

# **JTS Topology Suite**

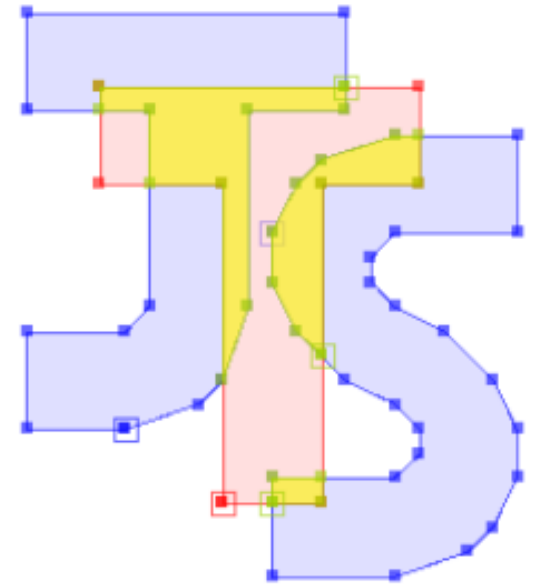
**A Library for Geometry Processing**

***Martin Davis***

***March 2011***

# JTS Topology Suite

- **Core API for processing Geometry**
- **Full implementation of *OpenGIS Consortium Simple Features for SQL* specification**
- **100% Java, Open Source (LGPL)**
- **750+ classes, 66K LOC**
- **Design Features:**
  - Fast
  - Robust
  - Complete
- **History**
  - Version 1.0 released Feb 2002
  - Version 1.11 released March 2010
  - Version 1.12 due Q1 2011

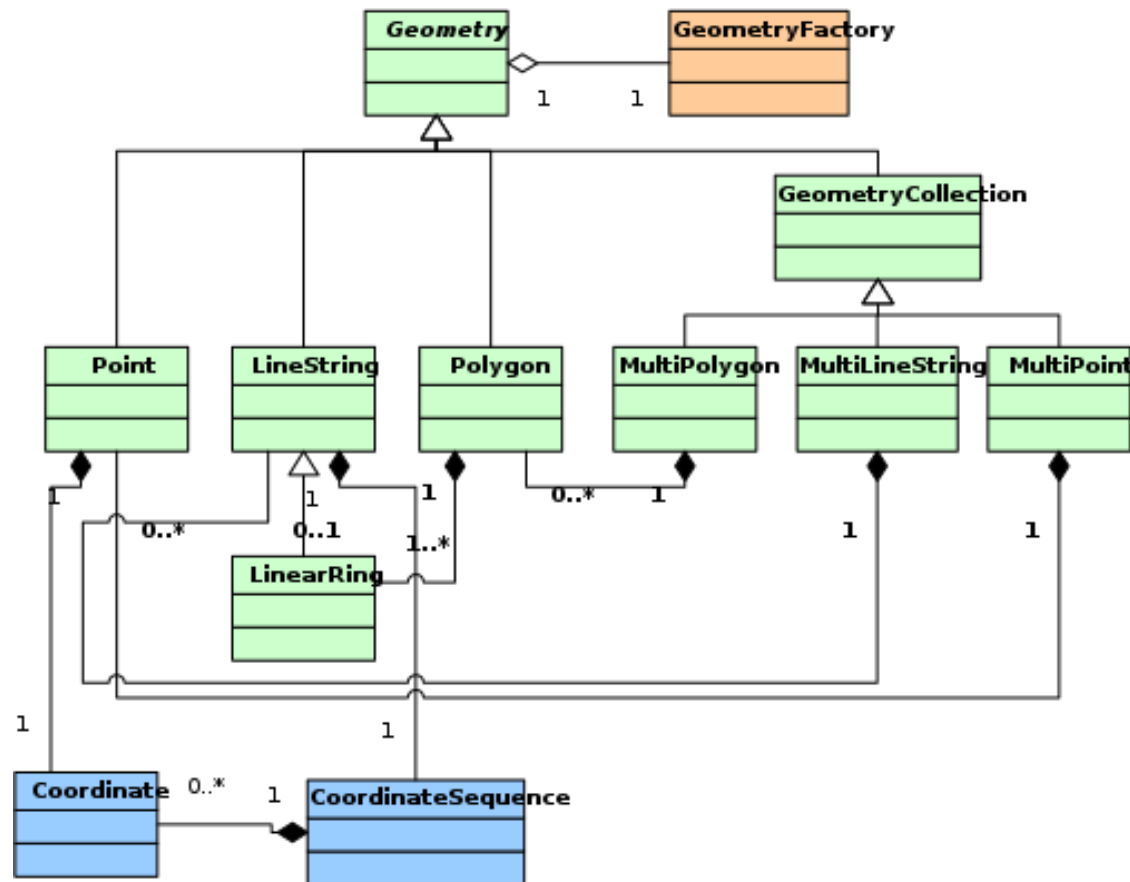
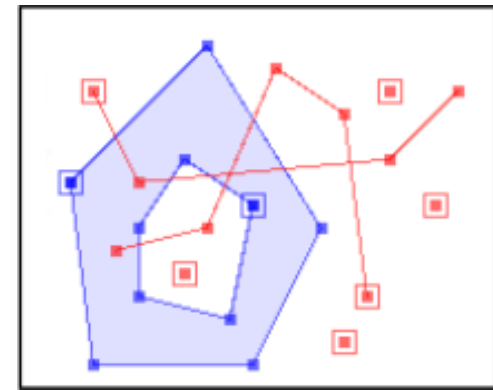


# Geometry Model

- Complete model for 2-D linear geometry (OGC SFS model)

- Point
- LineString, LinearRing
- Polygon (with holes)
- MultiPoint, MultiLineString, MultiPolygon
- GeometryCollection

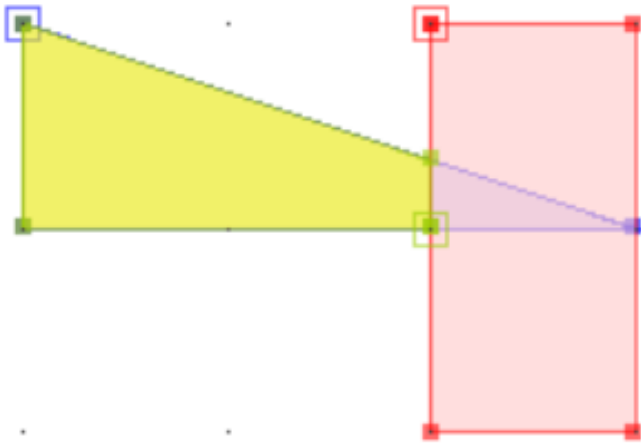
- User-defined coordinate representation



# Explicit Precision Model

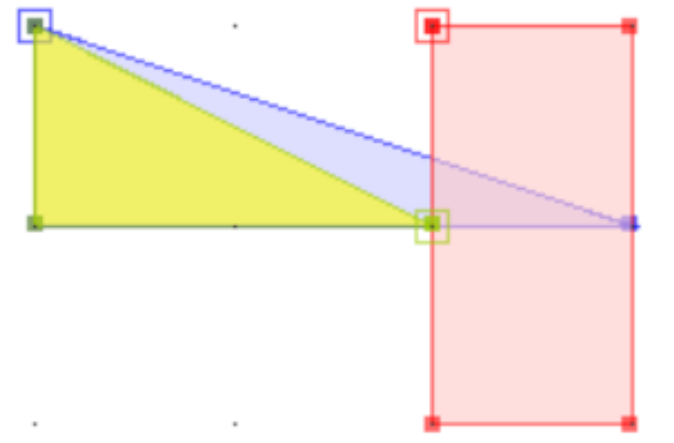
- Provides ability to specify Precision Model of coordinates
  - Floating - Double & Single Precision (IEEE-754)
  - Fixed - specified # of decimal places
- Ensures constructive geometry operations are closed over the specified coordinate space

*Floating*



POLYGON ((3 2, 1 2, 1 3, 3  
2.3333333333333333, 3 2))

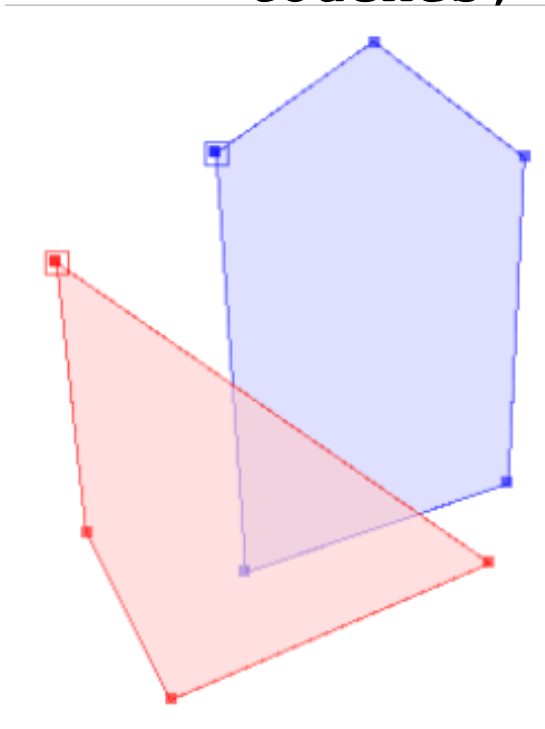
*Fixed*



POLYGON ((3 2, 1 2, 1 3, 3 2))

# Spatial Predicates

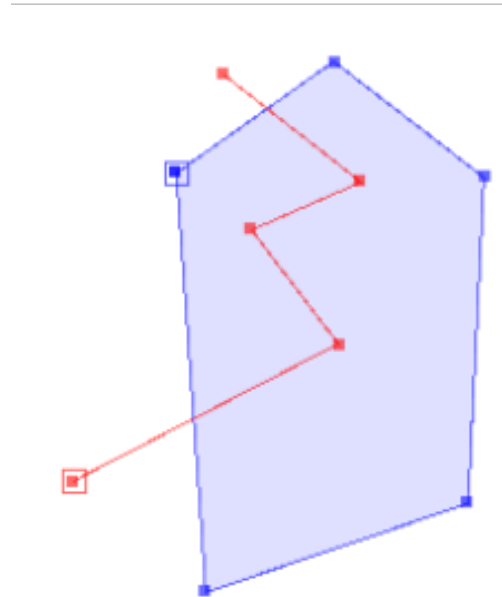
- Computes the spatial relationship of two Geometries
- Implements the *Dimensionally Extended 9-Intersection Model (DE-9IM)*
  - Computes dimension of intersection of Interior, Boundary, Exterior
- General function
  - `relate( IMpattern )`
- Named predicates
  - `intersects`, `contains`, `within`, `equals`, `disjoint`, `touches`, `crosses`, `overlaps`, `covers`, `coveredBy`



		B		
		Int	Bdy	Ext
A	Int	2	1	2
	Bdy	1	0	1
	Ext	2	1	2

Binary Predicates		
	AB	BA
Equals	F	F
Disjoint	F	F
Intersects	T	T
Touches	F	F
Crosses	F	F
Within	F	F
Contains	F	F
Overlaps	T	T



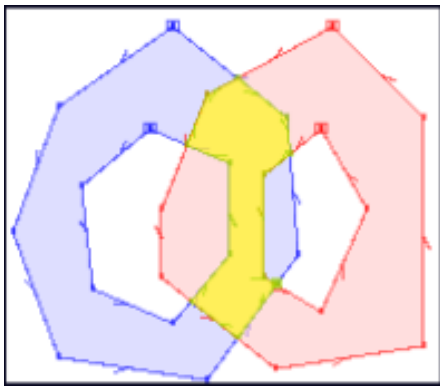
		B		
		Int	Bdy	Ext
A	Int	1	F	2
	Bdy	0	F	1
	Ext	1	0	2

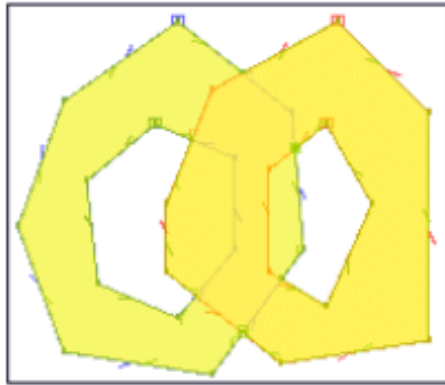
Binary Predicates		
	AB	BA
Equals	F	F
Disjoint	F	F
Intersects	T	T
Touches	F	F
Crosses	T	T
Within	F	F
Contains	F	F
Overlaps	F	F

# Overlay Methods

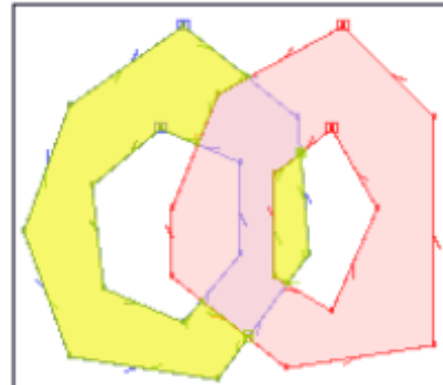
- AKA Boolean functions, Set-theoretic functions



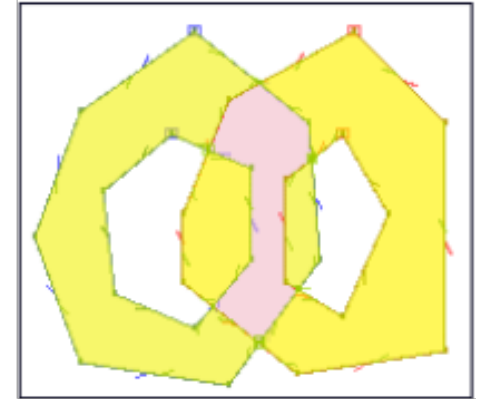
*Intersection*



*Union*

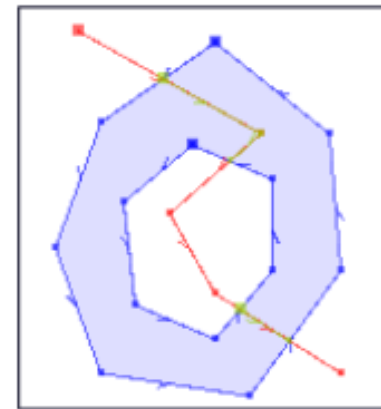


*Difference*



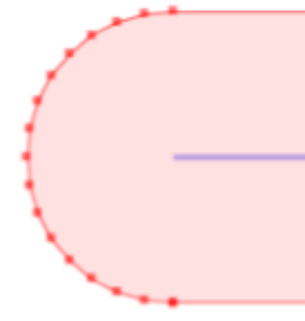
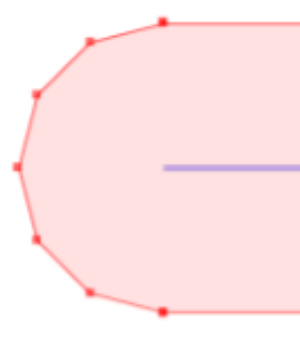
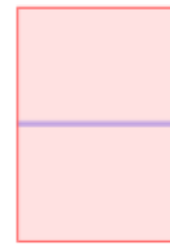
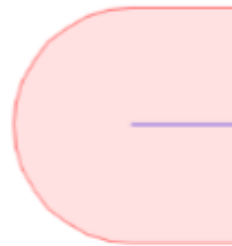
*Symmetric Difference*

- Heterogeneous – all geometry types supported



# Buffering

- **Positive & Negative buffers**
  - All Geometry types
  - Robust, efficient algorithm
- **Choice of End Cap Styles**
  - Round, Square, Butt
- **Curve Densification is user controllable**



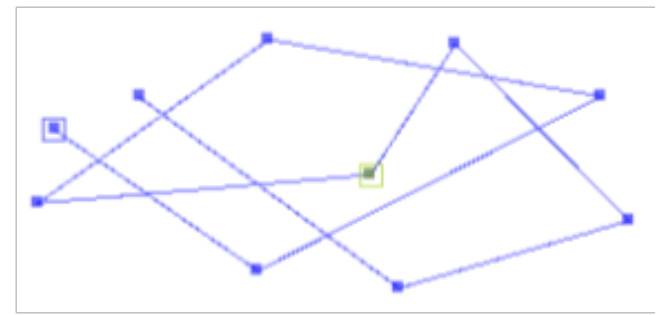
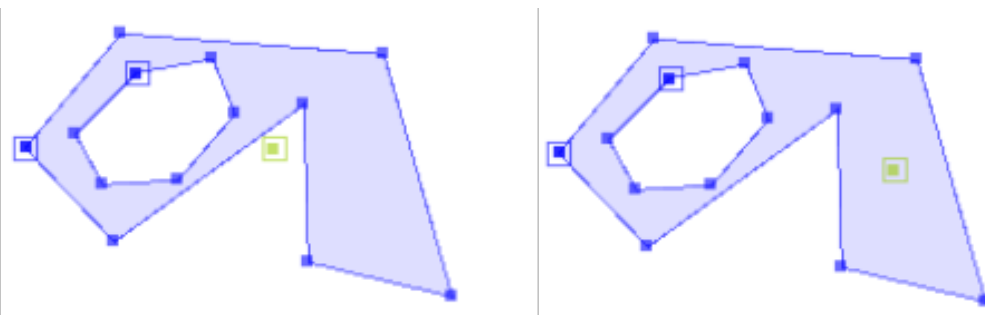
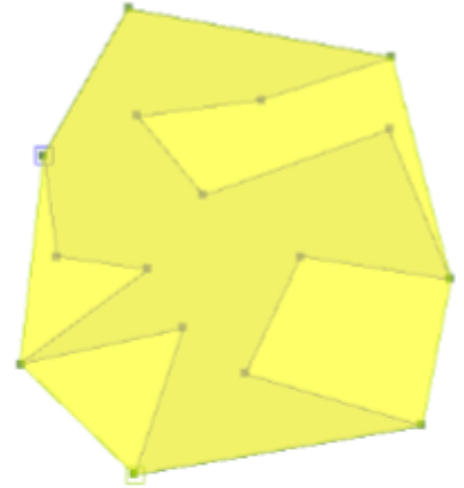
# Other Constructive Methods

- **Convex Hull**

- Standard Computational Geometry algorithm

- **Centroid & InteriorPoint**

- Centroid is center of mass (not necessarily in interior)
- Interior point *always* in interior, as close to centre as possible
- all Geometry types supported





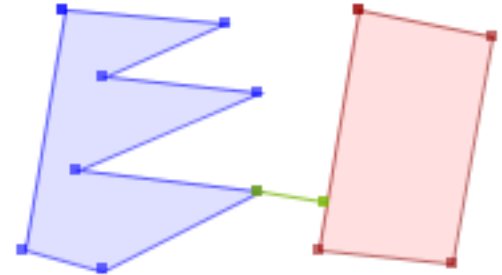
# Metric Methods

- **Area, Length**

- Length = Perimeter, for Area geometries

- **Distance**

- Constructive – computes location of points providing minimum distance

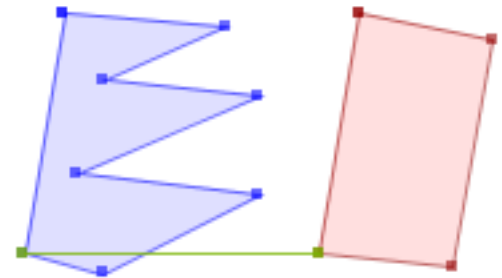


- **WithinDistance**

- “Limited predicate” allows optimized computation

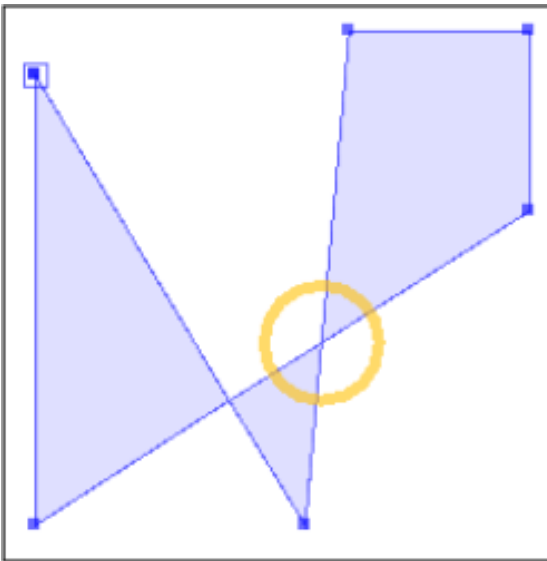
- **Hausdorff Distance**

- "How far apart"

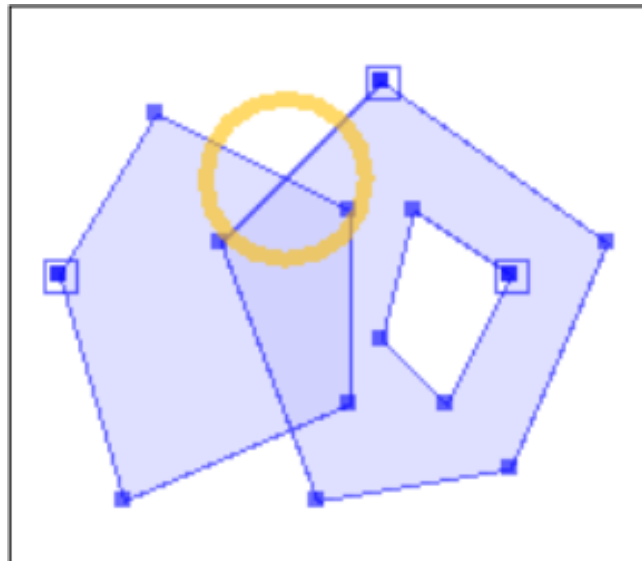


# Geometry Validation

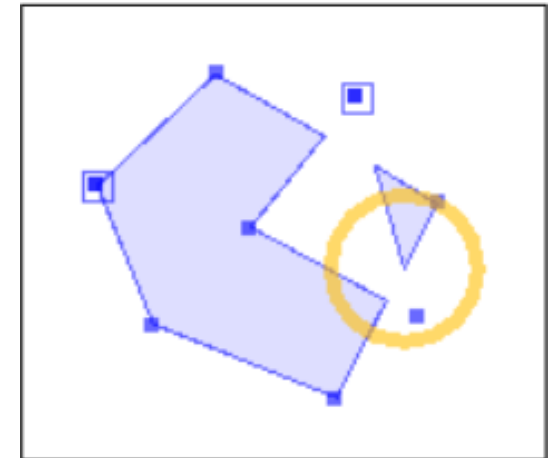
- **Validation of Geometry topology essential to ensure correct spatial processing**
  - Polygons in particular – many possible invalid situations
- **JTS provides full Validation of Topology**
  - **isValid** provides simple good/bad test
  - **ValidOp** class provides detailed error information, including location



***Self-intersection***



***Overlapping Rings***

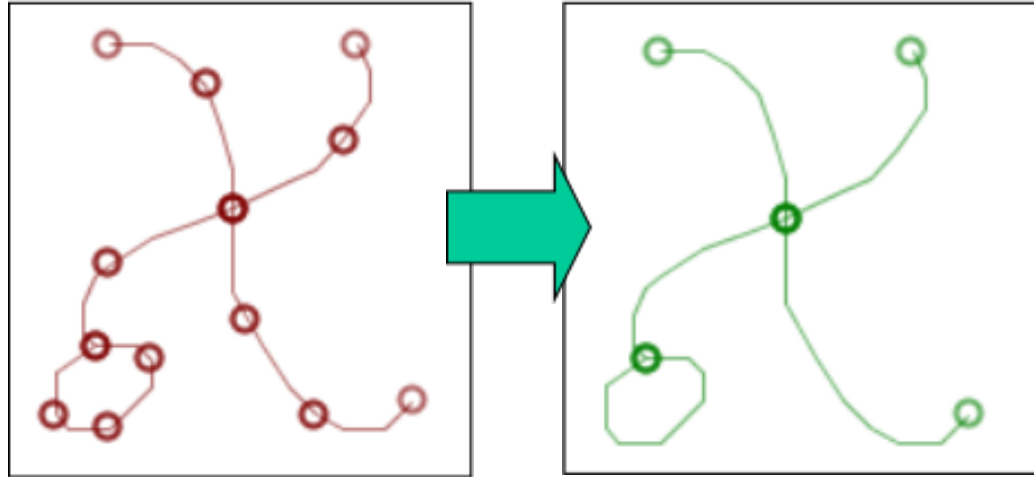


***Hole intersects shell***

# Line Merging, Polygonization

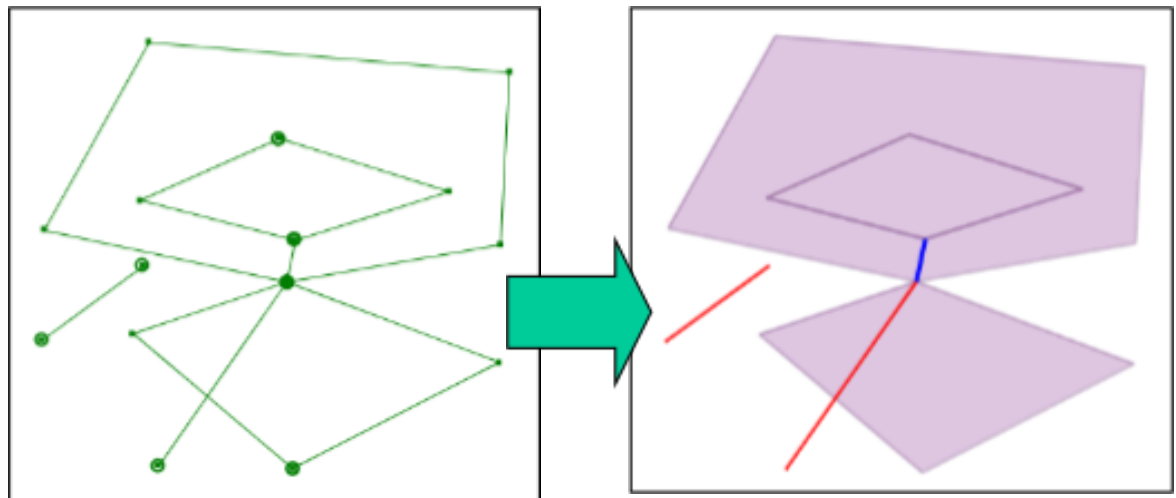
- **Line Merging**

- Removes 2-nodes from set of LineStrings



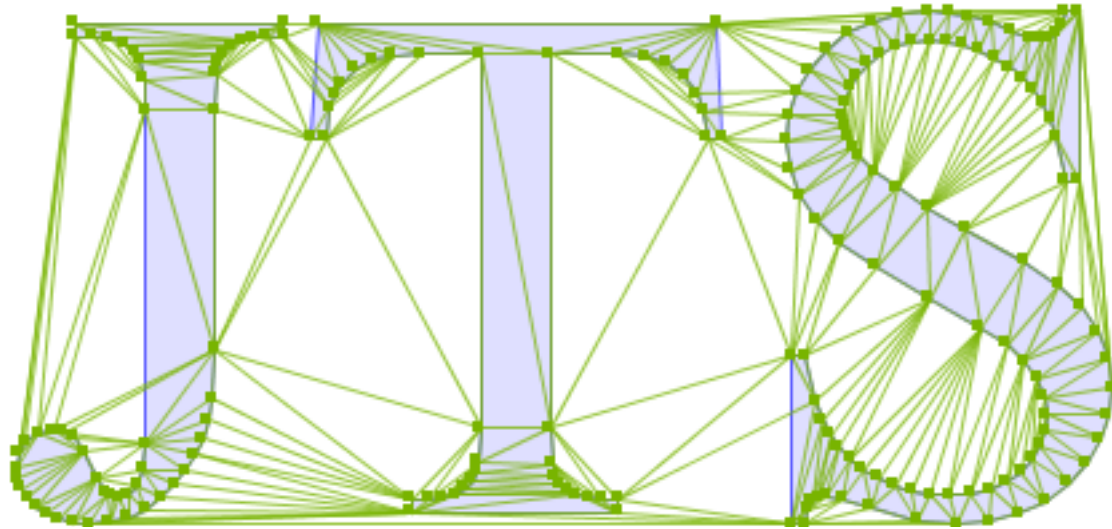
- **Polygonization**

- Finds errors (Dangles, Cutlines)

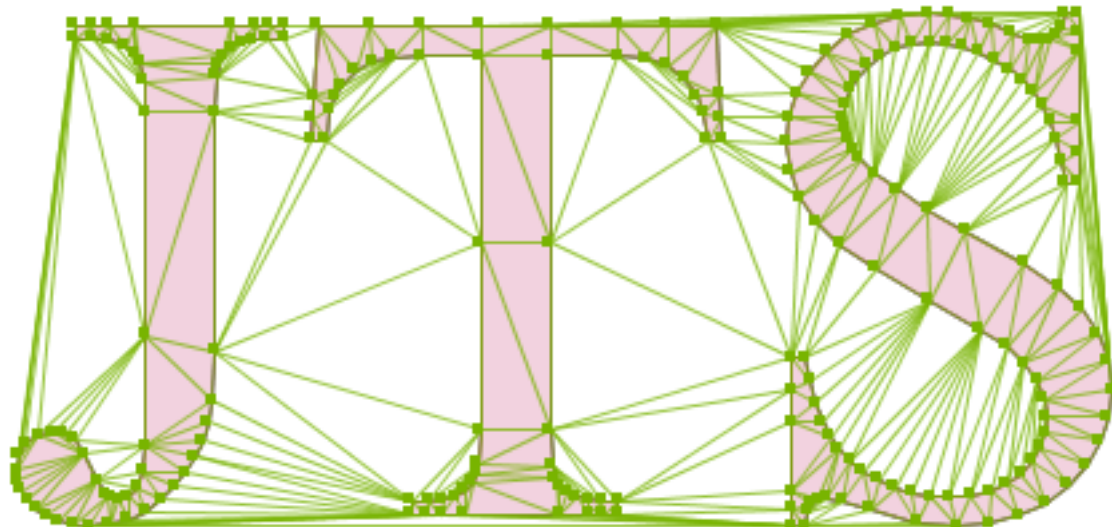


# Delaunay Triangulation

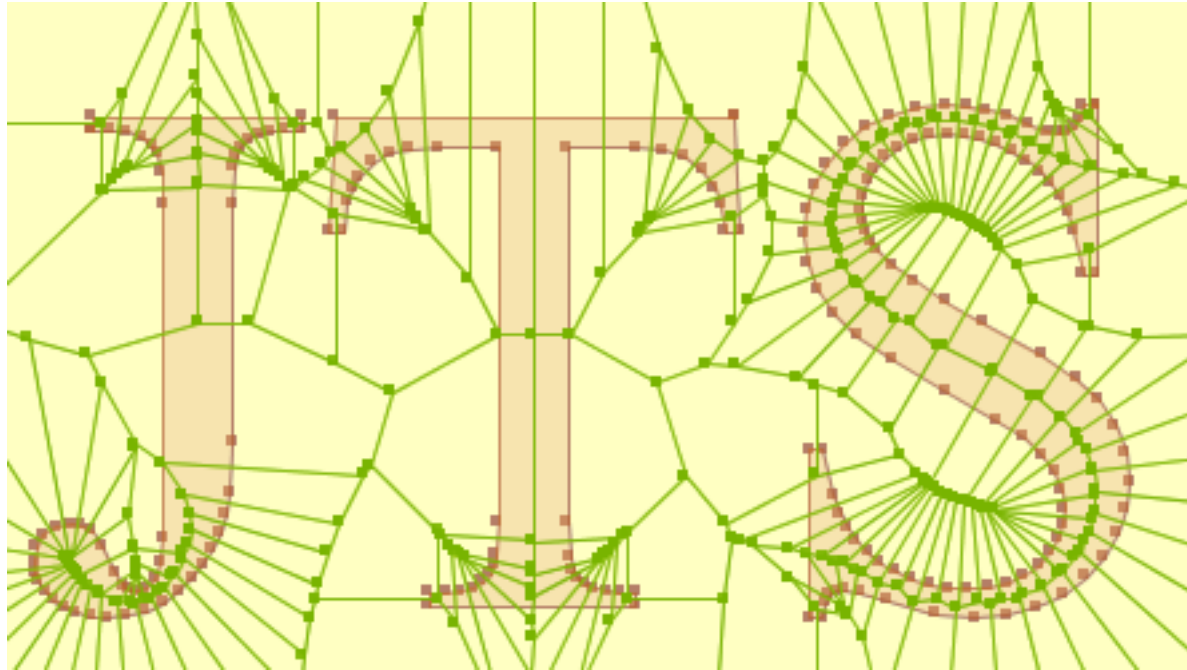
- Delaunay Triangulation



- Conforming Delaunay Triangulation



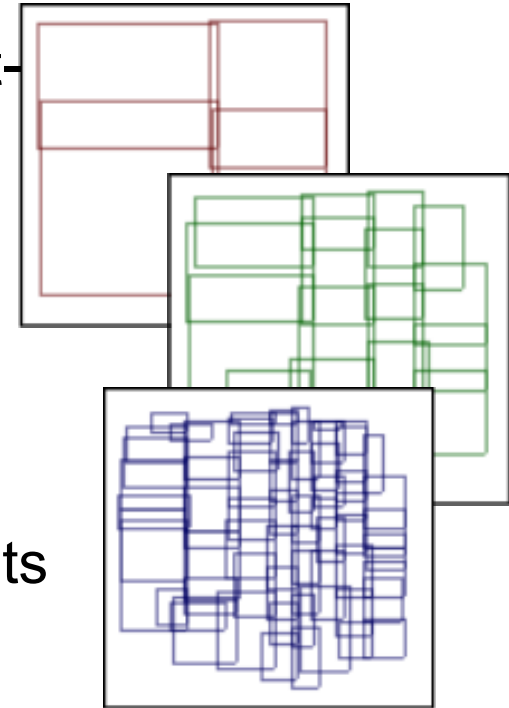
# Voronoi Diagram



- AKA Thiessen Polygons

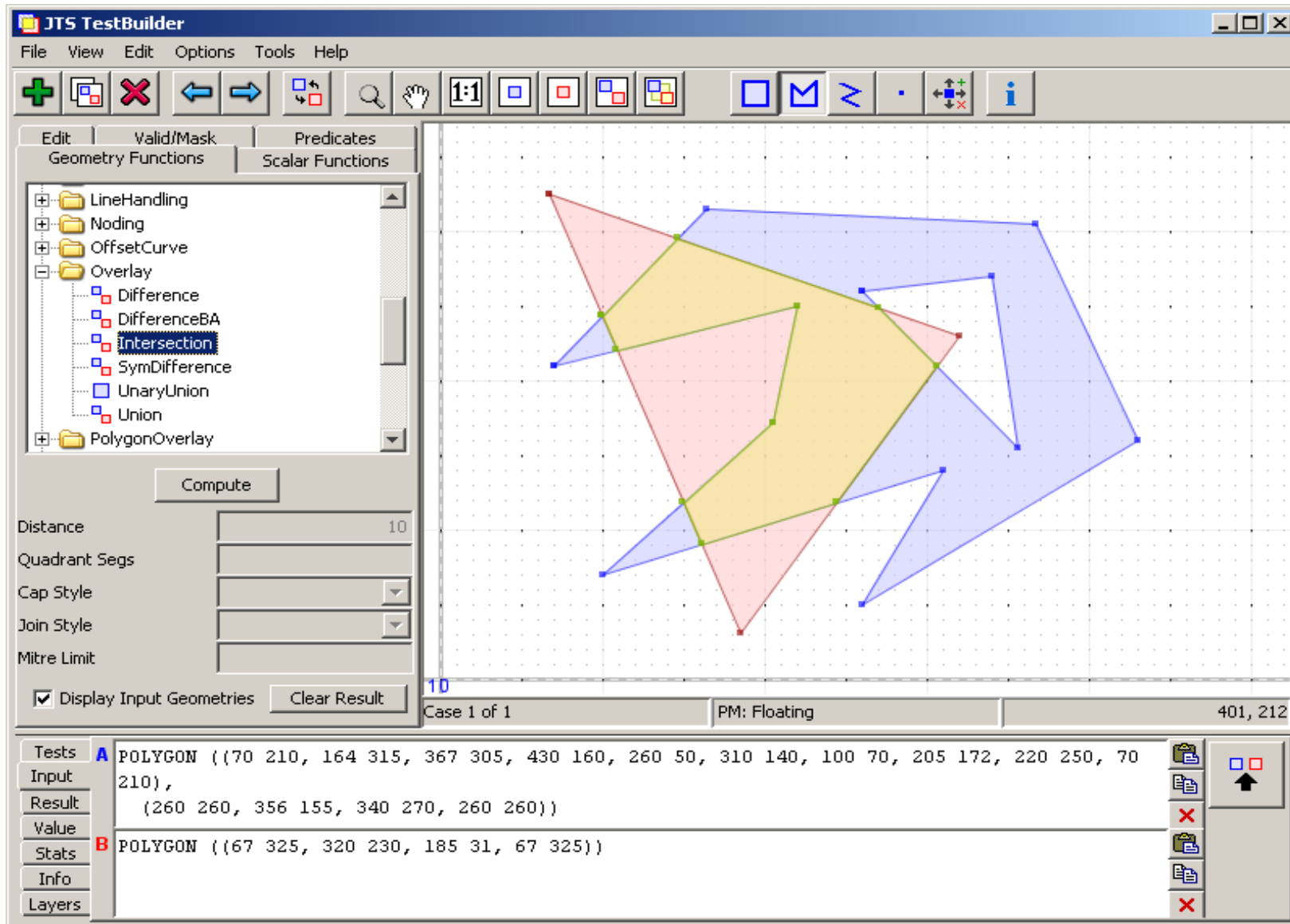
# Spatial Algorithms & Structures

- **Numerous fundamental CG algorithms**
  - Line segment intersection, Ring orientation, Point-Line orientation, Point-line distance, etc.
- **Spatial Indexes**
  - Quadtree, STRtree, kD-tree, Bintree, MonotoneChains, SweepLine
- **Line segment Noding**
  - find/create all intersections in set of Line Segments
  - Snap-Rounding
- **Planar Graph framework**
- **Precision Reduction**



# JTS TestBuilder

- Create/edit/view geometry
- Compute & view results of all JTS operations



# JTS In Use

- **JTS used for geometry processing in numerous open source and commercial geospatial applications**
  - JUMP
  - BC Gov't Electronic Submission Framework
  - Internet Mapping Framework
  - GeoServer / GeoTools
  - Deegree
  - PostGIS (as GEOS)
  - Tlogica (Bulgaria)
- **Other interesting applications**
  - Font Creator (RobMeek.com)



# Future Work

- **Fully robust Overlay Operations**
- **Improve performance**
  - e.g. line noding, distance computation
- **Optimize repeated method calls on single Geometry**
  - e.g. *“find all geometries which intersect this geometry”*
- **Improve internal structure**
- **Geometry simplification / generalization methods**
  - Douglas-Peucker line simplification, etc
- **Generalized Distance methods**
  - Hausdorff distance, Frechet distance, etc
- **User-defined Geometry representation**
  - Allows easier adaptation to other Geometry APIs, database structures
- **Affine Transform**
- **Linear Referencing operations**
- **Improved/Extended Spatial Indexes**
  - Updatable Quadtree, B tree, Visitor pattern