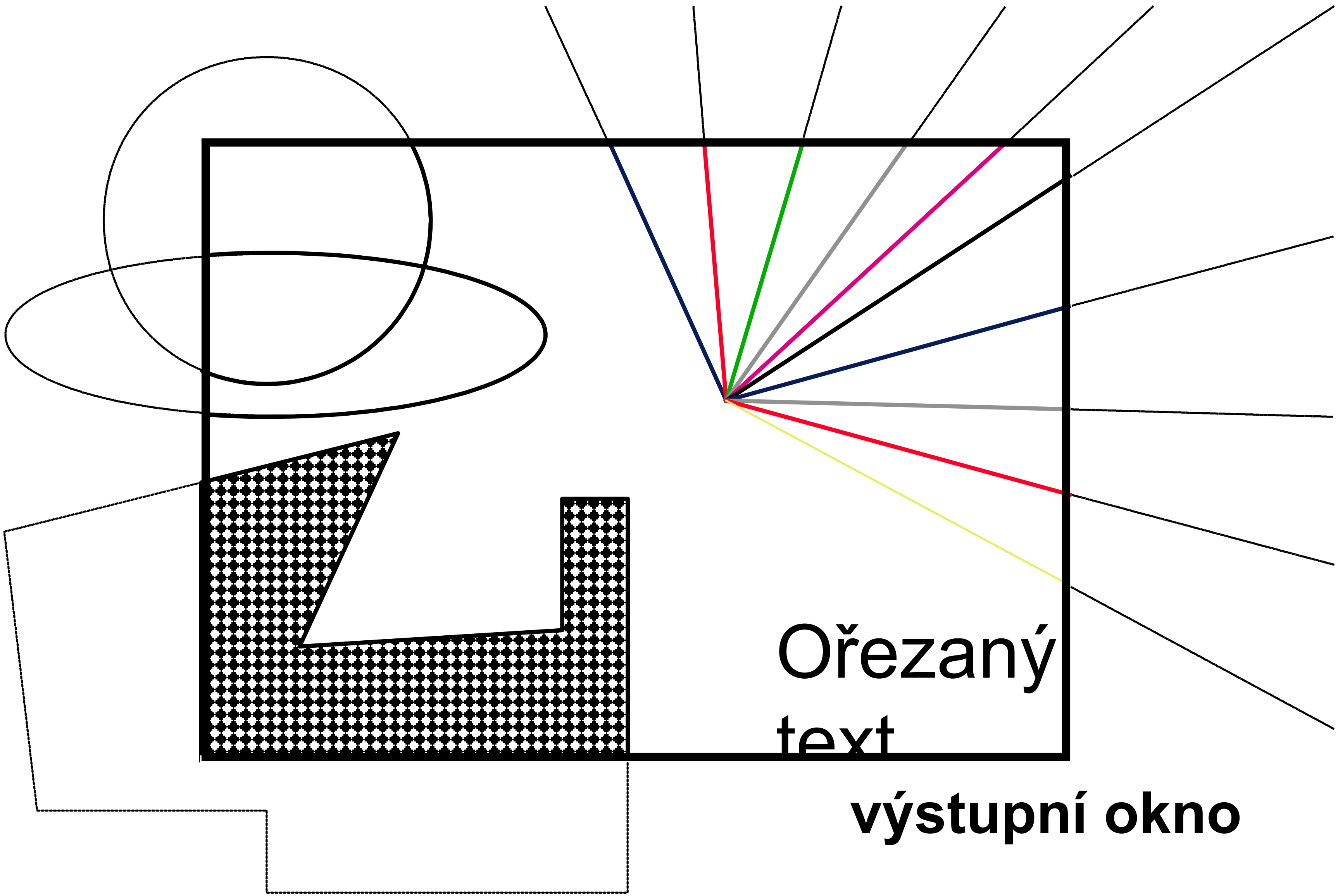
Ořezávání v rovině

© 1995-2015 Josef Pelikán
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
http://cgg.mff.cuni.cz/~pepca/

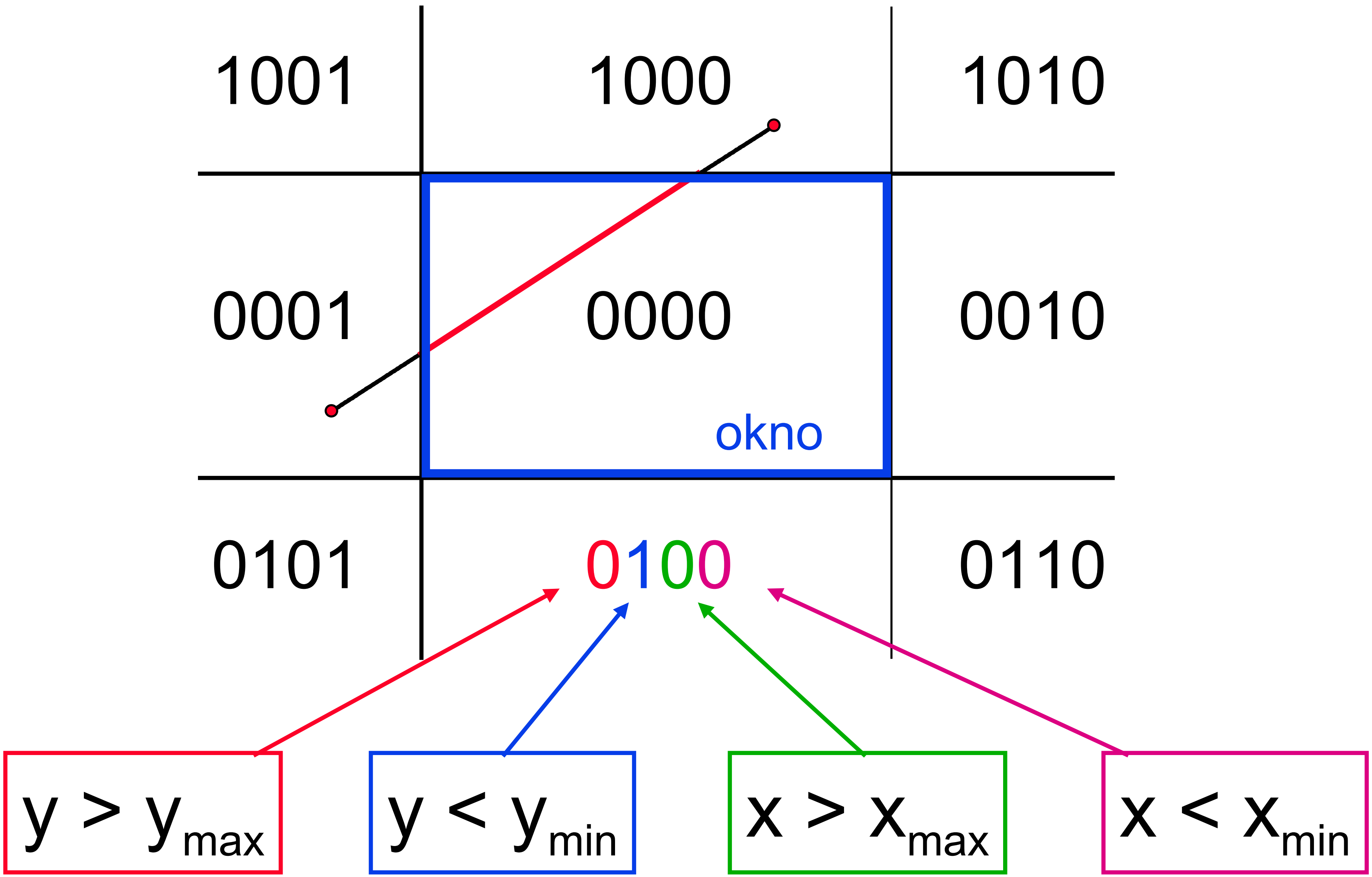
2D ořezávání



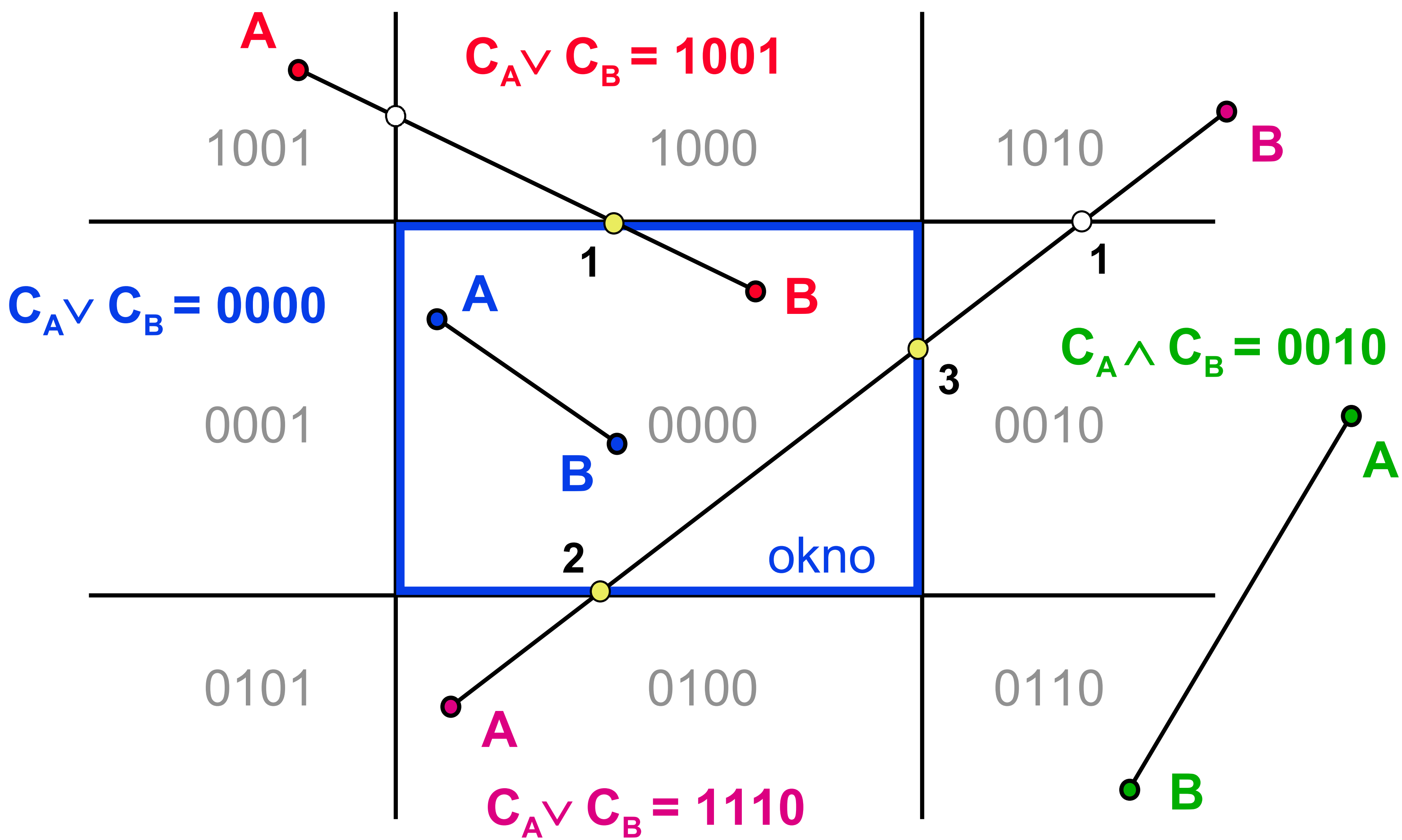
Ořezávání úseček

- přepočítávají se **koncové body** úseček:
  - $[x_1,y_1]$-$[x_2,y_2] \rightarrow [x_A,y_A]$-$[x_B,y_B]$ nebo $\varnothing$
- běžně jsou x_A,y_A,x_B,y_B v **celočíselném** formátu
  - nepřesnosti v kresbě
- **racionální** souřadnice jsou nepraktické
  - algoritmus kreslení úsečky mívá celočíselný vstup
- nejlepší by bylo spočítat **mezivýsledky** algoritmu na kreslení úsečky (Bresenham)

Kódy oblastí (Cohen-Sutherland)



Cohen-Sutherland



Cohen-Sutherland

❶ spočítám **kódy koncových bodů** $\mathbf{C_A}$, $\mathbf{C_B}$

❷ je-li $\mathbf{C_A \lor C_B = 0}$, celá úsečka leží **uvnitř okna**

❸ je-li $\mathbf{C_A \land C_B \neq 0}$, celá úsečka leží **mimo okno**

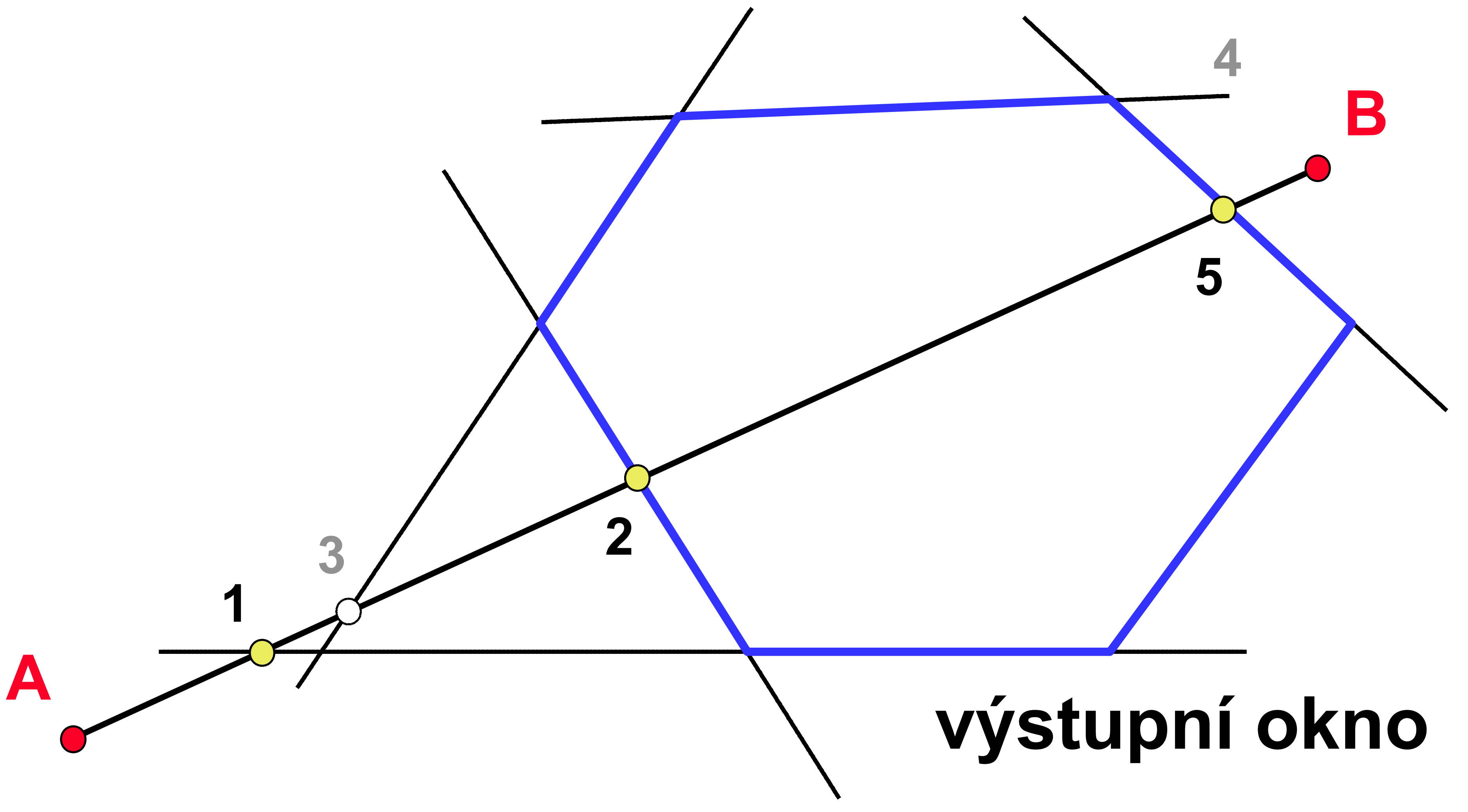
❹ pro každou jedničku v $\mathbf{C_A \lor C_B}$ postupně úsečku ořezávám - např. pro $\mathbf{Y_{max}}$ počítám:

– $\mathbf{X := X_A + (X_B - X_A) * (Y_{max} - Y_A) / (Y_B - Y_A)}$

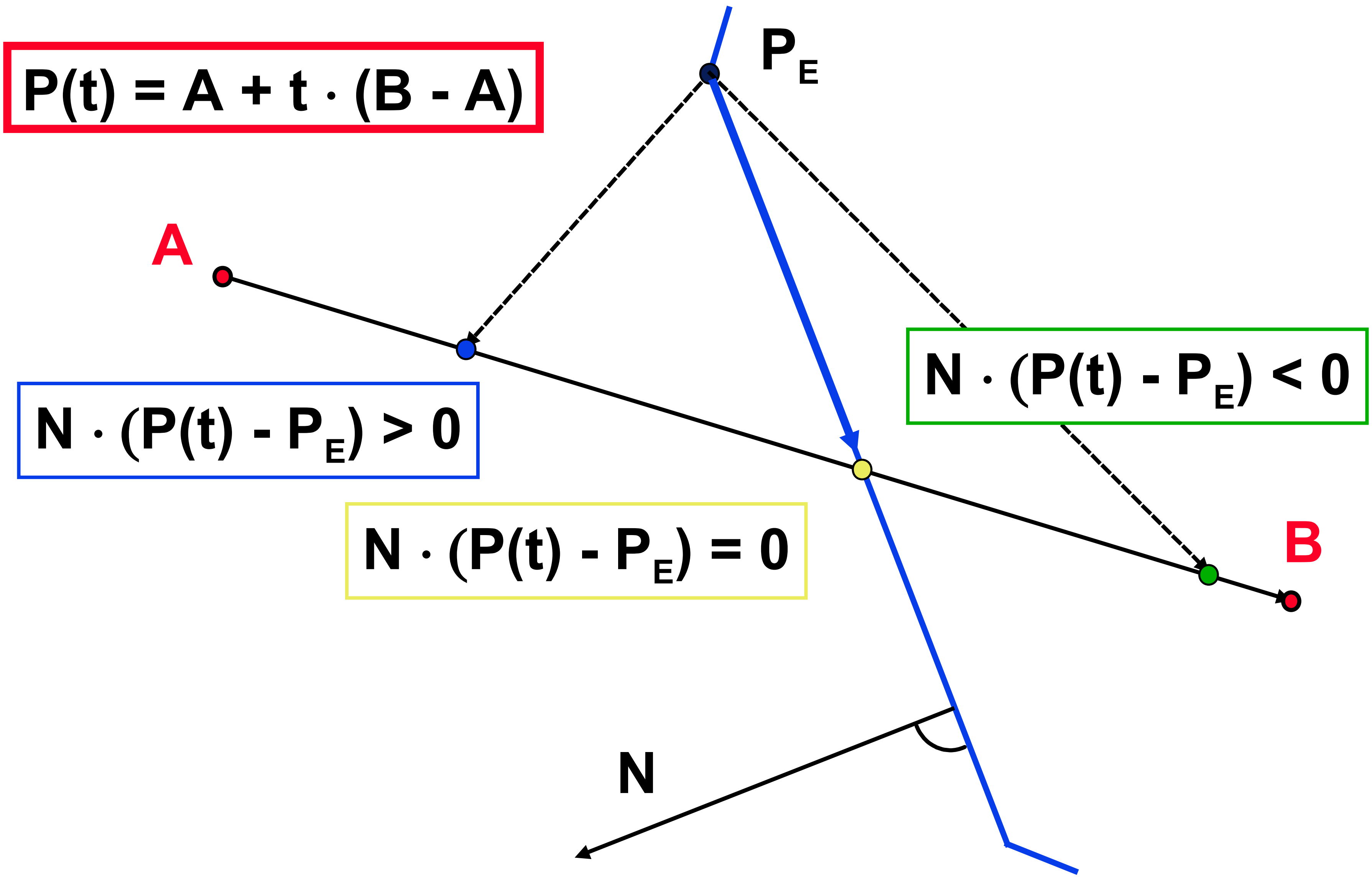
– $\mathbf{Y := Y_{max}}$

– (**A** nebo **B**) **:= [X,Y]**, opravím kód $\mathbf{C_A}$ nebo $\mathbf{C_B}$

Okno = konvexní n-úhelník



Parametrické ořezávání (Cyrus-Beck)



Výpočet průsečíku

$$N \cdot \left(P\left(t_0 \right) - P_E \right) = 0$$
$$N \cdot \left[A + t_0 \cdot \left(B - A \right) - P_E \right] = 0$$

$$\boxed{t_0 = -\frac{N \cdot \left(A - P_E \right)}{N \cdot \left(B - A \right)}}$$

$N \cdot \left(B - A \right) = 0 \rightarrow$ úsečka je rovnoběžná s hranicí

$N \cdot \left(B - A \right) < 0 \rightarrow$ úsečka míří dovnitř okna

$N \cdot \left(B - A \right) > 0 \rightarrow$ úsečka míří ven z okna

Cyrus-Beck

❶ $\mathbf{t_{min} := 0.0; t_{max} := 1.0}$

❷ dokud $\mathbf{t_{min} < t_{max}}$, opakuji pro každou hraniční přímku kroky ❸ a ❹ [pak spočtu $\mathbf{P(t_{min})}$ a $\mathbf{P(t_{max})}$]

❸ je-li $\mathbf{N \cdot (B - A) = 0}$, **celá úsečka** leží v jedné polorovině (rozhodnu podle znaménka $\mathbf{N \cdot (A - P_E)}$)

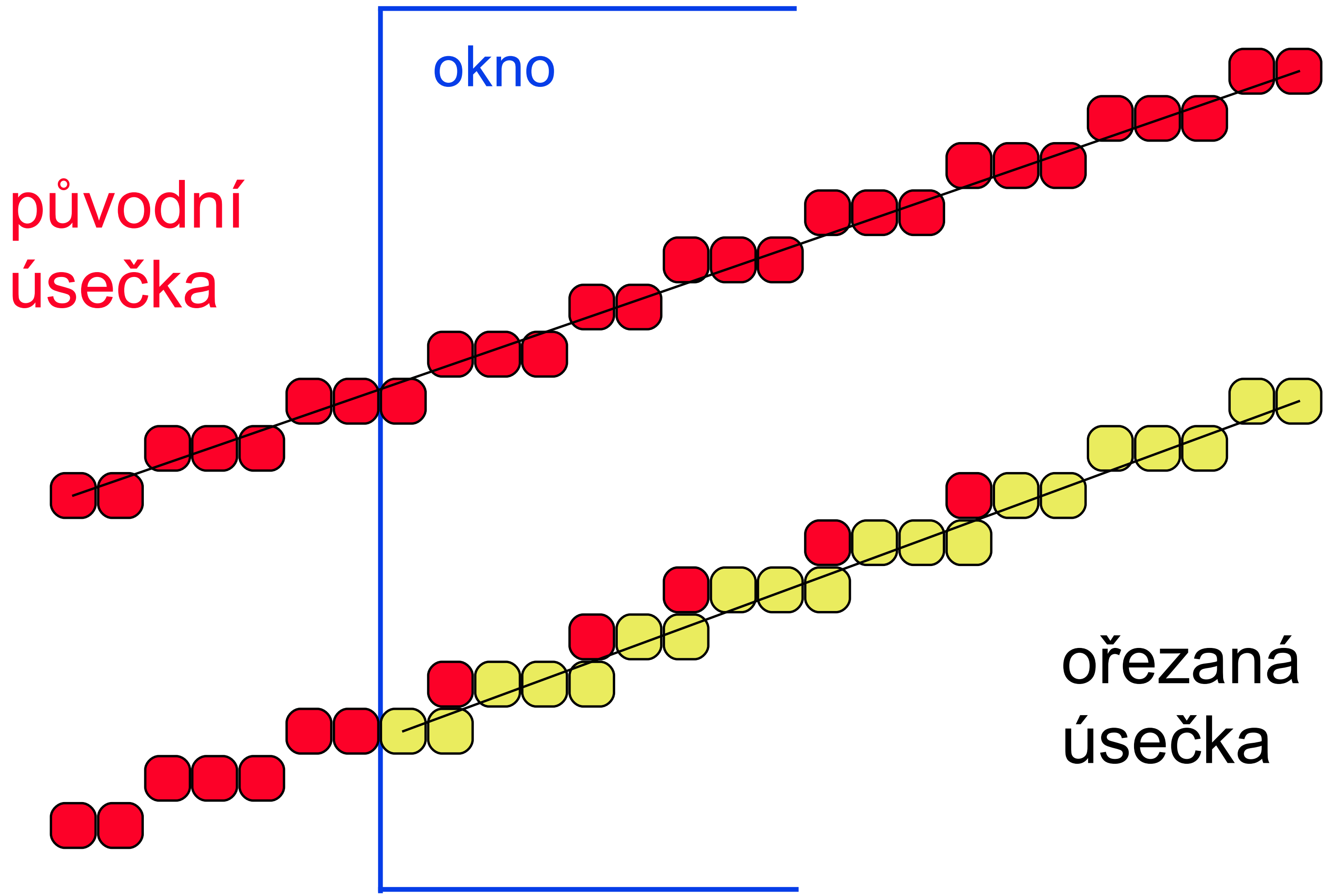
❹ jinak spočítám $\mathbf{t_0}$ a podle znaménka $\mathbf{N \cdot (B - A)}$ opravím $\mathbf{t_{min}}$ nebo $\mathbf{t_{max}}$:

$$\mathbf{t_{min} := max\{\, t_{min}, t_0 \,\}} \text{ nebo } \mathbf{t_{max} := min\{\, t_{max}, t_0 \,\}}$$

Liang-Barsky

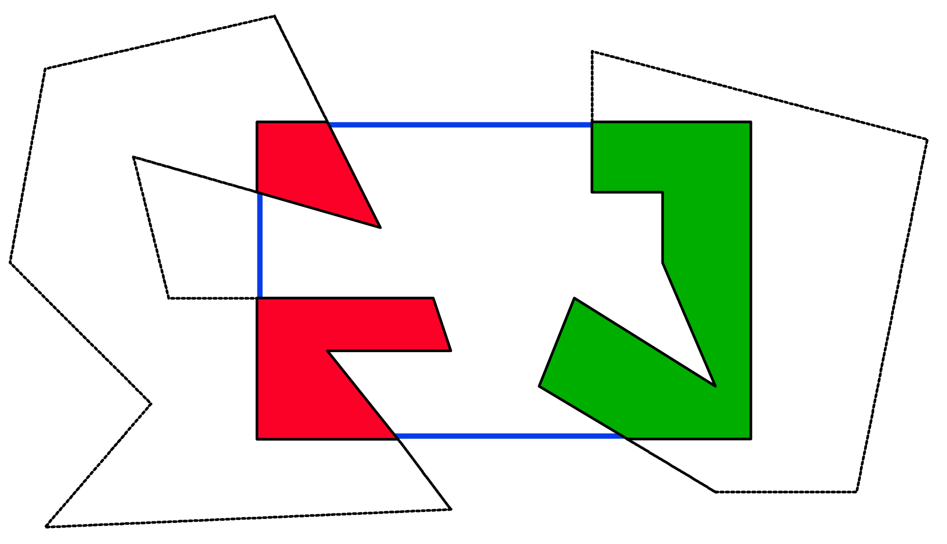
- efektivní úprava parametrického algoritmu pro **obdélníkové okno**
  - původní algoritmus Cyrus-Beck počítá s obecným konvexním n-úhelníkem
  - normálové vektory jsou triviální ([1,0], [0,-1], ...)
  - maximálně: 8 add/sub , 14 cmp , 4 div, 4 mul
- **Cohen-Sutherland**: většina případů je triviálních
- **Liang-Barsky**: většina úseček se ořezává
  - čím více se ořezává, tím je výhodnější než C-S

Zaokrouhlovací chyby

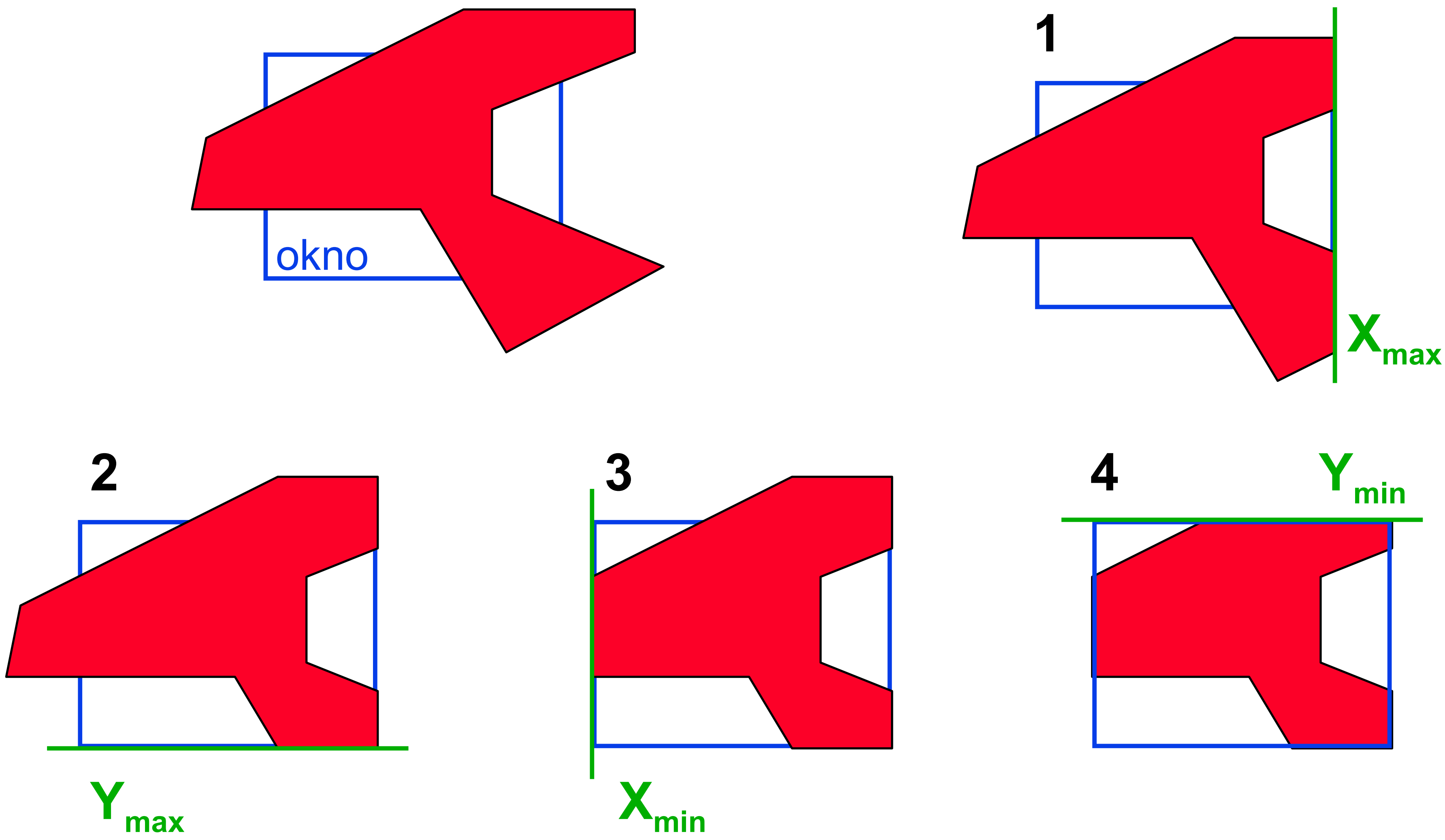


Ořezávání n-úhelníků

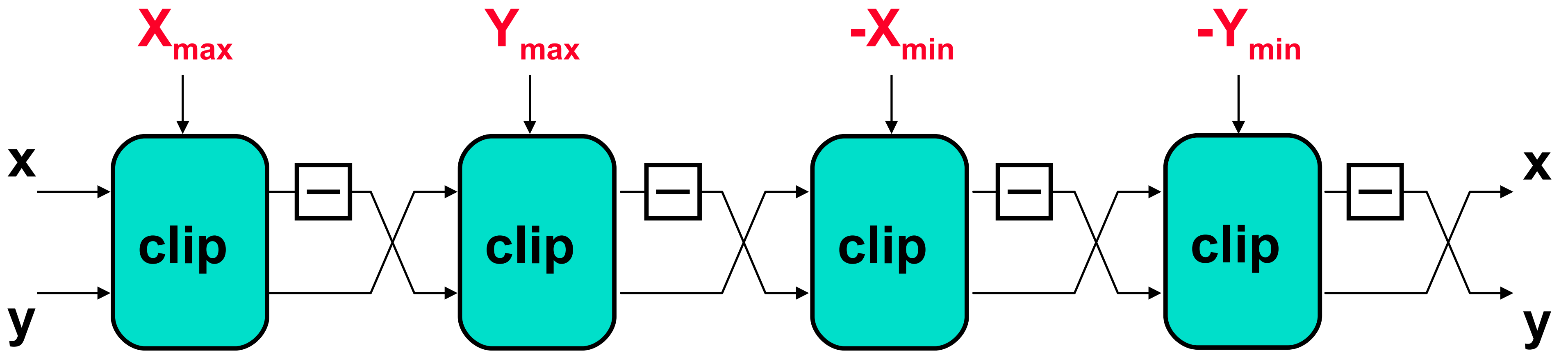
- jestliže chceme kreslit pouze **obrys n-úhelníku**, pak stačí ořezávat hrany samostatně jako úsečky
- chceme-li **vybarvovat vnitřek n-úhelníku**, musíme ho oříznout speciálním algoritmem:



Proudové ořez. (Sutherland-Hodgman)



Sutherland-Hodgman (HW)

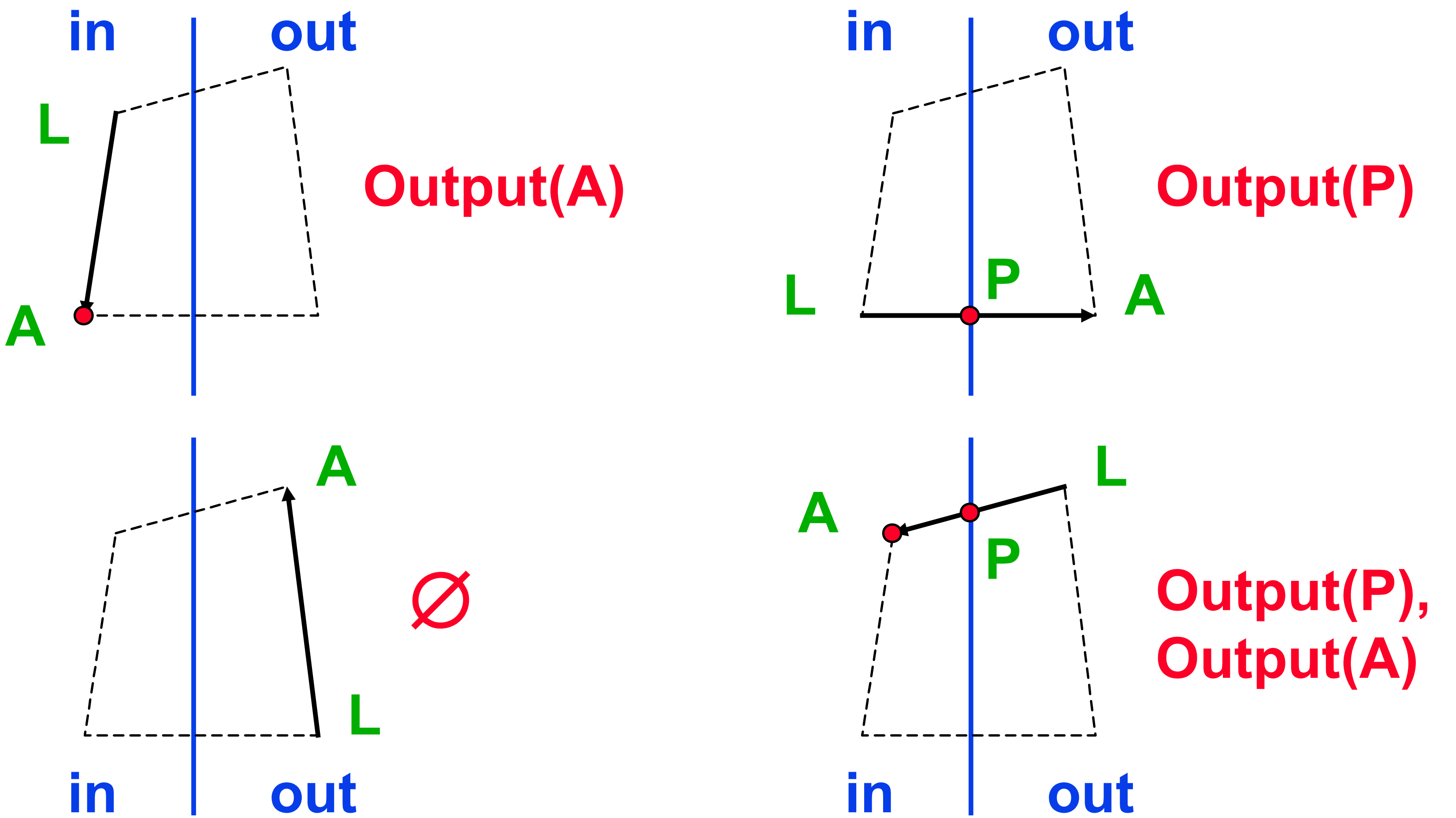


modul „clip“ ořízne n-úhelník podle hranice $\mathbf{x = X_{max}}$

- mezi jednotlivými moduly se souřadnice **otáčejí o 90°**

Modul „**clip**”

- pamatuje si poslední dva vrcholy **L** a **A**



Konec

Další informace:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 111-127
- **Jiří Žára a kol.**: ***Počítačová grafika***, principy a algoritmy, 153-168