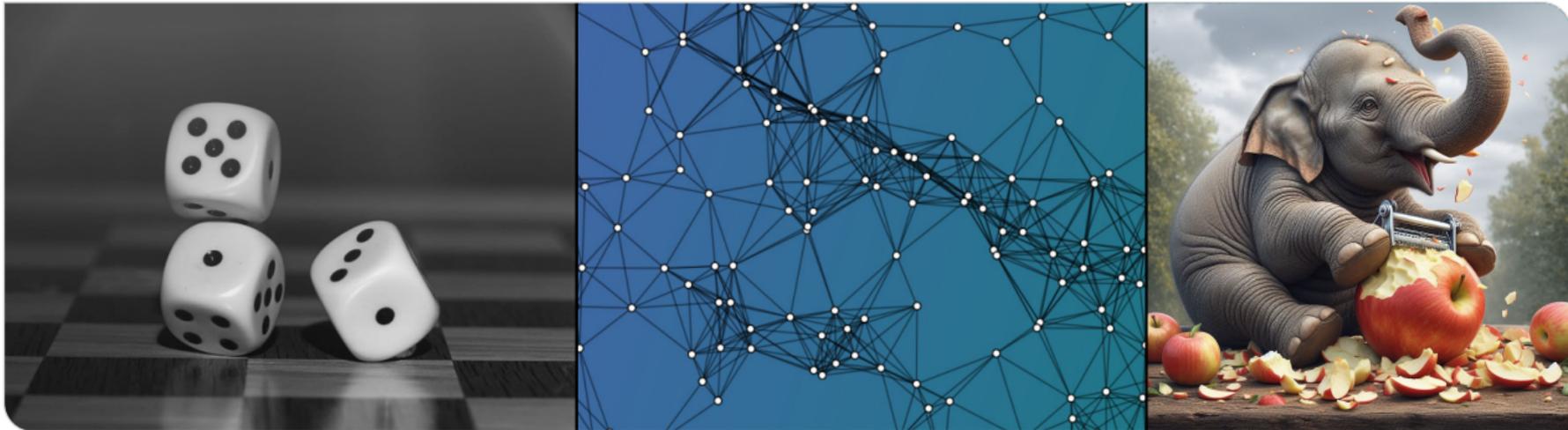


Probability and Computing – The Peeling Algorithm

Stefan Walzer | WS 2025/2026



Static hash table

`construct(S):` builds table T with key set S // no insertions or deletions after construction

`lookup(x):` checks if x is in T or not

Static hash table

`construct(S):` builds table T with key set S // no insertions or deletions after construction

`lookup(x):` checks if x is in T or not

Constructing cuckoo hash tables:

- solved by Khosla 2013: “Balls into Bins Made Faster”
- matching algorithm resembling preflow push
- expected running time $\mathcal{O}(n)$, finds placement whenever one exists
- not in this lecture

Static hash table

construct(S): builds table T with key set S // no insertions or deletions after construction
lookup(x): checks if x is in T or not

Constructing cuckoo hash tables:

- solved by Khosla 2013: “Balls into Bins Made Faster”
- matching algorithm resembling preflow push
- expected running time $\mathcal{O}(n)$, finds placement whenever one exists
- not in this lecture

Greedily constructing cuckoo hash tables

- Peeling: simple algorithm but sophisticated analysis
- interesting applications beyond hash tables (see “retrieval” in next lecture)

1. The Peeling Algorithm

2. The Peeling Theorem

3. Conclusion

The Peeling Algorithm

```
constructByPeeling( $S \subseteq D, h_1, h_2, h_3 \in [m]^D$ )
```

```
 $T \leftarrow [\perp, \dots, \perp]$  // empty table of size  $m$ 
```

```
while  $\exists i \in [m] : \exists$  exactly one  $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$  do
```

```
    //  $x$  is only unplaced key that may be placed in  $i$ 
```

```
     $T[i] \leftarrow x$ 
```

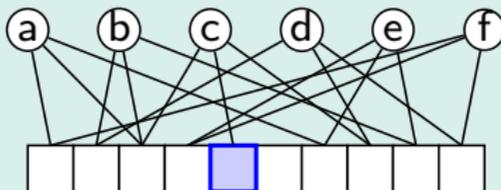
```
     $S \leftarrow S \setminus \{x\}$ 
```

```
if  $S = \emptyset$  then
```

```
    return  $T$ 
```

```
else
```

```
    return NOT-PEELABLE
```



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

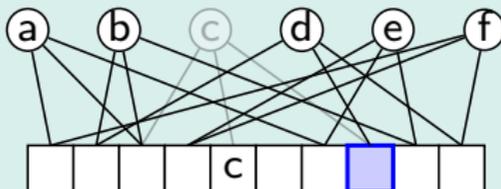
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

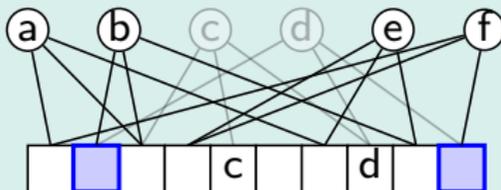
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

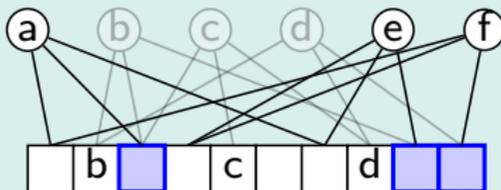
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

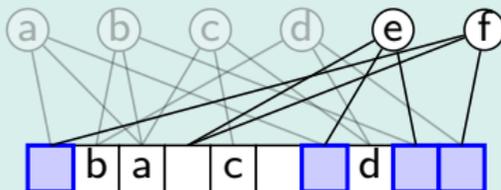
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

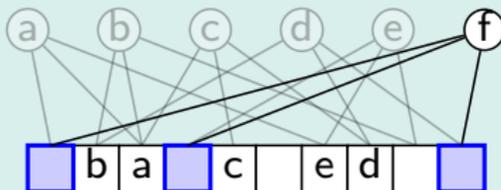
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

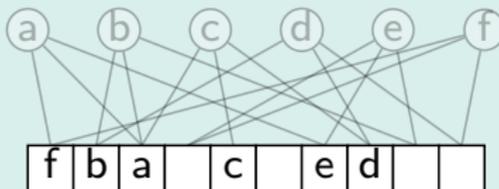
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



The Peeling Algorithm

$\text{constructByPeeling}(S \subseteq D, h_1, h_2, h_3 \in [m]^D)$

$T \leftarrow [\perp, \dots, \perp]$ // empty table of size m

while $\exists i \in [m] : \exists$ *exactly one* $x \in S : i \in \{h_1(x), h_2(x), h_3(x)\}$ **do**

 // x is only unplaced key that may be placed in i

$T[i] \leftarrow x$

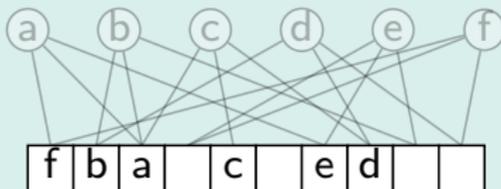
$S \leftarrow S \setminus \{x\}$

if $S = \emptyset$ **then**

return T

else

return NOT-PEELABLE



Exercise

- Success of `constructByPeeling` does not depend on choices for i made by `while`.
- `constructByPeeling` can be implemented in linear time.

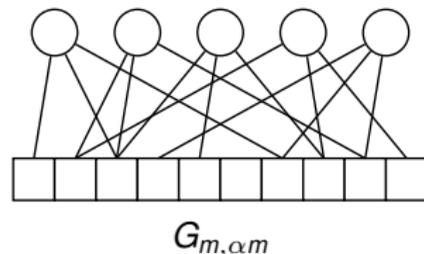
Peelability and the Cuckoo Graph

Cuckoo Graph and Peelability

- The **Cuckoo Graph** is the bipartite graph

$$G_{S, h_1, h_2, h_3} = (S, [m], \{(x, h_i(x)) \mid x \in S, i \in [3]\})$$

- Call G_{S, h_1, h_2, h_3} **peelable** if `constructByPeeling(S, h_1, h_2, h_3)` succeeds.
- If $h_1, h_2, h_3 \sim \mathcal{U}([m]^D)$ then the distribution of G_{S, h_1, h_2, h_3} does not depend on S . We then simply write $G_{m, \alpha m}$.
 - m \square -nodes and $\lfloor \alpha m \rfloor$ \circ -nodes
 - think: α is constant and $m \rightarrow \infty$.



1. The Peeling Algorithm

2. The Peeling Theorem

3. Conclusion

Peeling Theorem

Peeling Threshold

Let $c_3^\Delta = \min_{y \in [0,1]} \frac{y}{3(1-e^{-y})^2} \approx 0.81$.

Theorem (today's goal)

Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.

Peeling Theorem

Peeling Threshold

Let $c_3^\Delta = \min_{y \in [0,1]} \frac{y}{3(1-e^{-y})^2} \approx 0.81$.

Theorem (today's goal)

Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.

Remark: More is known.

- For “ $\alpha < c_3^\Delta$ ” we get “peelable” with probability $1 - \mathcal{O}(1/m)$.
- For “ $\alpha > c_3^\Delta$ ” we get “not peelable” with probability $1 - \mathcal{O}(1/m)$.
- Corresponding thresholds c_k^Δ for $k \geq 3$ hash functions are also known.

Peeling Theorem

Peeling Threshold

Let $c_3^\Delta = \min_{y \in [0,1]} \frac{y}{3(1-e^{-y})^2} \approx 0.81$.

Theorem (today's goal)

Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.

Remark: More is known.

- For “ $\alpha < c_3^\Delta$ ” we get “peelable” with probability $1 - \mathcal{O}(1/m)$.
- For “ $\alpha > c_3^\Delta$ ” we get “not peelable” with probability $1 - \mathcal{O}(1/m)$.
- Corresponding thresholds c_k^Δ for $k \geq 3$ hash functions are also known.

Exercise: What about $k = 2$?

Peeling does not reliably work for $k = 2$ for any $\alpha > 0$.

Peeling Theorem: Proof outline

Theorem (today's goal)

Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.

Proof Idea

The random (possibly) infinite tree T_α can be peeled for $\alpha < c_3^\Delta$ and T_α is locally like $G_{m,\alpha m}$.

Peeling Theorem: Proof outline

Theorem (today's goal)

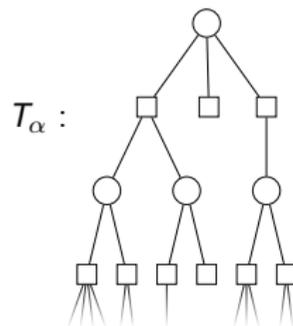
Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.

Proof Idea

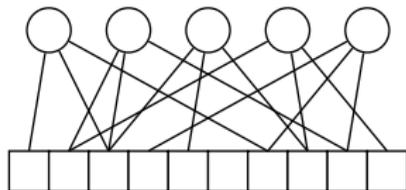
The random (possibly) infinite tree T_α can be peeled for $\alpha < c_3^\Delta$ and T_α is locally like $G_{m,\alpha m}$.

Steps

- I What is T_α ?
- II What does peeling mean in this setting?
- III What role does c_3^Δ play?
- IV What does it mean for T_α to be locally like $G_{m,\alpha m}$?
- V What is the probability that a fixed key of $G_{m,\alpha m}$ is peeled?
- VI What is the probability that *all* keys of $G_{m,\alpha m}$ are peeled?



i What is T_α ?



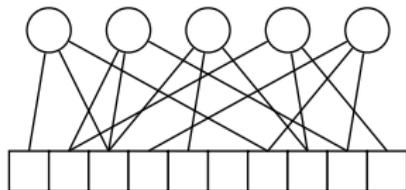
Observations for the finite Graph $G_{m, \alpha m}$

- each \bigcirc has 3 \square as neighbours (rare exception: $h_1(x), h_2(x), h_3(x)$ not distinct)
- each \square has random number X of \bigcirc as neighbours with $X \sim \text{Bin}(3n, \frac{1}{m}) = \text{Bin}(3\lfloor \alpha m \rfloor, \frac{1}{m})$. In an exercise we have seen that

$$X \xrightarrow{d} Y \text{ for } Y \sim \text{Pois}(3\alpha) \text{ and } m \rightarrow \infty$$

// i.e. $\forall i : \Pr[X = i] \xrightarrow{m \rightarrow \infty} \Pr[Y = i]$.

i What is T_α ?

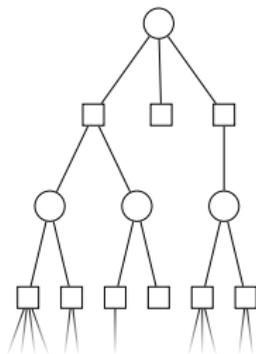


Observations for the finite Graph $G_{m, \alpha m}$

- each \bigcirc has 3 \square as neighbours (rare exception: $h_1(x), h_2(x), h_3(x)$ not distinct)
- each \square has random number X of \bigcirc as neighbours with $X \sim \text{Bin}(3n, \frac{1}{m}) = \text{Bin}(3\lfloor \alpha m \rfloor, \frac{1}{m})$. In an exercise we have seen that

$$X \xrightarrow{d} Y \text{ for } Y \sim \text{Pois}(3\alpha) \text{ and } m \rightarrow \infty$$

// i.e. $\forall i : \Pr[X = i] \xrightarrow{m \rightarrow \infty} \Pr[Y = i]$.



Definition of the (possibly) infinite random tree T_α

- root is \bigcirc and has three \square as children
- each \square has random number of \bigcirc children, sampled $\text{Pois}(3\alpha)$ (independently for each \square).
- each non-root \bigcirc has two \square as children.

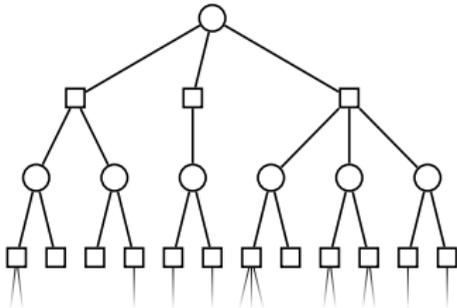
Remark: T_α is finite with positive probability > 0 , e.g. when the first three $\text{Pois}(3\alpha)$ random variables come out as 0. But T_α is also infinite with positive probability for $\alpha > \frac{1}{2}$.

ii What does peeling mean in this setting?

Peeling Algorithm

while \exists \square -node of degree 1 **do**
 | remove it and its incident \circ

\leftrightarrow not well defined outcome on infinite T_α !



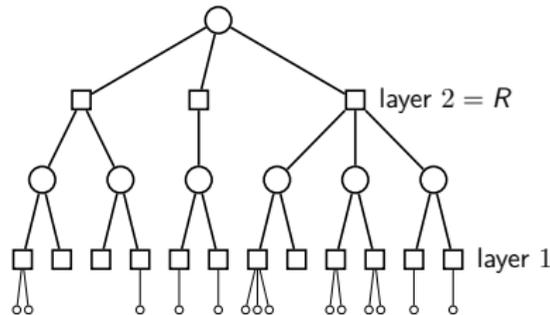
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

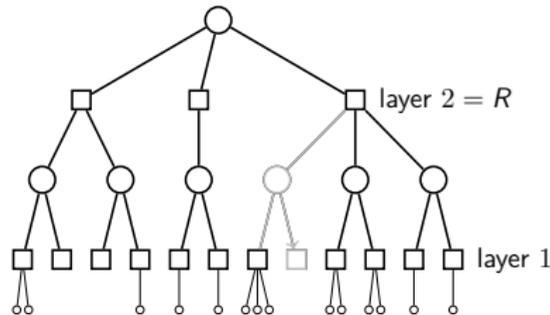
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

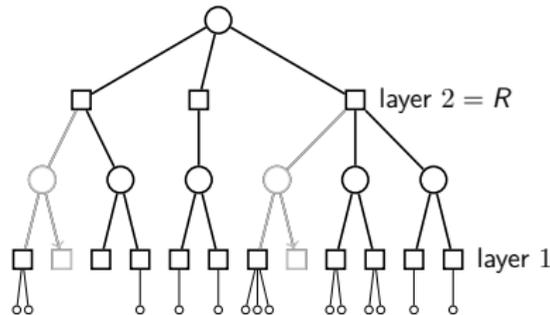
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

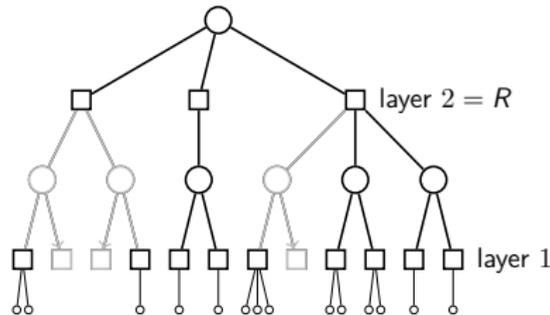
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

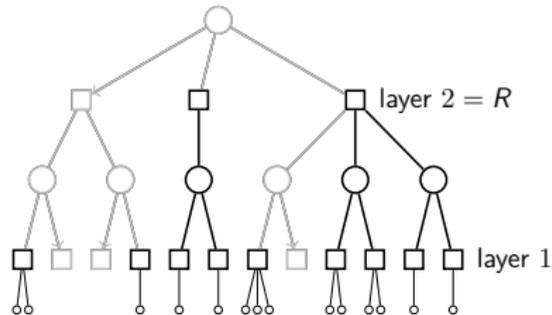
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

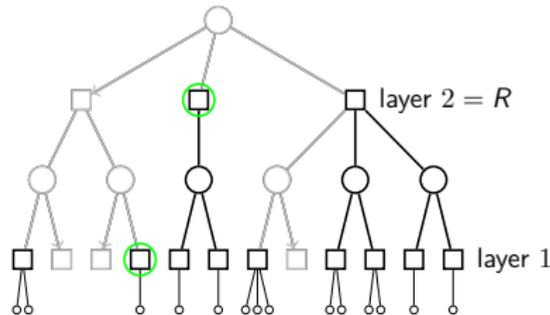
ii What does peeling mean in this setting?

Peeling Algorithm

while \exists childless \square -node **do**
 | remove it and its incident \circ

\hookrightarrow not well defined outcome on infinite T_α !

\hookrightarrow but well defined on T_α^R !



Peel only the first $R \in \mathbb{N}$ layers

- Let T_α^R be the first $2R + 1$ levels of T_α .
- R layers of \square -nodes, labeled bottom to top.
- Run peeling on T_α^R (later $R \rightarrow \infty$).

\hookrightarrow Why not consider the first $2R$ levels? (without +1)

Only care whether root is removed (root represents arbitrary node in $G_{m,\alpha m}$)

We may then simplify the peeling algorithm.

- replace “ \square -node of degree 1” condition with stronger “childless \square -node”.
 - prevents peeling of \square -nodes with one child and no parent
 - no matter: such nodes are disconnected from the root anyway
- whether node is peeled only depends on subtree
 \hookrightarrow one bottom up pass suffices for peeling

ii What does peeling mean in this setting? (2)

Observation

Let $q_R = \Pr[\text{root survives when peeling } T_\alpha^R]$.
The values q_R are decreasing in R .

Peeling Algorithm

```
while  $\exists$  childless  $\square$ -node do  
   $\sqsubset$  remove it and its incident  $\circ$ 
```

ii What does peeling mean in this setting? (2)

Observation

Let $q_R = \Pr[\text{root survives when peeling } T_\alpha^R]$.
 The values q_R are decreasing in R .

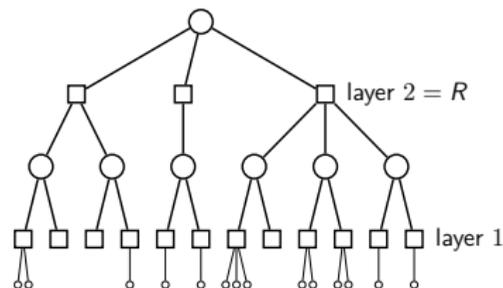
Peeling Algorithm

while \exists *childless* \square -node **do**
 \square remove it and its incident \circ

Proof.

Consider any \circ -node x that is a leaf.

- peeling cannot remove x
- *had* we generated more layers, x *may* be removable
- adding more layer can only cause more nodes to be peelable □

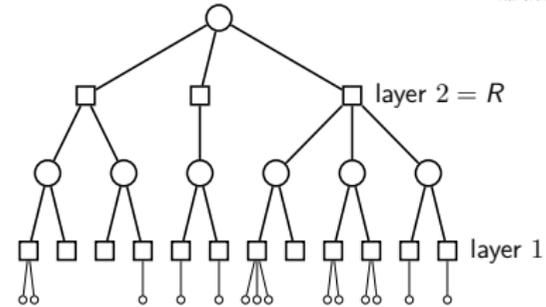


ii What does peeling mean in this setting? (3)

Peeling T_α^R bottom up

```

for  $i = 1$  to  $R$  do //  $\square$ -layers bottom to top
  for each  $\square$ -node  $v$  in layer  $i$  do
    if  $v$  has no children then
      remove  $v$  and its parent  $\circ$ 
  
```



Survival probabilities $p_i := \Pr[\square\text{-node in layer } i \text{ is not peeled}]$

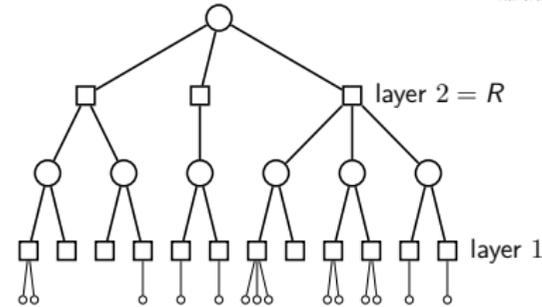
$$\begin{aligned}
 p_1 &= \Pr[\square\text{-node has at least 1 child}] \\
 &= \Pr_{Y \sim \text{Pois}(3\alpha)}[Y > 0] = 1 - e^{-3\alpha}.
 \end{aligned}$$

ii What does peeling mean in this setting? (3)

Peeling T_α^R bottom up

```

for  $i = 1$  to  $R$  do //  $\square$ -layers bottom to top
  for each  $\square$ -node  $v$  in layer  $i$  do
    if  $v$  has no children then
      remove  $v$  and its parent  $\circ$ 
  
```



Survival probabilities $p_i := \Pr[\square\text{-node in layer } i \text{ is not peeled}]$

$$\begin{aligned}
 p_1 &= \Pr[\square\text{-node has at least 1 child}] \\
 &= \Pr_{Y \sim \text{Pois}(3\alpha)}[Y > 0] = 1 - e^{-3\alpha}. \\
 p_i &= \Pr[\text{layer } i \text{ } \square\text{-node } v \text{ has at least 1 surviving child}] \\
 &= \Pr_{X \sim \text{Pois}(3\alpha p_{i-1}^2)}[X > 0] = 1 - e^{-3\alpha p_{i-1}^2}.
 \end{aligned}$$

\square -survival probabilities. With $p_0 := 1$ we have

$$p_i = \begin{cases} 1 & \text{if } i = 0 \\ 1 - e^{-3\alpha p_{i-1}^2} & \text{if } i = 1, 2, \dots \end{cases}$$

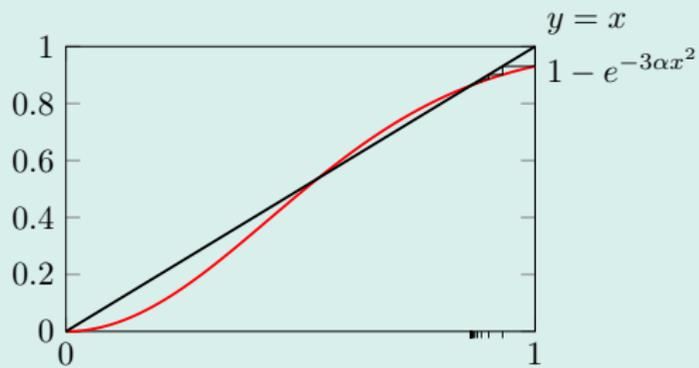
Moreover: $q_R := \Pr[\text{root survives}] = p_R^3$.

iii What role does $c_3^\Delta \approx 0.81$ play?

$$p_i = \begin{cases} 1 & \text{if } i = 0 \\ 1 - e^{-3\alpha p_{i-1}^2} & \text{if } i = 1, 2, \dots \end{cases}$$

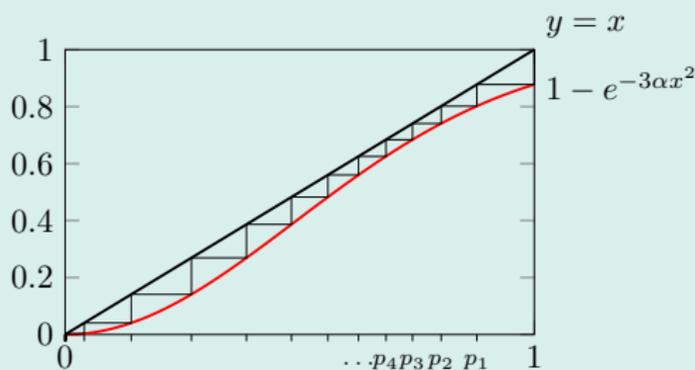
\hookrightarrow consider $f(x) = 1 - e^{-3\alpha x^2}$

Case 1: $\exists x > 0 : f(x) = x$.



$$\Rightarrow \lim_{i \rightarrow \infty} p_i = x^* = \max\{x \in [0, 1] \mid f(x) = x\}.$$

Case 2: $\forall x \in (0, 1] : f(x) < x$



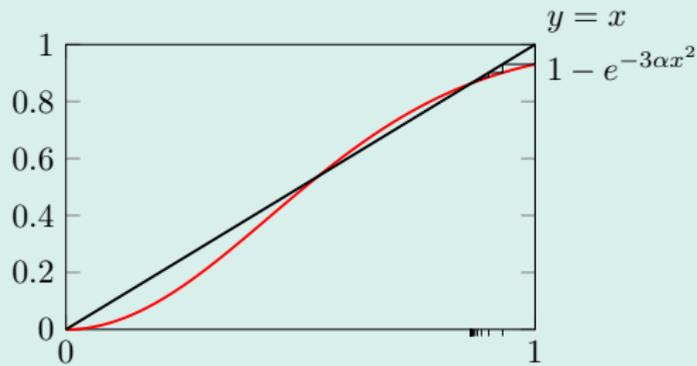
$$\Rightarrow \lim_{i \rightarrow \infty} p_i = 0.$$

iii What role does $c_3^\Delta \approx 0.81$ play?

$$p_i = \begin{cases} 1 & \text{if } i = 0 \\ 1 - e^{-3\alpha p_{i-1}^2} & \text{if } i = 1, 2, \dots \end{cases}$$

\hookrightarrow consider $f(x) = 1 - e^{-3\alpha x^2}$

Case 1: $\exists x > 0 : f(x) = x$.



$$\Rightarrow \lim_{i \rightarrow \infty} p_i = x^* = \max\{x \in [0, 1] \mid f(x) = x\}.$$

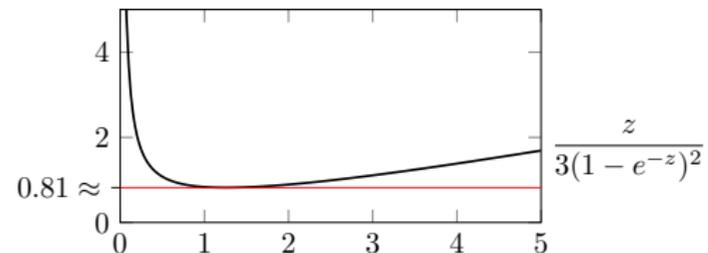
$$\text{Case 1} \Leftrightarrow \exists x > 0 : x = 1 - e^{-3\alpha x^2}$$

$$\Leftrightarrow \exists x > 0 : x^2 = (1 - e^{-3\alpha x^2})^2$$

$$\Leftrightarrow \exists z > 0 : \frac{z}{3\alpha} = (1 - e^{-z})^2 // z = 3\alpha x^2$$

$$\Leftrightarrow \exists z > 0 : \alpha = \frac{z}{3(1 - e^{-z})^2}$$

$$\Leftrightarrow \alpha \geq \min_{z > 0} \frac{z}{3(1 - e^{-z})^2} =: c_3^\Delta \approx 0.81$$



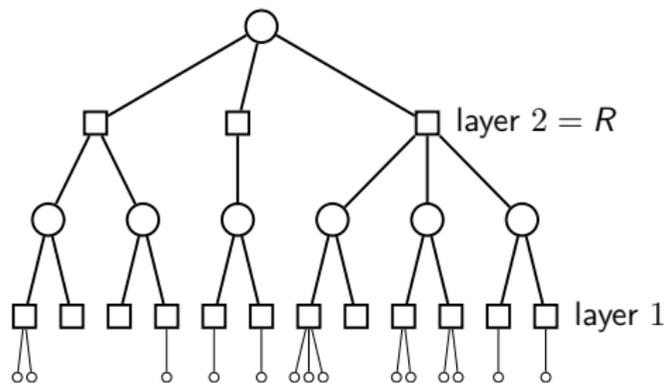
iii Interim Conclusion: What we learned about peeling T_α

Lemma

For $\alpha < c_3^\Delta \approx 0.81$ we have

- $\lim_{i \rightarrow \infty} p_i = 0.$
- $\lim_{R \rightarrow \infty} q_R = \lim_{R \rightarrow \infty} p_R^3 = 0.$

“Root rarely survives for large R .”

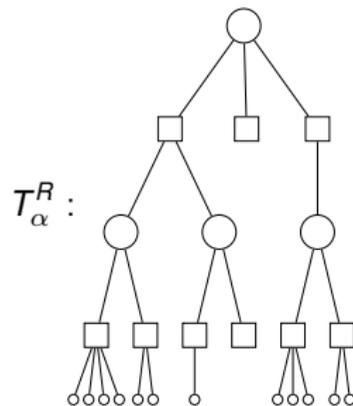
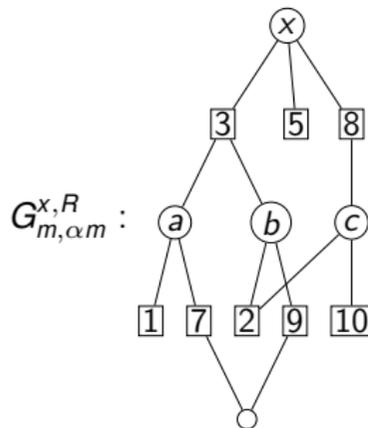
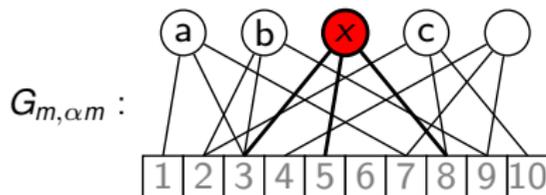


iv What does it mean for T_α to be locally like $G_{m,\alpha m}$?

Neighbourhoods in T_α and G

Let $R \in \mathbb{N}$. We consider

- T_α^R as before and
- for any fixed $x \in S$ the subgraph $G_{m,\alpha m}^{x,R}$ of $G_{m,\alpha m}$ induced by all nodes with distance at most $2R$ from x .



iv What does it mean for T_α to be locally like $G_{m,\alpha m}$?

Neighbourhoods in T_α and G

Let $R \in \mathbb{N}$. We consider

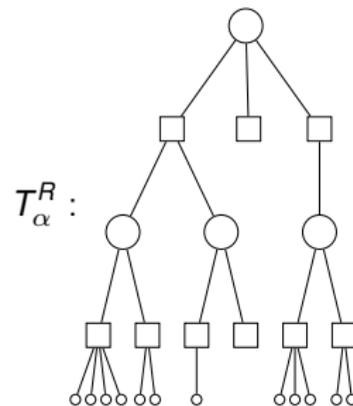
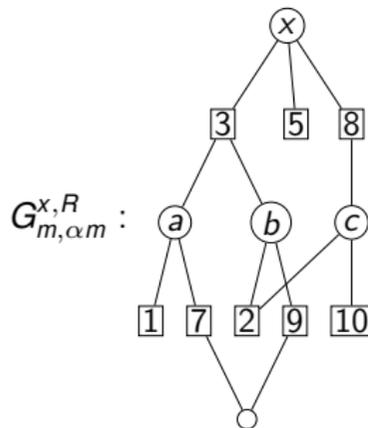
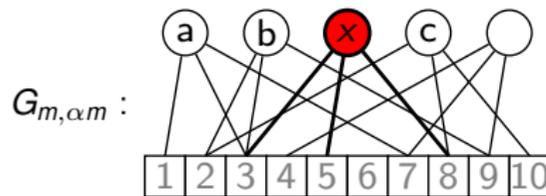
- T_α^R as before and
- for any fixed $x \in S$ the subgraph $G_{m,\alpha m}^{x,R}$ of $G_{m,\alpha m}$ induced by all nodes with distance at most $2R$ from x .

Lemma

For any $R \in \mathbb{N}$ and $m \rightarrow \infty$ we have

$$G_{m,\alpha m}^{x,R} \xrightarrow{d} T_\alpha^R$$

i.e. $\forall T : \lim_{m \rightarrow \infty} \Pr[G_{m,\alpha m}^{x,R} = T] = \Pr[T_\alpha^R = T]$.



iv Distribution of T_α^R

Lemma

Let T_y be a possible outcome of T_α^R given by a finite sequence $y = (y_1, \dots, y_k) \in \mathbb{N}_0^k$ specifying the number of children of \square -nodes in level order. Then

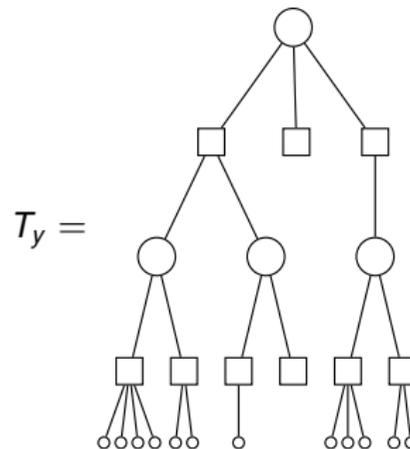
$$\Pr[T_\alpha^R = T_y] = \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

Remark on the meaning of “=”

We consider rooted unlabeled graphs where neighbours of vertices are ordered.



e.g. for $y = (2, 0, 1, 4, 2, 1, 0, 3, 2)$:



iv No cycles in $G_{m,\alpha m}^{x,R}$

Lemma

Assume $R = \mathcal{O}(1)$. The probability that $G_{m,\alpha m}^{x,R}$ contains a cycle is $\mathcal{O}(1/m)$.

iv No cycles in $G_{m,\alpha m}^{x,R}$

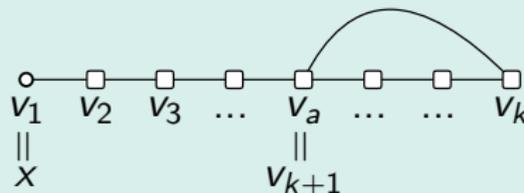
Lemma

Assume $R = \mathcal{O}(1)$. The probability that $G_{m,\alpha m}^{x,R}$ contains a cycle is $\mathcal{O}(1/m)$.

Proof.

If $G_{m,\alpha m}^{x,R}$ contains a cycle then we have

- a sequence $(v_1 = x, v_2, \dots, v_k, v_{k+1} = v_a)$ of nodes with $a \in [k]$
- of length $k \leq 4R$ (consider BFS tree for x and additional edge in it)
- for each $i \in \{1, \dots, k\}$ an index $j_i \in \{1, 2, 3\}$ of the hash function connecting v_i and v_{i+1} . (If $a = k - 1$ then $j_k \neq j_{k-1}$.)



iv No cycles in $G_{m,\alpha m}^{x,R}$

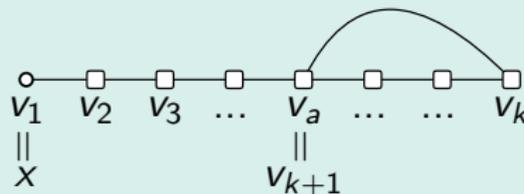
Lemma

Assume $R = \mathcal{O}(1)$. The probability that $G_{m,\alpha m}^{x,R}$ contains a cycle is $\mathcal{O}(1/m)$.

Proof.

If $G_{m,\alpha m}^{x,R}$ contains a cycle then we have

- a sequence $(v_1 = x, v_2, \dots, v_k, v_{k+1} = v_a)$ of nodes with $a \in [k]$
- of length $k \leq 4R$ (consider BFS tree for x and additional edge in it)
- for each $i \in \{1, \dots, k\}$ an index $j_i \in \{1, 2, 3\}$ of the hash function connecting v_i and v_{i+1} . (If $a = k - 1$ then $j_k \neq j_{k-1}$.)



$\Pr[\exists \text{ cycle in } G_{m,\alpha m}^{x,R}] \leq \Pr[\exists 2 \leq k \leq 4R : \exists v_2, \dots, v_k : \exists a \in [k] : \exists j_1, \dots, j_k \in [3] : \forall i \in [k] : h_{j_i} \text{ connects } v_i \text{ to } v_{i+1}]$

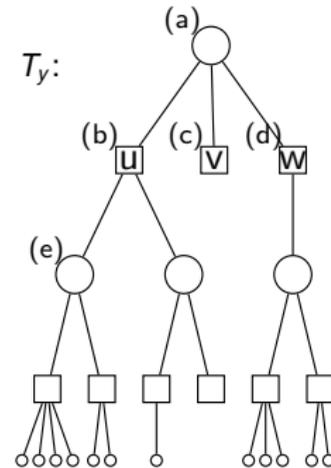
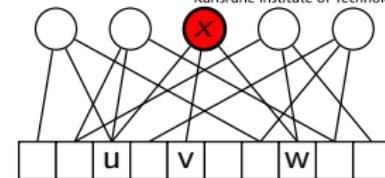
$$\leq \sum_{k=2}^{4R} \sum_{v_2, \dots, v_k} \sum_{a=1}^k \sum_{j_1, \dots, j_k} \prod_{i=1}^k \Pr[h_{j_i} \text{ connects } v_i \text{ to } v_{i+1}] \leq \sum_{k=2}^{4R} (\max\{m, n\})^{k-1} \cdot k \cdot 3^k \left(\frac{1}{m}\right)^k = \frac{1}{m} \sum_{k=2}^{4R} k \cdot 3^k = \mathcal{O}(1/m). \quad \square$$

iv Distribution of $G_{m,\alpha m}^{x,R}$

Lemma

Let T_y be a possible outcome of T_α^R as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \rightarrow \infty} \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$



iv Distribution of $G_{m,\alpha m}^{x,R}$

Lemma

Let T_y be a possible outcome of T_α^R as before. Then

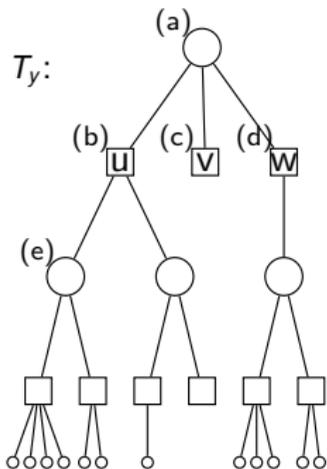
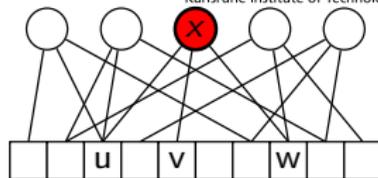
$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \rightarrow \infty} \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

“Proof by example”, using T_y shown on the right.

The following things have to “go right” for $G_{m,\alpha m}^{x,R} = T_y$.

- a $h_1(x), h_2(x), h_3(x)$ pairwise distinct: probability $\xrightarrow{m \rightarrow \infty} 1$
 \hookrightarrow non-distinct would give cycle of length 2. Unlikely by lemma.

Note: $3 \lfloor \alpha m \rfloor - 3$ remaining hash values $\sim \mathcal{U}([m])$.



iv Distribution of $G_{m,\alpha m}^{x,R}$

Lemma

Let T_y be a possible outcome of T_α^R as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \rightarrow \infty} \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

“Proof by example”, using T_y shown on the right.

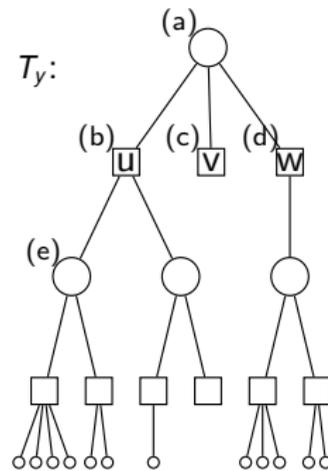
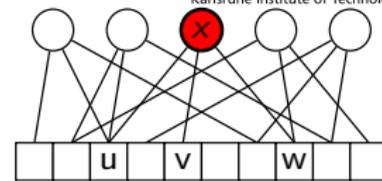
b Exactly $y_1 = 2$ of the remaining hash values are u .

$$\hookrightarrow \Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - 3, \frac{1}{m})} [Y = 2] \xrightarrow{m \rightarrow \infty} \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = 2]. \rightarrow \text{exercise}$$

Moreover: The two hash values must belong to 2 distinct keys. Probability $\xrightarrow{m \rightarrow \infty} 1$.

\hookrightarrow non-distinct would give cycle of length 2.

Note: The $3\lfloor \alpha m \rfloor - 5$ remaining hash values are $\sim \mathcal{U}([m] \setminus \{u\})$. \rightarrow exercise



iv Distribution of $G_{m,\alpha m}^{x,R}$

Lemma

Let T_y be a possible outcome of T_α^R as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \rightarrow \infty} \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

“Proof by example”, using T_y shown on the right.

c None of the remaining hash values are v .

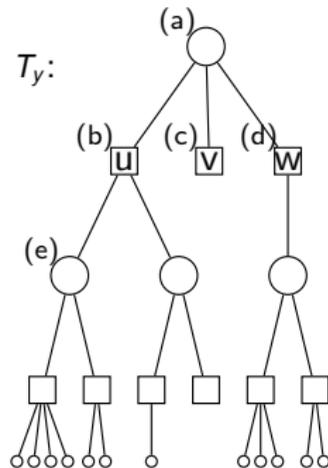
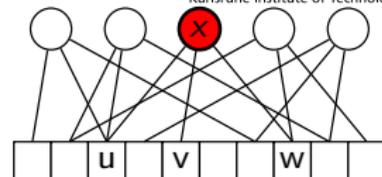
$$\hookrightarrow \Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - 5, \frac{1}{m-1})} [Y = 0] \xrightarrow{m \rightarrow \infty} \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = 0].$$

Note: The $3\lfloor \alpha m \rfloor - 5$ remaining hash values are $\sim \mathcal{U}([m] \setminus \{u, v\})$.

d One of the remaining hash values is w .

$$\hookrightarrow \Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - 5, \frac{1}{m-2})} [Y = 1] \xrightarrow{m \rightarrow \infty} \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = 1].$$

...



iv Distribution of $G_{m,\alpha m}^{x,R}$

Lemma

Let T_y be a possible outcome of T_α^R as before. Then

$$\Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [G_{m,\alpha m}^{x,R} = T_y] \xrightarrow{m \rightarrow \infty} \prod_{i=1}^k \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

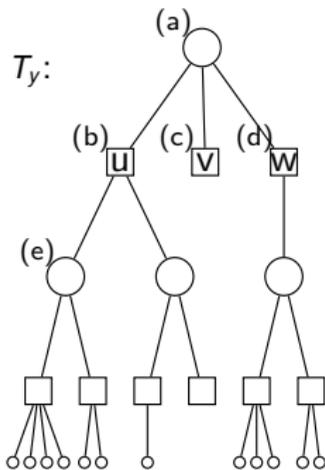
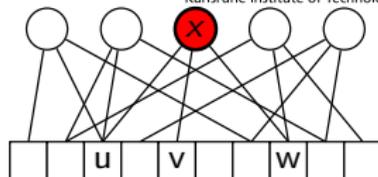
Proof sketch in general (some details omitted)

- General case at i -th \square -node. Want: probability that $G_{m,\alpha m}^{x,R}$ continues to match T_y . Note: T_y is fixed, so i and the number c_i of previously revealed hash values is bounded.

$$\Pr_{Y \sim \text{Bin}(3\lfloor \alpha m \rfloor - c_i, \frac{1}{m-i+1})} [Y = y_i] \xrightarrow{m \rightarrow \infty} \Pr_{Y \sim \text{Pois}(3\alpha)} [Y = y_i].$$

Moreover, those y_i hash values must belong to distinct fresh keys. Probability $\xrightarrow{m \rightarrow \infty} 1$
 \hookrightarrow otherwise we'd have a cycle.

- General case for \circ -node. The two children must be fresh: probability $\xrightarrow{m \rightarrow \infty} 1$
 \hookrightarrow otherwise there would be a cycle.



v Probability that a specific key survives peeling

Lemma

Let $\alpha < c_3^\Delta$. Let x be any \bigcirc -node in $G_{m,\alpha m}$ as before (chosen before sampling the hash functions). Let

$$\mu_m := \Pr_{h_1, h_2, h_3 \sim \mathcal{U}([m]^D)} [x \text{ is removed when peeling } G_{m,\alpha m}].$$

Then $\lim_{m \rightarrow \infty} \mu_m = 1$.

$\mu_m := \Pr[x \text{ is removed when peeling } G_{m,\alpha m}] \xrightarrow{m \rightarrow \infty} 1$

Let $\delta > 0$ be arbitrary. We will show $\lim_{m \rightarrow \infty} \mu_m \geq 1 - 2\delta$.

Let $R \in \mathbb{N}$ be such that $q_R < \delta$.

$\mathcal{Y}^R := \{\text{all possibilities for } T_\alpha^R\}$

$\mathcal{Y}_{\text{peel}}^R := \{T \in \mathcal{Y}^R \mid \text{peeling } T \text{ removes the root}\}$

Let $\mathcal{Y}_{\text{fin}}^R \subseteq \mathcal{Y}^R$ be a *finite* set such that $\Pr[T_\alpha^R \notin \mathcal{Y}_{\text{fin}}^R] \leq \delta$

possible because $\lim_{R \rightarrow \infty} q_R = 0$

note: $\Pr[T_\alpha^R \notin \mathcal{Y}_{\text{peel}}^R] = q_R \leq \delta$.

uses that \mathcal{Y}^R is countable and $\sum_{T \in \mathcal{Y}^R} \Pr[T_\alpha^R = T] = 1$.

\checkmark $\mu_m := \Pr[x \text{ is removed when peeling } G_{m,\alpha m}] \xrightarrow{m \rightarrow \infty} 1$

Let $\delta > 0$ be arbitrary. We will show $\lim_{m \rightarrow \infty} \mu_m \geq 1 - 2\delta$.

Let $R \in \mathbb{N}$ be such that $q_R < \delta$.

$\mathcal{Y}^R := \{\text{all possibilities for } T_\alpha^R\}$

$\mathcal{Y}_{\text{peel}}^R := \{T \in \mathcal{Y}^R \mid \text{peeling } T \text{ removes the root}\}$

Let $\mathcal{Y}_{\text{fin}}^R \subseteq \mathcal{Y}^R$ be a *finite* set such that $\Pr[T_\alpha^R \notin \mathcal{Y}_{\text{fin}}^R] \leq \delta$

$$\begin{aligned}
 \lim_{m \rightarrow \infty} \mu_m &\geq \lim_{m \rightarrow \infty} \Pr[G_{m,\alpha m}^{x,R} \in \mathcal{Y}_{\text{peel}}^R] \\
 &= \lim_{m \rightarrow \infty} \sum_{T \in \mathcal{Y}_{\text{peel}}^R} \Pr[G_{m,\alpha m}^{x,R} = T] \\
 &\geq \lim_{m \rightarrow \infty} \sum_{T \in \mathcal{Y}_{\text{peel}}^R \cap \mathcal{Y}_{\text{fin}}^R} \Pr[G_{m,\alpha m}^{x,R} = T] \\
 &= \sum_{T \in \mathcal{Y}_{\text{peel}}^R \cap \mathcal{Y}_{\text{fin}}^R} \lim_{m \rightarrow \infty} \Pr[G_{m,\alpha m}^{x,R} = T] \\
 &= \sum_{T \in \mathcal{Y}_{\text{peel}}^R \cap \mathcal{Y}_{\text{fin}}^R} \Pr[T_\alpha^R = T] \\
 &= \Pr[T_\alpha^R \in \mathcal{Y}_{\text{peel}}^R \cap \mathcal{Y}_{\text{fin}}^R] = 1 - \Pr[T_\alpha^R \notin \mathcal{Y}_{\text{peel}}^R \cap \mathcal{Y}_{\text{fin}}^R] \\
 &= 1 - \Pr[T_\alpha^R \notin \mathcal{Y}_{\text{peel}}^R \vee T_\alpha^R \notin \mathcal{Y}_{\text{fin}}^R] \\
 &\geq 1 - \Pr[T_\alpha^R \notin \mathcal{Y}_{\text{peel}}^R] - \Pr[T_\alpha^R \notin \mathcal{Y}_{\text{fin}}^R] \geq 1 - 2\delta.
 \end{aligned}$$

possible because $\lim_{R \rightarrow \infty} q_R = 0$

note: $\Pr[T_\alpha^R \notin \mathcal{Y}_{\text{peel}}^R] = q_R \leq \delta$.

uses that \mathcal{Y}^R is countable and $\sum_{T \in \mathcal{Y}^R} \Pr[T_\alpha^R = T] = 1$.

peeling only in R -neighbourhood of x is “weaker”

finite sums commute with limit

previous lemmas

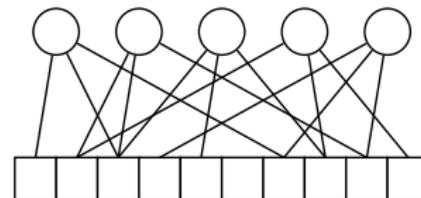
De Morgan's laws: $\overline{A \cap B} = \overline{A} \cup \overline{B}$

union bound: $\Pr[E_1 \vee E_2] \leq \Pr[E_1] + \Pr[E_2]$ \square

vi Proof of the Peeling Theorem

Theorem

Let $\alpha < c_3^\Delta$. Then $\Pr[G_{m,\alpha m} \text{ is peelable}] = 1 - o(1)$.



Proof

Let $n = \lfloor \alpha m \rfloor$ and $0 \leq s \leq n$ the number of \bigcirc nodes surviving peeling.

last lemma: each \bigcirc survives with probability $o(1)$.

linearity of expectation $\mathbb{E}[s] = n \cdot o(1) = o(n)$.

Exercise: $\Pr[s \in \{1, \dots, \delta n\}] = \mathcal{O}(1/m)$ if $\delta > 0$ is a small enough constant.

Markov: $\Pr[s > \delta n] \leq \frac{\mathbb{E}[s]}{\delta n} = \frac{o(n)}{\delta n} = o(1)$.

finally: $\Pr[s > 0] = \Pr[s \in \{1, \dots, \delta n\}] + \Pr[s > \delta n] = \mathcal{O}(1/m) + o(1) = o(1)$. \square

Peeling Process

- greedy algorithm for placing keys in cuckoo table
- works up to a load factor of $c_3^\Delta \approx 0.81$

We saw glimpses of important techniques

- *Local interactions in large graphs*. Also used in statistical physics.
- *Galton-Watson Processes / Trees*. Random processes related to T_α .
- *Local weak convergence*. How the finite graph $G_{m,\alpha m}$ is locally like T_α .

But wait, there's more!

- Further applications of peeling
 - retrieval data structures (next lecture)
 - perfect hash functions (next lecture)
 - set sketches
 - linear error correcting codes

Appendix: Possible Exam Questions I

- Cuckoo hashing and the peeling algorithm
 - (How) can cuckoo hashing be implemented with more than 2 hash functions?
 - What advantage does this provide compared to using 2 hash functions?
 - How does the peeling algorithm work for placing keys in a cuckoo hash table?
 - Peeling can be viewed as a simple process on graphs. How?
 - What is the main result that we proved about the peeling process?
- Proof of the peeling theorem. // *I am aware that the proof is extremely complicated.*
 - In the proof, two graphs played a role: a finite one and a (potentially) infinite one. How were these graphs defined?
 - What is the relationship between the distribution of T_α and the distribution of $G_{m,\alpha,m}$?