Proofs Regarding Specification of Spark Aggregation

Yu-Fang Chen\(^1\), Chih-Duo Hong\(^1\), Ondřej Lengál\(^{1,2}\), Shin-Cheng Mu\(^1\), Nishant Sinha\(^3\), and Bow-Yaw Wang\(^1\)

\(^1\)Academia Sinica, Taiwan
\(^2\)Brno University of Technology, Czech Republic
\(^3\)IBM Research, India

Laws regarding monads
Recall the monad laws

\[
\begin{align*}
    f \circ \text{\textit{return}} x &= f x, \quad & (1) \\
    \text{\textit{return}} &= \text{\textit{m}}, \quad & (2) \\
    f \circ (g \circ \text{\textit{m}}) &= (\lambda x \rightarrow f \circ g x) \circ \text{\textit{m}}. \quad & (3)
\end{align*}
\]

Monadic application and composition are defined as:

\[
\begin{align*}
    (\text{\textit{m}}) &:: (b \rightarrow m c) \rightarrow (a \rightarrow m b) \rightarrow a \rightarrow m c \\
    (f \circ g) x &= f \circ g x \\
    (\text{\textit{s}}) &:: (a \rightarrow b) \rightarrow m a \rightarrow m b \\
    f \circ m &= (\text{\textit{return}} \cdot f) \circ m \\
    (\text{\textit{r}}) &:: (b \rightarrow c) \rightarrow (a \rightarrow m b) \rightarrow (a \rightarrow m c) \\
    f \circ g &= ((\text{\textit{return}} \cdot f) \circ \text{\textit{r}}) \cdot g
\end{align*}
\]

Laws concerning them include:

\[
\begin{align*}
    (f \circ g) x &= f \circ g x, \quad & (4) \\
    f \circ (g \circ \text{\textit{m}}) &= (f \cdot g) \circ \text{\textit{m}}, \quad & (5) \\
    f \circ (g \circ \text{\textit{m}}) &= (f \cdot g) \circ m, \quad & (6) \\
    f \circ (g \circ \text{\textit{m}}) &= (f \cdot g) \circ \text{\textit{m}}, \quad & (7) \\
    f \circ (g \circ h) &= (f \cdot g) \circ h, \quad & (8) \\
    f \circ (g \circ h) &= (f \cdot g) \circ h. \quad & (9)
\end{align*}
\]

Proofs of the properties above will be given later, so as not to distract us from the main theorems.
Regarding monad-plus, we want \([\_\_\_]\) to be associative, with \textit{mzero} as identity. Monadic bind distributes into \([\_\_\_]\) from the end:

\[
    f \lln (m \lln n) = (f \lln m) \lln (f \lln n) .
\]  

(10)

It is less mentioned, but not uncommon, to demand that \([\_\_\_]\) is also commutative and idempotent.

1 Folding and Shuffling

Lemma 1. Given \((\odot) :: a \rightarrow b \rightarrow b\), we have

\[
    \text{foldr} (\odot) z \lln \text{insert } x \cdot xs = \text{return} \cdot \text{foldr} (\odot) z \cdot (x:xs) ,
\]

provided that \(x \odot (y \odot z) = y \odot (x \odot z)\) for all \(x, y :: a\) and \(z :: b\).

Proof. Prove \(\text{foldr} (\odot) z \lln \text{insert } x \cdot xs = \text{return} \cdot \text{foldr} (\odot) z \cdot (x:xs)\). Induction on \(xs\).

Case \(xs :: []\).

\[
    \begin{align*}
    \text{foldr} (\odot) z \lln \text{insert } x \cdot [] & = \{ \text{definition of } (\_\_\_\) \} \\
    (\text{return} \cdot \text{foldr} (\odot) z) \lln \text{insert } x \cdot [] & = \{ \text{definition of insert } \} \\
    (\text{return} \cdot \text{foldr} (\odot) z) \lln \text{return } [x] & = \{ \text{monadic law (1)} \} \\
    \text{return} \cdot \text{foldr} (\odot) z \cdot [x] & .
    \end{align*}
\]

Case \(xs :: y : xs\).

\[
    \begin{align*}
    \text{foldr} (\odot) z \lln \text{insert } x \cdot (y : xs) & = \{ \text{definition of } (\_\_\_\) \} \\
    (\text{return} \cdot \text{foldr} (\odot) z) \lln \text{insert } x \cdot (y : xs) & = \{ \text{definition of insert } \} \\
    (\text{return} \cdot \text{foldr} (\odot) z) \lln \\
    (\text{return} \cdot \text{foldr} (\odot) z \cdot (y : xs)) & = \{ \text{by (10)} \} \\
    \text{return} \cdot \text{foldr} (\odot) z \cdot (y : xs) \lln \\
    ((\text{return} \cdot \text{foldr} (\odot) z \cdot (y : xs)) \lln \text{insert } x \cdot xs) & .
    \end{align*}
\]

Focus on the second branch:

\[
    \begin{align*}
    (\text{return} \cdot \text{foldr} (\odot) z \cdot (y : xs)) \lln \text{insert } x \cdot xs & = \{ \text{definition of foldr } \} \\
    (\text{return} \cdot (y \odot) \cdot \text{foldr} (\odot) z) \lln \text{insert } x \cdot xs & = \{ \text{by (5)} \} \\
    (\text{return} \cdot (y \odot)) \lln ((\text{foldr} (\odot) z \lln \text{insert } x \cdot xs) & = \{ \text{induction } \}
    \end{align*}
\]
\[(\text{return} \cdot (y \circ )) \approx \text{return} \cdot (\text{foldr} \circ ) \cdot (x : \mathit{xs})\]

\[= \{ \text{monadic law (1)} \}\]
\[\text{return} \cdot (y \circ \text{foldr} \circ ) \cdot (x : \mathit{xs})\]

\[= \{ \text{definition of foldr} \}\]
\[\text{return} \cdot (y \circ (x \circ \text{foldr} \circ ) \cdot (z \cdot \mathit{xs}))\]

\[= \{ \text{since } x \circ (y \circ z) = y \circ (x \circ z) \}\]
\[\text{return} \cdot (\text{foldr} \circ ) \cdot (z \cdot (x : y : \mathit{xs}))\] .

Thus we have
\[
(\text{foldr} \circ ) \cdot z \cdot \text{shuffle}! (y : \mathit{xs})
\]

\[= \{ \text{calulation above} \}\]
\[\text{return} \cdot \text{foldr} (\circ) \cdot z \cdot \text{shuffle}! [\ ]\]

\[= \{ \text{idempotence of } (\text{shuffle}) \}\]
\[\text{return} \cdot \text{foldr} (\circ) \cdot z \cdot (x : y : \mathit{xs})\].

\[\Box\]

**Lemma 2.** Given \((\circ) :: a \to b \to b\), we have
\[\text{foldr} (\circ) \cdot z \cdot \text{shuffle}! = \text{return} \cdot \text{foldr} (\circ) \cdot z ,\]
provided that \(x \circ (y \circ z) = y \circ (x \circ z)\) for all \(x, y :: a\) and \(z :: b\).

**Proof.** Prove that \(\text{foldr} (\circ) \cdot z \cdot \text{shuffle} \cdot \mathit{xs} = \text{return} \cdot \text{foldr} (\circ) \cdot z \cdot \mathit{xs}\). Induction on \(\mathit{xs}\).

**Case \(\mathit{xs} := []\)**

\[\text{foldr} (\circ) \cdot z \cdot \text{shuffle}! [\ ]\]

\[= \{ \text{definitions of } (\text{shuffle}) \text{ and } \text{shuffle}! \}\]
\[\text{return} \cdot \text{foldr} (\circ) \cdot z \cdot \text{shuffle}! [\ ]\]

\[= \{ \text{monadic law (1)} \}\]
\[\text{return} \cdot \text{foldr} (\circ) \cdot z \cdot [\ ]\].

**Case \(\mathit{xs} := x : \mathit{xs}\).**

\[\text{foldr} (\circ) \cdot z \cdot \text{shuffle}! (x : \mathit{xs})\]

\[= \{ \text{definition of } \text{shuffle}! \}\]
\[\text{foldr} (\circ) \cdot z \cdot \text{shuffle}! (x : \mathit{xs})\]

\[= \{ \text{monadic law (3)} \}\]
\[\lambda (\mathit{xs} : \to \text{foldr} (\circ) \cdot z \cdot \text{insert } x \cdot \text{shuffle} \cdot \mathit{xs}) \approx \text{shuffle}! \cdot \mathit{xs}\]

\[= \{ \text{Lemma 1} \}\]
\[\lambda (\mathit{xs} : \to \text{return} \cdot \text{foldr} (\circ) \cdot z \cdot (x : \mathit{xs})) \approx \text{shuffle}! \cdot \mathit{xs}\]

\[= \{ \text{definition of } \text{foldr} \}\]
\[\text{return} \cdot (x \circ) \cdot \text{foldr} (\circ) \cdot z \approx \text{shuffle} \cdot \mathit{xs}\]

\[= \{ \text{by (5)} \}\]
\[\text{return} \cdot (x \circ) \approx \text{foldr} (\circ) \cdot z \cdot \text{shuffle} \cdot \mathit{xs}\]

\[= \{ \text{induction} \}\]
\[\text{return} \cdot (x \circ) \approx \text{return} \cdot \text{foldr} (\circ) \cdot z \cdot \mathit{xs}\].
\[
\begin{align*}
\text{return } (x \odot \text{foldr } (\odot) z \cdot \text{xs}) \\
= \{ \text{monadic law (1)} \} \\
\text{return } (x \odot \text{foldr } (\odot) z \cdot \text{xs}) \\
= \{ \text{definition of foldr} \} \\
\text{return } (\text{foldr } (\odot) z \cdot \text{x:xs}) \ .
\end{align*}
\]

\section{Map, Filter, and Shuffling}

\textbf{Lemma 3.} \( \text{insert } (f \cdot x) \cdot \text{map } f = \text{map } f \cdot \text{insert } x. \)

\textbf{Proof.} Prove that \( \text{map } f \cdot \text{insert } x \cdot \text{xs} = \text{insert } (f \cdot x) \cdot (\text{map } f \cdot \text{xs}) \) for all \( \text{xs} \), by induction on \( \text{xs} \).

\textbf{Case} \( \text{xs} := y : \text{xs}. \)

\( \text{map } f \cdot \text{insert } x \cdot (y : \text{xs}) \)

\( = \{ \text{definition of insert} \} \)

\( \text{map } f \cdot (\text{return } (x \cdot y \cdot \text{xs}) \parallel ((y) \cdot \text{insert } x \cdot \text{xs})) \)

\( = \{ (10), \text{definition of } (\cdot) \} \)

\( (\text{map } f \cdot \text{return } (x \cdot y \cdot \text{xs}) \parallel (\text{map } f \cdot ((y) \cdot \text{insert } x \cdot \text{xs})) \) .

The first branch, by definition of \((\cdot)\) and monadic law (1), simplifies to \( \text{return } (\text{map } f \cdot (x \cdot y \cdot \text{xs})) \). The second branch:

\( \text{map } f \cdot ((y) \cdot \text{insert } x \cdot \text{xs}) \)

\( = \{ (6) \} \)

\( (\text{map } f \cdot (y)) \cdot \text{insert } x \cdot \text{xs} \)

\( = \{ \text{definition of map} \} \)

\( ((f \cdot y) \cdot \text{map } f) \cdot \text{insert } x \cdot \text{xs} \)

\( = \{ (6) \} \)

\( (f \cdot y) \cdot (\text{map } f \cdot \text{insert } x \cdot \text{xs}) \)

\( = \{ \text{induction} \} \)

\( (f \cdot y) \cdot (\text{insert } (f \cdot x) \cdot (\text{map } f \cdot \text{xs})) \) .

Thus we have

\( \text{map } f \cdot \text{insert } x \cdot (y : \text{xs}) \)

\( = \{ \text{calculation above} \} \)

\( \text{return } (f \cdot x \cdot y \cdot \text{map } f \cdot \text{xs}) \parallel ((y) \cdot \text{insert } (f \cdot x) \cdot (\text{map } f \cdot \text{xs})) \)

\( = \{ \text{definition of insert} \} \)

\( \text{insert } (f \cdot x) \cdot (f \cdot y \cdot \text{map } f \cdot \text{xs}) \)

\( = \{ \text{definition of map} \} \)

\( \text{insert } (f \cdot x) \cdot (\text{map } f \cdot (y : \text{xs})) \) .

\textbf{Lemma 4.} \( \text{shuffle!} \cdot \text{map } f = \text{map } f \cdot \text{shuffle!}. \)

\textbf{Lemma 5.} \( \text{shuffle!} \cdot \text{filter } p = \text{filter } p \cdot \text{shuffle!}. \)
3 Homomorphism, etc

Lemma 6. \( h = \text{hom}(\oplus) z \) if and only if \( \text{foldr}(\oplus) z \cdot \text{map} \, h = h \cdot \text{concat} \).

Proof. A Ping-pong proof.

Direction (\( \Rightarrow \)). Prove \( \text{foldr}(\oplus) z (\text{map} \, h \, \text{xss}) = h (\text{concat} \, \text{xss}) \) by induction on \( \text{xss} \).

Case \( \text{xss} := [] \):

\[
\begin{align*}
\text{foldr}(\oplus) z (\text{map} \, h \, []) &= \text{foldr}(\oplus) z [] \\
&= z \\
&= h (\text{concat} []) .
\end{align*}
\]

Case \( \text{xss} := \text{xs} : \text{xss} \):

\[
\begin{align*}
\text{foldr}(\oplus) z (\text{map} h (\text{xs} : \text{xss})) &= h \, \text{xs} \oplus \text{foldr}(\oplus) z (\text{map} \, h \, \text{xss}) \\
&= \{ \text{induction} \} \\
&= h \, \text{xs} \oplus h (\text{concat} \, \text{xss}) \\
&= \{ h \, \text{homomorphism} \} \\
&= h (\text{concat} (\text{xs} : \text{xss})) .
\end{align*}
\]

Direction (\( \Leftarrow \)). Show that \( h \) satisfies the properties being a list homomorphism. On empty list:

\[
\begin{align*}
h [] &= h (\text{concat} []) \\
&= \{ \text{assumption} \} \\
&= \text{foldr}(\oplus) z (\text{map} h []) \\
&= z .
\end{align*}
\]

On concatenation:

\[
\begin{align*}
h (\text{xs} \oplus \text{ys}) &= h (\text{concat} [\text{xs}, \text{ys}]) \\
&= \{ \text{assumption} \} \\
&= \text{foldr}(\oplus) z (\text{map} h [\text{xs}, \text{ys}]) \\
&= h \, \text{xs} \oplus (h \, \text{ys} \oplus z) \\
&= h \, \text{xs} \oplus h \, \text{ys} .
\end{align*}
\]

Lemma 7. Let \( (\oplus) :: b \rightarrow b \rightarrow b \) be associative on \( \text{img}(\text{foldr}(\otimes) z) \) with \( z \) as its identity, where \( (\otimes) :: a \rightarrow b \rightarrow b \). We have \( \text{foldr}(\otimes) z = \text{hom}(\oplus) z \) if and only if \( x \otimes (y \oplus w) = (x \otimes y) \oplus w \) for all \( x :: a \) and \( y, w \in \text{img}(\text{foldr}(\otimes) z) \).

Proof. A Ping-pong proof.
Direction (⇐). We show that \( \text{foldr} (\otimes) z \) satisfies the homomorphic properties. It is immediate that \( \text{foldr} (\otimes) z [\ ] = z \). For \( xs + ys \), note that

\[
\text{foldr} (\otimes) z (xs + ys) = \text{foldr} (\otimes) (\text{foldr} (\otimes) ys) xs .
\]

The aim is thus to prove that

\[
\text{foldr} (\otimes) (\text{foldr} (\otimes) ys) xs = (\text{foldr} (\otimes) z xs) \oplus (\text{foldr} (\otimes) z ys) .
\]

We perform an induction on \( xs \). The case when \( xs :\ = [\ ] \) trivially holds. For \( xs :\ = x : xs \), we reason:

\[
\text{foldr} (\otimes) (\text{foldr} (\otimes) ys) (x : xs) = x \otimes \text{foldr} (\otimes) (\text{foldr} (\otimes) ys) xs \\
= \{ \text{induction} \}
\]

\[
x \otimes ((\text{foldr} (\otimes) z xs) \oplus (\text{foldr} (\otimes) z ys)) \\
= \{ \text{assumption: } x \otimes (y \oplus w) = (x \otimes y) \oplus w \}
\]

\[
(x \otimes (\text{foldr} (\otimes) z xs)) \oplus (\text{foldr} (\otimes) z ys) \\
= (\text{foldr} (\otimes) z (x : xs)) \oplus (\text{foldr} (\otimes) z ys) .
\]

Direction (⇒). Given \( \text{foldr} (\otimes) z = \text{hom} (\oplus) z \). Let \( y = \text{foldr} (\otimes) z xs \) and \( w = \text{foldr} (\otimes) z ys \) for some \( xs \) and \( ys \). We reason:

\[
x \otimes (y \oplus w) \\
= x \otimes (\text{foldr} (\otimes) z xs \oplus \text{foldr} (\otimes) z ys) \\
= \{ \text{since } \text{foldr} (\otimes) z = \text{hom} (\oplus) z \}
\]

\[
x \otimes (\text{foldr} (\otimes) z (xs + ys)) \\
= \text{foldr} (\otimes) z (x : xs + ys) \\
= \{ \text{since } \text{foldr} (\otimes) z = \text{hom} (\oplus) z \}
\]

\[
= \text{foldr} (\otimes) z (x : xs) \oplus \text{foldr} (\otimes) z ys \\
= (x \otimes \text{foldr} (\otimes) z xs) \oplus \text{foldr} (\otimes) z ys \\
= (x \otimes y) \oplus w .
\]

\[
\square
\]

4 Aggregation

Lemma 8. Given \( (\otimes) : a \rightarrow b \rightarrow b \) and define \( xs \odot w = \text{foldr} (\otimes) w xs \). We have

\[
\text{foldr} (\otimes) z \cdot \text{concat} = \text{foldr} (\odot) z .
\]

Proof. By \text{foldr} fusion the proof obligation is

\[
\text{foldr} (\otimes) z (xs + ys) = \text{foldr} (\otimes) (\text{foldr} (\otimes) z ys) xs .
\]

Induction on \( xs \).

Case \( xs :\ = [\ ]:\
$foldr (\odot) \left( foldr (\odot) z ys \right) []$

$= foldr (\odot) z ys$

$= foldr (\odot) z ([] \oplus ys)$.

Case $xs := x:xs$:

$foldr (\odot) \left( foldr (\odot) z ys \right) \left( x:xs \right)$

$= x \otimes foldr (\odot) \left( foldr (\odot) z ys \right) xs$

$= \{ \text{ induction } \}

x \otimes foldr (\odot) z (xs \oplus ys)$

$= foldr (\odot) z \left( x:xs \oplus ys \right)$.

\[ \square \]

**Theorem 9.** $aggregate z (\odot) (\oplus) = return \cdot foldr (\oplus) z \cdot map \left( foldr (\odot) z \right)$, provided that $(\oplus)$ is associative, commutative, and has $z$ as identity.

**Proof.** We reason:

\[
aggregate z (\odot) (\oplus)
= \{ \text{ definition of } aggregate \}
foldr (\oplus) z \circ \left( shuffle! \cdot map \left( foldr (\odot) z \right) \right)
= \{ \text{ Lemma 4 } \}
foldr (\oplus) z \circ \left( map \left( foldr (\odot) z \right) \circ shuffle! \right)
= \{ \text{ by (7) } \}
\left( foldr (\oplus) z \cdot map \left( foldr (\odot) z \right) \right) \circ shuffle!
= \{ \text{ Lemma 2 } \}
return \cdot foldr (\oplus) z \cdot map \left( foldr (\odot) z \right).
\]

The last step holds because by $foldr$-$map$ fusion, for all $h$,

\[
foldr (\oplus) z \cdot map h = foldr (\odot) z
\]

\[ \text{ where } xs \odot w = h \cdot xs \odot w, \]

and $(\odot)$ satisfies that $xs \odot (ys \odot w) = ys \odot (xs \odot w)$ if $(\oplus)$ is associative, commutative, and has $z$ as identity. \[ \square \]

**Corollary 10.** $aggregate z (\odot) (\oplus) = return \cdot foldr (\odot) z \cdot concat$, provided that $(\oplus)$ is associative, commutative, and has $z$ as identity, and that $foldr (\odot) z = hom (\oplus) z$.

**Proof.** We reason:

\[
aggregate z (\odot) (\oplus)
= \{ \text{ Theorem 9 } \}
return \cdot foldr (\oplus) z \cdot map \left( foldr (\odot) z \right)
= \{ \text{ Lemma 6 } \}
return \cdot foldr (\odot) z \cdot concat.
\]

\[ \square \]
Lemma 11. If aggregate \( z \odot (\oplus) = \text{return} \cdot \text{foldr} \ (\odot) \cdot \text{concat} \), and shuffle! \( xss = \text{return} \ yss \| m \), we have

\[
\begin{align*}
\text{foldr} \ (\odot) \ z \ (\text{concat} \ xss) &= \\
\text{foldr} \ (\oplus) \ z \ (\text{map} \ (\text{foldr} \ (\odot) \ z) \ xss) &= \\
\text{foldr} \ (\oplus) \ z \ (\text{map} \ (\text{foldr} \ (\odot) \ z) \ yss) .
\end{align*}
\]

Proof. We assume the following two properties of MonadPlus:

1. \( m_1 \parallel m_2 = \text{return} \ x \) implies that \( m_1 = m_2 = \text{return} \ x \).
2. \( \text{return} \ x_1 = \text{return} \ x_2 \) implies that \( x_1 = x_2 \).

For our problem, if \( \text{shuffle}! \ xss = \text{return} \ yss \| m \), we have

\[
\text{return} \cdot \text{foldr} \ (\odot) \ z \cdot \text{concat} \ yss
\]

\[
\begin{align*}
\text{aggregate} \ z \ (\odot) \ (\oplus) \ xss &= \\
\begin{cases}
\text{assumption} \\
\text{calculation in the previous lemma}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\text{foldr} \ (\oplus) \ z \cdot \text{map} \ (\text{foldr} \ (\odot) \ z) \ yss \parallel \\
\text{((foldr} \ (\oplus) \ z \cdot \text{map} \ (\text{foldr} \ (\odot) \ z)) \ yss \| m).
\end{align*}
\]

\[\square\]

Theorem 12. If aggregate \( z \odot (\oplus) = \text{return} \cdot \text{foldr} \ (\odot) \ z \cdot \text{concat} \), we have that \( (\oplus) \), when restricted to values in \( \text{img} \ (\text{foldr} \ (\odot) \ z) \), is associative, commutative, and has \( z \) as identity.

Proof. In the discussion below, let \( x, y, \) and \( w \) be in \( \text{img} \ (\text{foldr} \ (\odot) \ z) \). That is, there exists \( xs, ys, \) and \( ws \) such that \( x = \text{foldr} \ (\odot) \ z \cdot xs, y = \text{foldr} \ (\odot) \ z \cdot ys, \) and \( w = \text{foldr} \ (\odot) \ z \cdot ws. \)

Identity. We reason:

\[
y = \text{foldr} \ (\odot) \ z \ (\text{concat} \ [xs])
\]

\[
\begin{align*}
\begin{cases}
\text{shuffle!} \ [xs] = \text{return} \ [xs] \| mzero, \text{Lemma 11} \\
\text{foldr} \ (\oplus) \ z \ (\text{map} \ (\text{foldr} \ (\odot) \ z) \ [xs])
\end{cases}
\end{align*}
\]

\[
y = y \oplus z .
\]

Thus \( z \) is a right identity of \( (\oplus) \).

\[
y = \text{foldr} \ (\odot) \ z \ (\text{concat} \ [[],xs])
\]

\[
\begin{align*}
\begin{cases}
\text{shuffle!} \ [ [],xs] = \text{return} \ [ [],xs] \| m, \text{Lemma 11} \\
\text{foldr} \ (\oplus) \ z \ (\text{map} \ (\text{foldr} \ (\odot) \ z) \ [ [],xs])
\end{cases}
\end{align*}
\]

\[
z \oplus (y \oplus z)
\]

\[
\begin{cases}
z \ is \ a \ right \ identity \ of \ (\oplus) \\
z \oplus y .
\end{cases}
\]

8
Thus \( z \) is also a left identity of \( (\oplus) \).

**Commutativity.** We reason:

\[
\begin{align*}
x \oplus y &= \{ \text{\( z \) is a right identity} \} \\
x \oplus (y \oplus z) &= \text{foldr} (\oplus) z (\text{map} (\text{foldr} \ (\otimes) \ z) [xs, ys]) \\
&= \{ \text{shuffle} [xs, ys] = \text{return} [ys, xs] \parallel m, \text{Lemma 11} \} \\
\text{foldr} (\oplus) z (\text{map} (\text{foldr} \ (\otimes) \ z) [ys, xs]) &= y \oplus (x \oplus z) \\
&= y \oplus x .
\end{align*}
\]

**Associativity.** We reason:

\[
\begin{align*}
x \oplus (y \oplus w) &= \{ \text{\( z \) is a right identity} \} \\
x \oplus (y \oplus (w \oplus z)) &= \text{foldr} (\oplus) z (\text{map} (\text{foldr} \ (\otimes) \ z) [xs, ys, ws]) \\
&= \{ \text{Lemma 11} \} \\
\text{foldr} (\oplus) z (\text{map} (\text{foldr} \ (\otimes) \ z) [ws, xs, ys]) &= w \oplus (x \oplus (y \oplus z)) \\
&= \{ \text{\( z \) is a right identity} \} \\
w \oplus (x \oplus y) &= \{ (\oplus) \text{ commutative} \} \\
(x \oplus y) \oplus w &.
\end{align*}
\]

\(\square\)

**Theorem 13.** If aggregate \( z (\otimes) (\oplus) = \text{return} \cdot \text{foldr} (\otimes) z \cdot \text{concat} \), we have \( \text{foldr} (\otimes) z = \text{hom} (\oplus) z \).

**Proof.** Apparently \( \text{foldr} (\otimes) z [] = \). We are left with proving the case for concatenation.

\[
\begin{align*}
\text{foldr} (\otimes) z (xs + ys) &= \text{foldr} (\otimes) z (\text{concat} [xs, ys]) \\
&= \{ \text{Lemma 11} \} \\
\text{foldr} (\oplus) z (\text{map} (\text{foldr} (\otimes) z) [xs, ys]) &= \text{foldr} (\otimes) z xs \oplus (\text{foldr} (\oplus) z ys \oplus z) \\
&= \{ \text{Theorem 12, \( z \) is identity} \} \\
\text{foldr} (\otimes) z xs \oplus (\text{foldr} (\oplus) z ys) .
\end{align*}
\]

\(\square\)

**Corollary 14.** Given \( (\otimes) : a \to b \to b \) and \( (\oplus) : b \to b \to b \). aggregate \( z (\otimes) (\oplus) = \text{return} \cdot \text{foldr} (\otimes) z \cdot \text{concat} \) if and only if \( (\text{img} \ (\text{foldr} (\otimes) z), (\oplus), z) \) forms a commutative monoid, and that \( \text{foldr} (\otimes) z = \text{hom} (\oplus) z \).

**Proof.** A conclusion following from Corollary 10, Theorem 12, and Theorem 13. \(\square\)
5 Tree Aggregate

Lemma 15. \( \text{apply} \, (\oplus) = \text{return} \cdot \text{foldr} \, (\oplus) \, z \) if \((\oplus)\) is associative, commutative, and has \(z\) as identity.

Proof. To do. \qed

Theorem 16. \( \text{treeAggregate} \, z \, (\otimes) \, (\oplus) \, = \text{return} \cdot \text{foldr} \, (\oplus) \, z \cdot \text{map} \, (\text{foldr} \, (\otimes) \, z) \) if \((\oplus)\) is associative, commutative, and has \(z\) as identity.

Proof. \[
\begin{align*}
\text{treeAggregate} \, z \, (\otimes) \, (\oplus) &= \{\text{definition of treeAggregate}\} \\
\text{apply} \, (\oplus) &\ll (\text{shuffle}! \cdot \text{map} \, (\text{foldr} \, (\otimes) \, z)) \\
&= \{\text{Lemma 4}\} \\
\text{apply} \, (\oplus) &\ll (\text{map} \, (\text{foldr} \, (\otimes) \, z) \circ \text{shuffle}!) \\
&= \{\text{Lemma 15}\} \\
(\text{return} \cdot \text{foldr} \, (\oplus) \, z \cdot \text{map} \, (\text{foldr} \, (\otimes) \, z)) &\ll \text{shuffle}! \\
&= \{\text{definitions of (\ll) and (\circ)}\} \\
(\text{foldr} \, (\oplus) \, z \cdot \text{map} \, (\text{foldr} \, (\otimes) \, z)) &\circ \text{shuffle}! \\
&= \{\text{Lemma 2}\} \\
\text{return} \cdot \text{foldr} \, (\oplus) \, z \cdot \text{map} \, (\text{foldr} \, (\otimes) \, z) .
\]

\qed

6 Aggregation by Key

For proofs, it is helpful to have an inductive definition of \((\hat{\circ}_z)\).

\[
\begin{align*}
(k, v) \hat{\circ}_z [] &= [(k, v \circ z)] \\
(k, v) \hat{\circ}_z ((j, u):xs) &= \\
| k = j &\Rightarrow (k, v \circ u):xs \\
| \text{otherwise} &\Rightarrow (j, u):(k, v) \hat{\circ}_z xs .
\end{align*}
\]

Define the following auxiliary functions:

\[
\begin{align*}
kEq k &= (k \equiv) \cdot \text{fst} , \\
\text{lookUp} :: \text{Eq} \, k \Rightarrow k \rightarrow a \rightarrow \text{PairRDD} \, k \rightarrow a \\
\text{lookUp} \, k \, z &= \text{hd} \, z \cdot \text{map} \, \text{snd} \cdot \text{filter} \, (k\text{Eq} \, k) \cdot \text{concat} , \\
\text{filterKRDD} :: \text{Eq} \, k \Rightarrow k \rightarrow \text{PairRDD} \, k \rightarrow a \rightarrow \text{PairRDD} \, k \rightarrow a \\
\text{filterKRDD} \, k &= \text{filter} \, (\neg \cdot \text{null}) \cdot \text{map} \, (\text{filter} \, (k\text{Eq} \, k)) .
\end{align*}
\]

Lemma 17. For all \(j, k, xs, v, z,\) and \((\circ)\), we have:

1. \(\text{filter} \, (k\text{Eq} \, k) \, ((k, v) \hat{\circ}_z xs) = (k, v) \hat{\circ}_z \text{filter} \, (k\text{Eq} \, k) \, xs .\)
2. \( \text{filter}(k \text{Eq } k) \cdot (j,v) \cdot \hat{\odot}_z \cdot \text{xs} = \text{filter}(k \text{Eq } k) \cdot \text{xs} \), if \( k \neq j \).

Lemma 18. For all \( k \), \( (\oplus) \), and \( z \), we have
\[
\text{filter}(k \text{Eq } k) \cdot \text{foldr} (=z) [] = \text{foldr} (=z) [] \cdot \text{filter}(k \text{Eq } k).
\]

Lemma 19. For all \( k \), \( (\odot) \), and \( z \), we have
\[
\text{foldr}(=z) [] \cdot (\text{filter}(k \text{Eq } k) \cdot \text{xs}) = \text{foldr}(=z) [] \cdot \text{filter}(k \text{Eq } k).
\]
if there exists at least one \((k,v)\) in \( \text{xs} \). Otherwise \( \text{foldr}(=z) [] \cdot (\text{filter}(k \text{Eq } k) \cdot \text{xs}) = [] \).

Corollary 20. As a corollary, we have that for all \( k \), \( (\odot) \), and \( z \),
\[
\text{hd } z \otimes \text{shuffle} \cdot \text{map } \text{value} \cdot \text{foldr}(=z) [] \cdot \text{filter}(k \text{Eq } k) = \text{return} \cdot \text{foldr} \cdot \text{dot } z \cdot \text{map } \text{value} \cdot \text{filter}(k \text{Eq } k).
\]

It follows from Lemma 19 because, when the input is empty or does not contain entries with key \( k \), both sides reduce to \( \text{return } z \).

Corollary 21. For all \( k \), \( z \) and \( (\odot) \) we have:
\[
\text{concat} \cdot \text{map } (\text{map } \text{value} \cdot \text{filter}(k \text{Eq } k) \cdot \text{foldr}(=z) []) = \text{map } (\text{foldr}(=z) z) \cdot \text{filterKRDD } k.
\]

Theorem 22. If \( ..., \) we have
\[
\text{lookup } k z \otimes \text{aggregateByKey}_z (\otimes) (\oplus) = \text{return} \cdot \text{foldr}(=z) z \cdot \text{map } (\text{foldr}(=z) z) \cdot \text{filterKRDD } k.
\]

Proof. We reason:
\[
\begin{align*}
\text{lookup } k z \otimes \text{aggregateByKey}_z (\otimes) (\oplus) &= \{ \text{definitions} \} \\
&= (\text{hd } z \cdot \text{map } \text{snd} \cdot \text{filter}(k \text{Eq } k) \cdot \text{concat}) \cdot \text{repartition}! \ll (\text{foldr}(=z) []) \cdot \text{concat} \cdot \text{map}! (\text{foldr}(=z) []) \\
&= \{ \text{concat} \cdot \text{part} = \text{return} \} \\
&= (\text{hd } z \cdot \text{map } \text{snd} \cdot \text{filter}(k \text{Eq } k) \cdot \text{shuffle}! \ll (\text{foldr}(=z) []) \cdot \text{concat} \cdot \text{map}! (\text{foldr}(=z) []) \\
&= \{ \text{Lemma 4 and 5} \} \\
&= \text{hd } z \otimes \text{shuffle}! \ll (\text{map } \text{snd} \cdot \text{filter}(k \text{Eq } k) \cdot \text{foldr}(=z) []) \cdot \text{concat} \cdot \text{map}! (\text{foldr}(=z) []) \\
&= \{ \text{Lemma 4} \} \\
&= \text{hd } z \otimes \text{shuffle}! \ll (\text{map } \text{snd} \cdot \text{filter}(k \text{Eq } k) \cdot \text{foldr}(=z) []) \cdot \text{concat} \cdot \text{map}! (\text{foldr}(=z) []) \otimes \text{shuffle}! \\
&= \{ \text{by (8) and (9)} \} \\
&= (\text{hd } z \otimes \text{shuffle}! \cdot \text{map } \text{snd} \cdot \text{filter}(k \text{Eq } k) \cdot \text{foldr}(=z) []) \cdot \text{concat} \cdot \text{map}! (\text{foldr}(=z) []) \ll \text{shuffle}!.
\end{align*}
\]
We work on the part before the right-most shuffle!:

\[ \text{hd } z \leadsto \text{shuffle!} \cdot \text{map } \text{snd} \cdot \text{filter } (k\text{Eq } k) \cdot \text{foldr } (\triangle) \cdot \text{concat} \cdot \text{map } (\text{foldr } (\triangle)) \cdot [ ] \]

= \{ \text{Lemma 18} \}

\[ \text{hd } z \leadsto \text{shuffle!} \cdot \text{map } \text{snd} \cdot \text{foldr } (\triangle) \cdot \text{filter } (k\text{Eq } k) \cdot \text{concat} \cdot \text{map } (\text{foldr } (\triangle)) \cdot [ ] \]

= \{ \text{Corollary 20} \}

\[ \text{return } \cdot \text{foldr } (\oplus) \cdot \text{map } \text{snd} \cdot \text{filter } (k\text{Eq } k) \cdot \text{concat} \cdot \text{map } (\text{foldr } (\triangle)) \cdot [ ] \]

= \{ \text{naturality} \}

\[ \text{return } \cdot \text{foldr } (\oplus) \cdot \text{map } \text{snd} \cdot \text{filter } (k\text{Eq } k) \cdot \text{foldr } (\triangle) \cdot [ ] \]

= \{ \text{Corollary 21} \}

\[ \text{return } \cdot \text{foldr } (\oplus) \cdot \text{concat} \cdot \text{map } (\text{foldr } (\otimes) \cdot \text{filterKRDD } k) . \]

Back to the main proof:

\[ \text{lookUp } k z \leadsto \text{aggregateByKey } z (\otimes) (\oplus) \]

= \{ \text{calculation above} \}

\[ (\text{return } \cdot \text{foldr } (\oplus) \cdot \text{map } (\text{foldr } (\otimes) \cdot \text{filter } (\neg \cdot \text{null}) \cdot \text{map } (\text{filter } (k\text{Eq } k))) = (\text{shuffle!}) \]

= \{ \text{definitions of } (\triangleleft\triangleleft) \text{ and } (\triangleleft) \}

\[ (\text{foldr } (\oplus) \cdot \text{map } (\text{foldr } (\otimes) \cdot \text{filter } (\neg \cdot \text{null}) \cdot \text{map } (\text{filter } (k\text{Eq } k))) = (\text{shuffle!}) \]

= \{ \text{Lemma 4 and 5} \}

\[ (\text{foldr } (\oplus) \cdot \text{map } (\text{foldr } (\otimes)) \cdot \text{filterKRDD } k) . \]

= \{ \text{Lemma 2} \}

\[ \text{return } \cdot \text{foldr } (\oplus) \cdot \text{map } (\text{foldr } (\otimes) \cdot \text{filterKRDD } k) . \]

\[ \text{lookUp } k z \leadsto (\text{aggregateMessagesWithActiveVertices } \text{sendMsg } (\oplus) \cdot \text{active } (\text{Graph } v\text{Rdd } e\text{Rdd})) \]

= \{ \text{definitions} \}

\[ \text{lookUp } k z \leadsto \text{reduceByKey } (\oplus) \cdot (\text{map } (\text{concatMap } \text{sendIfActive}) \cdot e\text{Rdd}) \]

= \{ \}

\[ \text{return } \cdot \text{foldl } (\oplus) \cdot \text{concat} \cdot \text{filterKRDD } k \cdot \text{map } (\text{concatMap } \text{sendIfActive}) \cdot e\text{Rdd} \]

= \{ \text{naturality laws} \}

\[ \text{return } \cdot \text{foldl } (\oplus) \cdot \text{concatMap } \text{sendIfActive} \cdot \text{map } \text{value} \cdot \text{filter } (k\text{Eq } k) \cdot \text{concat} \cdot e\text{Rdd} . \]

7 Proofs of monadic properties

Proving (5) \( f \triangleleft\triangleleft (g \cdot m) = (f \cdot g) \triangleleft\triangleleft m. \)

Proof. We reason:

\[ f \triangleleft\triangleleft (g \cdot m) \]

= \{ definition of \((\cdot \cdot)\) \}

\[ f \triangleleft\triangleleft (g \cdot m) \]

= \{ definition of \((\cdot \cdot)\) \}

12
\[
f = \lll (\lll \cdot g) = m \\
\triangleq \{ \text{monadic law (3)} \} \\
(\lambda x \to f = \lll \cdot (\lll g x)) = m \\
\triangleq \{ \text{monadic law (1)} \} \\
(\lambda x \to f (g x)) = m \\
= (f \cdot g) = m.
\]

**Proving (6) \( f \comp (g \comp m) = (f \cdot g) \comp m. \)**

**Proof.** We reason:

\[
f \comp (g \comp m) \\
\triangleq \{ \text{definition of } (\comp) \} \\
(\lll f) = \lll (g \comp m) \\
\triangleq \{ \text{by (5)} \} \\
(\lll f \cdot g) = m \\
\triangleq \{ \text{definition of } (\comp) \} \\
(f \cdot g) = m.
\]

For the next results we prove a lemma:

\[
(f \lll) \cdot (g \lll) = ((f \lll) \cdot g) \lll . 
\]

**Proving (7) \( f \cdot (g \cdot m) = (f \cdot g) \cdot m. \)**

**Proof.** We reason:

\[
f \cdot (g \cdot m) \\
\triangleq \{ \text{definition of } (\cdot) \} \\
(((\lll f) \cdot (\lll g)) \cdot m \\
\triangleq \{ \text{by (11)} \} \\
(((\lll f) \cdot \lll g) \cdot m \\
\triangleq \{ \text{monadic law (1)} \} \\
(\lll f \cdot g) \cdot m \\
\triangleq \{ \text{definition of } (\cdot) \} \\
(f \cdot g) \cdot m.
\]

\[
\square
\]
Proving (8)  \( f \ll (g \cdot h) = (f \cdot g) \ll h \).

Proof. We reason:

\[
\begin{align*}
f \ll (g \cdot h) \\
&= \{ \text{definitions of } (\ll) \} \\
&= (f \ll) \cdot ((return \cdot g) \ll) \cdot h \\
&= \{ \text{by (11)} \} \\
&= (((f \ll) \cdot return \cdot g) \ll) \cdot h \\
&= \{ \text{monadic law (1)} \} \\
&= ((f \cdot g) \ll) \cdot h \\
&= \{ \text{definition of } (\ll) \} \\
&= (f \cdot g) \ll h .
\end{align*}
\]

Proving (9)  \( f \circ (g \ll h) = (f \circ g) \ll h \).

Proof. We reason:

\[
\begin{align*}
f \circ (g \ll h) \\
&= \{ \text{definitions of } (\ll) \text{ and } (\circ) \} \\
&= ((return \cdot f) \ll) \cdot (g \ll) \cdot h \\
&= \{ \text{by (11)} \} \\
&= (((return \cdot f) \ll) \cdot g) \ll) \cdot h \\
&= \{ \text{definition of } (\circ) \} \\
&= ((f \circ g) \ll) \cdot h \\
&= \{ \text{definition of } (\ll) \} \\
&= (f \circ g) \ll h .
\end{align*}
\]