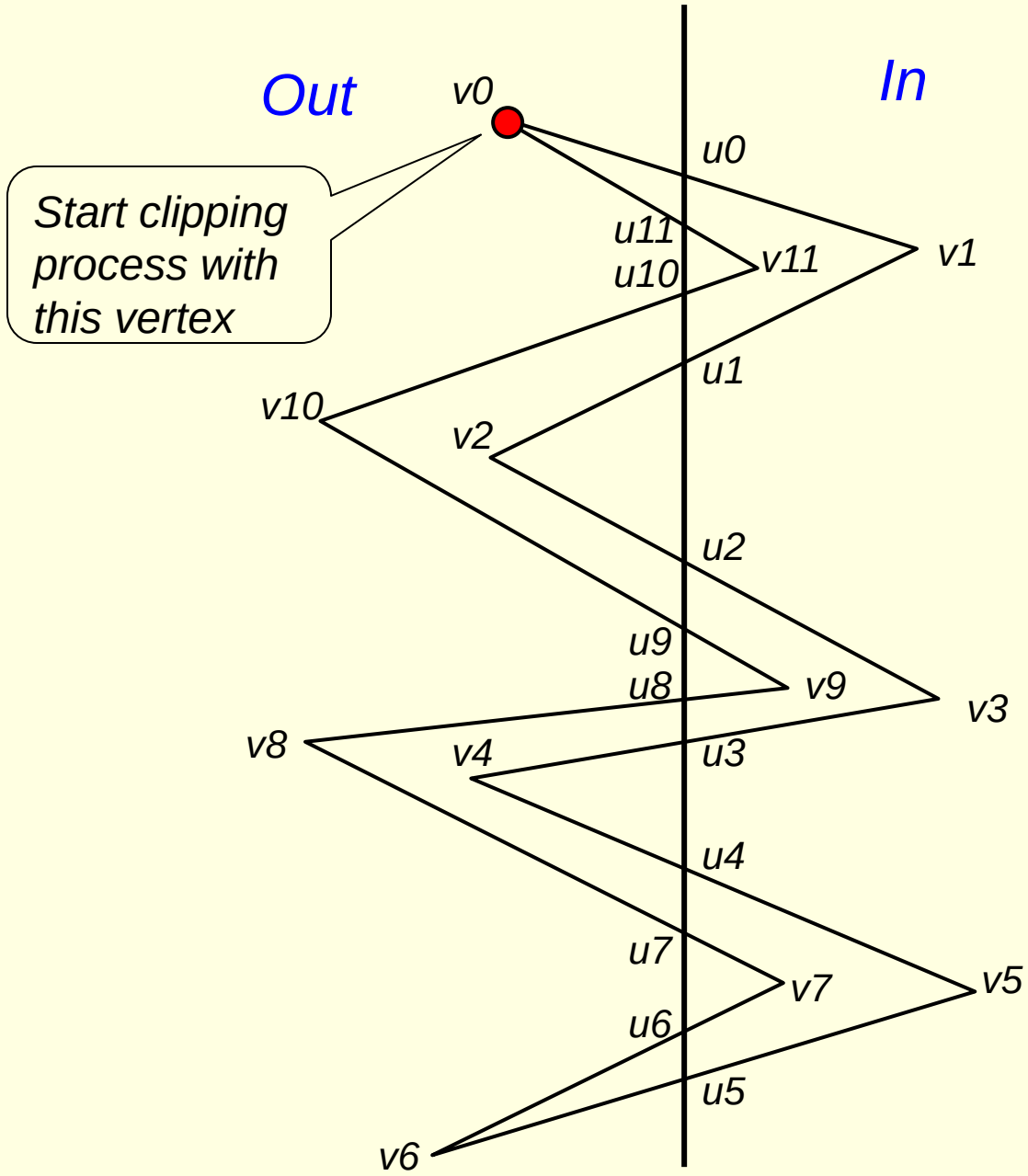
Polygon Clipping I

1. Always start the clipping process with the vertex that goes from outside to inside and has the largest (or, smallest) y-coordinate if the clipping edge is vertical (see the next slide for such a vertex). Otherwise start with the vertex that goes from outside to inside and has the largest (or, smallest) x-coordinate. (Totally, four cases)

2. Each time you move from an "In" vertex to an "out" vertex, compute the intersection point, append the intersection point to the current vertex list and then output the current vertex list. So for the following polygon, you would get the following output vertex lists.



P1: u0→v1→u1

P2: u2→v3→u3

P3: u4→v5→u5

P4: u6→v7→u7

P5: u8→v9→u9

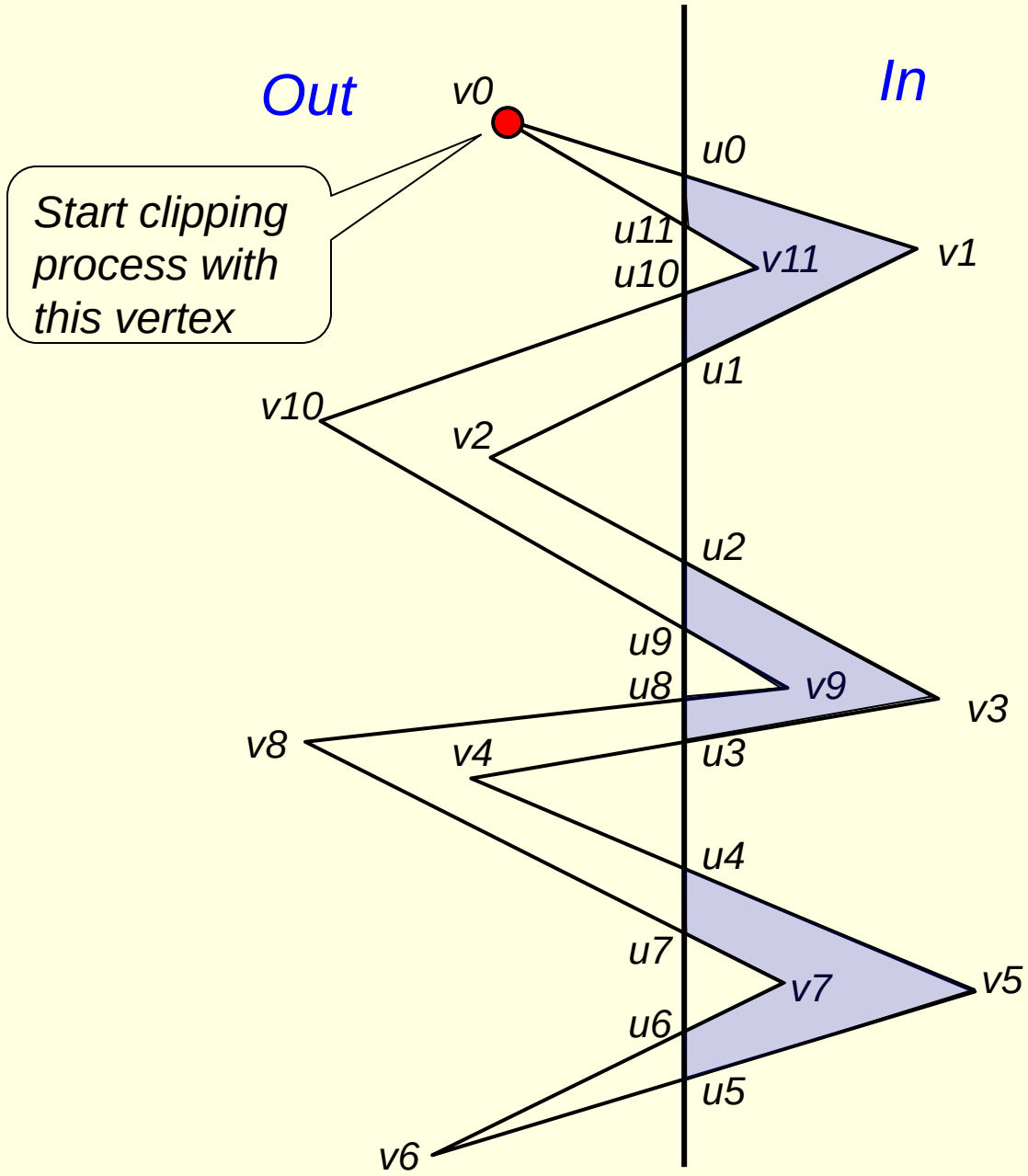
P6: u10→v11→u11

Polygon Clipping I

3. For each output vertex list, check if the start vertex and end vertex of any other vertex list are between the start vertex and end vertex of this list vertically (if the clipping edge is vertical) or horizontally (if the clipping edge is horizontal). If there is such a vertex list for this output vertex list, then these two vertex lists should be merged into a single list.

For instance, for the above polygon, vertex lists P1 and P6 should be merged into a single list. The merging process is simple, simply append P6 to P1.

After this step, we would have only 3 vertex lists for the given polygon. These sub-polygons are then clipped against other bounding edges of the window.



P1: u0→v1→u1

P2: u2→v3→u3

P3: u4→v5→u5

P4: u6→v7→u7

P5: u8→v9→u9

P6: u10→v11→u11

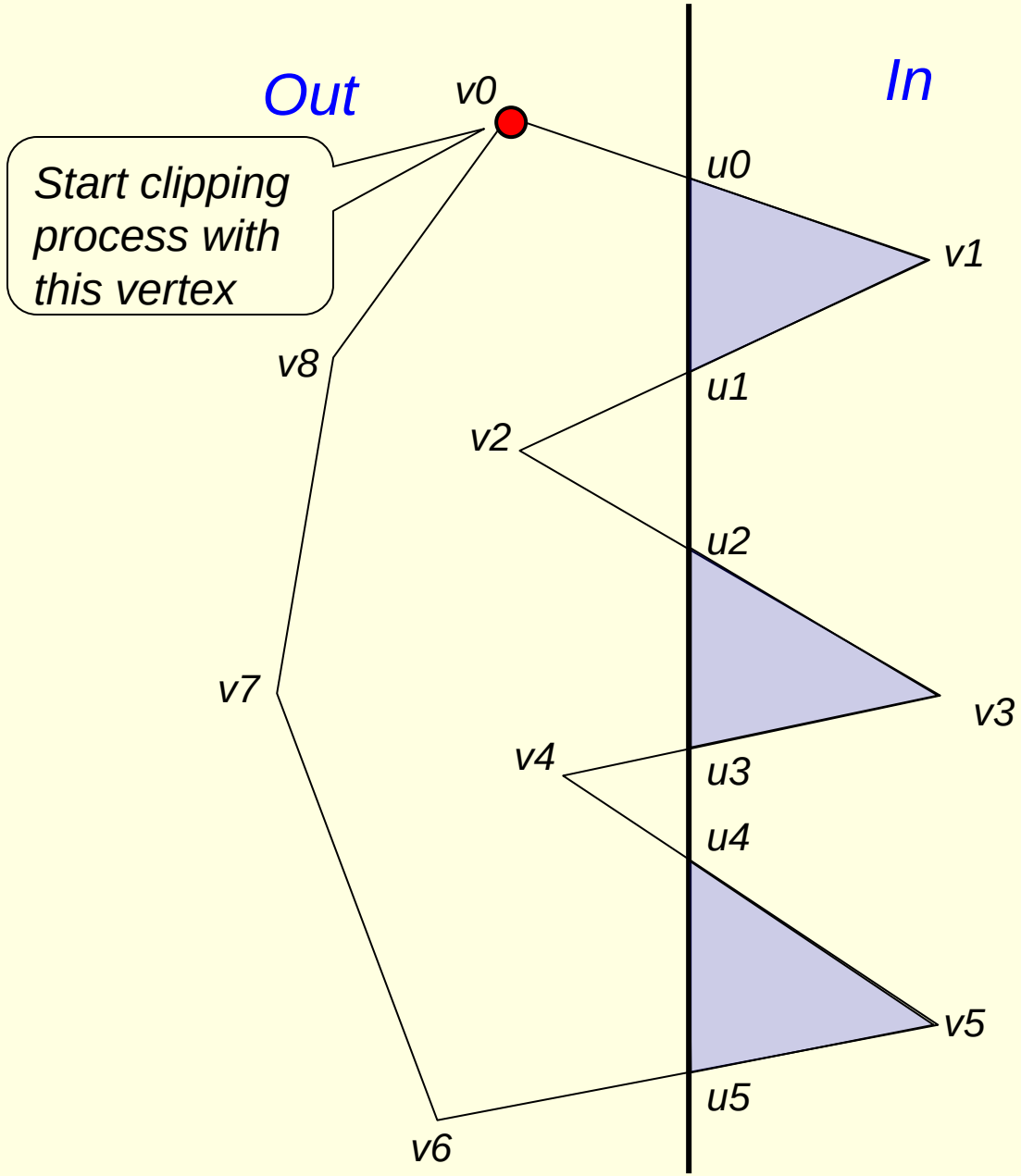
P1: u0→v1→u1→ u10→v11→u11

P2: u2→v3→u3→ u8→v9→u9

P3: u4→v5→u5→ u6→v7→u7

Polygon Clipping I

4. If a vertex list does not contain any other vertex list as a subset, such as the ones shown in the next slide, the vertex list forms closed sub-polygon by itself.



P1: u0→v1→u1

P2: u2→v3→u3

P3: u4→v5→u5