Applications of a Splitting Trick

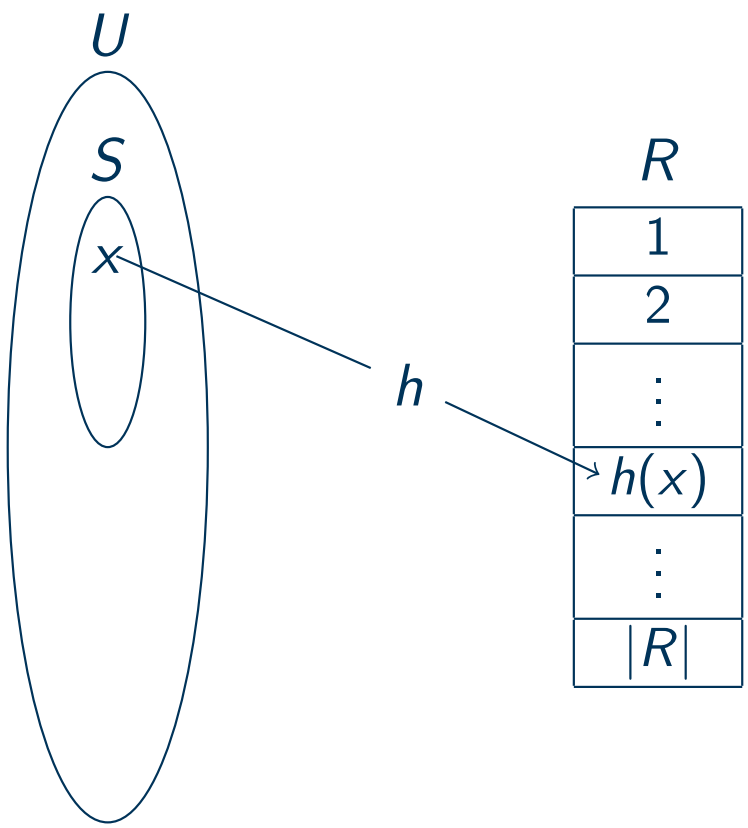
Martin Dietzfelbinger Michael Rink

Technische Universität Ilmenau, Germany

08.07.2009

Full Randomness Assumption

- hash function $h : U \longrightarrow R$ maps keys from a universe U to some range R
- many data structures have been analyzed under the full randomness assumption:
  - values $h(x)$ are fully random (uniform in R, independent) — often it is sufficient if this holds for all x from a key set $S \subseteq U$
  - evaluation time of h is constant
  - h needs no or marginal space
- a theoretical ideal!



Split-and-Share

- Topic: applications of a *simple method*, called split-and-share, for circumventing the "full randomness assumption" when building hashing based data structures
- General approach:
  - split the key set S into disjoint subsets, by a splitting hash function
  - for one fixed subset S_i construct a data structure, using more space / random bits than $|S_i|$
  - replicate the construction for the other subsets, but share some of the random bits among them
  - each small construction consists of an individual part and a shared part
- Result: a tighter overall space bound

Related Work

- The idea of "splitting" is *not new*!
- split-and-share trick:
  - sketched in [Dietzfelbinger and Weidling, 2007] (cuckoo hashing)
  - mentioned in [Fotakis et al., 2005] (hash tables)
  - presented in other works before, e.g. [Hagerup and Tholey, 2001] (minimal perfect hashing)
- the trick has wider applications, we show 2 improved constructions (3 in the paper), concerning: one-probe schemes and uniform hash functions

Before we start ...

- universe U, key set $S \subseteq U$; $|S| = n$, $|U| = n^r$, for $r > 1$
- $[i] = \{1, 2, 3, \ldots, i\}$
- $\mathcal{T} \mathrel{\hat{=}}$ time complexity [word ops], $\mathcal{S} \mathrel{\hat{=}}$ space complexity [bits]
- with high probability (whp) $\mathrel{\hat{=}} 1 - n^{-c}$, for some $c > 0$

Outline

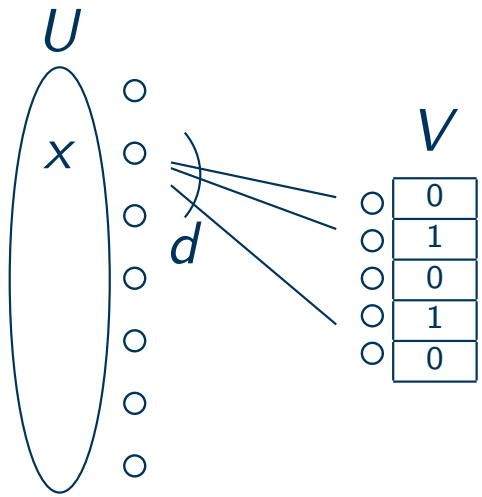
Martin Dietzfelbinger, Michael Rink
Technische Universität Ilmenau, Germany

TECHNISCHE UNIVERSITÄT
ILMENAU

One-probe Schemes

- data structures that answer a "question" with one memory probe
- bit-probe: [Buhrman et al., 2002] showed that there exists schemes which solve the membership problem (" is $x \in S$ "), with two-sided error probability ε, using near optimal space
- word-probe: [Östlin and Pagh, 2002] extended the result to arbitrary functions $f : U \to R$, where a function value is retrieved by probing one word of size $\log |R|$

- main parts: vector V and a d-left regular bipartite graph $G = (U, V, E)$ with certain expansion properties
- *lookup*(x): randomly chose a neighbor $y \in \Gamma(\{x\})$ and calculate the answer from $V[y]$



One-probe Schemes

- required expansion: $\forall T \subseteq U, |T| \leq 2n : |\Gamma(T)| \geq (1 - \frac{\varepsilon}{2}) \cdot |T|$
- probabilistic method: such graphs exist for parameters: $|V| = O(\frac{n \cdot d}{\varepsilon})$, $d = \Theta(\frac{\log |U|}{\varepsilon})$
- explicit constructions, like [Ta-Shma, 2002], need either more space or have non-constant evaluation time
- hence the expander is assumed to be given for free (counts not towards the space consumption)

Our Result

Theorem

Let n be sufficiently large. For every $\varepsilon \in (0,1)$ there exists an *explicit* one-probe membership tester with two sided error ε and:

(i) size $O\left(\frac{n \log |U|}{\varepsilon^2}\right)$ and

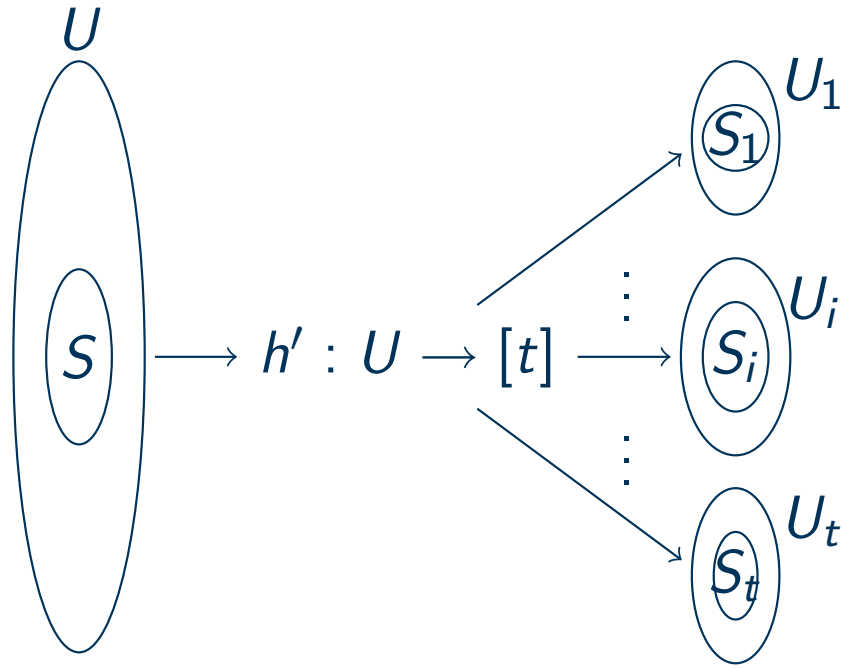
(ii) constant evaluation time.

The result can be transferred to the one-probe dictionary of [Östlin and Pagh, 2002].

Splitting the Key Set Evenly

- randomly choose h' from hash class $\mathcal{R} \subseteq \{h \mid h : U \to [t]\}$ [Dietzfelbinger and Meyer auf der Heide, 1990] , $t = n^{1-\delta}$
- split the key set S into disjoint subsets, via $S_i := \{x \in S \mid h'(x) = i\}$

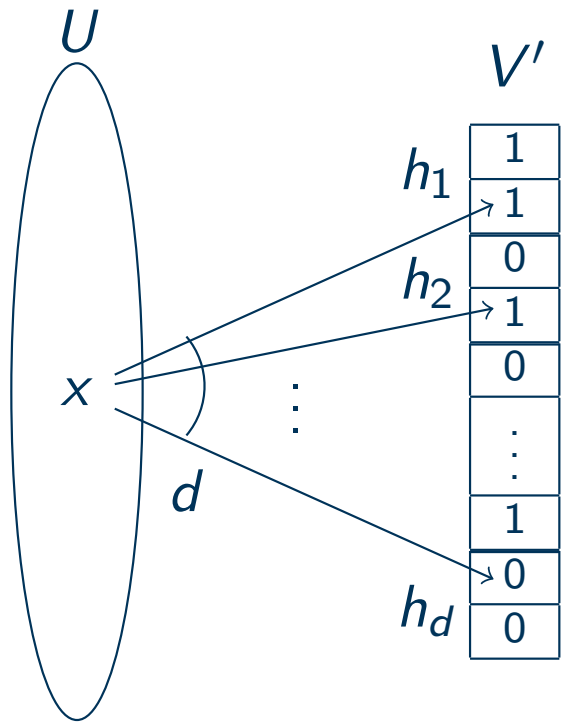
- $S(h') = o(n)$, $\mathcal{T}_{\text{eval}}(h') = O(1)$
- whp we have: $|S_i| \leq (1+\lambda)n^{\delta}$, for all $i \in [t]$
- since S is static, we can choose a new h' until this bound holds



Virtual Expander Graph via Siegel's Hash Class

- consider a universe with at most $(1+\lambda)n^{\delta}$ key elements
- randomly choose d hash functions $h_k : U \to [s'/d]$, $k \in [d]$, from a $2 \cdot (1+\lambda)n^{\delta}$-wise independent hash class [Siegel, 2004]
- $\mathcal{S}(h_k) = o(n^{\delta+\nu})$, for some $\nu \in (0, 1-\delta)$, $\mathcal{T}_{\text{eval}}(h_k) = O(1)$

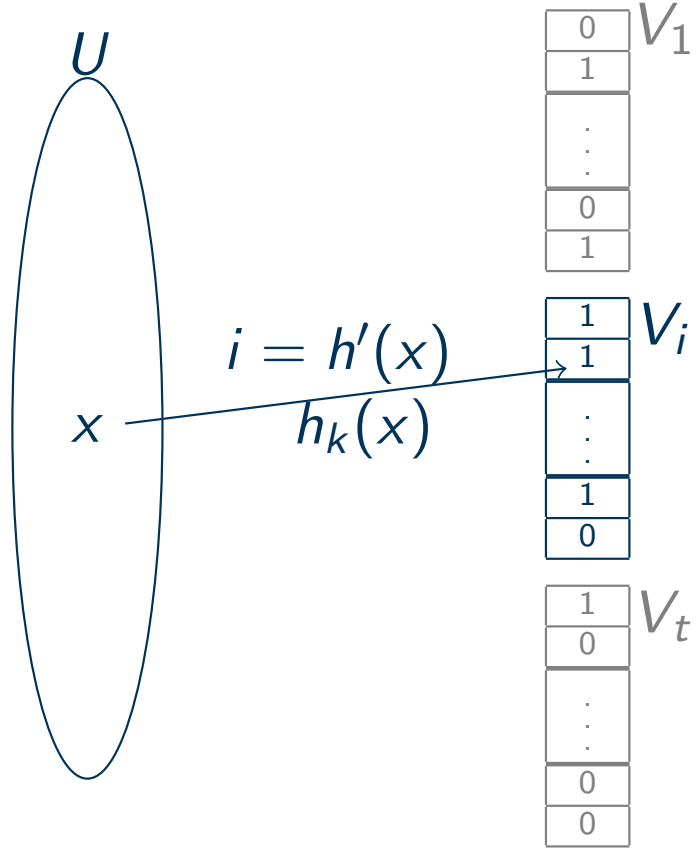
- build a graph $G = (U, V', E)$ and use the hash functions to obtain the edge set $E = \left\{ \left(x, (k-1) \cdot \frac{s'}{d} + h_k(x)\right) \mid k \in [d] \right\}$
- let $d = \Theta\left(\frac{\log |U|}{\varepsilon}\right)$ and $s' = O\left(\frac{n^{\delta} \cdot d}{\varepsilon}\right)$, then it holds: G has the desired expansion property whp



Sharing the Expander

- replicate the construction for each S_i, $i \in [n^{1-\delta}]$
- obtain a storing scheme for S by juxtaposing the distinct storing schemes for all S_i

- the expander / the hash functions can be shared, since the domain of the h_k is U and the expansion holds *for all* $T \subseteq U$ with $|T| \leq 2 \cdot (1+\lambda)n^{\delta}$
- overall space usage: $o(n)$ for the hash functions, $O\left(\frac{n \log |U|}{\varepsilon^2}\right)$ for the vectors V_i
- lookup time: $O(1)$ $(\mathcal{T}_{\text{eval}}(h') = \mathcal{T}_{\text{eval}}(h_k) = O(1))$



1 Constant Time Explicit One-probe Schemes

2 Uniform Hashing in Close to Optimal Space

Martin Dietzfelbinger, Michael Rink
Technische Universität Ilmenau, Germany

TECHNISCHE UNIVERSITÄT
ILMENAU

Simulating Full Randomness

- [Östlin and Pagh, 2003] and [Dietzfelbinger and Woelfel, 2003] gave solutions for the problem of constructing a hash function $h : U \to R$ that is fully random on S whp, needs space $O(n \log |R|)$, and has constant evaluation time
- [Pagh and Pagh, 2008] reduced the space bound to $(1 + \sigma)n \log |R|$, with evaluation time $O(1/\sigma^2)$, for $\sigma > 0$
- Note: the information theoretical space minimum is $n \log |R|$

Our Result

Main Theorem

For arbitrary $\sigma \in (0,1)$ there is a hash class $\mathcal{H} \subseteq \{h \mid U \to R\}$ such that for arbitrary $S \subseteq U$, with $|S| = n$, and $h \in \mathcal{H}$ chosen at random the following holds:

(i) h behaves fully randomly on S whp;
(ii) the space usage for h is asymptotically $(1+\sigma)n \log |R|$;
(iii) the evaluation time of h is $O(\log(1/\sigma))$.

The advantages over the construction from [Pagh and Pagh, 2008] are the exponentially faster evaluation time and the (we think) simpler structure.

Martin Dietzfelbinger, Michael Rink
Technische Universität Ilmenau, Germany

TECHNISCHE UNIVERSITÄT
ILMENAU

Split and Provide Fully Random Functions on S_i

- randomly choose a function $h' \in \mathcal{R}$ and split S into $t = n^{1-\delta}$ disjoint subsets via $S_i = \{x \in S \mid h'(x) = i\}$
- $\mathcal{T}_{\text{eval}}(h') = O(1)$, $\mathcal{S}(h') = o(n)$
- S is not given, therefore: $|S_i| \leq (1+\lambda)n^{\delta}$ only whp
- it is easy to get a function $\mathfrak{h} : U \to [s]$ that is fully random on S_i, with $\mathcal{S}(\mathfrak{h}) = O(n^{\delta+\nu})$, for $\nu \in (0, 1-\delta)$, and $\mathcal{T}_{\text{eval}}(\mathfrak{h}) = O(1)$

Linearly Independent Vectors

$$S_i \longrightarrow \varphi : U \to \{0,1\}^s \to M_i = \left.\begin{bmatrix} 0100000\ldots0010 \\ 1000100\ldots0000 \\ \vdots \qquad\qquad \vdots \\ 0000010\ldots0100 \end{bmatrix}\right\} |S_i|$$

- consider one subset S_i of size $|S_i| \leq (1+\lambda)n^\delta$
- use b independent functions $\mathfrak{h}_1, \mathfrak{h}_2, \ldots, \mathfrak{h}_b$ to obtain a mapping $\varphi : x \mapsto v_x \subseteq \{0,1\}^s$ with $||v_x||_1 = b$
- the probability that matrix M_i, with rows $v_x, x \in S_i$, has full row rank can be obtained as function of $s, |S_i|$ and b
- [Calkin, 1997]: to get full row rank whp, it is sufficient that:

$$s \geq (1+\sigma)n^\delta, \text{ for } \sigma > \lambda, \text{ and } b = O(\log(1/\sigma))$$

- $\mathcal{S}(\varphi) = O(n^{\delta+\nu})$, $\mathcal{T}_{\text{eval}}(\varphi) = O(\log(1/\sigma))$

Stochastic Independence

- create a vector $V_i \in R^s$ filled with fully random elements from R
- the space usage for V_i is $(1+\sigma)n^\delta \log |R|$
- with φ and V_i we get a new hash function $h_i : U \to R$ via

$$h_i(x) = <v_x, V_i>$$

- from the linear independence of the $v_x, x \in S_i$, easily follows stochastic independence
- that is h_i is uniform on S_i whp

Martin Dietzfelbinger, Michael Rink
Technische Universität Ilmenau, Germany

TECHNISCHE UNIVERSITÄT
ILMENAU

Sharing the Hash Functions

- until now: we have one uniform hash function h_i for one subset S_i
- replication of the construction for all $n^{1-\delta}$ subsets S_i doesn't work since $\mathcal{S}(\varphi) = O(n^{\delta+\nu})$, for some $\nu > 0$
- observation: φ (functions $\mathfrak{h}_1, \ldots, \mathfrak{h}_b$) can be shared among the S_i
- the vector V_i cannot be shared, one must choose a separate one for each subset S_i
- whp we get a function that is fully random on S via:

$$h(x) = <v_x, V_i> \in R, h'(x) = i \in [n^{1-\delta}], \varphi(x) = v_x \in \{0,1\}^s$$

- overall space usage: $o(n)$ for the hash functions, $(1+\sigma)n \log |R|$ for the $n^{1-\delta}$ vectors
- evaluation time: $O(\log(1/\sigma))$

Thank you!

Questions?


Buhrman, H., Miltersen, P. B., Radhakrishnan, J., and Venkatesh, S. (2002).
Are bitvectors optimal?
SIAM J. Comput., 31(6):1723–1744.


Calkin, N. J. (1997).
Dependent sets of constant weight binary vectors.
Combinatorics, Probability & Computing, 6(3):263–271.


Dietzfelbinger, M. and Meyer auf der Heide, F. (1990).
A new universal class of hash functions and dynamic hashing in real time.
In *Proc. 17th ICALP*, LNCS, pages 6–19. Springer-Verlag.


Dietzfelbinger, M. and Weidling, C. (2007).
Balanced allocation and dictionaries with tightly packed constant size bins.
Theor. Comput. Sci., 380(1-2):47–68.


Dietzfelbinger, M. and Woelfel, P. (2003).
Almost random graphs with simple hash functions.
In *Proc. 35th STOC*, pages 629–638. ACM.


Fotakis, D., Pagh, R., Sanders, P., and Spirakis, P. G. (2005).
Space efficient hash tables with worst case constant access time.
Theory Comput. Syst., 38(2):229–248.


Hagerup, T. and Tholey, T. (2001).
Efficient minimal perfect hashing in nearly minimal space.
In *Proc. 18th STACS*, LNCS, pages 317–326. Springer-Verlag.


Östlin, A. and Pagh, R. (2002).
One-probe search.
In *Proc. 29th ICALP*, LNCS, pages 439–450. Springer-Verlag.


Östlin, A. and Pagh, R. (2003).
Uniform hashing in constant time and linear space.
In *Proc. 35th STOC*, pages 622–628. ACM.


Pagh, A. and Pagh, R. (2008).
Uniform hashing in constant time and optimal space.
SIAM J. Comput., 38(1):85–96.


Siegel, A. (2004).
On universal classes of extremely random constant-time hash functions.
SIAM J. Comput., 33(3):505–543.


Ta-Shma, A. (2002).
Storing information with extractors.
Inf. Process. Lett., 83(5):267–274.