1 Monads and Kleisli categories

1.1 Monads

Let $\mathcal{C}$ be a category. Then the category $[\mathcal{C},\mathcal{C}]$ of functors $\mathcal{C} \to \mathcal{C}$ and natural transformations has a (strict) monoidal structure given by composition. A *monad* [Mac71, §VI] in $\mathcal{C}$ is a monoid in $[\mathcal{C},\mathcal{C}]$.

In other words, it is a functor $M\colon \mathcal{C} \to \mathcal{C}$ together with natural transformations $m_a\colon MMa \to Ma$ and $u_a\colon a \to Ma$ such that the following diagrams commute for all objects a of $\mathcal{C}$.

$$\begin{array}{ccc} MMMa & \xrightarrow{Mm_a} & MMa \\ \downarrow{\scriptstyle m_{Ma}} & & \downarrow{\scriptstyle m_a} \\ MMa & \xrightarrow{m_a} & Ma \end{array} \qquad \begin{array}{ccc} Ma & \xrightarrow{Mu_a} & MMa \\ & {\scriptstyle id}\searrow & \downarrow{\scriptstyle m_a} \\ & & Ma \end{array} \qquad \begin{array}{ccc} Ma & \xrightarrow{u_{Ma}} & MMa \\ & {\scriptstyle id}\searrow & \downarrow{\scriptstyle m_a} \\ & & Ma \end{array}$$

Example 1.1. In the category of sets, the *nonempty powerset functor* $\mathcal{P}_+$ sends a set A to the set of nonempty subsets of A. This has the structure of a monad on **Set**, since we have a natural transformation (union) from $\mathcal{P}_+\mathcal{P}_+A \to \mathcal{P}_+A$ and a natural transformation (singleton) from $A \to \mathcal{P}_+A$ that obey the diagrams given above.

Example 1.2. Let $\mathcal{M}$ be a monoidal category and let x be a monoid in $\mathcal{M}$. The *writer monad* W_x on $\mathcal{M}$ is defined by $W_x y = y \otimes x$, with natural transformations

$$m_y\colon y \otimes x \otimes x \to y \otimes x \qquad\qquad u_y\colon y \to y \otimes x$$

given by the monoid structure on x.

Going the other way, if $\mathcal{M}$ is monoidal closed with inner hom $\multimap$, and if z is a comonoid in $\mathcal{M}$, then the *reader monad* R_z is given by $R_z y = z \multimap y$. Then the monadic coherences

$$m_y\colon z \multimap z \multimap y \to z \multimap y \qquad\qquad u_y\colon y \to z \multimap y$$

are induced from the comonoid structure on z. This second example is particularly important in Cartesian closed categories, in which every object has the structure of a comonoid.

Example 1.3. If $\mathcal{C} \underset{R}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathcal{D}$ is an adjunction with counit $\epsilon\colon LR \to 1$ and unit $\eta\colon 1 \to RL$, then the composite $RL\colon \mathcal{C} \to \mathcal{C}$ has the structure of a monoid on $\mathcal{C}$, where the multiplication and unit are given by

$$R\epsilon L\colon RLRL \to RL \qquad\qquad \eta\colon 1 \to RL\,.$$

We will see in the next section that every monad is induced by an adjunction in this way.

As an example, if $\mathcal{M}$ is a monoidal closed category and w is an object of $\mathcal{M}$, then the *state monad* S_w on $\mathcal{M}$ is defined by

$$S_w x = w \multimap (x \otimes w)\,.$$

Example 1.4. Another example that arises from an adjunction is the *list monad* on **Set** that arises from the adjunction between the category of sets and the category of (set-valued) monoids. The underlying set of the free monoid on a set A is the set A^* of finite lists of elements of A, and the functor $A \mapsto A^*$ inherits a monoid structure where the multiplication $m_A \colon (A^*)^* \to A^*$ concatenates a list of lists into a single list and the unit $u_a \colon A \to A^*$ forms a list with a single element.

Example 1.5. A monad on $\mathcal{C}^{\mathrm{op}}$ is called a *comonad* on $\mathcal{C}$. The carrier of a comonad is still a functor $M \colon \mathcal{C} \to \mathcal{C}$, but now the multiplication and unit are natural transformations $M \Rightarrow MM$ and $M \Rightarrow 1$, rather than the other way round.

An adjunction $\mathcal{C} \underset{R}{\overset{L}{\underset{\longleftarrow}{\overset{\longrightarrow}{\perp}}}} \mathcal{D}$ gives rise to a comonad structure on LR in much the same way as it gives rise to a monad structure on RL. So, for example, we have the *store comonad* S'_r for any object r of a monoidal closed category $\mathcal{M}$, given by

$$S'_r x = (r \multimap x) \otimes x\,.$$

1.2 Kleisli Categories

Let $\mathcal{C}$ be a category and let M be a monad on $\mathcal{C}$. Then [Kle65] there is a category Kl_M, called the *Kleisli category* of M, whose objects are the objects of $\mathcal{C}$ and where a morphism from an object a to an object b is a morphism $a \to Mb$ in $\mathcal{C}$.

Identity arrows are given by the morphisms $u_c \colon c \to Mc$ (considered as a morphism $c \to c$ in Kl_M) and the composition of arrows $f \colon a \to Mb$ and $g \colon b \to Mc$ is given by the following composite in $\mathcal{C}$.

$$a \xrightarrow{f} Mb \xrightarrow{Mg} MMc \xrightarrow{m_c} Mc$$

There is a natural identity-on-objects functor $J \colon \mathcal{C} \to \mathrm{Kl}_M$ that sends a morphism $f \colon a \to b$ in $\mathcal{C}$ to the composite

$$a \xrightarrow{f} b \xrightarrow{u_b} Mb\,,$$

considered as a morphism $a \to b$ in Kl_M.

In the other direction, we have a functor $S \colon \mathrm{Kl}_M \to \mathcal{C}$ that sends an object a of Kl_M to the object Ma of $\mathcal{C}$ and sends a morphism $f \colon a \to Mb$ from a to b in Kl_M to the composite

$$Ma \xrightarrow{Mf} MMb \xrightarrow{m_b} Mb$$

in $\mathcal{C}$. Note that $SJ = M$, by one of our coherence conditions on m and u. Meanwhile, JS is the functor $\mathrm{Kl}_M \to \mathrm{Kl}_M$ that sends an object a to Ma and sends a morphism $f\colon a \to Mb$ from a to b to the morphism $Mf\colon Ma \to MMb$ from Ma to Mb.

Proposition 1.6 ([Kle65])**.** *S is a right adjoint to J. The unit of the adjunction is $u\colon \mathrm{id} \Rightarrow M$. The counit $e_a\colon J(Sa) \to a$ is given by the identity morphism $Ma \to Ma$ in $\mathcal{C}$, considered as a morphism $Ma \to a$ in Kl_M.*

Given a monad M on a category $\mathcal{C}$ and a functor $F\colon \mathcal{C} \to \mathcal{D}$, where $\mathcal{D}$ is another category, we say that a natural transormation $\psi_a\colon FMa \to Fa$ is *M-multiplicative* if it makes the following diagrams commute.

$$\begin{array}{ccc} FMMa & \xrightarrow{\psi_{Ma}} & FMa \\ {\scriptstyle Fm_a}\downarrow & & \downarrow{\scriptstyle \psi_a} \\ FMa & \xrightarrow{\psi_a} & Fa \end{array} \qquad\qquad \begin{array}{ccc} Fa & \xrightarrow{Fu_a} & FMa \\ & {\scriptstyle \mathrm{id}}\searrow & \downarrow{\scriptstyle \psi_a} \\ & & Fa \end{array}$$

Given two triples $(\mathcal{D}, F, \psi), (\mathcal{D}', F', \psi')$, where $F\colon \mathcal{C} \to \mathcal{D}, F'\colon \mathcal{C}' \to \mathcal{D}'$ are functors and $\psi\colon FM \Rightarrow F, \psi'\colon F'M \Rightarrow F'$ are functors, we define a *morphism* from $(\mathcal{D}', F', \psi')$ to $(\mathcal{D}, F, \psi)$ to be a functor $H\colon \mathcal{D}' \to \mathcal{D}$ such that $F = HF'$ and $\psi = H\psi'$. This gives us a category.

A defining property of the Kleisli category is that it is initial among such triples $(\mathcal{D}, F, \psi)$:

Proposition 1.7 ([Str72])**.** *i) Given an object a of $\mathcal{C}$, the identity morphism $Ma \to Ma$ may be considered as a morphism $\phi_a\colon JMa \to Ja$ in Kl_M. ϕ_a is an M-multiplicative natural transformation.*

ii) Let $\mathcal{D}$ be a category, let $F\colon \mathcal{C} \to \mathcal{D}$ be a functor and suppose that $\psi_a\colon FMa \to Ma$ is an M-multiplicative natural transformation. Then there is a unique functor $\hat{F}\colon \mathrm{Kl}_M \to \mathcal{D}$ such that $F = \hat{F}J$ and $\psi = \hat{F}\phi$.

Another way to characterize the Kleisli category Kl_M is to say that the the adjunction we described above is initial among all adjunctions giving rise to the monad M. This can be deduced from Proposition 1.7 using the following result.

Lemma 1.8 ([Str72])**.** *Let $\mathcal{C}$ be a category and let M be a monad on $\mathcal{C}$. If $\mathcal{C} \underset{R}{\overset{L}{\rightleftarrows}} \mathcal{D}$ is an adjunction (with counit ϵ and unit η), we say it* gives rise to *M if $M = RL$, $m = R\epsilon L$ and $u = \eta$.*

Any such adjunction gives rise to an M-multiplicative natural transformation $\psi\colon LM \Rightarrow L$. This gives us a fully faithful functor from the category of adjunctions giving rise to M to the category of triples $(\mathcal{D}, F, \psi)$ where ψ is M-multiplicative.

The proof of Proposition 1.7 essentially comes down to the following factorization result. If $f\colon a \to b$ is a morphism in Kl_M, then f may be factorized as

$$f = a \xrightarrow{Jf} Mb \xrightarrow{\phi_b} b\,,$$

where we use 'f' to refer both to the morphism $a \to b$ in Kl_M and to the underlying morphism $a \to Mb$ in $\mathcal{C}$. Indeed, if we compute this composite inside $\mathcal{C}$, we get

$$a \xrightarrow{f} Mb \xrightarrow{u_{Mb}} MMb \xrightarrow{M\,\mathrm{id}} MMb \xrightarrow{m_b} Mb\,,$$

which is equal to f by the coherence conditions on m and u. This means that the Kleisli category may be thought of as begin freely generated from the original category $\mathcal{C}$ and a multiplicative natural transformation ϕ.

Example 1.9. The morphisms in the Kleisli category for the nonempty powerset monad $\mathcal{P}_+$ on **Set** are functions $A \to \mathcal{P}_+B$, which can be thought of as nondeterministic functions. Given a set A, the morphism $\phi_A\colon \mathcal{P}_+A \to A$ in $\mathrm{Kl}_{\mathcal{P}_+}$ can be interpreted as a 'nondeterministic choice' function that accepts a nonempty set of elements of A and nondeterministically chooses one of them. The factorization then means that the category is freely generated over $\mathcal{C}$ by these nondeterministic choice morphisms.

Example 1.10. Let $\mathcal{C}$ be a Cartesian closed category and let z be some fixed object of $\mathcal{C}$. Then the Kleisli category for the reader monad R_z on $\mathcal{C}$ is generated over $\mathcal{C}$ by a natural transformation $\phi_y\colon (z \to y) \to y$. By the enriched Yoneda lemma, such a natural transformation is always given by precomposition with some fixed morphism $\mathsf{ask}\colon 1 \to z$. This means that Kl_{R_z} is suitable for modelling any situation in which we are generally working in $\mathcal{C}$, but need the ability to request a value of type z (for example, a config file, a piece of user input or something else that isn't being passed into the function in question).

A particularly important fact about the reader monad in Cartesian closed categories is the following.

Theorem 1.11 ([Lam74])**.** *Let $\mathcal{C}$ be a Cartesian closed category and let z be an object of $\mathcal{C}$. Then the Kleisli category Kl_{R_z} for the reader monad over z on $\mathcal{C}$ is Cartesian closed.*

The *functional completeness* theorem [Lam74] can be thought of as a special case of our remarks above.

1.3 Extended example: game semantics of time complexity

This is an extended example of using a reader monad, in the style of Example 1.10. Let $\mathcal{G}$ be the category of games and visible single-threaded strategies as in [AM96]. So $\mathcal{G}$ is a sound and adequate model of Idealized Algol satisfying compact definability. Let $\mathbb{C}$ be the denotation of the command type `com` of

Idealized Algol. Then the Kleisli category for the reader monad over $\mathbb{C}$ is freely generated over $\mathcal{G}$ by a single morphism $1 \to \mathbb{C}$.

Let us extend Idealized Algol with a single term, sleep, of type $\mathtt{com}$. This should be thought of as a command to the computer to pause for some fixed interval of time – let us say a second – before resuming execution. By Theorem 1.11, our Kleisli category is Cartesian closed. So it gives us a denotational semantics of IA+sleep, in which the terms of IA are interpreted within $\mathcal{G}$ as in [AM96] and the new term sleep is interpreted by the new morphism $1 \to \mathbb{C}$.

Let us now give an operational semantics to terms of IA+sleep. Let Γ be an IA-context. Then a *store* s over Γ is a list of pairs $(v \mapsto x)$, where each v is the name for a $\mathtt{Var}$-type variable occurring in Γ and x is a value held in v. We write $()$ for the empty store, $(s|v \mapsto x)$ for the store obtained from s by adding or updating the variable v to hold the value x and $(s|v \mapsto \bot)$ for the store obtained by removing the variable v from s. Given disjoint stores s, t (i.e., stores whose sets of variables are disjoint), we write $s + t$ for the store obtained by concatenating together s and t.

Given a term $\Gamma \vdash M$ in context, stores s,s' over Γ, a value c and a natural number n, we define a big-step relation

$$\Gamma, s \vdash M \Downarrow_n c, s' \,,$$

read as 'in the context Γ, s, M converges to c, s' in time n'. The rule for the constant sleep is given by

$$\frac{}{\Gamma, s \vdash \mathsf{sleep} \Downarrow_1 \mathsf{skip}, s} \quad ,$$

indicating that execution of the sleep command takes up an additional time step. The other rules are the same as those given in [AM96] for the $\Downarrow$ relation, except that we have to modify them in order to take account of the new information we have added to the reduction relation (i.e., the time taken to converge). For example:

$$\frac{}{\Gamma, s \vdash c \Downarrow_0 c, s} \qquad \frac{\Gamma, s \vdash M \Downarrow_m \mathsf{skip}, s' \qquad \Gamma, s' \vdash N \Downarrow_n \mathsf{skip}, s''}{\Gamma, s \vdash M; N \Downarrow_{m+n} \mathsf{skip}, s''} \quad .$$

If P is a program (i.e., a term of type $\mathtt{com}$) then we write $P \Downarrow_n$ as a shorthand for $, () \vdash P \Downarrow_n \mathsf{skip}, ()$. We say that P *diverges* and write $P \Uparrow$ if there is no n such that $P \Downarrow_n$.

We now want a way to capture this operational behaviour within the denotational semantics. First, we note the Computational Adequacy result proved for the game semantics of Idealized Algol.

Proposition 1.12 ([AM96, 21])**.** *Let P be a term of type $\mathtt{com}$. Then $P \Downarrow$ if and only if $[\![P]\!] \neq \bot$.*

To make sense of this strategy, we need to know that $\mathcal{G}$ is enriched in algebraic cpos. Then $\perp$ is the bottom element of $\mathcal{G}(1, \mathbb{C})$.

Before proceeding further, we prove an important lemma. This lemma tells us that the operational semantics of a term of IA+sleep may be modelled within Idealized Algol; i.e., that IA+sleep is a conservative extension of the original language.

Lemma 1.13. *Let $P : T$ be a term of IA+*sleep*. Suppose that*

$$\Gamma, s \vdash P \Downarrow_n c, s'$$

*in IA+*sleep*.*

Then for all $k \geq 0$, all contexts Δ disjoint from Γ, all stores t over Γ, Δ disjoint from s and all variables v not in Γ, Δ:

$$\Delta, \Gamma, v, (t{+}s|v \mapsto k) \vdash P[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] \Downarrow c, (t{+}s|v \mapsto k + n)$$

in Idealized Algol.

Proof. Induction on the derivation that $\Gamma, s \vdash P \Downarrow_n c, s'$. There are several possibilities for the last step in the derivation.

Structural congruence Suppose that the last step in the derivation is

$$\frac{P \equiv P' \qquad \Gamma, s \vdash P' \Downarrow_n \mathsf{skip}, s'}{\Gamma, s \vdash P \Downarrow_n \mathsf{skip}, s'} \ .$$

It is clear then by the definition of structural congruence that $P'[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] \equiv P[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}]$. So the claim follows by the inductive hypothesis applied to P' and the structural congruence rule for Idealized Algol.

sleep Suppose that the last step in the derivation is

$$\frac{}{\Gamma, s \vdash \mathsf{sleep} \Downarrow_1 \mathsf{skip}, s} \ .$$

Fix a context Δ disjoint from Γ and a store t over Γ, Δ that is disjoint from s. Fix $k \geq n$. Then it suffices to show that the following judgement may be proved within the operational semantics of Idealized Algol.

$$\Delta, \Gamma, v, (t{+}s|v \mapsto k) \vdash v \leftarrow \mathsf{succ}\ !v \Downarrow \mathsf{skip}, (t{+}s|v \mapsto k + 1)$$

A derivation for this obvious fact is given in Figure 1a.

Canonical forms Suppose that the last step in the derivation is

$$\frac{}{\Gamma, s \vdash_0 c \Downarrow c, s} \ .$$

Then, since sleep is not a canonical form, we have $c[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] = c$. Then we have the derivation

$$\frac{}{\Delta, \Gamma, (t{+}s|v \mapsto k) \vdash c \Downarrow c, (t{+}s|v \mapsto k)} \ .$$

$$\dfrac{\dfrac{\dfrac{}{\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash v \Downarrow v, (t{+}s | v \mapsto k)}}{\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash !v \Downarrow k, (t{+}s | v \mapsto k)} \quad \dfrac{}{\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash v \Downarrow v, (t{+}s | v \mapsto k)}}{\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash v \leftarrow \mathsf{succ}\ !v \Downarrow \mathsf{skip}, (t{+}s | v \mapsto k+1)}$$

(a) The crucial part of Lemma 1.13 pertaining to the sleep constant is a routine Idealized Algol derivation.

$$\dfrac{\begin{array}{c}\Delta, \Gamma_1, v, (t{+}s | v \mapsto k) \vdash M_1[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] \Downarrow c_1, (t{+}s | v \mapsto k + n_1) \\ \cdots \qquad \Delta, \Gamma_p, v, (t{+}s | v \mapsto k + n_1 + \cdots + n_{p-1}) \vdash M_p[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] \Downarrow c_1, (t{+}s | v \mapsto k + n_1 + \cdots + n_p)\end{array}}{\Delta, \Gamma, v, (t{+}s^{(0)} | v \mapsto k + n_1 + \cdots + n_p) \vdash M[v \leftarrow \mathsf{succ}\ !v/\mathsf{sleep}] \Downarrow c, (t{+}s^{(p)} | v \mapsto k)}$$

(b) The big-step rules of IA+sleep give rise to big-step rules of Idealized Algol via our translation.

Figure 1: The proof of Lemma 1.13 works by translating big-step derivations from IA+sleep into big-step derivations of Idealized Algol.

Sequencing Suppose instead that the last step in the derivation is

$$\frac{\Gamma, s \vdash M \Downarrow_m \mathsf{skip}, s' \qquad \Gamma, s' \vdash N \Downarrow_n \mathsf{skip}, s''}{\Gamma, s \vdash M; N \Downarrow_{m+n} \mathsf{skip}, s''} \quad .$$

Fix $k \geq 0$. By induction on M, N, we have

$$\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash M[v \leftarrow \mathtt{pred}\ !v/\mathsf{sleep}] \Downarrow \mathsf{skip}, (t{+}s' | v \mapsto k+m)\,;$$

$$\Delta, \Gamma, v, (t{+}s' | v \mapsto k+m) \vdash N[v \leftarrow \mathtt{pred}\ !v/\mathsf{sleep}] \Downarrow \mathsf{skip}, (t{+}s'' | v \mapsto k+m+n)\,.$$

It follows that

$$\Delta, \Gamma, v, (t{+}s | v \mapsto k) \vdash (M; N)[v \leftarrow \mathtt{pred}\ !v/\mathsf{sleep}] \Downarrow \mathsf{skip}, (t{+}s'' | v \mapsto k+m+n)\,,$$

by the sequencing rule of Idealized Algol.

Remaining rules Having given the rule for canonical forms and sequencing explicitly, we now give a general technique by which we can deal with all the remaining rules.

Each rule is of the form

$$\frac{\Gamma_1, s^{(0)} \vdash z_1 \Downarrow_{n_1} \gamma_1, s^{(1)} \qquad \cdots \qquad \Gamma_p, s^{(p-1)} \vdash z_p \Downarrow_{n_p} \gamma_p, s^{(p)}}{\Gamma, s^{(0)} \vdash z \Downarrow_{n_1+\cdots+n_p} \gamma, s^{(p)}}$$

(for $p \in \{0, 1, 2, 3\}$), where

$$\frac{\Gamma_1, s^{(0)} \vdash z_1 \Downarrow \gamma_p, s^{(1)} \qquad \cdots \qquad \Gamma_p, s^{(p-1)} \vdash z_p \Downarrow \gamma_1, s^{(p)}}{\Gamma, s^{(0)} \vdash z \Downarrow \gamma, s^{(p)}}$$

is a rule for Idealized Algol.

Here, each z_i, z, γ_i, γ is a term of Idealized Algol (without sleep) that may have free variables that are not included in the Γ_i, Γ. These free variables (x_α) may be shared between the z_i, z, γ_i, γ, introducing a dependence between them. For example, in the rule for sequencing, the terms on the top are single-variable terms of IA, where 'M' and 'N' are themselves free variables. These variables are bound in the expression '$M; N$' on the bottom of the rule.

Fix some such rule, and suppose that it is used in the last step of the deduction. So the last step looks like this.

$$\frac{\Gamma_1, s^{(0)} \vdash M_1 \Downarrow_{n_1} c_1, s^{(1)} \qquad \cdots \qquad \Gamma_p, s^{(p-1)} \vdash M_p \Downarrow_{n_p} c_p, s^{(p)}}{\Gamma, s^{(0)} \vdash M \Downarrow_{n_1+\cdots+n_p} c, s^{(p)}}$$

Here, the M_i, M, c_i, c fit the pattern given by the z_i, z, γ_i, γ. By induction, for each $i = 1, \cdots, p$, the following judgement may be proved in Idealized Algol.

$$\Delta, \Gamma, v, (t{+}s^{(i)} | v \mapsto k + n_1 + \cdots + n_{i-1}) \\ \vdash M[v \leftarrow \mathtt{succ}\ !v] \Downarrow c_i, (t{+}s^{(i)} | v \mapsto k + n_1 + \cdots + n_i)$$

Now we know that the M_i, M, c_i, c may be obtained from the z_i, z, γ_i, γ by substituting in actual terms N_α for the free variables x_α in these expressions. Moreover, since the primitive sleep is not mentioned anywhere in these rules, it follows that we have

$$\begin{aligned} M_i[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] &= (z_i[N\alpha/x_\alpha])[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \\ &= z_i[(N[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}])/x_\alpha]\,, \end{aligned}$$

and similarly for the z, γ_i, γ. Therefore, the M_i, M, c_i, c still fit the pattern given by the z_i, z, γ_i, γ, even after making our substitution. It follows that we may derive

$$\Delta, \Gamma, v, (t{+}s^{(0)} | v \mapsto k) \\ \vdash M[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \Downarrow c, (t{+}s^{(p)} | v \mapsto k + n_1 + \cdots + n_p)$$

using the derivation shown in Figure 1b.

This completes the induction. □

In light of this result, we can make a definition relating the denotational and operational semantics of terms of type `nat`. First, observe that since the β rule is valid in our semantics, we may write the denotation of a term $M\colon \mathtt{com}$ of IA+sleep as the composite

$$1 \xrightarrow{\mathsf{sleep}} \mathbb{C} \xrightarrow{[\![c \vdash M[c/\mathsf{sleep}]]\!]} \mathbb{C}$$

in $\mathrm{Kl}_{R_\mathbb{C}}$.

Working in the original category $\mathcal{G}$, however, we may identify this composite with the morphism

$$[\![\lambda c.M[c/\mathsf{sleep}]]\!] : 1 \to (\mathbb{C} \to \mathbb{C})\,.$$

Now we have a morphism

$$\eta = [\![f\colon \mathtt{com} \to \mathtt{com} \vdash \lambda v.f(v \leftarrow \mathtt{succ}\ !v); !v]\!] : (\mathbb{C} \to \mathbb{C}) \to (\mathtt{Var} \to \mathbb{N})\,.$$

If we form the composite

$$1 \xrightarrow{[\![\lambda c.M[c/\mathsf{sleep}]]\!]} (\mathbb{C} \to \mathbb{C}) \xrightarrow{\eta} (\mathtt{Var} \to \mathbb{N}) \xrightarrow{\mathtt{new}_\mathbb{N}} \mathbb{N}$$

then the morphism we get will be the denotation in $\mathcal{G}$ of the term

$$\vdash \mathtt{new}\ v = 0 \text{ in } M[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]; !v : \mathtt{nat}$$

(since the β rule is valid for the semantics in $\mathcal{G}$). Lemma 1.13 tells us that if $M \Downarrow_n \mathsf{skip}$ then this term converges to n in Idealized Algol.

Before we introduce our adequacy result, we need one more definition in order to take account of the fact that Proposition 1.12 applies to terms of type `com`, not `nat`.

Definition 1.14. Define terms $\mathtt{test}_n : \mathtt{nat} \to \mathtt{com}$ inductively by

$$\mathtt{test}_0 = \lambda m.\mathtt{If0}\ m \text{ then } \mathsf{skip} \text{ else } \Omega\,;$$
$$\mathtt{test}_{n+1} = \lambda m.\mathtt{If0}\ m \text{ then } \Omega \text{ else } \mathtt{test}_n\,.(\mathtt{pred}\ m)$$

So $\mathtt{test}_n$ converges if its input evaluates to n and diverges otherwise.

Let $t_n : \mathbb{N} \to \mathbb{C}$ be the denotation in $\mathcal{G}$ of $\mathtt{test}_n$.
Remark 1.15. In the model $\mathcal{G}$ of Idealized Algol, every morphism $1 \to \mathbb{N}$ is definable, so if $\sigma : 1 \to \mathbb{N}$ is such a morphism, then there must be at most one $n \in \omega$ such that the composite $\sigma; t_n$ is not equal to $\perp$.

Definition 1.16. Let $\sigma : 1 \to \mathbb{C}$ be a Kleisli morphism in $\mathrm{Kl}_{R_\mathbb{C}}$, considered as a morphism $1 \to (\mathbb{C} \to \mathbb{C})$ in $\mathcal{G}$.

If there is some (necessarily unique) $n \in \omega$ such that the composite

$$1 \xrightarrow{\sigma} (\mathbb{C} \to \mathbb{C}) \xrightarrow{\eta} (\mathtt{Var} \to \mathbb{N}) \xrightarrow{\mathtt{new}_\mathbb{N}} \mathbb{N} \xrightarrow{t_n} \mathbb{C}$$

is not equal to $\perp$, then we say that n is the *time taken for σ to converge*, or $\mathrm{tt}(\sigma)$. If there is no such n, we say that $\mathrm{tt}(\sigma) = \infty$.

From our discussion above, we have proved the following result.

Proposition 1.17 (Soundness for IA+sleep)**.** *Let $P : \mathtt{com}$ be a term of IA+*sleep *and let $\sigma : \mathbb{C} \to \mathbb{C}$ be its denotation in $\mathrm{Kl}_{R_\mathbb{C}}$, considered as a morphism in $\mathcal{G}$. If $P \Downarrow_n$ then $\mathrm{tt}(\sigma) = n$.*

In order to prove the other direction (adequacy), we need to prove a kind of converse to Lemma 1.13.

Lemma 1.18. *Let $\Gamma \vdash M : \mathtt{com}$ be a term of IA+*sleep *in context. Let s, s' be stores, let $a, b \in \mathbb{N}$, let c be a canonical form and let v be a variable not included in Γ. Suppose that*

$$\Gamma, v, (s|v \mapsto a) \vdash M \Downarrow c, (s'|v \mapsto b)$$

in Idealized Algol. Then there exists n such that

$$\Gamma, s \vdash M \Downarrow_n c, s'$$

*in IA+*sleep*.*

Proof. Induction on the derivation that $\Gamma, v, (s|v \mapsto a) \vdash M \Downarrow c, (s'|v \mapsto b)$. Most cases are self-explanatory. For example, suppose that the last step in the derivation is an instance of the sequencing rule:

$$\frac{\Gamma, v, (s|v \mapsto a) \vdash M[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \Downarrow \mathsf{skip}, s' \quad \Gamma, v, s' \vdash N[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \Downarrow c, (s''|v \mapsto b)}{\Gamma, v(s|v \mapsto a) \vdash (M; N)[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \Downarrow c, (s''|v \mapsto b)}\,.$$

Since there is no provable sequent of Idealized Algol in which a defined variable on the left ceases to be defined on the right, we may deduce that s' can be written as $(t|v \mapsto d)$. Therefore, we may apply the inductive hypothesis to M and N, which tells us that there must be m, n such that

$$\Gamma, s \vdash M \Downarrow_m \mathsf{skip}, t\,; \qquad\qquad \Gamma, t \vdash M \Downarrow_n c, s''\,.$$

From this it follows that

$$\Gamma, s \vdash M; N \Downarrow_{m+n} c, s''\,.$$

Crucial to this argument is the fact that

$$(M[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]); (N[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]) = (M; N)[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]\,.$$

For almost all the other rules, a similar equality holds. The only case where there is a problem is the first rule for $\_ \leftarrow \_$, since this may introduce a new copy of $v \leftarrow \mathtt{succ}\ !v$ on the bottom that does not arise from copies of $v \leftarrow \mathtt{succ}\ !v$ on the top.

The offending term occurs when $M = \mathsf{sleep}$ and so $M[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] = v \leftarrow \mathtt{succ}\ !v$. In that case, the inference is

$$\frac{\Gamma, (s|v \mapsto a) \vdash \mathtt{succ}\ !v \Downarrow b, s' \qquad \Gamma, s' \vdash v \Downarrow v, s''}{\Gamma, (s|v \mapsto a) \vdash v \leftarrow \mathtt{succ}\ !v \Downarrow \mathsf{skip}, (s''|v \mapsto b)}\,.$$

By following the derivation upwards, we may deduce that in fact $s = s' = s''$. But in that case the derivation of $\Gamma, s \vdash_n M \Downarrow \mathsf{skip}, s''$ is one of the base rules:

$$\frac{}{\Gamma, s \vdash \mathsf{sleep} \Downarrow_1 \mathsf{skip}, s}\,.$$

The other cases are routine. This completes the induction. □

Theorem 1.19 (Computational adequacy for IA+sleep)**.** *Let* $P\colon \mathtt{com}$ *be a term of IA+*sleep *and let* $\sigma\colon \mathbb{C} \to \mathbb{C}$ *be its denotation in* $\mathrm{Kl}_{R_{\mathbb{C}}}$*, considered as a morphism in* $\mathcal{G}$*. Then* $P \Downarrow_n$ *if and only if* $\mathrm{tt}(\sigma) = n$*.*

Proof. The forward direction is Proposition 1.17. For the reverse direction, note that if $\mathrm{tt}(\sigma) = n$, it means that

$$[\![\mathtt{test}_n(\mathtt{new} v = 0 \text{ in } P[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]; !v])]\!] \neq \bot\,,$$

which, by Proposition 1.12, means that

$$\mathtt{test}_n(\mathtt{new}v = 0 \text{ in } P[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}]; !v]) \Downarrow$$

in Idealized Algol, and therefore that

$$v, (v \mapsto 0) \vdash P[v \leftarrow \mathtt{succ}\ !v/\mathsf{sleep}] \Downarrow \mathsf{skip}, (v \mapsto n)\,.$$

From Lemma 1.3, we deduce that we must have

$$, () \vdash P \Downarrow_m \mathsf{skip}, ()$$

in IA+sleep for some m, and by Lemma 1.13 we must have $m = n$. Therefore, $P \Downarrow_n$. □

The last definition we need to make is the standard intrinsic equivalence relation.

Definition 1.20. Let $\sigma, \tau \colon A \to B$ be morphisms in $\mathrm{Kl}_{R_\mathbb{C}}$. By currying, we may consider σ, τ to be morphisms $1 \to (A \to B)$. We say that $\sigma \sim \tau$ if for all morphisms $\alpha \colon (A \to B) \to \mathbb{C}$, we have $\mathrm{tt}(\sigma; \alpha) = \mathrm{tt}(\sigma; \beta)$.

We now have all the tools necessary to prove that $Kl_{R_\mathbb{C}}$ is fully abstract for IA+sleep. The technique we use is standard and relies on a factorization result, as with the proofs given in [AM96] and [HM99]. The difference here is that our factorization result is immediate, and does not rely on any combinatorial arguments.

Theorem 1.21. *Let $M, N \colon T$ be two terms of IA+sleep. Then M and N are observationally equivalent if and only if $[\![M]\!] \sim [\![N]\!]$.*

Proof. First suppose that M and N are not observationally equivalent. So there is some context $- \colon T \vdash C[-] \colon \mathtt{com}$ such that $C[M]$ and $C[N]$ have different operational behaviours in IA+sleep. By Theorem 1.19 we know that for any program P we have $P \Downarrow_n$ if and only if $\mathrm{tt}([\![P]\!]) = n$ and $P \Uparrow$ if and only if $\mathrm{tt}([\![P]\!]) = \infty$. Therefore, we must have $\mathrm{tt}([\![C[M]]\!]) \neq \mathrm{tt}([\![C[n]]\!])$. But since $[\![C[M]]\!] = [\![M]\!]\,;[\![C]\!]$ and $[\![C[N]]\!] = [\![N]\!]\,;[\![C]\!]$, we must have $[\![M]\!] \not\sim [\![N]\!]$.

Conversely, suppose that $[\![M]\!] \not\sim [\![N]\!]$. So there is some $\alpha \colon [\![T]\!] \to \mathbb{C}$ such that $[\![M]\!]\,;\alpha \neq [\![N]\!]\,;\alpha$.

Let us now work in the category $\mathcal{G}$. So $[\![M]\!], [\![N]\!]$ may be regarded as morphisms $1 \to (\mathbb{C} \to [\![T]\!])$ and α may be regarded as a morphism $[\![T]\!] \to (\mathbb{C} \to \mathbb{C})$. Then the composites $[\![M]\!]\,;\alpha$ and $[\![N]\!]\,;\alpha$ within $\mathrm{Kl}_{R_\mathbb{C}}$ are given by the following composites in $\mathcal{G}$.

$$1 \xrightarrow{[\![M]\!]} (\mathbb{C} \to [\![T]\!]) \xrightarrow{\mathbb{C} \to \alpha} (\mathbb{C} \to (\mathbb{C} \to \mathbb{C})) \xrightarrow{\mu} (\mathbb{C} \to \mathbb{C})$$

$$1 \xrightarrow{[\![N]\!]} (\mathbb{C} \to [\![T]\!]) \xrightarrow{\mathbb{C} \to \alpha} (\mathbb{C} \to (\mathbb{C} \to \mathbb{C})) \xrightarrow{\mu} (\mathbb{C} \to \mathbb{C})$$

Without loss of generality, suppose that there exists some n such that

$$\mathrm{tt}([\![M]\!]\,;(\mathbb{C}\to\alpha);\mu)=n \qquad\qquad \mathrm{tt}([\![N]\!]\,;(\mathbb{C}\to\alpha);\mu)\neq n\,.$$

This means that we must have

$$[\![M]\!]\,;(\mathbb{C}\to\alpha);\mu;\eta;\mathtt{new}_{\mathbb{C}};t_n\neq\bot \qquad\qquad [\![N]\!]\,;(\mathbb{C}\to\alpha);\mu;\eta;\mathtt{new}_{\mathbb{C}};t_n=\bot$$

in $\mathcal{G}$ (where η is as defined in Definition 1.16).

$\mathcal{G}$ is enriched in algebraic cpos, so α is the least upper bound of its compact approximants. It follows that there is some compact $\alpha'\subseteq\alpha$ such that

$$[\![M]\!]\,;(\mathbb{C}\to\alpha');\mu;\eta;\mathtt{new}_{\mathbb{C}};t_n\neq\bot \qquad\qquad [\![N]\!]\,;(\mathbb{C}\to\alpha');\mu;\eta;\mathtt{new}_{\mathbb{C}};t_n=\bot\,.$$

In other words:

$$\mathrm{tt}([\![M]\!]\,;(\mathbb{C}\to\alpha');\mu)=n\,; \qquad\qquad \mathrm{tt}([\![M]\!]\,;(\mathbb{C}\to\alpha');\mu)\neq n\,.$$

Now the compact definability result for Idealized Algol [AM96, 17] tells us that α' must be the denotation of some IA term $x\colon T\vdash C[x]\colon \mathtt{com}\to\mathtt{com}$, which is therefore the denotation of $C[\mathsf{sleep}]\colon \mathtt{com}\to\mathtt{com}$ in $\mathrm{Kl}_{R_{\mathbb{C}}}$. So we get

$$\mathrm{tt}([\![C[\mathsf{sleep}]M]\!])=n\,; \qquad\qquad \mathrm{tt}([\![C[\mathsf{sleep}]N]\!])\neq n\,.$$

By Theorem 1.19, $C[\mathsf{sleep}]M\Downarrow_n$ and $C[\mathsf{sleep}]N\not\Downarrow_n$. Therefore, M and N are observationally inequivalent in IA+sleep. □

References

[AM96] Samson Abramsky and Guy McCusker. Linearity, sharing and state: a fully abstract game semantics for idealized algol with active expressions: Extended abstract. *Electronic Notes in Theoretical Computer Science*, 3:2 – 14, 1996. Linear Logic 96 Tokyo Meeting.

[HM99] R. Harmer and G. McCusker. A fully abstract game semantics for finite nondeterminism. In *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*, pages 422–430, 1999.

[Kle65] H. Kleisli. Every standard construction is induced by a pair of adjoint functors. *Proceedings of the American Mathematical Society*, 16(3):544–546, 1965.

[Lam74] J. Lambek. Functional completeness of cartesian categories. *Annals of Mathematical Logic*, 6(3):259 – 292, 1974.

[Mac71] Saunders MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

[Str72] Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149 – 168, 1972.