Non-linearity in Davenport-Schinzel Sequences

Seth Pettie
University of Michigan
Isomorphism and Subsequences

- Political Isomorphism
  - BUSH is isomorphic to GORE

Seth Pettie
Isomorphism and Subsequences

• Political Isomorphism
  • **BUSH** is isomorphic to **GORE**
  • **C** is isomorphic to **A**
Isomorphism and Subsequences

- Political Isomorphism
  - BUSH is isomorphic to GORE
  - C is isomorphic to A
  - THOMAS is isomorphic to SOUTER
Isomorphism and Subsequences

- Political Isomorphism
  - BUSH is isomorphic to GORE
  - C is isomorphic to A
  - THOMAS is isomorphic to SOUTER
  - CIA, NSA, DOD is not isomorphic to NSF, EPA, NIH
Isomorphism and Subsequences

- Political Isomorphism
  - \textbf{BUSH} is isomorphic to \textbf{GORE}
  - \textbf{C} is isomorphic to \textbf{A}
  - \textbf{THOMAS} is isomorphic to \textbf{SOUTER}
  - \textbf{CIA,NSA,DOD} is \textit{not} isomorphic to \textbf{NSF,EPA,NIH}

- Happiness via Subsequences
  - \texttt{WITH\_WHOM\_WOULD\_I\_RATHER\_HAVE\_A\_BEER?}
Isomorphism and Subsequences

- Political Isomorphism
  - **BUSH** is isomorphic to **GORE**
  - **C** is isomorphic to **A**
  - **THOMAS** is isomorphic to **SOUTER**
  - **CIA, NSA, DOD** is not isomorphic to **NSF, EPA, NIH**

- Happiness via Subsequences
  - **WITH_WHOM_WOULD_I_RATHER_HAVE_A_BEER?**
  - **TH_WHO LD R VE**?
Isomorphism and Subsequences

- Political Isomorphism
  - **BUSH** is isomorphic to **GORE**
  - **C** is isomorphic to **A**
  - **THOMAS** is isomorphic to **SOUTER**
  - **CIA, NSA, DOD** is **not** isomorphic to **NSF, EPA, NIH**

- Happiness via Subsequences
  - **WITH_WHOM_WOULD_I_RATHER_HAVE_A_BEER?**
  - **TH_WHO LD_ R_ VE ?**
  - **TARJAN FOR PR EZ ?**
Definitions

- $x \subset y : x$ is isomorphic to a subsequence of $y$

- $Ex(\sigma, n) = \max |S| :$
  
  $S \in \{1, \ldots, n\}^*$
  
  $\sigma \not\subset S$

  $S$ is $|\sigma|$-regular (technical condition)

- How fast does $Ex(\sigma, n)$ grow as a function of $n$?
Original application: lower envelopes

(1) Give each object (line segment, quadratic, etc.) a symbol
(2) Map the lower envelope to a sequence $|S|$
(3) Show $|S| \leq \text{Ex}(\sigma,n)$ for some forbidden subseq. $\sigma$

this sequence does not contain ababaababa
Original motivation: lower envelopes

(1) Give each object (line segment, quadratic, etc.) a symbol

(2) Map the lower envelope to a sequence $|S|$

(3) Show $|S| \leq \text{Ex}(\sigma, n)$ for some forbidden subseq. $\sigma$

standard case: $\sigma = \text{ababab...a}$ length $k+2$

“order k Davenport-Schinzel sequence”
Splay trees and Davenport-Schinzel sequences

- Amortized analysis: Normally pay for time consuming ops with a reduction in potential

Seth Pettie
Splay trees and Davenport-Schinzel sequences

- New kind of amortized analysis:
  - *Label nodes* that cannot be paid for by other means
  - *Transcribe the labels* as a sequence $S$:
    \[ |S| \leq \operatorname{Ex}(\sigma, n) \]

- In [SODA’08] $\sigma = \text{abaabba}$ or $\text{abababa}$
  Thm. $n$ deque operations take $O(n\alpha^*(n))$ time
Splay trees and Davenport-Schinzel sequences

- New kind of amortized analysis:
- *Label nodes* that cannot be paid for by other means
- *Transcribe the labels* as a sequence $S$:
  $$|S| \leq \text{Ex}(\sigma, n)$$

A much better way to end the proof:

... where $\text{Ex}(\sigma, n) = O(n)$
Standard Davenport-Schinzel seqs.

- $\alpha = \alpha(n)$: $\alpha$ is the inverse-Ackermann function

<table>
<thead>
<tr>
<th></th>
<th>$\text{Ex}(\text{aba}, n)$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>trivial</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$\text{Ex}(\text{abab}, n)$</td>
<td>$2n-1$</td>
</tr>
</tbody>
</table>

Seth Pettie
Standard Davenport-Schinzel seqs.

\[ \alpha = \alpha(n) \quad \alpha \text{ is the inverse-Ackermann function} \]

<table>
<thead>
<tr>
<th></th>
<th>Ex(a\text{ba}, n)</th>
<th>\ n</th>
</tr>
</thead>
<tbody>
<tr>
<td>trivial</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ex(ab\text{ab}, n)</td>
<td>2n-1</td>
<td></td>
</tr>
<tr>
<td>Hart-Sharir</td>
<td>Ex(ab\text{aba}, n)</td>
<td>\Theta(n\alpha)</td>
</tr>
</tbody>
</table>

Seth Pettie
Standard Davenport-Schinzel seqs.

\( \alpha = \alpha(n) \)  \( \alpha \) is the inverse-Ackermann function

<table>
<thead>
<tr>
<th></th>
<th>( \text{Ex}(ab\alpha, n) )</th>
<th>( n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>trivial</td>
<td>( \text{Ex}(\text{abab}, n) )</td>
<td>( 2n-1 )</td>
</tr>
<tr>
<td>Hart-Sharir</td>
<td>( \text{Ex}(\text{ababa}, n) )</td>
<td>( \Theta(n\alpha) )</td>
</tr>
<tr>
<td>Agarwal-Sharir-Shor</td>
<td>( \text{Ex}(\text{ababab}, n) )</td>
<td>( \Theta(n2^{\alpha}) )</td>
</tr>
</tbody>
</table>
Standard Davenport-Schinzel seqs.

$\alpha = \alpha(n)$ \textit{\textbf{is the inverse-Ackermann function}}

<table>
<thead>
<tr>
<th></th>
<th>$\text{Ex}(\text{aba}, n)$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Ex}(\text{abab}, n)$</td>
<td>$2n-1$</td>
<td></td>
</tr>
</tbody>
</table>

$\alpha = \Theta(n\alpha)$

<table>
<thead>
<tr>
<th></th>
<th>$\text{Ex}(\text{ababa}, n)$</th>
<th>$\Theta(n\alpha)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Ex}(\text{ababab}, n)$</td>
<td>$\Theta(n2^\alpha)$</td>
<td></td>
</tr>
</tbody>
</table>

$\alpha = \Theta(\alpha^2)$

<table>
<thead>
<tr>
<th></th>
<th>$\text{Ex}(\text{ababababa}, n)$</th>
<th>$n \exp(O(\alpha \log \alpha))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{Ex}(\text{ababababab}, n)$</td>
<td>$n \exp(O(\alpha^2 \log \alpha))$</td>
<td></td>
</tr>
</tbody>
</table>

$\alpha = \Theta(\alpha^3)$
Standard Davenport-Schinzel seqs.

\[ \alpha = \alpha(n) \quad \alpha \text{ is the inverse-Ackermann function} \]

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Ex(aba, n)</th>
<th>Ex(abab, n)</th>
<th>Ex(ababa, n)</th>
<th>Ex(abababa, n)</th>
<th>Ex(abababab, n)</th>
<th>Ex(ababababa, n)</th>
<th>Ex(abababababa, n)</th>
<th>Ex(abababababab, n)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n</td>
<td>2n-1</td>
<td>( \Theta(n\alpha) )</td>
<td>( n \exp(O(\alpha \log \alpha)) )</td>
<td>( n \exp(\Theta(\alpha^2)) )</td>
<td>( n \exp(O(\alpha^2 \log \alpha)) )</td>
<td>( n \exp(\Theta(\alpha^3)) )</td>
<td>( n \exp(O(\alpha^{</td>
</tr>
</tbody>
</table>

Trivial

- Hart-Sharir
- Agarwal-Sharir-Shor
- Klazar

Seth Pettie
Two-Letter Forbidden Subsequences

[Adamec-Klazar-Valtr]

\[ \text{Ex}(abbaab,n) = O(n) \]

**The Two-Letter Theorem:**

For any \( \sigma \in \{a,b\}^* \)

\[ \text{Ex}(\sigma,n) = \omega(n) \quad \text{if and only if} \quad ababa \subseteq \sigma \]

(i.e., there is only one “cause” of superlinearity over two symbols)
The Three-Letter Theorem

[ Klazar-Valtr ]

For $\sigma \in \{a,b,c\}^*$

$\text{Ex}(\sigma,n) = O(n)$

unless...

$ababa \subset \sigma$ or
$abcacbc \subset \sigma$ or
$abcbcac \subset \sigma$

or their reversals

non-linear

status still open

Seth Pettie
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i,n) = O(n)$
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) \( \text{Ex}(a^i,n) = O(n) \)

(2) If \( \text{Ex}(uw,n) = O(n) \) and \( \text{Ex}(v,n) = O(n) \)

\[ \text{Ex}(uvw,n) = O(n) \]

For Example: \( \text{Ex}(aabbaabcdddcefgfefgcccbbccdd) = O(n) \)

uw and v have disjoint alphabets
Recipe for linear forbidden sequences

[Klazar-Valtr]

1. \( \text{Ex}(a^i, n) = O(n) \)
2. If \( \text{Ex}(uw, n) = O(n) \) and \( \text{Ex}(v, n) = O(n) \)
   \[ \text{Ex}(uvw, n) = O(n) \]
3. If \( \text{Ex}(uawa, n) = O(n) \)
   \[ \text{Ex}(uabiwabii) = O(n) \]

\( uw \) and \( v \) have disjoint alphabets
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i,n) = O(n)$

(2) If $\text{Ex}(uw,n) = O(n)$ and $\text{Ex}(v,n) = O(n)$

   $\text{Ex}(uvw,n) = O(n)$

(3) If $\text{Ex}(uawa,n) = O(n)$

   $\text{Ex}(uabiwabi) = O(n)$

[Note: $uw$ and $v$ have disjoint alphabets]

aaaaa
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) \( \text{Ex}(a^i, n) = O(n) \)

(2) If \( \text{Ex}(uw, n) = O(n) \) and \( \text{Ex}(v, n) = O(n) \)

\[ \text{Ex}(uvw, n) = O(n) \]

(3) If \( \text{Ex}(uawa, n) = O(n) \)

\[ \text{Ex}(uab^i wab^i) = O(n) \]

\( u \) and \( v \) have disjoint alphabets

\( \text{aabbaabbb} \)
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i, n) = O(n)$

(2) If $\text{Ex}(uw, n) = O(n)$ and $\text{Ex}(v, n) = O(n)$
    
    $\text{Ex}(uvw, n) = O(n)$

(3) If $\text{Ex}(uawa, n) = O(n)$

    $\text{Ex}(uabiwabi) = O(n)$

aabbaabccccccbbccc
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i, n) = O(n)$

(2) If $\text{Ex}(uw, n) = O(n)$ and $\text{Ex}(v, n) = O(n)$
   \[\text{Ex}(uvw, n) = O(n)\]

(3) If $\text{Ex}(uawa, n) = O(n)$
   \[\text{Ex}(uabiwabi) = O(n)\]

aabbaabcdddccccbbccdd
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i,n) = O(n)$

(2) If $\text{Ex}(uw,n) = O(n)$ and $\text{Ex}(v,n) = O(n)$

\[
\text{Ex}(uvw,n) = O(n)
\]

(3) If $\text{Ex}(uawa,n) = O(n)$

\[
\text{Ex}(uabi wabi) = O(n)
\]

aabbaabcdddddcccccbccdd
eee
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i, n) = O(n)$

(2) If $\text{Ex}(uw, n) = O(n)$ and $\text{Ex}(v, n) = O(n)$

\[
\text{Ex}(uvw, n) = O(n)
\]

(3) If $\text{Ex}(uawa, n) = O(n)$

\[
\text{Ex}(uab^i wab^i) = O(n)
\]

aabbaabcdddcceccbbccdd
effef
effef

Seth Pettie
Recipe for linear forbidden sequences

[Klazar-Valtr]

(1) $\text{Ex}(a^i, n) = O(n)$

(2) If $\text{Ex}(uw, n) = O(n)$ and $\text{Ex}(v, n) = O(n)$

   $\text{Ex}(uvw, n) = O(n)$

(3) If $\text{Ex}(uawa, n) = O(n)$

   $\text{Ex}(uabiwabi) = O(n)$

\abcdefgabcdefg

Seth Pettie
Recipe for linear forbidden sequences

[ Klazar-Valtr ]

(1) \( \text{Ex}(a^i,n) = O(n) \)

(2) If \( \text{Ex}(uw,n) = O(n) \) and \( \text{Ex}(v,n) = O(n) \)

\( \text{Ex}(uvw,n) = O(n) \)

(3) If \( \text{Ex}(uawa,n) = O(n) \)

\( \text{Ex}(uabiwabi) = O(n) \)

aabbaabcddddcefgfefgccccbbccddd
efgfefg
More than one cause of non-linearity

- [Klazar]
  - $\sigma$ is a sequence without repetitions
  - $(x,y)$ is in $G(\sigma)$ iff $xyyx \subset \sigma$ or $yxyx \subset \sigma$

- If $G(\sigma)$ is strongly connected then
  $$Ex(\sigma,n) = \Omega(n\alpha(n))$$
More than one cause of non-linearity

- [Klazar]
  - $\sigma$ is a sequence without repetitions
  - $(x,y)$ is in $G(\sigma)$ iff $x y y x \subseteq \sigma$ or $y x y x \subseteq \sigma$

- If $G(\sigma)$ is strongly connected then
  \[ Ex(\sigma, n) = \Omega(n \alpha(n)) \]

\[ G(ababa) \]
\[ G(abcbadadbcdbcd) \]

Only two examples known

Seth Pettie
Another cause of non-linearity

- [Klazar]
  - $\sigma$ is a sequence without repetitions
  - $(x,y)$ is in $G'(\sigma)$ iff $xxyx \subset \sigma$ or $yxyx \subset \sigma$

- If $G'(\sigma)$ is strongly connected then
  \[ \text{Ex}(\sigma,n) = \Omega(n\alpha(n)) \leq \Omega(n2^{\alpha(n)}) \]

G'(ababab)

G'(abcbadabecfdefedef)

Only two examples known

Seth Pettie
Defn. $\Phi = \text{minimal non-linear forbidden seqs.}$

What we know about $\Phi$:
- $ababa \in \Phi$
- $|\Phi| \geq 2$ (the other a subseq of abcbadadbcd)
Defn. $\Phi = \text{minimal non-linear forbidden seqs.}$

What we know about $\Phi$:
- $ababa \in \Phi$
- $|\Phi| \geq 2$ (the other a subseq of abcbadaddbcd)

Q: Is $|\Phi|$ infinite?

A: Still Open. But we have a candidate!
Defn. $\Phi = \text{minimal non-linear forbidden seqs.}$

What we know about $\Phi$:
- $ababa \in \Phi$
- $|\Phi| \geq 2$ (the other a subseq of abcbadadbcd)

Q: Is $|\Phi|$ infinite?
A: Still Open. But we have a candidate!

Q: How big is it $\Phi$?
A: New result: $|\Phi| \geq 5$
Constructing Sequences

- $T(1,j) : a$ binary tree with height $j+1$
  - $j$ distinct letters at each leaf

Seth Pettie
Constructing Sequences

- $T(1,j)$: a bin. tree with height $j+1$, $j$ letters at each leaf
- $i^{th}$ letter at a leaf added to label of $i^{th}$ ancestor
Constructing Sequences

- $T(1,j)$: a bin. tree w/height $j+1$, $j$ letters at each leaf
- $i^{th}$ letter at a leaf added to label of $i^{th}$ ancestor
Constructing Sequences

- $T(1,j)$: a bin. tree w/height $j+1$, $j$ letters at each leaf.
- $i^{th}$ letter at a leaf added to label of $i^{th}$ ancestor.
Constructing Sequences

- $T(1,j)$: a bin. tree w/height $j+1$, $j$ letters at each leaf
- $i^{th}$ letter at a leaf added to label of $i^{th}$ ancestor
Constructing Sequences
$T(k,j)$: composition of $j$ $T(k-1, \cdot)$ trees, $j$ distinct letters at each leaf.
the \( i^{th} \) letter at a leaf is assigned to the \( i^{th} \) \((k-1)\)-node ancestor of the leaf.
the $i^{\text{th}}$ letter at a leaf is assigned to the $i^{\text{th}}$ $(k-1)$-node ancestor of the leaf.

...and the $T(k-1, \cdot)$ trees are defined in terms of their leaf labels...

$T(k,j) :$ composition of $j$ $T(k-1, \cdot)$ trees, $j$ distinct letters at each leaf.

Seth Pettie
Constructing Sequences

- \( v_1, v_2, \ldots, v_n \): nodes listed in **postorder**
- \( L(v) \): the label of \( v \) in **reverse order**
- The final sequence: \( \Sigma = L(v_1), L(v_2), \ldots, L(v_n) \)

The sequence for \( T(1,4) \):

```
cba fed da ihg lkj jg kheb
onm rqp pm uts xwv vs wtqn xurolifx ...
```
Seth Pettie

0-node

first 1-node

first (k-2)-node

first (k-1)-node

first k-node
Forbidden subseq: \textit{ababa}

• $\Sigma$ is (ababa)-free:

\begin{itemize}
  \item [a]
  \begin{itemize}
    \item [a]
      \begin{itemize}
        \item [a]
          \begin{itemize}
            \item [b]
          \end{itemize}
        \end{itemize}
      \end{itemize}
  \end{itemize}
\end{itemize}
Forbidden subseq: \textit{ababa}

- $\Sigma$ is $(ababa)$-free:
Forbidden subseq: \textit{ababa}

\[ \Sigma \text{ is } (ababa)-\text{free:} \]

these are in the wrong order!

\begin{center}
\begin{tikzpicture}
\node at (0,0) [circle,fill,inner sep=2pt,label=right:\textcolor{red}{ba}](a) [circle,fill,inner sep=2pt,] at (0,0) [circle,fill,inner sep=2pt,] at (0,0) [circle,fill,inner sep=2pt] {a} [circle,fill,inner sep=2pt] {a} [circle,fill,inner sep=2pt] {b} [circle,fill,inner sep=2pt] {b}
\node at (1,1) [circle,fill,inner sep=2pt,label=right:\textcolor{red}{ba}](a) [circle,fill,inner sep=2pt,] at (0,0) [circle,fill,inner sep=2pt,] at (0,0) [circle,fill,inner sep=2pt] {a} [circle,fill,inner sep=2pt] {a} [circle,fill,inner sep=2pt] {b} [circle,fill,inner sep=2pt] {b}
\end{tikzpicture}
\end{center}

Seth Pettie
Forbidden subseq: \(abcaccbc\)

\(\Sigma\) is \((abcaccbc)\)-free:

necessarily a common ancestor
Forbidden subseq: $abcaccbc$

- $\Sigma$ is $(abcaccbc)$-free:

![Diagram showing nodes and necessary different nodes]
Forbidden subseq: $abcaccbc$

• $\Sigma$ is $(abcaccbc)$-free:

- necessarily different nodes
- first $i$-node
- first $j$-node
Forbidden subseq: \textcolor{blue}{abcaccbc}

“a” does not appear in the final contradiction
(an implied occurrence of \textcolor{red}{bcbbc})

Why is it necessary?

\[ \begin{array}{c}
\text{the “binder”}
\end{array} \]
Forbidden subseq: \textit{abcdeaebdce}

\textbullet{} $\Sigma$ is (abcdeaebdce)-free:

\begin{itemize}
\item necessarily different nodes
\end{itemize}

\begin{align*}
  \Sigma & \text{ is (abcdeaebdce)-free:} \\
  e & \rightarrow c \rightarrow d \rightarrow eb \rightarrow a \\
  \text{necessarily different nodes} & \\
  \text{first } i\text{-node} & 
\end{align*}
Forbidden subseq: \textit{abcdeaebdce}

\[ \Sigma \text{ is } (abcdeaebdce)-\text{free:} \]

\[ \text{necessarily different nodes} \]

first \( i \)-node

Seth Pettie
Forbidden subseq: \textit{abcdeaebdce}\

\[ \text{\( \Sigma \) is (abcdeaebdce)-free:} \]

- Necessarily different nodes:
  - "guard"
  - "binder"
  - "trapped elements"

```
  e
  |
  c
  |
  d
  |
  edeb
  |
  a
```

Seth Pettie
Forbidden subseq: abcdeaebdce

Succinct Encoding: ♥ ♦ ♠♠♣

- ♥ : a = binder
- ♦ : b = guard
- ♠ : c = 1st trapped
- ♠ : d = 2nd trapped
- ♣ : e = trapper

"trapper"

"trapped elements"

necessarily different nodes

"guard"

"binder"
All of these encodings make sense & work:

♥♦♠♣♠
♥♦♥♠♠♣
♥♠♠♦♣
♥♣♣♦♠♣

These don’t:
♦♠♣♣♥♣ ← the binder doesn’t bind (but this can be fixed!)
♥♠♣♣♦♣ ← the guard doesn’t guard
♥♣♥♣♠♣ ← this doesn’t make any sense
Forbidden subseq: \textit{abcdeafefbdcf}

\textbf{Encoding:} \begin{itemize}
  \item ★ : \texttt{a} = half-binder
  \item ♦ : \texttt{b} = guard
  \item ♠ : \texttt{c} = 1st trapped
  \item ♠ : \texttt{d} = 2nd trapped
  \item ♥ : \texttt{e} = binder
  \item ♣ : \texttt{f} = trapper
\end{itemize}

Seth Pettie
Forbidden subseq: \textit{abcdeafegfhgihijjbdcj}

Half-binders can be “daisy-chained”
Seventeen legal encodings

♥♠♣(♦♠♣♣)
♥♠♣♠♣♥♠♣
♥♠♣♣remennt
♥♠♣♠♣♥♠♣
♥♠♣♠♣♥♠♣
♥♠♣♠♣♥♠♣
Some open problems

- Are there infinitely many “causes” of non-linearity?
- Are there any more linear seqs. to be discovered?
- For each $c$, is there an (ababa)-free $\sigma$ such that:

$$\text{Ex}(\sigma, n) = n \exp(\alpha^c(n))$$