Cat. theory Assignment 2 - Xiang Li

u6319368

Page 1

2.1 Idempotent completion

1) Kar(primeVec) is equivalent to FdVec; ie

$$\begin{array}{ccc} (V,p) & \xmapsto{\alpha} & pV \\ f\downarrow & \longmapsto & \downarrow {\scriptstyle pv \mapsto f(v)} \\ (W,q) & \longmapsto & qW \end{array}$$

is full, faithfull and surjective

proof: well defined: if $pv = pv'$, then $fv = fpv = fpv' = fv'$ □

α is a functor $\alpha(1_{(V,p)}) = \alpha(p : V \to V) = \begin{pmatrix} pV \to pV \\ pv \mapsto p(v) \end{pmatrix} = 1_{pV}$

$(V,p) \xrightarrow{f} (W,q) \xrightarrow{g} (X,r)$

$\alpha g \alpha f(pv) = \alpha g(f(v)) \underset{qf=f}{=} \alpha g q f(v) \underset{gq=g}{=} g f(v) = \alpha(gf)(v)$ □

α is surjective: given $W \in$ FdVec one can form $W \times K^n \in$ primeVec (for a suitable n) and $\alpha(W \times K^n, \pi_W) = W$

α is full give $h : pV \to qW$ define: $\tilde{h} : \begin{matrix} V \to W \\ v \mapsto h(pv) \end{matrix}$

- $\tilde{h}$ is a map $(V,p) \to (W,q)$ since $\tilde{h}pv = hppv = hv$ and $q\tilde{h}v = qhpv = hv$
- $h = \alpha(\tilde{h})$ since $\alpha(\tilde{h}(v)) = \alpha(h(pv)) = h(v)$

α is faithful: $(V,p) \overset{f,f'}{\rightrightarrows} (W,q)$ If $\alpha f = \alpha f'$ then $fv = \alpha f\, pv = \alpha f'\, pv = f'v$

Page 2

2.1.2 $\operatorname{Kar}\operatorname{Kar} C \cong \operatorname{Kar} C$

Notation: For maps f in C, sometimes f, $\tilde{f}$ and $\hat{f}$ denote the corresponding maps in C, $\operatorname{Kar} C$, and $\operatorname{Kar}\operatorname{Kar} C$.

$$
\begin{array}{ccc}
\operatorname{Kar} C & \xrightarrow{\alpha} & \operatorname{Kar}\operatorname{Kar} C \\
(X,p) & \longmapsto & ((X,p),\tilde{p}) \\
\downarrow \tilde{f} & \longmapsto & \alpha(\tilde{f}) := \hat{f} \; \downarrow \\
(Y,q) & \longmapsto & ((Y,q),\tilde{q})
\end{array}
$$

The inverse natural transformation is $\beta: ((X,p),\tilde{q}) \longmapsto (X,p)$,
$f \longmapsto f$.

$\alpha\beta((X,p),\tilde{q}) = ((X,p),\tilde{p}) \cong ((X,p),\tilde{q})$
see ess. surj below

$\beta\alpha(X,p) = X,p$

$\Rightarrow \beta$ is α's inverse

Since $\hat{e}$ and $\tilde{f}$ are basically f, α is like the identity on maps. Therefore α is full and faithful.

full: given $((X,p),\tilde{p}) \xrightarrow{\hat{f}} ((Y,q),\tilde{q})$. This implies $\hat{f}\tilde{p} = \hat{f} = \tilde{q}\hat{f}$ and $fp = f = qf$.
And therefore for $\tilde{f}: (X,p) \xrightarrow{f} (Y,q)$, $\alpha(\tilde{f}) = \hat{f}$

faithful:

$\tilde{f}$ underlies $\alpha(\tilde{f})$

ess. surj:

$$((X,p),\tilde{q}) \underset{p}{\overset{p}{\rightleftarrows}} ((X,p),\tilde{p})$$

is an isomorphism since $id_{(X,p)} = p = p^2$

2.1.3

$$
\begin{array}{ccc}
C & \overset{\alpha}{\hookrightarrow} & \operatorname{Kar} C \\
X & \longmapsto & (X,1_X) \\
\downarrow g & \longmapsto & g \; \downarrow \\
Y & \longmapsto & (Y,1_Y)
\end{array}
$$

$\alpha(g)$. well defined bec. of the identy property of 1_X, 1_Y

$\alpha(g)$ is basically g. Therefore α is full and faithful.

2.1.4 $I : \mathrm{Kar}\,C \to \mathrm{Kar}\,D \simeq L_D \circ \mathrm{Kar}\,F_{IC}$

This only a collection of ideas ~~draft~~, not a solution.
I only came up with a one-sided inverse on $F(X,p)$.

1. Understand the functors.

$$F : \mathrm{Kar}\,C \to \mathrm{Kar}\,D, \quad (X,p) \mapsto F(X,p), \quad h \mapsto Fh$$

$$F_{IC} : C \xrightarrow{(-,1)} \mathrm{Kar}\,C \xrightarrow{F} \mathrm{Kar}\,D$$

$$\begin{tikzcd}
X \arrow[r, mapsto] \arrow[d, "g"] & (X,1_X) \arrow[r, mapsto] \arrow[d, "g"] & F(X,1_X) \arrow[d, "Fg"] \\
X' \arrow[r, mapsto] & (X',1_{X'}) \arrow[r, mapsto] & F(X',1_{X'})
\end{tikzcd}$$

$$L_D \circ \mathrm{Kar}\,F_{IC} : \mathrm{Kar}\,C \xrightarrow{\mathrm{Kar}\,F_{IC}} \mathrm{Kar}\,\mathrm{Kar}\,D \xrightarrow{L_D} \mathrm{Kar}\,D$$

$$\begin{tikzcd}
(X,p) \arrow[r, mapsto] \arrow[d, "g"] & (F_{IC}(X), F_{IC}(p)) = (F(X,1_X), Fp) \arrow[r, mapsto] \arrow[d, "Fg"] & F(X,1_X) \arrow[d, "Fg"] \\
(X',p') & (F(X',1_{X'}), Fp') & F(X',1_{X'})
\end{tikzcd}$$

2. Construct Natural Transformations $\left(F(X,p) \underset{Fp}{\overset{Fp}{\rightleftarrows}} F(X,1_X)\right)_{(X,p)\in \mathrm{Kar}\,C}$

well defined:

Fp is a map: p is a map $(X,p) \underset{p}{\overset{p}{\rightleftarrows}} (X,1_X)$ because $p^2 = p = 1_X p = p 1_X$.

Since F is a functor Fp is a map $F(X,p) \rightleftarrows F(X,1_X)$ in $\mathrm{Kar}\,D$

Fp is a natural transformation

given $(X,p) \xrightarrow{h} (Y,q)$ we have to check

$$\begin{tikzcd}
F(X,p) \arrow[r, "Fh"] \arrow[d, "Fp"] & F(Y,q) \arrow[d, "Fq"] \\
F(X,1_X) \arrow[r, "F_{IC}h = Fh"] & F(Y,1_Y)
\end{tikzcd}$$

This is true since $FqFh = F(qh) = F(hp) = FhFp$

Fp is a self inverse isomorphism?

$f_0 = Fp \circ Fp$ is the identity on $F(X,p)$ since $id_{(X,p)} = p$ and F preserves isomorphism.

How about $Fp \circ Fp \overset{?}{=} id_{F(X,1_X)} = F(1_X)$?

Page 4

2.2 Cones

Define a functor: Cones $\longrightarrow (\mathrm{val} \downarrow \tilde{F})$ = Comma Category

object: $X \in C, (X \xrightarrow{f_j} F(j))_{j\in J} \longmapsto (X, \cdot, x = \mathrm{val}_X(j) \xmapsto{f_j} F(j))_{j \in J}$

This means

$$C \xrightarrow{\mathrm{val}} \mathrm{Fun}(J, C) \xleftarrow{\tilde{F}} 1$$

$$X \longmapsto \left(\begin{array}{l} \mathrm{val}_X : J \to C \\ j \mapsto X \\ \downarrow h \mapsto \downarrow id_X \\ j' \mapsto X \end{array} \right) \qquad \begin{array}{l} F \longleftarrow\!\mapsto \cdot \\ \downarrow id \longleftarrow\!\mapsto \downarrow id \\ F \longleftarrow\!\mapsto \cdot \end{array}$$

Satisfying for $g : X \to X'$

$$\begin{tikzcd} \mathrm{val}X \arrow[r, "\mathrm{val}g = (-\circ g)_{j\in J}"] \arrow[d, "(f_j)_j"] & \mathrm{val}(X') \arrow[d, "(f_j')_j"] \\ F \arrow[r, "id"] & F \end{tikzcd}$$

ie $\forall j \in J : f_j = f_j' \circ g$ (1)

map:

$$\begin{array}{c} X, (X \xrightarrow{f_j} F_j)_j \\ \downarrow g \\ X', (X' \xrightarrow{f_j'} F_j)_j \end{array} \longmapsto (g, id) \quad \begin{array}{c} (X, \cdot, (f_j)_j) \\ g\downarrow \quad \downarrow id \\ (X', \cdot, (f_j')_j) \end{array}$$

Satisfying $\forall j$ $f_j = f_j' g$ (2)

This functor is full: Since 1 only has the identity map. all maps in $\mathrm{val} \downarrow \tilde{F}$ are of the form (g, id).

g is a preimage as (2) and (1) are the same

faithful: Let (g, id) and (g', id) be arbitrary maps.

Then $(g, id) = (g', id) \iff g = g'$

surjective: $(X, \cdot, (x = \mathrm{val}_X(j) \xmapsto{f_j} F_j)_j)$ has the preimage $X, (X \xrightarrow{f_j} F_j)_j$ because

for $h : j \to j'$ $\begin{tikzcd} X \arrow[r, "f_j"] \arrow[dr, "f_{j'}"'] & F_j \arrow[d, "Fh"] \\ & F_{j'} \end{tikzcd}$ is satisfied ~~as the~~ due to the naturality

Page 5

of $(f_j)_j$:

$$\begin{tikzcd} X = \mathrm{val}_X(j) \arrow[r, "f_j"] \arrow[d, "\mathrm{id}_X"'] \arrow[dr, phantom, "\circlearrowleft"] & Fj \arrow[d, "Fh"] \\ X = \mathrm{val}_X(j') \arrow[r, "f_{j'}"] & Fj' \end{tikzcd}$$

$\Rightarrow$ The category of Cones is equivalent to the comma category

Cat. theory - Assign. 2 - Xiangli Page ~~5~~ 6

Yoneda Lemma **2.3.1**

Let D be a locally small category. Then $[D^{op}, Set](H_A, X) \cong X(A)$ naturally in $A \in C$, $X \in [D^{op}, Set]$

ie

$$\begin{tikzcd}[column sep=huge] D^{op} \times [D^{op}, Set] \arrow[r, bend left=40, "F"] \arrow[r, bend right=40, "G"'] \arrow[r, phantom, "\alpha \Uparrow \; \Downarrow \beta_A"] & Set \end{tikzcd}$$

$$nt(H_A, X) \qquad (X_{-}(y))_{B \in A} \xleftarrow{\alpha} y \in X(A) \qquad \eta \xmapsto{\beta_A} \eta_A(1_A) \qquad X(A)$$

α and β are natural transformations and mutually inverse.

F denotes the functor

$$\begin{tikzcd} D^{op} \times [D^{op}, Set] \arrow[r, "H_\bullet^{op} \times 1"] & {[D^{op}, Set]^{op} \times [D^{op}, Set]} \arrow[r, "Hom_{[D^{op}, Set]}"] & Set \\ (A, X) \arrow[r, mapsto] \arrow[d, "g"] & (H_A, X) \arrow[r, mapsto] \arrow[d, dashed, "((-\circ f)^{op}, g) = (f\circ -, g)"] & nt(H_A, X) \arrow[d, dashed, "g \circ - \circ (f \circ -)"] \\ (A', X') \arrow[u, "f"] \arrow[r, mapsto] & (H_{A'}, X') \arrow[r, mapsto] & nt(H_{A'}, X') \end{tikzcd}$$

G denotes the functor

$$\begin{tikzcd} D^{op} \times [D^{op}, Set] \arrow[r] & Set \\ (A, X) \arrow[r, mapsto] \arrow[d, "g"] & X(A) \arrow[d, dashed, "g_{A'} (Xf\ -)"] \\ (A', X') \arrow[u, "f"] \arrow[r, mapsto] & X'(A') \end{tikzcd}$$

β is natural in X **2.3.1 (continued)**

Let $g: X \to X'$ be a map in $[D^{op}, Set]$ (i.e. a nat tr)

$$\begin{tikzcd}
n \arrow[rrr, mapsto] \arrow[ddd, mapsto] & & & g_B(n_B(-)) \arrow[ddd, mapsto] \\
& nt(H_A, X) \arrow[r, "g\circ -"] \arrow[d, "\beta_A"'] & nt(H_A, X') \arrow[d, "\beta_A"] & \\
& X(A) \arrow[r, "g_A -"] & X'(A) & \\
n_A(1_A) \arrow[rrr, mapsto] & & & g_A(n_A(1_A))
\end{tikzcd}$$

□

β is natural in X and A

It follows from the following lemma:

~~Given~~ $\cancel{A \times B \underset{G}{\overset{F}{\rightrightarrows}} C}$

$$\begin{tikzcd}
F(A,X) \arrow[r, "b"] \arrow[d, "a"'] & F(A',X) \arrow[r, "d"] \arrow[d, "e"] & F(A',X') \arrow[d, "g"] \\
G(A,X) \arrow[r, "c"] & G(A',X) \arrow[r, "f"] & G(A',X')
\end{tikzcd}$$

commutativity follows from the naturality in A and X:

$g\,d\,b = f\,e\,b = f\,c\,a$

2.3.2

$fdVec^{op} \simeq fdVec$

$V \overset{\alpha}{\longmapsto} V^*$

$f \longmapsto (- \circ f) \equiv f^*$

$V^* \overset{\beta}{\longleftarrow\!\!\shortmid} V$

$f^* \longleftarrow\!\!\shortmid f$

$\alpha(\beta(V)) = V^{**} \cong \beta(\alpha(V))$

$id_{nt}(V) = V$

Since V and V^{**} are naturally isomorphic, α and β are part of the equivalence

Since $fdVec \xrightarrow{(-)^*} fdVec^{op}$ is a functor, it is also functor as $Vec^{op} \to fdVec$ because of $Functor(C^{op} \to D) = Fun(C \to D^{op})$

proof: $\alpha: C^{op} \xrightarrow{\alpha} D$. Then in D: $\alpha(\underset{\text{in } C}{fg}) = \alpha(\underset{\text{in } C^{op}}{gf}) = \alpha g\, \alpha f$

given two maps f, g in C: $\alpha(\underset{\text{in } C}{fg}) = \alpha g\ \alpha f$ in D

$= \alpha f\ \alpha g$ in D^{op}

The identity property is trivial

2: replace C and D by C^{op} and D^{op} and use $C^{op\,op} = C$

Page 18

α is well-defined

ie For $A \in D$, $y \in X(A)$ $\quad \alpha_A(y)$ is a nat. tr.

$$\begin{tikzcd} D^{op} \arrow[r, bend left, "H_A"] \arrow[r, bend right, "X"'] \arrow[r, phantom, "\Downarrow \alpha_A(y)"] & Set \end{tikzcd}$$

$$\alpha_A(y) = \left(H_A(B) \xrightarrow{X_-(y)} X(B)\right)_{B \in D}$$

check nat. diagram:

given $f: B' \to B$

$$\begin{tikzcd} H_A(B) \arrow[r, "\_ \circ f"] \arrow[d, "{\alpha_A(y)_B = X_-(y)}"'] & H_A(B') \arrow[d, "X_-(y)"] \\ X(B) \arrow[r, "Xf\_"'] & X(B') \end{tikzcd}$$

$$\begin{tikzcd} g \arrow[r, mapsto] \arrow[d, mapsto] & g \circ f \arrow[d, mapsto] \\ Xg(y) \arrow[r, mapsto, "X \text{ is contrav. functor}"'] & X(g\circ f)y = Xf\,Xg(y) \end{tikzcd}$$

α∘β = id

given $A \in D^{op}$, $X \in [D^{op}, set]$ we need to check that for any $n \in nt(H_A, X)$ ie $n = \left(H_A(B) \xrightarrow{n_B} X(B)\right)_{B \in D}$

n equals $\alpha \circ \beta(n) = \left(H_A(B) \xrightarrow{X_-(n_A(1_A))} X(B)\right)_{B \in D}$

proof. For $f \in H_A(B)$ $n_B(f) = Xf\, n_A(1_A)$ follows from the naturality of n and $1_A \in H_A(A)$

$$\begin{tikzcd} H_A A \arrow[r, "\_ \circ f"] \arrow[d, "n_A"'] & H_A B \arrow[d, "n_B"] \\ XA \arrow[r, "Xf\_"'] & XB \end{tikzcd}$$

β∘α = id

Given X, A, $y \in X(A)$:

$$\beta\circ\alpha(y) = X_{1_A}(y) = 1_{X(A)}(y) = y$$

β is natural in A

Let $f: A' \to A$ in D. Then

$$\begin{tikzcd} nt(H_A, X) \arrow[r, "n \circ (f \circ \_)"] \arrow[d, "\beta_A"'] & nt(H_{A'}, X) \arrow[d, "\beta_B"] \\ X(A) \arrow[r, "Xf\_"'] & X(A') \end{tikzcd}$$

$$\begin{tikzcd} n \arrow[r, mapsto] \arrow[d, mapsto] & n \circ f \circ \_ \arrow[d, mapsto] \\ n_A(1_A) \arrow[r, mapsto] & Xf\, n_A(1_A) = n \circ f \circ 1_{A'} = n_B \circ f \end{tikzcd}$$

equal as seen above

2.3.2 continued

$Set^{op} \not\simeq Set$

If two categories are equivalent, then there is a full, ess. surj and faithfull functor F between them. Therefore there is a bijection between maps $A \to A'$ and maps $FA \to FA'$. It follows that A is terminal iff FA is terminal.

In Set there is a map from its terminal object to a non-terminal object. (Any function with codomain $\neq \emptyset$). If there is an equivalence this must also be true in Set^{op}.

This would mean that there is a function to the initial object in Set ($\emptyset$) from a non-initial object (non-empty set). However, there is no such function. ⨳