Simple aggregation Recall the definitions:

$$\textbf{type}\ \mathsf{Partition\ a} = [\mathsf{a}]$$
$$\textbf{type}\ \mathsf{RDD\ a} = [\mathsf{Partition\ a}]$$

$$\mathsf{aggregate} :: \mathsf{b} \to (\mathsf{b} \to \mathsf{a} \to \mathsf{b}) \to (\mathsf{b} \to \mathsf{b} \to \mathsf{b}) \to \mathsf{RDD\ a} \to \mathsf{b}$$
$$\mathsf{aggregate\ z}\ (\otimes)\ (\oplus) = \mathsf{foldl}\ (\oplus)\ \mathsf{z} \circ \mathsf{map}\ (\mathsf{foldl}\ (\otimes)\ \mathsf{z})$$

Aggregation for pair RDDs Extending aggregation to pair RDDs:

$$\textbf{type}\ \mathsf{PairRDD\ a\ b} = \mathsf{RDD}\ (\mathsf{a}, \mathsf{b})$$

$$\mathsf{key} :: (\mathsf{a}, \mathsf{b}) \to \mathsf{a}$$
$$\mathsf{key} = \mathsf{fst}$$
$$\mathsf{value} :: (\mathsf{a}, \mathsf{b}) \to \mathsf{b}$$
$$\mathsf{value} = \mathsf{snd}$$

The functional call $\mathsf{hasValye\ k\ v\ ps}$ locates the rightmost occurrence of (k, u) in ps, if any, and returns u. Otherwise it returns the default value v.

$$\mathsf{hasValue} :: \mathsf{Eq\ a} \Rightarrow \mathsf{a} \to \mathsf{b} \to [(\mathsf{a}, \mathsf{b})] \to \mathsf{b}$$
$$\mathsf{hasValue\ k\ v\ ps} = \mathsf{last}_{\mathsf{v}}\ (\mathsf{map\ value}\ (\mathsf{filter}\ ((\equiv \mathsf{k}) \circ \mathsf{key})\ \mathsf{ps}))$$

where $\mathsf{last}_{\mathsf{v}} = \mathsf{last} \circ (\mathsf{v}{:})$ (thus $\mathsf{last}_{\mathsf{v}}\ [\,] = \mathsf{v}$). The definition above is equivalent to $\mathsf{foldl}\ (\lambda \mathsf{r\ p} \to \textbf{if}\ \mathsf{k} \equiv \mathsf{key\ p}\ \textbf{then}\ \mathsf{value\ p}\ \textbf{else}\ \mathsf{r})\ \mathsf{v}$. Also, let $\mathsf{keyEq\ k} = (\equiv \mathsf{k}) \circ \mathsf{key}$, since we use it often

The function addTo updates a key-value list by a given key-value pair.

$$\mathsf{addTo} :: \mathsf{Eq\ a} \Rightarrow \mathsf{a} \to \mathsf{b} \to [(\mathsf{a}, \mathsf{b})] \to [(\mathsf{a}, \mathsf{b})]$$
$$\mathsf{addTo\ k\ v\ ps} = (\mathsf{k}, \mathsf{v}) : \mathsf{filter}\ (\neg \circ (\equiv \mathsf{k}) \circ \mathsf{key})\ \mathsf{ps}$$

Given $(\odot) :: \mathsf{c} \to \mathsf{b} \to \mathsf{c}$, $(\hat{\odot}_z) :: [(\mathsf{a}, \mathsf{c})] \to (\mathsf{a}, \mathsf{b}) \to [(\mathsf{a}, \mathsf{c})]$ is $(\odot)$ lifted to list of key-value pairs:

$$\hat{\cdot}. :: \mathsf{Eq\ a} \Rightarrow (\mathsf{c} \to \mathsf{b} \to \mathsf{c}) \to \mathsf{c} \to [(\mathsf{a}, \mathsf{c})] \to (\mathsf{a}, \mathsf{b}) \to [(\mathsf{a}, \mathsf{c})]$$
$$\mathsf{r}\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) = \mathsf{addTo\ k}\ (\mathsf{hasValue\ k\ z\ r} \odot \mathsf{v})\ \mathsf{r}$$

It is helpful to give $(\hat{\odot}_z)$ an inductive characterization:

$$\begin{array}{lll}
[\,] & \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) = [(\mathsf{k}, \mathsf{z} \odot \mathsf{v})] & \\
(\mathsf{xs} +\!\!\!+ [(\mathsf{j}, \mathsf{u})]) & \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) \mid \mathsf{k} \equiv \mathsf{j} & = \mathsf{xs} +\!\!\!+ [(\mathsf{k}, \mathsf{u} \odot \mathsf{v})] \\
 & \qquad\qquad \mid \mathsf{otherwise} & = (\mathsf{xs}\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v})) +\!\!\!+ [(\mathsf{j}, \mathsf{u})]
\end{array}$$

The function $\mathsf{repartition}$ is unspecified. The following dummy definition is merely there to allow the code to type check.

$$\begin{aligned}&\mathsf{repartition} :: [(\mathsf{a},\mathsf{b})] \to \mathsf{PairRDD\ a\ b}\\&\mathsf{repartition\ xs} = [\mathsf{xs}]\end{aligned}$$

Finally, aggregateByKey may be defined by:

$$\begin{aligned}&\mathsf{aggregateByKey} :: \mathsf{Eq\ a} \Rightarrow \mathsf{c} \to (\mathsf{c} \to \mathsf{b} \to \mathsf{c}) \to (\mathsf{c} \to \mathsf{c} \to \mathsf{c}) \to\\&\qquad\qquad\qquad\qquad \mathsf{PairRDD\ a\ b} \to \mathsf{PairRDD\ a\ c}\\&\mathsf{aggregateByKey\ z}\ (\otimes)\ (\oplus) =\\&\quad \mathsf{repartition} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{foldl}\ (\hat{\otimes}_z)\ [\,])\end{aligned}$$

Aggregation with a given key The function aggregateWithKey

$$\begin{aligned}&\mathsf{aggregateWithKey} :: \mathsf{Eq\ a} \Rightarrow\\&\quad \mathsf{a} \to \mathsf{c} \to (\mathsf{c} \to \mathsf{b} \to \mathsf{c}) \to (\mathsf{c} \to \mathsf{c} \to \mathsf{c}) \to \mathsf{PairRDD\ a\ b} \to \mathsf{c}\\&\mathsf{aggregateWithKey\ k\ z}\ (\otimes)\ (\oplus) =\\&\quad \mathsf{aggregate\ z}\ (\otimes)\ (\oplus) \circ\\&\qquad \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ\\&\qquad\quad \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\end{aligned}$$

$$\begin{aligned}&\mathsf{lookUp} :: \mathsf{Eq\ a} \Rightarrow \mathsf{a} \to \mathsf{b} \to \mathsf{PairRDD\ a\ b} \to \mathsf{b}\\&\mathsf{lookUp\ k\ z} = \mathsf{last_z} \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ ((\equiv \mathsf{k}) \circ \mathsf{key}))\end{aligned}$$

$$\begin{aligned}&\mathsf{lookUpL} :: \mathsf{Eq\ a} \Rightarrow \mathsf{a} \to \mathsf{b} \to [(\mathsf{a},\mathsf{b})] \to \mathsf{b}\\&\mathsf{lookUpL\ k\ z} = \mathsf{last_z} \circ \mathsf{map\ value} \circ \mathsf{filter}\ ((\equiv \mathsf{k}) \circ \mathsf{key})\end{aligned}$$

Lemma 1. *For all* j, k, xs, v, z, *and* $(\odot)$, *we have:*

1. $\mathsf{filter\ (keyEq\ k)\ (xs}\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})) = \mathsf{filter\ (keyEq\ k)\ xs}\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})$.
2. $\mathsf{filter\ (keyEq\ k)\ (xs}\ \hat{\odot}_z\ (\mathsf{j},\mathsf{v})) = \mathsf{filter\ (keyEq\ k)\ xs}$, *if* $\mathsf{k} \not\equiv \mathsf{j}$.

Proof. We prove 1. only. **Case** $\mathsf{xs} := [\,]$.

$$\begin{aligned}&\ \ \mathsf{filter\ (keyEq\ k)}\ ([\,]\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}))\\&= \mathsf{filter\ (keyEq\ k)}\ [(\mathsf{k}, \mathsf{z} \odot \mathsf{v})]\\&= [(\mathsf{k}, \mathsf{z} \odot \mathsf{v})]\\&= [\,]\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})\\&= \mathsf{filter\ (keyEq\ k)}\ [\,]\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}) \quad .\end{aligned}$$

Case $\mathsf{xs} := \mathsf{xs} +\!\!+ [(\mathsf{k},\mathsf{u})]$.

$$\begin{aligned}&\ \ \mathsf{filter\ (keyEq\ k)}\ ((\mathsf{xs} +\!\!+ [(\mathsf{k},\mathsf{u})])\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}))\\&= \quad \{\text{ inductive definition of } (\hat{\odot}_z) \}\\&\ \ \mathsf{filter\ (keyEq\ k)}\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{u} \odot \mathsf{v})])\\&= \mathsf{filter\ (keyEq\ k)\ xs} +\!\!+ [(\mathsf{k}, \mathsf{u} \odot \mathsf{v})]\end{aligned}$$

$$
\begin{aligned}
&= \quad \{\text{ inductive definition of } (\hat{\odot}_z) \ \}\\
&\quad (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs} \mathbin{+\!\!+} [(\mathsf{k},\mathsf{u})])\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{k},\mathsf{u})])\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}) \quad .
\end{aligned}
$$

Case $\mathsf{xs} := \mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{u})]$, where $\mathsf{j} \not\equiv \mathsf{k}$.

$$
\begin{aligned}
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ ((\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{u})])\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}))\\
&= \quad \{\text{ inductive definition of } (\hat{\odot}_z) \ \}\\
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ ((\mathsf{xs}\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})) \mathbin{+\!\!+} [(\mathsf{j},\mathsf{u})])\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs}\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}))\\
&= \quad \{\text{ induction }\}\\
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{l})\ \mathsf{xs}\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v})\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{u})])\ \hat{\odot}_z\ (\mathsf{k},\mathsf{v}) \quad .
\end{aligned}
$$

□

Lemma 2. *For all* k, $(\oplus)$, *and* z, *we have*

$$
\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k}) \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] = \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k}) \quad .
$$

Proof. Induction on the input.
Case $\mathsf{xs} := [\,]$. Both sides reduces to $[\,]$.
Case $\mathsf{xs} := \mathsf{xs} \mathbin{+\!\!+} [(\mathsf{k},\mathsf{v})]$

$$
\begin{aligned}
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{k},\mathsf{v})]))\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ \mathsf{xs}\ \hat{\oplus}_z\ (\mathsf{k},\mathsf{v}))\\
&= \quad \{\text{ Lemma 1 }\}\\
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ \mathsf{xs})\ \hat{\oplus}_z\ (\mathsf{k},\mathsf{v})\\
&= \quad \{\text{ induction }\}\\
&\quad \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs})\ \hat{\oplus}_z\ (\mathsf{k},\mathsf{v})\\
&= \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{k},\mathsf{v})]))
\end{aligned}
$$

Case $\mathsf{xs} := \mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{v})]$, where $\mathsf{j} \not\equiv \mathsf{k}$.

$$
\begin{aligned}
&\quad \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{v})]))\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ \mathsf{xs}\ \hat{\oplus}_z\ (\mathsf{j},\mathsf{v}))\\
&= \quad \{\text{ Lemma 1 }\}\\
&= \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ \mathsf{xs})\\
&= \quad \{\text{ induction }\}\\
&\quad \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs})\\
&= \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} \mathbin{+\!\!+} [(\mathsf{j},\mathsf{v})]))
\end{aligned}
$$

□

Lemma 3. *For all* k*,* $(\odot)$*, and* z*, we have*

$$\begin{array}{l}\mathsf{foldl}\ (\hat{\odot}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs} = \\ \quad [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs})]\end{array}$$

if there exists at least one (k, v) *in* xs*. Otherwise* $\mathsf{foldl}\ (\hat{\odot}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs} = [\,]$.

Proof. The "otherwise" part is trivially true. For the first part, we perform an induction on the input.
Case $\mathsf{xs} := [(\mathsf{k}, \mathsf{v})]$. Both sides reduce to $[(\mathsf{k}, \mathsf{z} \odot \mathsf{v})]$.
Case $\mathsf{xs} := \mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]$.

- **Case** there exists at least one $[(\mathsf{k}, \mathsf{u})]$ in xs.

$$\begin{array}{ll} & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})])) \\ = & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]) \\ = & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs})\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) \\ = & \quad \{\ \text{induction}\ \} \\ & [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs})]\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) \\ = & \quad \{\ \text{definition of } (\hat{\oplus}_z)\ \} \\ & [(\mathsf{k}, (\mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs}) \odot \mathsf{v})] \\ = & [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]))] \quad . \end{array}$$

- **Case** $[(\mathsf{k}, \mathsf{u})]$ does not occur in xs.

$$\begin{array}{ll} & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})])) \\ = & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]) \\ = & \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs})\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) \\ = & \quad \{\ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs} = [\,]\ \} \\ = & [\,]\ \hat{\odot}_z\ (\mathsf{k}, \mathsf{v}) \\ = & [(\mathsf{k}, \mathsf{z} \odot \mathsf{v})] \\ = & [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]))] \quad . \end{array}$$

Case $\mathsf{xs} := \mathsf{xs} +\!\!+ [(\mathsf{j}, \mathsf{v})]$ where $\mathsf{j} \not\equiv \mathsf{k}$. In this case there must exists some $[(\mathsf{k}, \mathsf{u})]$ in xs.

$$\begin{array}{ll} & \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ (\mathsf{xs} +\!\!+ [(\mathsf{j}, \mathsf{v})])) \\ = & \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,]\ (\mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \mathsf{xs}) \\ = & \quad \{\ \text{induction}\ \} \\ & [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ \mathsf{xs})] \\ = & [(\mathsf{k}, \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map}\ \mathsf{value} \circ \mathsf{filter}\ (\mathsf{keyEq}\ \mathsf{k})\ \$\ (\mathsf{xs} +\!\!+ [(\mathsf{k}, \mathsf{v})]))] \quad . \end{array}$$

$\square$

Corollary 4. *As a corollary, we have that for all* k, $(\odot)$, *and* z,

$$\begin{aligned}&\mathsf{last\_z} \circ \mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\odot}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) = \\ &\quad \mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \quad .\end{aligned}$$

When the input is empty or does not contain entries having key k, *both sides reduce to* z.

Corollary 5. *For all* k, z *and* $(\odot)$ *we have:*

$$\begin{aligned}&\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \mathsf{foldl}\ (\hat{\odot}_z)\ [\,]) = \\ &\quad \mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map\ filter}\ (\mathsf{keyEq\ k}) \quad .\end{aligned}$$

Proof. We reason:

$$\begin{aligned}&\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\ = &\quad \{\ \text{Lemma 2}\ \} \\ &\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\end{aligned}$$

Let the input be xss. Induction.
Case $\mathsf{xss} := [\,]$. Both sides reduce to $[\,]$
Case $\mathsf{xss} := \mathsf{xss} +\!\!+ [\mathsf{xs}]$

- there exists at least one (k, v) in xs

$$\begin{aligned}&\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ (\mathsf{xss} +\!\!+ [\mathsf{xs}]) \\ = &(\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ \mathsf{xss}) +\!\!+ \\ &\quad (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z) \circ \mathsf{filter}\ (\mathsf{keyEq\ k})\ \$\ \mathsf{xs}) \\ = &\quad \{\ \text{induction}\ \} \\ &(\mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ \mathsf{xss}) +\!\!+ \\ &\quad (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z) \circ \mathsf{filter}\ (\mathsf{keyEq\ k})\ \$\ \mathsf{xs}) \\ = &\quad \{\ \text{Lemma 3}\ \} \\ &(\mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ \mathsf{xss}) +\!\!+ \\ &\quad [\mathsf{foldl}\ (\odot)\ \mathsf{z} \circ \mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k})\ \$\ \mathsf{xs}] \\ = &\ \mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ (\mathsf{xss} +\!\!+ [\mathsf{xs}])\end{aligned}$$

- there exists no (k, v) in xs.

$$\begin{aligned}&\mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ (\mathsf{xss} +\!\!+ [\mathsf{xs}]) \\ = &\quad \{\ \text{same as above}\ \} \\ &(\mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ \mathsf{xss}) +\!\!+ \\ &\quad (\mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z) \circ \mathsf{filter}\ (\mathsf{keyEq\ k})\ \$\ \mathsf{xs}) \\ = &\quad \{\ \text{Lemma 3}\ \} \\ &\mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ \mathsf{xss} \\ = &\ \mathsf{map}\ (\mathsf{foldl}\ (\odot)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}))\ \$\ (\mathsf{xss} +\!\!+ [\mathsf{xs}]) \quad .\end{aligned}$$

$\square$

Finally, the main theorem:

Theorem 6. *For all* k, z, $(\otimes)$ *and* $(\oplus)$, *we have:*

$$\mathsf{lookUp\ k} \circ \mathsf{aggregateByKey\ z}\ (\otimes)\ (\oplus) = \mathsf{aggregateWithKey\ k\ z}\ (\otimes)\ (\oplus) \quad .$$

Proof. We reason:

$$\begin{aligned}
& \mathsf{lookUp\ k} \circ \mathsf{aggregateByKey\ z}\ (\otimes)\ (\oplus) \\
= \quad & \{ \text{ definition of } \mathsf{lookUp} \text{ and } \mathsf{aggregateByKey} \} \\
& \mathsf{last_z} \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k})) \circ \\
& \quad \mathsf{repartition} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\
= \quad & \{ \text{ routine naturalty laws. See below. } \} \\
& \mathsf{last_z} \circ \mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \\
& \quad \mathsf{concat} \circ \mathsf{map}\ (\mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\
= \quad & \{ \text{ Lemma 2 } \} \\
& \mathsf{last_z} \circ \mathsf{map\ value} \circ \mathsf{foldl}\ (\hat{\oplus}_z)\ [\,] \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \\
& \quad \mathsf{concat} \circ \mathsf{map}\ (\mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\
= \quad & \{ \text{ corollary of Lemma 3 } \} \\
& \mathsf{foldl}\ (\oplus)\ \mathsf{z} \circ \mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\
= \quad & \{ \text{ naturalty, } \mathsf{filter\ p} \circ \mathsf{concat} = \mathsf{concat} \circ \mathsf{map}\ (\mathsf{filter\ p}) \} \\
& \mathsf{foldl}\ (\oplus)\ \mathsf{z} \circ \mathsf{concat} \circ \mathsf{map}\ (\mathsf{map\ value} \circ \mathsf{filter}\ (\mathsf{keyEq\ k}) \circ \mathsf{foldl}\ (\hat{\otimes}_z)\ [\,]) \\
= \quad & \{ \text{ Corollary 5 } \} \\
& \mathsf{foldl}\ (\oplus)\ \mathsf{z} \circ \mathsf{map}\ (\mathsf{foldl}\ (\otimes)\ \mathsf{z}) \circ \mathsf{filter}\ (\neg \circ \mathsf{null}) \circ \mathsf{map\ filter}\ (\mathsf{keyEq\ k}) \\
= \quad & \{ \text{ definition } \} \\
& \mathsf{aggregateWithKey\ k\ z}\ (\otimes)\ (\oplus) \quad .
\end{aligned}$$

The "routine naturalty laws" in the second step we need are:

- $\mathsf{map}\ (\mathsf{filter\ p}) \circ \mathsf{repartition} = \mathsf{repartition} \circ \mathsf{filter\ p}$,
- $\mathsf{map}\ (\mathsf{map\ f}) \circ \mathsf{repartition} = \mathsf{repartition} \circ \mathsf{map\ f}$,
- $\mathsf{concat} \circ \mathsf{partition} = \mathsf{id}$.

$\square$