Spatial Analysis and Modeling (GIST 4302/5302)

Guofeng Cao
Department of Geosciences
Texas Tech University
Class Outlines

Spatial Point Pattern

Regional Data (Areal Data)

Continuous Spatial Data (Geostatistical Data)

Representation and Operations of Spatial Data

What is Special about Spatial? Characteristics of Spatial Data and Map Projection
Representation of Spatial Data
Representation of Spatial Data Models

• **Object-based model:** treats the space as populated by discrete, identifiable entities each with a geospatial reference
  – Buildings or roads fit into this view
  – GIS Softwares: ArcGIS

• **Field-based model:** treats geographic information as collections of spatial distributions
  – Distribution may be formalized as a mathematical function from a spatial framework to an attribute domain
  – Patterns of topographic altitudes, rainfall, and temperature fit neatly into this view.
  – GIS Software: Grass
Object-based Approach
Entity

• Object-based models decompose an information space into objects or *entities*

• An entity must be:
  – Identifiable
  – Relevant (be of interest)
  – Describable (have characteristics)

• The frame of spatial reference is provided by the entities themselves
Example: House object

Has several attributes, such as registration date, address, owner and boundary, which are themselves objects.
# Attribute Tables

<table>
<thead>
<tr>
<th>ID</th>
<th>Point</th>
<th>13</th>
<th>7.181452</th>
<th>70074</th>
<th>34.199444</th>
<th>-118.534722</th>
<th>-118.53472</th>
<th>34.19944</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Point</td>
<td>13</td>
<td>6.076613</td>
<td>70075</td>
<td>34.066944</td>
<td>-117.751389</td>
<td>-117.75139</td>
<td>34.06694</td>
</tr>
<tr>
<td>5</td>
<td>Point</td>
<td>8</td>
<td>3.157258</td>
<td>70084</td>
<td>33.929167</td>
<td>-118.209722</td>
<td>-118.20972</td>
<td>33.92917</td>
</tr>
<tr>
<td>6</td>
<td>Point</td>
<td>11</td>
<td>5.201613</td>
<td>70085</td>
<td>34.015</td>
<td>-118.059722</td>
<td>-118.05972</td>
<td>34.015</td>
</tr>
<tr>
<td>7</td>
<td>Point</td>
<td>11</td>
<td>4.717742</td>
<td>70087</td>
<td>34.067222</td>
<td>-118.226389</td>
<td>-118.22639</td>
<td>34.06722</td>
</tr>
<tr>
<td>8</td>
<td>Point</td>
<td>15</td>
<td>6.532258</td>
<td>70088</td>
<td>34.083333</td>
<td>-118.106944</td>
<td>-118.10694</td>
<td>34.08333</td>
</tr>
<tr>
<td>9</td>
<td>Point</td>
<td>15</td>
<td>7.540323</td>
<td>70089</td>
<td>34.3875</td>
<td>-118.534722</td>
<td>-118.53472</td>
<td>34.3875</td>
</tr>
<tr>
<td>10</td>
<td>Point</td>
<td>10</td>
<td>4.891129</td>
<td>70091</td>
<td>34.050833</td>
<td>-118.451417</td>
<td>-118.45147</td>
<td>34.05083</td>
</tr>
<tr>
<td>11</td>
<td>Point</td>
<td>10</td>
<td>4.149194</td>
<td>70094</td>
<td>33.92357</td>
<td>-118.37085</td>
<td>-118.37085</td>
<td>33.92357</td>
</tr>
<tr>
<td>12</td>
<td>Point</td>
<td>13</td>
<td>7.266129</td>
<td>70096</td>
<td>34.69012</td>
<td>-118.1334</td>
<td>-118.1334</td>
<td>34.69012</td>
</tr>
<tr>
<td>13</td>
<td>Point</td>
<td>9</td>
<td>4.100806</td>
<td>30176</td>
<td>33.820278</td>
<td>-117.9125</td>
<td>-117.9125</td>
<td>33.82028</td>
</tr>
<tr>
<td>14</td>
<td>Point</td>
<td>10</td>
<td>4.737903</td>
<td>30177</td>
<td>33.926111</td>
<td>-117.951389</td>
<td>-117.95139</td>
<td>33.92611</td>
</tr>
<tr>
<td>15</td>
<td>Point</td>
<td>11</td>
<td>4.600806</td>
<td>30186</td>
<td>33.6275</td>
<td>-117.690278</td>
<td>-117.69028</td>
<td>33.6275</td>
</tr>
<tr>
<td>16</td>
<td>Point</td>
<td>8</td>
<td>4.358871</td>
<td>30195</td>
<td>33.67511</td>
<td>-117.9269</td>
<td>-117.9269</td>
<td>33.67511</td>
</tr>
<tr>
<td>17</td>
<td>Point</td>
<td>13</td>
<td>8.241935</td>
<td>33137</td>
<td>33.85</td>
<td>-116.543056</td>
<td>-116.54306</td>
<td>33.85</td>
</tr>
<tr>
<td>18</td>
<td>Point</td>
<td>17</td>
<td>8.794355</td>
<td>33144</td>
<td>34.010278</td>
<td>-117.426389</td>
<td>-117.42639</td>
<td>34.01028</td>
</tr>
<tr>
<td>19</td>
<td>Point</td>
<td>15</td>
<td>8.899194</td>
<td>33149</td>
<td>33.708333</td>
<td>-117.243056</td>
<td>-117.24306</td>
<td>33.70833</td>
</tr>
<tr>
<td>20</td>
<td>Point</td>
<td>14</td>
<td>7.404203</td>
<td>33150</td>
<td>33.6725</td>
<td>-117.1334</td>
<td>-117.1334</td>
<td>33.6725</td>
</tr>
</tbody>
</table>
Attribute Tables

• Basic operators on attribute tables
  – Selection: picking certain rows
  – Projection: picking certain columns
  – Join: compositions of relations
Project Operator

- The **project** operator is unary
  - It outputs a new relation that has a subset of attributes
  - Identical tuples in the output relation are coalesced

<table>
<thead>
<tr>
<th>Relation Sells:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>bar</td>
<td>beer</td>
<td>price</td>
</tr>
<tr>
<td>Chimy’s</td>
<td>Bud</td>
<td>2.50</td>
</tr>
<tr>
<td>Chimy’s</td>
<td>Miller</td>
<td>2.75</td>
</tr>
<tr>
<td>Cricket’s</td>
<td>Bud</td>
<td>2.50</td>
</tr>
<tr>
<td>Cricket’s</td>
<td>Miller</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Prices := PROJ_{beer,price}(Sells):

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>beer</td>
<td>price</td>
</tr>
<tr>
<td>Bud</td>
<td>2.50</td>
</tr>
<tr>
<td>Miller</td>
<td>2.75</td>
</tr>
<tr>
<td>Miller</td>
<td>3.00</td>
</tr>
</tbody>
</table>
Select Operator

• The **select** operator is unary
  – It outputs a new relation that has a subset of tuples
  – A condition specifies those tuples that are required

Relation Sells:

<table>
<thead>
<tr>
<th>bar</th>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimy’s Bud</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td>Chimy’s Miller</td>
<td>2.75</td>
<td></td>
</tr>
<tr>
<td>Cricket’s Bud</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td>Cricket’s Miller</td>
<td>3.00</td>
<td></td>
</tr>
</tbody>
</table>

ChimyMenu := SELECT_{bar = “Chimy’s”}(Sells):

<table>
<thead>
<tr>
<th>bar</th>
<th>beer</th>
<th>price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimy’s Bud</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td>Chimy’s Miller</td>
<td>2.75</td>
<td></td>
</tr>
</tbody>
</table>
Join Operator

- The **join** operator is binary
  - It outputs the combined relation where tuples agree on a specified attribute (natural join)

<table>
<thead>
<tr>
<th>Sells(bar, beer, price)</th>
<th>Bars(bar, address)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimy’s Bud 2.50</td>
<td>Chimy’s 2417 Broadway St.</td>
</tr>
<tr>
<td>Chimy’s Miller 2.75</td>
<td>Cricket’s 2412 Broadway St.</td>
</tr>
<tr>
<td>Cricket’s Bud 2.50</td>
<td></td>
</tr>
<tr>
<td>Cricket’s Coors 3.00</td>
<td></td>
</tr>
</tbody>
</table>

BarInfo := Sells JOIN Bars
Note Bars.name has become Bars.bar to make the natural join “work.”

<table>
<thead>
<tr>
<th>BarInfo(bar, beer, price, address)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimy’s Bud 2.50 2417 Broadway St.</td>
</tr>
<tr>
<td>Chimy’s Miller 2.75 2417 Broadway St.</td>
</tr>
<tr>
<td>Cricket’s Bud 2.50 2412 Broadway St.</td>
</tr>
<tr>
<td>Cricket’s Coors 3.00 2412 Broadway St.</td>
</tr>
</tbody>
</table>
Join Operator

• Join is the most time-consuming of all relational operators to compute
  – In general, relational operators may not be arbitrarily reordered (left join, right join)
  – Query optimization aims to find an efficient way of processing queries, for example reordering to produce equivalent but more efficient queries
<table>
<thead>
<tr>
<th>FID</th>
<th>Shape</th>
<th>STATION</th>
<th>MAXDAY</th>
<th>AV8TOP</th>
<th>MONITOR</th>
<th>LAT</th>
<th>LON</th>
<th>X_COORD</th>
<th>Y_COORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Point</td>
<td>60</td>
<td>16</td>
<td>7.225806</td>
<td>70060</td>
<td>34.135833</td>
<td>-117.92361</td>
<td>-117.92361</td>
<td>34.13583</td>
</tr>
<tr>
<td>1</td>
<td>Point</td>
<td>69</td>
<td>9</td>
<td>5.899194</td>
<td>70069</td>
<td>34.176111</td>
<td>-118.31528</td>
<td>-118.31528</td>
<td>34.17611</td>
</tr>
<tr>
<td>2</td>
<td>Point</td>
<td>72</td>
<td>13</td>
<td>4.052885</td>
<td>70072</td>
<td>33.823611</td>
<td>-118.1875</td>
<td>-118.1875</td>
<td>33.82361</td>
</tr>
<tr>
<td>3</td>
<td>Point</td>
<td>74</td>
<td>13</td>
<td>7.181452</td>
<td>70074</td>
<td>34.199444</td>
<td>-118.53472</td>
<td>-118.53472</td>
<td>34.19944</td>
</tr>
<tr>
<td>4</td>
<td>Point</td>
<td>75</td>
<td>8</td>
<td>6.076613</td>
<td>70075</td>
<td>34.669944</td>
<td>-117.75139</td>
<td>-117.75139</td>
<td>34.66994</td>
</tr>
<tr>
<td>5</td>
<td>Point</td>
<td>84</td>
<td>8</td>
<td>3.157258</td>
<td>70084</td>
<td>33.929167</td>
<td>-118.20972</td>
<td>-118.20972</td>
<td>33.92917</td>
</tr>
<tr>
<td>6</td>
<td>Point</td>
<td>85</td>
<td>11</td>
<td>5.201613</td>
<td>70085</td>
<td>34.015</td>
<td>-118.05972</td>
<td>-118.05972</td>
<td>34.015</td>
</tr>
<tr>
<td>7</td>
<td>Point</td>
<td>87</td>
<td>9</td>
<td>4.717742</td>
<td>70087</td>
<td>34.067222</td>
<td>-118.22639</td>
<td>-118.22639</td>
<td>34.06722</td>
</tr>
<tr>
<td>8</td>
<td>Point</td>
<td>88</td>
<td>8</td>
<td>6.532258</td>
<td>70088</td>
<td>34.038333</td>
<td>-118.10694</td>
<td>-118.10694</td>
<td>34.03833</td>
</tr>
<tr>
<td>9</td>
<td>Point</td>
<td>89</td>
<td>13</td>
<td>7.540323</td>
<td>70089</td>
<td>34.3875</td>
<td>-118.53472</td>
<td>-118.53472</td>
<td>34.3875</td>
</tr>
<tr>
<td>10</td>
<td>Point</td>
<td>91</td>
<td>8</td>
<td>4.891129</td>
<td>70091</td>
<td>34.050833</td>
<td>-118.45417</td>
<td>-118.45417</td>
<td>34.05083</td>
</tr>
<tr>
<td>11</td>
<td>Point</td>
<td>94</td>
<td>8</td>
<td>4.14914</td>
<td>70094</td>
<td>33.92357</td>
<td>-118.37085</td>
<td>-118.37085</td>
<td>33.92357</td>
</tr>
<tr>
<td>12</td>
<td>Point</td>
<td>96</td>
<td>7</td>
<td>7.266129</td>
<td>70096</td>
<td>34.69012</td>
<td>-118.1334</td>
<td>-118.1334</td>
<td>34.69012</td>
</tr>
<tr>
<td>13</td>
<td>Point</td>
<td>3176</td>
<td>9</td>
<td>4.100806</td>
<td>30176</td>
<td>33.820278</td>
<td>-117.9125</td>
<td>-117.9125</td>
<td>33.82028</td>
</tr>
<tr>
<td>14</td>
<td>Point</td>
<td>3177</td>
<td>10</td>
<td>4.737903</td>
<td>30177</td>
<td>33.926111</td>
<td>-117.95139</td>
<td>-117.95139</td>
<td>33.92611</td>
</tr>
<tr>
<td>15</td>
<td>Point</td>
<td>3186</td>
<td>9</td>
<td>4.600086</td>
<td>30186</td>
<td>33.6275</td>
<td>-117.69028</td>
<td>-117.69028</td>
<td>33.6275</td>
</tr>
<tr>
<td>16</td>
<td>Point</td>
<td>3195</td>
<td>8</td>
<td>4.358871</td>
<td>30195</td>
<td>33.675711</td>
<td>-117.9269</td>
<td>-117.9269</td>
<td>33.67571</td>
</tr>
<tr>
<td>17</td>
<td>Point</td>
<td>4137</td>
<td>13</td>
<td>8.241935</td>
<td>33137</td>
<td>33.38</td>
<td>-116.54306</td>
<td>-116.54306</td>
<td>33.38</td>
</tr>
<tr>
<td>18</td>
<td>Point</td>
<td>4143</td>
<td>17</td>
<td>8.974355</td>
<td>33144</td>
<td>34.01028</td>
<td>-117.42639</td>
<td>-117.42639</td>
<td>34.01028</td>
</tr>
<tr>
<td>19</td>
<td>Point</td>
<td>4149</td>
<td>8</td>
<td>8.899194</td>
<td>33149</td>
<td>33.708333</td>
<td>-117.24306</td>
<td>-117.24306</td>
<td>33.70833</td>
</tr>
<tr>
<td>20</td>
<td>Point</td>
<td>4150</td>
<td>14</td>
<td>7.16129</td>
<td>33150</td>
<td>33.925</td>
<td>-116.87639</td>
<td>-116.87639</td>
<td>33.925</td>
</tr>
<tr>
<td>21</td>
<td>Point</td>
<td>4157</td>
<td>17</td>
<td>8.133065</td>
<td>33157</td>
<td>33.714555</td>
<td>-116.23389</td>
<td>-116.23389</td>
<td>33.71455</td>
</tr>
<tr>
<td>22</td>
<td>Point</td>
<td>4158</td>
<td>7</td>
<td>7.858871</td>
<td>33158</td>
<td>33.67401</td>
<td>-117.32114</td>
<td>-117.32114</td>
<td>33.67401</td>
</tr>
<tr>
<td>23</td>
<td>Point</td>
<td>4162</td>
<td>17</td>
<td>7.962097</td>
<td>33162</td>
<td>33.97562</td>
<td>-117.33263</td>
<td>-117.33263</td>
<td>33.97562</td>
</tr>
<tr>
<td>24</td>
<td>Point</td>
<td>5175</td>
<td>8</td>
<td>8.14914</td>
<td>33175</td>
<td>34.10361</td>
<td>-117.62917</td>
<td>-117.62917</td>
<td>34.10361</td>
</tr>
<tr>
<td>25</td>
<td>Point</td>
<td>5181</td>
<td>22</td>
<td>11.64914</td>
<td>33181</td>
<td>34.243889</td>
<td>-117.27361</td>
<td>-117.27361</td>
<td>34.24389</td>
</tr>
<tr>
<td>26</td>
<td>Point</td>
<td>5197</td>
<td>18</td>
<td>8.604839</td>
<td>33197</td>
<td>34.09447</td>
<td>-117.50417</td>
<td>-117.50417</td>
<td>34.09447</td>
</tr>
<tr>
<td>27</td>
<td>Point</td>
<td>5203</td>
<td>20</td>
<td>9.947581</td>
<td>33203</td>
<td>34.107222</td>
<td>-117.27361</td>
<td>-117.27361</td>
<td>34.10722</td>
</tr>
<tr>
<td>28</td>
<td>Point</td>
<td>5204</td>
<td>20</td>
<td>10.27419</td>
<td>33204</td>
<td>34.066667</td>
<td>-117.15139</td>
<td>-117.15139</td>
<td>34.06667</td>
</tr>
<tr>
<td>29</td>
<td>Point</td>
<td>5212</td>
<td>9</td>
<td>8.524194</td>
<td>33212</td>
<td>33.89484</td>
<td>-117.51237</td>
<td>-117.51237</td>
<td>33.89484</td>
</tr>
<tr>
<td>30</td>
<td>Point</td>
<td>5213</td>
<td>9</td>
<td>4.447917</td>
<td>33213</td>
<td>34.262276</td>
<td>-117.18854</td>
<td>-117.18854</td>
<td>34.26228</td>
</tr>
<tr>
<td>31</td>
<td>Point</td>
<td>591</td>
<td>18</td>
<td>8.241935</td>
<td>70591</td>
<td>34.143889</td>
<td>-117.85139</td>
<td>-117.85139</td>
<td>34.14389</td>
</tr>
</tbody>
</table>

SELECT * FROM oz96 WHERE "MAXDAY">=9;
Example: GIS analysis

• For Italy’s capital city, Rome, calculate the total length of the River Tiber which lies within 2.5 km of the Colosseum
  – First we need to model the relevant parts of Rome as objects
  – Operation *length* will act on *arc*, and *intersect* will apply to form the piece of the *arc* in common with the disc
Example: GIS analysis

- A process of discretization must convert the objects to types that are computationally tractable
- A circle may be represented as a discrete polygonal area, arcs by chains of line segments, and points may be embedded in some discrete space
Primitive Objects

• *Euclidean Space*: coordinatized model of space
  – Transforms spatial properties into properties of tuples of real numbers
  – Coordinate frame consists of a fixed, distinguished point (origin) and a pair of orthogonal lines (axes), intersecting in the origin

• Point objects
• Line objects
• Polygonal objects
Polygonal objects

- Polyline
- Simple closed polyline
- Polygon
- Convex polygon
- Star-shaped polygon
Convexity

Visibility between points $x$, $y$, and $z$
Example: Triangulation

• Every simple polygon has a triangulation. Any triangulation of a simple polygon with $n$ vertices consists of exactly $n - 2$ triangles.

• Art Gallery Problem
  – How many cameras are needed to guard a gallery and how should they be placed?
  – Upper bound $N/3$
Related: Convex Hull
Related: Voronoi Diagram
Voronoi Diagram on Road Network
John Snow, Pumps and Cholera Outbreak
Primitive GIS Operations

- in Euclidean spaces
  - Length, bearing, area
    - How many ways you can think of to calculate the area of a polygon?
    - How to test which side of a point corresponding to a line?
  - Distance between objects (points, lines, polygons)
    - Distance could be ambiguous, e.g., what is the difference from Lubbock to Dallas (from city center or city boundary?).
  - Centroid
    - Not necessarily within the boundary of polygon
  - Point in polygon
    - Ray casting method
  - Point on line
    - area
  - Buffer
  - Intersection/overlay
- In topological spaces
  - Spatial relations (within, touch, cover, ...)

Distance and angle between points

- Length of a line segment can be computed as the distance between successive pairs of points

\[ |pq| = \sqrt{(x_q - x_p)^2 + (y_q - y_p)^2} \]

- The bearing, \( \theta \), of \( q \) from \( p \) is given by the unique solution in the interval \([0,360]\) of the simultaneous equations:

\[
\cos \theta = \frac{y_q - y_p}{|pq|} \quad \text{and} \quad \sin \theta = \frac{x_q - x_p}{|pq|}
\]
Area

• Let $P$ be a simple polygon (no boundary self-intersections) with vertex vectors: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where $(x_1, y_1) = (x_n, y_n)$. Then the area is:

$$\text{area}(P) = \frac{1}{2} \sum_{i=1}^{n-1} x_i y_{i+1} - x_{i+1} y_i$$

• In the case of a triangle $pqr$

$$\text{area}(pqr) = \frac{xy_q - x_y p + x_q y_r - x_r y_q + x_r y_p - x_p y_r}{2}$$
Area of a simple polygon

• Note that the area may be positive or negative
• In fact, $\text{area}(pqr) = -\text{area}(qpr)$
• If $p$ is to the left of $qr$ then the area is positive, if $p$ is to the right of $qr$ then the area is negative

$$\text{side}(p, q, r) = \begin{cases} 
1 & \text{if } \text{area}(pqr) > 0 \quad (p \text{ is left of } qr) \\
0 & \text{if } \text{area}(pqr) = 0 \quad (pqr \text{ are collinear}) \\
-1 & \text{if } \text{area}(pqr) < 0 \quad (p \text{ is right of } qr) 
\end{cases}$$
Point in polygon

- Determining whether a point is inside a polygon is one of the most fundamental operations in a spatial database
- **Semi-line method (ray casting)**: checks for odd or even numbers of intersections of a semi-line with polygon
- **Winding method**: sums bearings from point to polygon vertices
Is Biloxi in Mississippi?
1 intersection
= Hey Y'all!

Is New Orleans in Mississippi?
2 Intersections
= Reject-shon!
Primitive GIS operations: Overlay

- Union
- Intersect
- Erase
- Identity
- Update
- Spatial Join
- Symmetrical Difference
Overlay

• Union
  
  — Computes a geometric union of the input features. All features and their attributes will be written to the output feature class.
Overlay

- **Intersect**
  - Computes a geometric intersection of the input features. Features or portions of features which overlap in all layers and/or feature classes will be written to the output feature class.
Overlay

• Erase
  – Creates a feature class by overlaying the Input Features with the polygons of the Erase Features. Only those portions of the input features falling outside the erase features outside boundaries are copied to the output feature class
Overlay

• Identity
  – Computes a geometric intersection of the input features and identity features. The input features or portions thereof that overlap identity features will get the attributes of those identity features
Overlay

• Update
  
  – Computes a geometric intersection of the Input Features and Update Features. The attributes and geometry of the input features are updated by the update features in the output feature class.
Overlay

• Symmetrical difference
  
  – Features or portions of features in the input and update features that do not overlap will be written to the output feature class.
Overlay

• Spatial join
  - Joins attributes from one feature to another based on the spatial relationship. The target features and the joined attributes from the join features are written to the output feature class.
Which of the following is the result of identity, intersect, symmetrical difference, union and update respectively?

A  B  C  D  E
Buffer

Vector buffers

points
lines
polygons

input source features

output buffer features
• Primitive operators
  – you might already realized that these primitive operators are often used collaboratively with each other, and other analytical methods (e.g., dissolve, surface analysis, interpolation) that we will introduce in the coming lectures.
Topological spatial operations: spatial relationship

• Object types with an assumed underlying topology are **point, arc, loop** and **area**

• Operations:
  
  – **boundary, interior, closure** and **connected** are defined in the usual manner
  
  – **components** returns the set of maximal connected components of an area
  
  – **extremes** acts on each object of type arc and returns the pair of points of the arc that constitute its end points
  
  – **is within** provides a relationship between a point and a simple loop, returning true if the point is enclosed by the loop
Topological spatial operations for areas

- $X$ *meets* $Y$ if $X$ and $Y$ touch externally in a common portion of their boundaries

- $X$ *overlaps* $Y$ if $X$ and $Y$ impinge into each other’s interiors

$X$ is not *disjoint from* $Y$
Topological spatial operations for areas

- $X$ is inside $Y$ if $X$ is a subset of $Y$ and $X$, $Y$ do not share a common portion of boundary

- $X$ covers $Y$ if $Y$ is a subset of $X$ and $X$, $Y$ touch externally in a common portion of their boundaries
Topological spatial operations

- There are an infinite number of possible topological relationships that are available between objects of type cell.
Match Option (optional)

- CONTAINS_CLEMENTINI
- INTERSECT
- INTERSECT_3D
- WITHIN_A_DISTANCE
- WITHIN_A_DISTANCE_3D
- CONTAINS
- COMPLETELY_CONTAINS
- CONTAINS_CLEMENTINI
- WITHIN
- COMPLETELY_WITHIN
- WITHIN_CLEMENTINI
- ARE_IDENTICAL_TO
- BOUNDARY_TOUCHES
- SHARE_A_LINE_SEGMENT_WITH
- CROSSED_BY_THE_OUTLINE_OF
- HAVE_THEIR_CENTER_IN
- CLOSEST

• Contain vs CONTAINS_CLEMENTINI:
  – the results of CONTAINS_CLEMENTINI will be identical to CONTAINS with the exception that if the feature in the Selecting Features layer is entirely on the boundary of the Input Feature Layer, with no part of the contained feature properly inside the feature in the Input Feature Layer, the input feature will not be selected.
Spaghetti

- Spaghetti data structure represents a planar configuration of points, arcs, and areas
- Geometry is represented as a set of lists of straight-line segments
Spaghetti- example

• Each polygonal area is represented by its boundary loop
• Each loop is discretized as a closed polyline
• Each polyline is represented as a list of points

A: [1, 2, 3, 4, 21, 22, 23, 26, 27, 28, 20, 19, 18, 17]
B: [4, 5, 6, 7, 8, 25, 24, 23, 22, 21]
C: [8, 9, 10, 11, 12, 13, 29, 28, 27, 26, 23, 24, 25]
D: [17, 18, 19, 20, 28, 29, 13, 14, 15, 16]
Issues

• There is **NO** explicit representation of the topological interrelationships of the configuration, such as adjacency

• Data consistence issues
  – *Silver polygons*
  – *Data redundancy*
NAA: node arc area

• Each directed arc has exactly one start and one end node.
• Each node must be the start node or end node (maybe both) of at least one directed arc.
• Each area is bounded by one or more directed arcs.
• Directed arcs may intersect only at their end nodes.
• Each directed arc has exactly one area on its right and one area on its left.
• Each area must be the left area or right area (maybe both) of at least one directed arc.
NAA: planar decomposition

<table>
<thead>
<tr>
<th>Arc</th>
<th>Begin</th>
<th>End</th>
<th>Left</th>
<th>Right</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>2</td>
<td>A</td>
<td>X</td>
</tr>
<tr>
<td>b</td>
<td>4</td>
<td>1</td>
<td>B</td>
<td>X</td>
</tr>
<tr>
<td>c</td>
<td>3</td>
<td>4</td>
<td>C</td>
<td>X</td>
</tr>
<tr>
<td>d</td>
<td>2</td>
<td>3</td>
<td>D</td>
<td>X</td>
</tr>
<tr>
<td>e</td>
<td>5</td>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>f</td>
<td>4</td>
<td>5</td>
<td>C</td>
<td>B</td>
</tr>
<tr>
<td>g</td>
<td>6</td>
<td>2</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>h</td>
<td>5</td>
<td>6</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>i</td>
<td>3</td>
<td>6</td>
<td>D</td>
<td>C</td>
</tr>
</tbody>
</table>
• End of this topic