and $\sigma' = \sigma_2[l' \mapsto v']$. By induction $l \in \text{dom}(\sigma_1)$, by induction again $l \in \text{dom}(\sigma_2)$, so also $l \in \text{dom}(\sigma_2[l' \mapsto v'])$.

- E-Expr: Then $\sigma' = \sigma$, so since $l \in \text{dom}(\sigma)$, also $l \in \text{dom}(\sigma')$.

$\square$

Lemma 3.8 If $\Gamma \vdash l : \overline{q}(\texttt{ref}\ \tau)$, then there exists $\overline{q}'$ such that $\Gamma(l) = \overline{q}'(\texttt{ref}\ \tau)$.
Proof By induction on the depth of the derivation of $\Gamma \vdash l : \overline{q}(\texttt{ref}\ \tau)$. Case analysis of the last rule in the derivation.

- T-Var: Then $\Gamma(l) = \overline{q}(\texttt{ref}\ \tau)$, so the result follows, where $\overline{q}' = \overline{q}$.

- T-QualCase: Then $\overline{q} = q\overline{q}''$ and $\Gamma \vdash l : \overline{q}''(\texttt{ref}\ \tau)$, so by induction there exists $\overline{q}'$ such that $\Gamma(l) = \overline{q}'(\texttt{ref}\ \tau)$.

- T-QualVal: Then $\overline{q} = q\overline{q}''$ and $\Gamma \vdash l : \overline{q}''(\texttt{ref}\ \tau)$, so by induction there exists $\overline{q}'$ such that $\Gamma(l) = \overline{q}'(\texttt{ref}\ \tau)$.

- T-Sub: Then $\Gamma \vdash l : \tau'$ and $\tau' \leq \overline{q}(\texttt{ref}\ \tau)$. By Lemma 3.9 τ' has the form $\overline{q}''(\texttt{ref}\ \tau)$. Then by induction there exists $\overline{q}'$ such that $\Gamma(l) = \overline{q}'(\texttt{ref}\ \tau)$.

$\square$

Lemma 3.9 If $\tau_0 \leq \overline{q}(\texttt{ref}\ \tau)$, then τ_0 has the form $\overline{q}'(\texttt{ref}\ \tau)$.
Proof By induction on the depth of the derivation of $\tau_0 \leq \overline{q}(\texttt{ref}\ \tau)$. Case analysis of the last rule in the derivation.

- SubValQual: Then $\tau_0 = q\overline{q}(\texttt{ref}\ \tau)$, and the result follows.

- SubQualReorder: Then $\overline{q} = q_2 q_1 \overline{q}''$ and $\tau_0 = q_1 q_2 \overline{q}''(\texttt{ref}\ \tau)$, and the result follows.

- SubRef: Then $\tau_0 = \overline{q}(\texttt{ref}\ \tau)$, and the result follows.

- SubTrans: Then $\tau_0 \leq \tau'$ and $\tau' \leq \overline{q}(\texttt{ref}\ \tau)$. By induction τ' has the form $\overline{q}''(\texttt{ref}\ \tau)$. By induction again τ_0 has the form $\overline{q}'(\texttt{ref}\ \tau)$.

□

Lemma 3.10 If $\Gamma \vdash \lambda x.s : \tau$, then τ has the form $\overline{q}(\tau_1 \to \tau_2)$.
Proof By induction on the depth of the derivation of $\Gamma \vdash \lambda x.s : \tau$. Case analysis of the last rule in the derivation.

- T-Fun: Then τ has the form $\tau_1 \to \tau_2$ and the result is shown, with $\overline{q}$ being empty.
- T-QualCase: Then τ has the form $q\tau'$ and $\Gamma \vdash \lambda x.s : \tau'$. By induction τ' has the form $\overline{q}(\tau_1 \to \tau_2)$, so $\tau = q\overline{q}(\tau_1 \to \tau_2)$ and the result follows.
- T-QualVal: Then τ has the form $q\tau'$ and $\Gamma \vdash \lambda x.s : \tau'$. By induction τ' has the form $\overline{q}(\tau_1 \to \tau_2)$, so $\tau = q\overline{q}(\tau_1 \to \tau_2)$ and the result follows.
- T-Sub: Then $\Gamma \vdash \lambda x.s : \tau'$ and $\tau' \leq \tau$. By induction τ' has the form $\overline{q}(\tau_1 \to \tau_2)$ and the result follows from Lemma 3.11.

□

Lemma 3.11 If $\overline{q}(\tau_1 \to \tau_2) \leq \tau'$, then τ' has the form $\overline{q}'(\tau_1' \to \tau_2')$.
Proof By induction on the depth of the derivation of $\overline{q}(\tau_1 \to \tau_2) \leq \tau'$. Case analysis of the last rule used in the derivation.

- SubValQual: Then $\overline{q} = q\overline{q}'$ and $\tau' = \overline{q}'(\tau_1 \to \tau_2)$, so the result follows.
- SubQualReorder: Then $\overline{q} = q_1 q_2 \overline{q}'$ and $\tau' = q_2 q_1 \overline{q}'(\tau_1 \to \tau_2)$, so the result follows.
- SubRef: Then $\tau' = \overline{q}(\tau_1 \to \tau_2)$, so the result follows.
- SubTrans: Then $\overline{q}(\tau_1 \to \tau_2) \leq \tau''$ and $\tau'' \leq \tau'$. By induction τ'' has the form $\overline{q}''(\tau_1'' \to \tau_2'')$. Then by induction again, τ' has the form $\overline{q}'(\tau_1' \to \tau_2')$.
- SubFun: Then τ' has the form $\tau_1' \to \tau_2'$, so the result follows with $\overline{q}'$ as the empty sequence.

□

Lemma 3.12 If $\Gamma \vdash \lambda x.s : \overline{q}(\tau_1 \to \tau_2)$, then there exist τ_1' and τ_2' such that $\Gamma, x : \tau_1' \vdash s : \tau_2'$, where $\tau_1 \leq \tau_1'$ and $\tau_2' \leq \tau_2$.
Proof By induction on the depth of the derivation of $\Gamma \vdash \lambda x.s : \overline{q}(\tau_1 \to \tau_2)$. Case analysis of the last rule used in the derivation.

- T-Fun: Then $\overline{q}$ is empty and $\Gamma, x : \tau_1 \vdash s : \tau_2$. By SubRef we have $\tau_1 \leq \tau_1$ and $\tau_2 \leq \tau_2$, so the result follows.
- T-QualCase: Then $\overline{q} = q\overline{q}'$ and $\Gamma \vdash \lambda x.s : \overline{q}'(\tau_1 \to \tau_2)$, so the result follows by induction.
- T-QualVal: Then $\overline{q} = q\overline{q}'$ and $\Gamma \vdash \lambda x.s : \overline{q}''(\tau_1 \to \tau_2)$, so the result follows by induction.
- T-Sub: Then $\Gamma \vdash \lambda x.s : \tau'$ and $\tau' \leq \overline{q}(\tau_1 \to \tau_2)$. By Lemma 3.13 τ' has the form $\overline{q}'(\tau_1'' \to \tau_2'')$, where $\tau_1 \leq \tau_1''$ and $\tau_2'' \leq \tau_2$. By induction $\Gamma, x : \tau_1' \vdash s : \tau_2'$, where $\tau_1'' \leq \tau_1'$ and $\tau_2' \leq \tau_2''$. Then by SubTrans also $\tau_1 \leq \tau_1'$ and $\tau_2' \leq \tau_2$, so the result follows.

□

Lemma 3.13 If $\tau' \leq \overline{q}(\tau_1 \rightarrow \tau_2)$, then τ' has the form $\overline{q}'(\tau_1' \rightarrow \tau_2')$, where $\tau_1 \leq \tau_1'$ and $\tau_2' \leq \tau_2$.
Proof By induction on the depth of the derivation of $\tau' \leq \overline{q}(\tau_1 \rightarrow \tau_2)$. Case analysis of the last rule used in the derivation.

- SubValQual: Then $\tau' = q\overline{q}(\tau_1 \rightarrow \tau_2)$. By SubRef we have $\tau_1 \leq \tau_1$ and $\tau_2 \leq \tau_2$, so the result follows.

- SubQualReorder: Then $\overline{q} = q_2 q_1 \overline{q}'$ and $\tau' = q_1 q_2 \overline{q}'(\tau_1 \rightarrow \tau_2)$. By SubRef we have $\tau_1 \leq \tau_1$ and $\tau_2 \leq \tau_2$, so the result follows.

- SubRef: Then $\tau' = \overline{q}(\tau_1 \rightarrow \tau_2)$. By SubRef we have $\tau_1 \leq \tau_1$ and $\tau_2 \leq \tau_2$, so the result follows.

- SubTrans: Then $\tau' \leq \tau''$ and $\tau'' \leq \overline{q}(\tau_1 \rightarrow \tau_2)$. By induction τ'' has the form $\overline{q}''(\tau_1'' \rightarrow \tau_2'')$, where $\tau_1 \leq \tau_1''$ and $\tau_2'' \leq \tau_2$. Then by induction again, τ' has the form $\overline{q}'(\tau_1' \rightarrow \tau_2')$, where $\tau_1'' \leq \tau_1'$ and $\tau_2' \leq \tau_2''$. Then by SubTrans we have $\tau_1 \leq \tau_1'$ and $\tau_2' \leq \tau_2$, so the result follows.

- SubFun: Then $\overline{q}$ is empty and $\tau' = \tau_1' \rightarrow \tau_2'$, where $\tau_1 \leq \tau_1'$ and $\tau_2' \leq \tau_2$, so the result follows with $\overline{q}'$ as the empty sequence.

□

Lemma 3.14 If $\Gamma; \tau' \vdash < \sigma, v >$ and $\Gamma; \texttt{ref}\ \tau \vdash < \sigma, l >$ and τ' is a component of τ, then $\sigma \vdash v \not\hookrightarrow l$.
Proof By induction on the depth of the derivation of $\Gamma; \tau' \vdash < \sigma, v >$. Case analysis of the last rule used in the derivation.

- Q-Int: Then $v = c$ and the result follows from UnreachInt.

- Q-Unit: Then $v = ()$ and the result follows from UnreachUnit.

- Q-Fun: Then $v = \lambda x.s$ and the result follows from UnreachFun.

- Q-Ref: Then $v = l'$ and $\tau' = \texttt{ref}\ \tau''$ and $\Gamma \vdash l' : \texttt{ref}\ \tau''$ and $\Gamma; \tau'' \vdash < \sigma, \sigma(l') >$ and $l' \in \mathrm{dom}(\sigma)$.

  First we show that $l' \neq l$. Suppose not, so $l' = l$. Since $\Gamma \vdash l' : \texttt{ref}\ \tau''$, by Lemma 3.8 $\Gamma(l') = \Gamma(l)$ has the form $\overline{q}(\texttt{ref}\ \tau'')$. We're given that $\Gamma; \texttt{ref}\ \tau \vdash < \sigma, l >$, so by Q-Ref we have $\Gamma \vdash l : \texttt{ref}\ \tau$, so again by Lemma 3.8 $\Gamma(l)$ also has the form $\overline{q}'(\texttt{ref}\ \tau)$. Therefore, it must be the case that $\tau = \tau''$. But we know that $\texttt{ref}\ \tau''$ is a component of τ, so we have a contradiction.

  Since $\texttt{ref}\ \tau''$ is a component of τ, so is τ''. Since $\Gamma; \tau'' \vdash < \sigma, \sigma(l') >$ and $\Gamma; \texttt{ref}\ \tau \vdash < \sigma, l >$, by induction we have $\sigma \vdash \sigma(l') \not\hookrightarrow l$. Therefore, we have shown $l' \neq l$ and $l' \in \mathrm{dom}(\sigma)$ and $\sigma \vdash \sigma(l') \not\hookrightarrow l$, so by UnreachLoc we have $\sigma \vdash l' \not\hookrightarrow l$.

- Q-Qual: Then $\tau' = q\tau''$ and $\Gamma; \tau'' \vdash < \sigma, v >$. Since τ' is a component of τ, so is τ''. Then by induction we have $\sigma \vdash v \not\hookrightarrow l$.

□

Lemma 3.15 If $\Gamma; \tau \vdash < \sigma, v >$ and $\sigma \vdash v \not\hookrightarrow l$, then $\Gamma; \tau \vdash < \sigma[l \mapsto v'], v >$.
Proof By induction on the depth of the derivation of $\Gamma; \tau \vdash < \sigma, v >$. Case analysis of the last rule used in the derivation.

- Q-Int: Then $\tau = \texttt{int}$ and $v = c$, and the result follows by Q-Int.

- Q-Unit: Then $\tau = \texttt{unit}$ and $v = ()$, and the result follows by Q-Unit.

- Q-Fun: Then $\tau = \tau_1 \rightarrow \tau_2$ and $v = \lambda x.s$ and $\Gamma \vdash \lambda x.s : \tau_1 \rightarrow \tau_2$. Then the result follows by Q-Fun.

- Q-Ref: Then $\tau = \texttt{ref}\ \tau'$ and $v = l'$ and $\Gamma \vdash l' : \texttt{ref}\ \tau'$ and $\Gamma; \tau' \vdash< \sigma, \sigma(l') >$ and $l' \in \mathrm{dom}(\sigma)$. Then also $l' \in \mathrm{dom}(\sigma[l \mapsto v'])$. The result follows by Q-Ref if we can prove $\Gamma; \tau' \vdash< \sigma[l \mapsto v'], \sigma[l \mapsto v'](l') >$. Since $\sigma \vdash l' \not\rightarrow l$ and $l' \in \mathrm{dom}(\sigma)$, by UnreachLoc we have $l' \neq l$ and $\sigma \vdash \sigma(l') \not\rightarrow l$. Since $\Gamma; \tau' \vdash< \sigma, \sigma(l') >$ and $\sigma \vdash \sigma(l') \not\rightarrow l$, by induction $\Gamma; \tau' \vdash< \sigma[l \mapsto v'], \sigma(l') >$. Since $l' \neq l$, $\sigma(l') = \sigma[l \mapsto v'](l')$, so we have proven $\Gamma; \tau' \vdash< \sigma[l \mapsto v'], \sigma[l \mapsto v'](l') >$.

- Q-Qual: Then $\tau = q\tau'$ and $[[q]](v)$ and $\Gamma; \tau' \vdash< \sigma, v >$. By induction we have $\Gamma'; \tau' \vdash< \sigma[l \mapsto v'], v >$, and the result follows by Q-Qual.

□

Lemma 3.16 If $\Gamma \sim \sigma$ and it is not the case that $\sigma \vdash v \not\rightarrow l$, then there exists a location l' such that $v = l'$ and a nonnegative integer k such that $\sigma^k(l') = l$.

Proof We prove this lemma by induction on the *depth* of v, which we define as follows. If v is not a location, then $\mathrm{depth}(v) = 0$. Otherwise v has the form l'. If $l' \notin \mathrm{dom}(\sigma)$ then $\mathrm{depth}(l') = 0$. Otherwise $l' \in \mathrm{dom}(\sigma)$. Since $\Gamma \sim \sigma$, by Lemma 3.24 we have $\sigma \vdash \sigma(l') \not\rightarrow l'$. Then by Lemma 3.17 there exists a positive integer k' and a value v' such that $\sigma^{k'}(l') = v'$, where v' is not a location, and $\mathrm{depth}(l')$ is defined to be k'. Note that k' is unique.

- v has depth 0: Then either v is not a location or $v = l'$ and $l' \notin \mathrm{dom}(\sigma)$. If v is not a location, then it is either an integer constant c, the unit value $()$, or a function value $\lambda x.s$. But then $\sigma \vdash v \not\rightarrow l$ by UnreachInt, UnreachUnit, and UnreachFun, contradicting our initial assumptions. Therefore $v = l'$ and $l' \notin \mathrm{dom}(\sigma)$. Since it is not the case that $\sigma \vdash l' \not\rightarrow l$, by UnreachLoc we have that either $l' = l$ or $l' \in \mathrm{dom}(\sigma)$ and it is not the case that $\sigma \vdash \sigma(l') \not\rightarrow l$. Therefore $l' = l$, so $\sigma^0(l') = l$ and the result follows with $k = 0$.

- v has depth $d > 0$: Then $v = l'$ and $l' \in \mathrm{dom}(\sigma)$ and there exists a value v' such that $\sigma^d(l') = v'$, where v' is not a location. Since it is not the case that $\sigma \vdash l' \not\rightarrow l$, by UnreachLoc we have that either $l' = l$ or $l' \in \mathrm{dom}(\sigma)$ and it is not the case that $\sigma \vdash \sigma(l') \not\rightarrow l$. If $l' = l$ then $\sigma^0(l') = l$ and the result follows with $k = 0$. Otherwise $l' \in \mathrm{dom}(\sigma)$ and it is not the case that $\sigma \vdash \sigma(l') \not\rightarrow l$. If we can show that $\sigma(l')$ has a smaller depth than l', then by induction we have that there exists a location l'' such that $\sigma(l') = l''$ and a nonnegative integer k'' such that $\sigma^{k''}(l'') = l$, so $\sigma^{k''+1}(l') = l$ and the result follows.

  To see that $\sigma(l')$ has smaller depth than l', we analyze the form of $\sigma(l')$. If it is a value other than a location or it is a location that is not in $\mathrm{dom}(\sigma)$, then $\mathrm{depth}(\sigma(l')) = 0$. Since the depth of l' is $d > 0$, the result follows. Otherwise, $\sigma(l')$ is some location $l'' \in \mathrm{dom}(\sigma)$. Since there exists a value v' such that $\sigma^d(l') = v'$, where v' is not a location, also $\sigma^{d-1}(l'') = v'$, so the depth of l'' is $d - 1$, which is smaller than d.

□

Lemma 3.17 If $\Gamma \sim \sigma$ and $l \in \mathrm{dom}(\sigma)$ and $\sigma \vdash \sigma^k(l) \not\rightarrow l$ for some nonnegative integer k, then there exists a positive integer k' and a value v such that $\sigma^{k'}(l) = v$, where v is not a location.

Proof By induction on the depth of the derivation of $\sigma \vdash \sigma^k(l) \not\hookrightarrow l$. Case analysis of the last rule used in the derivation.

- UNREACHINT, UNREACHUNIT, or UNREACHFUN: Then $\sigma^k(l)$ is not a location. Since l is a location, $k > 0$, so the result follows with $k' = k$.

- UNREACHLOC: Then $\sigma^k(l) = l'$ and $l' \neq l$ and $l' \in \text{dom}(\sigma) \Rightarrow \sigma \vdash \sigma(l') \not\hookrightarrow l$. Then by Lemma 3.18 $l' \in \text{dom}(\sigma)$. Therefore we have $\sigma \vdash \sigma(l') \not\hookrightarrow l$, or equivalently $\sigma \vdash \sigma^{k+1}(l) \not\hookrightarrow l$. Then by induction there exists a positive integer k' and a value v such that $\sigma^{k'}(l) = v$, where v is not a location.

$\square$

Lemma 3.18 If $\Gamma \sim \sigma$ and $l \in \text{dom}(\sigma)$ and $\sigma^k(l) = l'$ for some nonnegative integer k, then $l' \in \text{dom}(\sigma)$.

Proof By induction on k.

- $k = 0$: Then $l = l'$ and since $l \in \text{dom}(\sigma)$ also $l' \in \text{dom}(\sigma)$.

- $k > 0$: Let $\sigma^{k-1}(l) = l''$. By induction $l'' \in \text{dom}(\sigma)$. Since $\Gamma \sim \sigma$, we have $\Gamma; \Gamma(l'') \vdash < \sigma, l'' >$. By Lemma 3.22, $\Gamma(l'')$ has the form $\overline{q}(\texttt{ref}\ \tau)$, and by Lemma 3.23 we have $\Gamma; \tau \vdash < \sigma, l' >$. By Lemma 3.22 again, τ has the form $\overline{q}'(\texttt{ref}\ \tau')$, and by Lemma 3.23 again we have $l' \in \text{dom}(\sigma)$.

$\square$

Lemma 3.19 If $\Gamma_2 \sim \sigma_2$ and $\Gamma_2; \Gamma_2(l) \vdash < \sigma_2, l >$ and $\sigma_2^k(l) = l_1$ for some nonnegative integer k and $\sigma' = \sigma_2[l_1 \mapsto v_2]$ and $\Gamma_2; \tau' \vdash < \sigma_2, v_2 >$ and $l_1 \in \text{dom}(\sigma_2)$ and $\Gamma_2 \vdash l_1 : \texttt{ref}\ \tau'$, then $\Gamma_2; \Gamma_2(l) \vdash < \sigma', l >$.

Proof Assume WLOG that k is the smallest nonnegative integer such that $\sigma_2^k(l) = l_1$. We prove this lemma by induction on k.

- $k = 0$: Then $l = l_1$, so we must show $\Gamma_2; \Gamma_2(l_1) \vdash < \sigma', l_1 >$. Since $\Gamma_2 \vdash l_1 : \texttt{ref}\ \tau'$, by Lemma 3.8, there exists some $\overline{q}$ such that $\Gamma_2(l_1) = \overline{q}(\texttt{ref}\ \tau')$. Since $\Gamma_2 \sim \sigma_2$, we know that $\Gamma_2; \overline{q}(\texttt{ref}\ \tau') \vdash < \sigma_2, l_1 >$. Then by Lemma 3.20 also $\Gamma_2; \overline{q}(\texttt{ref}\ \tau') \vdash < \sigma', l_1 >$.

- $k > 0$: Since $\Gamma_2; \Gamma_2(l) \vdash < \sigma_2, l >$, by Lemma 3.22 $\Gamma_2(l)$ has the form $\overline{q}(\texttt{ref}\ \tau'')$. Then the result follows by Lemma 3.21.

$\square$

Lemma 3.20 If $\Gamma_2; \overline{q}(\texttt{ref}\ \tau') \vdash < \sigma_2, l_1 >$ and $\Gamma_2; \tau' \vdash < \sigma_2, v_2 >$ and $\sigma' = \sigma_2[l_1 \mapsto v_2]$ and $l_1 \in \text{dom}(\sigma_2)$, then $\Gamma_2; \overline{q}(\texttt{ref}\ \tau') \vdash < \sigma', l >$.

Proof By induction on the length of $\overline{q}$.

- $\overline{q}$ has length 0: So $\Gamma_2; \texttt{ref}\ \tau' \vdash < \sigma_2, l_1 >$. Then by Q-REF, $\Gamma_2 \vdash l_1 : \texttt{ref}\ \tau'$. Since $\sigma' = \sigma_2[l_1 \mapsto v_2]$, we have $l_1 \in \text{dom}(\sigma')$. Since $\sigma'(l_1) = v_2$, if we can show that $\Gamma_2; \tau' \vdash < \sigma', v_2 >$, then by Q-REF we have $\Gamma_2; \texttt{ref}\ \tau' \vdash < \sigma', l_1 >$, which is what we are trying to prove.

  Since $\Gamma_2; \tau' \vdash < \sigma_2, v_2 >$ and $\Gamma_2; \texttt{ref}\ \tau' \vdash < \sigma_2, l_1 >$, by Lemma 3.14 we have $\sigma_2 \vdash v_2 \not\hookrightarrow l_1$. Then by Lemma 3.15 we have $\Gamma_2; \tau' \vdash < \sigma_2[l_1 \mapsto v_2], v_2 >$, or equivalently $\Gamma_2; \tau' \vdash < \sigma', v_2 >$.

- $\overline{q}$ has length greater than zero: So $\overline{q}$ has the form $q\overline{q}'$ and $\Gamma_2; q\overline{q}'(\texttt{ref}\ \tau') \vdash < \sigma_2, l_1 >$. By Q-QUAL we have $[[q]](l_1)$ and $\Gamma_2; \overline{q}'(\texttt{ref}\ \tau') \vdash < \sigma_2, l_1 >$. By induction $\Gamma_2; \overline{q}'(\texttt{ref}\ \tau') \vdash < \sigma', l_1 >$, and by Q-QUAL also $\Gamma_2; q\overline{q}'(\texttt{ref}\ \tau') \vdash < \sigma', l_1 >$.

$\square$

Lemma 3.21 If $\Gamma_2 \sim \sigma_2$ and $\Gamma_2; \overline{q}(\texttt{ref}\ \tau'') \vdash < \sigma_2, l >$ and $\sigma_2^k(l) = l_1$ for some positive integer k and $\sigma' = \sigma_2[l_1 \mapsto v_2]$ and Lemma 3.19 holds for all nonnegative integers i such that $0 \leq i < k$, then $\Gamma_2; \overline{q}(\texttt{ref}\ \tau'') \vdash < \sigma', l >$.

Proof We prove this lemma by induction on the length of $\overline{q}$.

- $\overline{q}$ has length 0: So $\Gamma_2; \texttt{ref}\ \tau'' \vdash < \sigma_2, l >$. By Q-Ref, $\Gamma_2 \vdash l : \texttt{ref}\ \tau''$ and $l \in \mathrm{dom}(\sigma_2)$ and $\Gamma_2; \tau'' \vdash < \sigma_2, \sigma_2(l) >$. Since $k > 0$ we have that $l \neq l_1$, so $\sigma_2(l) = \sigma'(l)$ and $\Gamma_2; \tau'' \vdash < \sigma_2, \sigma'(l) >$. Since $l \in \mathrm{dom}(\sigma_2)$, by definition of σ' we have $l \in \mathrm{dom}(\sigma')$. Then the result holds by Q-Ref if we can show that $\Gamma_2; \tau'' \vdash < \sigma', \sigma'(l) >$.

  Since $\sigma_2^k(l) = l_1$ we have $\sigma_2^{k-1}(\sigma_2(l)) = l_1$, and $\sigma_2(l)$ must be some location l'. Then we have $\Gamma_2; \tau'' \vdash < \sigma_2, l' >$, so by Lemma 3.22 τ'' has the form $\overline{q}_0(\texttt{ref}\ \tau_0)$, and by Lemma 3.23 we have $l' \in \mathrm{dom}(\sigma_2)$. Since $\Gamma_2 \sim \sigma_2$, also $l' \in \mathrm{dom}(\Gamma_2)$ and we have $\Gamma_2; \Gamma_2(l') \vdash < \sigma_2, l' >$. Therefore by induction on Lemma 3.19 we have $\Gamma_2; \Gamma_2(l') \vdash < \sigma', l' >$.

  Since $\Gamma_2; \tau'' \vdash < \sigma_2, l' >$, by Lemma 3.3 we have $\Gamma_2 \vdash l' : \overline{q}_0(\texttt{ref}\ \tau_0)$. Then by Lemma 3.8 we have $\Gamma_2(l') = \overline{q}_0'(\texttt{ref}\ \tau_0)$. Then since $\Gamma_2; \Gamma_2(l') \vdash < \sigma', l' >$, by Lemma 3.25 we have $\Gamma_2; \texttt{ref}\ \tau_0 \vdash < \sigma', l' >$. Finally, since $\Gamma_2; \overline{q}_0(\texttt{ref}\ \tau_0) \vdash < \sigma_2, l' >$ by Lemma 3.26 we also $\Gamma_2; \overline{q}_0(\texttt{ref}\ \tau_0) \vdash < \sigma', l' >$, which is what we were trying to prove.

- $\overline{q}$ has length greater than zero, so $\overline{q}$ has the form $q\overline{q}'$ and $\Gamma_2; q\overline{q}'(\texttt{ref}\ \tau'') \vdash < \sigma_2, l >$. By Q-Qual we have $[[q]](l)$ and $\Gamma_2; \overline{q}'(\texttt{ref}\ \tau'') \vdash < \sigma_2, l >$. By induction $\Gamma_2; \overline{q}'(\texttt{ref}\ \tau'') \vdash < \sigma', l >$, and by Q-Qual also $\Gamma_2; q\overline{q}'(\texttt{ref}\ \tau'') \vdash < \sigma', l >$ as desired.

$\square$

Lemma 3.22 If $\Gamma; \tau \vdash < \sigma, l >$ then τ has the form $\overline{q}(\texttt{ref}\ \tau')$.

Proof By induction on the depth of the derivation of $\Gamma; \tau \vdash < \sigma, l >$. Case analysis of the last rule used in the derivation.

- Q-Ref: Then τ has the form $\texttt{ref}\ \tau'$, so the result follows with $\overline{q}$ being empty.
- Q-Qual: Then $\tau = q\tau''$ and $\Gamma; \tau'' \vdash < \sigma, l >$. By induction τ'' has the form $\overline{q}'(\texttt{ref}\ \tau')$, so τ has the form $\overline{q}(\texttt{ref}\ \tau')$, where $\overline{q} = q\overline{q}'$.

$\square$

Lemma 3.23 If $\Gamma; \overline{q}(\texttt{ref}\ \tau) \vdash < \sigma, l >$, then $l \in \mathrm{dom}(\sigma)$ and $\Gamma; \tau \vdash < \sigma, \sigma(l) >$.

Proof By induction on the depth of the derivation of $\Gamma; \overline{q}(\texttt{ref}\ \tau) \vdash < \sigma, l >$. Case analysis of the last rule used in the derivation.

- Q-Ref: Then $\overline{q}$ is empty and $l \in \mathrm{dom}(\sigma)$ and $\Gamma; \tau \vdash < \sigma, \sigma(l) >$.
- Q-Qual: Then $\overline{q} = q\overline{q}'$ and $\Gamma; \overline{q}'(\texttt{ref}\ \tau) \vdash < \sigma, l >$. By induction, $l \in \mathrm{dom}(\sigma)$ and $\Gamma; \tau \vdash < \sigma, \sigma(l) >$.

$\square$

Lemma 3.24 If $\Gamma \sim \sigma$ and $l \in \mathrm{dom}(\sigma)$ then $\sigma \vdash \sigma(l) \not\rightsquigarrow l$.

Proof Since $\Gamma \sim \sigma$ and $l \in \mathrm{dom}(\sigma)$, also $l \in \mathrm{dom}(\Gamma)$. Then since $\Gamma \sim \sigma$, we have $\Gamma; \Gamma(l) \vdash < \sigma, l >$. Then by Lemma 3.22 $\Gamma(l)$ has the form $\overline{q}(\texttt{ref}\ \tau)$, and by Lemma 3.23 also $\Gamma; \tau \vdash < \sigma, \sigma(l) >$. Then since τ is a component of $\overline{q}(\texttt{ref}\ \tau)$, by Lemma 3.14 we have $\sigma \vdash \sigma(l) \not\rightsquigarrow l$. $\square$

Lemma 3.25 If $\Gamma; \overline{q}\tau \vdash < \sigma, v >$ then $\Gamma; \tau \vdash < \sigma, v >$.
Proof By induction on the length of $\overline{q}$.

- $\overline{q}$ has length 0: Then $\overline{q}\tau = \tau$ and the result follows.
- $\overline{q}$ has length $k > 0$: Then $\overline{q} = q\overline{q}'$. Since $\Gamma; \overline{q}\tau \vdash < \sigma, v >$, by Q-Qual we have $\Gamma; \overline{q}'\tau \vdash < \sigma, v >$, and the result follows by induction.

$\square$

Lemma 3.26 If $\Gamma; \tau \vdash < \sigma, v >$ and $\Gamma; \overline{q}\tau \vdash < \sigma', v >$, then $\Gamma; \overline{q}\tau \vdash < \sigma, v >$.
Proof By induction on the length of $\overline{q}$.

- $\overline{q}$ has length 0: Then $\overline{q}\tau = \tau$ and the result follows.
- $\overline{q}$ has length $k > 0$: Then $\overline{q} = q\overline{q}'$. Since $\Gamma; \overline{q}\tau \vdash < \sigma', v >$, by Q-Qual we have $[[q]](v)$ and $\Gamma; \overline{q}'\tau \vdash < \sigma', v >$. Then by induction we have $\Gamma; \overline{q}'\tau \vdash < \sigma, v >$, and the result follows by Q-Qual.

$\square$