Local Linear Presentations and Sparse Factorization Complexity

Luca Blanchi

June 2026

Abstract

We introduce two algebraic invariants for exact linear representation problems: local presentation rank and sparse factorization complexity. Let $P \subseteq \mathbb{F}^n$ be a family of linear observables. The t-local presentation rank $\lambda_t(P)$ is the minimum number of stored linear atoms $B = \{b_1, \ldots, b_s\} \subseteq \mathbb{F}^n$ such that every observable in P is a linear combination of at most t atoms from B. We prove that $\lambda_t(P)$ exactly characterizes static exact linear data structures with s stored linear cells and t-probe linear queries. Over finite fields, $\lambda_t(P)$ satisfies a sphere-covering lower bound, and for the full observable space $P = \mathbb{F}_q^r$ it is exactly the minimum length of a q-ary linear code of codimension r and covering radius at most t.

We then study dynamic exact linear representations. Given a workload matrix $A \in \mathbb{F}^{m\times n}$, we prove that an exact dynamic linear data structure with s cells, query locality q, and update locality u exists if and only if

$$A = RM,$$

where $M \in \mathbb{F}^{s\times n}$, $R \in \mathbb{F}^{m\times s}$, each row of R has sparsity at most q, and each column of M has sparsity at most u. Thus exact dynamic linear data structures and exact dynamic materialized aggregate views are precisely sparse matrix factorizations with asymmetric row and column locality constraints.

We prove generic finite-field lower bounds for sparse factorization, formulate the prefix-matrix sparse-factorization problem, and isolate the cancellation-free support-faithful obstruction via chain-intersection set systems. We also give a workload-biased interval-tree construction for dynamic prefix sums, showing that nonuniform update/query distributions admit entropy-sensitive expected costs via optimal alphabetic search trees.

Declaration of generative AI and AI-assisted technologies in the writing process

This article belongs to the Presentation theory section. Generative AI, including ChatGPT by OpenAI, was used to assist with mathematical drafting, formalization, review, and editing. The note may be preliminary, may have been only partially reviewed, and in some cases may be only AI-reviewed.

1 Introduction

Many data structures store linear summaries of an input vector. A static sketch stores linear cells

$$b_1x, \ldots, b_sx$$

and answers linear queries by combining some of these cells. A dynamic linear data structure maintains such cells under coordinate updates. A materialized aggregate view stores precomputed linear aggregates and answers workload queries by combining a few stored views.

This paper isolates the exact algebraic structure of such representations. Let $x \in \mathbb{F}^n$ be an input vector. A linear observable is a row vector $p \in \mathbb{F}^n$ which asks for px. A static exact linear representation stores cells $b_1 x, \ldots, b_s x$. A query has local access t if every observable p is answerable by reading at most t cells and forming a linear combination.

The natural invariant is the t-local presentation rank

$$\lambda_t(P) = \min\{|B| : P \subseteq \mathrm{span}_{\leq t}(B)\}.$$

It is a storage/access frontier for exact static linear representations.

For dynamic workloads, let $A \in \mathbb{F}^{m \times n}$ be a workload matrix. Query i asks for the i-th coordinate of Ax. A dynamic representation stores $Mx \in \mathbb{F}^s$. If query i reads at most q cells, row i of A must be a q-sparse linear combination of rows of M. If an update to coordinate j changes at most u cells, column j of M has sparsity at most u. Therefore dynamic exact linear representations are exactly factorizations

$$A = RM$$

with row-sparse R and column-sparse M.

The purpose of the paper is not to replace general cell-probe lower bounds. Rather, it identifies a representation-native algebraic model in which exact linear workloads become local generation and sparse factorization problems.

1.1 Presentation-theoretic interpretation

In the terminology of Presentation Theory, a workload $P \subseteq \mathbb{F}^n$ is a finite linear observable system on the ambient object $x \in \mathbb{F}^n$. A stored family

$$B = \{b_1, \ldots, b_s\} \subseteq \mathbb{F}^n$$

is a presentation layer: instead of accessing the observables in P directly, one presents them through a smaller or more structured set of stored atoms. The inclusion

$$P \subseteq \mathrm{span}_{\leq t}(B)$$

means that every observable in the original system factors through at most t atoms of the presentation layer. Thus $\lambda_t(P)$ is not only a storage invariant; it is the exact size of the smallest presentation layer whose induced observable profile reconstructs all of P with local access overhead t.

The dynamic factorization

$$A = RM$$

has the same reading with two simultaneous budgets. The matrix M is the maintained presentation of the changing input, the matrix R is the compiler from stored atoms to requested observables, row sparsity of R is observable-access cost, and column sparsity of M is update cost. In this sense sparse factorization is a controlled transfer package from coordinate-update access to workload-query access. The overhead is not an auxiliary estimate added after the fact; it is encoded in the support constraints on R and M.

1.2 Contributions

The main contributions are:

- We define the t-local presentation rank $\lambda_t(P)$ and prove that it exactly characterizes exact static linear data structures with local query access.
- Over finite fields, we prove the sphere-counting lower bound

$$|P| \le \sum_{j=0}^{t} \binom{s}{j} (q-1)^j$$

for every t-local presentation of size s.
- For the full observable space $P = \mathbb{F}_q^r$, we identify $\lambda_t(P)$ with the minimum length of a linear q-ary code of codimension r and covering radius at most t.
- For a workload matrix A, we define $\mathrm{sf}_{\mathbb{F}}(A; q, u)$ as the minimum s for which $A = RM$ with row sparsity of R at most q and column sparsity of M at most u, and prove that this exactly characterizes exact dynamic linear data structures.
- We prove generic lower bounds over finite fields by counting sparse factorizations.
- We formulate the prefix sparse-factorization problem for the lower-triangular all-ones matrix L_n, isolate a support-faithful cancellation-free obstruction, and state the arbitrary-field cancellation case as a central open problem.
- We give a workload-biased interval-tree construction for dynamic prefix sums, where the expected cost is a weighted path-length objective minimized by optimal alphabetic trees.

2 Static Local Linear Presentations

Let $\mathbb{F}$ be a field. For a vector v, write $|v|_0$ for its support size. For a finite set $B = \{b_1, \dots, b_s\} \subseteq \mathbb{F}^n$, define

$$\mathrm{span}_{\le t}(B) = \left\{ \sum_{j \in J} \alpha_j b_j : J \subseteq [s],\ |J| \le t,\ \alpha_j \in \mathbb{F} \right\}.$$

Definition 2.1 (Local presentation rank)**.** *Let $P \subseteq \mathbb{F}^n$ be a finite set of linear observables. The t-local presentation rank of P is*

$$\lambda_t(P) = \min \left\{ s : \exists B \subseteq \mathbb{F}^n,\ |B| = s,\ P \subseteq \mathrm{span}_{\le t}(B) \right\}.$$

A set B achieving this condition is called a t-local presentation of P.

Theorem 2.2 (Static local-linear characterization)**.** *There exists an exact static linear data structure for P with s stored linear cells and at most t probes per query if and only if*

$$\lambda_t(P) \le s.$$

Proof. Suppose the data structure exists. Its cells are $b_1 x, \dots, b_s x$. For every $p \in P$, the query algorithm reads cells indexed by some set $J_p \subseteq [s]$ with $|J_p| \le t$ and outputs

$$\sum_{j \in J_p} \alpha_{p,j} b_j x.$$

Correctness for all $x \in \mathbb{F}^n$ implies

$$p = \sum_{j \in J_p} \alpha_{p,j} b_j.$$

Thus $p \in \operatorname{span}_{\leq t}(B)$ for every $p \in P$, so $\lambda_t(P) \leq s$.

Conversely, suppose $P \subseteq \operatorname{span}_{\leq t}(B)$ for some $B = \{b_1, \ldots, b_s\}$. Store the cells $b_j x$. For each $p \in P$, fix a representation of p as a linear combination of at most t elements of B. The query reads those cells and combines them using the fixed coefficients. This returns px exactly. □

Proposition 2.3 (Basic properties)**.** *The invariant λ_t satisfies:*

1. *if $P \subseteq Q$, then $\lambda_t(P) \leq \lambda_t(Q)$;*
2. *if $t \leq t'$, then $\lambda_{t'}(P) \leq \lambda_t(P)$;*
3. *if $r = \operatorname{rank}(P)$, then $\lambda_r(P) \leq r$;*
4. *$\lambda_1(P)$ is the number of distinct projective directions among nonzero elements of P.*

Proof. The first two assertions are monotonicity in the observable family and in locality. For the third, choose a basis of the linear span of P. For the fourth, a nonzero observable lies in the one-local span of B precisely when it is a scalar multiple of an atom of B. □

3 Finite Fields and Covering Codes

Assume now that $\mathbb{F} = \mathbb{F}_q$. Define

$$V_q(s,t) = \sum_{j=0}^{t} \binom{s}{j} (q-1)^j.$$

This is the number of formal vectors of Hamming weight at most t in $\mathbb{F}_q^s$.

Theorem 3.1 (Local sphere-counting lower bound)**.** *For every finite $P \subseteq \mathbb{F}_q^n$,*

$$|P| \leq V_q(\lambda_t(P), t) = \sum_{j=0}^{t} \binom{\lambda_t(P)}{j} (q-1)^j.$$

Equivalently,

$$\lambda_t(P) \geq \min \left\{ s : \sum_{j=0}^{t} \binom{s}{j} (q-1)^j \geq |P| \right\}.$$

For fixed q and t,

$$\lambda_t(P) = \Omega_{q,t}(|P|^{1/t}).$$

Proof. Let B be a t-local presentation of P with $|B| = s$. Every vector in $\operatorname{span}_{\leq t}(B)$ is obtained by choosing at most t atoms and nonzero coefficients on those atoms. The number of such formal choices is at most $V_q(s,t)$. Different choices may produce the same vector, so this is an upper bound on $|\operatorname{span}_{\leq t}(B)|$. Since $P \subseteq \operatorname{span}_{\leq t}(B)$, we have $|P| \leq V_q(s,t)$. Taking $s = \lambda_t(P)$ proves the first claim. The asymptotic lower bound follows from $V_q(s,t) = O_{q,t}(s^t)$. □

Let

$$\Lambda_q(r,t) = \lambda_t(\mathbb{F}_q^r).$$

Theorem 3.2 (Local presentation rank equals covering-code length)**.**

$$\Lambda_q(r,t) = \min\left\{ s : \exists C \le \mathbb{F}_q^s \textit{ linear of codimension } r \textit{ and covering radius } \le t \right\}.$$

Proof. Let $B = \{b_1, \dots, b_s\} \subseteq \mathbb{F}_q^r$ be a t-local presentation of $\mathbb{F}_q^r$. Define $G : \mathbb{F}_q^s \to \mathbb{F}_q^r$ by $G(e_j) = b_j$. Since B locally spans all of $\mathbb{F}_q^r$, the map G is surjective. Let $C = \ker G$. Then C has codimension r. For every $a \in \mathbb{F}_q^r$, t-local spanning means there exists $z \in \mathbb{F}_q^s$ of Hamming weight at most t such that $Gz = a$. The fiber $G^{-1}(a)$ is a coset of C, so every coset of C contains a word of weight at most t.

Conversely, suppose $C \le \mathbb{F}_q^s$ has codimension r and covering radius at most t. Let $G : \mathbb{F}_q^s \to \mathbb{F}_q^r$ be a parity-check map with kernel C, and set $b_j = G(e_j)$. For every $a \in \mathbb{F}_q^r$, the coset $G^{-1}(a)$ contains some z of Hamming weight at most t. Therefore $a = Gz$ is a linear combination of at most t columns b_j. Hence the b_j form a t-local presentation of $\mathbb{F}_q^r$. □

Corollary 3.3. *For fixed q and t,*

$$\Lambda_q(r,t) \ge \Omega_{q,t}(q^{r/t}).$$

4 Dynamic Sparse Factorization

Let $A \in \mathbb{F}^{m\times n}$ be a workload matrix. The database is $x \in \mathbb{F}^n$, and query $i \in [m]$ asks for $(Ax)_i$. A dynamic linear representation stores $Mx \in \mathbb{F}^s$ for some $M \in \mathbb{F}^{s\times n}$. A coordinate update $x_j \leftarrow x_j + \delta$ changes precisely those stored cells k for which $M_{k,j} \ne 0$. Thus update locality is column sparsity of M. A query reads cells of Mx and combines them linearly. Thus query locality is row sparsity of a decoding matrix R.

Definition 4.1 (Sparse factorization complexity)**.** *For $A \in \mathbb{F}^{m\times n}$ and locality parameters q, u, define $\mathrm{sf}_{\mathbb{F}}(A; q, u)$ as the minimum s for which there are matrices*

$$R \in \mathbb{F}^{m\times s}, \qquad M \in \mathbb{F}^{s\times n}$$

such that

$$A = RM, \qquad |R_{i,*}|_0 \le q \ \forall i, \qquad |M_{*,j}|_0 \le u \ \forall j.$$

Here s is storage width, q is query locality, and u is update locality.

Theorem 4.2 (Dynamic linear representations are sparse factorizations)**.** *There exists an exact dynamic linear data structure for workload A with s stored linear cells, at most q cell probes per query, and at most u cell changes per coordinate update if and only if*

$$A = RM$$

with $R \in \mathbb{F}^{m\times s}$, $M \in \mathbb{F}^{s\times n}$, every row of R having sparsity at most q, and every column of M having sparsity at most u.

Proof. Suppose a dynamic linear data structure exists. Since stored cells are linear functions of x, there is a matrix $M \in \mathbb{F}^{s\times n}$ such that the stored state is Mx. An update to coordinate j changes cell k exactly when $M_{k,j} \ne 0$, so update locality gives $|M_{*,j}|_0 \le u$.

For query i, the algorithm reads at most q cells and returns a linear combination. Therefore there is a row vector $R_i \in \mathbb{F}^s$ with $|R_i|_0 \le q$ such that $(Ax)_i = R_i M x$ for all x. Hence $A_i = R_i M$. Stacking these rows gives $A = RM$.

Conversely, given such a factorization, store Mx. A coordinate update to x_j changes exactly the cells corresponding to nonzero entries of $M_{*,j}$, at most u of them. Query i reads the cells corresponding to the support of R_i, at most q cells, and returns $R_i M x = (Ax)_i$. □

5 Generic Lower Bounds Over Finite Fields

Let A be uniformly random in $\mathbb{F}_q^{m\times n}$. Define

$$N_R(s,q) = \sum_{a=0}^{q} \binom{s}{a}(q-1)^a, \qquad N_M(s,u) = \sum_{b=0}^{u} \binom{s}{b}(q-1)^b.$$

Theorem 5.1 (Generic sparse-factorization lower bound)**.** *For random $A \sim \mathrm{Unif}(\mathbb{F}_q^{m\times n})$,*

$$\Pr[\mathrm{sf}_{\mathbb{F}_q}(A;q,u) \le s] \le q^{-mn} N_R(s,q)^m N_M(s,u)^n.$$

Equivalently,

$$\Pr[\mathrm{sf}_{\mathbb{F}_q}(A;q,u) \le s] \le q^{m\log_q N_R(s,q)+n\log_q N_M(s,u)-mn}.$$

Proof. There are q^{mn} possible $m\times n$ matrices. The number of possible matrices R with row sparsity at most q is at most $N_R(s,q)^m$, and the number of possible matrices M with column sparsity at most u is at most $N_M(s,u)^n$. Each pair (R,M) determines at most one matrix $A = RM$. Dividing by q^{mn} gives the probability bound. □

Corollary 5.2. *If*

$$m\log_q N_R(s,q) + n\log_q N_M(s,u) \le mn - c,$$

then

$$\Pr[\mathrm{sf}_{\mathbb{F}_q}(A;q,u) \le s] \le q^{-c}.$$

Thus almost all workloads require large storage, large query locality, or large update locality.

6 The Prefix Matrix and Cancellation

Let $L_n \in \mathbb{F}^{n\times n}$ be the lower-triangular all-ones matrix:

$$(L_n)_{i,j} = \begin{cases} 1, & j \le i, \\ 0, & j > i. \end{cases}$$

This matrix represents dynamic prefix sums. Fenwick trees and segment trees give

$$s = O(n), \qquad q = O(\log n), \qquad u = O(\log n).$$

In sparse-factorization language, these are factorizations $L_n = RM$ with $s = O(n)$, row sparsity $O(\log n)$, and column sparsity $O(\log n)$.

Definition 6.1 (Support-faithful factorization)**.** *A factorization $A = RM$ is support-faithful if, for every (i,j),*

$$A_{i,j} = 0 \iff \text{there is no middle index } k \text{ such that } R_{i,k}M_{k,j} \ne 0.$$

Equivalently, the support of A is the Boolean product of the supports of R and M.

Support-faithfulness holds in semigroup or noncancelling models. It is stronger than ordinary field-linear factorization, because over fields zeros may arise by cancellation.

Proposition 6.2 (Prefix systems and intersections). *Suppose $L_n = RM$ is support-faithful. For each coordinate j, define*

$$A_j = \mathrm{supp}(M_{*,j}),$$

and for each prefix query i, define

$$B_i = \mathrm{supp}(R_{i,*}).$$

Then

$$j \le i \quad \Longleftrightarrow \quad A_j \cap B_i \neq \varnothing,$$

with $|A_j| \le u$ and $|B_i| \le q$.

Proof. Support-faithfulness says $(L_n)_{i,j} = 1$ if and only if there is some middle index k with $R_{i,k}M_{k,j} \neq 0$, equivalently $A_j \cap B_i \neq \varnothing$. Since $(L_n)_{i,j} = 1$ exactly when $j \le i$, the claim follows. □

Proposition 6.3 (Support-faithful prefix systems induce skew set-pair systems). *Every support-faithful factorization $L_n = RM$ with row sparsity at most q, column sparsity at most u, and middle dimension s induces a skew set-pair system*

$$(A_{k+1}, B_k)_{k=1}^{n-1}$$

on a ground set of size s, with $|A_{k+1}| \le u$ and $|B_k| \le q$. Consequently, any upper bound on the size of such skew set-pair systems gives a lower bound on the possible parameters (s, q, u) of support-faithful prefix factorizations.

Proof. For each k, the prefix relation gives $A_{k+1} \cap B_k = \varnothing$ because $k + 1 > k$. If $\ell \ge k + 1$, then $A_{k+1} \cap B_\ell \neq \varnothing$. These are precisely triangular intersection conditions on the middle universe $[s]$. □

Remark 6.4. *This support-faithful obstruction is deliberately modest. Earlier informal versions of this program suggested a ground-set-free Bollobas-type bound for the triangular condition. That is too strong in this generality. The correct cancellation-free obstruction is via skew set-pair systems on a middle universe of size s, and the resulting inequality depends on the precise skew-Bollobas theorem used. Stronger logarithmic lower bounds require additional structure or a cancellation-resistant algebraic argument.*

Problem 6.5 (Prefix sparse-factorization lower bound). *Let $\mathbb{F}$ be a field. Suppose*

$$L_n = RM, \qquad R \in \mathbb{F}^{n \times s}, \qquad M \in \mathbb{F}^{s \times n},$$

every row of R has sparsity at most q, and every column of M has sparsity at most u. Determine the optimal tradeoff among s, q, u. In particular, for $s = O(n)$, is it true that

$$\max\{q, u\} = \Omega(\log n)?$$

7 Workload-Biased Prefix Presentations

The classical balanced interval decomposition gives worst-case $O(\log n)$ update and query cost. If the workload is nonuniform, a biased tree can do better in expectation.

Let point update j occur with probability p_j. Let prefix query at boundary i occur with probability r_i. Construct an alphabetic binary tree on the ordered leaves $1, \ldots, n$. Store at each internal node the sum of its interval, or equivalently maintain a tree-based interval representation.

Theorem 7.1 (Biased interval presentation)**.** *For every alphabetic binary tree T on $[n]$, there is a dynamic prefix-sum representation with expected operation cost*

$$C_T = \sum_{j=1}^{n} p_j d_T(j) + \sum_{i=0}^{n} r_i d_T^{\mathrm{gap}}(i) + O(1),$$

where $d_T(j)$ is the depth of leaf j and $d_T^{\mathrm{gap}}(i)$ is the depth of the boundary or gap corresponding to prefix query i. Consequently, choosing an optimal alphabetic tree minimizes the expected operation cost over this interval-tree representation class.

Proof. A point update modifies exactly the stored interval sums on the path from leaf j to the root, so its cost is $d_T(j) + O(1)$. A prefix query at boundary i can be decomposed into the disjoint union of $O(d_T^{\mathrm{gap}}(i))$ canonical subtrees encountered along the search path to the boundary. Reading their stored sums yields the prefix sum. Taking expectation over update and query probabilities gives the formula. Optimizing over alphabetic trees gives the final statement. □

Corollary 7.2 (Entropy-sensitive expected cost)**.** *For distributions with entropy H over update leaves and query gaps, an optimal alphabetic tree achieves expected cost*

$$O(H + 1)$$

within the standard alphabetic-coding overheads.

8 Materialized Aggregate Views

Let $x \in \mathbb{F}^n$ represent base data values and let $A \in \mathbb{F}^{m\times n}$ be a linear aggregate workload. A materialized view design stores Mx. A query plan combines stored views, so query locality is row sparsity of R. A base update to coordinate j updates the views whose definitions depend on x_j, so update locality is column sparsity of M.

Theorem 8.1 (Materialized-view sparse-factorization theorem)**.** *An exact linear materialized-view design for workload A with s stored views, at most q views combined per query, and at most u views updated per base-coordinate update exists if and only if*

$$A = RM$$

where $R \in \mathbb{F}^{m\times s}$ and $M \in \mathbb{F}^{s\times n}$, with $|R_{i,*}|_0 \le q$ for every i and $|M_{*,j}|_0 \le u$ for every j.

Proof. This is Theorem 4.2 translated into materialized-view language. □

9 Variants and Open Problems

The exact theory suggests several relaxations. One may allow approximate sparse factorization $A \approx RM$ under a workload norm, randomized representations with correctness in expectation or with high probability, semiring factorizations over Boolean, tropical, or nonnegative semirings, or multi-layer factorizations

$$A = M_d M_{d-1} \cdots M_1$$

with sparsity constraints per layer.

Problem 9.1. *Prove or refute the field-linear prefix lower bound*

$$L_n = RM,\ s = O(n) \quad \Longrightarrow \quad \max\{q, u\} = \Omega(\log n)$$

over every field.

Problem 9.2. *Determine the optimal lower bound on q and u as a function of storage s for $L_n = RM$.*

Problem 9.3. *Extend the sparse-factorization analysis from prefix matrices to interval matrices and multidimensional box-query matrices.*

Problem 9.4. *Develop lower bounds for approximate factorizations $A \approx RM$ under query-relevant norms.*

Problem 9.5. *Given A, determine the computational complexity of finding near-optimal sparse factors R, M.*

10 Conclusion

This paper introduced local presentation rank and sparse factorization complexity as algebraic invariants for exact linear representations. For static linear query workloads, $\lambda_t(P)$ exactly characterizes storage under t-local query access. For the full observable space over a finite field, it is equivalent to the length of a linear code of specified codimension and covering radius.

For dynamic linear workloads, exact representations are precisely sparse factorizations

$$A = RM,$$

where row sparsity of R is query locality and column sparsity of M is update locality. This framework gives exact characterizations of static and dynamic linear representations, finite-field generic lower bounds, coding-theoretic connections, a sparse-factorization formulation of dynamic prefix sums, a materialized-view interpretation, workload-biased prefix presentations via optimal alphabetic trees, and concrete open problems in algebraic lower bounds.

In Presentation Theory terms, a workload matrix is a finite linear observable system. Compressing that observable system, or refining a residual fibre by adding local linear probes, is exactly the problem of factoring the corresponding workload through a small family of stored atoms. Thus sparse factorization is the linear algebraic mechanism behind local observable compression; it is not a metaphor for compression but the precise presentation model for exact linear access.

The main thesis is:

exact linear representation complexity is local generation and sparse factorization.

The strongest open direction is to prove sharp arbitrary-field lower bounds for sparse factorizations of the prefix matrix. Such a result would give a representation-native algebraic explanation of dynamic prefix-sum hardness.

References

[1] P. M. Fenwick. A new data structure for cumulative frequency tables. *Software: Practice and Experience* 24 (1994), 327–336.

[2] M. L. Fredman and M. E. Saks. The cell probe complexity of dynamic data structures. In *Proceedings of STOC 1989*, 345–354.

[3] M. Patrascu and E. D. Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing* 35 (2006), 932–963.

[4] G. Cohen, I. Honkala, S. Litsyn, and A. Lobstein. *Covering Codes*. North-Holland, 1997.

[5] G. D. Cohen, M. G. Karpovsky, H. F. Mattson, and J. R. Schatz. Covering radius: survey and recent results. *IEEE Transactions on Information Theory* 31 (1985), 328–343.

[6] B. Bollobas. On generalized graphs. *Acta Mathematica Academiae Scientiarum Hungaricae* 16 (1965), 447–452.

[7] P. Frankl. An extremal problem for two families of sets. *European Journal of Combinatorics* 3 (1982), 125–127.

[8] D. E. Knuth. Optimum binary search trees. *Acta Informatica* 1 (1971), 14–25.

[9] T. C. Hu and A. C. Tucker. Optimal computer search trees and variable-length alphabetical codes. *SIAM Journal on Applied Mathematics* 21 (1971), 514–532.

[10] A. M. Garsia and M. L. Wachs. A new algorithm for minimum cost binary trees. *SIAM Journal on Computing* 6 (1977), 622–642.

[11] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record* 26 (1997), 65–74.

[12] A. Gupta and I. S. Mumick, editors. *Materialized Views: Techniques, Implementations, and Applications*. MIT Press, 1999.