Lenses and View Update Translation

Benjamin C. Pierce and Alan Schmitt

University of Pennsylvania

Working draft of April 24, 2003

This manuscript is a technical sketch of some results that became too lengthy (and interesting) to fit in an earlier paper, "A Language For Bi-Directional Tree Transformations," by Greenwald, Moore, Pierce, and Schmitt [3]. We repeat the necessary basic technical definitions here for convenient reference, but please refer to that paper for background, motivations, intuitions, related work, etc.

Abstract

We draw precise connections between lenses and some "classical" structures studied in the context of the view update translation problem: the notion of *view update under a constant complement* of Bancilhon and Spyratos and the *dynamic views* of Gottlob, Paolini, and Zicari.

1 Introduction

We establish three main results:

1. The set of *very-well-behaved lenses* in the sense of [3] is isomorphic to the set of *translators under constant complement* in the sense of Bacilhon and Spyratos [1].

2. The set of *well-behaved lenses* is isomorphic to the set of *dynamic views* in the sense of Gottlob, Paolini, and Zicari [2].

   To be precise, both of these results must be qualified by an additional condition regarding partiality. The frameworks of Bacilhon and Spyratos and of Gottlob, Paolini, and Zicari are both formulated in terms of translating *update functions* on A into update functions on C—i.e., their *put* functions have type $(A \longrightarrow A) \longrightarrow (C \longrightarrow C)$—while our lenses translate abstract *states* into update functions on C—i.e., our *put* functions have type (isomorphic to) $A \longrightarrow (C \longrightarrow C)$. Moreover, in both of these frameworks, "update translators" (the analog of our *put* functions) are defined only over some particular chosen set U of abstract update functions, not over all functions from A to A. These update translators return *total* functions from C to C. Our *put* functions, on the other hand, are more general as they are defined over all abstract states and return *partial* functions from C to C. Finally, the *get* functions of lenses are allowed to be partial, whereas the corresponding functions (called *views*) in the other two frameworks are assumed to be total. In order to make the correspondences tight, the sets of well-behaved and very-well-behaved lenses need to be restricted to subsets that are "total" in a suitable sense.

3. If we restrict both *get* and *put* to be total functions (i.e., *put* must be total with respect to *all* abstract update functions), then our lens laws (including PutPut) characterize the set C as isomorphic to $A \times B$ for some B.

2 The Results

2.1 Basic Structure

Let C and A be two sets. Let Ω an element that does not occur in C. We write C_Ω for $C \cup \{\Omega\}$.

2.1.1 Definition [Lenses]: A *lens* l comprises two partial functions:

- a *get* function from C to A, written $l\nearrow$ and
- a *put* function from $A \times C_\Omega$ to C, written $l\searrow$.

We write $\mathsf{dom}(l\nearrow)$ for the subset of C on which $l\nearrow$ is defined and $\mathsf{dom}(l\searrow)$ for the subset of $A \times C_\Omega$ on which $l\searrow$ is defined, and similarly $\mathsf{ran}(l\nearrow)$ and $\mathsf{ran}(l\searrow)$ for the ranges of the *get* and *put* functions. Note that neither $l\nearrow$ nor $l\searrow$ may return Ω.

We now define some laws that well-behaved lenses should obey.

2.1.2 Definition [Well-behaved lenses]: A lens is *well behaved* iff its *get* and *put* functions obey the following laws:

(GetPut) $c \in \mathsf{dom}(l\nearrow) \implies l\searrow(l\nearrow c,\, c) = c$

(PutGet) $(a, c) \in \mathsf{dom}(l\searrow) \implies l\nearrow(l\searrow(a,\, c)) = a$

We write $L(C, A)$ for the set of well-behaved lenses between C and A.
A well-behaved lens is *very-well-behaved* if its *get* and *put* functions also obey the following law:

(PutPut) $a' \in A$ and $(a, c) \in \mathsf{dom}(l\searrow) \implies l\searrow(a',\, l\searrow(a,\, c)) = l\searrow(a',\, c)$

We write $L^+(C, A)$ for the set of very-well-behaved lenses between C and A.

2.2 Lenses and View Update Translation

In this section we establish a precise correspondance between the view update setting of [1, 2] and our lenses. Up to a small restriction concerning partiality, the set of very-well-behaved lenses is isomorphic to the set of *translators under constant complement* of Bacilhon and Spyratos [1], whereas the set of well-behaved lenses is isomorphic to the set *dynamic views* of Gottlob, Paolini, and Zicari[2].

We restrict our attention to lenses that are total with respect to a given set of update functions—that is lenses whose *put* function is defined for every possible update function from some set $U \subseteq A \longrightarrow A$.

2.2.1 Definition: Let P be a set of functions from C to C and U a set of functions from A to A. A lens l is said to be *total* with respect to U and P iff

- $\mathsf{dom}(l\nearrow) = C$;
- $l\nearrow(C) = A$;
- $\mathsf{dom}(l\searrow) = \{(u(l\nearrow c), c) \mid u \in U, c \in C\}$;
- $l\searrow(a,\, c) = c' \implies \exists p \in P.c' = p(c)$.

We write $L_t^+(C, A, U, P)$ for the set of very-well-behaved total lenses with respect to U and P and $L_t(C, A, U, P)$ for the set of well-behaved total lenses with respect to U and P.

The following definition is used to characterize databases C and views f that have a translator T for a set of complete updates U, as defined by Bancilhon and Spyratos [1].

2.2.2 Definition: A set U of functions from some set A to A is said to be complete iff it contains the identity and satisfies the following conditions:

1. $\forall u \in U, \forall v \in U, uv \in U$;
2. $\forall s \in A, \forall u \in U, \exists v \in U$ such that $vu(s) = s$;

[Comment that their inverse solution is very weak, in the sense that it does not require an inverse function, but an inverse at one point, which is closer to a state based approach –as]

2.2.3 Definition: Let C and A be sets, P be a set of functions from C to C, and U be a set of functions from A to A that is complete. We define the set $BS(C, A, U, P)$ as the set of all tuples (f, T) such that:

- f is a surjective function from C to A;
- T is a function from U to P;
- $\forall u \in U, fT(u) = uf$;
- $\forall u \in U, \forall s \in C, uf(s) = f(s) \implies T(u)(s) = s$;
- $\forall u \in U, \forall v \in V, T(uv) = T(u)T(v)$.

[think about the following comment by Zhe: Also, I find the statement about set isomorphism here and in Theorem 3.3.6 weak: conceivably the cardinalities will the be same anyway, both being N or R. I believe that your proof is constructive: for any function f in the first set you can write down its counterpart in the second set as a simple expression. This would make the correpondence natural *in all the arguments in the sense of category theory. –as]*

2.2.4 Theorem: Let C and A be some sets, and U be a complete set of functions from A to A. The sets $L_t^+(C, A, U, P)$ and $BS(C, A, U, P)$ are isomorphic.

Proof: We first introduce two functions $\mathcal{LB}$ from $L_t^+(C, A, U, P)$ to $BS(C, A, U, P)$ and $\mathcal{BL}$ from $BS(C, A, U, P)$ to $L_t^+(C, A, U, P)$ and show that both $\mathcal{LB} \circ \mathcal{BL}$ and $\mathcal{BL} \circ \mathcal{LB}$ are the identity.

Let l be a total very-well-behaved lens of $L_t^+(C, A, U, P)$. We define $\mathcal{LB}(l)$ as the tuple (f, T), where:

- $f = l\nearrow$;
- $T = u \mapsto s \mapsto l\searrow(u(l\nearrow s),\, s)$.

We show that we have $(f, T) \in BS(C, A, U, P)$.

Since l is total, f is a surjective function from C to A and T is defined for every $u \in U$ and $s \in C$.

Let $u \in U$ and $s \in C$. We have $f(T(u)(s)) = l\nearrow l\searrow(u(l\nearrow s),\, s) = u(l\nearrow s) = u(f(s))$ by law PutGet.

Let $u \in U$ and $s \in C$ such that $u(f(s)) = f(s)$. We have $T(u)(s) = l\searrow(u(l\nearrow s),\, s) = l\searrow(u(f(s)),\, s) = l\searrow(f(s),\, s) = l\searrow(l\nearrow s,\, s) = s$ by law GetPut.

Let u and v in U. We have

$$\begin{array}{rl} & T(u)(T(v))(s) \\ = & l\searrow(u(l\nearrow l\searrow(v(l\nearrow s),\, s)),\, l\searrow(v(l\nearrow s),\, s)) \\ = & l\searrow(u(v(l\nearrow s),\, l\searrow(v(l\nearrow s),\, s)) \\ = & l\searrow(u(v(l\nearrow s)),\, s) \\ = & T(uv)(s) \end{array}$$

by laws PutGet and PutPut.

We now define $\mathcal{BL}$. Let (f, T) in $BS(C, A, U, P)$, we define l as:

- $l\nearrow = f$;
- $\forall u \in U, \forall s \in C, l\searrow(u(f(s)),\, s) = T(u)(s)$.

The function $l\searrow$ is undefined otherwise.

We now check that l is a very-well-behaved total lens of $L_t^+(C, A, U, P)$.

We first check GetPut. We have:

$$\begin{array}{rcl} l\searrow(l\nearrow c,\, c) & = & l\searrow(f(c),\, c) \\ & = & l\searrow(id(f(c)),\, c) \\ & = & T(id)(c) \\ & = & c \end{array}$$

since $id(f(c)) = f(c)$ implies $T(id)(c) = c$.

We now check PutGet. We recall that $l\searrow(a,\, c)$ is defined iff $a = u(f(c))$.

$$l\nearrow l\searrow(u(f(c)),\, c) = f(T(u)(c)) = u(f(c))$$

We now check PutPut. As before, $l\searrow(a,\, c)$ is defined iff $a = u(f(c))$. We consider $l\searrow(a',\, l\searrow(u(f(c)),\, c))$.

We show that if $l\searrow(a',\, c)$ is defined then it is equal to $l\searrow(a',\, l\searrow(u(f(c)),\, c))$. Conversely, we show that if $l\searrow(a',\, l\searrow(u(f(c)),\, c))$ is defined then it is equal to $l\searrow(a',\, c)$. We conclude by noting that if any is undefined, then the other is undefined.

Let us assume that $l\searrow(a',\, c)$ is defined. Then we have $a' = u'(f(c))$ for some u'. Let v be such that $vu(f(c)) = f(c)$ (we know that v exists by definition of U). We have $a' = u'vu(f(c))$. We have:

$$\begin{array}{rl} & l\searrow(u'vu(f(c)),\, l\searrow(u(f(c)),\, c)) \\ = & l\searrow(u'v(u(f(c))),\, T(u)(c)) \\ = & l\searrow(u'v(f(T(u)(c))),\, T(u)(c)) \\ = & T(u'v)(T(u)(c)) \\ = & T(u')(T(vu)(c)) \\ = & T(u')(c) = l\searrow(u'(f(c)),\, c) = l\searrow(a',\, c) \end{array}$$

since $vu(f(c)) = f(c)$ thus $T(vu)(c) = c$.

Let us assume that $l\searrow(a',\, l\searrow(u(f(c)),\, c)) = l\searrow(a',\, T(u)(c))$ is defined. Thus for some u' we have $a' = u'(f(T(u)(c))$. We have:

$$\begin{array}{rl} & l\searrow(u'(f(T(u)(c)),\, T(u)(c)) \\ = & T(u')(T(u)(c)) \\ = & T(u'u)(c) \\ = & l\searrow(u'u(f(c)),\, c) \\ = & l\searrow(u'(f(T(u)(c))),\, c) \\ = & l\searrow(a',\, c) \end{array}$$

As f is surjective and is defined on all of C, we only need to check that $\mathsf{dom}(l\searrow) = \{(u(l\nearrow c), c) \mid u \in U, c \in C\}$. It is the case by definition for the *put* function since $f = l\nearrow$. Moreover, since T is a function from U to P, if $l\searrow(u(f(c)),\, c) = c'$ then we have $c' = T(u)(c)$ for some $u \in U$, and we conclude by $T(u) \in P$.

We now show that $\mathcal{LB} \circ \mathcal{BL}$ is the identity. Let (f, T) be in $BS(C, A, U, P)$. We have:

$$\begin{array}{rl} & \mathcal{LB}(\mathcal{BL}(f, T)) \\ = & \mathcal{LB}(f, \{(u(f(s)), s) \mapsto T(u)(s) \mid u \in U, s \in C\}) \\ = & \big(f, \big\{u' \mapsto s' \mapsto (\{(u(f(s)), s) \mapsto T(u)(s) \mid u \in U, s \in C\})(u'(f(s')), s') \mid u' \in U, s' \in C\big\}\big) \\ = & (f, \{u' \mapsto s' \mapsto T(u')(s') \mid u' \in U, s' \in C\} \\ = & (f, T) \end{array}$$

We now show that $\mathcal{BL} \circ \mathcal{LB}$ is the identity. Let l be a very-well-behaved total lens $(l\nearrow, l\searrow)$. We have:

$$\begin{array}{ll} & \mathcal{BL}(\mathcal{LB}(l\nearrow, l\searrow)) \\ = & \mathcal{BL}(l\nearrow, \{u \mapsto s \mapsto l\searrow(u(l\nearrow s),\ s) \mid u \in U, s \in C\} \\ = & (l\nearrow, \big\{(u'(l\nearrow s'), s') \mapsto (\{u \mapsto s \mapsto l\searrow(u(l\nearrow s),\ s) \mid u \in U, s \in C\})u'\ s' \mid u' \in U, s' \in C\big\}) \\ = & (l\nearrow, \{(u'(l\nearrow s'), s') \mapsto l\searrow(u'(l\nearrow s'),\ s') \mid u' \in U, s' \in C\}) \\ = & (l\nearrow, l\searrow) \end{array}$$

since $\mathsf{dom}(l\searrow) = \{(u(l\nearrow c), c) \mid u \in U, c \in C\}$. $\square$

We now state the relation between our setting and [2]: the set of dynamic views is isomorphic to the set of total well-behaved lenses.

2.2.5 Definition: The tuple $((C, P), (A, U), f, \tau)$ is a dynamic view iff:

- C and A are sets of states;
- P (respectivly U) is a set of update operators, that is functions from C to C (respectively from A to A) containing the identity;
- f is a surjective function from C to A;
- τ is a *translator*, a function from U to P such that for all $u \in U$ and $s \in C$, $f((\tau u)(s)) = u(f(s))$ and such that $\tau id_A = id_C$.

We write $DV(C, A, U, P)$ for the set of dynamic views on C, P, A and U.

2.2.6 Theorem: For any sets C, A, P and U, the set $DV(C, A, U, P)$ is isomorphic to $L_t(C, A, U, P)$.

Proof: Let C and A be two sets and P and U be two operator sets containing the identity.

We define a function $\mathcal{DL}$ which transforms a dynamic view of $DV(C, A, U, P)$ into a lens of $L_t(C, A, U, P)$, and a function $\mathcal{LD}$ which transforms a lens from $L_t(C, A, U, P)$ to a dynamic view $DV(C, A, U, P)$. We then prove that $\mathcal{DL} \circ \mathcal{LD} = id_{L_t(C,A,U,P)}$ and that $\mathcal{LD} \circ \mathcal{DL} = id_{DV(C,A,U,P)}$.

Let $((C, P), (A, U), f, \tau)$ be a dynamic view. We define $\mathcal{DL}$ on this dynamic view as l. Let l be the lens defined as:

$$\begin{array}{c} \forall s \in C\ .\ \ l\nearrow s = f(s) \\ \forall s, u \in C \times U\ .\ \ l\searrow(u(f(s)),\ s) = (\tau u)(s) \end{array}$$

For all other arguments, $l\searrow$ is undefined.

We show that we have $l \in L_t(C, A, U, P)$. First, we show that l is well-behaved.

GetPut Let $s \in C$.

$$\begin{array}{rcl} l\searrow(l\nearrow s,\ s) & = & l\searrow(f(s),\ s) \\ & = & l\searrow(id_A(f(s)),\ s) \\ & = & (\tau id_A)(s) \\ & = & id_C(s) \\ & = & s \end{array}$$

PutGet Let $s \in C$ and $u \in U$.

$$\begin{array}{rcl} l\nearrow l\searrow(u(f(s)),\ s) & = & l\nearrow(\tau u)(s) \\ & = & f((\tau u)(s)) \\ & = & u(f(s)) \end{array}$$

We now show that l is total. We immediately have $\mathsf{dom}(l\nearrow) = \mathsf{dom}(f) = C$. Moreover, we have $l\nearrow C = f(C) = A$ as f is surjective. By definition, we have $\mathsf{dom}(l\searrow) = \{(u(f(s)), s) \mid u \in U, s \in C\}$ with $f = l\nearrow$. Moreover, if $l\searrow(a, c) = c'$, then $a = u(f(c))$ and $c' = (\tau u)(c) = p(c)$ for some $p \in P$ by definition of τ. Thus $l \in L_t(C, A, U, P)$.

Let l be a lens of $L_t(C, A, U, P)$. We define $\mathcal{LD}$ on this lens as the tuple $((C, P), (A, U), f, \tau)$ where:

- $f = l\nearrow$;
- $(\tau u)s = l\searrow(u(l\nearrow s), s)$.

Note that τ is defined for any u and s by the requirement on the domain of $l\searrow$.

We show that $((C, P), (A, U), f, \tau)$ is a dynamic view. The function f is surjective because l is total. The function τ is from U to P because l is total.

Let $u \in U$ and $s \in C$, we have:

$$\begin{aligned} f((\tau u)(s)) &= l\nearrow l\searrow(u(l\nearrow s), s) \\ &= u(l\nearrow s) \\ &= u(f(s)) \end{aligned}$$

We also have

$$\begin{aligned} (\tau id_A)(s) &= l\searrow(id_A(l\nearrow s), s) \\ &= l\searrow(l\nearrow s, s) \\ &= s \\ &= id_C(s) \end{aligned}$$

We now show that $\mathcal{DL}$ and $\mathcal{LD}$ form an isomorphism.

Let $((C, P), (A, U), f, \tau)$ be a dynamic view. Through $\mathcal{DL}$ we obtain the lens $l\nearrow = f$ and $l\searrow(u(f(s)), s) = (\tau u)(s)$ for any $u \in U$ and $s \in C$. Through $\mathcal{LD}$ we obtain the dynamic view $f' = l\nearrow = f$ and for any $u \in U$ and $s \in C$:

$$\begin{aligned} (\tau' u)(s) &= l\searrow(u(l\nearrow s), s) \\ &= l\searrow(u(f(s)), s) \\ &= (\tau u)(s) \end{aligned}$$

Thus $\mathcal{LD} \circ \mathcal{DL} = id_{DV(C,A,U,P)}$.

Let l be a lens of $L_t(C, A, U, P)$. Through $\mathcal{DL}$ we obtain the dynamic view $f = l\nearrow$ and for any $u \in U$ and $s \in C$, $(\tau u)(s) = l\searrow(u(l\nearrow s), s)$. Through $\mathcal{LD}$ we obtain the lens $l'\nearrow = f = l\nearrow$ and for any $u \in U$ and $s \in C$:

$$\begin{aligned} l'\searrow(u(f(s)), s) &= (\tau u)(s) \\ &= l\searrow(u(l\nearrow s), s) \\ &= l\searrow(u(f(s)), s) \end{aligned}$$

Thus $\mathcal{DL} \circ \mathcal{LD} = id_{L_t(C,A,U,P)}$. □

2.3 Partitioning the Concrete Set

In this section, we show that given any fully total very-well-behaved lens (whose *put* function is total), the concrete set C is isomorphic to the cross product of the abstract set A and some other set.

2.3.7 Definition: A very-well-behaved lens is said to be *fully total* iff $\mathsf{dom}(l\searrow) = A \times C$ and if $\mathsf{ran}(l\searrow) = C$.

2.3.8 Lemma: Let l be a fully total very-well-behaved lens. Then we have $\mathsf{dom}(l\nearrow) = C$ and, if $C \neq \emptyset$, $\mathsf{ran}(l\nearrow) = A$.

Proof: Let $c \in C$. As l is fully total, there are some a and c' such that $l\searrow(a, c') = c$. Thus by rule PutGet, we have $l\nearrow c = a$ thus $c \in \mathsf{dom}(l\nearrow)$.

Let $a \in A$. Let $c \in C$ (which is not empty, thus there is such a c). As l is fully total, $l\searrow(a, c)$ is defined, thus by PutGet $l\nearrow l\searrow(a, c) = a$, thus $a \in \mathsf{ran}(l\nearrow)$. □

2.3.9 Theorem: Let l be a fully total very-well-behaved lens on C and A. The set C is isomorphic to $A \times \{\lambda x.l\searrow(x, c) \mid c \in C\}$.

Proof: Let f be the function from C to $A \times \{\lambda x.l\searrow(x, c) \mid c \in C\}$ defined as:

$$f(c) = (l\nearrow c, \lambda x.l\searrow(x, c))$$

Let g be the function from $A \times \{\lambda x.l\searrow(x, c) \mid c \in C\}$ to C defined as:

$$g(a, h) = h(a)$$

As l is fully total and by lemma 2.3.8, both f and g are well defined.

We now show that both $f \circ g$ and $g \circ f$ are the identity.

Let $c \in C$. We have

$$g(f(c)) = g(l\nearrow c, \lambda x.l\searrow(x, c)) = l\searrow(l\nearrow c, c) = c$$

by GetPut.

Let $(a, h) \in A \times \{\lambda x.l\searrow(x, c) \mid c \in C\}$. For any c such that $h = \lambda x.l\searrow(x, c)$, we have:

$$f(g(a, h)) = f(l\searrow(a, c)) = (l\nearrow l\searrow(a, c), \lambda x.l\searrow(x, l\searrow(a, c))$$

By rule PutGet we have $l\nearrow l\searrow(a, c) = a$. By rule PutPut we have $l\searrow(x, l\searrow(a, c)) = l\searrow(x, c)$. Thus $f(g(a, h)) = (a, h)$. □

We should write down the proof of the "strongest isomorphism" when both get and put are total. This proof should cite [4], where a similar thing is proved (and maybe also [5]?)

We also remark that, if we restrict both get and put to be total functions, our lens laws characterize the set C as exactly (up to isomorphism) $A \times B$ for some B. [This is just a "remark" because it is well known to mathematicians.]

3 Related work

To be written.

4 Future Work

To be done

5 Acknowledgements

Michael Greenwald Jonathan Moore

References

[1] F. Bancilhon and N. Spyratos. Update semantics of relational views. *TODS*, 6(4):557–575, 1981.

[2] G. Gottlob, P. Paolini, and R. Zicari. Properties and update semantics of consistent views. *TODS*, 13(4):486–524, 1988.

[3] M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt. A language for bi-directional tree transformations. Manuscript; available at `http://www.cis.upenn.edu/~bcpierce/papers/lenses.pdf`, 2003.

[4] M. Hofmann and B. Pierce. Positive subtyping. In *ACM Symposium on Principles of Programming Languages (POPL), San Francisco, California*, pages 186–197, Jan. 1995. Full version in *Information and Computation*, volume 126, number 1, April 1996. Also available as University of Edinburgh technical report ECS-LFCS-94-303, September 1994.

[5] F. J. Oles. Type algebras, functor categories, and block structure. In M. Nivat and J. C. Reynolds, editors, *Algebraic Methods in Semantics*. Cambrige University Press, 1985.