Space-Efficient Representations for Algebraic Structures (代数的構造の省メモリ表現)

数理情報学専攻 48-216232 柳田 達也
指導教員 定兼 邦彦 教授

1 はじめに

本発表では, 半順序集合と束を保持する空間効率の良いデータ構造について述べる. 半順序集合のデータ構造の応用先として, 仮想通貨や NFT の取引の記録に用いられるブロックチェーンが挙げられる. また, 束のデータ構造の応用先としては, データベース検索やプログラミング言語の継承関係の処理等が例として挙げられる.

2 準備

反射律, 反対称律, 推移律を満たす $(P, \leq)$ を半順序集合と言う. 記法として $a <: b$ は, $a < b$ であり, かつ $a < c < b$ となる c が存在しないことを言う.

半順序集合 P のトポロジカルラベリングとは, 全単射写像 $l : P \to \{1, 2, \ldots, |P|\}$ であって, $a \leq b \Rightarrow l(a) \leq l(b)$ を満たすものである.

半順序集合 P の Transitive Closure Graph は, 頂点集合 P と辺の集合 $E_C = \{(a, b) \in P^2;\ a < b\}$ のペア $G_C = (P, E_C)$ で定義される. また, Transitive Reduction Graph は, 頂点集合 P と辺の集合 $E_R = \{(a, b) \in P^2;\ a <: b\}$ のペア $G_R = (P, E_R)$ である.

半順序集合 $(P, \leq)$ のチェーン分解とは, $P = \coprod_{i=1}^{k} C_i$ かつ各 C_i が $\leq$ に関して全順序となるような $\{C_i\}_{i=1}^{k}$ である. k として取り得る値の最小値を Dilworth の幅 [1], 若しくは単に幅と呼ぶ.

可換律, 結合律, 吸収律を満たす $(L, \vee, \wedge)$ を束と言う. $\vee$ を結び (join), $\wedge$ を交わり (meet) という. 束は $a \vee b = a \Leftrightarrow b \leq a$ と順序関係を定義することで半順序集合となる.

本発表では有限要素数の束のみを考える. この時, 束は上で定めた順序関係において最大の元と最小の元を持つ. これらをそれぞれ $\top, \bot$ と表記する.

束 L における結び既約元とは, $x \neq \bot$ かつ ($a \vee b \Rightarrow a = x$ または $b = x$) となる $x \in L$ を指す. 交わり既約元とは, $x \neq \top$ かつ ($a \wedge b \Rightarrow a = x$ または $b = x$) を満たす x である.

ビットベクトル [2, 3] は 0 または 1 のビット列であり, ランダムアクセスと rank, select という操作を定数時間で行う. 長さ N のビットベクトルは $N + o(N)$ ビットで表現される. ストリング [4] は $1, 2, \ldots, K$ のアルファベットの列で, 同じくランダムアクセスと rank を $O(\log \log K)$ 時間, select を $O(1)$ 時間でサポートする. 長さ N のとき, 空間計算量は $N \log K + o(N \log K)$ ビットである.

3 半順序集合のデータ構造

要素数 n のラベル付き半順序集合を格納するのに必要な空間の情報理論的下限は, $\frac{n^2}{4} + \frac{3}{2}n + O(\log n)$ ビットであることが知られている [5]. また, 幅 w のラベル付き半順序集合に関しては $n \log n + 2n(w-1) - \Theta(w^2 \log n)$ ビットよりも情報理論的下限が大きい [6]. トポロジカルラベリングの半順序集合の情報理論的下限は $2n(w-1) - \Theta(w^2 \log n)$ ビットよりも大きい [7].

先行研究として, Munro と Nicholson [8] による $\frac{n^2}{4} + o(n^2)$ ビットのデータ構造, Farzan と Fischer [9] による $(1+\varepsilon)n \log n + 2nw + o(nw)$ ビットのデータ構造が知られている. ここで, $\varepsilon \in (0, 1]$ は任意の定数である. これらはともにラベル付きの半順序集合を格納する.

本発表ではトポロジカルラベリングの半順序集合を格納するデータ構造を 3 つ提案する. これは SWAT 2022 で発表された内容である [7]. $\{C_p\}_{p=1}^{w}$ を最小のチェーン分解とし, 各 C_p に対し $C_p = \{c_{p,1}, c_{p,2}, \ldots, c_{p,|C_p|}\}$ かつ $c_{p,i} < c_{p,i+1}$ $(i = 1, 2, \ldots, |C_p| - 1)$ と定義する.

1 つ目のデータ構造は Farzan と Fischer のデータ構造をトポロジカルラベリングの半順序集合へ応用したものである. これはストリング S とビットベクトル $B_{p,q}$ で構成される. S は $S[x] = p \Leftrightarrow x \in C_p$ というように定義される. $B_{p,q}$ は $B_{p,q} = 0^{d_{1,p,q}}10^{d_{2,p,q}}1 \cdots 10^{d_{|C_p|,p,q}}10^{d_{r,p,q}}$ と表される. ここで 0^d は 0 が d 個並んだもの, $d_{i,p,q} = |\{x \in C_q;\ x \leq c_{p,i}\}| - \sum_{j=1}^{i-1} d_{j,p,q}$, $d_{r,p,q} = |C_q| - \sum_{i=1}^{|C_p|} d_{i,p,q}$ である. これは合計で $2n(w-1) + o(nw)$ ビットを要する.

2 つ目のデータ構造はストリング S とビットベクトル $I_{p,q}$, $O_{p,q}$ からなる. S は上と全く同じように定義される. また各ビットベクトルは $I_{p,q}[i] = 1 \Leftrightarrow \exists x \in C_q, x <: c_{p,i}$ 及び, $O_{p,q}[i] = 1 \Leftrightarrow \exists x \in C_q, c_{p,i} <: x$

となるように格納される．空間計算量は合計で $2nw + o(nw)$ ビットとなる．

3 つ目のデータ構造はストリング S とビットベクトル $B_{p,q}$, T_x で構成される．S と $B_{p,q}$ は 1 つ目のものと同じである．T_x は $T_x[p] = 1 \Leftrightarrow \exists y \in C_p, y <: x$ となるように定義される．

半順序集合に関するクエリとして, adj_{G_C}, succ_{G_C}, pred_{G_C}, adj_{G_R}, succ_{G_R}, pred_{G_R} の 6 つを考える．それぞれ, G_C における隣接関係, 入力頂点, 出力頂点, G_R における隣接関係, 入力頂点, 出力頂点を求める．表 1 にそれぞれの計算量を記す.

表 1. 半順序集合の各データ構造の時間/空間計算量の比較.

	Proposal 1	Proposal 2	Proposal 3
Space [bits]	$2n(w-1)+o(nw)$	$2nw+o(nw)$	$3nw-2n+o(nk)$
$\mathrm{adj}_{G_C}(x,y)$	$O(\log\log w)$	$O(w^2)$	$O(\log\log w)$
$\mathrm{pred}_{G_C}(x)$	$O(w+t)$	$O(w^2+t)$	$O(w+t)$
$\mathrm{succ}_{G_C}(x)$	$O(w+t)$	$O(w^2+t)$	$O(w+t)$
$\mathrm{adj}_{G_R}(x,y)$	$O(w)$	$O(\log\log w)$	$O(\log\log w)$
$\mathrm{pred}_{G_R}(x)$	$O(w^2)$	$O(w)$	$O(\log\log w+t)$
$\mathrm{succ}_{G_R}(x)$	$O(w^2)$	$O(w)$	$O(w)$

4 束のデータ構造

要素数 n の束の情報理論的下限は $\Theta(n^{1.5})$ ビットであることが知られている [10, 11]. また先行研究として, Munro ら [12] による $O(n^{1.5}\log n)$ ビットのデータ構造が提案されている．本発表においては 2 つの新しいデータ構造を提案する．

J, M をそれぞれ結び既約元, 交わり既約元の集合, $n_J = |J|$, $n_M = |M|$ とする．また, J は半順序集合であるから幅が定義できて, それを w_J と表す．J の最小のチェーン分解を $\{C_p\}_{p=1}^{w_J}$, $x \in J$ に対して $p(x) = q \Leftrightarrow x \in C_q$ とする．また, 適当な変数 $a_1, a_2, \ldots, a_{w_J}$ を導入する．

この時, 束の元 $x \in L$ を $Enc(x) = \prod_{y \in J,\, y \leq x} a_{p(y)}$ という単項式としてエンコードする．これは既知 [13] の埋め込み $L \to \prod_{p=1}^{w_J} C_i$ と本質的に等しい．これを全ての元に関して, 保持することで束を表現できる．具体的には, ストリング S_p を用いて, $S_p[x]$ が $Enc(x)$ 内の a_p の次数を保持する．これが 1 つ目のデータ構造である．

2 つ目のデータ構造では, 交わり既約元のみのエンコードを保持することで束を表現する．これは交わり既約元のみから全ての元を復元できる [14] ことに由来する．

束におけるクエリとして, 順序関係を調べる lessthan, 結びを返す join, 交わりを返す meet の 3 つが挙げられる．それぞれにかかる計算量は表 2 の通りである．なお, 2 つ目のデータ構造に関してはこれらのクエリをサポートしない．

表 2. 束の各データ構造の時間/空間計算量の比較.

	Proposal 1	Proposal 2
Space [bits]	$O\left(nw_J \log \frac{n_J}{w_J}\right)$	$O\left(n_M w_J \log \frac{n_J}{w_J}\right)$
$\mathrm{lessthan}(x,y)$	$O\left(w_J \log\log \frac{n_J}{w_J}\right)$	-
$\mathrm{join}(x,y)$	$O\left(w_J \log n \log\log \frac{n_J}{w_J}\right)$	-
$\mathrm{meet}(x,y)$	$O\left(w_J \log\log \frac{n_J}{w_J}\right)$	-

参考文献

[1] R.P. Dilworth, “A Decomposition Theorem for Partially Ordered Sets,” Annals of Mathematics, 51(1), 1950.

[2] G.J. Jacobson, “Succinct Static Data Structures,” Ph.D Thesis, Carnegie Mellon University, 1988.

[3] A. Brodnik, J.I. Munro, “Membership in Constant Time and Almost-Minimum Space,” SIAM Journal on Computing, 28(5), 1999.

[4] A. Golynski, R. Raman, S. S. Rao, “On the Redundancy of Succinct Data Structures,” In Proceedings of 11th Scandinavian Workshops on Algorithm Theory, 2008.

[5] D.J. Kleitman, B.L. Rothschild, “Asymptotic Enumeration of Partial Orders on a Finite Set,” Transactions of the American Mathematical Society, 205, 1975.

[6] G. Brightwell, S. Goodall, “The Number of Partial Orders of Fixed Width,” Order, 13, 1996.

[7] T. Yanagita, S. Chakraborty, K. Sadakane, S.R. Satti, “Space-Efficient Data Structures for Posets with Applications,” In Proceedings of 18th Scandinavian Symposium and Workshops on Algorithm Theory, 2022.

[8] J.I. Munro, P.K. Nicholson, “Succinct Posets,” Algorithmica, 76, 2016.

[9] A. Farzan, J. Fischer, “Compact Representation of Posets,” In Proceeding of the 22nd International Symposium on Algorithms and Computation, 2011.

[10] W. Klotz, L. Lucht, “Endliche Verbände,” Journal Für die Reine und Angewandte Mathematik, 247, 1971.

[11] D.J. Kleitman, K.J. Winston, “The Asymptotic Number of Lattices,” Annals of Discrete Mathematics, 6, 1980.

[12] J.I. Munro, B. Sandlund, C. Sinnamon, “Space-Efficient Data Structure for Lattices,” In Proceedings of 17th Scandinavian Symposium and Workshops on Algorithm Theory, 2020.

[13] M. Siggers, “On the Representations of Finite Distributive Lattices,” arXiv:1412.0011v2, 2016.

[14] G. Birkhoff, O. Frink, “Representations of Lattices by Sets,” Transactions of the American Mathematical Society, 1948.