Tabulation-Based Hashing

Mihai Pătrașcu       Mikkel Thorup

April 23, 2010
Applications of Hashing

Hash tables:
- chaining

\[ x \rightarrow \bullet \rightarrow a \rightarrow t \]
\[ \rightarrow \bullet \rightarrow v \]
\[ \rightarrow \bullet \rightarrow f \rightarrow s \rightarrow r \]
Applications of Hashing

Hash tables:
- chaining
- linear probing

\[
\begin{array}{ccccccc}
\bullet & m & \bullet & c & a & f & t & \bullet & x & y \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & t \\
\end{array}
\]
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

\[ x \rightarrow a \]
\[ b \]
\[ s \]
\[ z \]
\[ f \]
\[ r \]
\[ b \]
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

\[ x \rightarrow \]

\[ \begin{array}{c}
\text{a} \\
\bullet \\
\text{y} \\
\text{w} \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\text{s} \\
\text{z} \\
\text{f} \\
\bullet \\
\text{r} \\
\text{b} \\
\end{array} \]
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

\[ x \rightarrow \begin{array}{c}
\text{a} \\
\bullet \\
\bullet \\
\bullet \\
\text{y} \\
\text{w} \\
\bullet \\
\bullet \\
\end{array} \quad \begin{array}{c}
\bullet \\
\text{s} \\
\text{z} \\
\text{f} \\
\bullet \\
\text{r} \\
\text{b} \\
\end{array} \]
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

Sketching and streaming:
- moment estimation: \( F_2(\bar{x}) = \sum_i x_i^2 \)
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

Sketching and streaming:
- moment estimation: \( F_2(\overline{x}) = \sum_i x_i^2 \)
- sketch \( A \) and \( B \) to later find \( \frac{|A \cap B|}{|A \cup B|} \)
Applications of Hashing

Hash tables:
- chaining
- linear probing
- cuckoo hashing

Sketching and streaming:
- moment estimation: $F_2(\bar{x}) = \sum_i x_i^2$
- sketch $A$ and $B$ to later find $\frac{|A \cap B|}{|A \cup B|}$
- etc, etc.
Minwise independence

Hash each set through \( h \), keen the minimum

\[
\frac{|A \cap B|}{|A \cup B|} = \Pr_h[\min h(A) = \min h(B)]
\]

- repeat with \( k \) different \( h \);
- keep smallest \( k \) items with one \( h \)
Minwise independence

Hash each set through $h$, keen the minimum

$$\frac{|A \cap B|}{|A \cup B|} = \Pr_{h}[\min h(A) = \min h(B)]$$

- repeat with $k$ different $h$;
- keep smallest $k$ items with one $h$

The guarantee we need on $h$: “minwise independence”

$$(\forall) S, x : \Pr[x < \min h(S)] = \frac{1}{|S|+1}$$
Minwise independence

Hash each set through $h$, keep the minimum

$$\frac{|A \cap B|}{|A \cup B|} = \Pr_h[\min h(A) = \min h(B)]$$

- repeat with $k$ different $h$;
- keep smallest $k$ items with one $h$

The guarantee we need on $h$: “minwise independence”

$$(\forall) S, x : \ Pr[x < \min h(S)] = \frac{1}{|S|+1}$$

Not feasible... Approximate:

$$(\forall) S, x : \ Pr[x < \min h(S)] = \frac{1+\epsilon}{|S|+1}$$

Approximation $= \epsilon + f(\# \text{ repetitions})$
Minwise independence

Hash each set through $h$, keen the minimum

$$\frac{|A \cap B|}{|A \cup B|} = \Pr[h(\min h(A)) = \min h(B)]$$

- repeat with $k$ different $h$;
- keep smallest $k$ items with one $h$

The guarantee we need on $h$: “minwise independence”

$$(\forall)S, x : \quad \Pr[x < \min h(S)] = \frac{1}{|S|+1}$$

Not feasible... Approximate:

$$(\forall)S, x : \quad \Pr[x < \min h(S)] = \frac{1{\pm}\varepsilon}{|S|+1}$$

Approximation $= \varepsilon + f(\# \text{repetitions})$

NB: for weighted $A, B$ the generalization is priority sampling
A family $\mathcal{H} = \{h : [u] \to [b]\}$ is $k$-independent iff:

- $(\forall) x \in u, \ h(x) \text{ is uniform in } [b]$;
- $(\forall) x_1, \ldots, x_k \in [u], \ h(x_1), \ldots, h(x_k) \text{ are independent.}$
A family $\mathcal{H} = \{h : [u] \to [b]\}$ is $k$-independent iff:

1. $(\forall) x \in u, \ h(x) \text{ is uniform in } [b]$;
2. $(\forall) x_1, \ldots, x_k \in [u], \ h(x_1), \ldots, h(x_k) \text{ are independent.}$

Prototypical example: degree $k$ polynomial

- $u$ prime;
- choose $a_0, a_1, \ldots, a_{k-1}$ randomly in $[u]$;
- $h(x) = (a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}) \mod b$. 
### How much independence?

<table>
<thead>
<tr>
<th>Method</th>
<th>Independence</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaining</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Linear probing</td>
<td>$\leq 5$</td>
<td>[Pagh, Ružić'07]</td>
</tr>
<tr>
<td></td>
<td>$\geq 5$</td>
<td>[PT'10]</td>
</tr>
<tr>
<td>Cuckoo hashing</td>
<td>$O(\lg n)$</td>
<td>$\geq 6$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Cohen, Kane'05]</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>[Thorup, Zhang'04]</td>
</tr>
<tr>
<td>$\varepsilon$-minwise indep.</td>
<td>$O(\lg \frac{1}{\varepsilon})$</td>
<td>[Indyk'99]</td>
</tr>
<tr>
<td></td>
<td>$\Omega(\lg \frac{1}{\varepsilon})$</td>
<td>[PT'10]</td>
</tr>
</tbody>
</table>
## How much independence?

<table>
<thead>
<tr>
<th>Method</th>
<th>Independence</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaining</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Linear probing</td>
<td>$\leq 5$</td>
<td>$\geq 5$ [Pagh$^2$, Ružić'07]</td>
</tr>
<tr>
<td>Cuckoo hashing</td>
<td>$O(\lg n)$</td>
<td>$\geq 6$ [Cohen, Kane'05]</td>
</tr>
<tr>
<td>$F_2^2$ estimation</td>
<td>4</td>
<td>[Thorup, Zhang'04]</td>
</tr>
<tr>
<td>$\epsilon$-minwise indep.</td>
<td>$O(\lg \frac{1}{\epsilon})$</td>
<td>$\Omega(\lg \frac{1}{\epsilon})$ [PT'10]</td>
</tr>
</tbody>
</table>

Chaining: $\text{time} = \#\{x \mid h(x) = h(\text{query})\}$

$E[\text{time}] = n \cdot Pr[h(x) = h(\text{query})] = n \cdot \frac{1}{b} = O(1)$
### How much independence?

<table>
<thead>
<tr>
<th>Method</th>
<th>Independence Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chaining</td>
<td>2</td>
</tr>
<tr>
<td>Linear probing</td>
<td>( \leq 5 ) [Pagh(^2), Ružič’07] ( \geq 5 ) [PT’10]</td>
</tr>
<tr>
<td>Cuckoo hashing</td>
<td>( O(\lg n) ) ( \geq 6 ) [Cohen, Kane’05]</td>
</tr>
<tr>
<td>( F_2 ) estimation</td>
<td>4                     [Thorup, Zhang’04]</td>
</tr>
<tr>
<td>( \varepsilon )-minwise indep.</td>
<td>( O(\lg \frac{1}{\varepsilon}) ) [Indyk’99] ( \Omega(\lg \frac{1}{\varepsilon}) ) [PT’10]</td>
</tr>
</tbody>
</table>

**Chaining:**

\[
\text{time} = \# \{ x \mid h(x) = h(\text{query}) \} \\
\mathbb{E}[\text{time}] = n \cdot \Pr[h(x) = h(\text{query})] = n \cdot \frac{1}{b} = O(1)
\]

**Cuckoo hashing:**

components in random graphs have size \( O(\lg n) \)
## How much independence?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chaining</strong></td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Linear probing</strong></td>
<td>≤ 5</td>
<td>[Pagh², Ružić’07]</td>
<td>≥ 5</td>
</tr>
<tr>
<td><strong>Cuckoo hashing</strong></td>
<td>$O(\lg n)$</td>
<td>≥ 6</td>
<td>[Cohen, Kane’05]</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>[Thorup, Zhang’04]</td>
<td></td>
</tr>
<tr>
<td><strong>$\varepsilon$-minwise indep.</strong></td>
<td>$O(\lg \frac{1}{\varepsilon})$ [Indyk’99]</td>
<td>$\Omega(\lg \frac{1}{\varepsilon})$ [PT’10]</td>
<td></td>
</tr>
</tbody>
</table>

**Chaining:**

\[
\text{time} = \#\{x \mid h(x) = h(\text{query})\}
\]

\[
\mathbb{E}[\text{time}] = n \cdot \Pr[h(x) = h(\text{query})] = n \cdot \frac{1}{b} = O(1)
\]

**Cuckoo hashing:**

components in random graphs have size $O(\lg n)$

**Minwise independence:**

$k$-level inclusion/exclusion estimates probabilities to $\pm 2^{-k}$. 
Linear probing
Implementing \( k \)-independence

Goals:
- constant time for \( \omega(1) \) independence
- practical solution?

Lower bound [Siegel'90s]:
With space \( u \), query time \( \geq \min \{ k, q \} \).

Tabulation hashing:
- \( q \) basic characters: \( x \mapsto (x_1, \ldots, x_q) \)
- \( d \) derived characters: \( y_i = f_i(x_1, \ldots, x_q) \)
- store \( q + d \) random tables \( T_i[u/q] \)
  
  \[ h(x) = T_1[q_1] \oplus \cdots \oplus T_q[x_q] \oplus T_{q+1}[y_1] \oplus \cdots \]
Implementing $k$-independence

Goals:

- constant time for $\omega(1)$ independence
- practical solution?

Lower bound [Siegel’90s]:

With space $u^{1/q}$, query time $\geq \min\{k, q\}$. 

Tabulation hashing:

- basic characters: $x \mapsto (x_1, \ldots, x_q)$
- derived characters: $y_i = f_i(x_1, \ldots, x_q)$
- store $q+d$ random tables $T_i[u^{1/q}]$
- $h(x) = T_1[q] \oplus \cdots \oplus T_q[x_q] \oplus T_{q+1}[y_1] \oplus \cdots$
Implementing $k$-independence

Goals:
- constant time for $\omega(1)$ independence
- practical solution?

Lower bound [Siegel’90s]:
With space $u^{1/q}$, query time $\geq \min\{k, q\}$.

Tabulation hashing:
- $q$ basic characters: $x \mapsto (x_1, \ldots, x_q)$
- $d$ derived characters: $y_i = f_i(x_1, \ldots, x_q)$
- store $q + d$ random tables $T_i[u^{1/q}]$
- $h(x) = T_1[q_1] \oplus \cdots \oplus T_q[x_q] \oplus T_{q+1}[y_1] \oplus \cdots$
## Tabulation-Based Hashing

<table>
<thead>
<tr>
<th></th>
<th>Independence</th>
<th># characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carter, Wegman’77</td>
<td>3</td>
<td>( q ) (⋆)</td>
</tr>
<tr>
<td>Siegel’90s</td>
<td>( n^{\Omega(1)} )</td>
<td>( q^{O(q)} )</td>
</tr>
<tr>
<td>Dietzfel, Woelfel’03</td>
<td>( k )</td>
<td>( k \cdot q )</td>
</tr>
<tr>
<td>Thorup, Zhang’04</td>
<td>( k )</td>
<td>((k - 1)(q - 1))</td>
</tr>
<tr>
<td>Thorup, Zhang’10</td>
<td>5</td>
<td>( 2q - 1 )</td>
</tr>
<tr>
<td>recent</td>
<td>( \omega(1) )</td>
<td>( O(q^2) )</td>
</tr>
</tbody>
</table>

(⋆) simple tabulation (no derived characters)
Peeling \((q = 2, k = 3)\)

\((x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2]\)

Let’s prove independence of \(\{a, b, c\}\).
Peeling \((q = 2, k = 3)\)

\[(x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2]\]

Let’s prove independence of \(\{a, b, c\}\).

<table>
<thead>
<tr>
<th>(a_1)</th>
<th>(a_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(b_1)</td>
<td>(b_2)</td>
</tr>
<tr>
<td>(c_1)</td>
<td>(c_2)</td>
</tr>
</tbody>
</table>

**Peeling:**
If \(a_i\) is unique \((a_i \neq b_i, c_i)\)
\[\implies h(a) \text{ independent of } h(b), h(c)\]
Peeling \((q = 2, k = 3)\)

\((x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2]\)

Let’s prove independence of \(\{a, b, c\}\).

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)</td>
<td>(a_2)</td>
</tr>
<tr>
<td>(b_1)</td>
<td>(b_2)</td>
</tr>
<tr>
<td>(c_1)</td>
<td>(c_2)</td>
</tr>
</tbody>
</table>

Peeling:
If \(a_i\) is unique \((a_i \neq b_i, c_i)\)

\(\implies h(a)\) independent of \(h(b), h(c)\)

Any set of \(\leq 3\) keys is peelable, thus independent.
Peeling \((q = 2, k = 4)\)

\((x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2] \oplus T_3[x_1 + x_2]\)

Let’s prove \(\{a, b, c, d\}\) are independent.

- if we can peel, reduce to 3-independence.
- the only non-peelable configuration:
Peeling \( (q = 2, k = 4) \)

\[(x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2] \oplus T_3[x_1 + x_2] \]

Let’s prove \( \{a, b, c, d\} \) are independent.

- if we can peel, reduce to 3-independence.
- the only non-peelable configuration:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>(s)</td>
<td>(x + s)</td>
</tr>
<tr>
<td>(x)</td>
<td>(t)</td>
<td>(x + t)</td>
</tr>
<tr>
<td>(y)</td>
<td>(s)</td>
<td>(y + s)</td>
</tr>
<tr>
<td>(y)</td>
<td>(t)</td>
<td>(y + t)</td>
</tr>
</tbody>
</table>
Peeling \((q = 2, k = 4)\)

\[(x_1, x_2) \mapsto T_1[x_1] \oplus T_2[x_2] \oplus T_3[x_1 + x_2]\]

Let’s prove \(\{a, b, c, d\}\) are independent.
- if we can peel, reduce to 3-independence.
- the only non-peelable configuration:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>(s)</td>
<td>(x + s)</td>
</tr>
<tr>
<td>(x)</td>
<td>(t)</td>
<td>(x + t)</td>
</tr>
<tr>
<td>(y)</td>
<td>(s)</td>
<td>(y + s)</td>
</tr>
<tr>
<td>(y)</td>
<td>(t)</td>
<td>(y + t)</td>
</tr>
</tbody>
</table>

Only possible equalities: \(x + s = y + t\) or \(x + t = y + s\).
Both cannot hold, so we have peeling in derived character.
Theorem: Any 4-independent tabulation is 5-independent! Among any 5 keys, one is independent in the basic characters.
5-independence [PT’10]

**Theorem:** Any 4-independent tabulation is 5-independent!
Among any 5 keys, one is independent in the basic characters.
- any unique character $\Rightarrow$ peel
Theorem: Any 4-independent tabulation is 5-independent! Among any 5 keys, one is independent in the basic characters.

- any unique character $\Rightarrow$ peel
- otherwise, any dimension looks like: three “0”, two “1”
Theorem: Any 4-independent tabulation is 5-independent!
Among any 5 keys, one is independent in the basic characters.

- any unique character $\Rightarrow$ peel
- otherwise, any dimension looks like: three “0”, two “1”
- two columns have Hamming distance = 4

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$b$</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>$c$</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$d$</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>$e$</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>
Theorem: Any 4-independent tabulation is 5-independent! Among any 5 keys, one is independent in the basic characters.

- any unique character ⇒ peel
- otherwise, any dimension looks like: three “0”, two “1”
- two columns have Hamming distance = 4

\[
\begin{array}{c|c|c|c|c|c}
\hline
a & 0 & 1 & ... \\
\hline
b & 0 & 1 & ... \\
\hline
c & 1 & 0 & ... \\
\hline
d & 1 & 0 & ... \\
\hline
e & 1 & 1 & ... \\
\hline
\end{array}
\]

- all columns at Hamming distance = 2

\[
\begin{array}{c|c|c|c|c|c}
\hline
a & 0 & 0 & 0 & ... \\
\hline
b & 0 & 1 & 1 & ... \\
\hline
c & 1 & 0 & 1 & ... \\
\hline
d & 1 & 1 & 0 & ... \\
\hline
e & 1 & 1 & 1 & ... \\
\hline
\end{array}
\]

NB: \( h(a) = h(b) \oplus h(c) \oplus h(d) \)

If \( e \) independent of \( b, c, d \), also independent of \( f(b, c, d) \).
## Putting it together

### Algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>$q$</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>$(k - 1)(q - 1)$</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$</td>
<td>$q^O(q)$</td>
</tr>
</tbody>
</table>

### Scheme

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>[Thorup, Zhang’04]</td>
<td>4</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>“4 $\rightarrow$ 5”</td>
<td>5</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>[Thorup, Zhang’04]</td>
<td>$k$</td>
<td>$(k - 1)(q - 1)$</td>
</tr>
<tr>
<td>[Siegel’90s]</td>
<td>$n^{\Omega(1)}$</td>
<td>$q^{O(q)}$</td>
</tr>
</tbody>
</table>
## Putting it together

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>[Thorup, Zhang’04]</td>
<td>4</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>“4 $\rightarrow$ 5”</td>
<td>5</td>
<td>$2q - 1$</td>
</tr>
<tr>
<td>[Thorup, Zhang’04]</td>
<td>$k$</td>
<td>$(k - 1)(q - 1)$</td>
</tr>
<tr>
<td>[Siegel’90s]</td>
<td>$n^{\Omega(1)}$</td>
<td>$q^{O(q)}$</td>
</tr>
</tbody>
</table>

What exactly are we doing here?
### The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
<td>maybe...</td>
<td>3</td>
<td>q</td>
</tr>
</tbody>
</table>
# The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
</tbody>
</table>

**Simple tabulation:**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
</table>
| preserves 4th moment bound $\Rightarrow$ $F_2$ estimation $1$-in-$5$ indep. $\Rightarrow$ linear probing in expected $O(1)$ time
| $\varepsilon$-minwise independence with $\varepsilon = \varepsilon(n) = o(1)$.
| Chernoff concentration $\Rightarrow O(lg n)$ query time w.h.p.
| preserve moments in linear probing, chaining: $F_p$ w/ simple tabulation $= F_p$ w/ truly random $+ o(1)$

Mihai Pătrașcu, Mikkel Thorup

Tabulation-Based Hashing
## The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
</tbody>
</table>

**Simple tabulation:**

- Preserves 4th moment bound
- $\Rightarrow F_2$ estimation
- 1-in-5 independence $\Rightarrow$ linear probing in expected $O(1)$ time
- Minwise independence with $\varepsilon = \varepsilon(n) = o(1)$.
- Chernoff concentration $\Rightarrow O(\lg n)$ query time w.h.p.
- Preserves moments in linear probing, chaining: $F_p$ w/ simple tabulation = $F_p$ w/ truly random + $o(1)$
### The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
<td>maybe...</td>
<td>3</td>
<td>q</td>
</tr>
</tbody>
</table>

**Simple tabulation:**
- preserves 4\textsuperscript{th} moment bound $\Rightarrow F_2$ estimation
The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(1g \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(1g n)$?</td>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
</tbody>
</table>

Simple tabulation:
- preserves 4$^{th}$ moment bound $\Rightarrow F_2$ estimation
- 1-in-5 indep. $\Rightarrow$ linear probing in expected $O(1)$ time
The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>simple tabulation</td>
<td>3</td>
<td>$q$</td>
</tr>
<tr>
<td>maybe...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Simple tabulation:
- preserves $4^{th}$ moment bound $\Rightarrow F_2$ estimation
- 1-in-5 indep. $\Rightarrow$ linear probing in expected $O(1)$ time
- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$. 

Mihai Pătraşcu, Mikkel Thorup
Tabulation-Based Hashing
The Power of Simple Tabulation

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(\lg \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(\lg n)$?</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>maybe...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Simple tabulation:

- preserves 4th moment bound $\Rightarrow F_2$ estimation
- 1-in-5 indep. $\Rightarrow$ linear probing in expected $O(1)$ time
- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$.
- Chernoff concentration $\Rightarrow O(\lg n)$ query time w.h.p.
**The Power of Simple Tabulation**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Indep.</th>
<th>Scheme</th>
<th>Indep.</th>
<th>Characters</th>
</tr>
</thead>
<tbody>
<tr>
<td>chaining</td>
<td>2</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$F_2$ estimation</td>
<td>4</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>linear probing</td>
<td>5</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>$\varepsilon$-minwise</td>
<td>$\Theta(1g \frac{1}{\varepsilon})$</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td>cuckoo hashing</td>
<td>$O(1g n)$?</td>
<td>simple tabulation</td>
<td>3</td>
<td>q</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maybe...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Simple tabulation:
- preserves 4\(^{th}\) moment bound $\Rightarrow F_2$ estimation
- 1-in-5 indep. $\Rightarrow$ linear probing in expected $O(1)$ time
- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$.
- Chernoff concentration $\Rightarrow O(1g n)$ query time w.h.p.
- preserve moments in linear probing, chaining: $F_p$ w/ simple tabulation $= F_p$ w/ truly random $+ o(1)$
Simple tabulation as a PRG

Pseudorandom numbers \( \approx h(0), h(1), h(1), \ldots \)

<table>
<thead>
<tr>
<th>( g(0) )</th>
<th>( f(0) )</th>
<th>( h(0) )</th>
<th>( h(1) )</th>
<th>( \ldots )</th>
<th>( h(S - 1) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( g(1) )</td>
<td>( f(1) )</td>
<td>( h(S) )</td>
<td>( h(S + 1) )</td>
<td>( \ldots )</td>
<td>( h(2S - 1) )</td>
</tr>
<tr>
<td>( g(2) )</td>
<td>( f(S - 1) )</td>
<td>( h(2S) )</td>
<td>( h(2S + 1) )</td>
<td>( \ldots )</td>
<td>( h(3S - 1) )</td>
</tr>
</tbody>
</table>

\( \ldots \)
Simple tabulation as a PRG

Pseudorandom numbers \( \approx h(0), h(1), h(1), \ldots \)

<table>
<thead>
<tr>
<th>(f(0))</th>
<th>(f(1))</th>
<th>\ldots</th>
<th>(f(S - 1))</th>
</tr>
</thead>
<tbody>
<tr>
<td>(g(0))</td>
<td>(h(0))</td>
<td>(h(1))</td>
<td>\ldots</td>
</tr>
<tr>
<td>(g(1))</td>
<td>(h(S))</td>
<td>(h(S + 1))</td>
<td>\ldots</td>
</tr>
<tr>
<td>(g(2))</td>
<td>(h(2S))</td>
<td>(h(2S + 1))</td>
<td>\ldots</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
</tbody>
</table>

Use \( S = O(\lg n) \) truly random numbers. Compute \( \lg n \) independent \( g(\cdot) \), but rarely.

PRG has:
- concentration (load balancing, \ldots)
- minwise independence (treaps, \ldots)
Wait, there’s more!

What more can we ask for?
What more can we ask for?

- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$
  \[\rightarrow\] minwise independence with $\varepsilon = \varepsilon(u) = o(1)$
Wait, there’s more!

What more can we ask for?

- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$
  $\rightarrow$ minwise independence with $\varepsilon = \varepsilon(u) = o(1)$

- linear probing/chaining $O(1)$ exp. time, $O(\lg n)$ w.h.p.
  $\rightarrow$ for $k \geq \lg n$, any $k$ operations work in $O(k)$ time w.h.p.
Wait, there’s more!

What more can we ask for?

- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$
  $\rightarrow$ minwise independence with $\varepsilon = \varepsilon(u) = o(1)$

Experiments: $\{0, 1\}^q$ is counterexample?

- linear probing/chaining $O(1)$ exp. time, $O(\lg n)$ w.h.p.
  $\rightarrow$ for $k \geq \lg n$, any $k$ operations work in $O(k)$ time w.h.p.
Wait, there’s more!

What more can we ask for?

- minwise independence with \( \varepsilon = \varepsilon(n) = o(1) \)
  \( \rightarrow \) minwise independence with \( \varepsilon = \varepsilon(u) = o(1) \)
  Experiments: \( \{0, 1\}^q \) is counterexample?

- linear probing/chaining \( O(1) \) exp. time, \( O(\lg n) \) w.h.p.
  \( \rightarrow \) for \( k \geq \lg n \), any \( k \) operations work in \( O(k) \) time w.h.p.
  We only get \( k = n^\varepsilon \) for any \( \varepsilon > 0 \).
  Counterexample for \( n^{o(1)} \).

Simple++:

\[ h_1: [u] \rightarrow [b], \quad h_2: [u] \rightarrow [u_{1/q}] \]

Just simple tabulation. . .

\[ h(x) = h_1(x) \oplus T[h_2(x)] \]

All previous properties, plus:

- minwise independence with \( \varepsilon(u) = o(1) \)
- linear probing/chaining with buffer \( k = O(\lg n) \).
Wait, there’s more!

What more can we ask for?

- minwise independence with $\varepsilon = \varepsilon(n) = o(1)$
  $\rightarrow$ minwise independence with $\varepsilon = \varepsilon(u) = o(1)$
Experiments: $\{0, 1\}^q$ is counterexample?

- linear probing/chaining $O(1)$ exp. time, $O(lg n)$ w.h.p.
  $\rightarrow$ for $k \geq lg n$, any $k$ operations work in $O(k)$ time w.h.p.
We only get $k = n^\varepsilon$ for any $\varepsilon > 0$.
Counterexample for $n^{o(1)}$.

Simple++:

- $h_1 : [u] \rightarrow [b]$, $h_2 : [u] \rightarrow [u^{1/q}]$. Just simple tabulation…
- $h(x) = h_1(x) \oplus T[h_2(x)]$.

All previous properties, plus:

- minwise independence with $\varepsilon(u) = o(1)$.
- linear probing/chaining with buffer $k = O(lg n)$.
THE END