CUCKOO HASHING REVISITED

motivated by Luc Devroye, jointly worked out with Reinhard Kutzelnigg

Michael Drmota*

Institut für Diskrete Mathematik und Geometrie
TU Wien

michael.drmota@tuwien.ac.at
www.dmg.tuwien.ac.at/drmota/

* supported by the Austrian Science Foundation FWF, grant S9600.
Contents

- Cuckoo Hashing
- Cuckoo Graph
- Random Bipartite Graphs
- Asymptotic Results
- Generating Functions
- Double Saddle Points
Cuckoo Hashing

[Pa
gh and Rodler, 2001]

- 2 tables $T_1, T_2$ of size $m$

- 2 hash functions $h_1(x), h_2(x)$.

- Every key $x$ is stored at $h_1(x) \in T_1$ or at $h_2(x) \in T_2$. 

Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

```
T_1
0
1
2
3
4
```
```
T_2
```

**Cuckoo Hashing**

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Diagram: T₁ and T₂ with nodes 0, 1, 2, 3, 4, and an edge between node 3 in T₁ and node C in T₂.
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

\[
\begin{array}{c|c|c|c|c|c}
& T_1 & & & & T_2 \\
0 & & & & & \\
1 & A & & & & \\
2 & & & & & \\
3 & C & & & & \\
4 & & & & & \\
\end{array}
\]
Cuckoo Hashing

<table>
<thead>
<tr>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Cuckoo Hashing

\[ h_1 \] 3 1 1 2 3 1 3
\[ h_2 \] 3 4 3 0 2 4 3

\[ T_1 \] 0 1 2 3 4
\[ T_2 \]
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
## Cuckoo Hashing

<table>
<thead>
<tr>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table:**

<table>
<thead>
<tr>
<th></th>
<th>$T_1$</th>
<th></th>
<th>$T_2$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>V</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>C</td>
<td></td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Diagram:

```
   T1          T2
0  |
1  K        |
2  V        |
3  M  C    |
4          A
```
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Cuckoo Hashing is a hash table data structure that uses two hash functions and two hash tables to handle collisions in a way that allows for constant-time lookups. In the given example, two hash functions $h_1$ and $h_2$ are used to map keys to indices in two separate tables $T_1$ and $T_2$. The keys $K$, $V$, $M$, $C$, and $A$ are stored in the tables, demonstrating how keys can be relocated between tables to resolve collisions.
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

$T_1$ and $T_2$ with a conflict resolution strategy for key $K$. The figure shows the insertion process where $K$ is placed in the first table, and then migrated to the second table to resolve the collision.
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

```
T_1

0
1
2
3
4

T_2

K
C
A
```

```
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

![Diagram of Cuckoo Hashing]
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Diagram showing Cuckoo Hashing with keys $C, A, K, V, M, F, H$ and their hashed values in $T_1$ and $T_2$. Keys are represented in boxes and hashed values are connected with arrows.
Cuckoo Hashing

<table>
<thead>
<tr>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

$$ T_1 \quad T_2 $$

0

1  $F$

2  $V$

3  $C$

4

M

K

A
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Diagram:

- $T_1$ with nodes labeled $H$, $V$, $F$.
- $T_2$ with nodes labeled $K$, $M$, $A$.
- Cuckoo hashing interconnections shown between $T_1$ and $T_2$. 
Cuckoo Hashing

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>A</th>
<th>K</th>
<th>V</th>
<th>M</th>
<th>F</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h_1$</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>$h_2$</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

REHASH !!

Diagram: Two tables $T_1$ and $T_2$ with keys C, A, K, V, M, F, H. Key C gets rehashed to bucket 3.
Cuckoo Graph

1 — 4
3 — 3
2 — 2
0 — 0
4 — 1
Cuckoo Graph

1 \rightarrow 4

3 \rightarrow 3

2 \rightarrow 2

0 \rightarrow 0

4 \rightarrow 1
Cuckoo Graph

![Cuckoo Graph Diagram]
Cuckoo Graph

Graph representation:

- Node 1 is connected to node 4.
- Node 3 is connected to node 3.
- Node 2 is connected to node 2.
- Node 0 is connected to node 0.
- Node 4 is connected to node 1.
Cuckoo Graph
Cuckoo Graph

1 -> 4
3 -> 3
2 -> 2
0 -> 0
4 -> 1
Cuckoo Graph

1 -> 4
3 -> 3
2 -> 2
0 -> 0
4 -> 1
Cuckoo Graph

1 -> 4
3 -> 3
2 -> 2
0 -> 0
4 -> 1
Cuckoo Graph

1 \rightarrow 4
3 \rightarrow 3
2 \rightarrow 2
0 \rightarrow 0
4 \rightarrow 1
Cuckoo Graph

1 → 4

3 → 3

2 → 2

0 → 0

4 → 1
Cuckoo Graph

1 → 4
3 → 3
2 → 2
0 → 0
4 → 1
Cuckoo Graph

1 → 4
3 → 3
2 → 2
0 → 0
4 → 1
Cuckoo Graph
Cuckoo Graph

- Bipartite Graph

\[ G = (V_1, V_2, E) \]

\[ V_1, V_2 \ldots \text{tables}, \text{labeled vertex sets} \]
\[ E \ldots \text{collects hash values} \ e = (h_1(x), h_2(x)), \text{labeled edges} \]

- \(|V_1| = |V_2| = m \ldots \text{table size} \]
  \[ |E| = n \ldots \text{number of keys} \]

- Hashing works \( \iff \) \( G \) contains no complex component !!!
  (only trees or unicyclic components)
Cuckoo Graph

Edge Insertion: Good case 1: joining 2 trees
Cuckoo Graph

Edge Insertion: Good case 1: joining 2 trees
Cuckoo Graph

Edge Insertion: Good case 1: joining 2 trees
Cuckoo Graph

Edge Insertion: **Good case 1:** joining 2 trees
Cuckoo Graph

Edge Insertion: Good case 1: joining 2 trees
Cuckoo Graph

Edge Insertion: Good case 2: inserting an edge into a tree
Cuckoo Graph

Edge Insertion: Good case 2: inserting an edge into a tree
Cuckoo Graph

Edge Insertion: Good case 2: inserting an edge into a tree
Cuckoo Graph

Edge Insertion: Good case 2: inserting an edge into a tree
Cuckoo Graph

Edge Insertion: **Good case 2**: inserting an edge into a tree
Cuckoo Graph

Edge Insertion: Good case 2: inserting an edge into a tree
Cuckoo Graph

**Edge Insertion:** Bad case: inserting an edge into a cyclic component
### Cuckoo Hashing – Cuckoo Graph

<table>
<thead>
<tr>
<th>Cuckoo Hashing</th>
<th>Cuckoo Graph $G = (V_1, V_2, E)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ size of hash tables $T_1, T_2$</td>
<td>$</td>
</tr>
<tr>
<td>$n$ number of keys</td>
<td>$</td>
</tr>
<tr>
<td>successful hashing</td>
<td>no complex components</td>
</tr>
<tr>
<td>running time</td>
<td>size of components</td>
</tr>
</tbody>
</table>
Random Bipartite Graph

- $G_{m_1,m_2,n}$ random bipartite multigraph (labeled)
- $|V_1| = m_1$, $|V_2| = m_2$ labeled vertex sets
- $|E| = n$ labeled multi edges
- Each of these $\#G_{m_1,m_2,n} = m_1^n m_2^n$ graphs is equally likely

The edge labels encode the insertion procedure (dynamic model)

**Notation:** $G_{m_1,m_2,n}^o$ ... graphs with no complex component
Asymptotic Results

Probability of Succussfull Hashing

Theorem 1

Suppose that \( n = (1 - \varepsilon)m \) for some \( \varepsilon > 0 \).

\[
\frac{\#G_{m,m,n}}{\#G_{m,m,n}} = 1 - \frac{h(\varepsilon)}{m} + O(m^{-2}) \quad (m \to \infty)
\]

with

\[
h(\varepsilon) = \frac{(2\varepsilon^2 - 5\varepsilon + 5)(1 - \varepsilon)^3}{12(2 - \varepsilon)^2\varepsilon^3} = \frac{5}{48}\varepsilon^{-3} - \frac{5}{16}\varepsilon^{-2} + \frac{21}{64}\varepsilon^{-1} - \frac{13}{96} + \frac{3}{256}\varepsilon + \frac{1}{256}\varepsilon^2 + \frac{1}{1024}\varepsilon^3 + O(\varepsilon^4)
\]
Asymptotic Results

Remark 1. [Devroye and Morin]: $1 - \frac{\#G_{m,m,n}^o}{\#G_{m,m,n}} = O(1/m)$.

Remark 2. The probability that Cuckoo hashing fails (with table sizes $m$ and $n = (1 - \varepsilon)m$ keys) is

$$\frac{h(\varepsilon)}{m} + O(m^{-2}).$$
Asymptotic Results

Probability of Successful Hashing

Theorem 2

Suppose that \( n = m \).

\[
\frac{\#G_{m,m,n}}{\#G_{m,m,n}} = \frac{\sqrt{2}}{3} + o(1) = 0.8164965809 \ldots + o(1) \quad (m \to \infty)
\]

Remark 3. The same results holds for usual random graphs [Flajolet, Knuth, and Pittel, 1989]

Remark 4. Threshold appears at \( n = m - \Theta(m^{2/3}) \)
(as for random graphs – birth of a giant component).
Asymptotic Results

Unicyclic Components

Theorem 3. Let $X_{m,n}$ denote the number of points in unicyclic components. Suppose that $\varepsilon > 0$ and $n = (1 - \varepsilon)m$. Then, as $m \to \infty$, $X_{m,n}$ has a discrete limiting distribution with expected value

$$E X_{m,n} = \frac{(1 - \varepsilon)^2}{(2 - \varepsilon)\varepsilon^2} + O\left(\frac{1}{m}\right)$$

and variance

$$\text{Var} X_{m,n} = \frac{(1 - \varepsilon)^2(\varepsilon^2 - 3\varepsilon + 4)}{(2 - \varepsilon)^2\varepsilon^4} + O\left(\frac{1}{m}\right).$$
Asymptotic Results

Tree Sizes

**Theorem 4.** Let $T_{k;m,n}$ denote the number of trees of size $k$. Suppose that $\varepsilon > 0$ and $n = (1 - \varepsilon)m$. Then, as $m \to \infty$, $T_{k;m,n}$ satisfies a central limit theorem with expected value

$$E T_{k;m,n} = 2m \frac{k^{k-2}(1 - \varepsilon)^{k-1}e^{k(\varepsilon-1)}}{k!} + O(1)$$

and variance

$$\text{Var} T_{k;m,n} = 2m \left( \frac{k^{k-2}(1 - \varepsilon)^{k-1}e^{k(\varepsilon-1)}}{k!} - \frac{k^{2k-4}(1 - \varepsilon)^{2k-3}e^{2k(\varepsilon-1)}(k^2\varepsilon^2 + k^2\varepsilon - 4k\varepsilon + 2)}{(k!)^2} \right) + O(1)$$
Asymptotic Results

Remark 1. Cyclic components are negligible (for $\varepsilon > 0$)

Remark 2. Expected average tree size $\leq 2$  
$\implies$ expected running time $\leq 2m$

(This explains the extremely good performance of Cuckoo hashing.)
Generating Functions

Bipartite Trees

t_{1,m_1,m_2} \ldots \text{number of bipartite rooted trees with } m_1 \text{ nodes of type 1, } m_2 \text{ nodes of type 2, and the root is of type 1.}

t_{2,m_1,m_2} \ldots \text{number of bipartite rooted trees with } m_1 \text{ nodes of type 1, } m_2 \text{ nodes of type 2, and the root is of type 2.}

\[
t_1(x, y) = \sum_{m_1, m_2} t_{1,m_1,m_2} \frac{x^{m_1} y^{m_2}}{m_1! m_2!}, \quad t_2(x, y) = \sum_{m_1, m_2} t_{2,m_1,m_2} \frac{x^{m_1} y^{m_2}}{m_1! m_2!},
\]

\[
t_1(x, y) = xe^{t_2(x,y)}, \quad t_2(x, y) = ye^{t_1(x,y)}
\]
Generating Functions

Bipartite Trees

\( \tilde{t}_{m_1,m_2} \) is the number of bipartite \textbf{unrooted} trees with \( m_1 \) nodes of type 1 and \( m_2 \) nodes of type 2.

\[
\tilde{t}(x, y) = \sum_{m_1,m_2} \tilde{t}_{m_1,m_2} \frac{x^{m_1} y^{m_2}}{m_1! m_2!},
\]

\[\tilde{t}(x, y) = t_1(x, y) + t_2(x, y) - t_1(x, y)t_2(x, y)\]

Lemma. [Scoins, 1962]

\[
\tilde{t}_{m_1,m_2} = \frac{m_2-1}{m_1} \frac{m_1-1}{m_2}
\]
Generating Functions

Bipartite Trees

**Remark 1** \( t_{1,m_1,m_2} = m_1 \tilde{t}_{m_1,m_2} \), \( t_{2,m_1,m_2} = m_2 \tilde{t}_{m_1,m_2} \).

**Remark 2** \( t_1(x, x) = t_2(x, x) = t(x) \) is the usual tree function given by
\[
 t(x) = xe^{t(x)}.
\]

**Remark 3** \( t_{m_1,m_2,n} \) ... number of unrooted labeled bipartite trees with \( m_1 \) nodes of type 1, \( m_2 \) nodes of type 2, and \( n \) (labeled) edges.

\[
 \sum_{m_1,m_2,n} t_{m_1,m_2,n} \frac{x^{m_1} y^{m_2} u^n}{m_1! m_2! n!} = \frac{1}{u} \tilde{t}(xu, yu).
\]
Generating Functions

Lemma

\[
g^\circ(x, y, u) = \sum_{m_1, m_2, n} \#G^\circ_{m_1, m_2, n} \frac{x^{m_1} y^{m_2} u^n}{m_1! m_2! n!} = \frac{e^{\frac{1}{u} \tilde{t}(xu, yu)}}{\sqrt{1 - t_1(xu, yu) t_2(xu, yu)}}.
\]

Proof.

Cyclic component with 2\(k\) cyklic points: \[
\frac{1}{2k} t_1(xu, yu)^k t_2(xu, yu)^k
\]

\[
\implies g^\circ(x, y, u) = \exp \left( \frac{1}{u} \tilde{t}(xu, yu) + \sum_{k \geq 1} \frac{1}{2k} t_1(xu, yu)^k t_2(xu, yu)^k \right)
\]
Generating Functions

Corollary

\[ \#G_{m_1,m_2,n}^{\circ} = \frac{m_1!m_2!n!}{(m_1 + m_2 - n)!} \left[ x^{m_1} y^{m_2} \right] \frac{\tilde{t}(x, y)^{m_1+m_2-n}}{\sqrt{1 - t_1(x, y)t_2(x, y)}} \]

\[ = -\frac{m_1!m_2!n!}{4\pi(m_1 + m_2 - n)!} \times \int_{|x|=x_0} \int_{|y|=y_0} \frac{\tilde{t}(x, y)^{m_1+m_2-n}}{\sqrt{1 - t_1(x, y)t_2(x, y)}} \frac{dx}{x^{m_1+1}} \frac{dy}{y^{m_2+1}}. \]

\[ \rightarrow \text{DOUBLE SADDLE POINT} \]
Generating Functions

Extensions

E.g., in

\[
\frac{e^u \tilde{t}(xu, yu) + x(w-1) + y(w-1)}{\sqrt{1 - t_1(xu, yu)t_2(xu, yu)}}.
\]

the additional variable \( w \) counts the number of isolated nodes
(= tree components of size 1).

etc.
**Double Saddle Point**

**Lemma**

\[ f(x, y), \ g(x, y) \ldots \text{analytic functions around } (0, 0) \]

(+ minor technical assumptions)

\[ \Longrightarrow \left[ x^{m_1}y^{m_2} \right] g(x, y)f(x, y)^k = \frac{g(x_0, y_0)f(x_0, y_0)^k}{2\pi x_0^{m_1}y_0^{m_2}k\sqrt{\Delta}} \left( 1 + \frac{h}{24\Delta^3 k} + O \left( \frac{1}{k^2} \right) \right), \]

where \( x_0 \) and \( y_0 \) are uniquely defined by

\[ \frac{m_1}{k} = \frac{x_0}{f(x_0, y_0)} \left[ \frac{\partial}{\partial x} f(x, y) \right]_{(x_0, y_0)}, \quad \frac{m_2}{k} = \frac{y_0}{f(x_0, y_0)} \left[ \frac{\partial}{\partial y} f(x, y) \right]_{(x_0, y_0)} \]

and are contained in a finite interval of the positive real line, that is, \( m_1, m_2, \) and \( k \) have to be of the same order of magnitude.
Double Saddle Point

Set

\[ \kappa_{ij} = \left[ \frac{\partial^i}{\partial u^i} \frac{\partial^j}{\partial v^j} \log f(x_0 e^u, y_0 e^v) \right]_{(0,0)}, \quad \bar{\kappa}_{ij} = \left[ \frac{\partial^i}{\partial u^i} \frac{\partial^j}{\partial v^j} \log g(x_0 e^u, y_0 e^v) \right]_{(0,0)}. \]

Then \( \Delta = \kappa_{20} \kappa_{02} - \kappa_{11}^2 \), and with

\[ \alpha = 54 \kappa_{21} \kappa_{11} \kappa_{12} \kappa_{20} \kappa_{02} + 6 \kappa_{22} \kappa_{20} \kappa_{02} \kappa_{11}^2 - 12 \kappa_{22} \kappa_{11}^4 + 4 \kappa_{03} \kappa_{11} \kappa_{30} + 36 \kappa_{21} \kappa_{11}^3 \kappa_{12} + 6 \kappa_{22} \kappa_{20} \kappa_{02}^2 + 6 \kappa_{03} \kappa_{11} \kappa_{30} \kappa_{20} \kappa_{02}, \]
\[ \beta = -5 \kappa_{02} \kappa_{11}^2 + 30 \kappa_{02} \kappa_{30} \kappa_{11} \kappa_{21} - 24 \kappa_{02} \kappa_{30} \kappa_{12} \kappa_{11}^2 - 6 \kappa_{02} \kappa_{30} \kappa_{12} \kappa_{20} - 12 \kappa_{11} \kappa_{02} \kappa_{31} \kappa_{20} - 36 \kappa_{02} \kappa_{21} \kappa_{11}^2 - 9 \kappa_{02} \kappa_{21} \kappa_{20} + 3 \kappa_{02} \kappa_{40} \kappa_{20} - 3 \kappa_{02} \kappa_{40} \kappa_{11} + 12 \kappa_{11} \kappa_{02} \kappa_{31}, \]
\[ \gamma = 12 \Delta \left( \kappa_{02}^2 \kappa_{30} - \kappa_{11} \kappa_{20} \kappa_{03} - 3 \kappa_{21} \kappa_{11} \kappa_{02} + \kappa_{12} \kappa_{11}^2 + \kappa_{12} (\kappa_{02} \kappa_{20} + \kappa_{11}^2) \right), \]
\[ \delta = 24 \Delta (\kappa_{11} \kappa_{20} \kappa_{02} - \kappa_{11}^3), \]
\[ \eta = 12 \Delta (\kappa_{02} \kappa_{11}^2 - \kappa_{02}^2 \kappa_{20}) \]

we have

\[ h = \alpha + \beta + \beta + \gamma \bar{\kappa}_{10} + \gamma \bar{\kappa}_{01} + \delta \bar{\kappa}_{10} \bar{\kappa}_{01} + \eta \bar{\kappa}_{10}^2 + \eta \bar{\kappa}_{01}^2 + 4 \eta \bar{\kappa}_{20} + 4 \eta \bar{\kappa}_{02} + 4 \delta \bar{\kappa}_{11}, \]

where \( \bar{\ } \) indicates to replace all functions of type \( \kappa_{ij} \) by \( \kappa_{ji} \).
Double Saddle Point

Proof of Theorem 1

\[ m_1 = m_2 = m, \quad n = (1 - \varepsilon)m \text{ for some } \varepsilon > 0 \]

\[ \rightarrow \quad x_0 = y_0 = (1 - \varepsilon)e^{\varepsilon-1} < \frac{1}{e}. \]

Application with \( f(x, y) = \bar{t}(x, y) \) and \( k = 2m - n = (1 + \varepsilon)n \)

\[ \rightarrow \quad \text{Theorem } 1. \]
Double Saddle Point

Proof of Theorem 2

- Saddle point $x_0 = y_0 = \frac{1}{e}$ and squareroot singularity coincide !!!
- Apply Lagrange inversion for $t_1 = x \exp(y e^{t_1})$.
- Series expansion for $\frac{1}{\sqrt{1 - v}}$
- Infinite series of double saddle point integrals
  - one integral with scale $e^{-t^2/2}$
  - the second integral with scale $e^{-cs^3}$
- **Lommel functions** (similar to Airy functions)
- (Explicit) Mellin transform of Lommel function

$$\rightarrow \sqrt{\frac{2}{3}} \quad \text{(Theorem 2)}$$
Double Saddle Point

Lommel Functions

- Fundamental system of the inhomogeneous Bessel equation

\[ x^2 y'' + xy' - (x^2 + \nu^2) = kx^{\mu+1}. \]

- Closely related to the integral

\[ \int_0^\infty e^{-t^3 + kt} t \, dt. \]
Double Saddle Point

Remark

The *analytic structure* of generating functions for bipartite random graphs is more difficult than that of usual random graphs. This is due to the double saddle point (that comes from the additional variable) and the squareroot singularity that is now in 2 variables (instead of 1).

Nevertheless the results look the same. Thus, one can expect that most properties of random graphs have a counterpart in random bipartite graphs (birth of giant component etc.)
Thank You!