An Introduction to Tries

Kevin Leckey

Monash University

21.09.2015

Given: Words, e.g. in binary code

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$

Given: Words, e.g. in binary code

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$

Task: Storage that allows fast search and insert/delete operations

Given: Words, e.g. in binary code

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$

Task: Storage that allows fast search and insert/delete operations

$\rightarrow$ Use tree-like data structures such as a Trie

Given: Words, e.g. in binary code

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$

Task: Storage that allows fast search and insert/delete operations

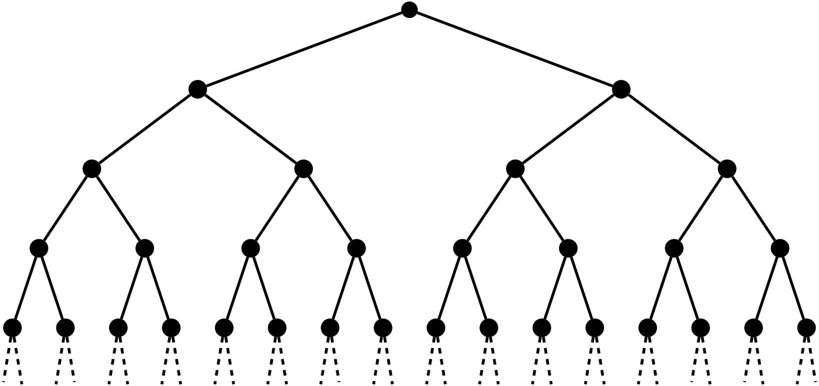
$\rightarrow$ Use tree-like data structures such as a Trie (Information re*trie*val)

Constructing a Trie

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$

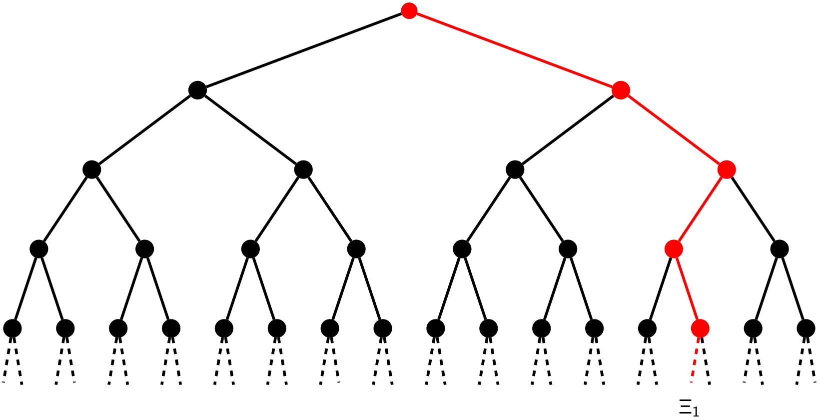
Constructing a Trie

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$



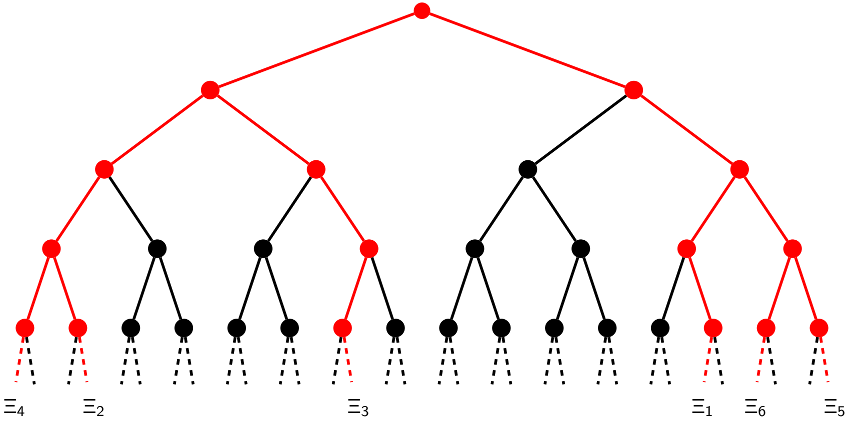
Constructing a Trie

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$



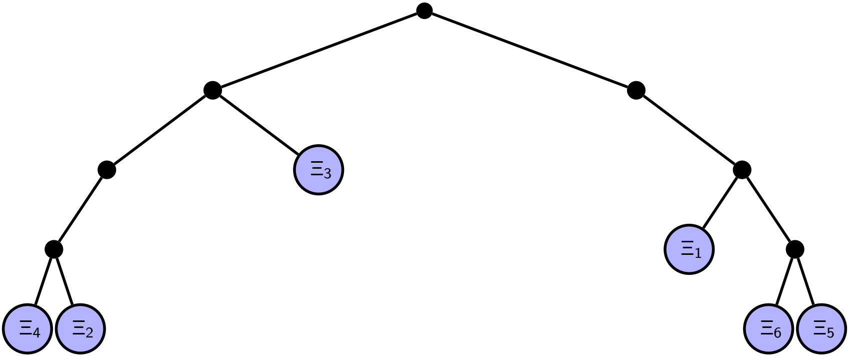
Constructing a Trie

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$



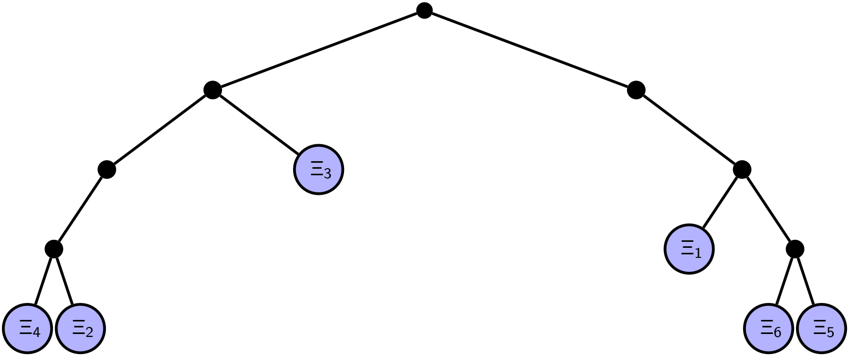
Constructing a Trie

$$\Xi_1 = 11010\ldots, \quad \Xi_2 = 00011\ldots, \quad \Xi_3 = 01101\ldots,$$
$$\Xi_4 = 00000\ldots, \quad \Xi_5 = 11111\ldots, \quad \Xi_6 = 11100\ldots$$



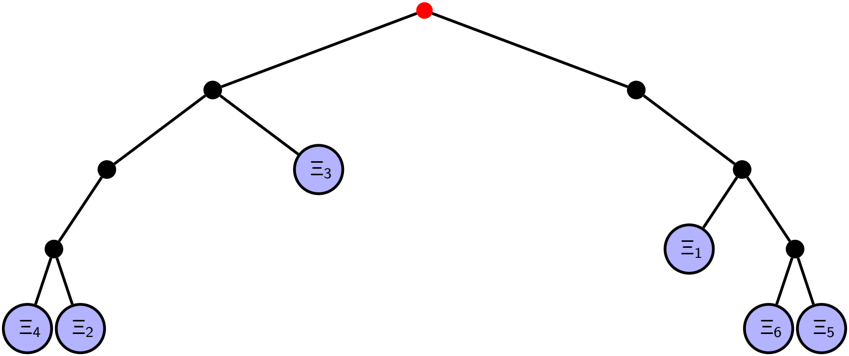
Searching

Search for $\Xi_1 = 11010\ldots$



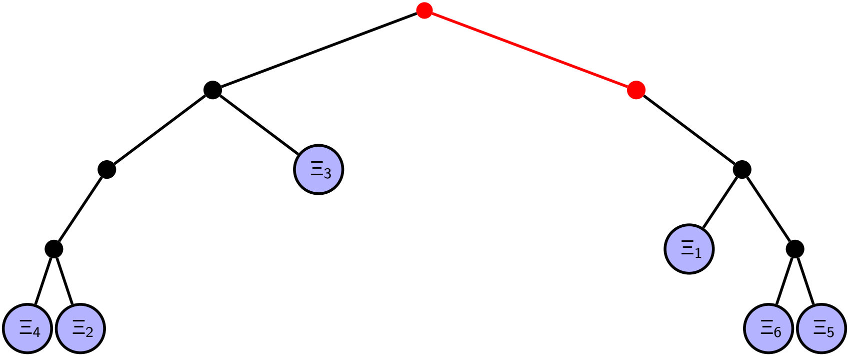
Searching

Search for $\Xi_1 = 11010\ldots$



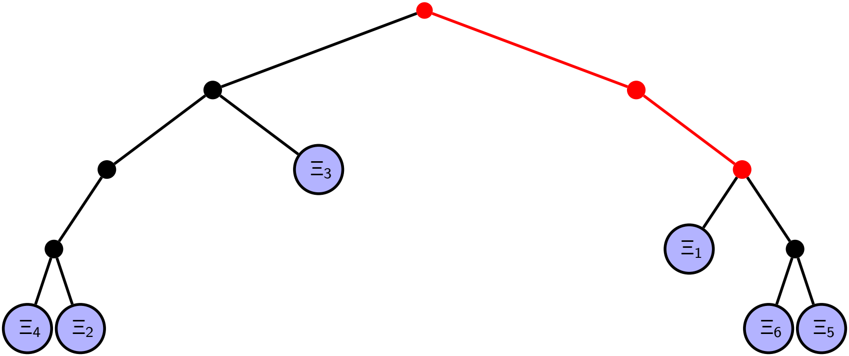
Searching

Search for $\Xi_1 = 11010\ldots$



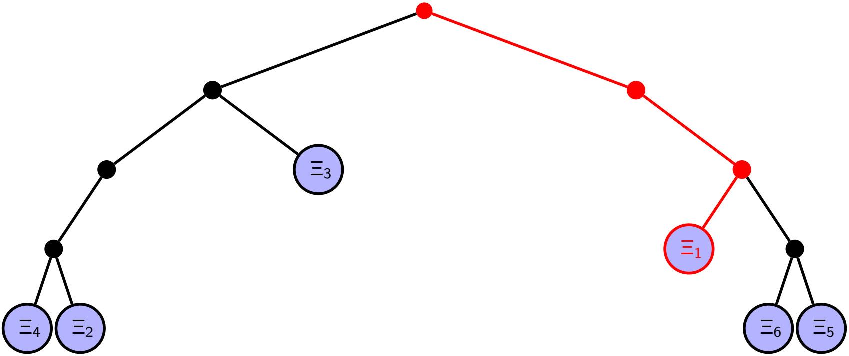
Searching

Search for $\Xi_1 = 11010\ldots$



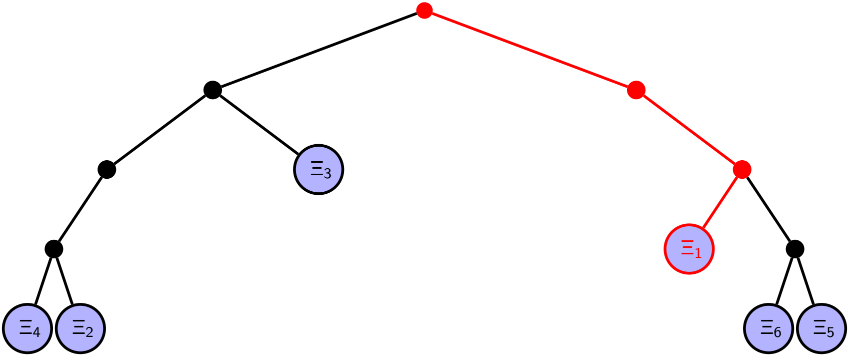
Searching

Search for $\Xi_1 = 11010\ldots$



Searching

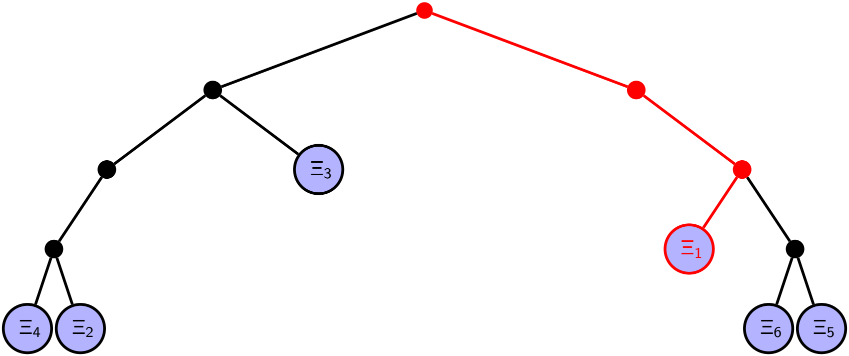
Search for $\Xi_1 = 11010\ldots$



- Searching cost $=$ Depth of Ξ_1

Searching

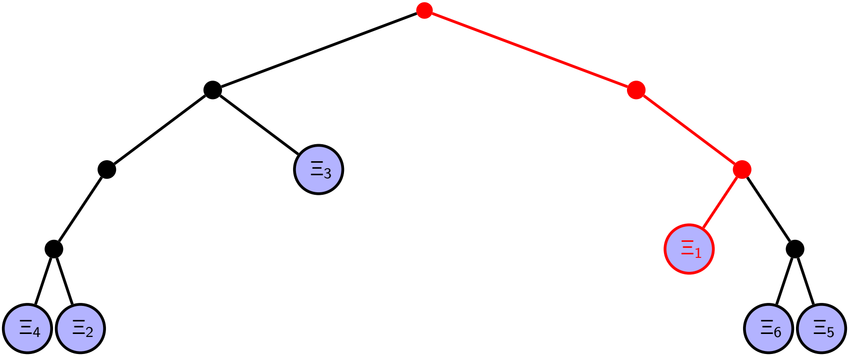
Search for $\Xi_1 = 11010\ldots$



- Searching cost = Depth of Ξ_1
  = Shortest prefix of Ξ_1 not shared by $\Xi_2, \ldots, \Xi_6$

Searching

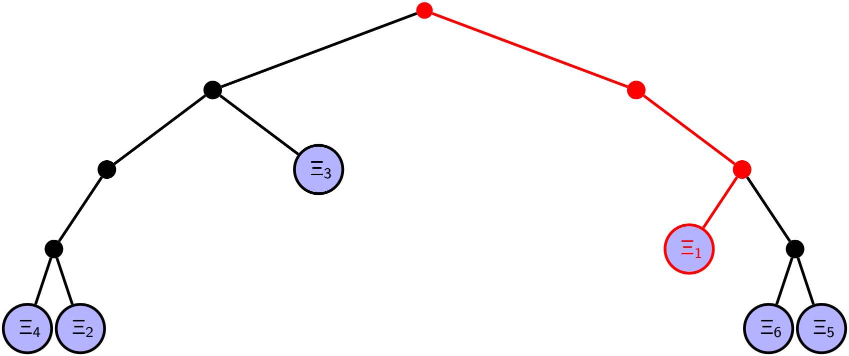
Search for $\Xi_1 = 11010\ldots$



- Searching cost = Depth of Ξ_1
  = Shortest prefix of Ξ_1 not shared by $\Xi_2, \ldots, \Xi_6$
- Worst case = Height of the Trie

Searching

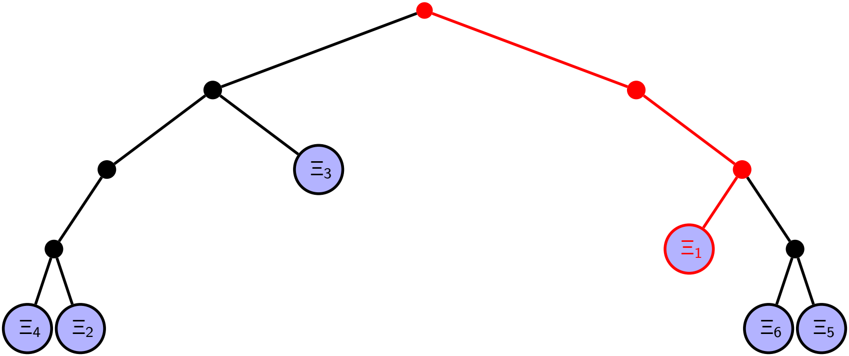
Search for $\Xi_1 = 11010\ldots$



- Searching cost $=$ Depth of $\Xi_1 = 3$
  $=$ Shortest prefix of Ξ_1 not shared by $\Xi_2, \ldots, \Xi_6$
- Worst case $=$ Height of the Trie

Searching

Search for $\Xi_1 = 11010\ldots$



- Searching cost $=$ Depth of $\Xi_1 = 3$
  $=$ Shortest prefix of Ξ_1 not shared by $\Xi_2, \ldots, \Xi_6$
- Worst case $=$ Height of the Trie $= 4$

Input Model

Generate the words $\Xi_1, \Xi_2, \ldots$ to be stored $\rightarrow$ Probabilistic Model

Input Model

Generate the words $\Xi_1, \Xi_2, \ldots$ to be stored $\rightarrow$ Probabilistic Model

- The words $\Xi_1, \Xi_2, \ldots$ are **independent** and identically distributed

Input Model

Generate the words $\Xi_1, \Xi_2, \ldots$ to be stored $\rightarrow$ Probabilistic Model

- The words $\Xi_1, \Xi_2, \ldots$ are **independent** and identically distributed
- Each word $\Xi_i = \xi_1\xi_2\xi_3\xi_4\ldots$ consists of letters $\xi_1, \xi_2, \ldots$ that are:

Input Model

Generate the words $\Xi_1, \Xi_2, \ldots$ to be stored $\rightarrow$ Probabilistic Model

- The words $\Xi_1, \Xi_2, \ldots$ are **independent** and identically distributed
- Each word $\Xi_i = \xi_1\xi_2\xi_3\xi_4 \ldots$ consists of letters $\xi_1, \xi_2, \ldots$ that are:
    - independent,

Input Model

Generate the words $\Xi_1, \Xi_2, \dots$ to be stored $\rightarrow$ Probabilistic Model

- The words $\Xi_1, \Xi_2, \dots$ are **independent** and identically distributed
- Each word $\Xi_i = \xi_1\xi_2\xi_3\xi_4\dots$ consists of letters $\xi_1, \xi_2, \dots$ that are:
  - independent,
  - $\mathbb{P}(\xi_j = 0) = 1/2 = \mathbb{P}(\xi_j = 1)$

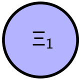
Input Model

Generate the words $\Xi_1, \Xi_2, \ldots$ to be stored $\rightarrow$ Probabilistic Model

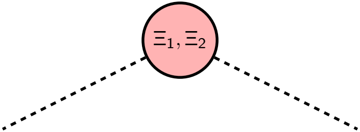
- The words $\Xi_1, \Xi_2, \ldots$ are **independent** and identically distributed
- Each word $\Xi_i = \xi_1\xi_2\xi_3\xi_4\ldots$ consists of letters $\xi_1, \xi_2, \ldots$ that are:
  - independent,
  - $\mathbb{P}(\xi_j = 0) = 1/2 = \mathbb{P}(\xi_j = 1)$

More general models allow $\xi_1, \xi_2, \ldots$ to be dependent
(e.g. evolving as a Markov chain)

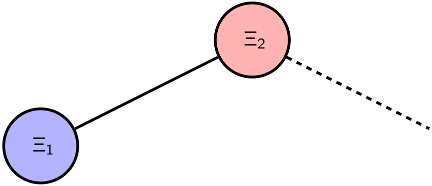
A recursive construction of the Trie



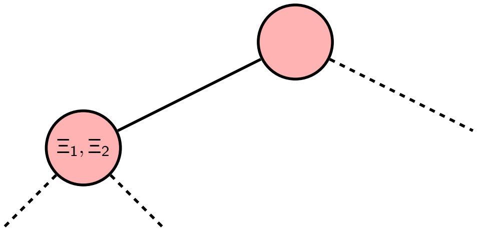
A recursive construction of the Trie



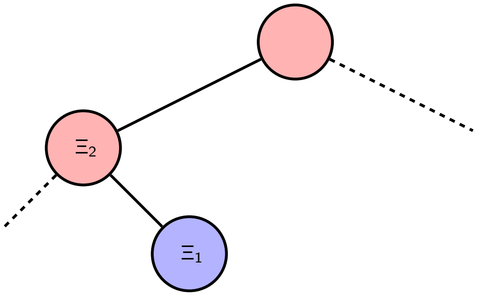
A recursive construction of the Trie



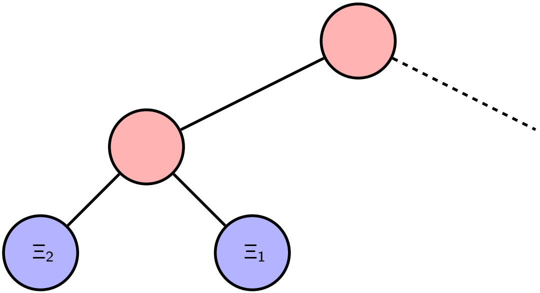
A recursive construction of the Trie



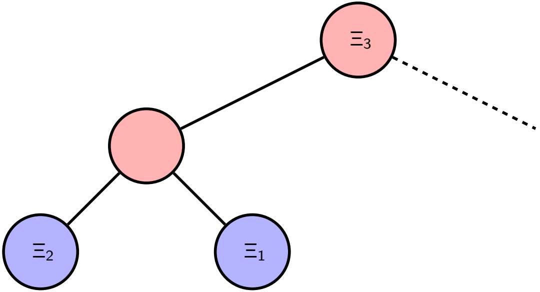
A recursive construction of the Trie



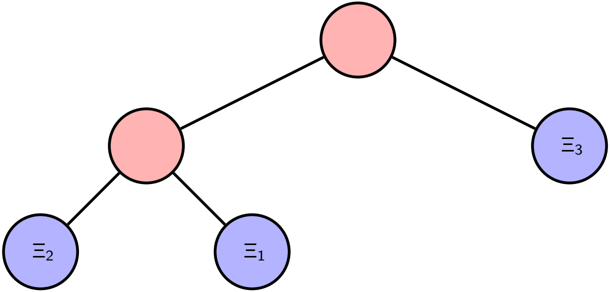
A recursive construction of the Trie



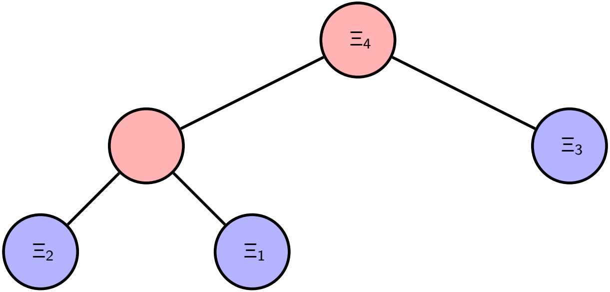
A recursive construction of the Trie



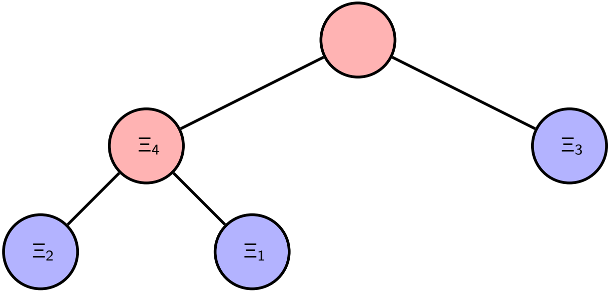
A recursive construction of the Trie



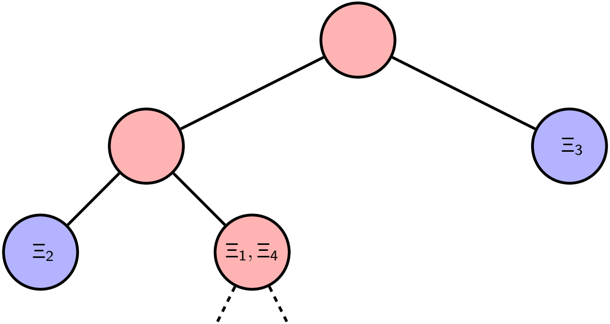
A recursive construction of the Trie



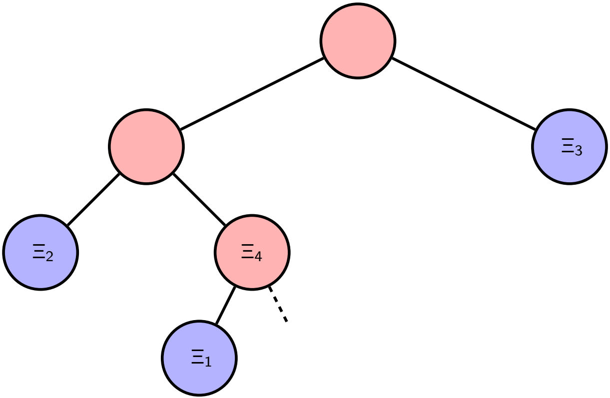
A recursive construction of the Trie



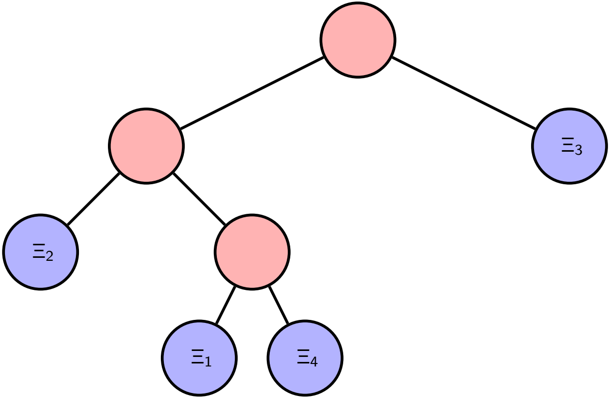
A recursive construction of the Trie



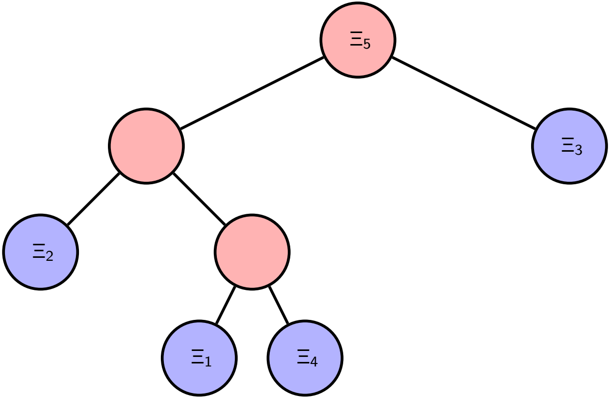
A recursive construction of the Trie



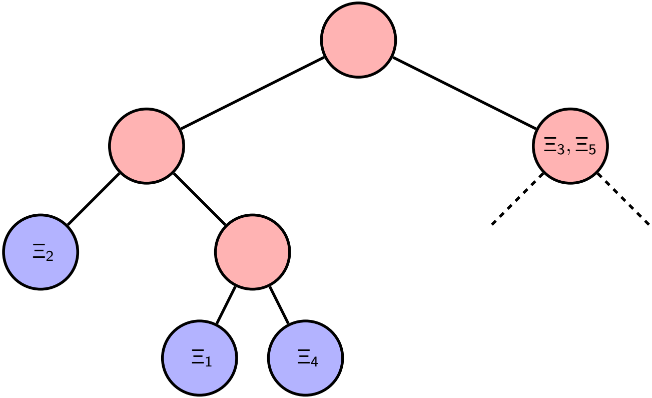
A recursive construction of the Trie



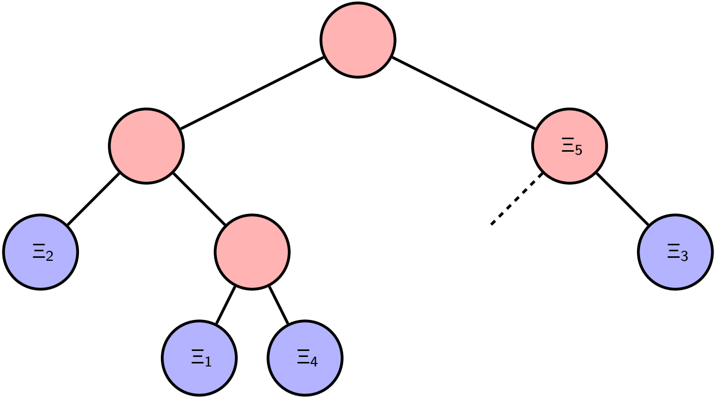
A recursive construction of the Trie



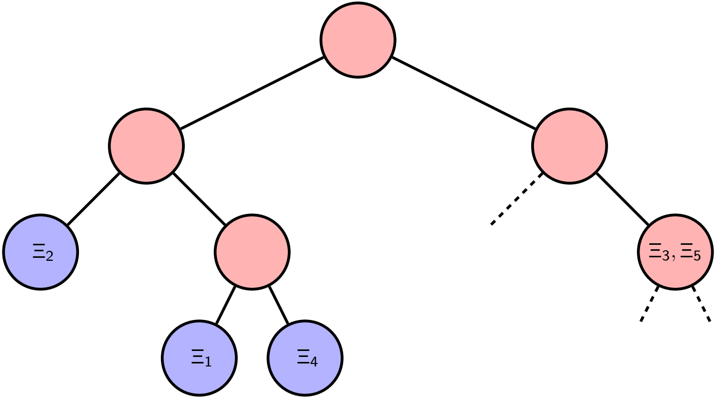
A recursive construction of the Trie



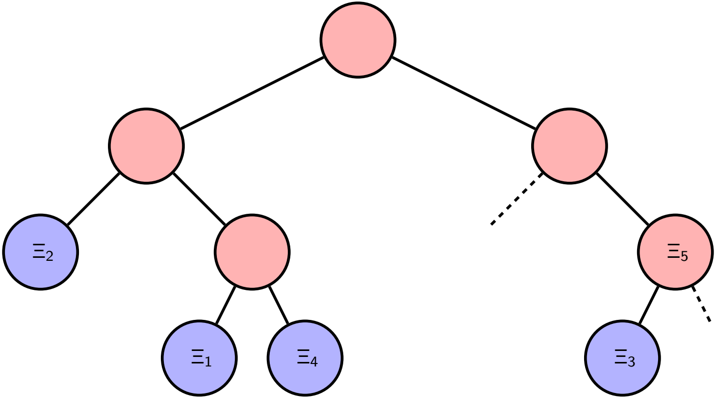
A recursive construction of the Trie



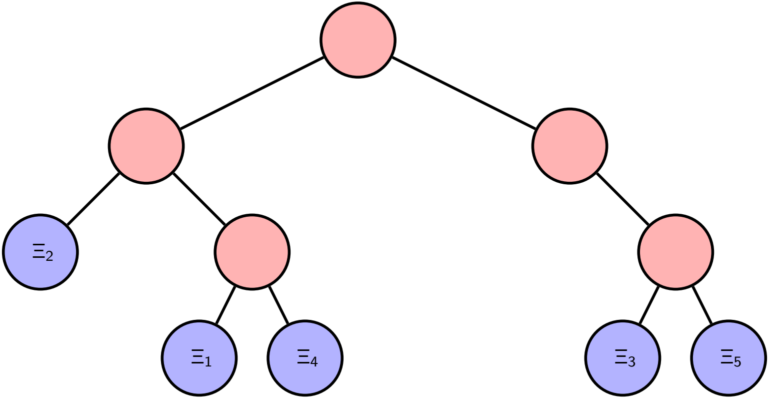
A recursive construction of the Trie



A recursive construction of the Trie



A recursive construction of the Trie



Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Recall:

Depth D_n = Length of the shortest unique prefix of $\Xi_1 = \xi_1 \xi_2 \xi_3 \ldots$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Recall:

Depth D_n = Length of the shortest unique prefix of $\Xi_1 = \xi_1 \xi_2 \xi_3 \ldots$

$$\mathbb{P}(D_n \leq k) = \mathbb{P}(\Xi_2, \ldots, \Xi_n \text{ do not start with } \xi_1 \ldots \xi_k)$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Recall:
Depth D_n = Length of the shortest unique prefix of $\Xi_1 = \xi_1\xi_2\xi_3\ldots$

$$\begin{aligned}\mathbb{P}(D_n \leq k) &= \mathbb{P}(\Xi_2, \ldots, \Xi_n \text{ do not start with } \xi_1 \ldots \xi_k) \\ &= \left(1 - \left(\frac{1}{2}\right)^k\right)^{n-1}\end{aligned}$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Recall:

Depth D_n = Length of the shortest unique prefix of $\Xi_1 = \xi_1\xi_2\xi_3\ldots$

$$\mathbb{P}(D_n \leq k) = \mathbb{P}(\Xi_2, \ldots, \Xi_n \text{ do not start with } \xi_1 \ldots \xi_k)$$
$$= \left(1 - \left(\frac{1}{2}\right)^k\right)^{n-1}$$

Consequence:

$$\mathbb{P}(D_n \leq \alpha \log_2(n)) = \left(1 - n^{-\alpha}\right)^{n-1}$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the depth of the vertex Ξ_1?

Recall:
Depth D_n = Length of the shortest unique prefix of $\Xi_1 = \xi_1\xi_2\xi_3\ldots$

$$\mathbb{P}(D_n \le k) = \mathbb{P}(\Xi_2, \ldots, \Xi_n \text{ do not start with } \xi_1 \ldots \xi_k)$$
$$= \left(1 - \left(\frac{1}{2}\right)^k\right)^{n-1}$$

Consequence:

$$\mathbb{P}(D_n \le \alpha \log_2(n)) = \left(1 - n^{-\alpha}\right)^{n-1} \overset{n\to\infty}{\longrightarrow} \begin{cases} 1, & \text{if } \alpha > 1, \\ 0, & \text{if } \alpha < 1. \end{cases}$$

Results on D_n

- Shown on the previous slide:

$$\frac{D_n}{\log_2(n)} \xrightarrow{\mathbb{P}} 1 \qquad (n \to \infty)$$

Results on D_n

- Shown on the previous slide:

$$\frac{D_n}{\log_2(n)} \xrightarrow{\mathbb{P}} 1 \qquad (n \to \infty)$$

- Considering the previous slide more carefully:

$$\mathbb{P}\left(D_n - \log_2(n) < x\right) \approx \left(1 - \frac{2^{-x}}{n}\right)^{n-1} \overset{n\to\infty}{\longrightarrow} \mathrm{e}^{-2^{-x}}$$

(Limit is a Gumbel distribution known from extreme value theory)

Results on D_n

- Shown on the previous slide:

$$\frac{D_n}{\log_2(n)} \xrightarrow{\mathbb{P}} 1 \qquad (n \to \infty)$$

- Considering the previous slide more carefully:

$$\mathbb{P}\left(D_n - \log_2(n) < x\right) \approx \left(1 - \frac{2^{-x}}{n}\right)^{n-1} \xrightarrow{n\to\infty} \mathrm{e}^{-2^{-x}}$$

(Limit is a Gumbel distribution known from extreme value theory)

- **Thm (Knuth '72):** $\mathbb{E}[D_n] = \log_2(n) + \Psi(\log_2(n)) + o(1)$ with periodic function Ψ

Results on D_n

- Shown on the previous slide:

$$\frac{D_n}{\log_2(n)} \xrightarrow{\mathbb{P}} 1 \qquad (n \to \infty)$$

- Considering the previous slide more carefully:

$$\mathbb{P}\left(D_n - \log_2(n) < x\right) \approx \left(1 - \frac{2^{-x}}{n}\right)^{n-1} \overset{n\to\infty}{\longrightarrow} \mathrm{e}^{-2^{-x}}$$

(Limit is a Gumbel distribution known from extreme value theory)

- **Thm (Knuth '72):** $\mathbb{E}[D_n] = \log_2(n) + \Psi(\log_2(n)) + o(1)$ with periodic function Ψ
- **Thm (Szpankowski '86):** $\mathrm{Var}(D_n) \sim \Phi(\log_2(n))$ with periodic function Φ

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \,:\, i = 1, \ldots, n\}$.

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) : i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \leq k) = (1 - 2^{-k})^{n-1}$ implies:

$$\mathbb{P}\left(H_n > \alpha \log_2(n)\right) =$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \,:\, i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \leq k) = (1 - 2^{-k})^{n-1}$ implies:

$$\mathbb{P}\left(H_n > \alpha \log_2(n)\right) = \mathbb{P}\left(D_n(\Xi_i) > \alpha \log_2(n) \text{ for some } i \in \{1, \ldots, n\}\right)$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \,:\, i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \le k) = (1 - 2^{-k})^{n-1}$ implies:

$$\begin{aligned}\mathbb{P}\left(H_n > \alpha \log_2(n)\right) &= \mathbb{P}\left(D_n(\Xi_i) > \alpha \log_2(n) \text{ for some } i \in \{1, \ldots, n\}\right) \\ &\le n \cdot \mathbb{P}\left(D_n > \alpha \log_2(n)\right)\end{aligned}$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \, : \, i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \leq k) = (1 - 2^{-k})^{n-1}$ implies:

$$\begin{aligned} \mathbb{P}\left(H_n > \alpha \log_2(n)\right) &= \mathbb{P}\left(D_n(\Xi_i) > \alpha \log_2(n) \text{ for some } i \in \{1, \ldots, n\}\right) \\ &\leq n \cdot \mathbb{P}\left(D_n > \alpha \log_2(n)\right) \\ &\leq n \cdot \left(1 - \left(1 - n^{-\alpha}\right)^n\right) \end{aligned}$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \,:\, i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \leq k) = (1 - 2^{-k})^{n-1}$ implies:

$$\begin{aligned}
\mathbb{P}\left(H_n > \alpha \log_2(n)\right) &= \mathbb{P}\left(D_n(\Xi_i) > \alpha \log_2(n) \text{ for some } i \in \{1, \ldots, n\}\right) \\
&\leq n \cdot \mathbb{P}\left(D_n > \alpha \log_2(n)\right) \\
&\leq n \cdot \left(1 - \left(1 - n^{-\alpha}\right)^n\right) \\
&\leq n^{2-\alpha}
\end{aligned}$$

Consider n words $\Xi_1, \ldots, \Xi_n$. What is the height of the resulting Trie?

Def: Height $H_n = \max\{D_n(\Xi_i) \,:\, i = 1, \ldots, n\}$.

The result $\mathbb{P}(D_n \leq k) = (1 - 2^{-k})^{n-1}$ implies:

$$\begin{aligned}
\mathbb{P}\left(H_n > \alpha \log_2(n)\right) &= \mathbb{P}\left(D_n(\Xi_i) > \alpha \log_2(n) \text{ for some } i \in \{1, \ldots, n\}\right) \\
&\leq n \cdot \mathbb{P}\left(D_n > \alpha \log_2(n)\right) \\
&\leq n \cdot \left(1 - \left(1 - n^{-\alpha}\right)^n\right) \\
&\leq n^{2-\alpha}
\end{aligned}$$

Consequence: $\mathbb{P}\left(H_n > \alpha \log_2(n)\right) \to 0$ for $\alpha > 2$

Results on H_n

- Partly proven on the previous slide:

$$\frac{H_n}{2\log_2(n)} \xrightarrow{\mathbb{P}} 1$$

Results on H_n

- Partly proven on the previous slide:

$$\frac{H_n}{2\log_2(n)} \xrightarrow{\mathbb{P}} 1$$

- **Thm (Devroye '84):**

$$\lim_{n\to\infty} \mathbb{P}(H_n - 2\log_2(n) - 1 \leq x) = \exp(-2^{-x}), \quad x \in \mathbb{R}$$

Results on H_n

- Partly proven on the previous slide:

$$\frac{H_n}{2\log_2(n)} \xrightarrow{\mathbb{P}} 1$$

- **Thm (Devroye '84):**

$$\lim_{n\to\infty} \mathbb{P}(H_n - 2\log_2(n) - 1 \le x) = \exp(-2^{-x}), \quad x \in \mathbb{R}$$

- **Thm (Regnier '82):**

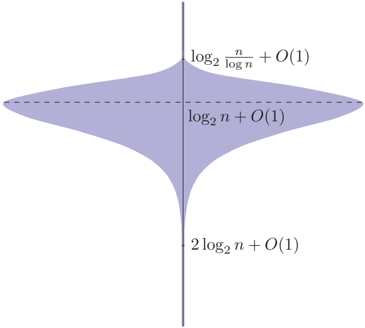
$$\mathbb{E}[H_n] \sim 2\log_2(n) \qquad (n \to \infty)$$

(Flajolet, Steyaert '82 $\rightarrow$ periodic second order term)

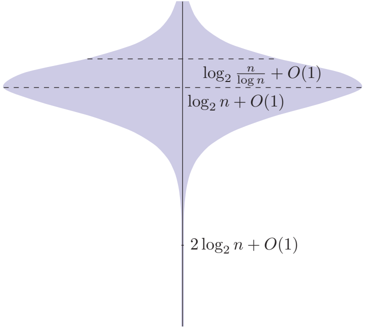
Summary: Typical depth: $\log_2(n)$, height: $2\log_2(n)$.

Summary: Typical depth: $\log_2(n)$, height: $2\log_2(n)$.

Profile (Park, Hwang, Nicodème, Szpankowski):



(External nodes/Leaves)



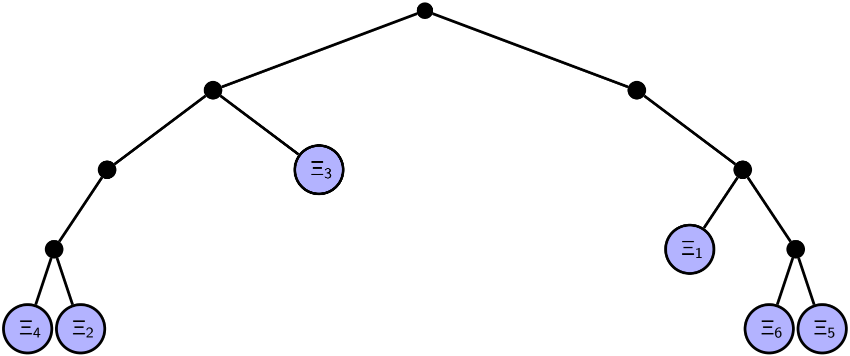
(Internal nodes)

Consider n words $\Xi_1, \ldots, \Xi_n$. External Path Length:

$$L_n := \sum_{i=1}^{n} D_{n,i}, \qquad D_{n,i} = D_n(\Xi_i).$$

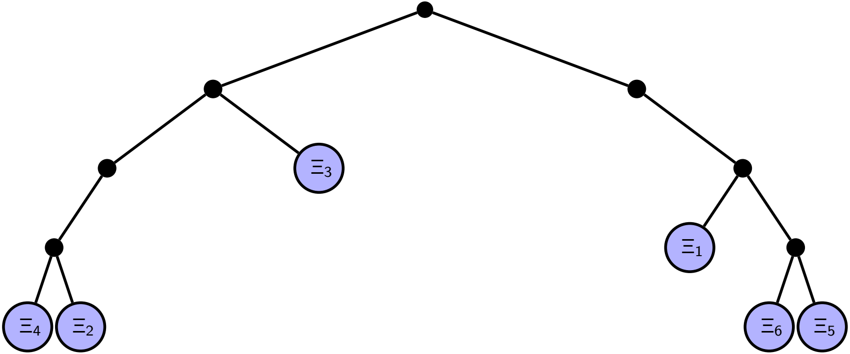
Consider n words $\Xi_1, \ldots, \Xi_n$. External Path Length:

$$L_n := \sum_{i=1}^{n} D_{n,i}, \qquad D_{n,i} = D_n(\Xi_i).$$



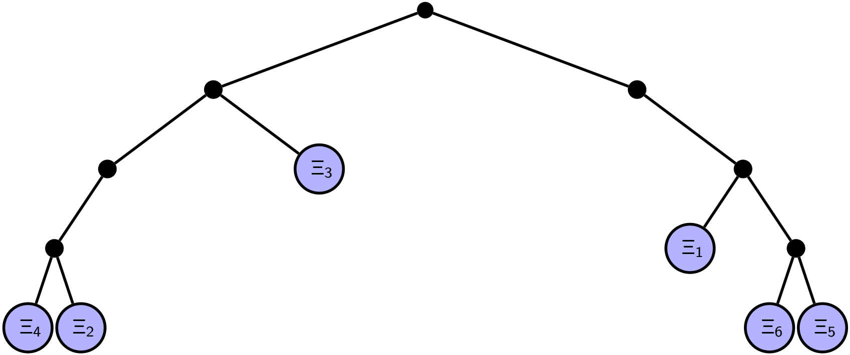
Consider n words $\Xi_1, \ldots, \Xi_n$. External Path Length:

$$L_n := \sum_{i=1}^{n} D_{n,i}, \qquad D_{n,i} = D_n(\Xi_i).$$

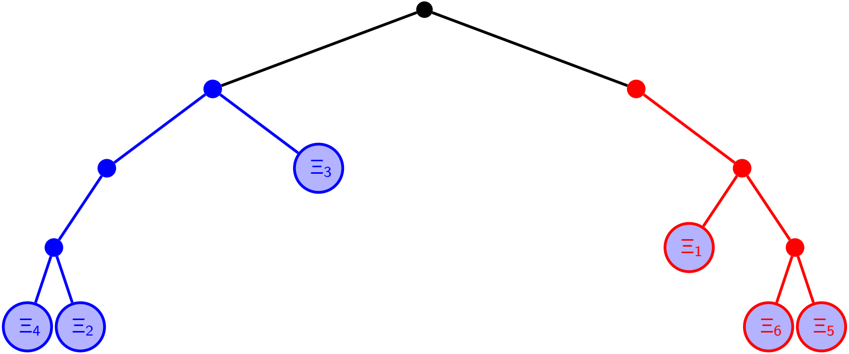


Example: $L_6 = 2 + 3 + 4 \cdot 4 = 21$

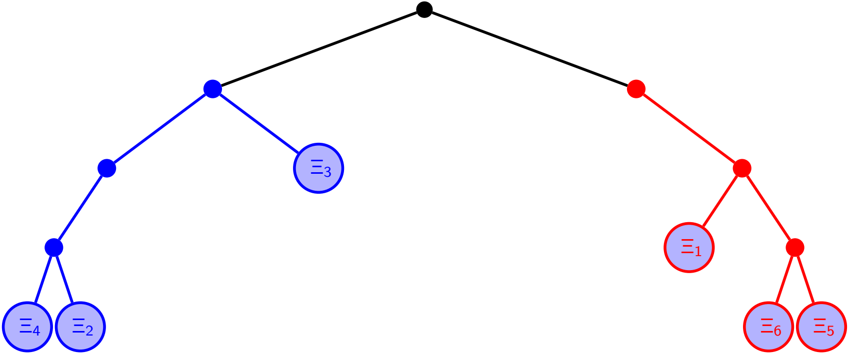
A Recursion for L_n



A Recursion for L_n

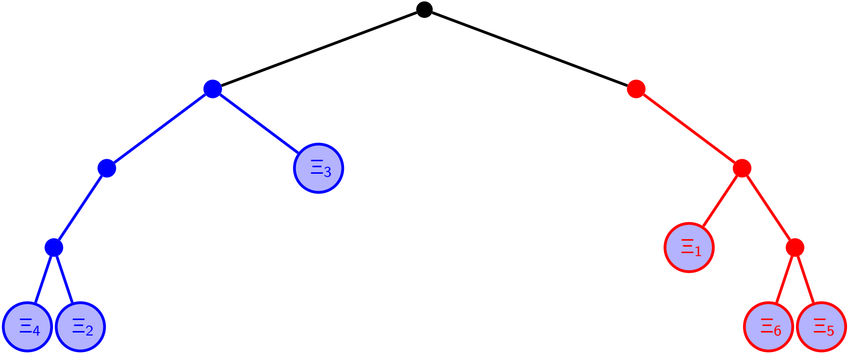


A Recursion for L_n



$K_n = \#$ words starting with 0

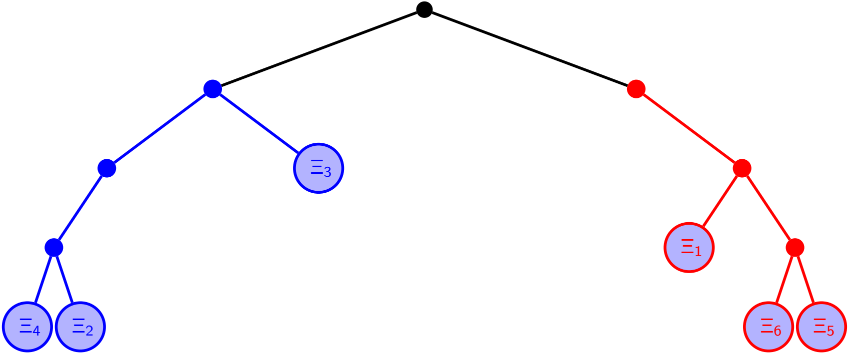
A Recursion for L_n



$K_n = \#$ words starting with 0

$$L_n \stackrel{d}{=}$$

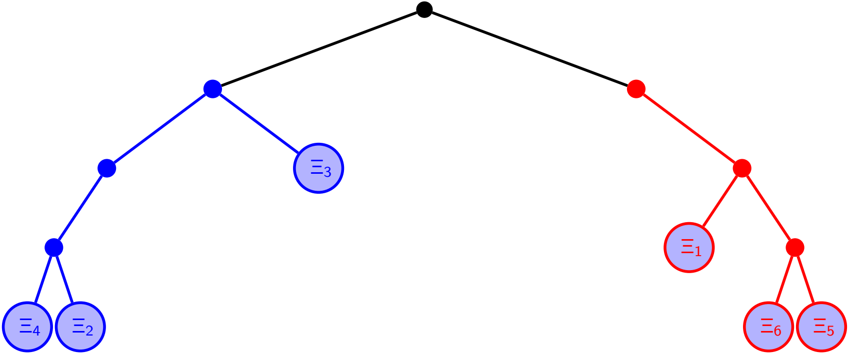
A Recursion for L_n



$K_n = \#$ words starting with 0

$$L_n \stackrel{d}{=} L_{K_n}$$

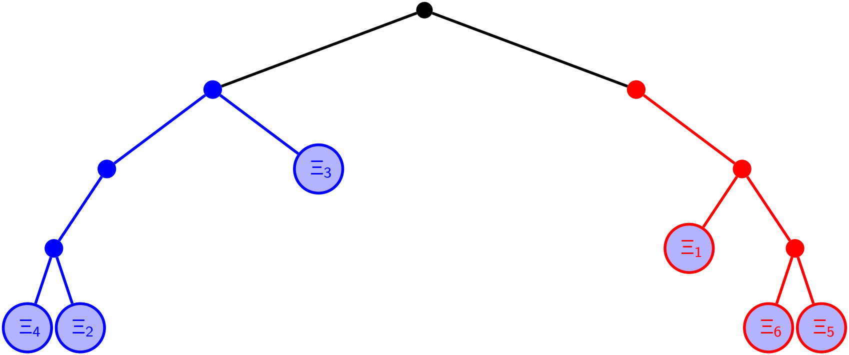
A Recursion for L_n



$K_n = \#$ words starting with 0

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n}$$

A Recursion for L_n



$K_n = \#$ words starting with 0

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ???$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ((\sqrt{2})^{-1}, (\sqrt{2})^{-1}, 0)$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ((\sqrt{2})^{-1}, (\sqrt{2})^{-1}, 0)$

$$X \stackrel{d}{=} \frac{1}{\sqrt{2}} X + \frac{1}{\sqrt{2}} \widetilde{X} \qquad (1)$$

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ((\sqrt{2})^{-1}, (\sqrt{2})^{-1}, 0)$

$$X \stackrel{d}{=} \frac{1}{\sqrt{2}} X + \frac{1}{\sqrt{2}} \widetilde{X} \tag{1}$$

3. Solution to (1)**:** Existence of a solution to (1). Here: Normal distribution with mean 0.

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ((\sqrt{2})^{-1}, (\sqrt{2})^{-1}, 0)$

$$X \stackrel{d}{=} \frac{1}{\sqrt{2}} X + \frac{1}{\sqrt{2}} \widetilde{X} \qquad (1)$$

3. Solution to (1): Existence of a solution to (1). Here: Normal distribution with mean 0.

4. Contraction: Find a metric such that (1) corresponds to the fixed point of a contracting map.

The Contraction Method in a Nutshell

Aim: Find a limit law for L_n (after rescaling properly)

$$L_n \stackrel{d}{=} L_{K_n} + \tilde{L}_{n-K_n} + n$$

1. Rescaling: $X_n = (L_n - \mathbb{E}[L_n])/\sqrt{\mathrm{Var}(L_n)}$

$$X_n \stackrel{d}{=} A_{n,1} X_{K_n} + A_{n,2} \widetilde{X}_{n-K_n} + b_n$$

2. Find the Limits: $(A_{n,1}, A_{n,2}, b_n) \longrightarrow ((\sqrt{2})^{-1}, (\sqrt{2})^{-1}, 0)$

$$X \stackrel{d}{=} \frac{1}{\sqrt{2}} X + \frac{1}{\sqrt{2}} \widetilde{X} \tag{1}$$

3. Solution to (1)**:** Existence of a solution to (1). Here: Normal distribution with mean 0.

4. Contraction: Find a metric such that (1) corresponds to the fixed point of a contracting map.

5. Convergence: Prove convergence with respect to that metric.

Results on L_n

- **Thm (Jacquet, Regnier '88; Neininger, Rüschendorf 2004):**

$$\frac{L_n - \mathbb{E}[L_n]}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{d} \mathcal{N}(0,1)$$

Results on L_n

- **Thm (Jacquet, Regnier '88; Neininger, Rüschendorf 2004):**

$$\frac{L_n - \mathbb{E}[L_n]}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{d} \mathcal{N}(0,1)$$

- From the analysis of D_n:

$$\mathbb{E}[L_n] = \mathbb{E}\left[\sum_{i=1}^{n} D_n(\Xi_i)\right]$$

Results on L_n

- **Thm (Jacquet, Regnier '88; Neininger, Rüschendorf 2004):**

$$\frac{L_n - \mathbb{E}[L_n]}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{d} \mathcal{N}(0,1)$$

- From the analysis of D_n:

$$\mathbb{E}[L_n] = \mathbb{E}\left[\sum_{i=1}^{n} D_n(\Xi_i)\right] = n\mathbb{E}[D_n]$$

Results on L_n

- **Thm (Jacquet, Regnier '88; Neininger, Rüschendorf 2004):**

$$\frac{L_n - \mathbb{E}[L_n]}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{d} \mathcal{N}(0,1)$$

- From the analysis of D_n:

$$\mathbb{E}[L_n] = \mathbb{E}\left[\sum_{i=1}^{n} D_n(\Xi_i)\right] = n\mathbb{E}[D_n] = n\log_2(n) + n\Psi(\log_2(n)) + o(n)$$

Results on L_n

- **Thm (Jacquet, Regnier '88; Neininger, Rüschendorf 2004):**

$$\frac{L_n - \mathbb{E}[L_n]}{\sqrt{\mathrm{Var}(L_n)}} \xrightarrow{d} \mathcal{N}(0,1)$$

- From the analysis of D_n:

$$\mathbb{E}[L_n] = \mathbb{E}\left[\sum_{i=1}^{n} D_n(\Xi_i)\right] = n\mathbb{E}[D_n] = n\log_2(n) + n\Psi(\log_2(n)) + o(n)$$

- **Thm (Kirschenhofer, Prodinger '86):**

$$\mathrm{Var}(L_n) = n\widetilde{\Psi}(\log_2(n)) + \mathrm{O}(log^2(n))$$

Trie:

- tree-like data structure to store words

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Performance:

- Consider input: n independent words, each word is a sequence of 'coin tosses'

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Performance:

- Consider input: n independent words, each word is a sequence of 'coin tosses'
    - Typical search/insert time (depth): around $\log_2(n)$

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Performance:

- Consider input: n independent words, each word is a sequence of 'coin tosses'
  - Typical search/insert time (depth): around $\log_2(n)$
  - Worst search/insert time (height): around $2\log_2(n)$

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Performance:

- Consider input: n independent words, each word is a sequence of 'coin tosses'
  - Typical search/insert time (depth): around $\log_2(n)$
  - Worst search/insert time (height): around $2\log_2(n)$
  - Construction cost (path length): around $n\log_2(n)$

Trie:

- tree-like data structure to store words
- position of a word in the tree $\leftrightarrow$ path given by shortest unique prefix

Performance:

- Consider input: n independent words, each word is a sequence of 'coin tosses'
  - Typical search/insert time (depth): around $\log_2(n)$
  - Worst search/insert time (height): around $2\log_2(n)$
  - Construction cost (path length): around $n\log_2(n)$
- Input model not very realistic, what about more general input models?

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**
- Each word $\Xi_k = \xi_1\xi_2\xi_3\ldots$ has letters $(\xi_j)_{j\geq 1}$ which are a Markov chain on $\{0,1\}$, i.e. for some $\mu = (\mu_0, \mu_1)$ and $P = (p_{ij})_{i,j\in\{0,1\}}$

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**
- Each word $\Xi_k = \xi_1\xi_2\xi_3\ldots$ has letters $(\xi_j)_{j\geq 1}$ which are a Markov chain on $\{0,1\}$, i.e. for some $\mu = (\mu_0, \mu_1)$ and $P = (p_{ij})_{i,j\in\{0,1\}}$
  - $\mathbb{P}(\xi_1 = a) = \mu_a$,

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**
- Each word $\Xi_k = \xi_1 \xi_2 \xi_3 \ldots$ has letters $(\xi_j)_{j \geq 1}$ which are a Markov chain on $\{0, 1\}$, i.e. for some $\mu = (\mu_0, \mu_1)$ and $P = (p_{ij})_{i,j \in \{0,1\}}$

$$\bullet\ \mathbb{P}(\xi_1 = a) = \mu_a, \qquad \bullet\ \mathbb{P}(\xi_{j+1} = a | \xi_1, \ldots, \xi_j) = p_{\xi_j a}$$

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**
- Each word $\Xi_k = \xi_1 \xi_2 \xi_3 \ldots$ has letters $(\xi_j)_{j \geq 1}$ which are a Markov chain on $\{0, 1\}$, i.e. for some $\mu = (\mu_0, \mu_1)$ and $P = (p_{ij})_{i,j \in \{0,1\}}$

$$\bullet\ \mathbb{P}(\xi_1 = a) = \mu_a, \qquad \bullet\ \mathbb{P}(\xi_{j+1} = a | \xi_1, \ldots, \xi_j) = p_{\xi_j a}$$

More general (Markov Model with k-dependency):
distribution of ξ_j depends only on the previous k letters for some fixed k

Markov Model

Generate n words $\Xi_1, \ldots, \Xi_n$ such that

- the words $\Xi_1, \ldots, \Xi_n$ are **independent** and **identically distributed**
- Each word $\Xi_k = \xi_1\xi_2\xi_3\ldots$ has letters $(\xi_j)_{j\geq 1}$ which are a Markov chain on $\{0,1\}$, i.e. for some $\mu = (\mu_0, \mu_1)$ and $P = (p_{ij})_{i,j\in\{0,1\}}$
  - $\mathbb{P}(\xi_1 = a) = \mu_a$,
  - $\mathbb{P}(\xi_{j+1} = a | \xi_1, \ldots, \xi_j) = p_{\xi_j a}$

More general (Markov Model with k-dependency):
distribution of ξ_j depends only on the previous k letters for some fixed k

Even more general: Dynamical Sources Model by Vallée

Effect on Depth and related parameters:

Are there very 'typical' long prefixes for the source (e.g. because p_{aa} is very large)?

Effect on Depth and related parameters:

Are there very 'typical' long prefixes for the source (e.g. because p_{aa} is very large)? $\rightarrow$ Depth/Height gets very large

Effect on Depth and related parameters:
Are there very 'typical' long prefixes for the source (e.g. because p_{aa} is very large)? $\rightarrow$ Depth/Height gets very large

Entropy in the Markov Source Model:

$$\mathcal{H} = \pi_0 \left(-p_{00} \log(p_{00}) - p_{01} \log(p_{01})\right) + \pi_1 \left(-p_{10} \log(p_{10}) - p_{11} \log(p_{11})\right)$$

with stationary distribution

$$(\pi_0, \pi_1) = \left(\frac{p_{10}}{p_{10} + p_{01}}, \frac{p_{01}}{p_{10} + p_{01}}\right)$$

Effect on Depth and related parameters:
Are there very 'typical' long prefixes for the source (e.g. because p_{aa} is very large)? $\rightarrow$ Depth/Height gets very large

Entropy in the Markov Source Model:

$$\mathcal{H} = \pi_0 \left(-p_{00} \log(p_{00}) - p_{01} \log(p_{01})\right) + \pi_1 \left(-p_{10} \log(p_{10}) - p_{11} \log(p_{11})\right)$$

with stationary distribution

$$(\pi_0, \pi_1) = \left(\frac{p_{10}}{p_{10} + p_{01}}, \frac{p_{01}}{p_{10} + p_{01}}\right)$$

Depth for Markov Sources:

$$\mathbb{E}[D_n] \sim \frac{1}{\mathcal{H}} \log(n)$$

Results for the Markov Source Model

- **Depth:** Jacquet, Szpankowski ’89
- **Height:** Szpankowski ’91
- **External Pathlength:** L., Neininger, Szpankowski (SODA 2013)

Results for the Markov Source Model

- **Depth:** Jacquet, Szpankowski '89
- **Height:** Szpankowski '91
- **External Pathlength:** L., Neininger, Szpankowski (SODA 2013)

Dynamical Sources: Clément, Flajolet, Vallée 2001

Results for the Markov Source Model

- **Depth:** Jacquet, Szpankowski '89
- **Height:** Szpankowski '91
- **External Pathlength:** L., Neininger, Szpankowski (SODA 2013)

Dynamical Sources: Clément, Flajolet, Vallée 2001

Some related problems:

- PATRICIA Tries and Digital Search Trees (Thesis L.$\rightarrow$ Pathlength)

Results for the Markov Source Model

- **Depth:** Jacquet, Szpankowski '89
- **Height:** Szpankowski '91
- **External Pathlength:** L., Neininger, Szpankowski (SODA 2013)

Dynamical Sources: Clément, Flajolet, Vallée 2001

Some related problems:

- PATRICIA Tries and Digital Search Trees (Thesis L.$\rightarrow$ Pathlength)
- Radix-Sort and -Select (Thesis L.)

Results for the Markov Source Model

- **Depth:** Jacquet, Szpankowski '89
- **Height:** Szpankowski '91
- **External Pathlength:** L., Neininger, Szpankowski (SODA 2013)

Dynamical Sources: Clément, Flajolet, Vallée 2001

Some related problems:

- PATRICIA Tries and Digital Search Trees (Thesis L.$\rightarrow$ Pathlength)
- Radix-Sort and -Select (Thesis L.)
- Lempel-Ziv Parsing Scheme (data compression)