A. Proofs

For simplicity, we only consider the unary and binary unlifting functions.

A.1 Free Theorem

Roughly speaking, free theorems are theorems obtained for free as corollaries of relational parametricity [2], which states that, for a closed term f of type T, (f, f) belongs to a certain relational interpretation of T. A simple example of a free theorem is that f of type $\forall a.\,[a] \to [a]$ satisfies $map\ g \circ f = f \circ map\ g$ for any function g of type $A \to B$. In addition to the binary relational parametricity, which is used in the original free theorems [6] and semantic bidirectionalization [3], we also use the unary version—it states that a closed term f of type T belongs to a certain unary-relational interpretation of T. Many of the proofs in this paper is based on the unary parametricity.

We start by introducing some notations. We write $\mathcal{R} :: \mathsf{Pred}(A)$ if $\mathcal{R}$ is a unary relation (i.e., predicate) on A, and $\mathcal{R} :: A_1 \leftrightarrow A_2$ if $\mathcal{R}$ is a binary relation between A_1 and A_2. For unary relations $\mathcal{R} :: \mathsf{Pred}(A)$ and $\mathcal{R}' :: \mathsf{Pred}(B)$, we write $\mathcal{R} \to \mathcal{R}' :: \mathsf{Pred}(A \to B)$ for the relation $\{(f, g) \mid \forall x \in \mathcal{R}.\ f\,x \in \mathcal{R}'\}$, and $(\mathcal{R}, \mathcal{R}') :: \mathsf{Pred}((A, B))$ for the relation $\{(x, y) \mid x \in \mathcal{R}, y \in \mathcal{R}'\}$. Similarly, for relations $\mathcal{R} :: A_1 \leftrightarrow A_2$ and $\mathcal{R}' :: B_1 \leftrightarrow B_2$, we write $\mathcal{R} \to \mathcal{R}' :: (A_1 \to B_1) \leftrightarrow (A_2 \to B_2)$ for the relation $\{(f_1, f_2) \mid \forall (x_1, x_2) \in \mathcal{R}.\,(f_1\,x_1, f_2\,x_2) \in \mathcal{R}'\}$, and $(\mathcal{R}, \mathcal{R}') :: (A_1, B_1) \leftrightarrow (A_2, B_2)$ for $\{((x_1, y_1), (x_2, y_2)) \mid (x_1, x_2) \in \mathcal{R}, (y_1, y_2) \in \mathcal{R}'\}$. For a polymorphic term f of type $\forall a.\ T$ and a type S, we write f_S for the instantiation of f to S, which has type $T[S/a]$. For simplicity, we sometimes omit the subscript and simply write f for f_S if S is clear from the context or irrelevant.

We introduce a unary/binary *relational interpretation* $[\![\tau]\!]^1_\rho$/$[\![\tau]\!]^2_\rho$ of types, where ρ is a mapping from type variable to unary/binary relations, as follows.

$$
\begin{aligned}
[\![a]\!]^1_\rho &= \rho(a)\\
[\![B]\!]^1_\rho &= \{e \mid e :: B\} \quad \text{if } B \text{ is a base type}\\
[\![T_1 \to T_2]\!]^1_\rho &= [\![T_1]\!]^1_\rho \to [\![T_2]\!]^1_\rho\\
[\![(T_1, T_2)]\!]^1_\rho &= ([\![T_1]\!]^1_\rho, [\![T_2]\!]^1_\rho)\\
[\![\forall a.\ T]\!]^1_\rho &= \left\{ u \,\middle|\, \forall \mathcal{R} :: \mathsf{Pred}(S).\ u_S \in [\![T]\!]^1_{\rho[\alpha \to \mathcal{R}]} \right\}.\\
[\![a]\!]^2_\rho &= \rho(a)\\
[\![B]\!]^2_\rho &= \{(e, e) \mid e :: B\} \quad \text{if } B \text{ is a base type}\\
[\![T_1 \to T_2]\!]^2_\rho &= [\![T_1]\!]^2_\rho \to [\![T_2]\!]^2_\rho\\
[\![(T_1, T_2)]\!]^2_\rho &= ([\![T_1]\!]^2_\rho, [\![T_2]\!]^2_\rho)\\
[\![\forall a.\ T]\!]^2_\rho &= \left\{ (u, v) \,\middle|\, \forall \mathcal{R} :: S_1 \leftrightarrow S_2.\ (u_{S_1}, v_{S_2}) \in [\![T]\!]_{\rho[a \mapsto \mathcal{R}]} \right\}
\end{aligned}
$$

Here, $\rho[a \mapsto \mathcal{R}]$ is an extension of ρ with $a \mapsto \mathcal{R}$. If $\rho = \emptyset$, we sometimes write $[\![T]\!]^1$/$[\![T]\!]^2$ instead of $[\![T]\!]^1_\emptyset$/$[\![T]\!]^2_\emptyset$. We abuse the notation to write $[\![\forall\alpha.\tau]\!]^1$/$[\![\forall\alpha.\tau]\!]^2$ as $\forall\mathcal{R}.\mathcal{F}$ where $\mathcal{F}$ is the interpretation $[\![\tau]\!]^1_{\{\alpha \mapsto \mathcal{R}\}}$/$[\![\tau]\!]^2_{\{\alpha \mapsto \mathcal{R}\}}$ depending on the context. For example, we write $\forall\mathcal{R}.\forall\mathcal{S}.\,\mathcal{R} \to \mathcal{S}$ for $[\![\forall\alpha.\forall\beta.\alpha \to \beta]\!]$. For a base type B, we also write B for $[\![B]\!]^1$/$[\![B]\!]^2$ depending on the context.

Then, parametricity states that, for a closed term f of a closed type τ, f is in $[\![\tau]\!]^1$ and (f, f) is in $[\![\tau]\!]^2$. Free theorems are theorems obtained by instantiating parametricity.

Voigtländer [4] extends parametricity to a type system with type constructors. A key notion in his result is *relational action*.

Definition 4 ((Binary) Relational Action [4])**.** For type constructors κ_1 and κ_2, $\mathcal{F}$ is called a *relational action* between κ_1 and κ_2, denoted by $\mathcal{F} :: \kappa_1 \leftrightarrow \kappa_2$, if $\mathcal{F}$ maps any relation $\mathcal{R} :: \tau_1 \leftrightarrow \tau_2$ for every closed type τ_1 and τ_2 to $\mathcal{F}\,\mathcal{R} :: \kappa_1\,\tau_1 \leftrightarrow \kappa_2\,\tau_2$. □

Similarly, we define a unary version of the relational action.

Definition 5 ((Unary) Relational Action)**.** For a type constructor κ, $\mathcal{F}$ is called a *relational action* on κ, denoted by $\mathcal{F} :: \mathsf{Pred}(\kappa)$, if $\mathcal{F}$ maps any relation $\mathcal{R} :: \mathsf{Pred}(\tau)$ for every closed type τ to $\mathcal{F}\,\mathcal{R} :: \mathsf{Pred}(\kappa\,\tau)$.

Accordingly, the relational interpretations are extended as:

$$
\begin{aligned}
[\![\kappa]\!]^1_\rho &= \rho(\kappa)\\
[\![\tau_1\,\tau_2]\!]^1_\rho &= [\![\tau_1]\!]^1_\rho\,[\![\tau_2]\!]^1_\rho\\
[\![\forall\kappa.\tau]\!]^1 &= \left\{ u \,\middle|\, \forall \mathcal{F} : \mathsf{Pred}(\kappa)\ u_\kappa.\ \in [\![\tau]\!]^1_{\rho[\kappa \mapsto \mathcal{F}]} \right\}\\
[\![\kappa]\!]^2_\rho &= \rho(\kappa)\\
[\![\tau_1\,\tau_2]\!]^2 &= [\![\tau_1]\!]^2_\rho\,[\![\tau_2]\!]^2_\rho\\
[\![\forall\kappa.\tau]\!]^2 &= \left\{ (u, v) \,\middle|\, \forall \mathcal{F} : \kappa_1 \leftrightarrow \kappa_2.\ (u_{\kappa_1}, v_{\kappa_2}) \in [\![\tau]\!]^2_{\rho[\kappa \mapsto \mathcal{F}]} \right\}
\end{aligned}
$$

Parametricity holds also on this relational interpretation [1, 5]. Here, κ, κ_1 and κ_2 are type constructors of kind $* \to *$, and thus the quantified $\mathcal{F}$ is a relational action. The notation of relational action can be extended to type constructors of kind $* \to * \to *$.

For unary/binary relations $\mathcal{R}$ and $\mathcal{S}$, we write $L\ \mathcal{R}\ \mathcal{S}$ for $(\mathcal{R} \to \mathcal{S}, \mathcal{R} \to \mathcal{S} \to \mathcal{R})$. The following lemma holds for L.

Lemma 3. *For binary relations $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{T}$, if $(f_1, f_2) \in L\ \mathcal{R}\ \mathcal{S}$ and $(g_1, g_2) \in L\ \mathcal{S}\ \mathcal{T}$, then $(g_1 \mathbin{\hat\circ} f_1, g_2 \mathbin{\hat\circ} f_2) \in L\ \mathcal{R}\ \mathcal{T}$. For unary relations $\mathcal{R}$, $\mathcal{S}$ and $\mathcal{T}$, if $f \in L\ \mathcal{R}\ \mathcal{S}$ and $g \in L\ \mathcal{S}\ \mathcal{T}$, then $g \mathbin{\hat\circ} f \in L\ \mathcal{R}\ \mathcal{T}$.* □

A.2 Proof of Lemma 1

Let us consider a function f of type $\forall s.\ L\ s\ A \to L\ s\ B$ in which L is abstract. This means that we have a function h

$$
\begin{aligned}
\forall \ell.\,(\forall a\ b.\ L\ a\ b \to (\forall s.\ \ell\ s\ a \to \ell\ s\ b))\\
\quad \to (\forall a\ b.\,(\forall s.\ \ell\ s\ a \to \ell\ s\ b) \to L\ a\ b)\\
\quad \to \forall s.\ \ell\ s\ A \to \ell\ s\ B
\end{aligned}
$$

such that $f = h\ lift\ unlift$.

An important property of h is that s needs not be a type. Although $lift$ and $unlift$ instantiate s as a type, h itself does not bring such a constraint. By using this fact, instead of h, we will consider the instantiation of h where $\ell\ s\ a$ is substituted with $Lifted\ s \Rightarrow s\ a$, where $Lifted$ is a type class defined below.

$$
\begin{aligned}
&\textbf{class}\ Lifted\ f\ \textbf{where}\\
&\quad lift' :: L\ a\ b \to (f\ a \to f\ b)
\end{aligned}
$$

The type class has the following instance.

$$
\begin{aligned}
&\textbf{instance}\ Lifted\ (L\ s)\ \textbf{where}\\
&\quad lift' = lift
\end{aligned}
$$

We also can define the function $unlift'$ that is a counterpart of $unlift$.

$$
\begin{aligned}
&unlift' :: \forall a\ b.\,(\forall f.\ Lifted\ f \Rightarrow f\ a \to f\ b) \to L\ a\ b\\
&unlift'\ f = unlift\ f
\end{aligned}
$$

For convenience, we name the instance of h as h' which has the type below.

$$
\begin{aligned}
&(\forall a\ b.\ L\ a\ b \to (\forall f.\ Lifted\ f \Rightarrow f\ a \to f\ b))\\
&\quad \to (\forall a\ b.\,(\forall f.\ Lifted\ f \Rightarrow f\ a \to f\ b) \to L\ a\ b)\\
&\quad \to \forall f.\ Lifted\ f \Rightarrow f\ A \to f\ B
\end{aligned}
$$

One can find that we have instantiated $\ell\ s\ a$ with $Lifted\ s \Rightarrow s\ a$. Since h' is merely an instance of h, and $lift' :: L\ a\ b \to L\ s\ a \to L\ s\ b$ and $unlift'$ have the same behaviors as $lift$ and $unlift$, we have $h\ lift\ unlift = (h'\ lift'\ unlift' :: \forall s.\ L\ s\ A \to L\ s\ B)$. Then, we consider the free theorem of $h'\ lift'\ unlift'$ instead of h to prove Lemma 1.

One might think that our discussion is somewhat indirect, and should be done in a more direct way. The technical problem is that

Voigtländer's extension [4] does not consider the case where we instantiate a type variable with a type with (type-class) constraints. Thus, we have to do this instantiation explicitly before using the free theorems.

The following is a free theorem obtained from the type of h'.

Theorem 5 (A Free Theorem)**.** *Let f be a function of type $\forall s.\, L\ s\ A \to L\ s\ B$ in which L is abstract. Suppose that $\mathcal{F} :: L\ S_1 \leftrightarrow L\ S_2$ be a relational action satisfying the following condition.*

- $(\mathit{lift}, \mathit{lift}) \in \forall \mathcal{T}.\forall \mathcal{U}.L\ \mathcal{T}\ \mathcal{U} \to (\mathcal{F}\ \mathcal{T} \to \mathcal{F}\ \mathcal{U})$

Then, we have $(f, f) \in \mathcal{F}\ A \to \mathcal{F}\ B$. □

Let $\ell :: L\ S_1\ S_2$ be a lens. Then, we define $\mathcal{F}$ as follows.

$$\mathcal{F}\ \mathcal{R} = \left\{ (x_1, x_2) \;\middle|\; \begin{array}{l} \exists (z_1, z_2) \in L\ S_1\ \mathcal{R}. \\ \quad (x_1, x_2) = (z_1, z_2 \mathbin{\hat{\circ}} \ell) \end{array} \right\}$$

Assume that we have proved the required conditions to use the free theorem above. Then, we can prove Lemma 1 by the free theorem. We have $(id_{\mathrm{L}}, \ell) \in \mathcal{F}\ A$ because we have $(id_{\mathrm{L}}, id_{\mathrm{L}}) \in L\ A\ A$. From the free theorem, we have $(f, f) \in \mathcal{F}\ A \to \mathcal{F}\ B$. Take $S_1 = A$. Thus, we have $(f\ id_{\mathrm{L}}, f\ \ell) \in \mathcal{F}\ B$. This means that there is a pair $(z_1, z_2) \in L\ A\ B$ such that $f\ id_{\mathrm{L}} = z_1$ and $f\ \ell = z_2 \mathbin{\hat{\circ}} \ell$. Since $L\ A\ B$ is a diagonal relation, we have $z_1 = z_2$. Thus, we obtain $f\ id_{\mathrm{L}} \mathbin{\hat{\circ}} \ell = f\ \ell$.

Then, we prove the required condition of the free theorem. Let $\mathcal{T}$ and $\mathcal{U}$ be relations. Let x_1 and x_2 be lenses such that $(x_1, x_2) \in L\ \mathcal{T}\ \mathcal{U}$, and let z_1 and z_2 be lenses such that $(z_1, z_2) \in \mathcal{F}\ \mathcal{T}$. Then, from Lemma 3, we have $(\mathit{lift}\ x_1\ z_1, \mathit{lift}\ x_2\ z_2) = (x_1 \mathbin{\hat{\circ}} z_1, x_2 \mathbin{\hat{\circ}} z_2) \in \mathcal{F}\ \mathcal{U}$.

A.3 Proof of Correctness

The correctness of our framework is stated by Theorem 4. However, here, we prove a more general version that covers the extension to observation discussed in Section 6. We do not show the proof of Theorem 1 here because it can be proved similarly to the statement in this section.

First, we extend the notion of the abstract nature.

Definition 6 (Abstract Nature of L^{T} and R)**.** We say L^{T} *and* R *are abstract* in $f :: \tau$ if there is a polymorphic function h of type

$$\begin{array}{l} \forall \ell\ r. \\ \quad (\forall a\ b.\, L\ a\ b \to (\forall s.\, \ell\ s\ a \to \ell\ s\ b)) \\ \quad \to (\forall a\ b.\, (\forall s.\, \ell\ s\ a \to \ell\ s\ b) \to L\ a\ b) \\ \quad \to (\forall s.\, \ell\ s\ ()) \\ \quad \to (\forall s\ a\ b.\, \ell\ s\ a \to \ell\ s\ b \to \ell\ s\ (a, b)) \\ \quad \to (\forall a\ b\ c.\, (\forall s.\, (\ell\ s\ a, \ell\ s\ b) \to \ell\ x\ c) \to L\ (a, b)\ c) \\ \quad \to (\forall s\ w.\, \mathit{Eq}\ w \Rightarrow \ell\ s\ w \to r\ s\ w) \\ \quad \to (\forall a\ b.\, (\forall s.\, \ell\ s\ a \to r\ s\ (\ell\ s\ b)) \to L\ a\ b) \\ \quad \to (\forall a\ b\ c.\, (\forall s.\, (\ell\ s\ a, \ell\ s\ b) \to r\ s\ (\ell\ x\ c)) \to L\ (a, b)\ c) \\ \quad \to \tau' \end{array}$$

satisfying $f = h\ \mathit{lift}\ \mathit{unlift}\ \mathit{unit}\ (\circledast)\ \mathit{unlift2}\ \mathit{observe}\ \mathit{unliftM}\ \mathit{unliftM2}$ and $\tau' = \tau[\ell/L^{\mathrm{T}}, r/R]$. □

In addition, we implicitly assume that $(\mathtt{==})$ of $\mathit{Eq}\ w$ above defines the semantic (i.e., observational) equality. For simplicity, we only consider unary and binary unlifting functions.

Then, we state a free theorem for functions of the type above.

Theorem 6 (A Free Theorem)**.** *Let $f :: \tau$ be a function in which L^{T} and R are abstract, and τ' be a type $\tau[\ell/L^{\mathrm{T}}, r/R]$. For any $\mathcal{F} :: \mathsf{Pred}(L^{\mathrm{T}})$ and $\mathcal{M} :: \mathsf{Pred}(R)$ satisfying the following conditions.*

- $\mathit{lift} \in \forall \mathcal{T}.\forall \mathcal{U}.L\ \mathcal{T}\ \mathcal{U} \to (\forall \mathcal{S}.\mathcal{F}\ \mathcal{S}\ \mathcal{T} \to \mathcal{F}\ \mathcal{S}\ \mathcal{U})$.
- $\mathit{unlift} \in \forall \mathcal{T}.\forall \mathcal{U}.(\forall \mathcal{S}.\mathcal{F}\ \mathcal{S}\ \mathcal{T} \to \mathcal{F}\ \mathcal{S}\ \mathcal{U}) \to L\ \mathcal{T}\ \mathcal{U}$.
- $\mathit{unit} \in \forall \mathcal{T}.L\ \mathcal{T}\ ()$.
- $(\circledast) \in \forall \mathcal{T}.\forall \mathcal{U}.\forall \mathcal{S}.\mathcal{F}\ \mathcal{S}\ \mathcal{T} \to \mathcal{F}\ \mathcal{S}\ \mathcal{U} \to \mathcal{F}\ \mathcal{S}\ (\mathcal{T}, \mathcal{U})$.
- $\mathit{unlift2} \in \forall \mathcal{T}.\forall \mathcal{U}.\forall \mathcal{R}.(\forall \mathcal{S}.(\mathcal{F}\ \mathcal{S}\ \mathcal{T}, \mathcal{F}\ \mathcal{S}\ \mathcal{U}) \to \mathcal{F}\ \mathcal{S}\ \mathcal{R}) \to L\ (\mathcal{T}, \mathcal{U})\ \mathcal{R}$.
- $\mathit{observe} \in \forall \mathcal{S}.\mathcal{F}\ \mathcal{S}\ \mathcal{T} \to \mathcal{M}\ \mathcal{S}\ \mathcal{T}$ *for any unary relation $\mathcal{T}$ on a type with decidable semantic equality* $(\mathtt{==})$.
- $\mathit{unliftM} \in \forall \mathcal{T}.\forall \mathcal{U}.(\forall \mathcal{S}.\mathcal{F}\ \mathcal{S}\ \mathcal{T} \to \mathcal{M}\ \mathcal{S}\ (\mathcal{F}\ \mathcal{S}\ \mathcal{U})) \to L\ \mathcal{T}\ \mathcal{U}$.
- $\mathit{unliftM2} \in \forall \mathcal{T}.\forall \mathcal{U}.\forall \mathcal{R}.(\forall \mathcal{S}.(\mathcal{F}\ \mathcal{S}\ \mathcal{T}, \mathcal{F}\ \mathcal{S}\ \mathcal{U}) \to \mathcal{M}\ \mathcal{S}\ (\mathcal{F}\ \mathcal{S}\ \mathcal{R})) \to L\ (\mathcal{T}, \mathcal{U})\ \mathcal{R}$.

Then, we have $f \in [\![\tau']\!]_{\{\ell \mapsto \mathcal{F}, r \mapsto \mathcal{M}\}}$. □

Then, we will prove the following theorem. We only discuss $\mathit{unliftM2}$ here because extension to other unlifting functions are straightforward.

Theorem 7. *Let f be a function of type $\forall s.\, (L^{\mathrm{T}}\ s\ A, L^{\mathrm{T}}\ s\ B) \to R\ s\ (L^{\mathrm{T}}\ s\ C)$ in which L^{T} and R are abstract. Then $\mathit{unliftM2}\ f$ is well-behaved, assuming that lift is applied only to well-behaved lenses in f.*

Proof. Thanks to the abstract nature of L^{T} and R in f, we can use Theorem 6. Concretely, we use the following $\mathcal{F}$ and $\mathcal{M}$.

$$\begin{array}{l} \mathcal{F}\ \mathcal{S}\ \mathcal{R} = \{\ell \mid \ell \in L^{\mathrm{T}}\ \mathcal{S}\ \mathcal{R}, \ell \text{ is locally well-behaved}\} \\ \mathcal{M}\ \mathcal{S}\ \mathcal{R} = \\ \quad \left\{ R\ m \;\middle|\; \begin{array}{l} \forall s \in \mathcal{S}.\ \text{let } (x, p) \text{ be } m\ s, \\ \qquad x \in \mathcal{R} \wedge \\ \qquad p\ s = \mathit{True} \wedge \\ \qquad \forall s' \in \mathcal{S}.\ p\ s' = \mathit{True} \Rightarrow m\ s = m\ s' \end{array} \right\} \end{array}$$

Then, the statement is obtained by simple instantiation. It is worth noting that $\mathcal{F}\ \mathcal{S}\ \mathcal{R} \subseteq L^{\mathrm{T}}\ \mathcal{S}\ \mathcal{R}$ and $\mathcal{M}\ \mathcal{S}\ \mathcal{R} \subseteq R\ \mathcal{S}\ \mathcal{R}$.

For the cases of lift and $(\circledast)$, we just use Lemma 2. For the case of unit, the proof is obvious. For the cases of unlift, $\mathit{unlift2}$, $\mathit{unliftM}$ and $\mathit{unliftM2}$, the proofs are straightforward because $\mathcal{F}\ \mathcal{S}\ \mathcal{R}$ is a subset of $L\ \mathcal{S}\ \mathcal{R}$ for any $\mathcal{S}$ on a type that is an instance of Poset.

For the case of $\mathit{observe}$, the proof is still straightforward. The last two lines of $\mathcal{M}$ are obtained by the fact that $(\mathtt{==})$ is the semantic equality.

Then, by the free theorem, we have $f\ (x, y) \in \mathcal{M}\ \mathcal{S}\ (L^{\mathrm{T}}\ \mathcal{S}\ C)$ for any x and y such that $x \in L^{\mathrm{T}}\ \mathcal{S}\ A$ and $y \in L^{\mathrm{T}}\ \mathcal{S}\ B$ for any $\mathcal{S}$. Thus, for each s, we have ℓ is the definition of $\mathit{mkLens}\ f\ s$ is locally well-behaved by the definition of $\mathcal{M}$. Then, we have that ℓ' is well-behaved because it is obtained by composing ℓ and $\mathit{tag2}_{\mathrm{L}}$ (Lemma 2). We have $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ (\mathit{get}\ (\mathit{mkLens}\ f\ s)\ s) = s$ from the consistency of ℓ' and $p\ (\mathit{get}\ \mathit{tag2}_{\mathrm{L}}\ s)\ \mathtt{==}\ \mathit{True}$ from the definition of $\mathcal{M}$. That is, $\mathit{unliftM2}\ f$ satisfies acceptability. Note that $\mathit{mkLens}\ f\ s$ itself can violate the acceptability when we pass $s' \not\equiv s$ to $\mathit{get}\ (\mathit{mkLens}\ f\ s)$. Also, we have $\mathit{get}\ (\mathit{mkLens}\ f\ s)\ s' = v$ if $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ v = s'$. However, this does not immediately imply the consistency of $\mathit{unliftM2}\ f$, which says that $\mathit{get}\ (\mathit{mkLens}\ f\ s')\ s' = v$ if $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ v = s'$. That is, we have to ensure that $\mathit{mkLens}\ f\ s = \mathit{mkLens}\ f\ s'$ where $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ v = s'$, because $\mathit{mkLens}\ f\ s$ depends on a source s.

Then, we will prove $\mathit{mkLens}\ f\ s = \mathit{mkLens}\ f\ s'$ if $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ v$ succeeds in s'. The last line of the definition of $\mathcal{M}$ plays the important role in this proof. From the fact that $\mathit{put}\ (\mathit{mkLens}\ f\ s)\ s\ v$ succeeds, we obtain that $p\ (\mathit{get}\ \mathit{tag2}_{\mathrm{L}}\ s') = p\ (\mathit{get}\ \mathit{tag2}_{\mathrm{L}}\ s) = \mathit{True}$. Thus, we have $\mathbf{let}\ R\ m = f\ (\mathit{fst}', \mathit{snd}')\ \mathbf{in}\ m\ (\mathit{get}\ \mathit{tag2}_{\mathrm{L}}\ s) = \mathbf{let}\ R\ m = f\ (\mathit{fst}', \mathit{snd}')\ \mathbf{in}\ m\ (\mathit{get}\ \mathit{tag2}_{\mathrm{L}}\ s)$ by $f\ (\mathit{fst}', \mathit{snd}') \in \mathcal{M}\ \mathcal{S}\ (L^{\mathrm{T}}\ \mathcal{S}\ C)$. The rest of construction in $\mathit{mkLens}\ f\ s$ does not depend on s, and thus $\mathit{mkLens}\ f\ s = \mathit{mkLens}\ f\ s'$ holds. □

It is worth noting that, for unlifting functions for data structures, we need to keep an additional invariant on $\mathcal{F}$: a lens ℓ in $\mathcal{F}\ \mathcal{S}\ \mathcal{R}$ must be shape-preserving if S of $\mathcal{S} :: \mathsf{Pred}(S)$ has a shape.

References

[1] J.-P. Bernardy, P. Jansson, and R. Paterson. Proofs for free - parametricity for dependent types. *J. Funct. Program.*, 22(2):107–152, 2012.

[2] J. C. Reynolds. Types, abstraction and parametric polymorphism. In R. Mason, editor, *Information Processing*, pages 513–523. Elsevier Science Publishers B.V. (North-Holland), 1983.

[3] J. Voigtländer. Bidirectionalization for free! (pearl). In Z. Shao and B. C. Pierce, editors, *POPL*, pages 165–176. ACM, 2009.

[4] J. Voigtländer. Free theorems involving type constructor classes: functional pearl. In G. Hutton and A. P. Tolmach, editors, *ICFP*, pages 173–184. ACM, 2009.

[5] D. Vytiniotis and S. Weirich. Parametricity, type equality, and higher-order polymorphism. *J. Funct. Program.*, 20(2):175–210, 2010.

[6] P. Wadler. Theorems for free! In *FPCA*, pages 347–359, 1989.