Lower bounds for succinct data structures

Emanuele Viola

Northeastern University

July 1 2009
Bits vs. trits

- Store $n$ “trits” $t_1, t_2, \ldots, t_n \in \{0,1,2\}$

- In $u$ bits $b_1, b_2, \ldots, b_u \in \{0,1\}$

- Want:
  - Small space $u$ (optimal = $\lceil n \log_2 3 \rceil$)
  - Fast retrieval: Get $t_i$ by probing few bits (optimal = 2)
Two solutions

• Arithmetic coding:
  Store bits of \((t_1, \ldots, t_n) \in \{0, 1, \ldots, 3^n - 1\}\)

  Optimal space: \(\lceil n \log_2 3 \rceil \approx n \cdot 1.584\)
  Bad retrieval: To get \(t_i\) probe all \(> n\) bits

• Two bits per trit

  Bad space: \(n \cdot 2\)
  Optimal retrieval: Probe 2 bits
Polynomial tradeoff

- Divide $n$ trits $t_1, \ldots, t_n \in \{0,1,2\}$ in blocks of $q$
- Arithmetic-code each block

Space: $\left\lceil q \log_2 3 \right\rceil \frac{n}{q} < (q \log_2 3 + 1) \frac{n}{q} = n \log_2 3 + \frac{n}{q}$

Retrieval: Probe $O(q)$ bits

polynomial tradeoff between redundancy, probes
Polynomial tradeoff

- Divide $n$ trits $t_1, ..., t_n \in \{0,1,2\}$ in blocks of $q$.

- Arithmetic-code each block.

\[
\text{Space: } \left\lceil q \log_2 3 \right\rceil \frac{n}{q} = (q \log_2 3 + \frac{1}{q^{\Theta(1)}}) \frac{n}{q}
\]

\[= n \log_2 3 + \frac{n}{q^{\Theta(1)}}\]

Retrieval: Probe $O(q)$ bits.

Logarithmic forms.
Expontential tradeoff

- Breakthrough [Pătraşcu '08, later + Thorup]

Space: $n \lg_2 3 + n/2^{\Omega(q)}$

Retrieval: Probe $q$ bits

- E.g., optimal space $\lceil n \lg_2 3 \rceil$, probe $O(\lg n)$
Our results

- **Theorem [V.]:**
  Store \( n \) trits \( t_1, \ldots, t_n \in \{0,1,2\} \) in \( u \) bits \( b_1, \ldots, b_u \in \{0,1\} \).

  If get \( t_i \) by probing \( q \) bits
  then space \( u > n \lg 2 3 + \frac{n}{2^{\Omega(q)}} \).

- Matches [Pătraşcu Thorup]: space \( < n \lg 2 3 + \frac{n}{2^{\Omega(q)}} \)

- Holds even for adaptive probes
Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- Cells vs. prefix sums
Theorem:
Store $n$ trits $t_1, ..., t_n \in \{0,1,2\}$
in $u$ bits $b_1, ..., b_u \in \{0,1\}$.

If get $t_i$ by probing $q$ bits
then space $u > n \lg_2 3 + \frac{n}{2^{O(q)}}$.

For now, assume non-adaptive probes:
$t_i = d_i(b_{i1}, b_{i2}, ..., b_{iq})$
Proof idea

- $t_i = d_i(b_{i1}, b_{i2}, ..., b_{iq})$

- Uniform $(t_1, ..., t_n) \in \{0,1,2\}^n$
  
  Let $(b_1, ..., b_u) := \text{Store}(t_1, ..., t_n)$

- Space $u \approx \text{optimal} \Rightarrow (b_1, ..., b_u) \in \{0,1\}^u \approx \text{uniform} \Rightarrow$
  
  
  $1/3 = \Pr [ t_i = 2 ] = \Pr [ d_i(b_{i1}, ..., b_{iq}) = 2 ] \approx A / 2^q \neq 1/3$

  Contradiction, so space $u \gg \text{optimal}$

Q.e.d.
Information-theory lemma
[Edmonds Rudich Impagliazzo Sgall, Raz, Shaltiel V.]

Lemma: Random \((b_1, \ldots, b_u)\) uniform in \(B \subseteq \{0,1\}^u\)

\[|B| \approx 2^u \Rightarrow \text{there is large set } G \subseteq [u]:\]

for every \(i_1, \ldots, i_q \in G : (b_{i_1}, \ldots, b_{i_q}) \approx \text{uniform in } \{0,1\}^q\)

Proof: \(|B| \approx 2^u \Rightarrow H(b_1, \ldots, b_u) \text{ large}\)

\[\Rightarrow H(b_i | b_1, \ldots, b_{i-1}) \text{ large for many } i (\in G)\]

Closeness\[ (b_{i_1}, \ldots, b_{i_q}), \text{ uniform } \] \(\geq H(b_{i_1}, \ldots, b_{i_q})\)

\(\geq H(b_{i_q} | b_1, \ldots, b_{i_q-1}) + \ldots + H(b_{i_1} | b_1, \ldots, b_{i_1-1}), \text{ large} \quad \text{Q.e.d.}\)
Proof

• Argument OK if probes in $G$

• $t_i = d_i (b_{i1}, b_{i2}, \ldots, b_{iq})$

• Uniform $(t_1, \ldots, t_n) \in \{0,1,2\}^n$

\[ \downarrow \]

uniform $(b_1, \ldots, b_u) \in B := \{\text{Store}(t) | t \in \{0,1,2\}^n\}$

\[ |B| = 3^n \approx 2^u \Rightarrow \text{(Lemma)} \Rightarrow (b_{i1}, \ldots, b_{iq}) \approx \text{uniform} \Rightarrow \]

\[ 1/3 = \Pr [ t_i = 2 ] = \Pr [ d_i (b_{i1}, \ldots, b_{iq}) = 2 ] \approx A / 2^q \neq 1/3 \]
If every $t_i$ probes bits not in $G$:

- Argue as in [Shaltiel V.]:
- Condition on heavy bits := probed by many $t_i$
- Can find $t_i \approx$ uniform in $\{0,1,2\}$, all probes in $G$
Handling adaptivity

- So far $t_i = d_i (b_{i1}, b_{i2}, \ldots, b_{iq})$

- In general, $q$ adaptively chosen probes = decision tree
  - $2^q$ bits
  - depth $q$

$$1/3 = \Pr [ t_i = 2 ] = \Pr [ d_i (b_{i1}, \ldots, b_{i2q}) = 2 ] \approx A / 2^q \neq 1/3$$
Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- Cells vs. prefix sums
Bits vs. sets

- Store $S \subseteq \{1, 2, \ldots, n\}$ of size $|S| = k$

In $u$ bits $b_1, \ldots, b_u \in \{0,1\}$

- Want:
  
  Small space $u$ (optimal $= \lceil \lg_2 (n \text{ choose } k) \rceil$)

  Answer “$i \in S?$” by probing few bits (optimal $= 1$)
Previous results

- Store $S \subseteq \{1, 2, \ldots, n\}$, $|S| = k$ in bits, answer “$i \in S$?”

- [Minsky Papert '69] Average-case study

- [Buhrman Miltersen Radhakrishnan Venkatesh; Pagh '00] Space $O(\text{optimal})$, probe $O(\lg(n/k))$
  
  Lower bounds for $k < n^{1-\varepsilon}$

- No lower bound was known for $k = \Omega(n)$
Our results

- **Theorem [V.]:**
  Store $S \subseteq \{1, 2, \ldots, n\}$, $|S| = n/3$ in $u$ bits $b_1, \ldots, b_u \in \{0,1\}$
  
  If answer “$i \in S$?” probing $q$ bits then space $u > \text{optimal} + n/2^{O(q)}$.

- First lower bound for $|S| = \Omega(n)$

- Holds even for adaptive probes
Outline

- Bits vs. trits
- Proof bits vs. trits
- Bits vs. sets
- **Cells** vs. prefix sums
Cell-probe model

- So far: \( q = \) number of bit probes

- Cell model: \( q = \) number of probes in cells of \( \lg(n) \) bits

![Diagram showing the relationship between data, store, and cells](diag.png)

- Relationship: \( q \text{ bit} \subseteq q \text{ cell} \subseteq q \lg(n) \) bit
Results in cell-probe model

- **Cells vs. trits:**
  
  \[ q = O(1), \text{ optimal space} = \left\lceil n \log_2 3 \right\rceil \quad [\text{Pătraşcu Thorup}] \]
  
  \[ q = 1 \Rightarrow \text{space} > n \log_2 3 + \frac{n}{\log^{O(1)} n} \quad [\text{this work}] \]

- **Cells vs. sets:**
  
  \[ q \text{ probes, space} = \text{optimal} + \frac{n}{\log^{\Omega(q)} n} \quad [\text{Pagh, Pătraşcu}] \]
  
  Lower bounds?

Work in progress on both fronts
Outline

• Bits vs. trits

• Proof bits vs. trits

• Bits vs. sets

• Cells vs. prefix sums
Prefix sums

- Store n bits $x_1, x_2, \ldots, x_n \in \{0,1\}$ in memory cells

- Want:
  - Small space
  - Fast answer prefix sum (a.k.a. Rank) queries:

$$\text{Sum}(i) := \sum_{k \leq i} x_k \in \{0, 1, 2, \ldots, n\}$$
History

- Fundamental problem: succinct trees, documents, ...

- Trivial
  - Space = $n \lg n$
  - Time = 1 cell probe

- [Jacobson '89]
  - Space = $n + O(n / \lg n)$
  - Time = $O(1)$ cell probes

- [Pătraşcu '08]
  - Space = $n + n / \lg^q n$
  - Time = $O(q)$ cell probes
Our results

• **Theorem[	extit{V.}]:**
  Store \( n \) bits in memory
  
  If answer \( \text{Sum}(i) := \sum_{k \leq i} x_k \) queries
  
  by **non-adaptively** probing \( q \) cells
  
  then space \( > n + \frac{n}{\lg^{O(q)} n} \).

• Matches [Pătraşcu]: space \( < n + \frac{n}{\lg^{\Omega(q)} n} \)
  
  non-adaptive
Proof idea

• Efficient data structure ⇒ **Break queries' correlations**

• For $i < j$, $A \subseteq \{0,1\}^n$

\[
0 = \Pr_{x \in A} [\text{Sum}(i) > t \text{ AND Sum}(j) < t]
\]

\[
> \Pr_{x \in A} [\text{Sum}(i) > t] \Pr_{x \in A}[\text{Sum}(j) < t]
\]

\[
> (1/10) \times (1/10) \gg 0
\]

• Contradiction, so data structure cannot be efficient
Previous proof focuses on single query. Works for bit probes, not cell.

New proof focuses on correlation between queries. Works for cell probes.
Balanced brackets

- Store \( n \) balanced brackets

- Want:
  Small space
  Fast answer \texttt{match} queries:

  - \textbf{Theorem[V.]}: space > optimal + \( \frac{n}{\lg \Omega(q)} \) \( n \).
    for non-adaptive \( q \) probes

  - \textbf{[Pătraşcu]}: space < optimal + \( \frac{n}{\lg \Omega(q)} \) \( n \) non-adaptive
New approach to lower bounds for basic data structures:

Representing trits, sets, prefix sums, balanced brackets using space = optimal + redundancy

Sometimes matching [Pătrașcu]

Open problems:
Some bounds are loose
    E.g. storing sets: 2 cell probes and optimal space?
Adaptive cell probes?
• $\Sigma \Pi \neg \cup \subseteq \subseteq \downarrow \Rightarrow \uparrow \leftarrow \leftarrow \forall \geq \forall \exists \Omega \alpha \beta \varepsilon \gamma \delta$

• $\neq \approx$