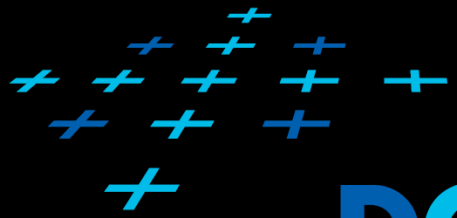


DCGI

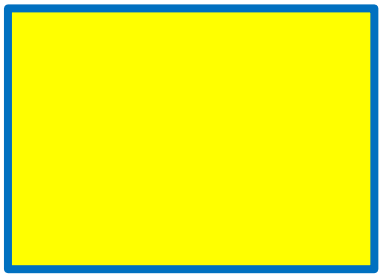
KATEDRA POČÍTAČOVÉ GRAFIKY A INTERAKCE

APG – Clipping

JIŘÍ ŽÁRA

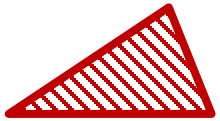
Clipping methods

- Window:
  - Parallel to X and Y axis (axis-aligned)

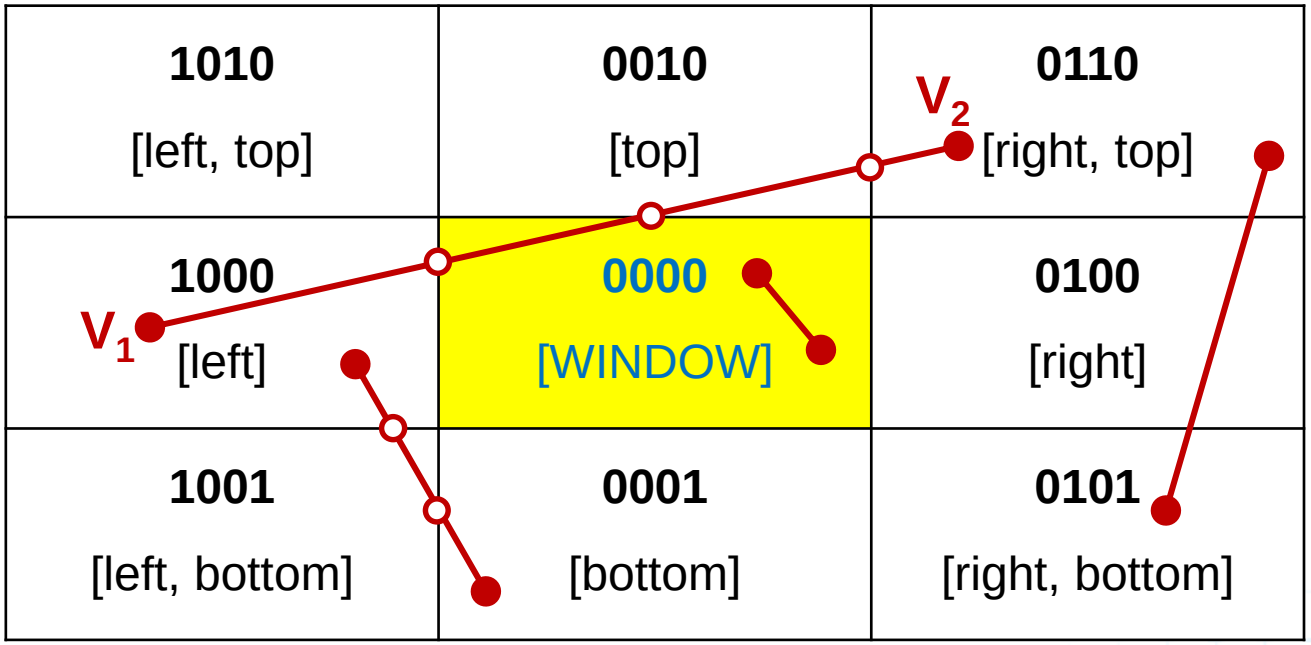


- Line clipping:
  1. OutCodes (Sutherland & Cohen)
  2. Line parametric expression (Liang & Barsky)

- Polygon clipping:
  - Reentrant polygon clipping (Sutherland & Hodgman)



OutCodes for line endpoints V_1 and V_2



1. **OutCode**(V_1) OR **OutCode** (V_2) == [0000] => line inside window
2. **OutCode** (V_1) AND **OutCode** (V_2) != [0000] => line outside window
3. otherwise => clip V_1 and/or V_2 and repeat tests 1 and 2

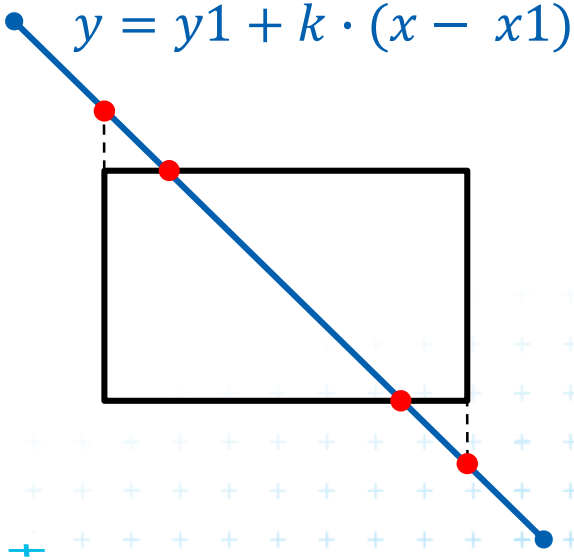
OutCodes - Sutherland & Cohen (1967)

1010	0010	0110
1000	0000	0100
1001	0001	0101

LEFT	RIGHT	TOP	BOTTOM
$sign(x - x_{min})$	$sign(x_{max} - x)$	$sign(y_{max} - y)$	$sign(y - y_{min})$

$$y = y1 + k \cdot (x - x1)$$

$$k = \frac{y2 - y1}{x2 - x1}$$

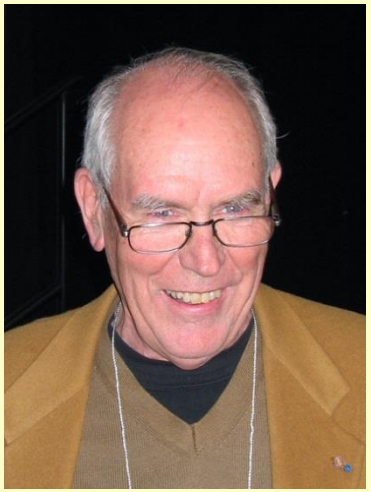


$$x = \begin{cases} x_{min} \\ x_{max} \end{cases}$$

Ivan E. Sutherland & Danny Cohen

Ivan Edward Sutherland

*1938



- MS, Caltech
- Ph.D., MIT, 1963
- Turing prize in 1988, for SketchPad program
- Evans & Sutherland company

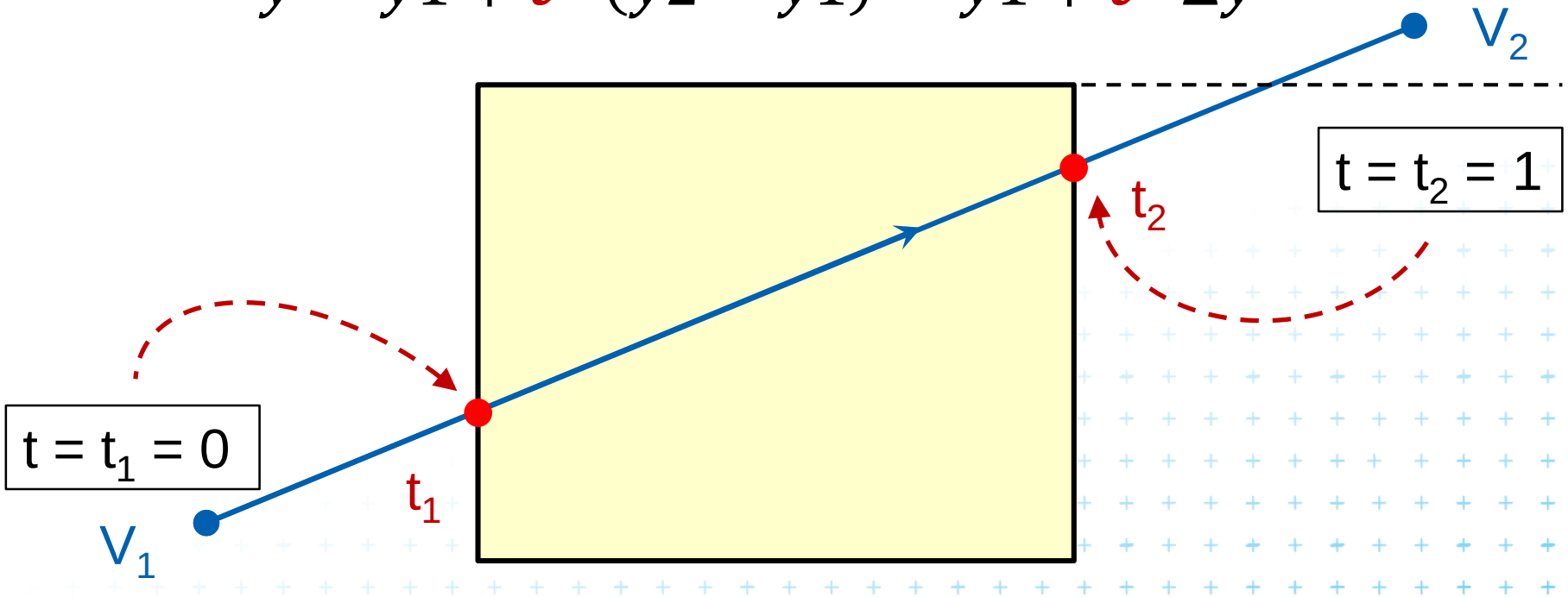
Danny Cohen

- BC, Technion, Israel, 1963
- Ph.D., Harvard, 1969
- Algorithm developed when programming the first real-time visual flight simulator (ARPANet)!
- Sun Microsystems, Oracle

Line parameters clipping (Liang & Barsky)

- More efficient: Do tests first, compute later…
- Parametric expression of a line:

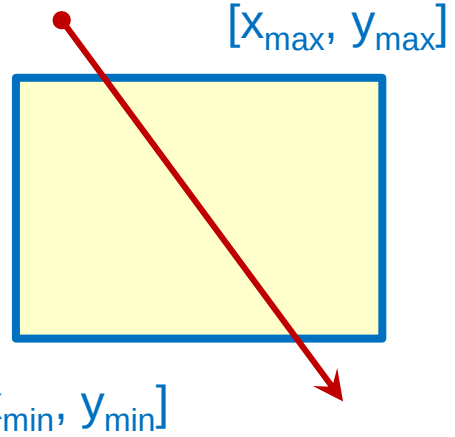
$$x = x1 + \boldsymbol{t} \cdot (x2 - x1) = x1 + \boldsymbol{t} \cdot \Delta x$$
$$y = y1 + \boldsymbol{t} \cdot (y2 - y1) = y1 + \boldsymbol{t} \cdot \Delta y$$



Line parameters - Liang & Barsky (1983)

$$x = x1 + t \cdot (x2 - x1) = x1 + t \cdot \Delta x$$
$$y = y1 + t \cdot (y2 - y1) = y1 + t \cdot \Delta y$$

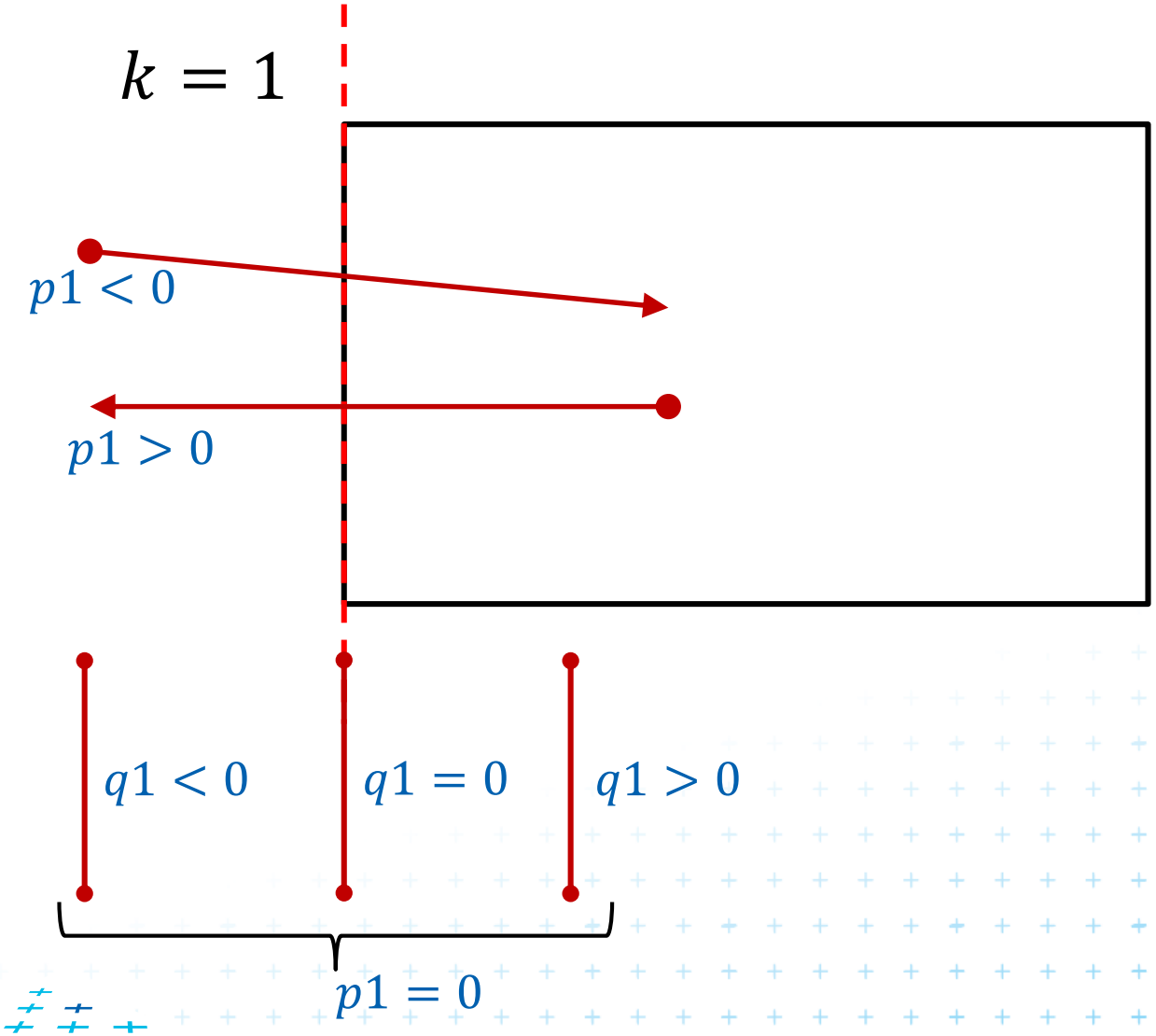
$$x_{min} \leq x1 + t \cdot \Delta x \leq x_{max}$$
$$y_{min} \leq y1 + t \cdot \Delta y \leq y_{max}$$



$$t \cdot p_k \leq q_k \qquad k = 1, 2, 3, 4$$

$p_1 = -\Delta x$	$q_1 = x1 - x_{min}$	// left window border
$p_2 = \Delta x$	$q_2 = x_{max} - x1$	// right
$p_3 = -\Delta y$	$q_3 = y1 - y_{min}$	// lower
$p_4 = \Delta y$	$q_4 = y_{max} - y1$	// upper

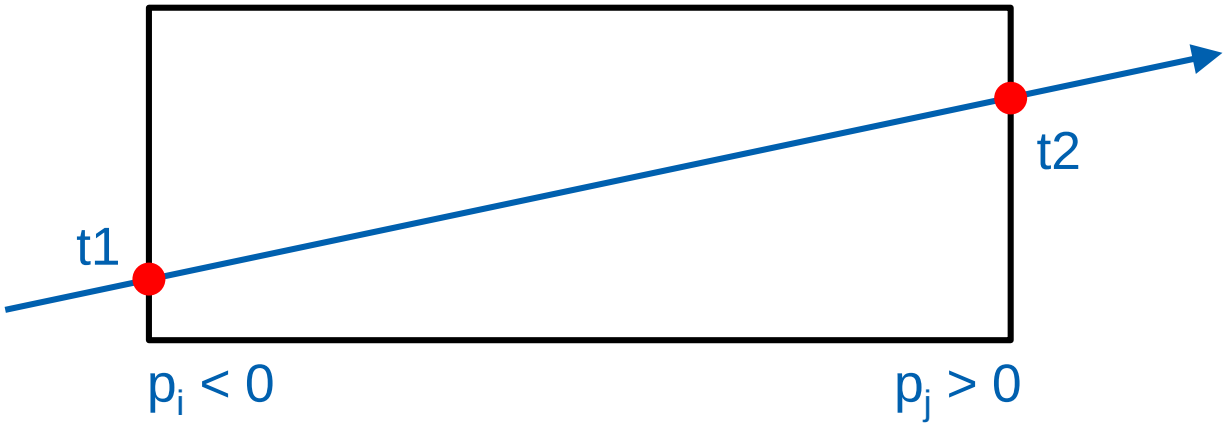
Liang & Barsky – *p* & *q* for the left border

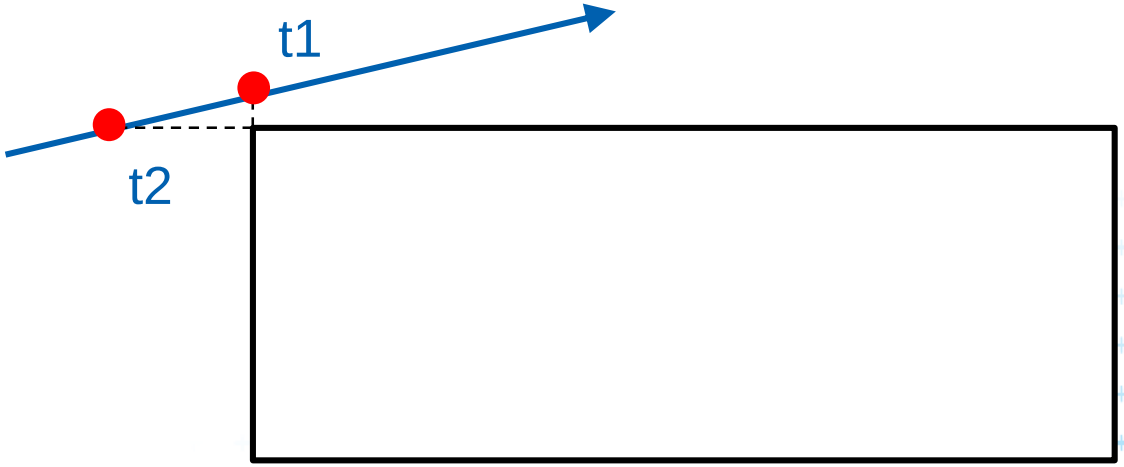


Intersection

$$t = \frac{q}{p}$$

Liang & Barsky – parameter *t*





t1 > t2: outside window

t1 < t2: clipping
(t1 > 0, t2 < 1)

Liang & Barsky – Test function

```
bool Test (double p, double q, double &t1, double &t2) {
        double t;
        if (p < 0) {  t = q/p;
                if (t > t2) { return false; }
                else if (t > t1) { t1 = t; }          // into window, update t1
        } else if (p > 0) { t = q/p;
                if (t < t1) { return false; }
                else if (t < t2) { t2 = t; }          // from window, update t2
        } else {
                if (q < 0) { return false; }          // parallel with border
        }
        return true;
}
```

$$t \cdot p_k \leq q_k$$

Liang & Barsky – Clip function

```
function Clip (double x1, double y1, double x2, double y2) {
        double t1, t2, dx, dy;
        t1 = 0;   t2 = 1;   dx = x2 – x1;   dy = y2 – y1;
        if (Test (-dx, x1-xmin, t1, t2) & Test (dx, xmax-x1, t1, t2)  &
           Test (-dy, y1-ymin, t1, t2) & Test (dy, ymax-y1, t1, t2) ) {
              if (t2 < 1) {     x2 = x1 + t2 * dx;
                                y2 = y1 + t2 * dy;
              }
              if (t1 > 0) {     x1 += t1 * dx;
                                y1 += t1 * dy;
              }
              DrawLine(x1, y1, x2, y2);
        }
}
```


Brian A. Barsky & You-Dong Liang

- ***„A New Concept and Method for Line Clipping*** “
  ACM TOG, Vol. 3, Issue 1, Jan. 1984, pp. 1-22
- 22-pages long paper introduces also a general concept for viewing frustum (pyramid) clipping in 3D!

Brian Barsky

- MS, Cornell
- Ph.D. Univ. of Utah
- Prof. @ Univ. of California, Berkeley

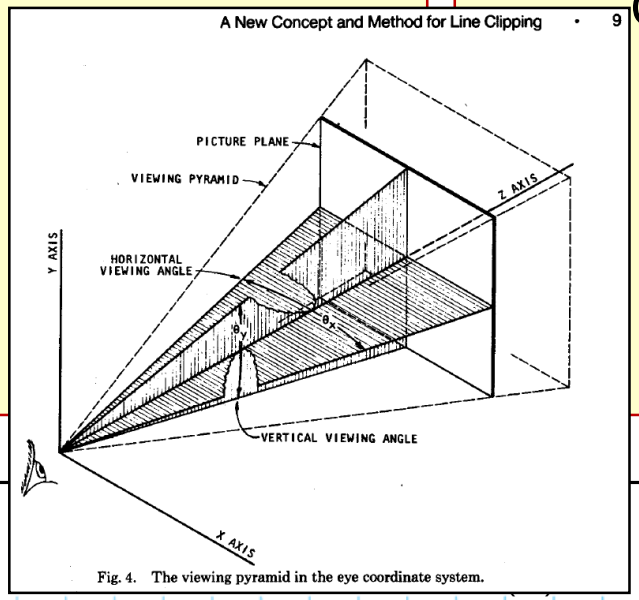


Fig. 4. The viewing pyramid in the eye coordinate system.

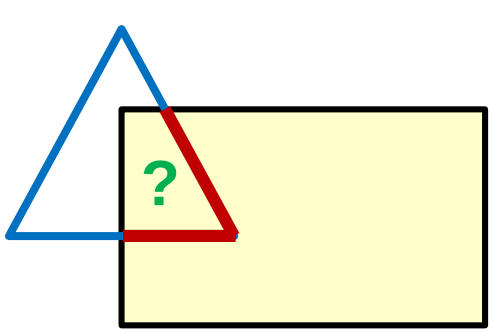
Y. D. Liang

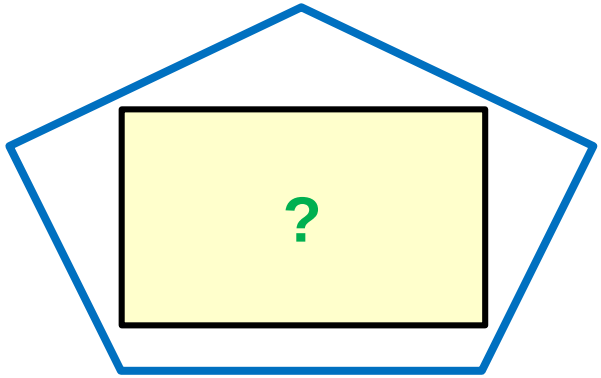
- Ph. D. student of prof. Barsky (Berkeley)
- Now at Zhejiang University, China (CAD/CAM)

DCGI

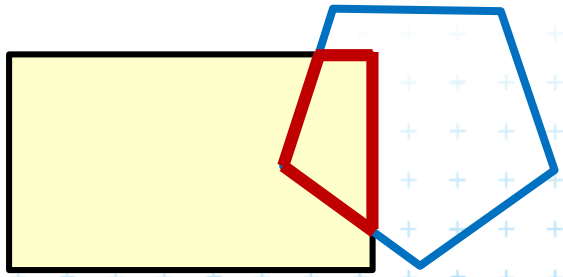
Polygon clipping

- Cannot be replaced by a sequence of line clippings:

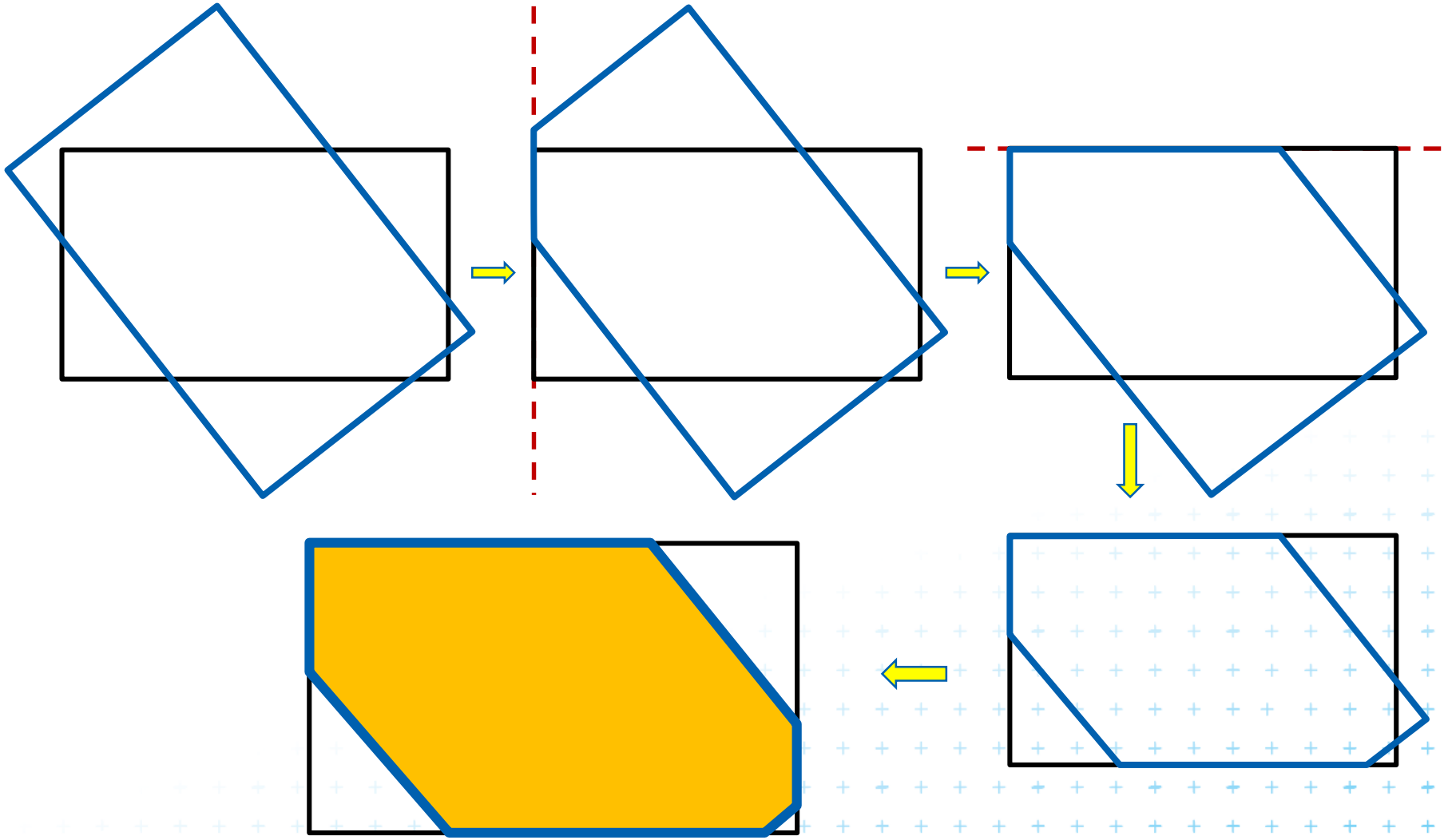




- Number of polygon vertices changes:



Incremental clipping by infinite border edges



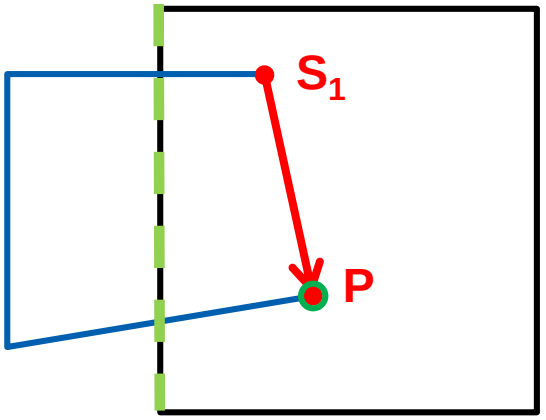
Ivan E. Sutherland & Garry W. Hodgman

- ***Reentrant polygon clipping***,
  CACM, Vol. 17, Issue 1, Jan. 1974, pp. 32-42
- Both authors from Evans & Sutherland company

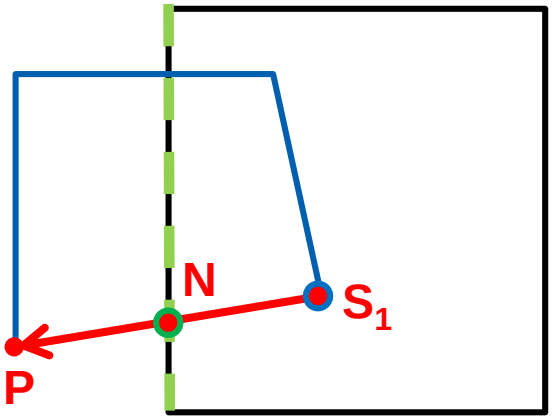
- Clipping by **arbitrary convex polygon**
- Incremental processing of input vertices by border/clipping edges.
- Data structures:
  - Si, already processed vertex by clipping edge ***i***
  - P, vertex currently processed

Sutherland & Hodgman

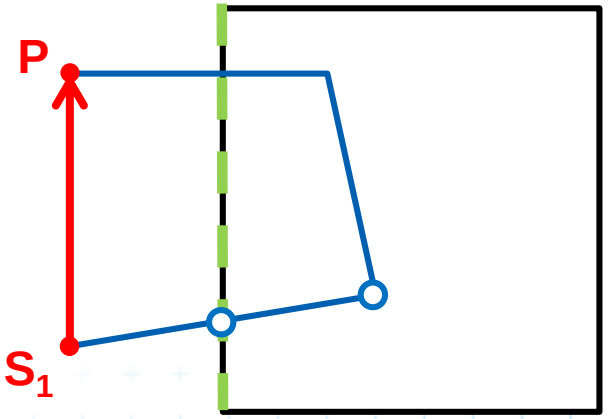
- Possible situations in clipping by left border edge:



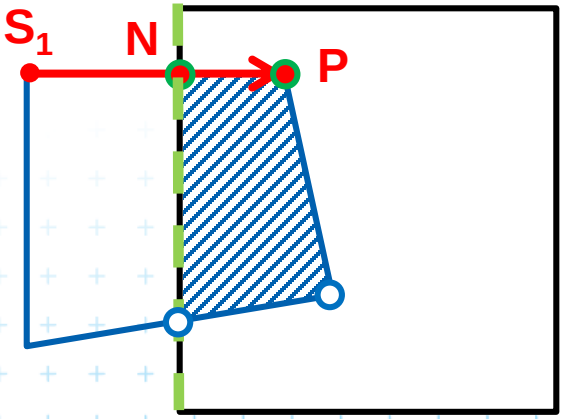
P



N

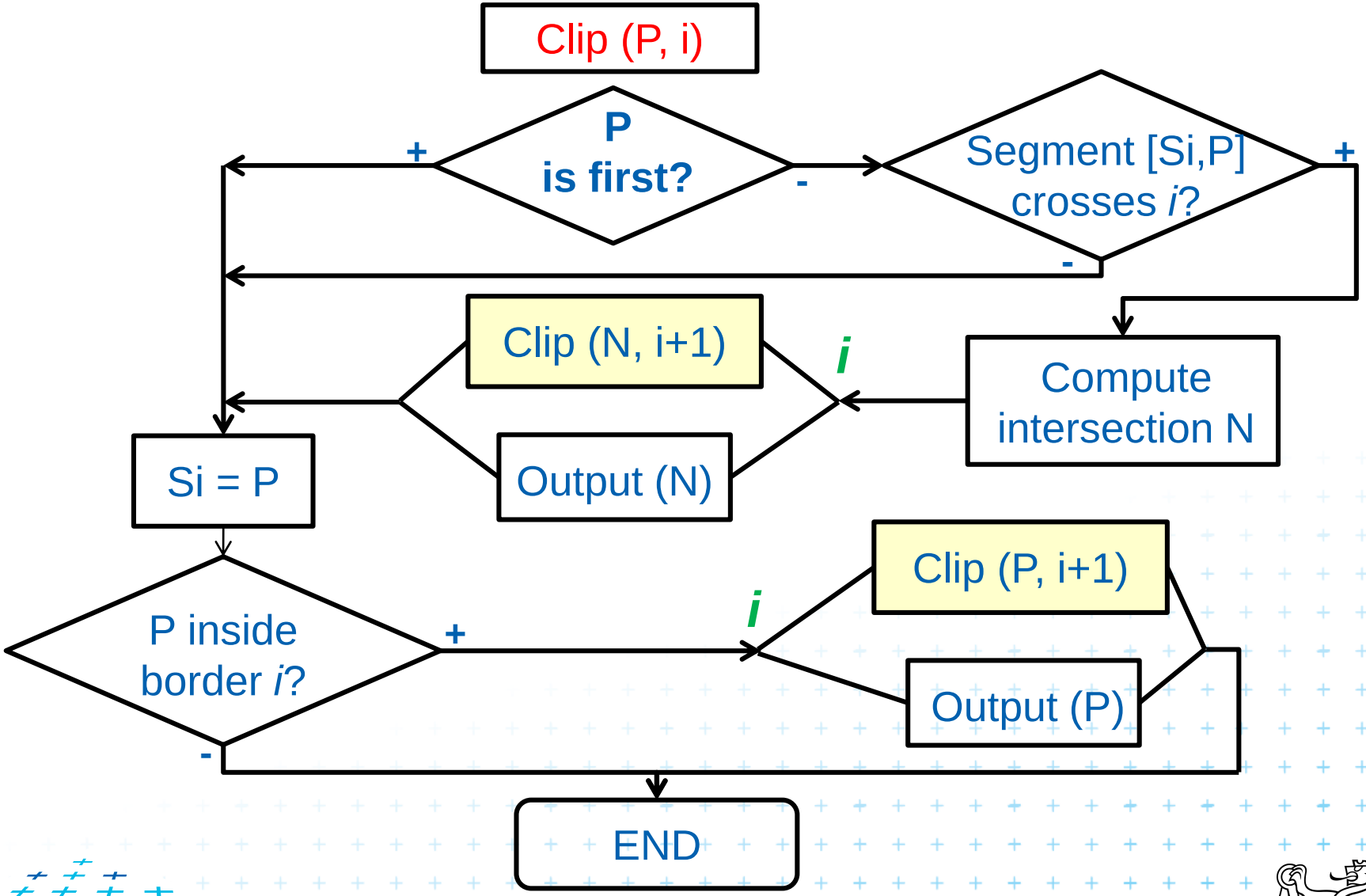


–

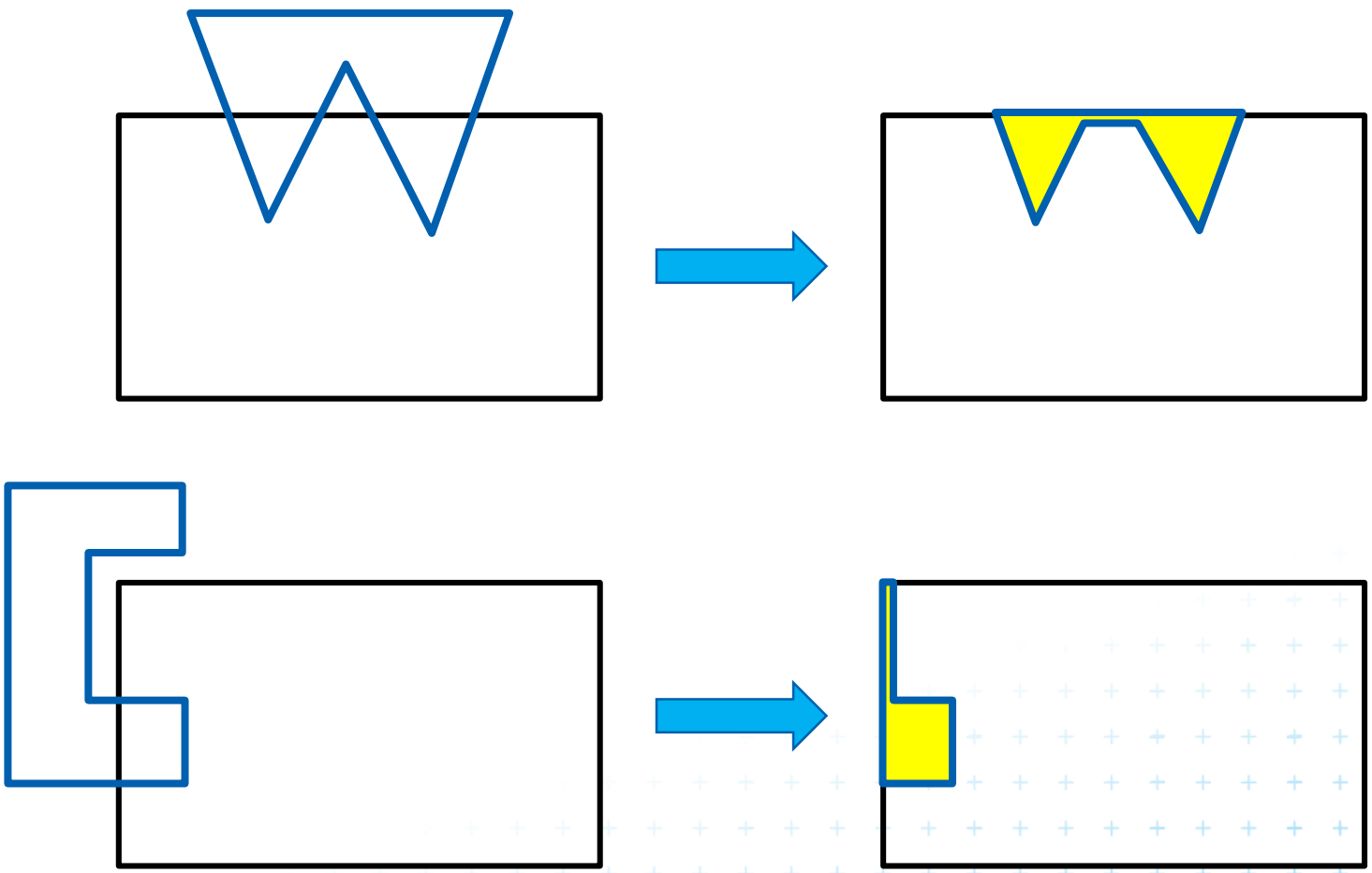


N + P

Sutherland & Hodgman – algorithm

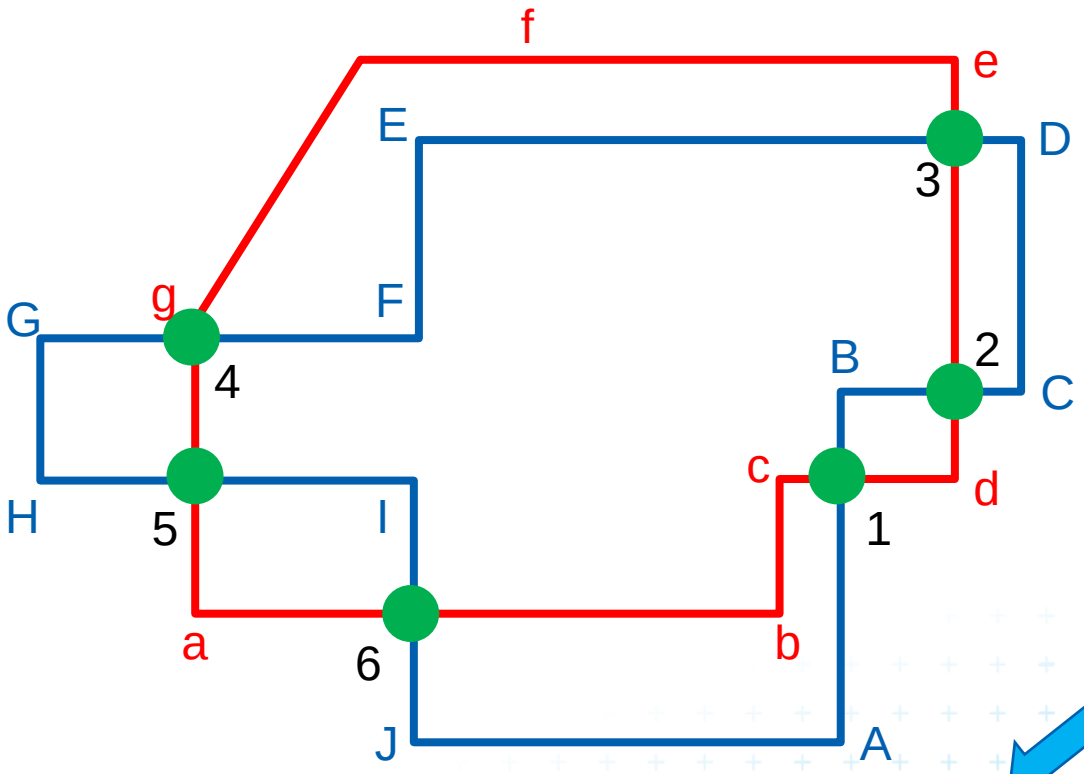


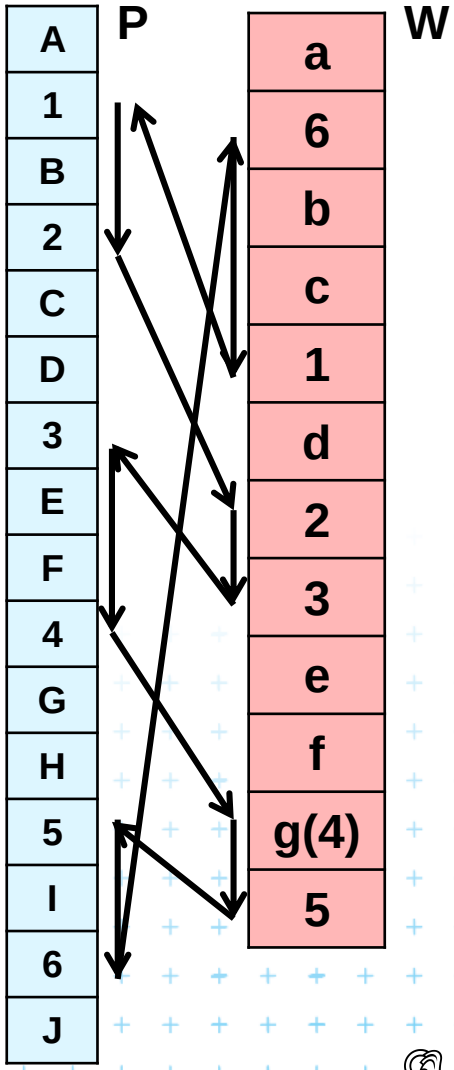
Issues in clipping



General polygon by polygon clipping

- Weiler-Atherton algorithm





List H	1	2	3	4	5	6

Thank you for your attention

Jiří Žára, 28.01.2021

DCGI