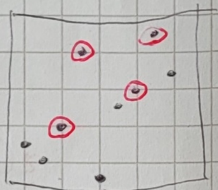
① A New Upper Bound for $g_r(Av(1324))$

Joint work with Chat GPT 5.6 sol

$\boxed{\S 1}$

- Permutations: in one-line notation
- π contains σ if $\exists$ subsequence of π with same relative ordering as σ.

Ex: $\pi = 314592687$



→ contains 1324

→ avoids 4321

- Object of study: $Av(1324) = \left\{ \begin{array}{l} \text{all perms} \\ \text{length } n \\ \text{avoiding } 1324 \end{array} \right\}$

Zeilberger: "Not even God knows $Av_{1000}(1324)$"

- By Marcus–Tardos (2004) and Arratia (1999) the growth rate:
$$c = \lim_{n\to\infty} \sqrt[n]{|Av_n(1324)|} \quad \text{exists}$$
- But what is it?!
- Bevan, B, Elvey Price, Pantone (2020):
$$10.24 \leq c \leq 13.5$$
- ~~Katzhang~~ Conway, Guttmann, Zinn-Justin (2018): Count up to $n=50$ and estimate $c \approx 11.600 \pm 0.003$

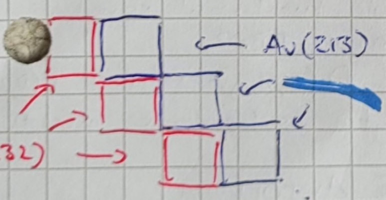
Why do we care?

Only length-4 avoidance class not counted exactly. The others were done by 1997.

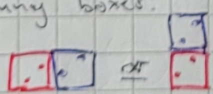
② Today: Thm: (B): $c \leq 13.16$.

$\boxed{\S 2}$ STAIRCASES AND DOMINOES

As noted by many & formalised in BBEP: if $\pi \in Av(1324)$, it looks like:



Also, the "avoid 1324" condition only happens in neighbouring boxes:

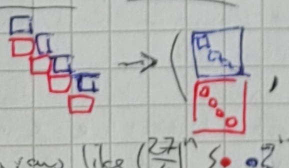


- THM (BBEP): The dominoes $\{\text{domino} : \text{avoiding } 1324\}$ are counted by
$$\frac{2(3n+3)!}{(n+2)!\,(2n+3)!}$$
[OEIS A000139. Counts loads of things]

Asymptotically, ~~[illegible]~~ $\sqrt[n]{|D_n|} \to \frac{27}{4}$

For the 13.5 upper bound:

Length n 1324 avoider



$|D_n|$ grows like $\left(\frac{27}{4}\right)^n$, $\{\bullet, \bullet\}^n$ like 2^n $\Rightarrow c \leq \frac{27}{4} \cdot 2$

$\boxed{\S 3}$ HOW TO DO BETTER ③

An earlier upper bound by Bóna (2015) used a 4-letter alphabet (rather than the 2-letter $\{\bullet, \bullet\}$).

So we map to:

$$\left(\text{domino}, \{\mathring{B}, B, R, \mathring{R}\}^n \right)$$

Now: we need to study shuffles of words over $\{\mathring{B}, B\}$ with $\{R, \mathring{R}\}$

NOTATION: $\delta \in D_n$ a domino length n.

$u(\delta) \in \{\mathring{B}, B\}^*$ ~~[illegible]~~ bottom-to-top encoding of upper cell.

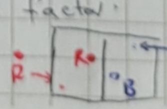
$l(\delta) \in \{R, \mathring{R}\}^*$ " of lower cell.

LEMMA (Bóna, 2015)

A valid 1324-avoiding shuffle contains no $\mathring{B}\mathring{R}$ factor.

$\mathring{B}$ and $\mathring{R}$ must exist.

PROOF:



④ Let $S(\delta) = \{\mathring{B}\mathring{R}\text{-avoiding shuffles of } u(\delta) \text{ and } l(\delta)\}$

$s(\delta) = |S(\delta)|$.

Then
$$|Av_n(1324)| \leq \sum_{\delta \in D_n} s(\delta)$$

Now: we "just" need to estimate $s(\delta)$...

$\boxed{\S 4}$ WEIGHTED DOMINOES ~~[illegible]~~

Let $i(\delta) = \#$ internal (B or R) points in δ

$r(\delta) = \#$ B runs $+ \#$ R runs

Then
$$D(z; x, y) = \sum_{\substack{\text{dominoes} \\ \delta}} z^{|\delta|} x^{i(\delta)} y^{r(\delta)}$$

Let $d_n(x,y) = [z^n]\, D(z; x, y) = \sum_{\delta \in D_n} x^i y^r$.

AIM: Find x, y, λ, C such that
$$s(u(\delta), l(\delta)) \leq C \lambda^{|\delta|} x^i y^r \quad (*)$$

Then
$$|Av_n(1324)| \leq \sum_{\delta \in D_n} s(\delta) \leq C \lambda^n \sum_{\delta \in D_n} x^i y^r = C \lambda^n d_n(x,y)$$

So: $c \leq \lambda \limsup_{n\to\infty} d_n(x,y)$

So we need to: (5)

1) ~~Estab~~ Find $\limsup_{n\to\infty} \sqrt[n]{d_n(x,y)}$ for any x, y

2) Establish $(*)$.

For 1) — build functional equations and solve as in BBCP

$\Rightarrow D(z;x,y)$ satisfies $P(D,z,x,y)=0$

Analytic combinatorics does the rest.

§5 SHUFFLE AUTOMATA

– Machines with 2 input tapes & 1 output tape.

e.g.: $B \circlearrowleft (\mathring{B}) \xrightarrow{\mathring{B}\ \mathring{R}} (N) \circlearrowleft \mathring{B}\ R\ \mathring{R}$, $(N) \xrightarrow{B} (\mathring{B})$

Transitions denote letter being written [choice between blue and red]

N.B. No R transition allowed from $(\mathring{B})$ – enforces BR avoidance.

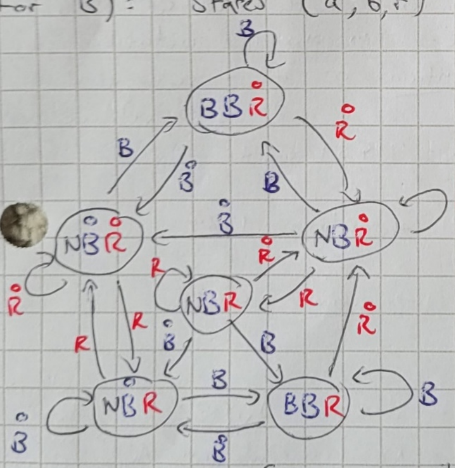
So: input $u(\delta)$, $\ell(\delta)$, and runs generate $S(u,\ell)$.

Our automata need to track $i(\delta)$ & $r(\delta)$ so we do two things:

A) Weight the transitions to track runs and internal letters.

B) Increase no. of states to remember what blue & red letters were last written.

For B): States $(a;b,r)$ (6)



~~NB~~ For A):

$$\tau(e) = \begin{cases} 1 & e \text{ writes } \mathring{B}\ \mathring{R} \\ 1/x & e \text{ reads } \begin{cases} \mathring{B} \text{ and } b=B \\ R \text{ and } r=R \end{cases} \\ 1/xy & \text{otherwise.} \end{cases}$$

So weight of a run is $x^{-i(\delta)} y^{-r(\delta)}$

Central idea: $\Rightarrow$ all possible runs is $s(u,\ell)x^{-i}y^{-r}$

Weighting K on states.

Say that K is λ-good if

$$\sum_{\substack{\text{edges } e \\ \text{available when} \\ \text{running automaton} \\ \text{from } q \to q_e}} \tau(e)K(q_e) \leq \lambda K(q) \quad (**)$$

PROPOSITION: If K is λ-good, ~~and~~ then the total weight of all possible runs on the input $u(\delta)$, $\ell(\delta)$ is $\leq C\lambda^n$ (7)

where $C = \dfrac{K_+}{K_-}$ ($K_+ \leftarrow$ max K, $K_- \leftarrow$ min K)

~~Part~~ So: $\sum s(u(\delta), \ell(\delta))x^{-i}y^{-r} \leq C\lambda^n$

hence $s(u,\ell) \leq C\lambda^n x^i y^r$ as required by $(*)$.

§6 k-LOOKAHEAD AUTOMATA

Basic idea: states "read ahead" k letters on input tapes.

For 1-lookahead: 24 $(=6\times4)$ states.

λ-good weighting requires 96 (24×4) inequalities $(**)$.

FORM IS NUMERICALLY OPTIMAL, IT SEEMS

Take x zero of

$$z^7 - z^6 - 6z^5 + 6z^4 + 12z^3 + 12z^2 - 8z - 8$$

$\Rightarrow x \approx 0.8040$

$y = \dfrac{2x}{x+1} \approx 0.8914$

$\lambda = \dfrac{x+1}{x} = 1 + 1/x \approx 2.2438$

Then $\sqrt[n]{d_n(x,y)} \longrightarrow 5.9679$ (8)

And hence

$c \leq \lambda \cdot 5.9679 \approx 13.39$.

To get the 13.16 advertised: use 3-lookahead:

384 states

1536 inequalities for λ-good.

§7 CONCLUDING REMARKS

- Bigger lookahead: better bounds, but harder to solve.
  e.g. 5-lookahead $\approx$ 13.15 ish.

- More than BR avoidance: use more letters.

ChatGPT says 3-lookahead gives $c \leq 13.14$, but automaton has 13 122 states and 118098 inequalities.

- The bound $s(u,\ell)x^{-i}y^{-r} \leq C\lambda^n$ seems to be reasonably tight.