Mathematical Principles and Algorithms for Spatial Data

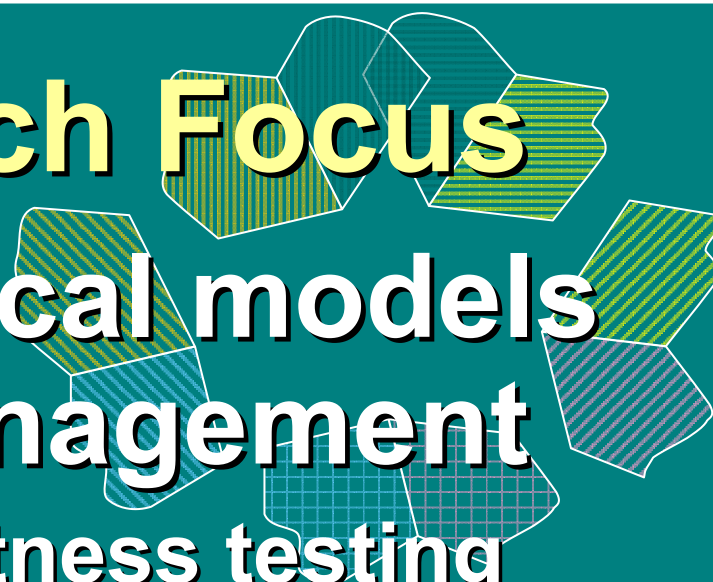
Alan J. Saalfeld

Tamal K. Dey

NMA201-01-1-2012

Contact: saalfeld.1@osu.edu

Primary Research Focus



Abstract mathematical models for spatial data management

- Completeness and correctness testing
  - Topology of a map without geometry
    - Combinatorial properties of a cell set determine when it is possible to build a map from the cells
- Primitive map components & operations
  - Faithfully represent map objects
  - Faithfully represent map operations
  - Employ exhaustive/exclusive cell definitions
  - Meet computational feasibility requirements

Completeness & Consistency

- What (minimal set of) combinatorial information encodes the map's topological structure as a graph on a surface to permit testing for completeness and consistency?
  - What combinatorial tests, when passed, guarantee the encoded map's completeness and consistency and, when failed, guarantee that no map exists with the purported cell structure?
  - Can we prove that no proper subset of the tests suffices? Yes, by finding non-map examples that pass any two, but not all three tests.
- Key math facts:
  - Cells form a surface if they do so locally
  - The topological type of any surface depends on a few easily computed topological invariants

Algorithms for Topology Editing

- Umbrella edit
  - Uses the winged-edge data structure
  - Builds a disk-like neighborhood at each 0-cell
- Euler characteristic edit
  - Counts the 0-cells, 1-cells, and 2-cells
  - Computes the graph's connected components
- Orientability edit
  - Checks that each interior edge v_1v_2 has its vertices v_1 and v_2 enumerated v_1 before v_2 in one adjacent 2-cell's cyclic ordering and v_2 before v_1 in the other adjacent 2-cell's ordering

$\chi = f - e + v - (cc - 1)$

		$\Delta\chi$	Δf	Δe	Δv	Δcc
Add a vertex to a region		0			+1	+1
Add a vertex to an edge, splitting the edge		0		+1	+1	
Add an (acyclic) edge linking components		0		+1		-1
Add an edge creating a cycle, and thus splitting a region		0	+1	+1		
Remove an acyclic edge, disconnecting a component		0		-1		+1
Remove an edge of a cycle, merging two regions		0	-1	-1		
Remove a vertex **u** and all its k edges, (b different regions at **u**)		0	-b+1	-k	-1	k-b *

* k-b = (# of edges in co-cycle at u) – (# of distinct faces in co-cycle at u)

Recent Research Results

A Better Combinatorial Model for maps

- **Define the pieces (primitive components)**
  - Graphs in the plane $\subset$ Graphs on surfaces
    - Straight-line-segment-graphs in the plane
  - Equivalence classes of graphs in the plane
    - "Equivalent" means having the same underlying point set
- **Define how the pieces fit and interact**
  - Fit information is encoded in winged-edge structure
  - Operations on (equivalence classes of) graphs
    - Map editing (feature addition, removal, alteration)
    - Map overlay
    - Map validation

There is a currently developing mathematical theory called “Graphs on Surfaces”

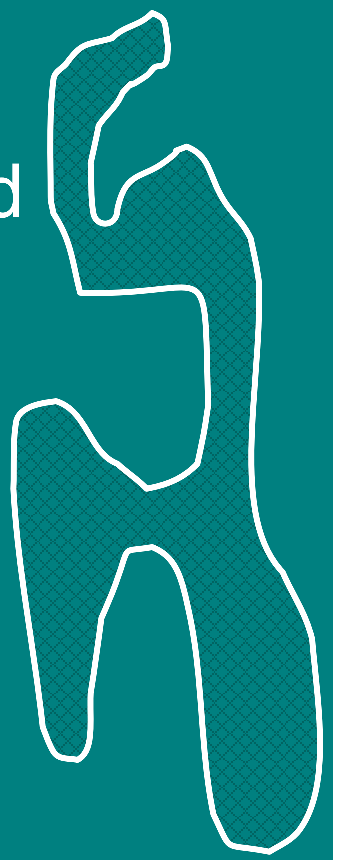
- What graphs can be drawn on surfaces?
  - K_5 and $K_{3,3}$ cannot be drawn on the plane
  - K_7 and $K_{4,4}$ can be drawn on the torus
- How do graph cycles separate or disconnect a surface? *Every cycle disconnects the plane*
- How can we identify and somehow equate all graphs having the same point set?
  - Subgraphs, inessential vertices and graph minors
  - Forbidden graph theory of Robertson & Seymour: *Every surface has a finite set of forbidden graph generators*

Jordan Curve Theorem

- Every simple closed curve on the plane separates the plane into two open connected disjoint regions.
- Each region is (path-)connected.
- One region, the exterior, is unbounded.
- The other region, the interior, is bounded.

Jordan Curve Theorem Corollaries

- Simple closed curves include simple graph cycles. Simple cycles are completely characterized by having no vertices repeated other than the first and last vertices.
- If a connected subset of the graph or of its complement does not contain any point on the simple closed curve, then it is either entirely inside or entirely outside the simple closed curve.
- An acyclic edge is completely characterized by the following property: left and right regions are the same.



Equivalence Relations and Equivalence Classes

- An equivalence relation on a set **S** is a binary relation "$\cong$" that satisfies three properties:
  1. Equivalence is reflexive: $\mathbf{x \cong x}$
  2. Equivalence is symmetric: $\mathbf{x \cong y \Leftrightarrow y \cong x}$
  3. Equivalence is transitive: $\mathbf{[x \cong y \ \&\ y \cong z] \Rightarrow x \cong z}$
- For each element **x**, the set $\mathbf{E_x = \{z \mid z \cong x\}}$ is called the equivalence class of **x**.
- Equivalence classes form a partition of **S** :
  – $\mathbf{S = \cup_{x \in S} E_x}$ [The sets $\mathbf{E_x}$ cover **S**]
  – For any pair of elements **x**, **y** in **S**, either $\mathbf{E_x = E_y}$, or $\mathbf{E_x \cap E_y = \varnothing}$. [There is no partial overlap]

Examples of Combinatorial Equivalence Classes

- Graphs that are minors of some common graph
  - $\mathbf{G_1} \cong \mathbf{G_2} \Leftrightarrow \exists$ a graph **G** such that both $\mathbf{G_1}$ and $\mathbf{G_2}$ are minors of **G**
- Nodes of a graph belonging to the same path-connected component of the graph
  - $\mathbf{a} \cong \mathbf{b} \Leftrightarrow \exists$ a path $(\mathbf{a}, \mathbf{v_{i1}}, \mathbf{v_{i2}}, \ldots, \mathbf{b})$ from **a** to **b**

Examples of Topological Equivalence Classes

- Points of a topological space T belonging to the same connected component of the space
  - **$a \cong b$** $\Leftrightarrow$ $\nexists$ open sets A, B with $a \in A$, $b \in B$, and $T = A \cup B$
- Two topological spaces themselves are equivalent if there exists a homeomorphism (i.e., an invertible continuous function having a continuous inverse) between the two spaces
  - **$X \cong Y \Leftrightarrow \exists\, f : X \rightarrow Y$ and $g : Y \rightarrow X$, such that $g \circ f = 1_X : X \rightarrow X$ and $f \circ g = 1_Y : Y \rightarrow Y$**

Examples of Combinatorial and Topological Equivalence Classes

1. Two Embedded Graphs may be called equivalent if they have the same underlying point set
   – $\mathbf{G_1} \cong \mathbf{G_2} \Leftrightarrow \mathbf{G_1}$ and $\mathbf{G_2}$ consist of the same points of the surface and in both cases the graph topology matches the topology induced by the embedding
2. Or alternatively, two Embedded Graphs may be called equivalent if they are homeomorphic and also have a homeomorphic extension to the entire embedding surface
   – $\mathbf{G_1} \cong \mathbf{G_2} \Leftrightarrow \mathbf{G_1}$ and $\mathbf{G_2}$ are both embedded in surfaces $\mathbf{S_1}$ and $\mathbf{S_2}$, $\mathbf{G_1} \subset \mathbf{S_1}$ and $\mathbf{G_2} \subset \mathbf{S_2}$ and there exists a homeomorphism $\varphi: \mathbf{S_1} \rightarrow \mathbf{S_2}$ such that $\varphi(\mathbf{G_1}) = \mathbf{G_2}$

Key Consequence of the First Equivalence Relation

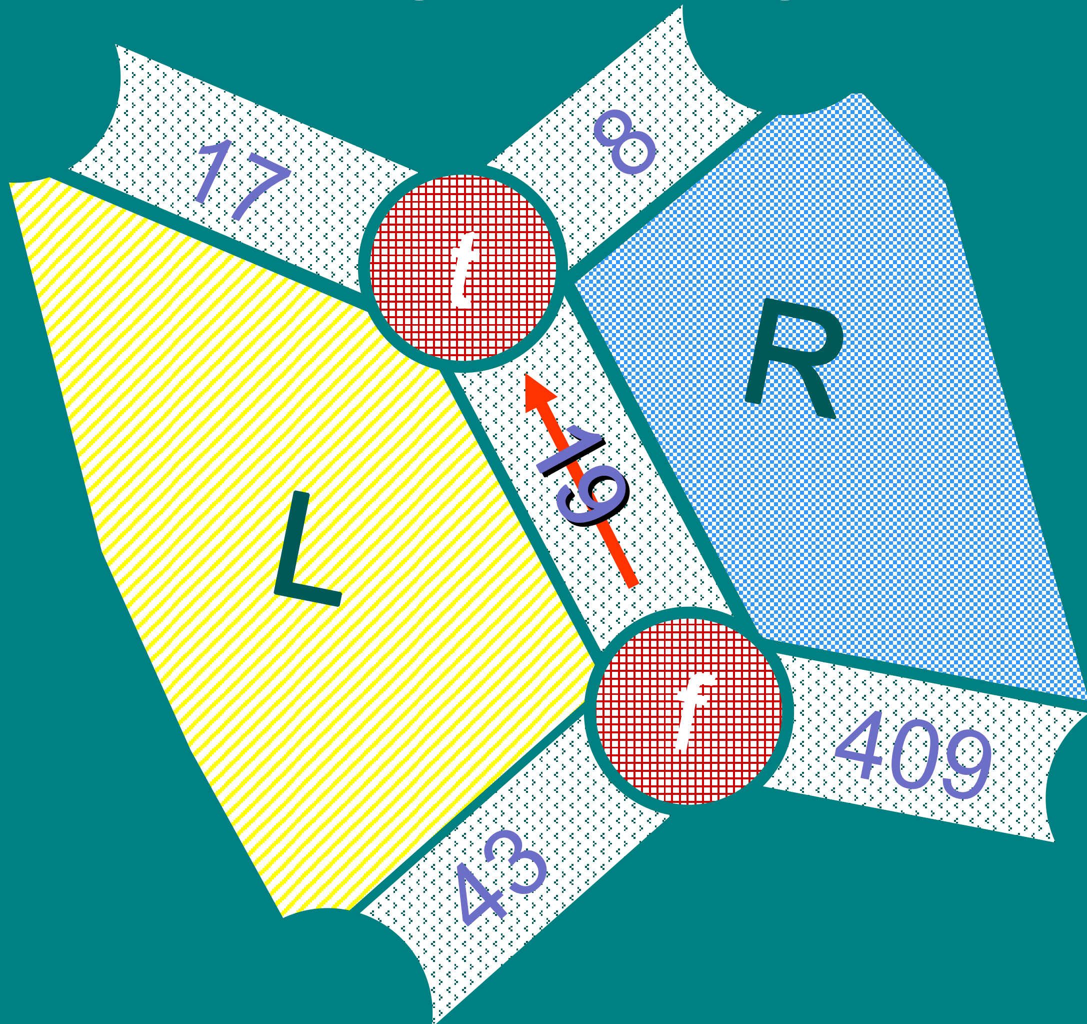
1. Call two Embedded Graphs equivalent if they have the same underlying point set:
   $G_1 \cong G_2 \iff G_1$ *and* G_2 *consist of the same points of the surface and in both cases the graph topology matches the topology induced by the embedding*

Consequence: we may freely add (or remove) vertices of degree 2 to (or from) the vertex set of a drawn graph without changing the underlying drawing or leaving the graph's equivalence class

Equivalence Class Issues

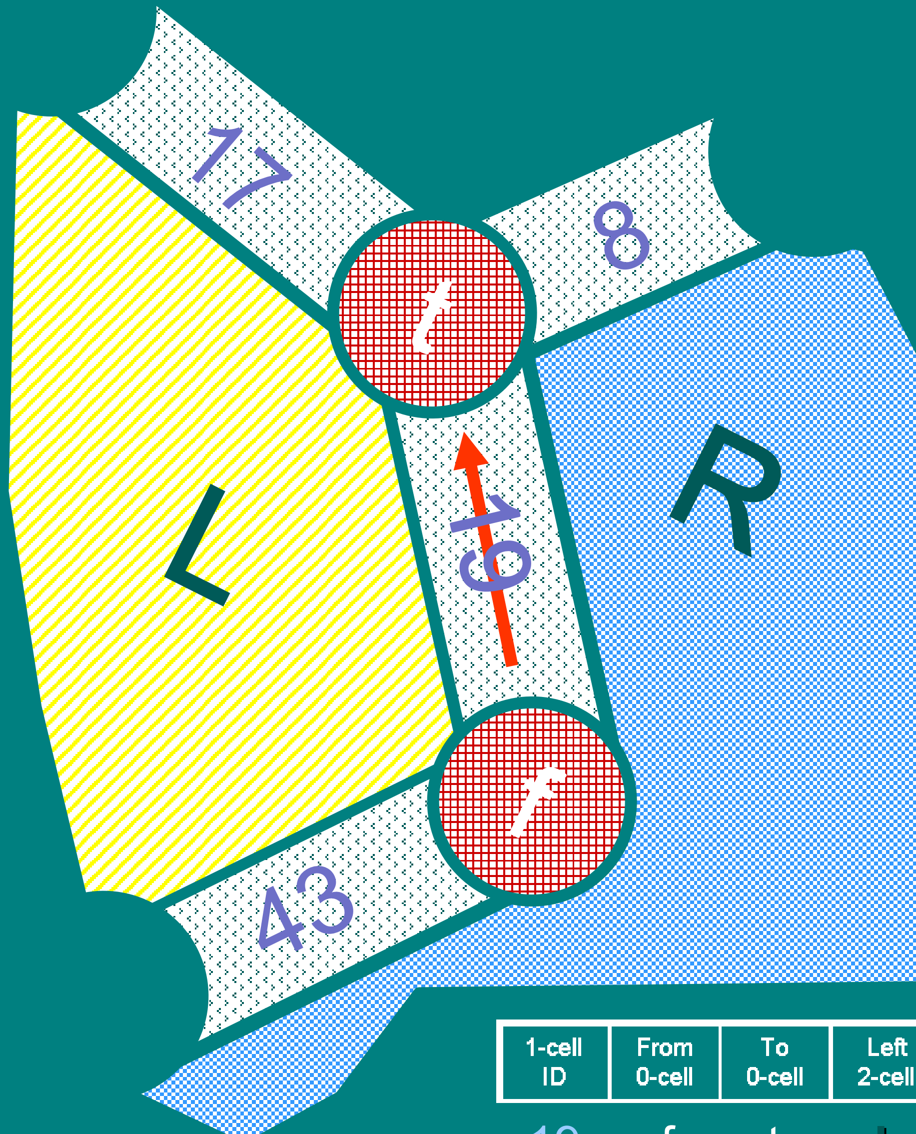
- When are two instances essentially the same?
  - **Consider that a single map instance can be viewed as two different embedded graphs**
  - **There must be an algorithm for testing equivalence of two different graph embeddings**
    - **If “equivalence” means “same point set”, then geometry can be used to verify equivalence.**
    - **If “equivalence” means “graphs are homeomorphic under a homeomorphism of the entire surface”, then an explicit homeomorphism (such as a map projection) suffices**

Winged-edge data record



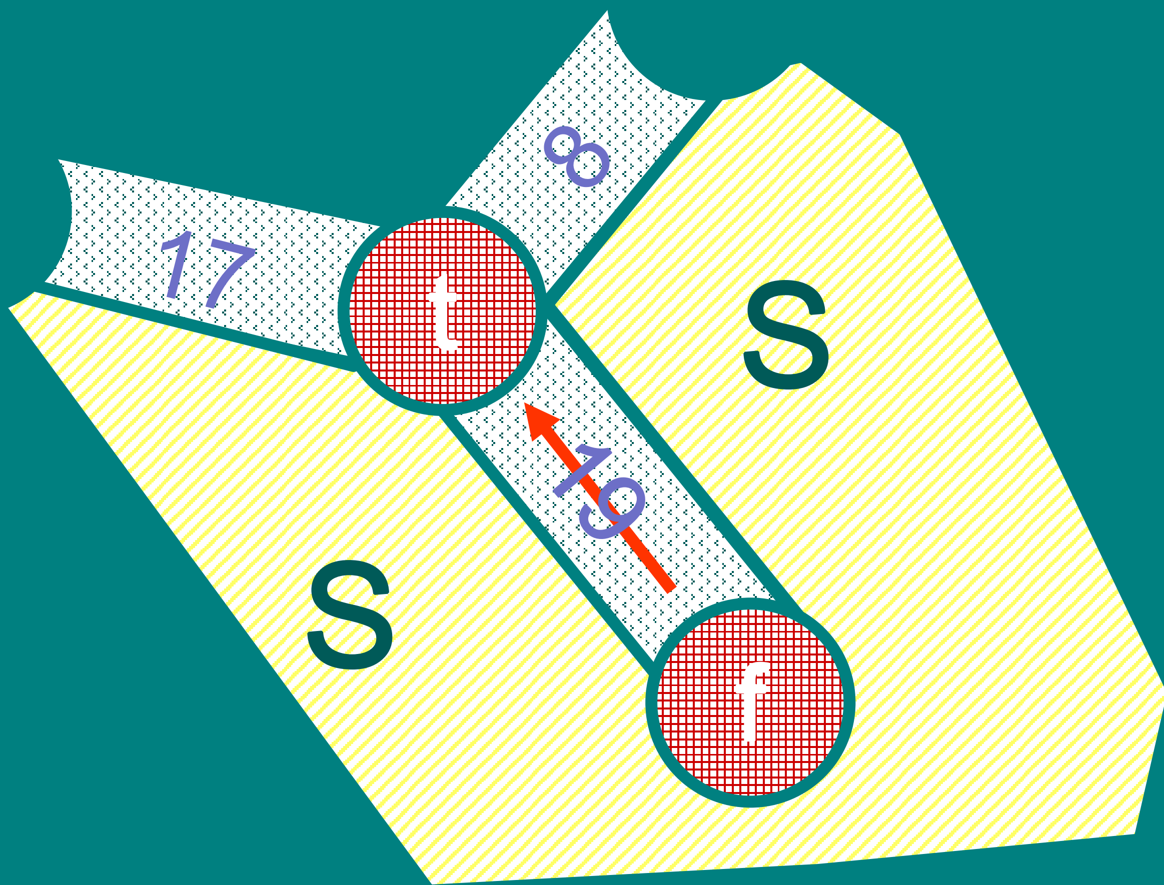
1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	**L**	**R**	43	8	409	17

Special case edge record: vertex of degree 2



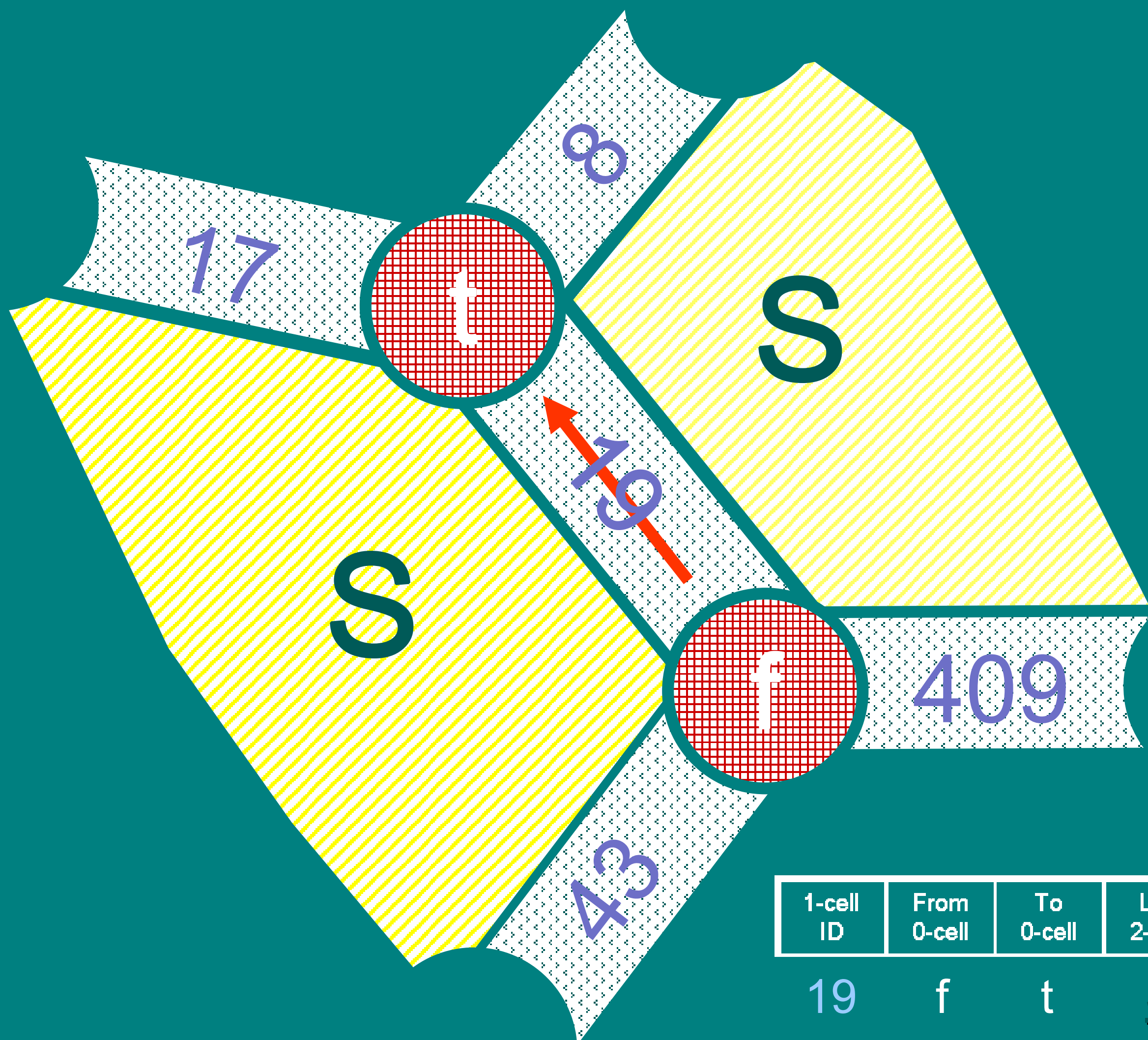
1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	L	R	43	8	43	17

Special case edge record: vertex of degree 1 (dead end)



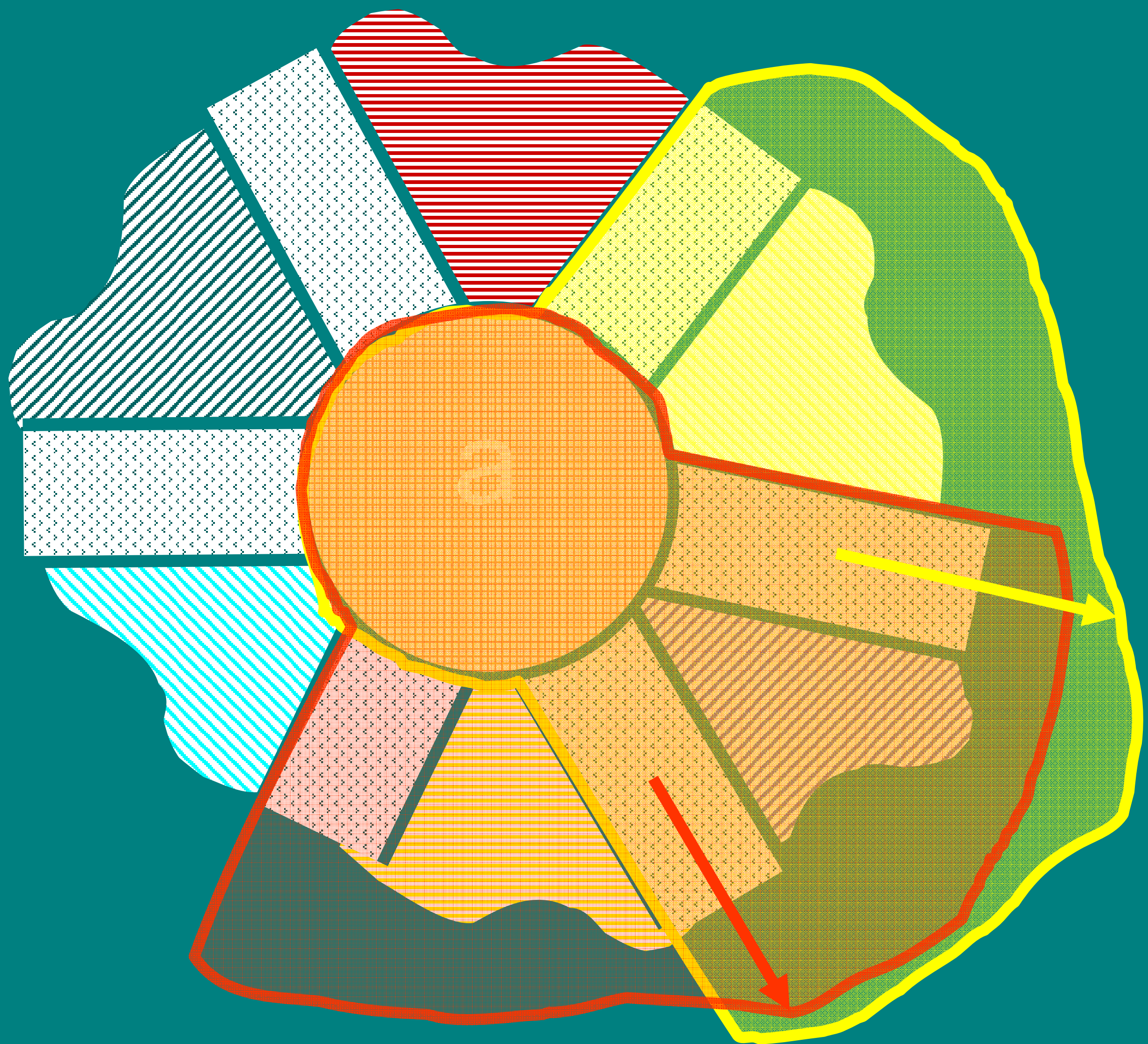
1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	S	S	19	8	19	17

Special case edge record: same 2-cell region on left and right

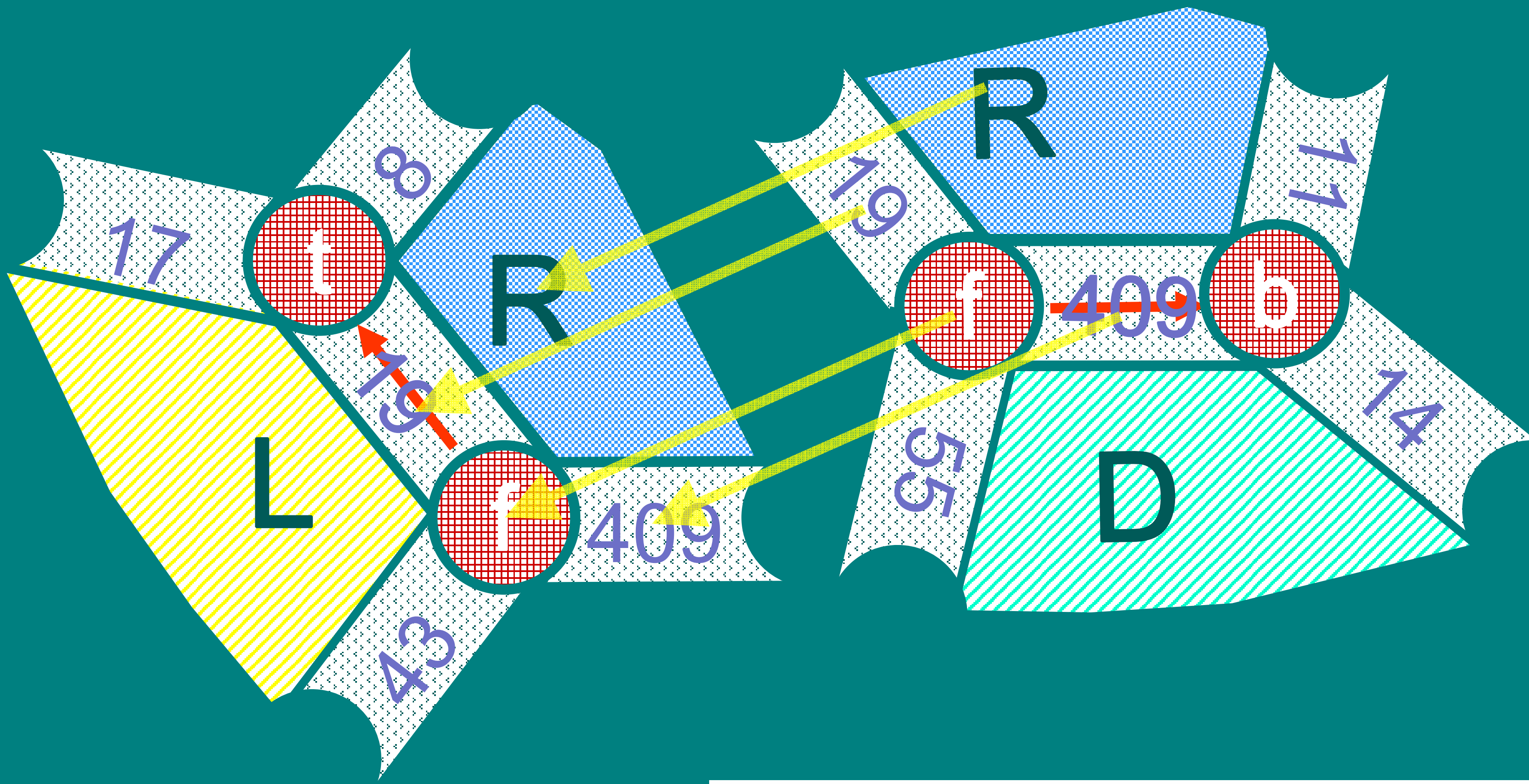


1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	<u>S</u>	<u>S</u>	43	8	409	17

An umbrella edit conceptually overlays attached winged edge components

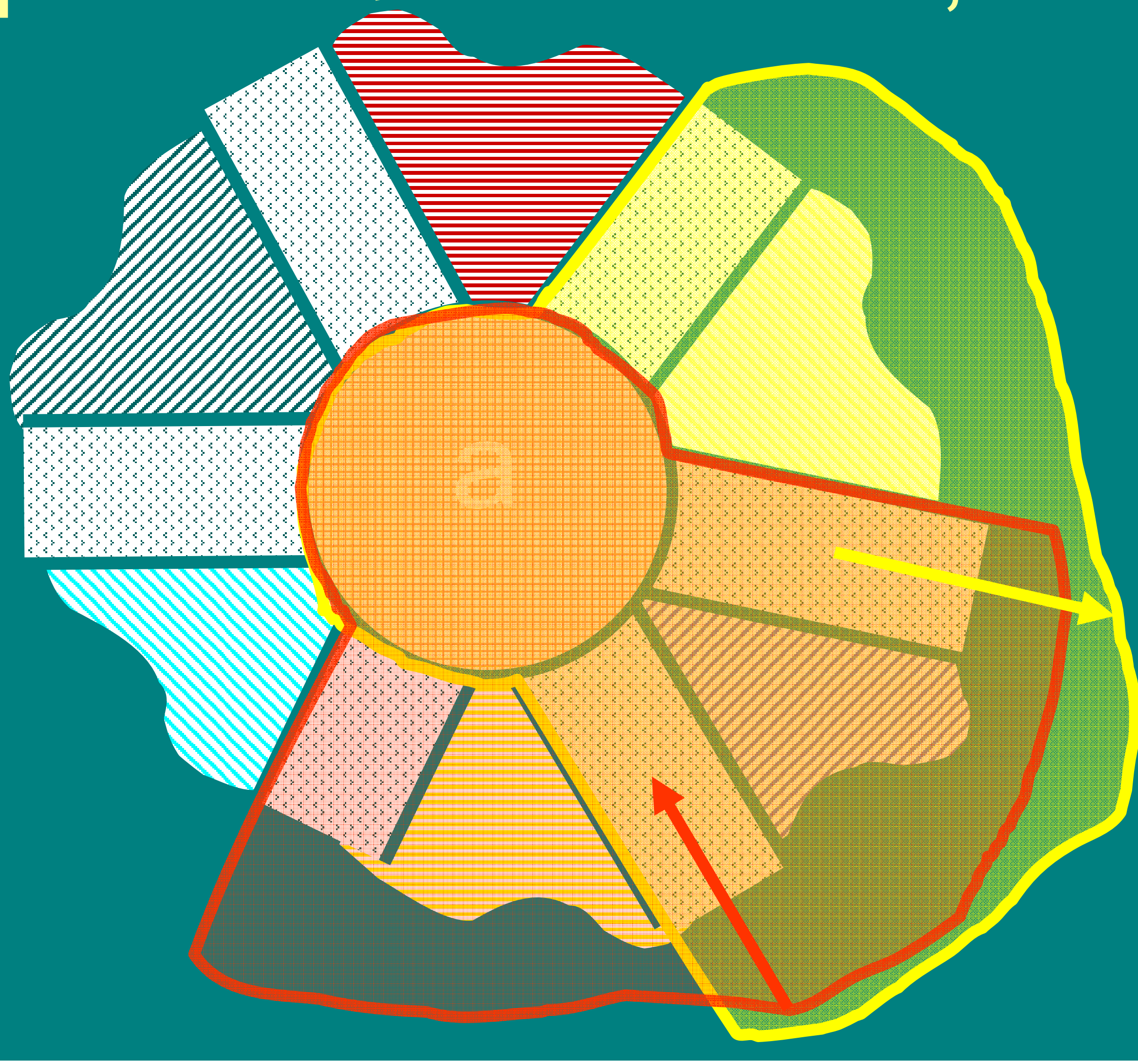


Adjacent winged-edges align

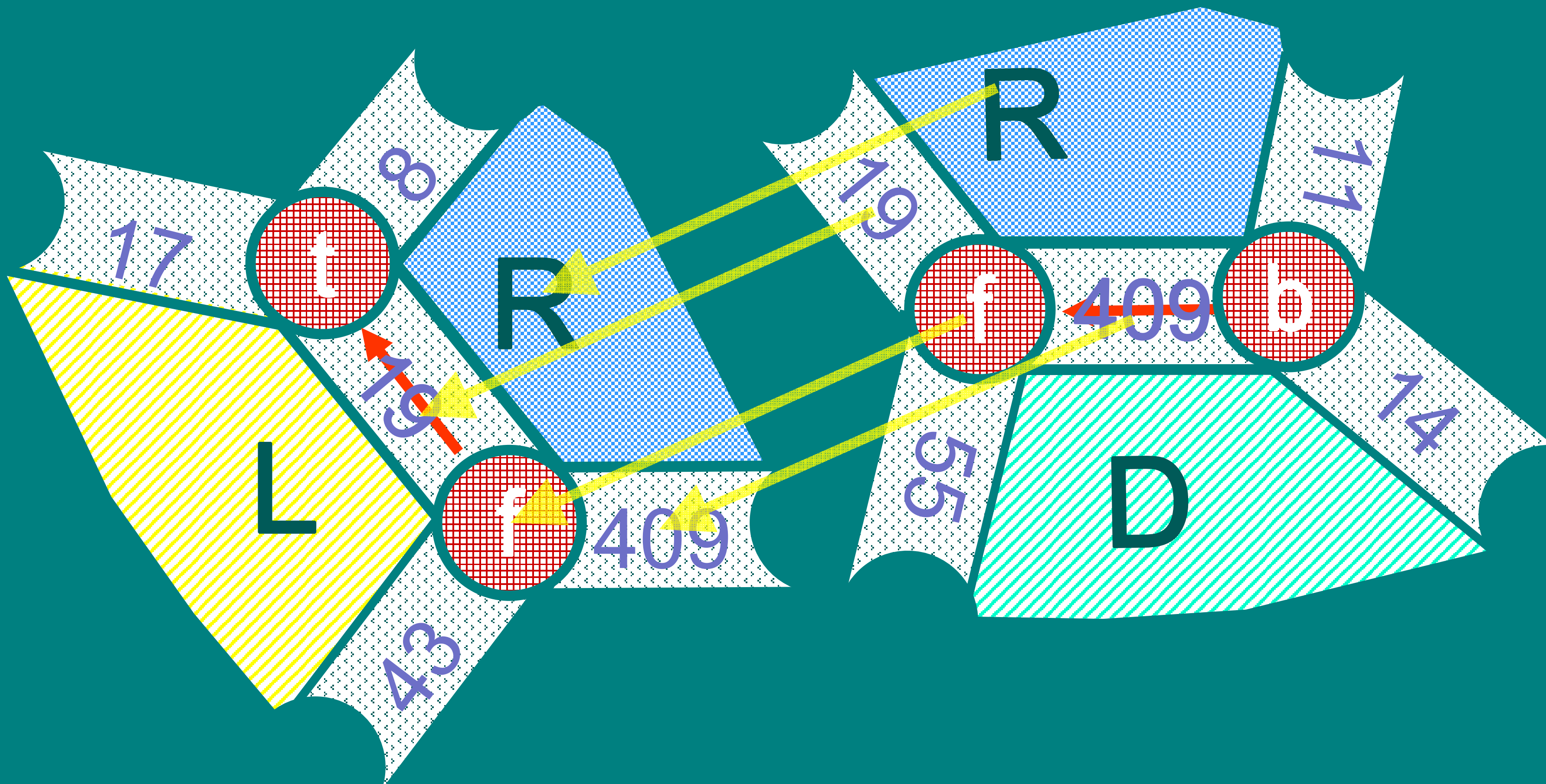


1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	L	R	43	8	409	17
409	f	b	R	D	19	14	55	11

An umbrella edit conceptually overlays attached winged edge components, in this case, head to tail

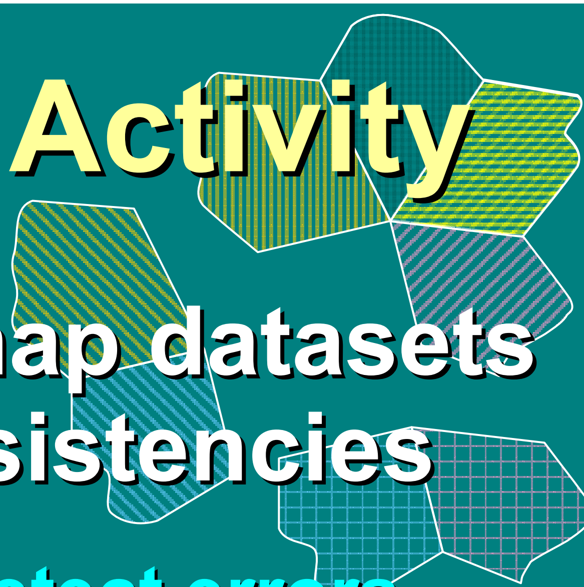


Adjacent winged-edges align



1-cell ID	From 0-cell	To 0-cell	Left 2-cell	Right 2-cell	From/ Left	To/ Right	From/ Right	To/ Left
19	f	t	L	R	43	8	409	17
409	b	f	D	R	14	19	11	55

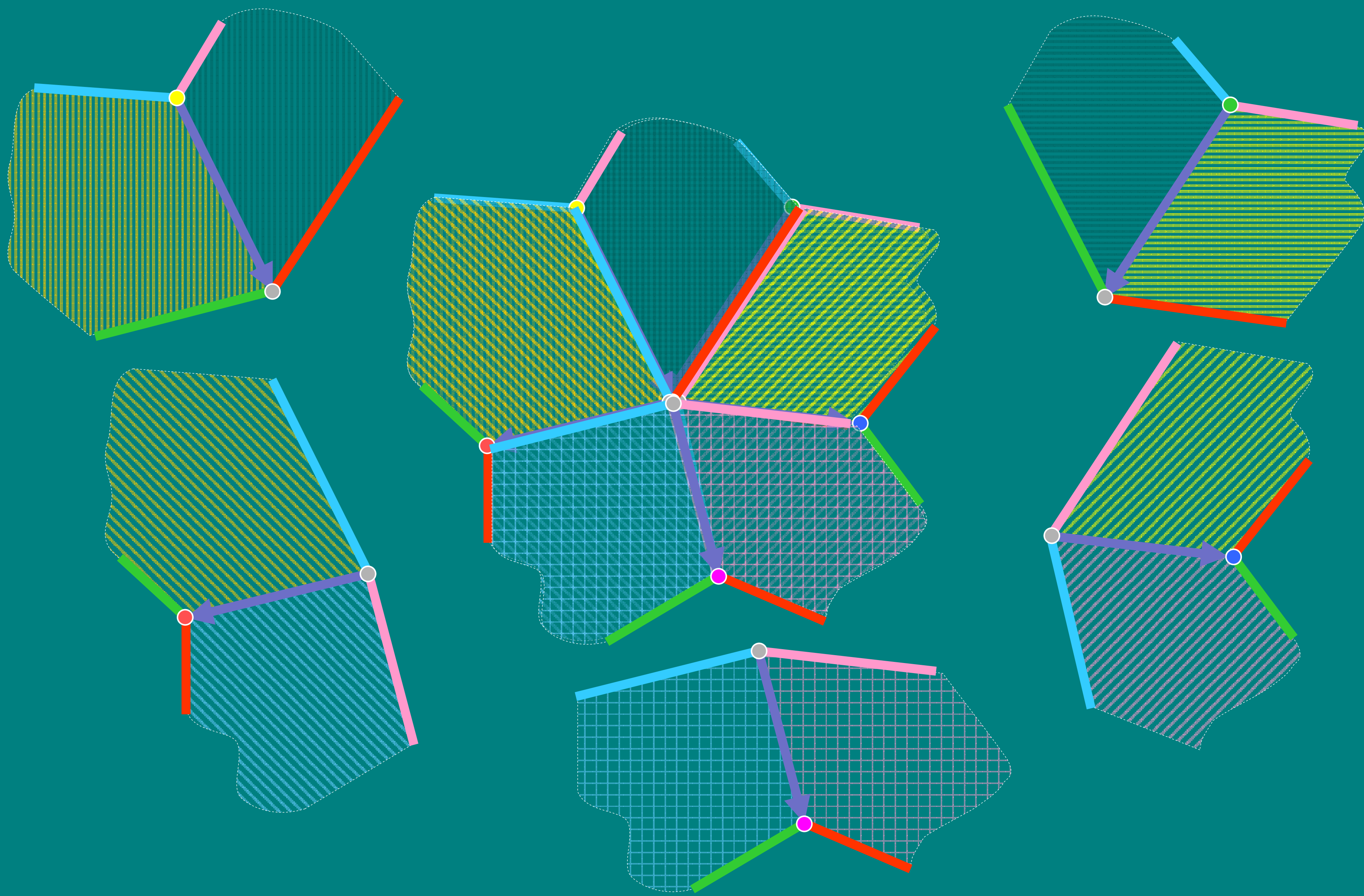
Ongoing Research Activity



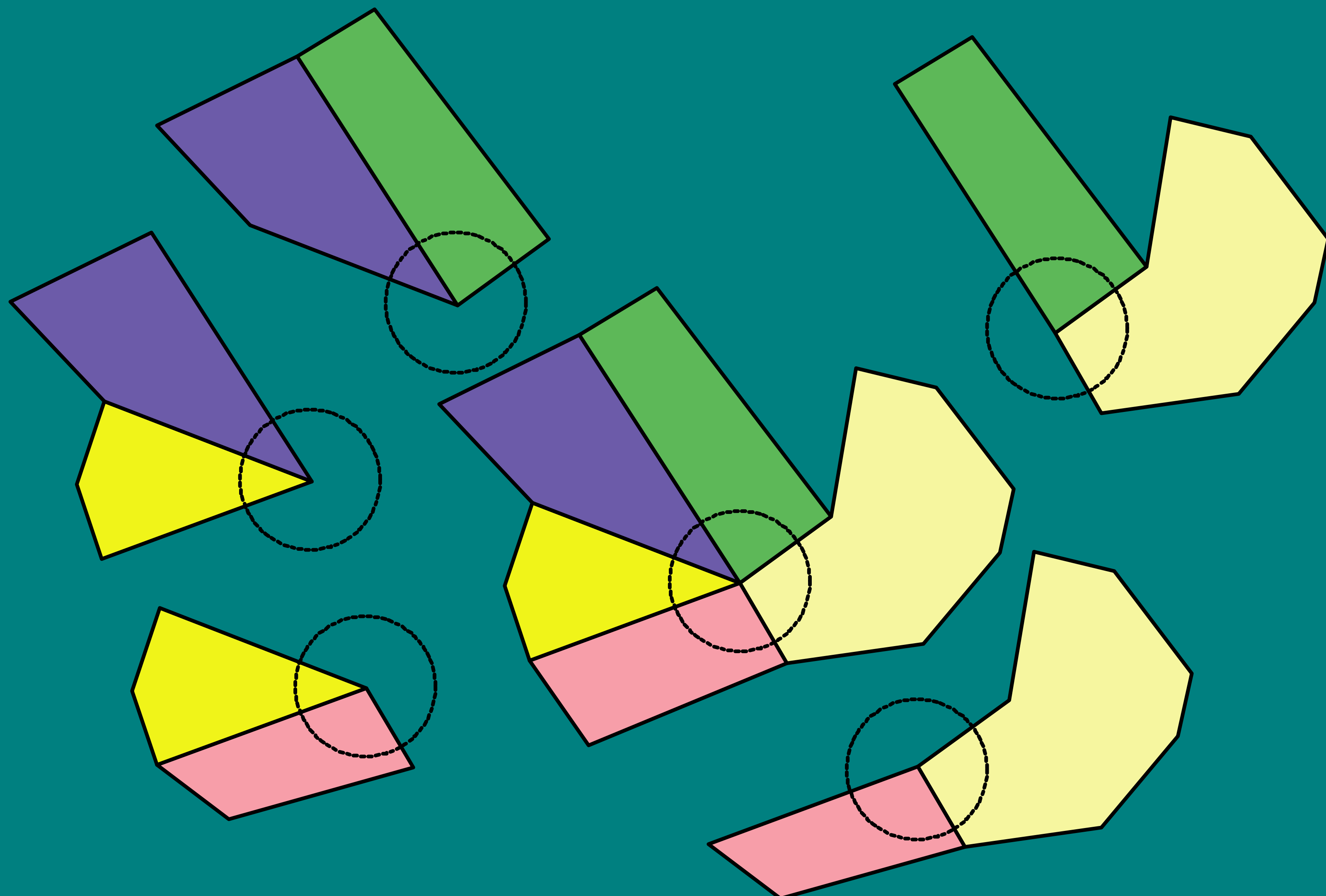
Algorithms for handling map datasets having errors or inconsistencies

- **Tests either validate planarity or detect errors**
  - **How to recognize errors of record omission?**
  - **How to recognize errors of adjacency relation mis-identification?**
- **Recovery of planarity via correction of errors**
  - **Main idea: exploit redundancy of winged-edge model.**
  - **How to fix a single error of omission?**
  - **How to fix a single error of mis-identification?**
  - **What about multiple errors?**
    - **Local vs non-local: only local matters.**

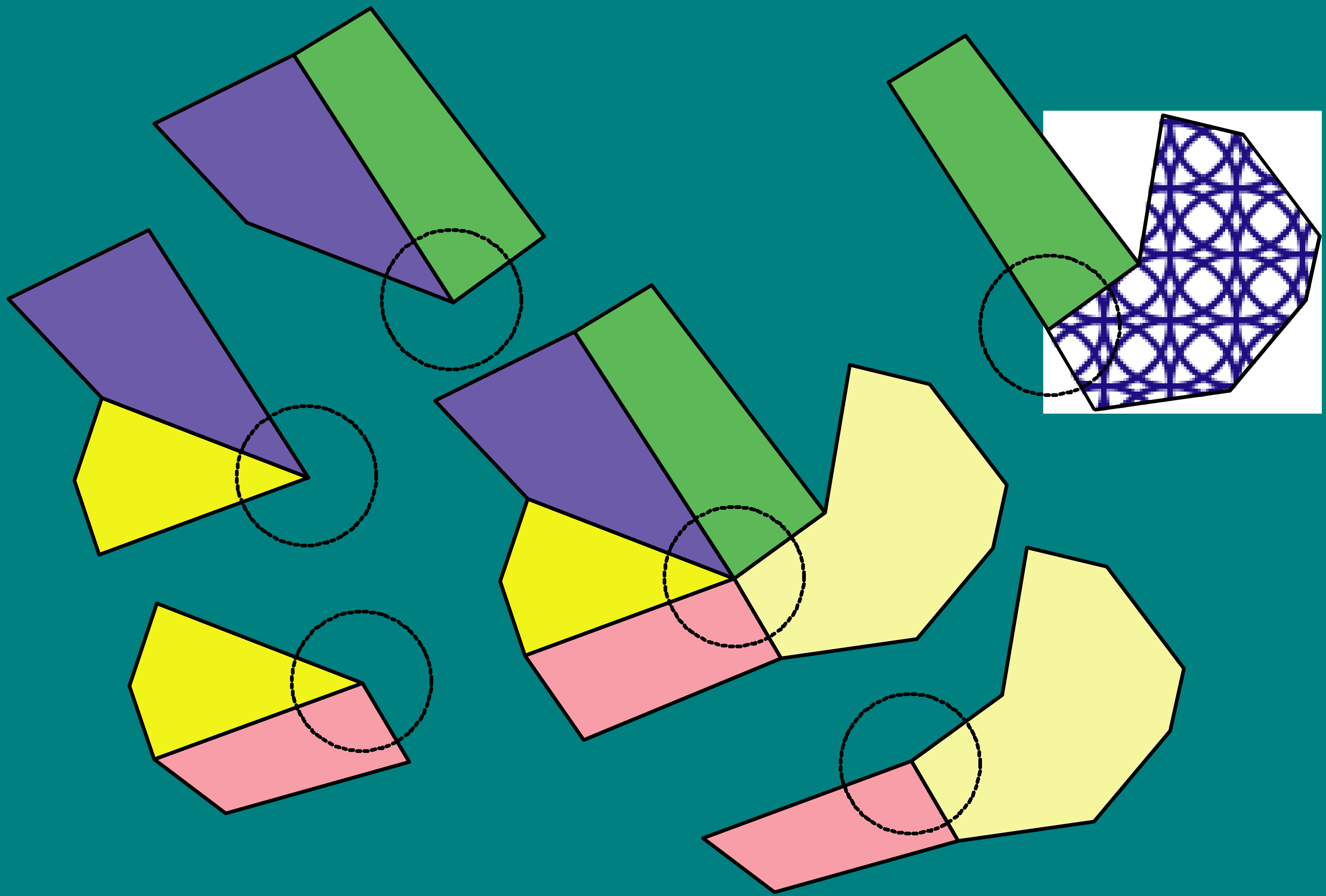
Successful umbrella edit



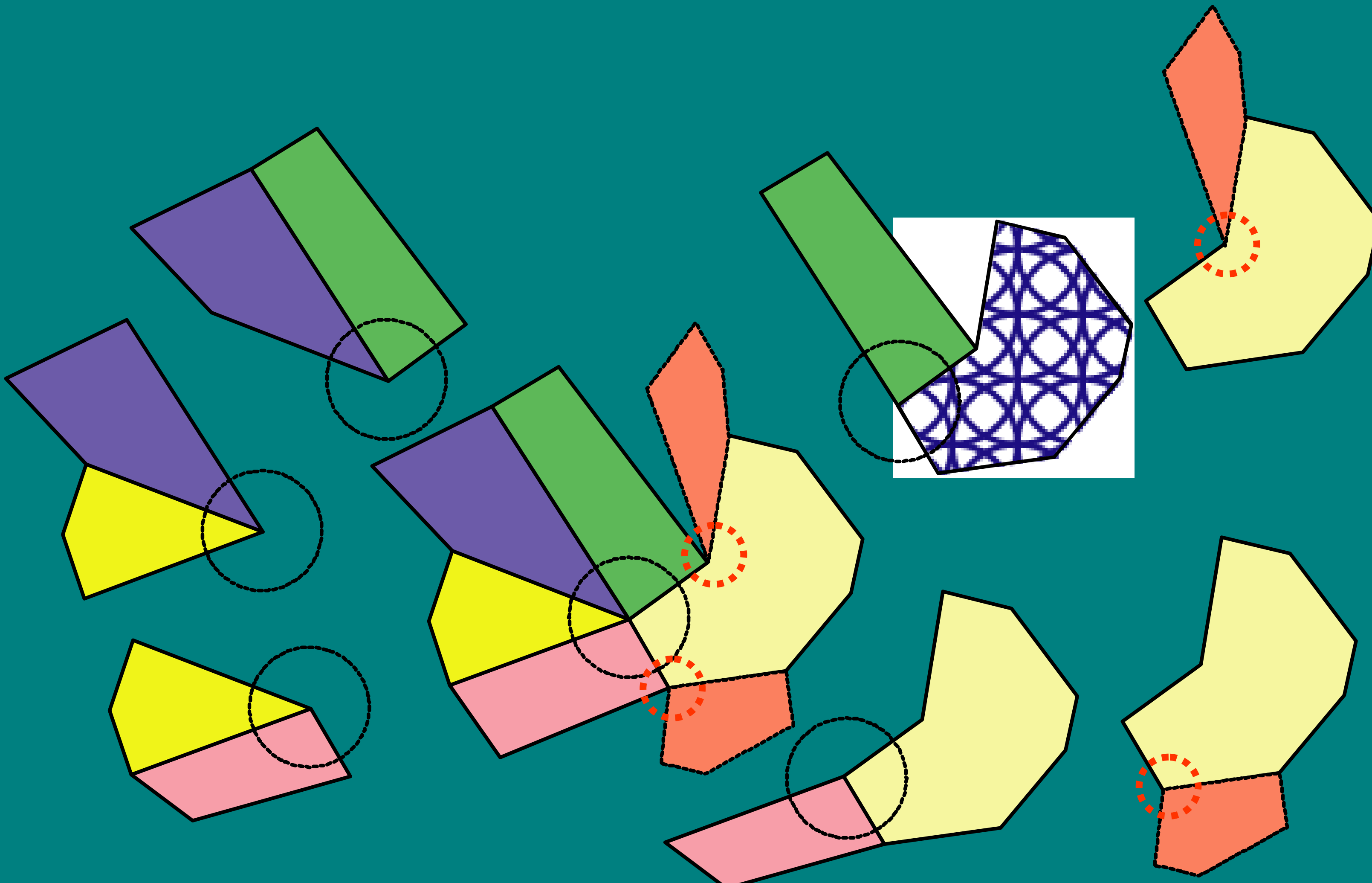
Successful umbrella edit



Unsuccessful umbrella edit



Follow unsuccessful umbrella edit by a cycle edit around conflict



To be continued…

- I expect to soon finish a tech report on the topological edits and a paper on maps as equivalence classes of graphs on the plane.
- Contact me at saalfeld.1@osu.edu if you wish me to send you a preprint of either document