Positively capjunctive cappings, and Galois

K. Rustan M. Leino
5 January 1998

Digital Equipment Corporation Systems Research Center
130 Lytton Ave., Palo Alto, CA 94301, U.S.A.
rustan@pa.dec.com

Abstract. This note defines an operator $\Box'$ which maps any predicate transformer to its least-refined positively conjunctive predicate transformer. Operator $\Box'$ is similar to Carroll Morgan's $\Box$ operator, which maps any predicate transformer to its least-refined universally conjunctive predicate transformer. The proof that $\Box'$ has the desired properties makes use of Morgan's results and Galois connections. The definition of $\Box'$ turns out to have an interesting, and, at least to me, novel ingredient.

0 Introduction

Carroll Morgan defines a function $\Box$, which has the following properties [2]: For any predicate transformer S,

$$\Box.S \in \mathcal{C}^{\top}_{\wedge} \tag{0}$$
$$\langle\, \forall c \mid c \in \mathcal{C}^{\top}_{\wedge} \triangleright S \sqsubseteq c \;\equiv\; \Box.S \sqsubseteq c \,\rangle \tag{1}$$

where $\mathcal{C}^{\top}_{\wedge}$ denotes the set of all universally conjunctive predicate transformers and $\sqsubseteq$ is the refinement ordering on predicate transformers.

Rajit Manohar and I have the need for a function $\Box'$ with the following properties [0]: For any predicate transformer S,

$$\Box'.S \in \mathcal{S}_0 \tag{2}$$
$$\langle\, \forall c \mid c \in \mathcal{S}_0 \triangleright S \sqsubseteq c \;\equiv\; \Box'.S \sqsubseteq c \,\rangle \tag{3}$$

where $\mathcal{S}_0$ denotes the set of all positively conjunctive predicate transformers. Paul Gardiner suggested to us that we define function $\Box'$ in terms of $\Box$ as follows: For any predicate transformer S and predicate F,

$$\Box'.S.F \;\equiv\; S.true \wedge \Box.S.F \tag{4}$$

Enticed by Gardiner's definition, we became interested in proving properties (2) and (3) from the definition of $\Box'$. That is the subject of this note.

As it turns out, if $\Box'$ is defined by (4), it will satisfy property (3) only for predicate transformers S that are *monotonic* (with respect to the refinement ordering). The desired

definition of $\Box'$ in terms of $\Box$ is instead: For any predicate transformer S and predicate F,

$$\Box'.S.F \;\equiv\; \langle\, \exists F \;\triangleright\; S.F \,\rangle \;\wedge\; \Box.S.F \tag{5}$$

Note that the right-hand sides of definitions (4) and (5) are the same if S is monotonic:

$$\begin{array}{ll} & S.true \\ \Rightarrow & \quad \{\;\; F := true \;\;\} \\ & \langle\, \exists F \;\triangleright\; S.F \,\rangle \\ \Rightarrow & \quad \{\;\; [F \;\Rightarrow\; true] \text{, so if } S \text{ is monotonic, then } [S.F \;\Rightarrow\; S.true] \;\;\} \\ & \langle\, \exists F \;\triangleright\; S.true \,\rangle \\ = & \quad \{\;\; \text{predicate calculus (since range of } F \text{ is nonempty)} \;\;\} \\ & S.true \end{array}$$

This calculation shows that if S is monotonic, then

$$[S.true \;\equiv\; \langle\, \exists F \;\triangleright\; S.F \,\rangle]$$

The rest of this note goes as follows. In Section 1, I show that property (3) does not hold for all predicate transformers S if $\Box'$ is defined by (4). Section 2 introduces two special kinds of predicate transformers that will be used in the sections that follow. In Sections 3 and 4, I show that definition (5) of $\Box'$ does live up to properties (2) and (3).

1 Definition (4) is not the desired one

For the record, I show that property (3) does not always hold if $\Box'$ is defined by (5) and the domain of $\Box'$ is the set of all predicate transformers. For use in this section, I define two predicate transformers, N and $skip$, as follows:

$$\langle\, \forall F \;\triangleright\; [N.F \;\equiv\; \neg F] \,\rangle \tag{6}$$
$$\langle\, \forall F \;\triangleright\; [skip.F \;\equiv\; F] \,\rangle \tag{7}$$

Note that N is not monotonic and that $skip \in \mathcal{S}_0$. We massage property (3) in light of $\Box'$ definition (4):

$$\begin{array}{ll} & \langle\, \forall S, c \;|\; c \in \mathcal{S}_0 \;\triangleright\; S \sqsubseteq c \;\equiv\; \Box'.S \sqsubseteq c \,\rangle \\ \Rightarrow & \quad \{\;\; \text{instantiate } S := N \;\;\} \\ & \langle\, \forall c \;|\; c \in \mathcal{S}_0 \;\triangleright\; N \sqsubseteq c \;\equiv\; \Box'.N \sqsubseteq c \,\rangle \\ = & \quad \{\;\; \text{definition of refinement} \;\;\} \end{array}$$

$$
\begin{array}{ll}
 & \langle \forall\, c \mid c \in \mathcal{S}_0 \triangleright N \sqsubseteq c \;\equiv\; \langle \forall\, F \triangleright [\Box'.N.F \;\Rightarrow\; c.F] \rangle \rangle \\
= & \quad \{ \text{ (4): definition of } \Box' \ \} \\
 & \langle \forall\, c \mid c \in \mathcal{S}_0 \triangleright N \sqsubseteq c \;\equiv\; \langle \forall\, F \triangleright [N.true \wedge \Box.N.F \;\Rightarrow\; c.F] \rangle \rangle \\
= & \quad \{ \text{ (6): definition of } N \ \} \\
 & \langle \forall\, c \mid c \in \mathcal{S}_0 \triangleright N \sqsubseteq c \;\equiv\; \langle \forall\, F \triangleright [false \wedge \Box.N.F \;\Rightarrow\; c.F] \rangle \rangle \\
= & \quad \{ \text{ predicate calculus } \} \\
 & \langle \forall\, c \mid c \in \mathcal{S}_0 \triangleright N \sqsubseteq c \rangle \\
= & \quad \{ \text{ definition of refinement } \} \\
 & \langle \forall\, c \mid c \in \mathcal{S}_0 \triangleright \langle \forall\, F \triangleright [N.F \;\Rightarrow\; c.F] \rangle \rangle \\
\Rightarrow & \quad \{ \text{ instantiate } c, F := skip, false \ \} \\
 & [N.false \;\Rightarrow\; skip.false] \\
= & \quad \{ \text{ (6) and (7): definitions of } N \text{ and } skip \ \} \\
 & [true \;\Rightarrow\; false] \\
= & \quad \{ \text{ predicate calculus } \} \\
 & false
\end{array}
$$

The rest of this note uses (5), not (4), as the definition of $\Box'$.

2 Specification statements and lifted predicates

In section section, I state a few properties of two particular kinds of predicate transformers: specification statements and lifted predicates.

A *specification statement* is a triple $w{:}[P, Q]$, where w is the set of programs variables of the state space under consideration, P is predicate on the pre-state, and Q is a two-state predicate, that is, a relation on the pre-state and post-state (in Q, w_0 and w refer to the pre- and post-state values, respectively, of the variables). A specification statement is defined as a predicate transformer in the following way [1]: For any post-state predicate F,

$$[\ (w{:}[P, Q]).F \;\equiv\; P \wedge \langle \forall\, w \mid Q \triangleright F \rangle [w_0 := w]\ ] \tag{8}$$

Manohar and I have proved the following properties [0]:

$$\mathcal{S}_0 \;=\; \{ P, Q \triangleright w{:}[P, Q] \} \tag{9}$$

$$\mathcal{C}^{\top}_{\wedge} \;=\; \{ Q \triangleright w{:}[true, Q] \} \tag{10}$$

For convenience, I also define another predicate transformer: For any predicate P, the *lifted predicate* $Lift.P$ is defined by

$$\langle \forall\, F \triangleright [Lift.P.F \;\equiv\; P] \rangle \tag{11}$$

Lifted predicates turn out to be handy when combined with demonic choice, written $\sqcap$. First, we can lift definition (5) of $\Box'$ and write it in terms of predicate transformers: For any predicate transformer S,

$$\Box'.S \;=\; \mathit{Lift}.\langle\, \exists F \triangleright S.F \,\rangle \sqcap \Box.S \tag{12}$$

Also, we can state the following property of specification statements:

$$\langle\, \forall P, Q \triangleright w{:}[P, Q] \;=\; \mathit{Lift}.P \sqcap w{:}[\mathit{true}, Q] \,\rangle \tag{13}$$

which follows from the definition of the specification statement, (8). Finally, we can prove the following nice property: For any predicate P and predicate transformer S,

$$S \sqsubseteq \mathit{Lift}.P \;\equiv\; [\langle\, \exists F \triangleright S.F \,\rangle \;\Rightarrow\; P] \tag{14}$$

The proof of (14) goes as follows:

$$\begin{array}{ll} & S \sqsubseteq \mathit{Lift}.P \\ = & \quad \{\ \text{definition of refinement}\ \} \\ & \langle\, \forall F \triangleright [S.F \;\Rightarrow\; \mathit{Lift}.P.F] \,\rangle \\ = & \quad \{\ \text{(11): definition of } \mathit{Lift}.P\ \} \\ & \langle\, \forall F \triangleright [S.F \;\Rightarrow\; P] \,\rangle \\ = & \quad \{\ \text{interchange quantifications } [\,] \text{ and } \forall\ \} \\ & [\langle\, \forall F \triangleright S.F \;\Rightarrow\; P \,\rangle] \\ = & \quad \{\ \text{predicate calculus}\ \} \\ & [\langle\, \exists F \triangleright S.F \,\rangle \;\Rightarrow\; P] \end{array}$$

3 Proof of property (2)

The proof of (2) is easy: For any predicate transformer S, we calculate,

$$\begin{array}{ll} & \Box'.S \in \mathcal{S}_0 \\ = & \quad \{\ \text{(12): lifted definition of } \Box'\ \} \\ & (\mathit{Lift}.\langle\, \exists F \triangleright S.F \,\rangle \sqcap \Box.S) \in \mathcal{S}_0 \\ = & \quad \{\ \text{(0) and (10) and some } Q\ \} \\ & (\mathit{Lift}.\langle\, \exists F \triangleright S.F \,\rangle \sqcap w{:}[\mathit{true}, Q]) \in \mathcal{S}_0 \\ = & \quad \{\ (13)\ \} \\ & w{:}[\langle\, \exists F \triangleright S.F \,\rangle, Q] \in \mathcal{S}_0 \\ = & \quad \{\ (9)\ \} \\ & \mathit{true} \end{array}$$

4 Proof of property (3)

In this section, I prove property (3). It will be profitable to start by reviewing Galois connections and some things from Morgan's paper.

Given two partial orders $(X, \leq_X)$ and $(Y, \leq_Y)$, two functions $f: X \to Y$ and $g: Y \to X$ are said to form a *Galois connection*, written $gal.(f, g)$, when

$$\langle \forall x, y \mid x \in X \wedge y \in Y \triangleright f.x \leq_Y y \equiv x \leq_X g.y \rangle$$

If $gal.(f, g)$, then function f is called the *lower adjoint*, and g the *upper adjoint*, of the Galois connection (f, g).

Morgan proves a useful lemma about adjoints (labeled Lemma 6.1): For any Galois connection (f, g) as just described,

$$\langle \forall x, y \triangleright x \leq_X g.y \equiv (g \circ f).x \leq_X g.y \rangle \tag{15}$$

Morgan considers two partial orders (complete lattices, in fact), $(\mathcal{T}, \sqsubseteq)$ and $(\mathcal{R}, \subseteq)$, where $\mathcal{T}$ denotes the set of all predicate transformers from post-states to pre-states, $\mathcal{R}$ denotes the set of all relations on pre- and post-states, and $\subseteq$ denotes relational containment. Morgan defines two functions $\underline{A}: \mathcal{T} \to \mathcal{R}$ and $\overline{A}: \mathcal{R} \to \mathcal{T}$. The definition of $\overline{A}$ is given operationally as follows: For any relation r, post-state predicate F, and pre-state p,

$$\overline{A}.r.F.p \equiv \langle \forall f \mid p\langle r \rangle f \triangleright F.f \rangle \tag{16}$$

Function $\underline{A}$ is defined in such a way that $(\underline{A}, \overline{A})$ forms a Galois connection: For every relation r and predicate transformer S,

$$\underline{A}.S \subseteq r \equiv S \sqsubseteq \overline{A}.r \tag{17}$$

(As Morgan shows, such a lower adjoint does exist, and a property of Galois connections is that if a function has a lower adjoint, it is unique.) Morgan then derives a definition of function $\Box$ in terms of $\underline{A}$ and $\overline{A}$:

$$\Box = \overline{A} \circ \underline{A} \tag{18}$$

I define two functions $\underline{B}$ and $\overline{B}$ that will be shown to form a Galois connection. First, $\overline{B}$ is defined operationally as follows: For any pre-state predicate P, relation r, post-state predicate F, and pre-state p,

$$\overline{B}.(P, r).F.p \equiv P.p \wedge \langle \forall f \mid p\langle r \rangle f \triangleright F.f \rangle \tag{19}$$

We calculate,

$$
\begin{array}{ll}
 & \overline{B}.(P,r).F.p \\
= & \quad \{ \ \text{(19): operational definition of } \overline{B} \ \} \\
 & P.p \ \wedge \ \langle \forall f \ | \ p\langle r\rangle f \ \triangleright \ F.f \ \rangle \\
= & \quad \{ \ \text{(16): operational definition of } \overline{A} \ \} \\
 & P.p \ \wedge \ \overline{A}.r.F.p
\end{array}
$$

This calculation establishes: For any r, P, and F,

$$[\overline{B}.(P,r).F \ \equiv \ P \wedge \overline{A}.r.F]$$

In fact, we can rewrite this property in a lifted way: For any relation r and predicate P,

$$\overline{B}.(P,r) \ = \ \mathit{Lift}.P \ \square \ \overline{A}.r \tag{20}$$

Morgan proves that the image of $\overline{A}$ is exactly the set of universally conjunctive predicate transformers:

$$\{ r \ \triangleright \ \overline{A}.r \ \} \ = \ \mathcal{C}^{\top}_{\wedge} \tag{21}$$

We wish to establish that the image of $\overline{B}$ is exactly the set of positively conjunctive predicate transformers:

$$\{ r, P \ \triangleright \ \overline{B}.(P,r) \ \} \ = \ \mathcal{S}_0 \tag{22}$$

Property (22) follows from (9) and the following calculation:

$$
\begin{array}{ll}
 & \overline{B}.(P,r) \\
= & \quad \{ \ \text{(20): lifted definition of } \overline{B} \ \} \\
 & \mathit{Lift}.P \ \square \ \overline{A}.r \\
= & \quad \{ \ \text{(21) and (10) and some } Q\text{; or looked at from the other direction:} \\
 & \qquad \text{(10) and (21) and some } r \ \} \\
 & \mathit{Lift}.P \ \square \ w{:}[\mathit{true}, Q] \\
= & \quad \{ \ \text{(13)} \ \} \\
 & w{:}[P, Q]
\end{array}
$$

I now define $\underline{B}$, which maps predicate transformers to relation-predicate pairs: For any predicate transformer S,

$$\underline{B}.S \ = \ (\langle \exists F \ \triangleright \ S.F \ \rangle, \underline{A}.S) \tag{23}$$

For use later, I also define an ordering $\leq$ on relation-predicate pairs: For any predicates P and P' and relations r and r',

$$(P,r) \leq (P',r') \ \equiv \ [P \ \Rightarrow \ P'] \wedge r \subseteq r' \tag{24}$$

We can now prove a property about $\underline{B}$, $\overline{B}$, and $\Box'$, corresponding to Morgan's property (18) about $\underline{A}$, $\overline{A}$, and $\Box$, namely

$$\overline{B} \circ \underline{B} \;=\; \Box' \tag{25}$$

For any predicate transformer S, we calculate,

$$\begin{array}{ll}
 & (\overline{B} \circ \underline{B}).S \\
= & \quad \{ \text{ function composition } \} \\
 & \overline{B}.(\underline{B}.S) \\
= & \quad \{ \text{ (23): definition of } \underline{B} \ \} \\
 & \overline{B}.(\langle\, \exists F \,\triangleright\, S.F \,\rangle, \underline{A}.S) \\
= & \quad \{ \text{ (20): lifted definition of } \overline{B} \ \} \\
 & \mathit{Lift}.\langle\, \exists F \,\triangleright\, S.F \,\rangle \;\square\; \overline{A}.(\underline{A}.S) \\
= & \quad \{ \text{ (18): } \overline{A} \circ \underline{A} \;=\; \Box \ \} \\
 & \mathit{Lift}.\langle\, \exists F \,\triangleright\, S.F \,\rangle \;\square\; \Box.S \\
= & \quad \{ \text{ (12): lifted definition of } \Box' \ \} \\
 & \Box'.S
\end{array}$$

To prove (3), I prove that $(\underline{B}, \overline{B})$ forms a Galois connection and then, following Morgan's footsteps in proving (1), apply lemma (15). Let's start by showing $\mathit{gal}.(\underline{B}, \overline{B})$: For any predicate P, relation r, and predicate transformer S, we calculate,

$$\begin{array}{ll}
 & \underline{B}.S \le (P, r) \\
= & \quad \{ \text{ (23): definition of } \underline{B} \ \} \\
 & (\langle\, \exists F \,\triangleright\, S.F \,\rangle, \underline{A}.S) \le (P, r) \\
= & \quad \{ \text{ (24): definition of } \le \ \} \\
 & [\langle\, \exists F \,\triangleright\, S.F \,\rangle \;\Rightarrow\; P] \;\wedge\; \underline{A}.S \subseteq r \\
= & \quad \{ \text{ lemma (14) } \} \\
 & S \sqsubseteq \mathit{Lift}.P \;\wedge\; \underline{A}.S \subseteq r \\
= & \quad \{ \text{ (17): } \mathit{gal}.(\underline{A}, \overline{A}) \ \} \\
 & S \sqsubseteq \mathit{Lift}.P \;\wedge\; S \sqsubseteq \overline{A}.r \\
= & \quad \{ \text{ lifting } \} \\
 & S \sqsubseteq \mathit{Lift}.P \;\square\; \overline{A}.r \\
= & \quad \{ \text{ (20): lifted definition of } \overline{B} \ \} \\
 & S \sqsubseteq \overline{B}.(P, r)
\end{array}$$

Using lemma (15), we can now prove property (3): For any predicate transformer S,

$$
\begin{array}{ll}
 & \langle\, \forall\, c \mid c \in \mathcal{S}_0 \,\triangleright\, S \sqsubseteq c \;\equiv\; \Box'.S \sqsubseteq c \,\rangle \\
= & \quad \{\; \text{(22): image of } \overline{B} \text{ is } \mathcal{S}_0 \;\} \\
 & \langle\, \forall\, P, r \,\triangleright\, S \sqsubseteq \overline{B}.(P,r) \;\equiv\; \Box'.S \sqsubseteq \overline{B}.(P,r) \,\rangle \\
= & \quad \{\; (25):\ \overline{B} \circ \underline{B} \;=\; \Box' \;\} \\
 & \langle\, \forall\, P, r \,\triangleright\, S \sqsubseteq \overline{B}.(P,r) \;\equiv\; (\overline{B} \circ \underline{B}).S \sqsubseteq \overline{B}.(P,r) \,\rangle \\
= & \quad \{\; \text{Morgan's lemma (15), with } f, g := \underline{B}, \overline{B} \text{, since } gal.(\underline{B}, \overline{B}) \;\} \\
 & \mathit{true}
\end{array}
$$

5 Epilogue

It is widely believed that monotonic predicate transformers have meaningful counterparts among computer programs, and that predicate transformers that are not monotonic do not. For a monotonic predicate transformer S, the predicate $S.\mathit{true}$ characterizes those pre-states from which execution of S will terminate. In the words of Rajit Manohar, $\langle\, \exists\, F \,\triangleright\, S.F \,\rangle$ characterizes those pre-states from which S "can be initiated". The notion of "can be initiated" applies to all predicate transformers, not just monotonic ones.

I have not previously seen the trick of using $\langle\, \exists\, F \,\triangleright\, S.F \,\rangle$ instead of $S.\mathit{true}$ in order to extend an argument about monotonic predicate transformers to all predicate transformers. An application of this trick turned out to be precisely what was required in problem described in this note.

References

[0] K. Rustan M. Leino and Rajit Manohar. Joining specification statements. *Theoretical Computer Science*, December 1997. To appear.

[1] Carroll Morgan. The specification statement. *ACM Transactions on Programming Languages and Systems*, 10(3):403–419, July 1988.

[2] Carroll Morgan. The cuppest capjunctive capping, and Galois. In A. W. Roscoe, editor, *A Classical Mind: Essays in Honour of C.A.R. Hoare*, International Series in Computer Science, pages 317–332. Prentice-Hall, 1994.