INTRODUCTION TO
GRAPH DRAWING

Fall 2010

Graph Drawing: Algorithms for the Visualization of Graphs. 1st.
Prentice Hall PTR.
Tessellation And Visibility Representation of Graphs
**Tile**

- A tile is a *rectangle* with sides parallel to the *coordinate axes*.
- A tile can be *unbounded* or can *degenerate* to a segment or a point.
- Two tiles are horizontally (vertically) *adjacent* if they share a portion of a vertical (horizontal) side.

Tiles: 🟢 | 🟠 | 🟡  
Vertically adjacent:
Let $G$ be a planar st-graph. A tessellation representation $\Theta$ for $G$ maps each object $o$ of $G$ into a tile $\Theta(o)$ such that:

- The interiors of tiles $\Theta(o_1)$ and $\Theta(o_2)$ are disjoint whenever $o_1 \neq o_2$.
- The union of all tiles $\Theta(o)$, $o \in V \cup E \cup F$, is a rectangle.
- Tiles $\Theta(o_1)$ and $\Theta(o_2)$ are horizontally adjacent if and only if $o_1 = \text{left}(o_2)$ or $o_1 = \text{right}(o_2)$ or $o_2 = \text{left}(o_1)$ or $o_2 = \text{right}(o_1)$.
- Tiles $\Theta(o_1)$ and $\Theta(o_2)$ are vertically adjacent if and only if $o_1 = \text{orig}(o_2)$ or $o_1 = \text{dest}(o_2)$ or $o_2 = \text{orig}(o_1)$ or $o_2 = \text{dest}(o_1)$. 
Algorithm Tessellation:

- Input: a planer st-graph.
- Output: a tessellation representation $\Theta$ for $G$ such that each vertex and face is a segment.

1. Construct planar st-graph $G^*$
2. Compute a topological numbering $Y$ of $G$
3. Compute a topological numbering $X$ of $G^*$
4. For each object $o \in V \cup E \cup F$, let the coordinates of tile $\Theta(o)$ be:
   \[
   x_L(o) = X(left(o)); \\
   x_R(o) = X(right(o)); \\
   y_B(o) = X(orig(o)); \\
   y_T(o) = X(dest(o));
   \]
**Planar Graph**

- A **planar graph** is a graph which can be embedded in the plane, i.e., it can be drawn on the plane in such a way that its edges intersect only at their endpoints.
ST-Graph

- An acyclic digraph with a **single source** $s$ and a **single sink** $t$ is called an st-graph.

Sink “$t$” has no outgoing edges!

Source “$s$” has no incoming edges!
A planar st-graph is an st-graph that is planar and embedded with vertices $s$ and $t$ on the boundary of the external face.
Planar ST-Graph

Let $G$ be a planer st-graph and $F$ be its set of faces. We conventionally assume that $F$ contains two representatives for the external face:

- The "left external face" $s^*$
- The "right external face" $t^*$
**Planar ST-Graph**

- For each edge $e = (u, v)$, we define $\text{orig}(e) = u$ and $\text{dest}(e) = v$.
- We define $\text{left}(e)$ (resp. $\text{right}(e)$) to be the face to the left (resp. right) of $e$. 

\begin{align*}
\text{dest}(e) &= v \\
\text{orig}(e) &= u \\
\text{left}(e) &= f_1 \\
\text{left}(e) &= f_2
\end{align*}
**Lemma 1:** Each face $f$ of $G$ consists of two directed paths with common origin called $\text{orig}(f)$, and common destination called $\text{dest}(f)$.

**Proof:** Let $f$ be a face of $G$ for which the lemma is not true.
**Lemma 2:** The incoming edges for each vertex of G appear *consecutively* around v, and so do the outgoing edges.

**Proof:** The lemma holds trivially for the vertices s and t. Let v be any *other* vertex, and suppose for a contradiction, that there are edges \((v,w_0), (w_1,v), (v,w_2), (w_3,v)\).
Since the incoming (outgoing) edges of each vertex of $G$ appear \textbf{consecutively} we define the face separating the incoming edges from the outgoing edges in clockwise order, $left(v)$, and the other separating face is called $right(v)$. 

\[left(v) \quad right(v)\]
**Planar ST-Graph G***

- We define a digraph $G^*$ associated with planar st-graph $G$, as follows:
  - The vertex set of $G^*$ is the set $F$ of faces (recall that $F$ has two representatives, $s^*$ and $t^*$, of the external face).
  - For every edge $e \neq (s,t)$ of $G$, $G^*$ has an $e^* = (f,g)$ where $f = \text{left}(e)$ and $g = \text{right}(e)$.

Notice that $G^*$ is a planar st-graph as well.
• An element of $V \cup E \cup F$ is called an object of planar st-graph G.

• For vertex $v$, we define $\text{orig}(v) = \text{dest}(v) = v$.

• For a face $f$, we define $\text{left}(f) = \text{right}(f) = f$.

* Reminder we have already defined previously: $\text{left}(v)$, $\text{right}(v)$, $\text{left}(e)$, $\text{right}(e)$, $\text{orig}(e)$, $\text{dest}(e)$, $\text{orig}(f)$ and $\text{dest}(f)$. 
LEMMA

- For any two objects $o_1$ and $o_2$ of a planar st-graph $G$, exactly one of the following holds:
  - $G$ has a directed path from $\text{dest}(o_1)$ to $\text{orig}(o_2)$.
  - $G$ has a directed path from $\text{dest}(o_2)$ to $\text{orig}(o_1)$.
  - $G^*$ has a directed path from $\text{right}(o_1)$ to $\text{left}(o_2)$.
  - $G^*$ has a directed path from $\text{right}(o_2)$ to $\text{left}(o_2)$. 
**Tessellation Representation**

- **Algorithm Tessellation:**
  - **Input:** a planer st-graph.
  - **Output:** a tessellation representation $\Theta$ for $G$ such that each vertex and face is a segment.
  1. Construct **planar st-graph $G^*$**
  2. Compute a **topological numbering $Y$ of $G$**
  3. Compute a **topological numbering $X$ of $G^*$**
  4. For each object $o \in V \cup E \cup F$, let the coordinates of tile $\Theta(o)$ be:
     - $x_L(o) = X(left(o))$
     - $x_R(o) = X(right(o))$
     - $y_B(o) = X(orig(o))$
     - $y_T(o) = X(dest(o))$
NUMBERING OF DIGRAPHS

- A **topological numbering** of $G$ is an assignment of numbers to the **vertices** of $G$, such that, for every edge $(u,v)$ of $G$, the number assigned to $v$ is **greater** than the one assigned to $u$. 

![Diagram of numbered digraph](image)
NUMBERING OF DIGRAPHS

- A topological sorting is a topological numbering of G, such that every vertex is assigned a distinct integer between 1 and n.
- A topological sorting is unique if G has a directed path that visits every vertex.
NUMBERING OF DIGRAPHS

- The following statements are equivalent:
  - G is acyclic.
  - G admits a topological numbering.
  - G admits a topological sorting.

In other words a topological numbering (sorting) can be done only on an *acyclic* graph.
ST-GRAPH

• Let G be an st-graph. The following simple properties hold:

  • Given a topological numbering of G, every directed path of G visits with increasing numbers.

  • For every vertex v of G, there exists a simple directed path from s (source) to t (sink) that contains v.
**Tessellation Representation**

- Algorithm **Tessellation**:
  - **Input**: a planer st-graph.
  - **Output**: a tessellation representation $\Theta$ for $G$ such that each vertex and face is a segment.

1. Construct planar st-graph $G^*$
2. Compute a topological numbering $Y$ of $G$
3. Compute a topological numbering $X$ of $G^*$
4. For each object $o \in V \cup E \cup F$, let the coordinates of tile $\Theta(o)$ be:
   
   \[
   \begin{align*}
   x_L(o) &= X(left(o)); \\
   x_R(o) &= X(right(o)); \\
   y_B(o) &= X(orig(o)); \\
   y_T(o) &= X(dest(o));
   \end{align*}
   \]

**Example**

- $X(left(e)) = 0$
- $X(right(e)) = 1$
- $Y(orig(e)) = 0$
- $Y(dest(e)) = 2$
Tessellation Representation

- Algorithm **Tessellation**:
  - Input: a planer st-graph.
  - Output: a tessellation representation $\Theta$ for $G$ such that each vertex and face is a segment.
  1. Construct planar st-graph $G^*$
  2. Compute a topological numbering $Y$ of $G$
  3. Compute a topological numbering $X$ of $G^*$
  4. For each object $o \in V \cup E \cup F$, let the coordinates of tile $\Theta(o)$ be:
     - $x_L(o) = X(left(o))$
     - $x_R(o) = X(right(o))$
     - $y_B(o) = X(ori(o))$
     - $y_T(o) = X(dest(o))$

$$X(left(e)) = 0$$
$$X(right(e)) = 1$$
$$Y(ori(e)) = 2$$
$$Y(dest(e)) = 4$$
Algorithm **Tessellation**:
- **Input:** a planer st-graph.
- **Output:** a tessellation representation \( \Theta \) for \( G \) such that each vertex and face is a segment.
1. Construct planar st-graph \( G^* \)
2. Compute a topological numbering \( Y \) of \( G \)
3. Compute a topological numbering \( X \) of \( G^* \)
4. For each object \( o \in V \cup E \cup F \), let the coordinates of tile \( \Theta(o) \) be:
   - \( x_L(o) = X(left(o)) \)
   - \( x_R(o) = X(right(o)) \)
   - \( y_B(o) = X(ori(o)) \)
   - \( y_T(o) = X(dest(o)) \)
MODIFICATION TO SUPPORT USER DEFINED CONSTRAINTS ON THE SIZE OF THE EDGE TILES:

- Algorithm Tessellation:
  - Input: a planer st-graph.
  - Output: a tessellation representation Θ for G such that each vertex and face is a segment.
  1. Construct planar st-graph G*
  2. Compute a topological numbering Y of G
  3. Compute a topological numbering X of G*
  4. For each object \( o \in V \cup E \cup F \), let the coordinates of tile Θ(o) be:
     \[
     x_L(o) = X(left(o));
     x_R(o) = X(right(o));
     y_B(o) = X(orig(o));
     y_T(o) = X(dest(o));
     \]

- Algorithm Tessellation:
  - Input: a planer st-graph.
  - Output: a tessellation representation Θ for G such that each vertex and face is a segment.
  1. Construct planar st-graph G*
  2. Assign weight \( h(e) \) to each edge \( e \) of G and compute an optimal weighted topological numbering Y of G
  3. Assign weight \( w(e) \) to each edge \( e^* \) of G* and compute an optimal weighted topological numbering X of G*
  4. For each object \( o \in V \cup E \cup F \), let the coordinates of tile Θ(o) be:
     \[
     x_L(o) = X(left(o));
     x_R(o) = X(right(o));
     y_B(o) = X(orig(o));
     y_T(o) = X(dest(o));
     \]
A weighted topological numbering is a topological numbering of G, such that for every edge $e=(u,v)$ of G the number assigned to $v$ is greater than or equal to the number assigned to $u$ plus the weight of $(u,v)$.

$$\text{number}(v) \geq \text{number}(u) + \text{weight}(u,v)$$
Tessellation Representation

- The correctness of the algorithm is based on Lemma 3:
  - Let there be tile $t_1$ and tile $t_2$, from Lemma 3 $t_1$ is either: above $t_2$, below $t_2$, left of $t_2$ or right of $t_2$. And only one of this directions is true.
- Since each line of the algorithm is $O(n)$, the total runtime of the algorithm is $O(n)$.
- The size of the Tessellation Representation can be modified by modifying the topological numbering (e.g. increasing the numbering to be 0..2..4 instead of 0..1..2 will make a Tessellation Representation twice bigger).
VISIBILITY REPRESENTATION

Let $G$ be a planar st-graph. A visibility representation $\Gamma$ of $G$ draws each vertex $v$ as a horizontal segment, called vertex segment $\Gamma(v)$, and each edge $(u,v)$ as vertical segment, called edge segment $\Gamma(u,v)$ such that:

- The vertex segments do not overlap.
- The edge segments do not overlap.
- Edge-segment $\Gamma(u,v)$ has its bottom end point on $\Gamma(u)$, its top end-point on $\Gamma(v)$, and does not intersect any other vertex segment.
VISIBILITY REPRESENTATION

- Algorithm Visibility
  - Input: a planar st-graph G with n vertices.
  - Output: visibility representation \( \Gamma \) of G with integer coordinates and area \( O(n^2) \)
  1. Construct planar st-graph \( G^* \)
  2. Assign unit weights to edges of G and compute an optimal weighted topological numbering \( Y \) of G
  3. Assign unit weights to edges of \( G^* \) and compute an optimal weighted topological numbering \( X \) of \( G^* \)
  4. For each vertex \( v \), draw the vertex-segment \( \Gamma(v) \) at y-coordinate \( Y(v) \) and between x-coordinates \( X(left(v)) \) and \( X(right(v) - 1) \).

    for each vertex \( v \) do
    
    draw \( \Gamma(v) \) as the horizontal segment with
    
    \[ y(\Gamma(v)) = Y(v); \]
    \[ x_L(\Gamma(v)) = X(left(v)); \]
    \[ x_R(\Gamma(v)) = X(right(v) - 1); \]

    endfor

  5. For each edge \( e \), draw the edge-segment \( \Gamma(e) \) at x-coordinate \( X(left(e)) \), between y-coordinates \( Y(orig(e)) \) and \( Y(dest(e) - 1) \).

    for each edge \( e \) do
    
    draw \( \Gamma(e) \) as the vertical segment with
    
    \[ x(\Gamma(e)) = X(left(e)); \]
    \[ y_B(\Gamma(e)) = Y(orig(e)); \]
    \[ y_T(\Gamma(e)) = Y(dest(e)); \]

    endfor


**VISIBILITY REPRESENTATION**

4. For each vertex $v$, draw the vertex-segment $\Gamma(v)$ at y-coordinate $Y(v)$ and between x-coordinates $X(left(v))$ and $X(right(v)-1)$.

   for each vertex $v$ do
   
   draw $\Gamma(v)$ as the horizontal segment with
   
   $y(\Gamma(v)) = Y(v)$;
   
   $x_L(\Gamma(v)) = X(left(v))$;
   
   $x_R(\Gamma(v)) = X(right(v)-1)$;

   endfor
**Visibility Representation**

5. For each edge $e$, draw the edge-segment $\Gamma(e)$ at x-coordinate $X(left(e))$, between y-coordinates $Y(orig(e))$ and $Y(dest(e)-1)$.

   for each edge $e$ do
   
   draw $\Gamma(e)$ as the vertical segment with

   $x(\Gamma(e)) = X(left(e))$;

   $y_B(\Gamma(e)) = Y(orig(e))$;

   $y_T(\Gamma(e)) = Y(dest(e))$;

   endfor
VISIBILITY REPRESENTATION

• The correctness of the algorithm By lemma 3 and the construction of the algorithm:
  • Any two **vertex segments** are separated by a horizontal or vertical strip of at least unit width (The vertex segments do not overlap).
  • Any two edge segments on opposite sides of a face are separated by a vertical strip of at least a unit width (The edge segments do not overlap).
  • Each edge segments (u,v) has its bottom point intersecting with u vertex segment, and his upper point intersecting with v vertex segment (sufficing the 3rd condition).

• The runtime of the algorithm is O(n) since each step is O(n).
• The area of the representation is O(n^2)