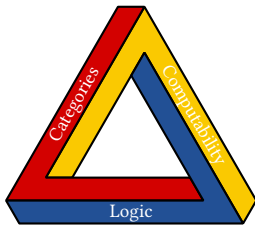


Exercise solutions for



# CATEGORICAL REALIZABILITY

MIDLANDS GRADUATE SCHOOL (MGS)

8–12 April 2024, Leicester, UK

JOSHUA CHEN

School of Computer Science  
University of Nottingham

April 2024

## Solutions to Chapter 2

**Exercise 2.11.** Each of the properties (i)–(iii) are proved by induction on the structure of the body term  $t$ .

- (i) The claim holds when  $t = x$ , when  $t = y$  is a variable distinct from  $x$ , and when  $t = a \in \mathcal{A}$ .

Finally, by the induction hypothesis the set of variables in the term

$$\langle x \rangle. (t_1 t_2) = S (\langle x \rangle. t_1) (\langle x \rangle. t_2)$$

is exactly  $(\mathcal{V}(t_1) \setminus x) \cup (\mathcal{V}(t_2) \setminus x)$ , where  $\mathcal{V}(t)$  denotes the set of variables in the term  $t$ . The result in this case now follows since

$$\mathcal{V}(t_1 t_2) \setminus x = (\mathcal{V}(t_1) \setminus x) \cup (\mathcal{V}(t_2) \setminus x).$$

- (ii) The term  $\langle x \rangle. t$  is defined when  $t$  is a variable or an element of  $\mathcal{A}$  since, by Definition 2.1,  $K a$  and  $S a b$  are defined for any  $a, b \in \mathcal{A}$ .

Finally, if  $\langle x \rangle. t_1$  and  $\langle x \rangle. t_2$  are defined then any substitution into the variables of

$$\langle x \rangle. (t_1 t_2) = S (\langle x \rangle. t_1) (\langle x \rangle. t_2)$$

yields an element of  $\mathcal{A}$  of the form  $S a b$  for some  $a, b \in \mathcal{A}$ .

- (iii) By straightforward computation in the case where  $t$  is not an application. When  $t = t_1 t_2$ ,

$$\begin{aligned} & (\langle x \rangle. (t_1 t_2)) a \\ &= S (\langle x \rangle. t_1) (\langle x \rangle. t_2) a \\ &\simeq ((\langle x \rangle. t_1) a) ((\langle x \rangle. t_2) a) \\ &\simeq (t_1[a/x]) (t_2[a/x]) \quad (\text{by the induction hypothesis}) \\ &\simeq (t_1 t_2)[a/x]. \end{aligned}$$

**Exercise 2.14.**

- (i)  $\text{pair } a b = (\langle xyz \rangle. zxy) a b = \langle z \rangle. z a b$  is defined by Exercise 2.11.  
(ii) By computation, taking care to note throughout that all applications are defined in  $\mathcal{A}$ .

**Exercise 2.15.** A possible set of definitions is

$$\begin{aligned} \text{iszero} &:= \text{fst} \\ \text{succ} &:= \text{pair false} \\ \text{pred} &:= \langle n \rangle. \text{if } (\text{iszero } n) \ \bar{0} \ (\text{snd } n) \end{aligned}$$

(check that these satisfy the required equations).

**Exercise 2.16.** From the specification of **primrec**, we would like our definition to satisfy the equation

$$\mathbf{primrec} \ a \ f \simeq \langle n \rangle. \left( \text{if } (\text{iszero } n) \ a \ (f \ (\text{pred } n) \ (\mathbf{primrec} \ a \ f \ (\text{pred } n))) \right)$$

for any  $a, f \in \mathcal{A}$ .

This suggests that the term  $\mathbf{primrec} \ a \ f$  should be constructed as a fixed point of the abstraction

$$\langle r \rangle. \left( \langle n \rangle. \left( \text{if } (\text{iszero } n) \ a \ (f \ (\text{pred } n) \ (r \ (\text{pred } n))) \right) \right), \quad (1)$$

and so we might try to define

$$\mathbf{spec}' := \langle af \rangle. \langle rn \rangle. \text{if } (\text{iszero } n) \ a \ (f \ (\text{pred } n) \ (r \ (\text{pred } n)))$$

and

$$\mathbf{primrec}' := \langle af \rangle. Z \ (\mathbf{spec}' \ a \ f).$$

However, this definition does not satisfy the requirement that  $\mathbf{primrec}' \ a \ f \ \bar{0}$  is always defined (expand the definition and check!).

Instead, we tweak the abstraction (1) whose fixed point we take, and define

$$\begin{aligned} \mathbf{spec} &:= \langle af \rangle. \langle rn \rangle. \text{if } (\text{iszero } n) \ (K \ a) \ (S \ f \ r) \ (\text{pred } n), \\ \mathbf{primrec} &:= \langle af \rangle. Z \ (\mathbf{spec} \ a \ f). \end{aligned}$$

We can then check (do so!) that the required equations are satisfied.

**Exercise 2.17.**

(i)  $\implies$  (ii): Assuming  $\mathbf{true} \neq \mathbf{false}$ , we show that  $\bar{m} \neq \bar{n}$  for all  $n \in \mathbb{N}$  and  $m < n$ , by case distinction on  $n$  and then induction on  $m < n$ . This is trivial for  $n = 0$ .

Assume that  $n = n' + 1$ . For  $m = 0$ ,

$$\text{iszero } \bar{m} = \mathbf{true} \neq \mathbf{false} = \text{iszero } \bar{n}$$

and so  $\bar{m} \neq \bar{n}$ . If  $m + 1 < n$  then  $m < n'$  and

$$\text{pred } \overline{m+1} = \bar{m} \neq \bar{n}' = \text{pred } \bar{n}$$

by induction, so  $\overline{m+1} \neq \bar{n}$ .

(ii)  $\implies$  (iii): Immediate.

(iii)  $\implies$  (i): Because if  $\mathbf{true} = \mathbf{false}$  then

$$a = \text{if } \mathbf{true} \ a \ b = \text{if } \mathbf{false} \ a \ b = b$$

for all  $a, b \in \mathcal{A}$ .

## Solutions to Chapter 3

**Exercise 3.20.** The inclusion maps

$$\text{inl}: X \rightarrow X + Y \quad \text{and} \quad \text{inr}: Y \rightarrow X + Y$$

are given by the usual coproduct inclusions in  $\text{Set}$ , and tracked by **left** and **right** respectively.

To check that the definition gives a coproduct in  $\text{Asm}_{\mathcal{A}}$ , it's enough to show that for any assembly  $Z$  and assembly maps  $f: X \rightarrow Z$  and  $g: Y \rightarrow Z$ , the induced function  $[f, g]: |X + Y| \rightarrow |Z|$  is tracked. That is, we need  $\tau_{[f, g]} \in \mathcal{A}$  such that for all  $x \in X$ ,  $y \in Y$  and realizers  $a \Vdash_X x$  and  $b \Vdash_Y y$ ,

$$\text{pair false } a \Vdash_{X+Y} \text{inl}(x) \implies \tau_{[f, g]}(\text{pair false } a) \Vdash_Z [f, g](\text{inl}(x)) = f(x)$$

and

$$\text{pair true } b \Vdash_{X+Y} \text{inr}(y) \implies \tau_{[f, g]}(\text{pair true } b) \Vdash_Z [f, g](\text{inr}(y)) = g(y).$$

Denoting the trackers of  $f$  and  $g$  by  $\tau_f$  and  $\tau_g$  respectively, we may define

$$\tau_{[f, g]} := \langle w \rangle. \text{if } (\text{fst } w) \tau_f \tau_g (\text{snd } w).$$

**Exercise 3.21.** The map out of the coproduct

$$\begin{aligned} 1 + 1 &\rightarrow 2 \\ \text{inl}(\star) &\mapsto 0 \\ \text{inr}(\star) &\mapsto 1 \end{aligned}$$

(induced by the constant maps  $1 \rightarrow 2$  at 0 and 1) is bijective on the carriers. Its inverse function is tracked by

$$\langle w \rangle. \text{pair } w \ a$$

for any  $a \in \mathcal{A}$ , and so we get a pair of inverse assembly isomorphisms.

**Exercise 3.23.**

- (i) The full details of this depend on the particular set theory, but the idea of the proof will hold for any “good” definition of  $\text{Set}$  (in particular, for ZFC) and is as follows.

The morphisms  $z: 1 \rightarrow \mathbb{N}$  and  $s: \mathbb{N} \rightarrow \mathbb{N}$  are given by the constant function at 0 and the successor function, respectively. Given any set  $X$  and functions  $x: 1 \rightarrow X$  and  $f: X \rightarrow X$ , recursively define a sequence of functions

$$(r_n: \{0, \dots, n\} \rightarrow X)_{n \in \mathbb{N}}$$

by

$$\begin{aligned} r_0(0) &:= x, \\ r_{n+1}(m) &:= \begin{cases} r_n(m) & \text{if } m \leq n \\ f(r_n(n)) & \text{if } m = n + 1 \end{cases} \end{aligned}$$

(where we have, as is customary, used the same name  $x$  to refer to the constant function  $1 \xrightarrow{x} X$  and its value  $x \in X$ ).

By induction,  $r_{n+1} \upharpoonright \{0, \dots, n\} = r_n$  for all  $n \in \mathbb{N}$ , and the universal morphism  $r: \mathbb{N} \rightarrow X$  is the union of this sequence of functions. In particular, we have that

$$r(n+1) = r_{n+1}(n+1) = f(r_n(n)) = f(r(n))$$

for all  $n \in \mathbb{N}$ .

- (ii) The zero and successor functions of  $|\mathbb{N}| = \mathbb{N}$  are tracked by  $\mathbf{K} \bar{0}$  and **succ** respectively. We claim that these are also the morphisms making  $\mathbb{N}$  an nno in  $\mathbf{Asm}_{\mathcal{A}}$ .

To show this, it's enough to show that given any assembly  $X$  and maps  $x: 1 \rightarrow X$  and  $f: X \rightarrow X$  such that  $a \Vdash_X x$  and  $\mathbf{t}_f$  tracks  $f$ , the function  $r: |\mathbb{N}| \rightarrow |X|$  defined in part (i) is tracked. And indeed it is: show, by induction, that

$$\mathbf{primrec} \ a \ (\mathbf{K} \ \mathbf{t}_f)$$

tracks  $r$ .

**Exercise 3.26.** Any constant function  $x: |1| \rightarrow |X|$  for any assembly  $X$  is tracked (by  $\mathbf{K} \ a$  for any realizer  $a \Vdash_X x$ ), so

$$\mathbf{Asm}_{\mathcal{A}}(1, X) \cong \mathbf{Set}(|1|, |X|) \cong |X|.$$

Naturality of this bijection is immediate because the action of any assembly map is just the action of its underlying function.

**Exercise 3.28.** For any assembly  $X$  and set  $Y$ , we have that any function  $|X| \rightarrow |\nabla(Y)|$  is tracked by  $\mathbf{I}$ , so

$$\mathbf{Set}(|X|, Y) = \mathbf{Set}(|X|, |\nabla(Y)|) \cong \mathbf{Asm}_{\mathcal{A}}(X, \nabla(Y)).$$

Again, naturality in the arguments holds straightforwardly (check!).

**Exercise 3.30.** If there is a nonconstant map  $\nabla\{0, 1\} \rightarrow 2$  with tracker  $\mathbf{t}$ , then for any  $a \in \mathcal{A}$  we have that  $a \Vdash_{\nabla\{0,1\}} 0$  and  $a \Vdash_{\nabla\{0,1\}} 1$ , and hence that  $\mathbf{t} \ a \Vdash_2 0$  and  $\mathbf{t} \ a \Vdash_2 1$ . But the realizers of 0 and 1 in 2 are **true** and **false** respectively, and so

$$\mathbf{true} = \mathbf{t} \ a = \mathbf{false}.$$

By Exercise 2.17, this means that  $\mathcal{A}$  is trivial.

**Exercise 3.31.** In particular, a right adjoint  $R: \mathbf{Asm}_{\mathcal{A}} \rightarrow \mathbf{Set}$  would satisfy

$$\mathbf{Asm}_{\mathcal{A}}(\nabla\{0, 1\}, 2) \cong \mathbf{Set}(\{0, 1\}, R(2)).$$

By Exercise 3.30 and the fact that constant functions are always tracked, the left hand side has size 2 when  $\mathcal{A}$  is nontrivial. On the other hand, the right hand set has size  $|R(2)|^2$ , and it cannot be the case that this is equal to 2.

**Exercise 3.32.**

- (i) We show that any two distinct assembly maps  $f, g: X \rightarrow 2$  must have distinct trackers. To this end assume that  $f(x) = 0$  and  $g(x) = 1$  for some  $x \in X$  with realizer  $a \Vdash_X x$ . If  $t \in \mathcal{A}$  tracks both  $f$  and  $g$ , then  $t a \Vdash_2 f(x)$  and also  $t a \Vdash_2 g(x)$ . Thus **true** = **false** and  $\mathcal{A}$  is trivial, which we assumed was not the case. Hence  $f$  and  $g$  have distinct trackers, and there can be at most  $|\mathcal{A}|$ -many.
- (ii) Because otherwise, by the universal property of coproducts there would be exactly  $2^{|\mathcal{A}|} > |\mathcal{A}|$  maps from  $\coprod_{a \in \mathcal{A}} \mathbf{1}$  to  $\mathbf{2}$ .
- (iii) Left adjoints preserve colimits, in particular coproducts. However, the  $\mathcal{A}$ -indexed coproduct of terminal objects exists in **Set** but not in  $\mathbf{Asm}_{\mathcal{A}}$ .