[**(September 9, 2010) Correction to**: Hadoop++: Making a Yellow Elephant Run Like a Cheetah (Without It Even Noticing). J. Dittrich et al., PVLDB'10.]

In Secion 4.1 the UDFs were incorrectly described for data co-partitioning. They should be replaced with the following UDFs:

$$CoPartition_{a_i,b_j}(T,S) \Mapsto \begin{cases} \texttt{map}(\text{key } k, \text{value } v) \mapsto \\ \begin{cases} [(\texttt{prj}_{a_i}(k \oplus v), k \oplus v)] & \text{if input}(k \oplus v) = T, \\ [(\texttt{prj}_{b_j}(k \oplus v), k \oplus v)] & \text{if input}(k \oplus v) = S. \end{cases} \\ \texttt{reduce}(\text{key } ik, \text{vset } ivs) \mapsto [(\{ik\} \times ivs)] \end{cases}$$

For each record in an input split, `itemize`.next() receives the `offset` as key and the record as value and `map` emits {`joinvalue, record`} as key-value pairs. For re-partitioning, sorting, and grouping the key-value pairs we use the entire key, i.e. we use the default `sh`, `cmp`, and `grp` UDFs. Figure 3(b) should be changed to show the Map Phase outputting {`joinvalue,record`} accordingly.