Combinatorial Lane Merging Problems

Experimental Math Seminar 10/15

Aurora Hiveley
Joint work with Doron Zeilberger

Department of Mathematics
Rutgers University

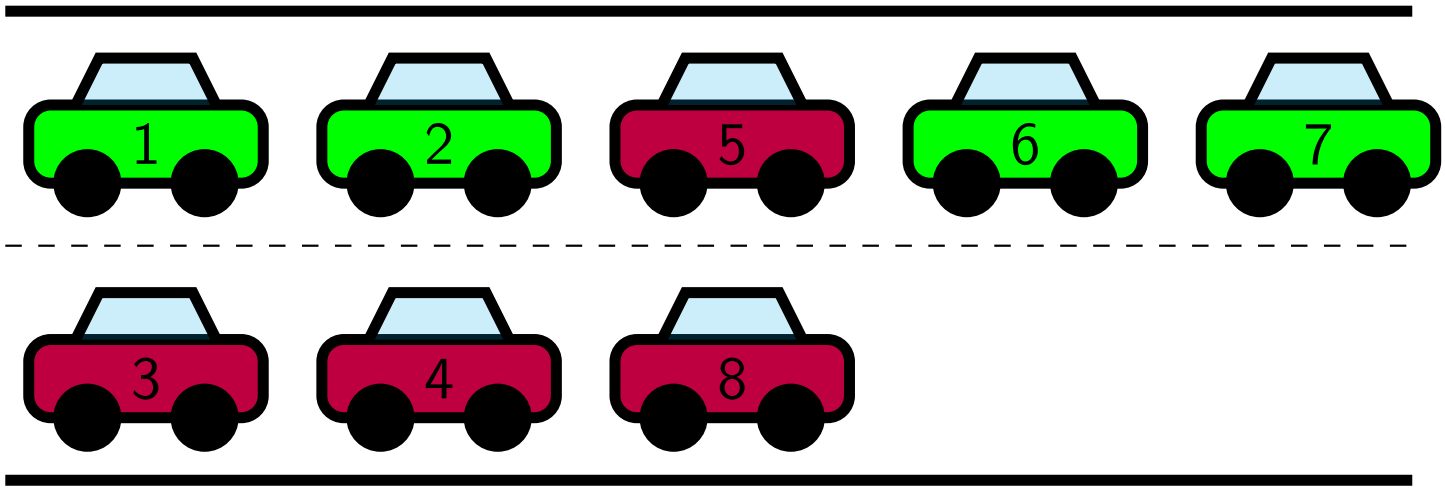
The Problem

A two lane road approaches a traffic light. Shortly after the intersection, the left lane merges into the right lane.

Car preferences: right (1) or shortest lane (2). (*Tie?* $\Longrightarrow$ *Right lane*)

The list of car preferences form an *arrival sequence*, denoted $A \in \{1, 2\}^n$.

Example: $A = 11222112$.



The Problem (cont.)

Motivating Question: How many arrival sequences in $\{1,2\}^n$ end with m cars in the right lane and l in the left? *Notationally:* $F(m,l) = ???$

Observation: The left lane is *always* at most as long as the right lane.
★ Visualize with a Dyck path!

Def. A **Dyck path** is an up-right path which starts at (0,0) and never crosses below the diagonal.

Define a path from an arrival seq.:

- Car in right lane $\mapsto$ Up edge
- Car in left lane $\mapsto$ Right edge

Example: $A = 11222112$

Another: $A' = 11221112$
★ Not a bijection!

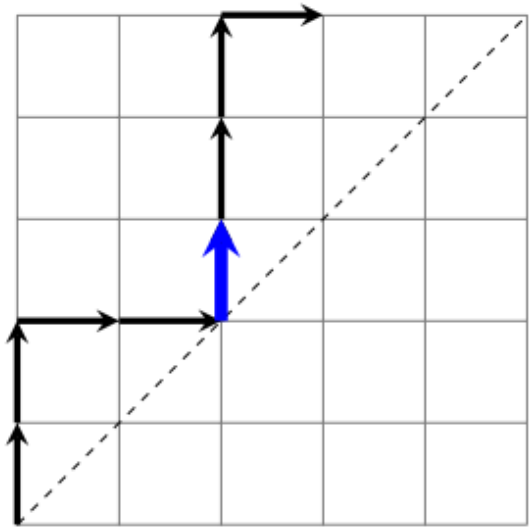


Image credit Bardenova et al. (2023)

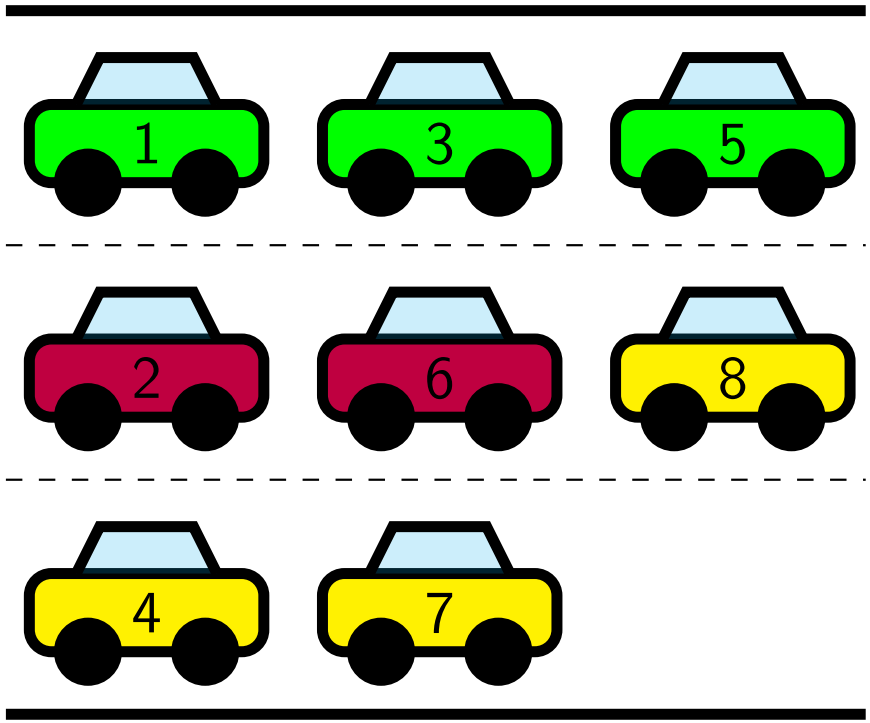
Three-Lane Problem

What if we now have three lanes that all merge into the rightmost lane?

Car prefs. for $A \in \{1, 2, 3\}^n$:

- (1) = Right lane
- (2) = Shortest of the middle or right lane
- (3) = Shortest of all lanes
- $*$ Tiebreak opts for rightmost tied lane

Example: $A = 12131233$



General Problem

Car preferences for $A \in \{1, 2, \ldots, k\}^n$:

- (i) = Shortest of the i rightmost lanes for any $1 \leq i \leq k$
- $*$ Tiebreak still opts for rightmost tied lane

Main Question: $F(x_1, x_2, \ldots, x_k) = ???$
★ x_i = length of i-th lane

To visualize our sequences and construct recurrences . . . Uh-oh.

- 3-dimensional Dyck paths? :/
- k-dimensional Dyck paths? :(

BUT we do still know that lane lengths are weakly decreasing from right to left! So . . .

Standard Young Tableaux

Def. A **SYT** is a left-justified array of n numbers such that:

- Each number in $\{1, 2, \ldots, n\}$ appears in exactly one position
- Row lengths are weakly decreasing from top to bottom
- Entries are strictly increasing from $L \to R$ and $T \to B$

Def. The **shape** of T is a list $[t_1, t_2, \ldots, t_k]$ of the row lengths from top to bottom.

Define a SYT T from an arrival seq. A:

- Car j in i-th lane $\implies$ entry j in i-th row of T

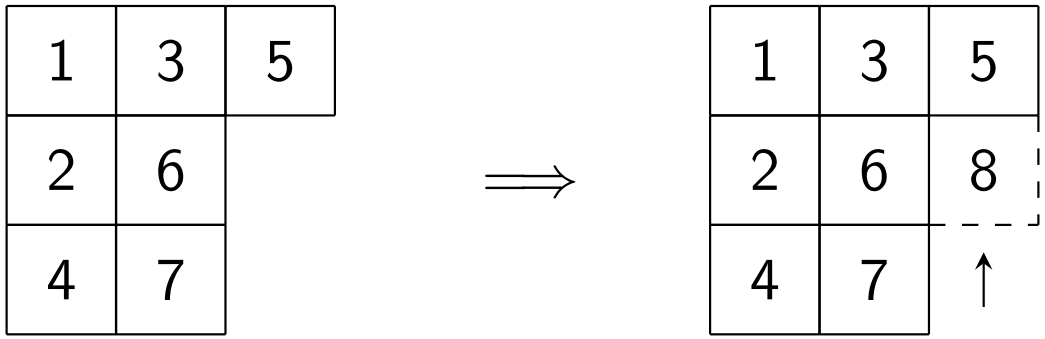
Example: $A = 1312223133$

1	3	5	8
2	4	6	
7	9	10	

Mapping to SYT

Question: How many A map to a given T? *Notationally:* $\#A(T)$.
Recall: Tied lanes $\implies$ multiple prefs. produce same result!

Example: $A = 12131233$



Car 8 could have preference 2 or 3 and would still end up in middle lane!

Def: A **bounce** occurs when a car w/ pref. $i > 1$ tie breaks to the right of lane i.

Bounces

Lemma.

If T is a SYT and T' is the SYT obtained by deleting the largest element, n, from cell c of T, then

$$\#A(T) = \beta(c) \cdot \#A(T')$$

where $\beta(c)$ is the size of bounce, or the number of open cells below c.

Proposition.

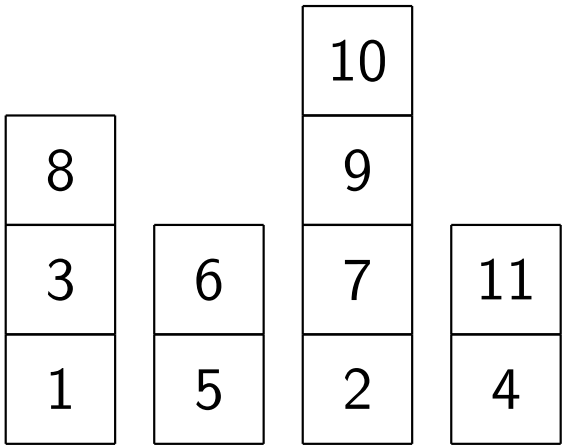
$$\#A(T) = \prod_{c \in T} \beta(c)$$

Note: First column has a special definition of $\beta(c)$. See Hive. & Zeil. for details.

Voting Sequences

In a voting sequence in $\{1, 2, \ldots, k\}^n$, each of n voters votes for one of k candidates. When finished, there are v_1 votes for the 1st place candidate, v_2 votes for the 2nd place candidate, etc.

Example: $V = 13142231334$



$v_1 = 4$ Candidate #3
$v_2 = 3$ Candidate #1
$v_3 = 2$ Candidate #2 (Tied)
$v_4 = 2$ Candidate #4 (Tied)

Final vote counts are weakly decreasing by popularity.
This feels familiar . . .

Voting Sequences (cont.)

We can also construct SYT for voting sequences!
Example: $V = 111223333$

1	2	3	9
4	5	8	
6	7		

Lemma.

$$\#V(T) = \beta(c) \cdot \#V(T')$$

Here, $\beta(c)$ is the number of candidates with tied popularity that Voter n could vote for and produce the same final vote count.

Closed Form

Lemma.

Majority voting sequences are counted by the closed form

$$F(v_1, \ldots, v_k) = \binom{n}{v_1, \ldots, v_k} \cdot \binom{k}{p_1, \ldots, p_r}$$

where $p_1, p_2, \ldots, p_r$ is the partition of k defined by $(v_1, \ldots, v_k)$ in frequency notation.

Example of frequency notation for a partition:

$$[1, 1, 2, 2, 2, 2, 3] \to 1^2 2^4 3^1 \to (2, 4, 1)$$

Proof outline:

- n votes cast. Choose v_1 for Candidate 1, v_2 for Candidate 2, etc.
- k candidates total. Choose p_1 to be (tied) for 1st place, p_2 to be (tied) for 2nd place, etc.

Conclusion

Corollary.

Since both arrival sequences and voting sequences satisfy the same recurrence,

$$F(x_1, \ldots, x_k) = \binom{n}{x_1, \ldots, x_k} \cdot \binom{k}{p_1, \ldots, p_r}$$

Other comments:

- Agrees with Bardenova et al. closed form!
- Mapping $A \to V$ can be constructed using our recurrence
- Tables for summed lane lengths can be found in Hive. & Zeil.
- ★ For right lane, these sequences are in the OEIS!
  $\hookrightarrow$ A230137, A265083, A265084, A265085, etc.

Thanks & Questions

Selected Reading:

- V. Bardenova, E. Insko, K. Johnson, and S. Sullivan, A combinatorial model for lane merging, *Journal of Integer Sequences* **26** (2023) Article #23.5.3
- E. Insko, K. Johnson, and S. Sullivan, A combinatorial model for lane merging: Limited capacity lanes, *ECA* **6:2** (2026) Article #S2R9.
- A. Hiveley and D. Zeilberger, The Combinatorics of Multi-Lane Merging, arXiv, arxiv.org/pdf/2607.26334

Thanks! Questions?

Bonus: Mapping is Hard

Recursion $\implies$ number of SYT with a given *shape* is the same. The seq. that produce them are different:

Example: $A = 211 \implies [3, 0]$, but $V = 211 \implies [2, 1]$

Different end states! Even within end states, there are variations over time:

$$A = 221 \implies [0, 0] \to [1, 0] \to [1, 1] \to [2, 1]$$
$$A = 212 \implies [0, 0] \to [1, 0] \to [2, 0] \to [2, 1]$$

Idea: Pick a sequence that resolves in the same way, so the SYT are the same at *every* step

Problem: You must keep track of when tie breaks happen. For lane merging, this isn't too bad. For voting sequences, you must track each permutation of the candidates' standings as the entire sequence unfolds.

Conclusion: Using the recurrence is best :)