# Proving Post's Functional Completeness Theorem Constructively

Evan Marzion
ILLC, Universiteit van Amsterdam
emarzion@gmail.com

**Abstract**

We modify Pelletier and Martin's proof of Post's Theorem on functionally complete sets of connectives so that it may be proven in the proof assistant Coq using constructive logic. Results on sets or types which enjoy the full duality between $\forall$ and $\exists$ are established in order to aid the extraction of witnesses from negated universally quantified propositions.

## 1 Background

Let $A$ be a set.

**Definition 1.1.** A **clone** $\mathcal{C}$ on $A$ is a set of operations of the form $A^n \to A$, for $n$ arbitrary, such that

1. For every $n \geq 1$ and $i \in \{1, \ldots, n\}$, $\mathcal{C}$ contains all projection functions $\pi_i^n$ given by the equation

$$\pi_i^n(a_1, \ldots, a_n) = a_i.$$

2. $\mathcal{C}$ is closed under generalized composition: If $f \in \mathcal{C}$ is $n$-ary and $g_1, \ldots, g_n \in \mathcal{C}$ are $k$-ary, then the function $f \circ [g_1, \ldots, g_n]$ given by the rule

$$(x_1, \ldots, x_k) \mapsto f(g_1(x_1, \ldots, x_k), \ldots, g_n(x_1, \ldots, x_k))$$

is in $\mathcal{C}$ as well.

**Definition 1.2.** The **full clone** on $A$, denoted $\mathcal{C}(A)$, is the set of all operations of the form $A^n \to A$, which naturally forms a clone.

**Definition 1.3.** Given $X \subseteq \mathcal{C}(A)$, the **closure** of $X$ is the smallest clone containing all elements of $X$.

**Definition 1.4.** A subset $X \subseteq \mathcal{C}(A)$ is said to be **functionally complete** if the closure of $X$ is all of $\mathcal{C}(A)$.

Let $\mathbf{B} = \{\mathsf{F}, \mathsf{T}\}$ denote the set of boolean truth values. In classical propositional logic, the semantics of an $n$-ary logical connective $C$ may be given in terms of a truth function (or truth table) $\dot{C} : \mathbf{B}^n \to \mathbf{B}$.

With this in mind, the notion of definability from a set of connectives can be cast in terms of clones over $\mathbf{B}$. For instance, a formula with two atomic letters $p$ and $q$ can now be thought of as generating a truth function carrying two arguments. As an example, the formula $\neg p \vee q$ can be thought of as the function of type $\mathbf{B}^2 \to \mathbf{B}$ given by

$$(p, q) \mapsto \dot{\vee}\big(\dot{\neg}(\pi_1^2(p, q)), (\pi_2^2(p, q))\big).$$

Presented in this fashion, the fact that $\to$ is definable from $\neg, \vee$ is recast as a statement that $\dot{\to}$ belongs to the closure of $\{\dot{\neg}, \dot{\vee}\}$.

From here on out, we shall conflate logical connectives with truth functions, and will write $\wedge$ in place of $\dot{\wedge}$, and so on.

In [1], Post characterizes the entire lattice structure of clones on the two-element set $\mathbf{B}$, and in doing so provides a simple characterization of all functionally complete sets of connectives. For a modern exposition on Post's lattice, we refer the reader to Lau [2], and for a modern proof of Post's characterization of functional completeness, we refer the reader to Pelletier and Martin [3]. In our formalization of the theorem, we follow the proof in [3].

In order to state Post's result on functional completeness, we must first define five classes of boolean-valued functions:

**Definition 1.5.** A function $f$ is **false-preserving** if $f(\mathsf{F}, \ldots, \mathsf{F}) = \mathsf{F}$.

**Definition 1.6.** A function $f$ is **truth-preserving** if $f(\mathsf{T}, \ldots, \mathsf{T}) = \mathsf{T}$.

**Definition 1.7.** Given a vector of booleans $\mathbf{x}$, let $\sim\!\mathbf{x}$ denote the vector resulting from negating each entry. A function $f$ of arity $n$ is **self-dual** if for all $\mathbf{x}$ of length $n$, $f(\sim\!\mathbf{x}) = \neg f(\mathbf{x})$.

**Definition 1.8.** Let $\leq$ denote the standard order on $\mathbf{B}$ given by $\mathsf{F} \leq \mathsf{T}$, and $\leq_n$[1] the product ordering on $\mathbf{B}^n$. A function $f$ of arity $n$ is **monotone** if for all $\mathbf{x}, \mathbf{y}$ of length $n$,

$$\mathbf{x} \leq_n \mathbf{y} \Rightarrow f(\mathbf{x}) \leq f(\mathbf{y}).$$

[1] elsewhere in this paper, the subscript $n$ will not be written where the length of the vectors in question is understood.

**Definition 1.9.** Given a vector $\mathbf{x}$ of length $n$ and $1 \leq i \leq n$, let $\mathbf{x}^{\neg i}$ denote the vector resulting from negating the $i$th entry of $\mathbf{x}$. Given a function $f$ of arity $n$, $i$ is said to be a dummy index (with respect to $f$) if for all $\mathbf{x}$,

$$f(\mathbf{x}^{\neg i}) = f(\mathbf{x}).$$

Likewise, $i$ is a counted index if for all $\mathbf{x}$,

$$f(\mathbf{x}^{\neg i}) = \neg f(\mathbf{x}).$$

A function $f$ is then said to be **counting** if every index is either a dummy index or a counted index with respect to $f$.

With these five classes of functions in mind, we may state Post's Theorem as follows:

**Theorem 1.10.** *A set of connectives $X$ is functionally complete if and only if $X$ contains functions $f_1, f_2, f_3, f_4, f_5$ which are not false-preserving, truth-preserving, self-dual, monotone, and counting, respectively.*

# 2 Formalizing Post's Theorem

We shall now run through Pelletier and Martin's proof, highlighting difficulties in their arguments which arise when formalization and constructive logic is considered.

## 2.1 Definitions

Given a set $A$ and a natural number $n$, we inductively define the set of $n$-length $A$ vectors as the $n$-times Cartesian product of $A$. In Coq:

```
Fixpoint vec (A : Set)(n : nat) : Set :=
  match n with
  | 0    => unit
  | S m  => (A * (vec m))%type
  end.
```

We shall consider vectors of booleans as the domains of our logical connectives.

Given a number $n$, we inductively define a canonical set of $n$ elements, called $\text{fin}(n)$, as the $n$-times disjoint sum of the unit set. In Coq:

```
Fixpoint fin (n : nat) : Set :=
  match n with
  | 0   => empty
  | S n => (unit + fin n)%type
  end.
```

where `empty` denotes the empty set. These sets shall serve the purpose of indexing our vectors.

## 2.2 Decidability Concerns

Throughout this section, we shall let $\mathrm{V}^n(A)$ denote the set of $n$-length $A$ vectors, $\varnothing$ the empty set, $\mathbf{1}$ the unit set, and $\star$ the unique element of $\mathbf{1}$. Given $A, B$ sets, $A + B$ shall denote their disjoint union, $A \times B$ their Cartesian product, and $\iota_1 : A \to A + B$, $\iota_2 : B \to A + B$ the canonical inclusion maps.

As we are working within a constructive framework, there are occasions where extra care needs to be taken in order to ensure that our arguments are valid constructively. For instance, in [3], there are a number of cases where one needs to 'flip' a negated universal quantifier to an existential. As an example, if $f$ is a non-self-dual connective, then we will need to find a vector $\mathbf{v}$ such that $f(\mathbf{v}) \neq f(\sim\mathbf{v})$. In logical notation, we will need to show that

$$\Big(\neg\forall \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\mathbf{x}) = f(\sim\mathbf{x})\Big) \to \Big(\exists \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\mathbf{x}) \neq f(\sim\mathbf{x})\Big).$$

Here, we run into a problem, since the logical principle

$$\Big(\neg\forall x \in A, \Phi(x)\Big) \to \Big(\exists x \in A, \neg\Phi(x)\Big)$$

is generally not constructively valid. However, when $A$ is finite and $\Phi$ is computable, it seems that it should be valid — after all, one could just run through every element $a \in A$ and check whether or not $\Phi(a)$ holds. As it turns out, this intuition is correct.

In this section, we shall show that sets with certain nice computational properties are closed under set operations such as disjoint union and Cartesian product. Later on, we will use these results to extract existential witnesses from negative statements.

In constructive type theory, the topic of sets which in some sense have computable quantifiers has been studied by Martín Escardó. We refer the interested reader to [4].

As our underlying logic is intuitionistic, the following definitions are non-trivial:

**Definition 2.1.** A proposition $P$ is **decidable** if $P \vee \neg P$.

**Definition 2.2.** A predicate $\Phi$ over a set $A$ is **decidable** if $\forall x \in A, \Phi(x) \vee \neg\Phi(x)$.

**Definition 2.3.** A set $A$ has the **existential excluded middle (ExEM) property** if for any decidable $\Phi$ over $A$, the proposition $\exists x \in A, \Phi(x)$ is decidable.

**Lemma 2.4.** *$\varnothing$ has the ExEM property.*

*Proof.* For any $\Phi$, $\neg\exists x \in \varnothing, \Phi(x)$.

□

**Lemma 2.5.** *$\mathbf{1}$ has the ExEM property.*

*Proof.* Let $\Phi$ be decidable. Observe that $\Big(\exists x \in \mathbf{1}, \Phi(x)\Big) \leftrightarrow \Phi(\star)$. Thus, $\exists x \in \mathbf{1}, \Phi(x)$ is decidable since $\Phi(\star)$ is.

□

**Lemma 2.6.** *$\mathbf{B}$ has the ExEM property.*

*Proof.* Let $\Phi$ be decidable. Observe that $\Big(\exists x \in \mathbf{B}, \Phi(x)\Big) \leftrightarrow \Phi(\mathsf{F}) \vee \Phi(\mathsf{T})$. The right-hand side is decidable, so the left is.

□

**Lemma 2.7.** *If $A, B$ have the ExEM property, then $A + B$ has the ExEM property.*

*Proof.* Let $\Phi$ be decidable. Observe that

$$\Big(\exists x \in A + B, \Phi(x)\Big) \leftrightarrow \Big(\exists a \in A, \Phi(\iota_1(a)) \vee \exists b \in B, \Phi(\iota_2(b))\Big)$$

The decidability of each disjunct on the right-hand side follows from the ExEM properties for $A$ and $B$ (and the fact that $\Phi \circ \iota_1$, $\Phi \circ \iota_2$ are both decidable). Thus, the left-hand side is also decidable.

□

**Lemma 2.8.** *If $A, B$ have the ExEM property, then $A \times B$ has the ExEM property.*

*Proof.* Let $\Phi$ be decidable. Observe that

$$\Big(\exists x \in A \times B, \Phi(x)\Big) \leftrightarrow \Big(\exists a \in A, \exists b \in B, \Phi(a, b)\Big)$$

Since $A$ has the ExEM property, in order to show that the right-hand side is decidable, it suffices to show that the predicate $\exists b \in B\Phi(a, b)$ is decidable. However this follows from the ExEM property for $B$ as well as the decidability of $\Phi$.

□

**Lemma 2.9.** *For all $n$, fin$(n)$ has the ExEM property.*

*Proof.* Induction on $n$, using the fact that $\varnothing$ and $\mathbf{1}$ have the ExEM property, and that the ExEM property is closed under disjoint union.

$\square$

**Lemma 2.10.** *If $A$ has the ExEM property, then for any $n$, $V^n(A)$ has the ExEM property.*

*Proof.* Induction on $n$, using the fact that $\mathbf{1}$ has the ExEM property and that the ExEM property is closed under Cartesian products. $\square$

**Definition 2.11.** A set $A$ has the **quantifier duality (QD) property** if for any decidable $\Phi$ over $A$,

$$\Big(\neg\forall x \in A, \Phi(x)\Big) \rightarrow \Big(\exists x \in A, \neg\Phi(x)\Big)$$

holds.

**Lemma 2.12.** *All sets which have the ExEM property have the QD property.*

*Proof.* Suppose $A$ has the ExEM property. Let $\Phi$ be decidable, and suppose $\neg\forall x \in A, \Phi(x)$. Since $\Phi$ is decidable, so is the negation of $\Phi$. Thus,

$$\exists x \in A, \neg\Phi(x) \vee \neg\exists x \in A, \neg\Phi(x)$$

If the left disjunct holds, we are done. Suppose then $\neg\exists x \in A, \neg\Phi(x)$. From this we may conclude that $\forall x \in A, \neg\neg\Phi(x)$. Since $\Phi$ is decidable, we may eliminate the double negation and arrive at a contradiction.

$\square$

**Definition 2.13.** A set $A$ has the **universal excluded middle (UnEM)** property if for any decidable $\Phi$ over $A$, $\forall x \in A, \Phi(x)$ is decidable.

**Lemma 2.14.** *All sets which have the ExEM property have the UnEM property.*

*Proof.* Suppose $A$ has the ExEM property and $\Phi$ is decidable. Since $\Phi$ is decidable, so is its negation. Thus,

$$\exists x \in A, \neg\Phi(x) \vee \neg\exists x \in A, \neg\Phi(x).$$

The left disjunct implies that $\neg\forall x \in A, \Phi(x)$, and the right disjunct implies that $\forall x \in A, \Phi(x)$.

$\square$

**Lemma 2.15.** *The properties of being false-preserving, truth-preserving, self-dual, monotone, and counting are all decidable.*

*Proof.* The decidability of the first two is trivial.

Since

$$f \text{ is self-dual} \equiv \forall \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\sim\mathbf{x}) = \neg f(\mathbf{x}),$$

decidability follows from the UnEM property for $\mathrm{V}^n(\mathbf{B})$ and the fact that equality for booleans is decidable.

Since

$$f \text{ is monotone} \equiv \forall \mathbf{x}, \mathbf{y} \in \mathrm{V}^n(\mathbf{B}), \mathbf{x} \leq \mathbf{y} \to f(\mathbf{x}) \leq f(\mathbf{y}),$$

decidability follows from the UnEM property for $\mathrm{V}^n(\mathbf{B})$ and the fact that $\leq$ is a decidable relation for vectors and booleans.

Since

$$f \text{ is counting} \equiv$$
$$\forall i \in \mathrm{fin}(n) \Big( \forall \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\mathbf{x}) = f(\mathbf{x}^{\neg i}) \vee \forall \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\mathbf{x}) \neq f(\mathbf{x}^{\neg i}) \Big),$$

decidability follows from the UnEM properties for $\mathrm{fin}(n)$ and $\mathrm{V}^n(\mathbf{B})$ and the fact that equality for booleans is decidable.

□

**Remark 2.16.** Given connectives which are not self-dual, monotone, or counting, we can extract witnesses to their negative properties. For instance, if $f$ is not self-dual, then

$$\neg \forall \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\sim\mathbf{x}) = \neg f(\mathbf{x}).$$

Since $\mathrm{V}^n(\mathbf{B})$ has the QD property, we may conclude from this that

$$\exists \mathbf{x} \in \mathrm{V}^n(\mathbf{B}), f(\sim\mathbf{x}) = f(\mathbf{x}).$$

Similar statements can be proven when we are given functions which are not monotone or not counting, and this fact will be freely used in the first half of our proof of Post's Theorem.

## 2.3 Defining Definability

In [3], the notion of definability from a set of connectives is treated intuively and, hence, informally. We therefore follow Post and use boolean clones as the basis of our definability predicate:[2]

[2] All relevant Coq definitions may be found in the appendix.

```
Inductive Definable(X : forall n:nat, conn n -> Prop) :
  forall k, conn k -> Prop :=
  | atom_def : forall (n : nat)(f : conn n), X n f
               -> Definable X f
  | null_def : forall x : bool, Definable X (const 1 x)
               -> Definable X (const 0 x)
  | project  : forall (n : nat) (i : fin n),
               Definable X (proj n i)
  | compose  : forall (n k : nat) (f : conn n)
               (gs : vec (conn k) n),
               Definable X f -> (forall i : fin n,
               Definable X (item_at (conn k) n i gs))
               -> Definable X (comp k f gs)
  | def_ext  : forall (n : nat) (f g : conn n),
               Definable X f -> f [=] g -> Definable X g.
```

The `atom_def` constructor states that an element of `X` is definable from `X`. Constructors `project` and `compose` state that the projection functions are always defined, and that definable functions are closed under generalized composition. The `def_ext` constructor allows us to treat our connectives extensionally; to be precise, it states that definable connectives are closed under extensional equivalence. One could also chose to add function extensionality as an axiom in order to avoid needing this constructor, but in the interest of proving the result without additional assumptions, this approach was not taken.

The `null_def` constructor requires elaboration. Technically, the nullary connectives (which in this case are just the two booleans) cannot be defined from `X` unless `X` already contains a boolean. For instance, the formula $p \wedge \neg p$ is often thought of as a definition of $\mathsf{F}$, but if $p$ is being viewed as a variable, then this formula is more appropriately viewed as defining the unary function $\lambda p.\mathsf{F}$. Post noted this and chose to remove nullary connectives from consideration. However, we choose to consider definitions for $\lambda x.\mathsf{F}$ and $\lambda x.\mathsf{T}$ to be 'just as good' as definitions for $\mathsf{F}$ and $\mathsf{T}$, and `null_def` reflects this choice.

## 2.4 Completeness of $\wedge, \neg$

Broadly, the approach taken by Pelletier and Martin is to show that boolean conjunction and negation are definable using functions outside of the five classes mentioned earlier. Thus, establishing the functional completeness of these two connectives is the first step in formalizing their proof.

Before we begin, a quick lemma:

**Lemma 2.17.** *If $g$ is $n$-ary definable, then $\lambda x, \mathbf{y}.g(\mathbf{y})$ is $n+1$-ary definable.*

*Proof.* Observe that

$$\lambda x, \mathbf{y}.g(\mathbf{y}) = g \circ \left[\pi_2^{n+1}, \ldots, \pi_{n+1}^{n+1}\right].$$

□

**Theorem 2.18.** $\{\wedge, \neg\}$ *is functionally complete.*

*Proof.* We prove this using induction on the arity of the function being defined. The key observation is that for $n+1$-ary $f$,

$$f = \lambda x, \mathbf{y}.(x \wedge f(\mathsf{T}, \mathbf{y})) \vee (\neg x \wedge f(\mathsf{F}, \mathbf{y})).$$

This gives us a definition $f$ from the functions $\wedge$,$\neg$,$\vee$, $\lambda x, \mathbf{y}.f(\mathsf{F}, \mathbf{y})$ and $\lambda x, \mathbf{y}.f(\mathsf{T}, \mathbf{y})$. The definability of the last two functions follows from the previous lemma and the inductive hypothesis.

□

## 2.5 Functional Completeness, First Direction

With functional completeness of $\wedge, \neg$ established, we can prove the first direction of our main theorem:

**Theorem 2.19.** *If $X$ contains connectives*

1. *$f_1$ which is not false-preserving*
2. *$f_2$ which is not truth-preserving*
3. *$f_3$ which is not self-dual*
4. *$f_4$ which is not monotone*
5. *$f_5$ which is not counting*

*then $X$ is functionally complete.*

The approach taken will be to progressively define more and more functions until we have constructed definitions for negation and conjunction.

**Lemma 2.20.** *From $f_1, f_2$, we may define either negation or both of the booleans.*

*Proof.* Given a function $f$, we can define a unary function $f^* := \lambda x.f(x, \ldots, x)$. To show that $f^*$ is definable from $f$, note that

$$f^* = f \circ \left[\pi_1^1, \ldots, \pi_1^1\right].$$

Since $f_1$ is not false-preserving, we see that $f_1^*(\mathsf{F}) = \mathsf{T}$. Depending on the value of $f_1^*(\mathsf{T})$, we see that $f_1^*$ is either negation or the constant true function. Similarly, $f_2^*$ is either negation or the constant false function.

□

**Lemma 2.21.** *From $f_1, f_2, f_4$, we may define negation.*

*Proof.* By the previous lemma, we either have negation defined (in which case we are done) or both of the booleans defined.

The argument made by Pelletier and Martin is as follows: $f_4$ is non-monotone, so there are vectors $\mathbf{v} \leq \mathbf{w}$ with $f_4(\mathbf{v}) = \mathsf{T}$ and $f_4(\mathbf{w}) = \mathsf{F}$. Without loss of generality, we can assume that $\mathbf{v}, \mathbf{w}$ differ in at most one position. Thus, $\mathbf{v} = (\mathbf{v}_1, \mathsf{F}, \mathbf{v}_2)$ and $\mathbf{w} = (\mathbf{v}_1, \mathsf{T}, \mathbf{v}_2)$. With this in mind, we define negation to be

$$\lambda x. f_4(\mathbf{v}_1, x, \mathbf{v}_2).$$

This is defined, since $f_4$ is defined, and both constants are defined, so $\mathbf{v}_1, \mathbf{v}_2$ can be 'hard-coded' into the definition using the constants $\mathsf{F}, \mathsf{T}$.

While this is a fine argument, it seems difficult to formalize. Thus, we develop a modified argument:

**Definition 2.22.** Given two vectors $\mathbf{v}, \mathbf{w}$ of length $n$, define $\star(\mathbf{v}, \mathbf{w})$, a vector of unary functions of length $n$, as follows:

$$\star(\mathbf{v}, \mathbf{w})_i = \begin{cases} \lambda x.\mathsf{F} & \mathbf{v}_i = \mathsf{F}, \mathbf{w}_i = \mathsf{F} \\ \pi_1^1 & \mathbf{v}_i = \mathsf{F}, \mathbf{w}_i = \mathsf{T} \\ \neg & \mathbf{v}_i = \mathsf{T}, \mathbf{w}_i = \mathsf{F} \\ \lambda x.\mathsf{T} & \mathbf{v}_i = \mathsf{T}, \mathbf{w}_i = \mathsf{T} \end{cases}$$

For instance, if $\mathbf{v} = (\mathsf{F}, \mathsf{F}, \mathsf{T}, \mathsf{T})$ and $\mathbf{w} = (\mathsf{F}, \mathsf{T}, \mathsf{F}, \mathsf{T})$, then

$$\star(\mathbf{v}, \mathbf{w}) = \left[\lambda x.\mathsf{F}, \pi_1^1, \neg, \lambda x.\mathsf{T}\right].$$

The $\star$ operation is defined in this manner so that the following lemma holds:

**Lemma 2.23.** *Let $f$ be $n$-ary and $v, w$ vectors of length $n$. Then*

$$(f \circ \star(\mathbf{v}, \mathbf{w}))\,(\mathsf{F}) = f(\mathbf{v})$$

$$(f \circ \star(\mathbf{v}, \mathbf{w}))\,(\mathsf{T}) = f(\mathbf{w}).$$

From here, we can complete our alternative proof of lemma 2.21: Since $f_4$ is non-monotonic, there are vectors $\mathbf{v} \leq \mathbf{w}$ with $f_4(\mathbf{v}) = \mathsf{T}$ and $f_4(\mathbf{w}) = \mathsf{F}$. We can then define negation as $f_4 \circ (\star(\mathbf{v}, \mathbf{w}))$, since

$$(f_4 \circ \star(\mathbf{v}, \mathbf{w}))\,(\mathsf{F}) = f_4(\mathbf{v}) = \mathsf{T}$$

and

$$(f_4 \circ \star(\mathbf{v}, \mathbf{w}))\,(\mathsf{T}) = f_4(\mathbf{w}) = \mathsf{F}$$

Finally, we must establish that $f_4 \circ (\star(\mathbf{v}, \mathbf{w}))$ is indeed definable from $f_4$, $\lambda x.\mathsf{F}$,$\lambda x.\mathsf{T}$: Since $\mathbf{v} \leq \mathbf{w}$, each component of $\star(\mathbf{v}, \mathbf{w})$ is either $\lambda x.\mathsf{F}$, $\pi^1_1$, or $\lambda x.\mathsf{T}$. □

At this stage, we have a definition of negation in hand. We will now show how to use negation and $f_3$ to define the constants $\lambda x.\mathsf{F}$ and $\lambda x.\mathsf{T}$.

**Lemma 2.24.** *$\lambda x.\mathsf{F}$, $\lambda x.\mathsf{T}$ are definable from $f_1, f_2, f_3, f_4$.*

*Proof.* Since one can clearly be defined from the other using negation, it suffices to prove that one unary constant is definable.

Since $f_3$ is not self-dual, there is some vector $\mathbf{v}$ such that $f_3(\mathbf{v}) = f_3(\sim\mathbf{v})$. We now claim that $f_3 \circ (\star(\mathbf{v}, \sim\mathbf{v}))$ is constant and definable.

First, it is constant, since

$$(f_3 \circ \star(\mathbf{v}, \sim\mathbf{v}))\,(\mathsf{F}) = f_3(\mathbf{v}) = f_3(\sim\mathbf{v}) = (f_3 \circ \star(\mathbf{v}, \sim\mathbf{v}))\,(\mathsf{T}).$$

It is indeed definable, since every component of $\star(\mathbf{v}, \sim\mathbf{v})$ will be either $\pi^1_1$ or $\neg$. □

**Lemma 2.25.** *Conjunction is definable from $f_1, f_2, f_3, f_4, f_5$.*

*Proof.* The strategy here is to define a binary function which has either three $\mathsf{F}$ and one $\mathsf{T}$ or three $\mathsf{T}$ and one $\mathsf{F}$ in its truth table. There are eight such functions, and it is routine to verify that conjunction is definable from each of these eight functions together with negation.

In the interest of assisting formalization, we again introduce a new definition:

**Definition 2.26.** Given two vectors $\mathbf{v}, \mathbf{w}$ of length $n$ and $i$ such that $1 \leq i \leq n$, define $\clubsuit(\mathbf{v}, \mathbf{w}, i)$, a vector of binary functions of length $n$, as follows:

$$\clubsuit(\mathbf{v}, \mathbf{w}, i)_j = \begin{cases} \pi^2_2 & j = i \\ \lambda x, y.\mathsf{F} & j \neq i, \mathbf{v}_j = \mathsf{F}, \mathbf{w}_j = \mathsf{F} \\ \pi^2_1 & j \neq i, \mathbf{v}_j = \mathsf{F}, \mathbf{w}_j = \mathsf{T} \\ \lambda x, y.\neg x & j \neq j, \mathbf{v}_j = \mathsf{T}, \mathbf{w}_j = \mathsf{F} \\ \lambda x, y.\mathsf{T} & j \neq i, \mathbf{v}_j = \mathsf{T}, \mathbf{w}_j = \mathsf{T} \end{cases}$$

We have chosen this definition so that the following lemma holds:

**Lemma 2.27.** *Let $f$ be $n$-ary and let $v, w$ be vectors of length $n$, both with $\mathsf{F}$ in the $i$ position. Then*

$$(f \circ \clubsuit(\mathbf{v}, \mathbf{w}, i))\,(\mathsf{F}, \mathsf{F}) = f(\mathbf{v})$$

$$(f \circ \clubsuit(\mathbf{v}, \mathbf{w}, i))\,(\mathsf{T}, \mathsf{F}) = f(\mathbf{w})$$

$$\left(f \circ \clubsuit(\mathbf{v}, \mathbf{w}, i)\right)(\mathsf{F}, \mathsf{T}) = f(\mathbf{v}^{\neg i})$$

$$\left(f \circ \clubsuit(\mathbf{v}, \mathbf{w}, i)\right)(\mathsf{T}, \mathsf{T}) = f(\mathbf{w}^{\neg i})$$

To finish the proof, $f_5$ is not counting, so there must be some index $i$ which is neither dummy nor counted. Therefore, there are vectors $\mathbf{v}, \mathbf{w}$ such that

$$f_5(\mathbf{v}^{\neg i}) = \neg f_5(\mathbf{v})$$
$$f_5(\mathbf{w}^{\neg i}) = f_5(\mathbf{w}).$$

Without loss of generality, we may assume that $\mathbf{v}, \mathbf{w}$ have $\mathsf{F}$ in the $i$ position. Therefore, in light of the above lemma, we see that $f_5 \circ \clubsuit(\mathbf{v}, \mathbf{w}, i)$ has the required 3/1 split.

Lastly, we must verify that $f_5 \circ \clubsuit(\mathbf{v}, \mathbf{w}, i)$ is definable, but this immediate, since each component in $\clubsuit(\mathbf{v}, \mathbf{w}, i)$ is composed of projections, the booleans, and negation, all of which have been previously defined.

$\square$

## 2.6 Functional Completeness, Second Direction

Let $Y$ be a class of functions.

**Definition 2.28.** $Y$ is **closed** if the closure of $Y$ is $Y$ itself.

**Theorem 2.29.** *If $X$ is functionally complete, then $X$ contains connectives*

1. *$f_1$ which is not false-preserving*
2. *$f_2$ which is not truth-preserving*
3. *$f_3$ which is not self-dual*
4. *$f_4$ which is not monotone*
5. *$f_5$ which is not counting.*

Our proof will mainly use the following lemma:

**Lemma 2.30.** *Let $X$ be functionally complete, and suppose that $Y$ is closed, decidable, and doesn't contain all functions. Then there is some $f \in X$ such that $f \notin Y$.*

*Proof.* Suppose $f \notin Y$. We proceed using induction on the definition of $f$ from $X$.

If $f \in X$ then we are done.

If $f$ is a projection function, then $f \in Y$, since $Y$ is closed. This is a contradiction.

Finally, suppose that the definition for $f$ takes the form $f = f' \circ [g_1, \ldots, g_n]$.

Since $Y$ is closed, we have that

$$\neg\Big(f' \in Y \wedge \forall_{1 \leq i \leq n}. g_i \in Y\Big).$$

Since $Y$ is decidable, we may conclude that

$$f' \notin Y \vee \exists_{1 \leq i \leq n}. g_i \notin Y.$$

In either case, we use the inductive hypothesis to complete the proof. □

As we have previously shown, each of the five special classes of functions is decidable and does not contain every function. Thus, it suffices to show that these classes are all closed. This amounts to showing that they contain the projection functions and are closed under the generalized composition operation.

**Lemma 2.31.** *The false-preserving functions are closed.*

*Proof.* The projection functions are false-preserving, since

$$\pi_i^n(\mathsf{F}, \ldots, \mathsf{F}) = \mathsf{F}.$$

Suppose that $f$ is $n$-ary false-preserving, and $g_1, \ldots, g_n$ are $k$-ary false-preserving. Then, $f \circ [g_1, \ldots, g_n]$ is false-preserving:

$$\begin{aligned} f \circ [g_1, \ldots, g_n]\,(\mathsf{F}, \ldots, \mathsf{F}) &= f\big(g_1(\mathsf{F}, \ldots, \mathsf{F}), \ldots, g_n(\mathsf{F}, \ldots, \mathsf{F})\big) \\ &= f(\mathsf{F}, \ldots, \mathsf{F}) \\ &= \mathsf{F}. \end{aligned}$$

□

**Lemma 2.32.** *The truth-preserving functions are closed.*

*Proof.* The proof is the same as in the case of false-preserving. □

**Lemma 2.33.** *The self-dual functions are closed.*

*Proof.* The projections are self-dual, since

$$\pi_i^n(\sim\mathbf{v}) = \neg\mathbf{v}_i = \neg\pi_i^n(\mathbf{v})$$

Suppose that $f$ is $n$-ary self-dual, and $g_1, \ldots, g_n$ are $k$-ary self-dual. Then, $f \circ [g_1, \ldots, g_n]$ is self-dual:

$$\begin{aligned} f \circ [g_1, \ldots, g_n]\,(\sim\mathbf{v}) &= f\big(g_1(\sim\mathbf{v}), \ldots, g_n(\sim\mathbf{v})\big) \\ &= f\big(\neg g_1(\mathbf{v}), \ldots, \neg g_n(\mathbf{v})\big) \\ &= \neg f\big(g_1(\mathbf{v}), \ldots, g_n(\mathbf{v})\big) \\ &= \neg f \circ [g_1, \ldots, g_n]\,(\mathbf{v}). \end{aligned}$$

□

**Lemma 2.34.** *The monotone functions are closed.*

*Proof.* The projections are monotone: Suppose $\mathbf{v} \leq \mathbf{w}$. Then for any $i$ we have that $\mathbf{v}_i \leq \mathbf{w}_i$. Thus,

$$\pi_i^n(\mathbf{v}) = \mathbf{v}_i \leq \mathbf{w}_i = \pi_i^n(\mathbf{w})$$

Suppose that $f$ is $n$-ary monotone, and $g_1, \ldots, g_n$ are $k$-ary monotone. Then, $f \circ [g_1, \ldots, g_n]$ is monotone: Suppose $\mathbf{v} \leq \mathbf{w}$. Then for any $i$ we have that $g_i(\mathbf{v}) \leq g_i(\mathbf{w})$. Thus,

$$\big(g_1(\mathbf{v}), \ldots, g_n(\mathbf{v})\big) \leq \big(g_1(\mathbf{w}), \ldots, g_n(\mathbf{w})\big)$$

so by monotonicity of $f$,

$$\begin{aligned} f \circ [g_1, \ldots g_n]\,(\mathbf{v}) &= f\big(g_1(\mathbf{v}), \ldots, g_n(\mathbf{v})\big) \\ &\leq f\big(g_1(\mathbf{w}), \ldots, g_n(\mathbf{w})\big) \\ &= f \circ [g_1, \ldots g_n]\,(\mathbf{w}) \end{aligned}$$

□

Before we continue, let us recall a bit of algebra.

Recall that $\mathbf{B}$ can be made into a ring (and in fact a field) by taking exclusive disjunction to be addition and conjunction to be multiplication. This allows us define many familiar notions from linear algebra. In particular, given two vectors $\mathbf{v}, \mathbf{w}$, let $\mathbf{v} \cdot \mathbf{w}$ denote their dot product. We may now introduce the class of affine functions:

**Definition 2.35.** Given $\mathbf{v}$ a vector and $b$ a boolean, consider the function taking a vector $\mathbf{x}$ to $\mathbf{v} \cdot \mathbf{x} + b$. We shall call functions of this form **affine**.

As it turns out, the affine functions coincide precisely with the counting functions.

**Lemma 2.36.** *All affine functions are counting.*

*Proof.* For $1 \leq i \leq n$, let $\mathbf{e}_i^n$ denote the vector of length $n$ whose $i$ entry is $\mathsf{T}$ and all other entries are $\mathsf{F}$. Observe that for $\mathbf{v}$ a length $n$ vector,

$$\mathbf{v}^{\neg i} = \mathbf{v} + \mathbf{e}_i^n$$

and that

$$\mathbf{v} \cdot \mathbf{e}_i^n = \mathbf{v}_i.$$

Therefore, for any $\mathbf{v}, \mathbf{w}$ and $b$ we have that

$$\begin{aligned} \mathbf{v} \cdot \mathbf{w}^{\neg i} + b &= \mathbf{v} \cdot (\mathbf{w} + \mathbf{e}_i^n) + b \\ &= \mathbf{v} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{e}_i^n + b \\ &= \big(\mathbf{v} \cdot \mathbf{w} + b\big) + \mathbf{v}_i. \end{aligned}$$

If $\mathbf{v}_i = \mathsf{F}$, we then have that

$$\mathbf{v} \cdot \mathbf{w}^{\neg i} + b = \big(\mathbf{v} \cdot \mathbf{w} + b\big) + \mathsf{F} = \mathbf{v} \cdot \mathbf{w} + b$$

and if $\mathbf{v}_i = \mathsf{T}$,

$$\mathbf{v} \cdot \mathbf{w}^{\neg i} + b = \big(\mathbf{v} \cdot \mathbf{w} + b\big) + \mathsf{T} = \neg\big(\mathbf{v} \cdot \mathbf{w} + b\big).$$

Thus, $i$ is a dummy or counted variable for the function $\lambda \mathbf{x}.\mathbf{v} \cdot \mathbf{x} + b$, depending on the value of $\mathbf{v}_i$.

□

**Lemma 2.37.** *All counting functions are affine.*

*Proof.* Let $f$ be a counting function. We define the vector $\mathbf{v}^f$ as follows:

$$\mathbf{v}_i^f = \begin{cases} \mathsf{F} & i \text{ is a dummy variable for } f \\ \mathsf{T} & i \text{ is a counted variable for } f \end{cases}$$

We now claim that for any $\mathbf{w}$,

$$\mathbf{v}^f \cdot \mathbf{w} + f(\mathsf{F}, \dots \mathsf{F}) = f(\mathbf{w}).$$

We proceed using induction on the number of $\mathsf{T}$ entries in $\mathbf{w}$. If there are zero, then

$$\begin{aligned}\mathbf{v}^f \cdot \mathbf{w} + f(\mathsf{F}, \dots \mathsf{F}) &= \mathbf{v}^f \cdot (\mathsf{F}, \dots, \mathsf{F}) + f(\mathsf{F}, \dots, \mathsf{F}) \\ &= \mathsf{F} + f(\mathsf{F}, \dots, \mathsf{F}) \\ &= f(\mathsf{F}, \dots, \mathsf{F}) \\ &= f(\mathbf{w}).\end{aligned}$$

Suppose then that there are $k+1$ $\mathsf{T}$ entries in $\mathbf{w}$. Let $i$ be such an entry. Then

$$\begin{aligned}\mathbf{v}^f \cdot \mathbf{w} + f(\mathsf{F}, \dots, \mathsf{F}) &= \mathbf{v}^f \cdot (\mathbf{w}^{\neg i} + \mathbf{e}_i^n) + f(\mathsf{F}, \dots, \mathsf{F}) \\ &= \mathbf{v}^f \cdot \mathbf{w}^{\neg i} + f(\mathsf{F}, \dots, \mathsf{F}) + \mathbf{v}^f \cdot \mathbf{e}_i^n \\ &= f(\mathbf{w}^{\neg i}) + \mathbf{v}_i^f\end{aligned}$$

If $i$ is a dummy variable for $f$, then we have that

$$\begin{aligned}\mathbf{v}^f \cdot \mathbf{w} + f(\mathsf{F}, \dots, \mathsf{F}) &= f(\mathbf{w}^{\neg i}) + \mathbf{v}_i^f \\ &= f(\mathbf{w}) + \mathsf{F} \\ &= f(\mathbf{w})\end{aligned}$$

and if $i$ is a counted variable,

$$\begin{aligned}\mathbf{v}^f \cdot \mathbf{w} + f(\mathsf{F}, \dots, \mathsf{F}) &= f(\mathbf{w}^{\neg i}) + \mathbf{v}_i^f \\ &= \neg f(\mathbf{w}) + \mathsf{T} \\ &= f(\mathbf{w}).\end{aligned}$$

□

**Lemma 2.38.** *The counting functions are closed.*

*Proof.* The projection functions are counting, since for any $\mathbf{v}$,

$$\pi_i^n(\mathbf{v}) = \mathbf{e}_i^n \cdot \mathbf{v}.$$

Suppose that $f$ is $n$-ary counting, and $g_1, \dots, g_n$ are $k$-ary counting. Then we have that there are $\mathbf{u}$ and $b$ such that for any $\mathbf{x}$ of length $n$,

$$f(\mathbf{x}) = \mathbf{u} \cdot \mathbf{x} + b$$

and for each $g_i$ there are $\mathbf{v}^i$ and $c_i$ such that for any $\mathbf{y}$ of length $k$,

$$g_i(\mathbf{y}) = \mathbf{v}^i \cdot \mathbf{y} + c_i$$

Let $V$ denote the matrix with rows $v_1, \dots v_n$ and let $\mathbf{c} = (c_1, \dots, c_n)$. For any $\mathbf{y}$, we then have that

$$\begin{aligned} f \circ [g_1, \ldots, g_n]\,(\mathbf{y}) &= f\big(g_1(\mathbf{y}), \ldots, g_n(\mathbf{y})\big) \\ &= \mathbf{u} \cdot \big(\mathbf{v}_1 \cdot \mathbf{y} + c_1, \ldots, \mathbf{v}_n \cdot \mathbf{y} + c_n\big) + b \\ &= \mathbf{u} \cdot \big(V\mathbf{y} + \mathbf{c}\big) + b \\ &= (V^T\mathbf{u}) \cdot \mathbf{y} + (\mathbf{u} \cdot \mathbf{c} + b). \end{aligned}$$

Thus, $f \circ [g_1, \ldots, g_n]$ is affine and thus counting.

□

# References


[1] E. L. Post *The Two-Valued Iterative Systems of Mathematical Logic.* Annals of Mathematics Studies 5, 1941.

[2] Dietlinde Lau *On Closed Subsets of Boolean functions (a new proof for Post's Theorem).* J. Inform. Process. Cybern. EIK 27 (1991) 167-78.

[3] Pelletier, F. J. and Martin, N. M. *Post's Functional Completeness Theorem.* Notre Dame Journal of Formal Logic Volume 31, Number 3, 1990, pages 462-475.

[4] Martín Escardó *Exhaustible Sets in Higher-Type Computation.* Logical Methods in Computer Science Vol. 4 (3:3), 2008, pages 1-37.


# A Coq Definitions

The vector application operation, which takes a vector of functions of type $A \to B$ and an element $a \in A$ and outputs the vector of type $B$ which results from applying $a$ to each such function:

```
Fixpoint vec_ap (A B : Set) (n : nat) : vec (A -> B) n -> A -> vec B n :=
  match n as n return vec (A -> B) n -> A -> vec B n with
    0   => fun _ _ => (tt : vec B 0)
  | S m => fun fs a => ( ((vhead fs) a) , (vec_ap B m (vtail fs) a) )
  end.
```

The item at index $i$ of an $n$-length vector:

```
Fixpoint item_at (n : nat) : fin n -> vec n -> A :=
  match n as n return fin n -> vec n -> A with
  |0   => fun i v => emptyf i
  |S m => fun i v => match i with
```

```
                     |inl tt => vhead v
                     |inr j  => item_at m j (vtail v)
                     end
  end.
```

The connectives of arity $k$:

```
Definition conn (k : nat) :=
  vec bool k -> bool.
```

Notation for function extensionality:

```
Definition conn_ext_eq (n : nat) (f g : conn n) :=
  forall v , f v = g v.

Notation "f [=] g" := (conn_ext_eq f g) (at level 38, right associativity).
```

The generalized composition operation between an $n$-ary connective and an $n$-length vector of $k$-ary connectives:

```
Definition comp (n k : nat) (f : conn n) (gs : vec (conn k) n) : conn k :=
  fun xs => f (vec_ap bool n gs xs).
```

Given a boolean $x$, the constant $k$-ary connective that always outputs $x$:

```
Definition const (k : nat)(x : bool) : conn k :=
  fun v => x.
```

The projection functions $\pi_i^n$:

```
Definition proj (n : nat) (i : fin n) : conn n :=
  fun v : vec bool n => item_at bool n i v .
```