Structure in solution spaces: Three lessons from Jean-Claude

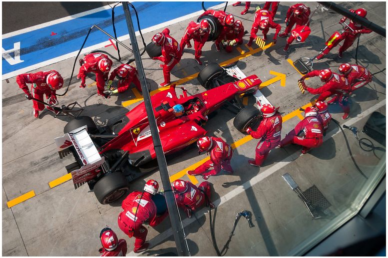
David Eppstein
Computer Science Department, Univ. of California, Irvine

Conference on Meaningfulness and Learning Spaces:
A Tribute to the Work of Jean-Claude Falmagne
February 27–28, 2014

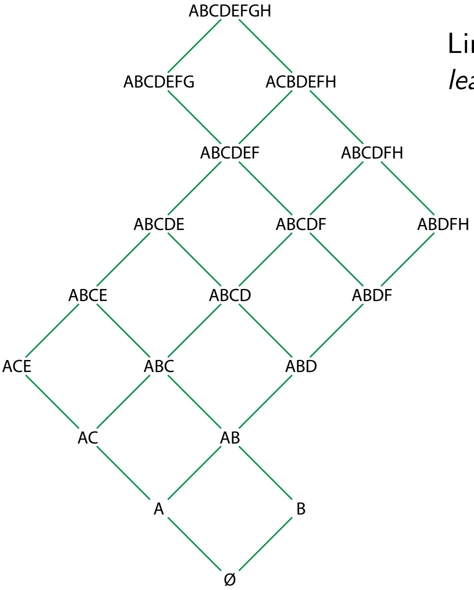
My specialty: Algorithm design

The algorithm design process:

- Find a computational task in need of solution
- Abstract away unimportant details
- Often, a *naive algorithm* exists but is too slow
- Design algorithms that are faster (scale better with problem size) without sacrificing solution quality



ALEKS circa 2000–2005



Limited to *quasi-ordinal learning spaces*:

- What a student knows is represented as a finite set, the set of concepts the student has mastered
- *Learning space*: the family of sets that could possibly be the state of knowledge of some student
- *Quasi-ordinal*: the intersection or union of any two sets in the family is another set in the family

What's wrong with quasi-ordinal spaces?

Closure under unions makes sense psychologically, but closure under intersections does not

This causes the spaces to have more sets than they should (intersections that can't really happen)

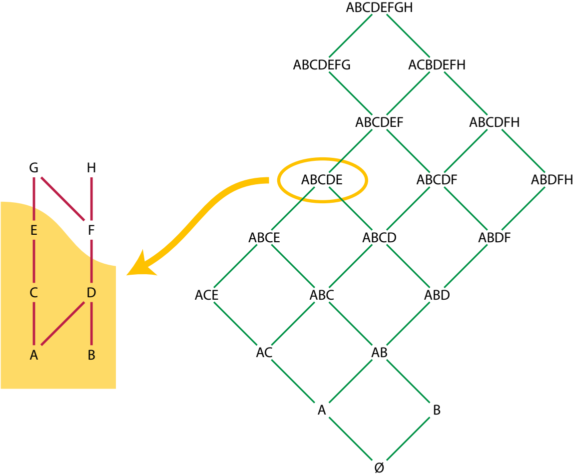
The extra sets increase the number of test questions needed to assess a student, slow down the assessment calculations, and lead to inaccuracies in the assessments

Because of these problems,
JCF was desperate to eliminate this restriction.

If quasi-ordinal spaces are bad, why use them?

Mathematically, quasi-ordinal spaces form *distributive lattices*

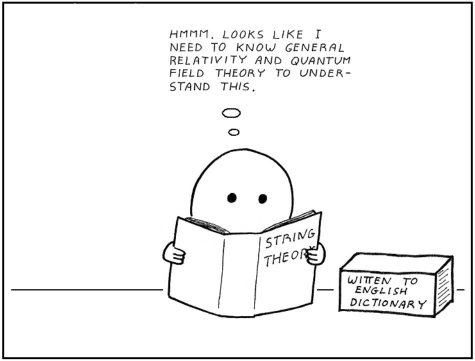
Birkhoff's representation theorem: the sets in these spaces can be represented as downward-closed subsets of a partial order



But what does it mean?

A partial order on a set of concepts to be learned describes a *prerequisite* relation

A student will only become ready to master a concept after he or she has mastered all its prerequisites



STEP 2: enter the rabbit hole

Advantages of using the underlying partial order

It's concise

- Only the prerequisite relation needs to be communicated to client software

It's fast

- Key computational bottleneck:
  listing all states in the learning space
- Time per state $\approx$ #concepts / machine word size
  [as implemented in early versions of ALEKS]
- Can theoretically be improved to O(log #concepts) per state
  [Squire 1995]

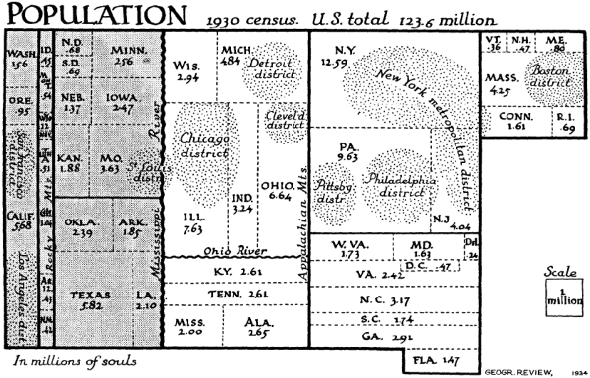
Lesson I

When your state space forms a distributive lattice,
find out what the underlying partial order means,
and take advantage of it for fast and space-efficient algorithms

Application: Rectangular cartograms

Stylize geographic regions as rectangles

Rectangle areas represent numerical data



[Raisz, "The rectangular statistical cartogram", *Geog. Rev.* 1934]

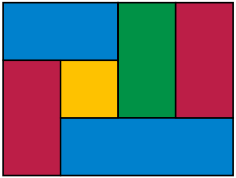
Stylization emphasizes the fact that it's not a map

The simplicity of the shapes makes areas easy to compare

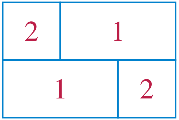
Formalization of cartogram construction

Find a partition of a rectangle into smaller rectangles, satisfying:

- Adjacency: Geographically adjacent regions should stay adjacent
- Orientation: Avoid gross geographic misplacement (e.g. California should not be north of Oregon)
- Area universality: Can adjust to any desired set of areas while preserving adjacency

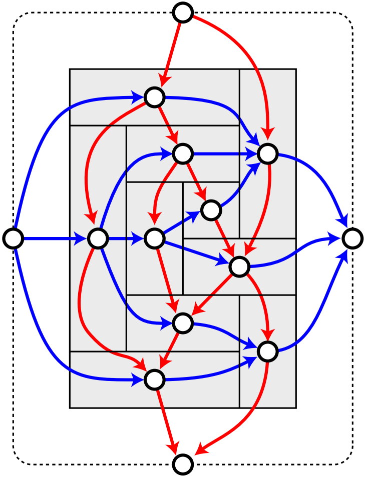


Area-universal



Not area-universal

Combinatorial language for describing layouts



Augment adjacency graph with four extra vertices, one per side of outer rectangle

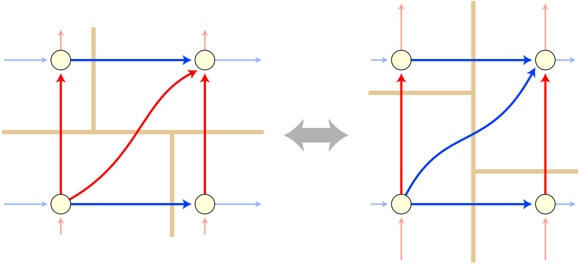
Color side-by-side adjacencies blue, orient left to right

Color above adjacencies red, orient top to bottom

Layouts for a given set of adjacencies correspond 1-for-1 with labelings in which the four colors and orientations have the correct clockwise order at all vertices

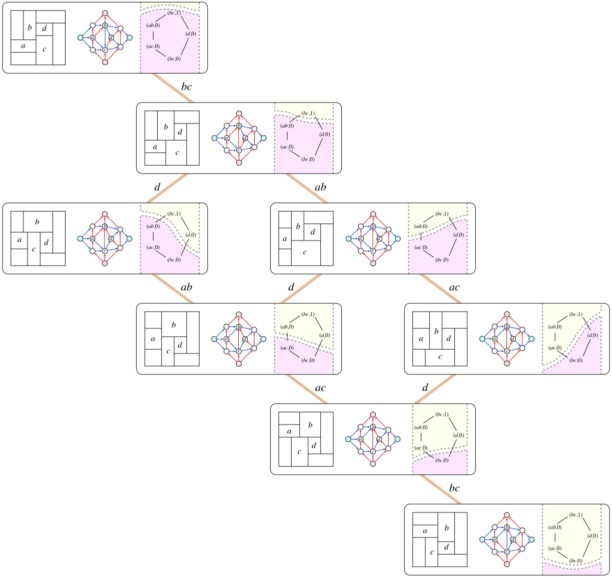
Local changes from one layout to another

Change colors/orientations within a quadrilateral

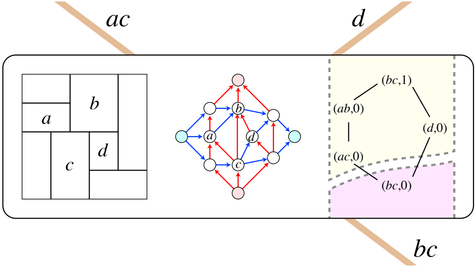


Corresponds to twisting either the boundary between two regions (as shown) or a rectangle surrounded by four others

The distributive lattice of layouts and labelings



What does it mean?



Elements = layouts

Neighbors = layouts that differ by a single twist

Upward in lattice order = twist counterclockwise

Elements of partial order = number of times each boundary has been twisted

Area-universal if and only if no edge twists are possible

Results of applying Lesson I

Although there may be exponentially many layouts,
the underlying partial order has polynomial size
and can be constructed in polynomial time

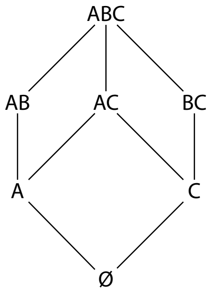
By working at the partial order level of abstraction, can efficiently find an area-universal layout (if it exists) with arbitrary constraints on boundary orientations

E., Mumford, Speckmann, & Verbeek, "Area-universal and constrained rectangular layouts", SIAM J. Comput. 2012

Beyond quasi-ordinal spaces

My contribution to ALEKS:

- Instead of prerequisites, describe a learning space by its *learning sequences*: orderings in which all concepts in the space could be learned
- States = unions of prefixes of the learning sequences



This example has four learning sequences:
A–B–C, A–C–B, C–A–B, and C–B–A

Only two, A–B–C and C–B–A, suffice to define the whole space

Advantages of the learning sequence formulation

- Capable of representing every learning space that is *accessible* (can be learned one concept at a time) and closed under unions

  Mathematically, such a space forms an *antimatroid*

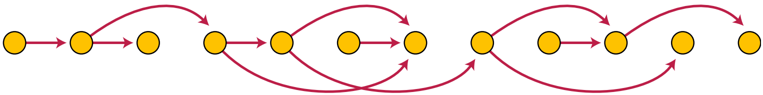
- As with quasi-ordinal spaces, can list all states quickly (the key step in student assessment)

- Still quite concise

- Can construct a description using the smallest possible set of learning sequences in time polynomial in the number of states

Relation to quasi-ordinal spaces and partial orders

A learning sequence of a quasi-ordinal space is a *linear extension* of its underlying partial order, or equivalently a *topological ordering* of its prerequisite relation. (A sequence of the vertices of a directed acyclic graph such that each edge is oriented from earlier to later in the sequence.)



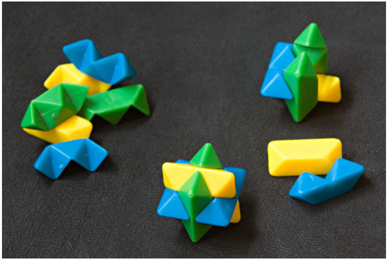
Thus, learning sequences provide a natural method of generalizing linear extensions and topological orderings to more general spaces

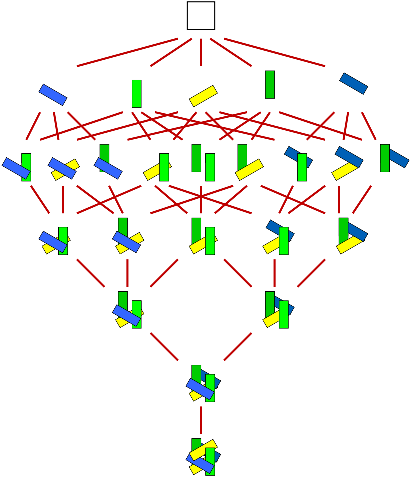
Lesson II

Antimatroids are a good way of describing sets of orderings.

When a problem involves linear extensions of partial orders
or topological orders of directed acyclic graphs,
generalize to antimatroids and learning sequences.

Example: Burr puzzle disassembly sequences





Application of Lesson II: The 1/3–2/3 conjecture

Conjecture: every partial order that is not a total order has two elements x and y such that the number of linear extensions with x earlier than y is between $1/3$ and $2/3$ of the total number

Formulated independently by Kislitsyn (1968), Fredman (circa 1976), and Linial (1984)

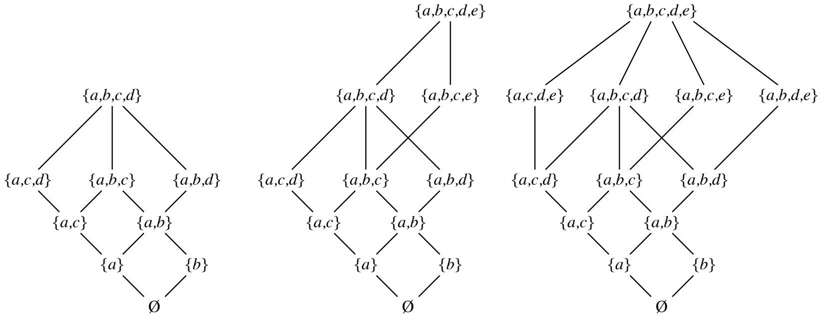
Equivalently, in comparison sorting, it is always possible to reduce the number of potential output sorted orderings by a $2/3$ factor, by making a single well-chosen comparison

(As a consequence, every partial order can be sorted in a number of comparisons logarithmic in its number of linear extensions.)

The 1/3–2/3 conjecture for antimatroids

Conjecture: every antimatroid that is not a total order has two elements x and y such that the number of learning sequences with x earlier than y is between $1/3$ and $2/3$ of the total number

[E., "Antimatroids and balanced pairs", *Order* 2014]

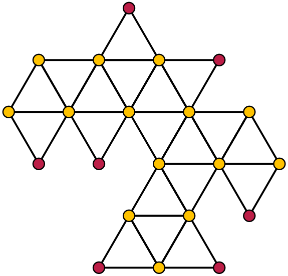


Three antimatroids for which the conjecture is tight

Partial results on the conjecture

True for:

- Antimatroids defined by two learning sequences (generalizing width-two partial orders)
- Antimatroids of height two (generalizing height-two partial orders)
- Antimatroids with at most six elements (by computer search)
- Several classes of antimatroids defined from graph searching



Example: Elimination orderings of maximal planar graphs

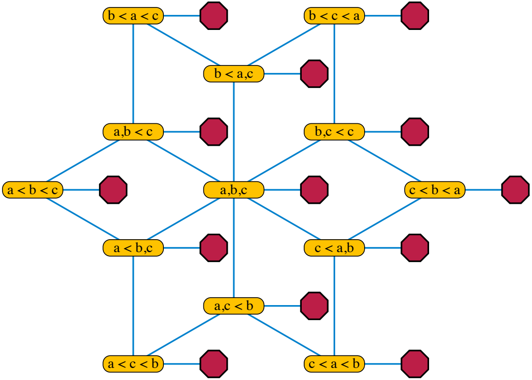
The red vertices have ≤ 2 neighbors and can safely be removed.

The yellow vertices have to wait until some neighbors have been removed.

Beyond learning spaces

Media: systems of states and transitions that can be embedded in a distance-preserving way into a Hamming cube $\{0,1\}^n$

Every learning space is a medium, but not conversely



Medium of voter preferences with "frozen" states $\Rightarrow \{0,1\}^{19}$
From Falmagne, Regenwetter, and Grofman, 1997

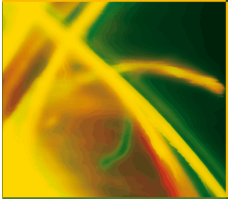
Fast shortest paths in media

The medium structure makes finding shortest paths between all pairs of states easier than in arbitrary state-transition systems

E. & Falmagne, *Disc. Appl. Math. 2008*

By combining a bit-parallel breadth-first-search based labeling phase with the fast shortest path algorithm, can recognize whether a state-transition system forms a medium in quadratic time

E., SODA 2008 & *J. Graph. Alg. Appl.* 2011



David Eppstein
Jean-Claude Falmagne
Sergei Ovchinnikov

Media Theory

Interdisciplinary Applied Mathematics

Springer

Lesson III

Large numbers of different state-transition systems have the structure of a medium

When they do, the underlying hypercube embedding allows fast construction of shortest paths

Unexplained dichotomy in computational complexity

Three important classes of computational problems

- **P**: problems that can be solved in polynomial time
- **NP**: problems that can be solved in exponential time by a simple brute-force search
- **NP**-hard: at least as difficult as *all problems* in **NP**

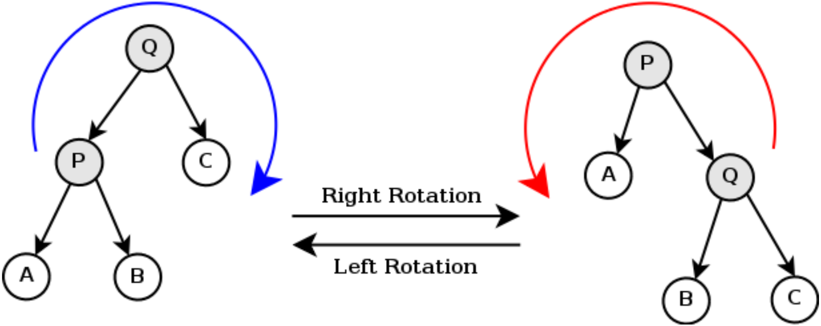
Most computational problems that have been studied are either known to be in **P** or known to be **NP**-hard

One of the rare exceptions: rotation distance in binary trees / flip distance in polygon triangulations

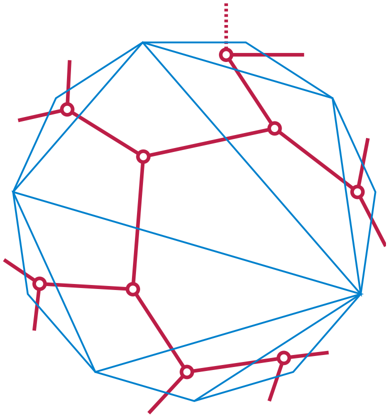
Rotation in binary trees

States: binary trees having a given ordered sequence of keys

Transitions: swap parent-child relation between two nodes
(rearranging their three other children to preserve key sequence)



Binary trees and polygon triangulations



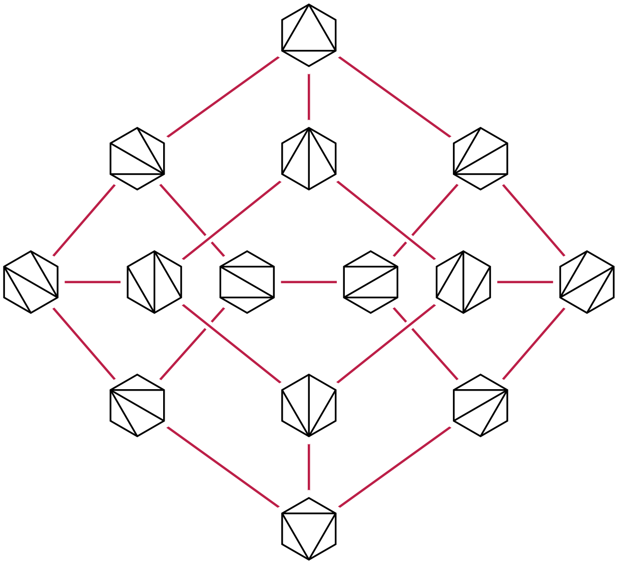
Binary trees with n leaves correspond by planar graph duality to triangulations of an $(n+1)$-sided polygon

The one extra side marks the root of the tree

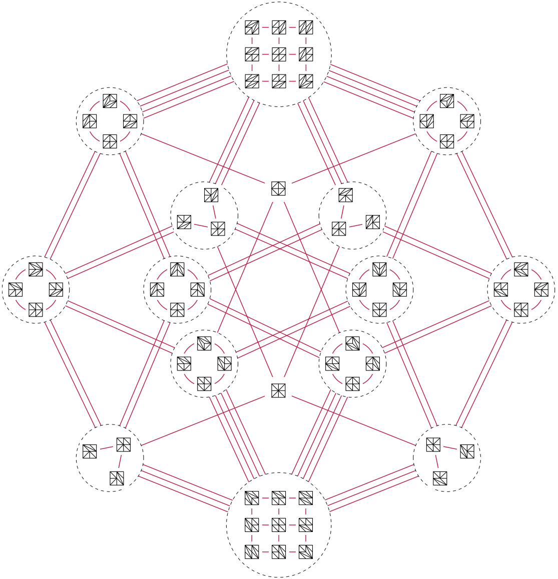
Tree rotations $\Leftrightarrow$ flips in the triangulation

Flip: retriangulate the quadrilateral formed by two adjacent triangles

Flip distance: Distance in the flip graph



Generalize flip distance to non-convex point sets



E.g. in this case the point set is a 3×3 grid

The generalized problem is NP-hard

Lubiw & Pathak, CCCG 2012

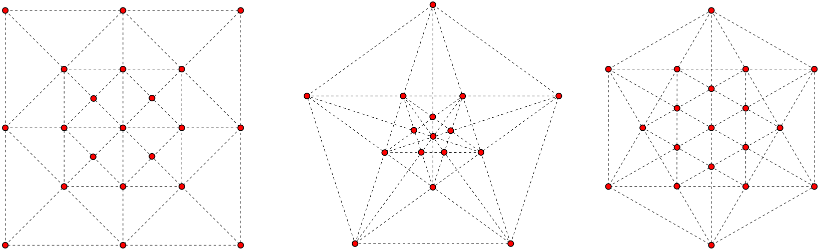
Pilz, *Comp. Geom.* 2014

But maybe some other special cases are easier?

Application of Lesson III to flip distance

Triangulations and flips form a medium if and only if the point set does not include the vertices of an empty convex pentagon

Includes all convex subsets of an integer grid, and some other sets:



When this is true, we can compute flip distance in polynomial time

E., SoCG 2007 & *J. Comp. Geom.* 2010

Conclusions

To compute efficiently with large state spaces, one must understand their mathematical structure

The structures identified by JCF — including quasi-ordinal spaces, learning spaces, and media — appear ubiquitously both in social science applications and beyond

Identifying one of these structures in an application is the first step to finding efficient algorithms for that application

Thank you, Jean-Claude!