Sorting with Bialgebras MGS 2015 #3

Type K of keys, ordered.

$LX = 1 + K \times X$ $\quad \underline{L} = L$, but "ordered"

$\mu F \underset{In}{\overset{in^{\circ}}{\rightleftarrows}} F \mu F \qquad \nu G \underset{Out^{\circ}}{\overset{out}{\rightleftarrows}} G \nu G$

Sorting has type $\mu L \to \nu \underline{L}$.

Functions of type $\mu F \to \nu G$ include

$\quad$ unfold (fold ($f :: FG\mu F \to G\mu F$))

and fold (unfold ($g :: F\nu G \to GF\nu G$))

Lambek: $G\mu F \approx GF\mu F \qquad G\, in^{\circ} \circ f : FG\mu F \to GF\mu F$

$\qquad F\nu G \approx FG\nu G \qquad g \circ F\, out^{\circ} : FG\nu G \to GF\nu G$

Apparently, we want distributive laws

$\quad FG \dot{\to} GF$

---

data $L\, x$ = Nil | Cons $K\, x$

data $\underline{L}\, x$ = $\underline{Nil}$ | $\underline{Cons}$ $K\, x$

Consider

isort = fold insert

insert = unfold ins

ins :: $L(\nu \underline{L}) \to \underline{L}(L(\nu \underline{L}))$

ins Nil $\quad = \underline{Nil}$

ins (Cons a (Out° $\underline{Nil}$)) = $\underline{Cons}$ a Nil

ins (Cons a (Out° ($\underline{Cons}$ b x)))

$\quad$ | a ≤ b $\quad = \underline{Cons}$ a (Cons b x)

$\quad$ | otherwise $\quad = \underline{Cons}$ b (Cons a x)

bsort then isort

Dually

$$\text{bsort} = \text{unfold bubble}$$
$$\text{bubble} = \text{fold bub}$$
$$\text{bub} :: L(\underline{L}(\mu L)) \to \underline{L}(\mu L)$$

```
bub Nil                  = Nil
bub (Cons a Nil)         = Cons a (In Nil)
bub (Cons a (Cons b x))
    | a ≤ b              = Cons a (In (Cons b x))
    | otherwise          = Cons b (In (Cons a x))
```

Note similarities between ins & bub.

$$\text{swap} :: L(\underline{L}\,x) \to \underline{L}(L\,x)$$

```
swap Nil                  = Nil
swap (Cons a Nil)         = Cons a Nil
swap (Cons a (Cons b x))
    | a ≤ b               = Cons a (Cons b x)
    | otherwise           = Cons b (Cons a x)
```

so

$$\text{ins} = \text{swap} \circ L\ \text{out}$$
$$\text{bub} = \underline{L}\ \text{In} \circ \text{swap}$$

Note also, now polymorphic.

$$
\begin{tikzcd}
L\mu L \arrow[d, "\mathrm{In}"'] \arrow[rr, "L\,\mathrm{bubble}"] & & L\underline{L}\mu L \arrow[d, "\mathrm{swap}"] \\
\mu L \arrow[d, "\mathrm{bubble}"'] & & \underline{L}L\mu L \arrow[dll, "\underline{L}\,\mathrm{In}"] \\
\underline{L}\mu L & &
\end{tikzcd}
\qquad
\begin{tikzcd}
L\nu\underline{L} \arrow[d, "\mathrm{insert}"'] \arrow[rr, "L\,\mathrm{out}"] & & L\underline{L}\nu\underline{L} \arrow[d, "\mathrm{swap}"] \\
\nu\underline{L} \arrow[d, "\mathrm{out}"'] & & \underline{L}L\nu\underline{L} \arrow[dll, "\underline{L}\,\mathrm{insert}"] \\
\underline{L}\nu\underline{L} & &
\end{tikzcd}
$$

For $\delta : FG \to GF$, a $\delta$-bialgebra $(X,a,c)$ is $F$-algebra $(X,a)$ ad $G$-coalgebra $(X,c)$ such that:

$$
\begin{tikzcd}
FX \arrow[d, "a"'] \arrow[rr, "Fc"] & & FGX \arrow[d, "\delta"] \\
X \arrow[d, "c"'] & & GFX \arrow[dll, "Ga"] \\
GX & &
\end{tikzcd}
$$

Homomorphism $h : (X,a,c) \to (Y,a',c')$ betwea $\delta$-bialgebras is $h : X \to Y$ such that $h : (X,a) \to (Y,a')$ ad $h : (X,c) \to (Y,c')$ homs.

Identity is a hom, and they compose; so they form a category.

$$
\begin{tikzcd}
FX \arrow[r, "Fh"] \arrow[d, "a"'] & FY \arrow[d, "a'"] \\
X \arrow[r, "h"] \arrow[d, "c"'] & Y \arrow[d, "c'"] \\
GX \arrow[r, "Gh"] & GY
\end{tikzcd}
$$

Initial algebras lift to initial bialgebras

$$\begin{tikzcd}
& F\mu F \arrow[rr, "F(\mathrm{fold}\,a)"] \arrow[dl, "Fg"'] \arrow[dd, "In"] & & FX \arrow[dr, "Fc"] \arrow[dd, "a"] & \\
FG\mu F \arrow[dd, "\delta"'] & \arrow[rr, phantom, "(3)"] & & & FGX \arrow[dd, "\delta"] \\
\arrow[r, phantom, "(1)"] & \mu F \arrow[rr, "\mathrm{fold}\,a"] \arrow[dd, "g"] & & X \arrow[dd, "c"] \arrow[r, phantom, "(2)"] & \\
GF\mu F \arrow[dr, "GIn"] & \arrow[rr, phantom, "(4)"] & & & GFX \arrow[dl, "Ga"'] \\
& G\mu F \arrow[rr, "G(\mathrm{fold}\,a)"] & & GX &
\end{tikzcd}$$

ie fold a is unique hom $(\mu F, In, g) \to (X, a, c)$
(1), (2) commute by defn.
fold a is unique arrow to make (3) commute
Need to check (4). Note (1) implies
$g : (\mu F, In) \to (G\mu F, G In \circ \delta)$, so $g = \mathrm{fold}(G In \circ \delta)$
Similarly $c$ a hom $(X, a) \to (GX, Ga \circ \delta)$
so $c \circ \mathrm{fold}\, a = \mathrm{fold}(Ga \circ \delta)$.
Then (4) follows by fusion:
$$G(\mathrm{fold}\,a) \circ G In \circ \delta = Ga \circ \delta \circ FG(\mathrm{fold}\,a)$$
ie $(\mu F, In, \mathrm{fold}(G In \circ \delta))$ initial $\delta$-bialgebra

Dually, unfold c is unique hom from $(X, a, c)$,
so $(\nu G, \mathrm{unfold}(\delta \circ F\,out), out)$ is final $\delta$-bialgebra.

Specifically, $F ::= L$, $G ::= \underline{L}$, $\delta ::= swap$

Then $(\mu L, In, fold\ bub)$ initial bialgebra

$(\nu \underline{L}, unfold\ ins, out)$ final

By initiality, fold (unfold ins) is unique hom between them. But by finality, so is unfold (fold bub). Hence they're equal — not just extensionally ("both sort") but intensionally ("same algorithm") too. cf diagrams on p138.

Also works for apo and para.

```
apo :: functor f ⇒ (a → f(νf + a)) → a → νf
insert = apo ins
ins :: L(νL) → L(νL + LνL)
ins Nil = Nil
ins (Cons a (In Nil)) = Cons a (Lft (In Nil))
ins (Cons a (In (Cons b x')))
      | a≤b   = Cons a (Lft (In (Cons b x')))
      | ow    = Cons b (Rgt (Cons a x'))
```

and dually...

$para :: Functor\ f \Rightarrow (f(\mu f \times a) \to a) \to \mu f \to a$

$select = para\ sel$

$sel :: L(\mu L \times \underline{L}\,\mu L) \to \underline{L}\,\mu L$

$sel\ Nil = \underline{Nil}$

$sel\ (Cons\ a\ (x, \underline{Nil})) = \underline{Cons}\ a\ x$

$sel\ (Cons\ a\ (x, \underline{Cons}\ b\ x'))$

$\quad |\ a \le b = \underline{Cons}\ a\ x$

$\quad |\ ow = \underline{Cons}\ b\ (In\ (Cons\ a\ x'))$

~~Unify~~ Make polymorphic: "swap & stop"

$ins = swop \circ L\ out$ $\qquad F_+ A = A + FA$

$swop :: L(\underline{L} X) \to \underline{L}(L_+ X)$ $\qquad F_\times A = A \times FA$

and unify

$swop_2 :: L(\underline{L}_\times X) \to \underline{L}(L_+ X)$

$swop_2\ Nil = \underline{Nil}$

$swop_2\ (Cons\ a\ (x, \underline{Nil})) = \underline{Cons}\ a\ (Lft\ (In\ Nil))$

$swop_2\ (Cons\ a\ (x, \underline{Cons}\ b\ x'))$

$\quad |\ a \le b = \underline{Cons}\ a\ (Lft\ (In\ (\underline{Cons}\ b\ x')))$

$\quad |\ ow = \underline{Cons}\ b\ (Rgt\ (Cons\ a\ x'))$

$ins = swop_2 \circ L(id \vartriangle out)$ ← invariant! so replace by $x$

$sel = \underline{L}\ (id \triangledown In) \circ swop_2$

With more work, $swopsy :: L_+ \underline{L}_\times \stackrel{\cdot}{\to} \underline{L}_\times L_+$.