Weighted citation index

Maarten Fokkinga

Version of February 26, 2013, 15:40

Abstract. A citation index is one way of the many possibilities to measure "the quality" of a researcher: count the citations to his work. The notion of citation index has many derived concepts, one of which is the *h-index*. In this note we consider one particular wish for a new definition of citation index of an author: when counting citations to the author's work, weigh (= multiply) each citation by the citation index of the citing author. We show how the weighted citation index can be computed in the MapReduce framework.

(1) Introduction. A citation index is one way of the many (sometimes contradictory and often unsatisfactory) possibilities to measure "the quality" of a researcher: count the citations to his work [6]. The notion of citation index has many variants and derived concepts, one of which is the frequently used *h-index*. The `H-index` page of Wikipedia [7] nicely describes the concept, discusses the weaknesses and strengths (and ways to cheat), and mentions related, relevant, and interesting literature. In this note we explore one particular wish, *weighing*, that is applicable to the basic citation index, and hence to every derived concept: don't treat all citations on equal footing, but weigh each citation by the citation index of the citing author (thus making the definition of citation index circular or recursive).

The kind of weighing that we propose, is *exactly* the same kind of weighing as done in ranking web pages by the Page Rank algorithm [1] and in ranking scientific journals [3], and is motivated by the belief that *citations from a high quality researcher are more valuable than those from a low quality researcher*. As far as we know, weighing has not been applied to the notion of citation index of authors, even though the Page Rank was originally motivated and introduced as a kind of weighted citation count [5]. We will show that the computation of such an index can be done easily using the MapReduce framework [2, 4]. Such a computation is feasible for a global authority, but the correctness of the resulting index cannot be easily checked by individuals. The latter observation might be a practical reason not to use weighing when a kind of citation index is used in decisions about a researcher's career.

(2) Basic citation index. Before explaining weighing, we recall the basic index on which other variants are built:

$$\mathit{index}\ a \quad = \quad \sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ (\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ 1)$$

Variables a and p range over authors and publications, respectively. The inner summation counts the citations to p; these counts are summed in the outer summation for publications of a. Self-citations can be excluded by an additional restriction on p', namely: $\neg\ (a\ \underline{\mathit{writes}}\ p')$. By taking a suitable fraction of the inner summation, it can be arranged that an author gets

only $\frac{1}{m}$ of the citations to p when p has m authors. After parameterizing *index* by the set from which the publications p are drawn, we can use it to express the h-index:

$$\begin{array}{lcl} \mathit{index}_{ps}\ a & = & \sum_{p \in ps \mid a\ \underline{\mathit{writes}}\ p}\ (\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ 1) \\ \mathit{h\text{-}index}\ a & = & \max_n\ (\exists_{ps \mid \#ps=n}\ \ \mathit{index}_{ps}\ a \geq n) \end{array}$$

We do not further elaborate the basic citation index and its derived variants, but focus on *weighing each count* by its author's index.

(3) Weighing. For the basic citation index we weigh (= multiply) each 'citation count 1' by the maximal index of the citing authors — and instead of the maximum we can also take the summation, the average, or any other aggregate:

$$\begin{array}{lcl} \mathit{index}\ a & = & \sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \Big(\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ (\max_{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')\Big) \\ \mathit{index}\ a & = & \sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \Big(\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ (\ \sum\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')\Big) \\ \mathit{index}\ a & = & \sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \Big(\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ (\ \mathrm{avg}\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')\Big) \\ \mathit{index}\ a & = & \sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \Big(\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ (\ \mathrm{agg}\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')\Big) \end{array}$$

In fact, the specification is now merely an equation with unknown *index*. We do not consider conditions under which a solution, or even a unique particular solution, exists. However, by *damping* (with some global constant d) we preclude the trivial solution that assigns each author the number 0:

$$\begin{array}{lcl} \mathit{damp}\ x & = & (1-d) + d \times x \\ \mathit{index}\ a & = & \mathit{damp}\ \Big(\sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \ \Big(\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ \ (\ \mathrm{agg}\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')\Big)\Big) \end{array}$$

Naturally, additional weighing factors suggest themselves. For example, believing that *the more publications are cited from within p', the less a single citation counts*, we weigh each 'aggregated citation count' by $\frac{1}{n}$, where n is the number of citations occurring in p':

$$\mathit{index}\ a \quad = \quad \mathit{damp}\ (\sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ (\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ \ \tfrac{1}{\#(\text{citations } p')} \times (\mathrm{agg}\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a')))$$

We can also normalize *index* so as to yield percentages rather than unbounded numbers:

$$\begin{array}{lcl} \mathit{index}\ a & = & \frac{1}{T} \times \mathit{damp}\ (\sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ \ \tfrac{1}{\#(\text{citations } p')} \times (\mathrm{agg}_{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \mathit{index}\ a')) \\ T & = & \sum_a\ \mathit{index}\ a \end{array}$$

Abstracting from particular choices, the weighted citation index has the following specification:

$$(4) \quad \mathit{index}\ a \quad = \quad f\ (\sum_{p \mid a\ \underline{\mathit{writes}}\ p}\ \ G\ (\sum_{p' \mid p'\ \underline{\mathit{cites}}\ p}\ \ H\ (\mathrm{agg}\ _{a' \mid a'\ \underline{\mathit{writes}}\ p'}\ \ \mathit{index}\ a'))) \quad ,$$

where f is a given function, and expressions G, H denote functions that may depend on p, and on p, p', respectively. Interestingly, this formula is has a three nested aggregates, whereas page Rank has only one; see Appendix (8).

(5) MapReduce computation. We assume that the intended solution for unknown *index* in (4) can be approximated to arbitrary degree with an iterative approach. That is, taking a "reasonable" $index_0$ as starting point, solution *index* is the limit of $index_0$, $index_1$, $index_2$, ... where $index_{i+1}$ is expressed in terms of $index_i$ according to specification (4):

$$(6) \quad index_{i+1}\, a \;=\; f\,(\textstyle\sum_{p \mid a\ \underline{writes}\ p}\ G\,(\sum_{p' \mid p'\ \underline{cites}\ p}\ H\,(\mathrm{agg}_{\,a' \mid a'\ \underline{writes}\ p'}\ index_i\ a')))$$

The step from $index_i$ to $index_{i+1}$ can be expressed (and computed) in the MapReduce framework. The computation is similar to the one for Page Rank, but three times as complicated because there is a nesting of three aggregates in the equation for *index* in contrast to the single aggregate in the equation for page rank. To express the computation, we assume that the following functions are globally available:

$$\begin{array}{lcl} writers\ p & = & \{a \mid a\ \underline{writes}\ p\} \\ citedby\ p' & = & \{p \mid p'\ \underline{cites}\ p\} \end{array}$$

Figure 1 on page 4 symbolically sketches the computation of $index_{i+1}$ given the availability of $index_i$. It uses several little map steps, and some adjacent steps can be combined into larger steps. The formal rendering of the computation reads as follows:

$Data_0 :=$ the set of all publications
$\mapsto$ map (providing each publication with the citation count and author list)
$Data_1 := \{p' : Data_0 \bullet (p',\ \#(citedby\ p'),\ writers\ p')\}$
$\mapsto$ map (replacing each author by the n^{th} approximation of his citation index)
$Data_2 := \{(p', c', as') : Data_1 \bullet (p',\ c',\ \{a' : as' \bullet index_i\ a'\})\}$
$\mapsto$ reduce-and-map (aggregate, and subject the result to H)
$Data_3 := \{(p', c', is') : Data_2 \bullet (p',\ H(\mathrm{agg}\ is'))\}$
$\mapsto$ map (from (p', v') produce all (p, v') for $p \in citedby\ p'$)
$Data_4 := \bigcup\{(p', v') : Data_3 \bullet \{p : citedby\ p' \bullet (p,\ v')\}\}$
$\mapsto$ map (adding the authors)
$Data_5 := \bigcup\{(p, v') : Data_4 \bullet \{a : writers\ p\ \bullet\ (a, p, v')\}\}$
$\mapsto$ group by author and publication
$Data_6 := \{(a, p, ...) : Data_5 \bullet (a, p, \{v' \mid (a, p, v') \in Data_5\})\}$
$\mapsto$ reduce-and-map (sum, and subject the result to G)
$Data_7 := \{(a, p, vs') : Data_6 \bullet (a, G(\sum vs'))\}$
$\mapsto$ group by author
$Data_8 := \{(a, ...) : Data_7 \bullet (a, \{v \mid (a, v) \in Data_7\})\}$
$\mapsto$ reduce-and-map (sum, and adapt the results)
$Data_9 := \{(a, vs) : Data_8 \bullet (a, f(\sum vs))\}$
$\approx$ theorem, proved below in paragraph (7)
$index_{i+1}$

The proof of the last step is a matter of simply back substituting $Data_8$ up to $Data_0$, and applying laws from pure set and predicate logic; see paragraph (7).

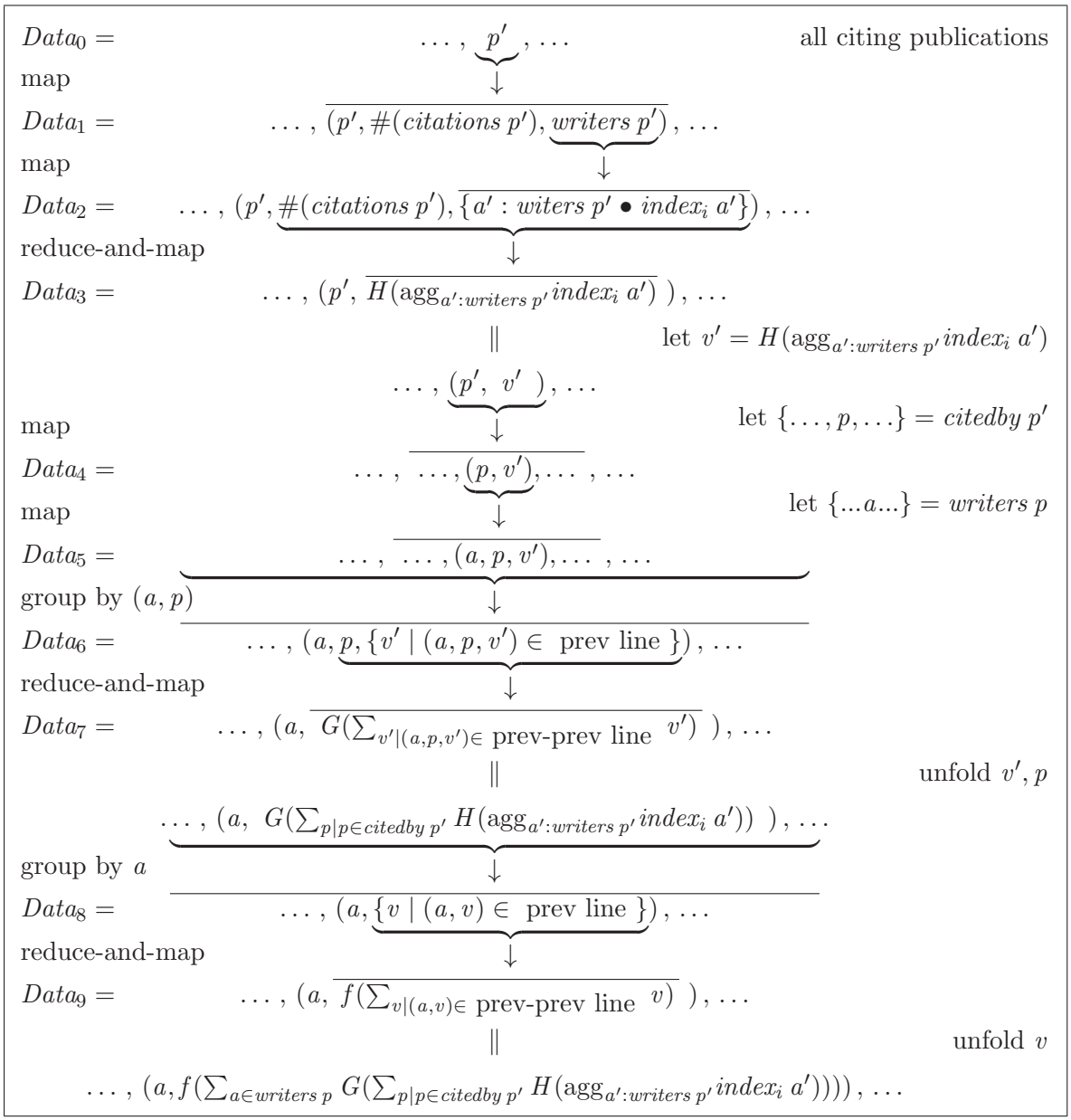

Figure 1: Symbolic formulation of the MapReduce computation of $index_{i+1}$ from $index_i$. Each step with label 'unfold' substitutes the defining expression for an abbreviating name; the correctness is formally shown the proof of Theorem 7.

(7) Theorem. Viewing function $index_{i+1}$ as a set of argument-result pairs, we expect that $Data_9$ contains the same pairs (a, i) as $index_{i+1}$, that is, $Data_9$ and $index_{i+1}$ are equal. In the proof attempt below, it turns out that this is *almost* the case. The true statement is:

$$Data_9 \quad = \quad \{(a, i) : index_{i+1} \mid \text{“}a \text{ has been cited”}\}$$

The condition “a has been cited” is, in retrospect, to be expected, but it came for us as a surprise, as a result of the formal manipulations. In order to try and prove the equality of $Data_9$ and $index_{i+1}$, it suffices to show that $(a, i) \in Data_9$ is equivalent to $(a, i) \in index_{i+1}$. Apart from the surprise, the calculation itself is straightforward, with only simple steps from set and predicate logic:

$$\begin{array}{ll}
 & (a, i) \in Data_9 \\
\Leftrightarrow & \quad \text{defn } Data_9\text{, and at the same time set membership} \\
 & \exists\, vs \mid (a, vs) \in Data_8 \bullet i = f(\sum vs) \\
\Leftrightarrow & \quad \text{defn } Data_8\text{, and at the same time set membership} \\
 & \exists\, vs \mid (\exists\, v \mid (a, v) \in Data_7 \bullet vs = \{v \mid (a, v) \in Data_7\}) \bullet i = f(\sum vs) \\
\Leftrightarrow & \quad \text{shunting } (v \text{ outwards}), \text{ equality and one-point rule (eliminating } vs) \\
 & \exists\, v \mid (a, v) \in Data_7 \bullet i = f(\sum\{v \mid (a, v) \in Data_7\}) \\
\Leftrightarrow & \\
 & (\exists\, v \bullet (a, v) \in Data_7) \wedge i = f(\textstyle\sum_{v \mid (a,v) \in Data_7} v) \\
\Leftrightarrow & \quad \text{twice: defn } Data_7\text{, and at the same time set membership} \\
 & (\exists\, v \bullet (\exists\, p, vs' \mid (a, p, vs') \in Data_6 \bullet (a, v) = (a, G(\textstyle\sum vs')))) \wedge \\
 & i = f(\textstyle\sum_{v \mid (\exists\, p, vs' \mid (a,p,vs') \in Data_6 \bullet (a,v) = (a, G(\sum vs')))} v) \\
\Leftrightarrow & \quad \text{twice: shunting } (p, vs' \text{ outward}), \text{ equality and one-point rule (eliminating } v) \\
 & (\exists\, p, vs' \bullet (a, p, vs') \in Data_6) \wedge i = f(\textstyle\sum_{p, vs' \mid (a,p,vs') \in Data_6} G(\sum vs')) \\
\Leftrightarrow & \quad \text{twice: defn } Data_6\text{, and at the same time set membership} \\
 & (\exists\, p, vs' \bullet (\exists\, v' \mid (a, p, v') \in Data_5 \bullet vs' = \{v' \mid (a, p, v') \in Data_5\})) \wedge \\
 & i = f(\textstyle\sum_{p, vs' \mid (\exists\, v' \mid (a,p,v') \in Data_5 \bullet vs' = \{v' \mid (a,p,v') \in Data_5\})} G(\sum vs')) \\
\Leftrightarrow & \quad \text{twice: shunting } (v' \text{ outward}), \text{ equality and one-point-rule (eliminating } vs') \\
 & (\exists\, p, v' \mid (a, p, v') \in Data_5) \wedge i = f(\textstyle\sum_{p, v' \mid (a,p,v') \in Data_5} G(\sum_{v' \mid (a,p,v') \in Data_5} v')) \\
\Leftrightarrow & \quad \text{thrice: defn } Data_5\text{, and at the same time set membership in a big union} \\
 & (\exists\, p, v' \mid (p, v') \in Data_4 \bullet a \in writers\, p) \wedge \\
 & i = f(\textstyle\sum_{p, v' \mid (p,v') \in Data_4 \wedge a \in writers\, p} G(\sum_{v' \mid (p,v') \in Data_4 \wedge a \in writers\, p} v')) \\
\Leftrightarrow & \quad \text{thrice: defn } Data_4\text{, and at the same time set membership in a big union} \\
 & (\exists\, p, v' \mid (\exists\, p' \mid (p', v') \in Data_3 \bullet p \in citedby\, p') \bullet a \in writers\, p) \wedge \\
 & i = f(\textstyle\sum_{p, v' \mid (\exists\, p' \mid (p', v') \in Data_3 \bullet p \in citedby\, p') \wedge a \in writers\, p} \\
 & \qquad G(\sum_{v' \mid (\exists\, p' \mid (p', v') \in Data_3 \bullet p \in citedby\, p') \wedge a \in writers\, p} v')) \\
\Leftrightarrow & \quad \text{shunting} \\
 & (\exists\, p, (p', v') : Data_3 \bullet p \in citedby\, p' \wedge a \in writers\, p) \wedge \\
 & i = f(\textstyle\sum_{p, (p', v') : Data_3 \mid p \in citedby\, p' \wedge a \in writers\, p} \\
 & \qquad G(\sum_{(p', v') : Data_3 \mid p \in citedby\, p' \wedge a \in writers\, p} v')) \\
\Leftrightarrow & \quad \text{thrice: defn } Data_3\text{, and at the same time set membership, shunting}
\end{array}$$

$$
\begin{array}{ll}
 & (\exists\, p, p', v', c', is' \bullet (p', c', is') \in Data_2 \wedge v' = H(\text{agg } is') \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p',v',c',is' \bullet (p',c',is') \in Data_2 \wedge v' = H(\text{agg } is') \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p',v',c',is' | (p',c',is') \in Data_2 \wedge v' = H(\text{agg } is') \wedge p \in citedby\ p' \wedge a \in writers\ p} v')) \\
\Leftrightarrow & \quad \text{eliminating } v' \text{ (thrice equality, twice one-point rule)} \\
 & (\exists\, p, p', c', is' \bullet (p', c', is') \in Data_2 \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p',c',is' \bullet (p',c',is') \in Data_2 \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p',c',is' | (p',c',is') \in Data_2 \wedge p \in citedby\ p' \wedge a \in writers\ p} H(\text{agg } is'))) \\
\Leftrightarrow & \quad \text{thrice: defn } Data_2\text{, and at the same time set membership, shunting} \\
 & (\exists\, p, p', c', is', as' \bullet \\
 & (p', c', as') \in Data_1 \wedge is' = \{a' : as' \bullet index_i\ a'\} \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p',c',is',as' \bullet (p',c',as') \in Data_1 \wedge is' = \{a' : as' \bullet index_i\ a'\} \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p',c',is',as' | (p',c',as') \in Data_1 \wedge is' = \{a' : as' \bullet index_i\ a'\} \wedge p \in citedby\ p' \wedge a \in writers\ p} H(\text{agg } is'))) \\
\Leftrightarrow & \quad \text{eliminating } is' \text{ (shunting, one-point rule)} \\
 & (\exists\, p, p', c', as' \bullet (p', c', as') \in Data_1 \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p',c',as' \bullet (p',c',as') \in Data_1 \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p',c',as' | (p',c',as') \in Data_1 \wedge p \in citedby\ p' \wedge a \in writers\ p} H(\text{agg } \{a' : as' \bullet index_i\ a'\}))) \\
\Leftrightarrow & \quad \text{thrice: defn } Data_1\text{, and at the same time set membership, shunting} \\
 & (\exists\, p, p', c', as' \bullet \\
 & p' \in Data_0 \wedge c' = \#(citedby\ p') \wedge as' = writers\ p' \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p',c',as' \bullet p' \in Data_0 \wedge c' = \#(citedby\ p') \wedge as' = writers\ p' \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p',c',as' | p' \in Data_0 \wedge c' = \#(citedby\ p') \wedge as' = writers\ p' \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad\qquad H(\text{agg } \{a' : as' \bullet index_i\ a'\}))) \\
\Leftrightarrow & \quad \text{eliminating } c' \text{ and } as' \text{ (by shunting and one-point rule)} \\
 & (\exists\, p, p' \bullet p' \in Data_0 \wedge p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p' \bullet p' \in Data_0 \wedge p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p' | p' \in Data_0 \wedge p \in citedby\ p' \wedge a \in writers\ p} H(\text{agg } \{a' : writers\ p' \bullet index_i\ a'\}))) \\
\Leftrightarrow & \quad \text{default domain of } p' \text{ is } Data_0 \\
 & (\exists\, p, p' \bullet p \in citedby\ p' \wedge a \in writers\ p) \wedge \\
 & i = f(\sum_{p,p' \bullet p \in citedby\ p' \wedge a \in writers\ p} \\
 & \qquad G(\sum_{p' | p \in citedby\ p' \wedge a \in writers\ p} H(\text{agg } \{a' : writers\ p' \bullet index_i\ a'\}))) \\
\Leftrightarrow & \quad \text{defn } index_{i+1} \text{ (6)} \\
 & (\exists\, p, p' \bullet p \in citedby\ p' \wedge a \in writers\ p) \wedge (a, i) \in index_{i+1}
\end{array}
$$

Writing the first condition in the last line as "a has been cited", the calculation proves:

$$Data_9 \;=\; \{(a, i) : index_{i+1} \mid \text{“}a \text{ has been cited”}\}$$

References

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *COMPUTER NETWORKS AND ISDN SYSTEMS*, pages 107–117. Elsevier

Science Publishers B. V., 1998.

[2] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In *OSDI'04: Sixth Symposium on Operating System Design and Implementation*, pages 137–150. USENIX, December 2004. Google Research Paper, `http://labs.google.com/papers/mapreduce.html`. Also: Comm. of the ACM, Jan 2008, Vol. 5, nr 1, pp. 107–113.

[3] Eigenfactor. Eigenfactor.org, ranking and mapping scientific knowledge. `http://www.eigenfactor.org` [Visited at 2013-02-22].

[4] M.M. Fokkinga. Mapreduce — a two-page explanation for laymen. Unpublished Technical Report, obtainable from `http://www.cs.utwente.nl/~fokkinga/mmf2008j.pdf`, 2008.

[5] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. In *Proceedings of the 7th International World Wide Web Conference*, pages 161–172, Brisbane, Australia, 1998.

[6] Wikipedia. Citation index — wikipedia, the free encyclopedia, 2013. `http://en.wikipedia.org/w/index.php?title=Citation_index&oldid=533014646`, [accessed 2013-02-22].

[7] Wikipedia. H-index — wikipedia, the free encyclopedia, 2013. [Online; accessed 18-February-2013].

(8) APPENDIX: eigenvector. It is well known that page ranks can be written as an eigenvector of a suitably defined matrix; the situation is more complicated for the weighted citation index. To see this, consider the page rank in its basic form, without damping. Let i, j denote (identifying numbers of) pages; then we have:

$$\begin{array}{lcl} \mathit{rank}\ i & = & \sum_{j \mid \text{page } j \text{ references page } i} \ \mathit{rank}\ j \\ & = & \sum_j \ (1 \text{ if page } j \text{ references page } i \text{ else } 0) \times \mathit{rank}\ j \\ & = & \sum_j \ A_{ij} \times \mathit{rank}\ j \quad , \end{array}$$

where $A_{ij} = (1 \text{ if page } j \text{ references page } i \text{ else } 0)$. So, viewing *rank* as a vector ($\mathit{rank}\,1$, $\mathit{rank}\,2$, ...) and denoting matrix multiplication by juxtaposition, we have:

$$\mathit{rank} \quad = \quad A\ \mathit{rank}$$

In contrast, the specification of *index* cannot be written as an eigenvector of a matrix, because the value *index a* is not a function of simply *one* aggregation of the values of *one* author set that depends on a.