Monade

Contents

Videli smo, da je operacijska semantika računskih učinkov precej raznolika. Pri denotacijski semantiki ni nič drugače. Recimo, da tip A predstavimo z množico (ali domeno, čeprav zaradi enostavnosti delajmo z množicami) vrednosti $[\![A]\!]$. Programe, ki računajo vrednosti tipa A smo do sedaj predstavljali s funkcijami, ki so slikale v $[\![A]\!]$. A če dodamo učinke, taka predstavitev ni več dovolj dobra.

- Če želimo v programu podpirati izjeme, bo njegova interpretacija morala razločevati programe, ki vračajo vrednosti iz A od tistih, ki sprožajo izjeme. Tako bi bila ustreznejša kodomena funkcij $[\![A]\!] + \mathbb{E}$.
- Če program izpisuje znake iz množice O, moramo poleg rezultata vrniti še seznam izpisanih znakov, torej bi bila kodomena $[\![A]\!] \times O^*$.
- Če naš program lahko bere in piše po pomnilniku z množico možnih stanj S, bo interpretacija programa odvisna od začetnega stanja, hkrati pa bo vrnila novo, spremenjeno stanje. Boljša interpretacija bi bila $([\![A]\!] \times S)^S$.
- Če je izvajanje programa nedeterministično in gre lahko po različnih poteh, bi bilo bolje, če bi rezultat predstavili z množico možnih vrnjenih vrednosti, torej kodomeno $\mathcal{P}([\![A]\!])$.

Opazimo, da so vsi zgornji primeri kodomen oblike $T[\![A]\!]$, kjer je T neka konstrukcija, ki množico X slika v množico TX. Poleg tega za interpretacijo programov, ki povzročajo učinke, potrebujemo še dve operaciji:

- Če imamo vrednost tipa A, moramo znati predstaviti tudi program, ki vrača samo to vrednost in ne sproža nobenih učinkov. Torej potrebujemo operacijo, ki $[\![A]\!]$ vloži v $T[\![A]\!]$.
- Če imamo program, ki vrača vrednosti tipa A, in še en program, ki vrača vrednosti tipa B, ju želimo izvesti enega za drugim. Pri tem je drugi program lahko odvisen od rezultata prvega. Torej potrebujemo operacijo, ki zna iz interpretacije v $T[\![A]\!]$ in funkcije $[\![A]\!] \to T[\![B]\!]$ sestaviti interpretacijo združenega programa v $T[\![B]\!]$.

Definicija monad

Tako strukturo opisujejo *monade*. Monada je podana s trojico $(T, \eta, \gg\!=)$, kjer je:

- T preslikava (funktor), ki vsaki množici X priredi množico TX,
- *enoto* η, ki je družina preslikav (naravna transformacija) $\eta_X : X \to TX$ za poljubno množico X,
- *veriženjem* $\gg\!=$, ki je družina preslikav (naravna transformacija) $\gg\!=_{X,Y} : TX \to (X \to TY) \to TY$ za poljubni množici X in Y,

ki zadoščajo zakonom:

- $\eta(x) \gg\!=_{X,Y} k = k(x)$ za poljuben $x \in X$ ter $k : X \to TY$,
- $m \gg\!=_{X,X} \eta_Y$ za poljuben $m \in TX$,
- $(m \gg\!=_{X,Y} k) \gg\!=_{Y,Z} k' = m \gg\!=_{X,Z} (x \mapsto k(x) \gg\!=_{Y,Z} k')$ za poljuben $m \in TX$, $k : X \to TY$ in $k' : Y \to TZ$.

Primeri monad

Izjeme

Monado za izjeme iz množice $\mathbb{E}$ podamo kot:

$$\begin{aligned} TX &= X + \mathbb{E} \\ \eta_X(x) &= \iota_1(x) \\ m \underset{X,Y}{\gg\!=} k &= \begin{cases} k(x) & m = \iota_1(x) \\ \iota_2(e) & m = \iota_2(e) \end{cases} \end{aligned}$$

Preverimo še veljavnost zakonov:

$$
\begin{aligned}
\eta(x) \underset{X,Y}{\gg\!=} k &= \iota_1(x) \underset{X,Y}{\gg\!=} k = k(x) \\
m \underset{X,Y}{\gg\!=} \eta_X(x) &= \left(\begin{cases} \iota_1(x) & m = \iota_1(x) \\ \iota_2(e) & m = \iota_2(e) \end{cases}\right) = m \\
(m \gg\!= \quad k) \gg\!= \quad k' &= \left(\begin{cases} k(x) & m = \iota_1(x) \\ \iota_2(e) & m = \iota_2(e) \end{cases}\right) \gg\!= \quad k' \\
&= \begin{cases} k(x) \gg\!= k' & m = \iota_1(x) \\ \iota_2(e) \gg\!= k' & m = \iota_2(e) \end{cases} \\
&= \begin{cases} k(x) \gg\!= k' & m = \iota_1(x) \\ \iota_2(e) & m = \iota_2(e) \end{cases} \\
&= m \gg\!= \quad (x \mapsto k(x) \gg\!= \quad k')
\end{aligned}
$$

Pomnilnik

Monado za delo s pomnilnikom nad množico stanj S podamo kot:

$$
\begin{aligned}
TX &= (X \times S)^S \\
\eta_X(x) &= (s \mapsto (x, s)) \\
m \underset{X,Y}{\gg\!=} k &= (s \mapsto k(\underbrace{\pi_1(m(s))}_{x})(\underbrace{\pi_2(m(s))}_{s'}))
\end{aligned}
$$

Veriženje deluje tako, da $m : S \to (X \times S)$ uporabi na začetnem stanju $s \in S$ in dobi vrednost $x = \pi_1(m(s)) \in X$ ter novo stanje $s' = \pi_2(m(s)) \in S$. Nato $k : X \mapsto (Y \times S)^S$ uporabi na x, da dobi $k(x) : S \to (Y \times S)$, ki ga uporabi še na s', da dobi končni rezultat $k(x)(s') : Y \times S$.

Preverimo še veljavnost zakonov::

$$
\begin{aligned}
(\eta(x) \underset{X,Y}{\gg\!=} k) &= (s \mapsto k(\pi_1(\eta(x)(s)))(\pi_2(\eta(x)(s)))) \\
&= (s \mapsto k(\pi_1((x, s)))(\pi_2((x, s)))) \\
&= (s \mapsto k(x)(s)) \\
&= k(x) \\
m \underset{X,Y}{\gg\!=} \eta(x) &= (s \mapsto \eta(\pi_1(m(s)))(\pi_2(m(s)))) \\
&= (s \mapsto (\pi_1(m(s)), \pi_2(m(s)))) \\
&= (s \mapsto m(s)) \\
&= m \\
(m \gg\!= \quad k) \gg\!= \quad k' &= (s \mapsto k(\pi_1(m(s)))(\pi_2(m(s)))) \gg\!= \quad k' \\
&= (s \mapsto k(\pi_1(m(s)))(\pi_2(m(s)))) \gg\!= \quad k' \\
&= (s \mapsto k'(\pi_1(k(\pi_1(m(s)))(\pi_2(m(s)))))(\pi_2(k(\pi_1(m(s)))(\pi_2(m(s)))))) \\
&= m \gg\!= \quad (x \mapsto (s \mapsto k'(\pi_1(k(x)(s)))(\pi_2(k(x)(s))))) \\
&= m \gg\!= \quad (x \mapsto k(x) \gg\!= \quad k')
\end{aligned}
$$
