Dispersing Hash Functions

Rasmus Pagh
BRICS
University of Aarhus

Plan:
• "Dispersing Hash Functions" by way of a game.
• Element distinctness: universal vs. dispersing.
• Explicit constructions.
• "Almost perfect" hash functions.
A TWO-PLAYER GAME

THE GAME "COLOURFUL" HAS ONE ROUND:

Alice chooses $S \subseteq U = \{1, \ldots, u\}$, $|S| = n$

Independently, Bob colours $U$ using colours $\{1, \ldots, r\}$.

Bob's aim: maximize #colours in $S$
(worst case over all $S$)

A randomized strategy

- Pick each colour at random.
- $r \cdot \log r$ bits.

Bob can't afford this :(
Dispersing Hash Functions

Assume $U \geq r^n$, $r \geq n$. For $S \subseteq U$, $|S| = n$, a random function $h: U \rightarrow \{1, \ldots, r\}$ has

$$E[|h(S)|] = n - \lambda,$$

where $\lambda = \Theta(n^2/r)$

Definition

A family $\{h_i : h_i : U \rightarrow \{1, \ldots, r\}\}$ is $c$-dispersing if

$$\forall S \subseteq U : |S| = n$$

$$\exists i \in \mathbb{N}^+ : |h_i(S)| \geq n - c \lambda,$$

for all $S \subseteq U$ with $|S| = n$. 
PSEUDO-RANDOMNESS

It is useful to be able to pick and store functions that "look" like random functions.

Main concerns:
- Size of family
- Explicitness

Properties:
- Low collision probability (universality)
- $k$-wise independence
- Near-uniformity on sufficiently large sets (Extractors).
MOTIVATING EXAMPLE

ELEMENT DISTINCTNESS:
ARE $x_1, \ldots, x_n \in \{1, \ldots, 2^b\}$ DISTINCT?

ALGORITHM:

1. PICK $h: \{1, \ldots, 2^b\} \rightarrow \{1, \ldots, n^2\}$ AT RANDOM FROM A $(n/\log n)$-DISPERSING FAMILY.

2. PUT $x_1, \ldots, x_n$ IN BUCKETS ACCORDING TO $h(x_1), \ldots, h(x_n)$. (RADIX SORT)

3. FOR EACH BUCKET, PUT THE ELEMENTS IN A BINARY SEARCH TREE (STOP IF DUPLICATE)

ANALYSIS:
- $O(n/\log n)$ DISTINCT ELEMENTS IN NON-TRIVIAL BUCKETS, $\Rightarrow O(n)$ TIME
- LINEAR SPACE
DISPERsING VS. (ALMOST) UNIVERSAL FAMILIES

**Size**

**Universal:** \( O(r \cdot \log u) \quad \Omega(r \cdot \log r \cdot \log u) \)

**C-Dispersing:** \( O(r \cdot \log d(w)/cn) \quad \Omega(r \cdot \log_{2n} (w)/cn) \)

**Non-constructive**

**Examples, continued**

- **Element Distinctness Algorithm Needs**
  \[ O \left( \log(n^2 \cdot \log n / \log n (2^b)/(n^2 / \log n)) \right) = O(\log b) \]
  Random bits (rather than \( O(\log n + \log b) \)).

- **Bob Needs, for \( c = 1 + \epsilon \),** \( O(\log \log u) \)
  Random bits.
**Explicit Construction I**

**Observation (Fredman et al):**

\[ |h(S)| \geq n - \# \text{Colliding Pairs} \]

(Apract) Universal Families are \(O(1)\)-dispersing
EXPLICIT CONSTRUCTION II\textsubscript{a}

EXTRACTOR (SIMPLIFIED):

FOR ANY $S \subseteq U$ WITH $|S| = n$

$\begin{align*}
&X \in_R S \\
i \in_R I
\end{align*}$

$E = \text{CLOSER TO UNIFORM}$

$E(x_i) \in \mathbb{Z}_1, \ldots, n^3$
EXPLICIT CONSTRUCTION II

- SELECT $f'$ FROM A STRONGLY UNIVERSAL FAMILY, $f': U \to I$.
- SELECT $f''': U \to \{1, \ldots, r/m\}$ FROM A UNIVERSAL FAMILY.

\[
\begin{array}{c}
X \xrightarrow{f'(x)} E^{m} \xrightarrow{E(x, f'(x))} f''(x) \\
\downarrow \quad \text{CONCAT} \\
E(x, f'(x)) \circ f'(x)
\end{array}
\]

ANALYSIS:

LOOK AT COLLISIONS OUTSIDE A SET OF "BAD" POINTS, TO WHICH FEW ELEMENTS MAP.
A DIFFERENT VIEW OF EXTRACTORS

EXTRACTOR WITH OPTIMAL $\log |\mathbb{F}|$

$\uparrow$

SEED LENGTH

$\downarrow$

NON-TRIVIAL

$O(1)$-DISPERISING FAMILY WITH

OPTIMAL SAMPLE COMPLEXITY

WHEN DESIGNING EXPLICIT EXTRACTORS, ONE MAY W.L.O.G.

CONSIDER COLOURFUL STRATEGIES
DETOUR: "ALMOST PERFECT" HASHING

- **MINIMAL PERFECT HASH FUNCTION:**
  \[h: U \rightarrow \{1, \ldots, n\}\]
  \[|h(S)| = |S| = n\]
  
  **PROGRAM SIZE:** \(\Omega(n)\) BITS

- "ALMOST PERFECT" HASH FUNCTION:
  \[h: U \rightarrow \{1, \ldots, n\}\]
  \[|h(S)| \geq \frac{2}{3} |S| = \frac{2}{3} n\]

  **PROGRAM SIZE:** \(\Omega(n)\) BITS
CONCLUSION

DISPERsING HASH FUNCTIONs

+ SEVERAL (MANY) APPLICATIONs
+ POTENTIALLY LOW SAMPLE COMPLEXITY

2. EXPLICIT CONSTRUCTION PROBABLY DIFFICULT (AT LEAST FOR $c = O(1)$).

2. EVER AS FAST AS UNIVERSAL HASH FUNCTIONs?

... THAT'S ALL