Introdução à Computação Gráfica
Recorte

Claudio Esperança
Paulo Roma Cavalcanti

O Problema de Recorte

- Dada uma superfície *M* fechada de co-dimensão 1 do R^n , o complemento de M, (R^n-*M*), possui duas componentes conexas.
- Se *S* é um subconjunto do R^n, chama-se de recorte de *S* por *M* à operação que consiste em determinar os subconjuntos de *S* que estão em cada uma das componentes conexas.

Recorte (*Clipping*)

- Problema definido por:
  - Geometria a ser recortada
    - Pontos, retas, planos, curvas, superfícies
  - Regiões de recorte
    - Janela (2D)
    - Volume de visibilidade
      - Frustum (tronco de pirâmide)
      - Paralelepípedo
    - Polígonos
      - Convexos
      - Genéricos (côncavos, com buracos, etc)

Resultado

- Depende da geometria:
  - Pontos: valor booleano (visível / não visível)
  - Retas: segmento de reta ou coleção de segmentos de reta
  - Planos: polígono ou coleção de polígonos

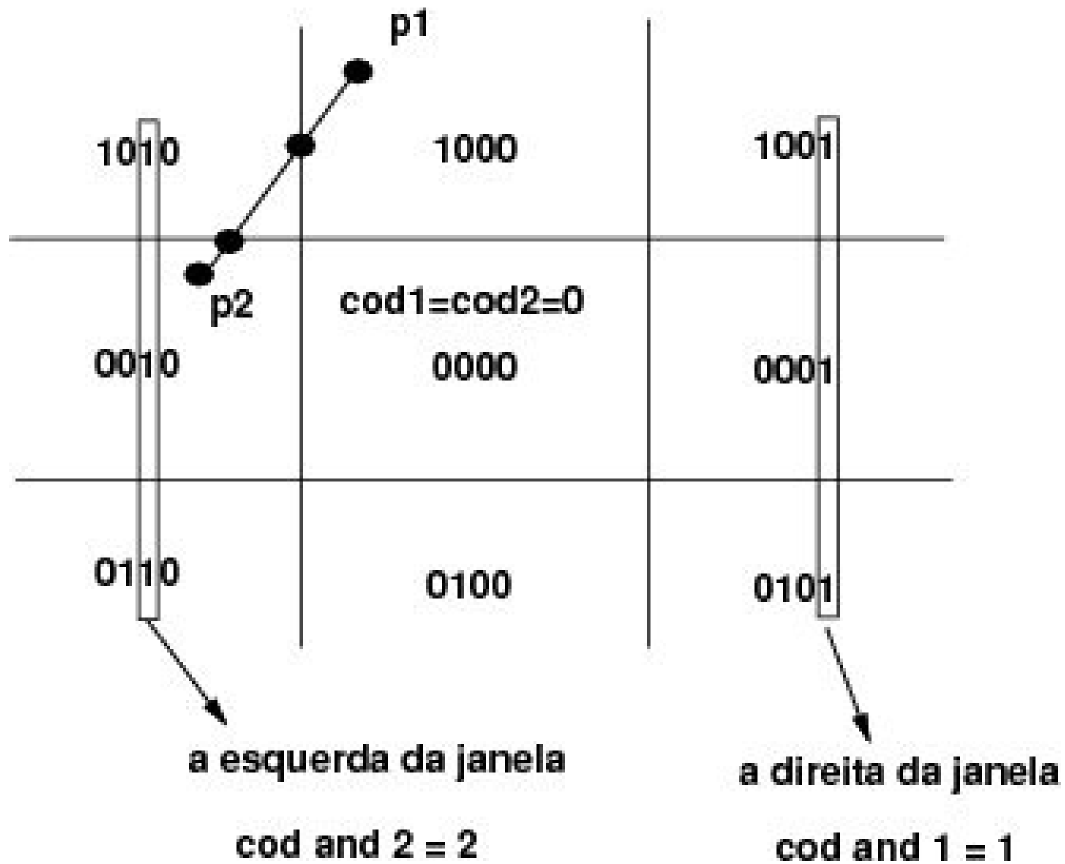
Recorte de Segmento de Reta x Retângulo

- Problema clássico 2D
- Entrada:
  - Segmento de reta P_1 - P_2
  - Janela alinhada com eixos (*xmin*, *ymin*) - (*xmax*, *ymax*)
- Saída: Segmento recortado (possivelmente nulo)
- Variantes
  - Cohen-Sutherland
  - Liang-Barksy / Cyrus-Beck
  - Nicholl-Lee-Nicholl

Cohen-Sutherland

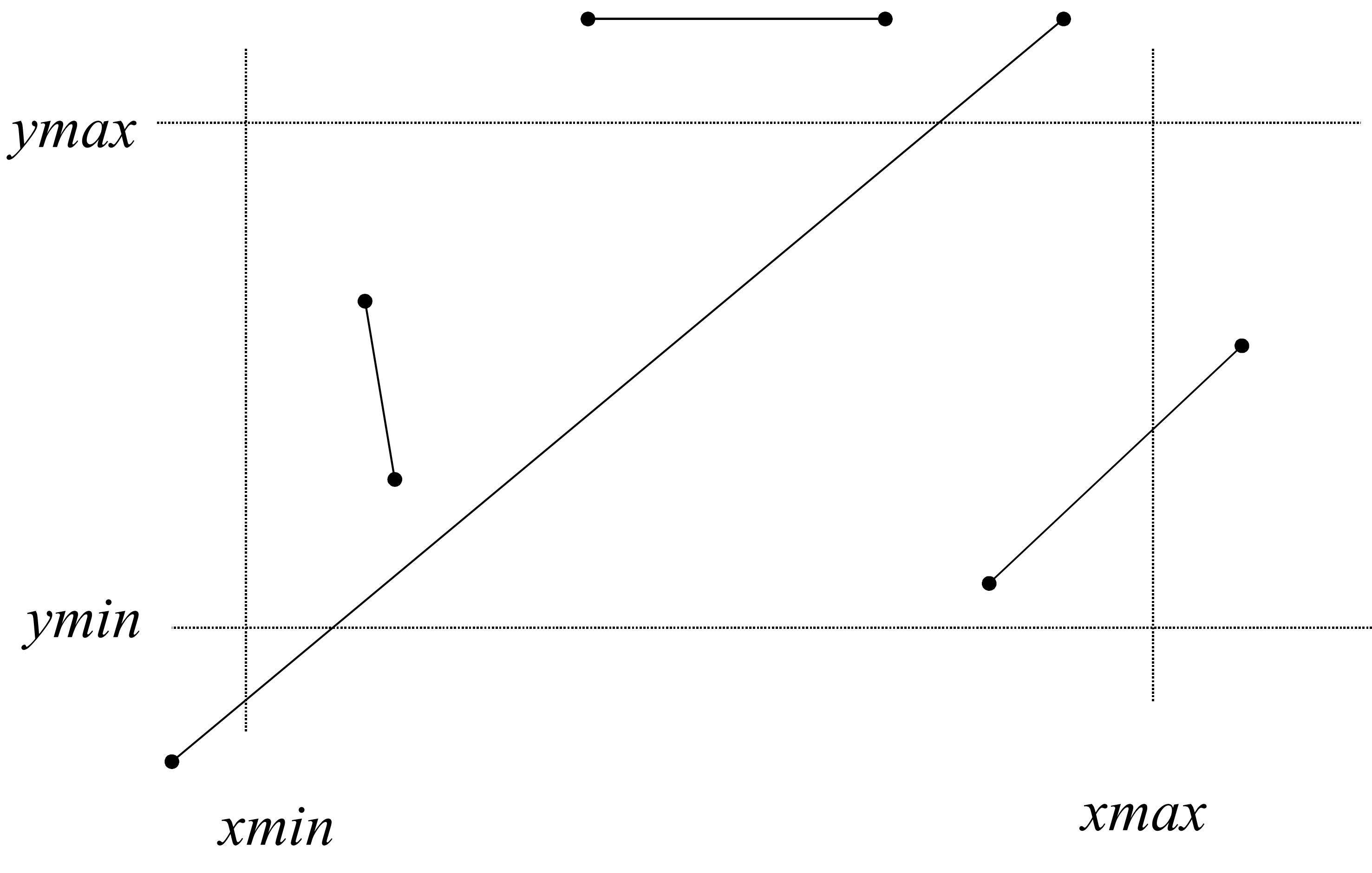
- A janela é definida pela interseção de 4 semi-planos:
  - $ymin \leq y \leq ymax$ e
  - $xmin \leq x \leq xmax$
- Os vértices do segmento são classificados em relação a cada semi-plano que delimita a janela, gerando um código de 4 bits:
  - $Bit1 = (y > ymax)$
  - $Bit2 = (y < ymin)$
  - $Bit3 = (x < xmin)$
  - $Bit4 = (x > xmax)$
- Se ambos os vértices forem classificados como fora, descartar o segmento (totalmente invisível)
- Se ambos forem classificados como dentro, testar o próximo semi-plano
- Se um vértice estiver dentro e outro fora, computar o ponto de interseção Q e continuar o algoritmo com o segmento recortado (P_1-Q ou P_2-Q)

Códigos

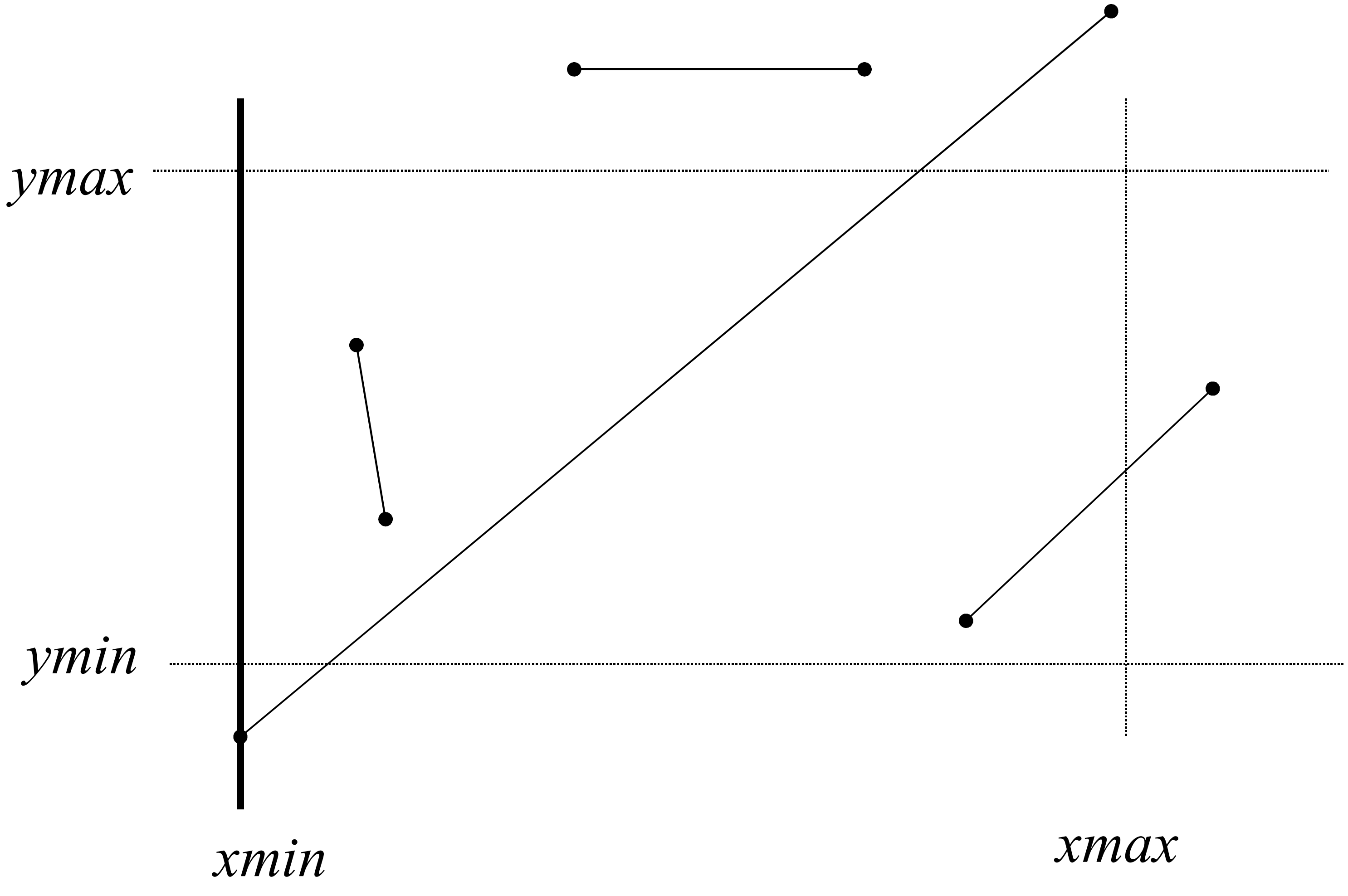


p1
1010
1000
1001
p2
cod1=cod2=0
0010
0000
0001
0110
0100
0101
a esquerda da janela
cod and 2 = 2
a direita da janela
cod and 1 = 1

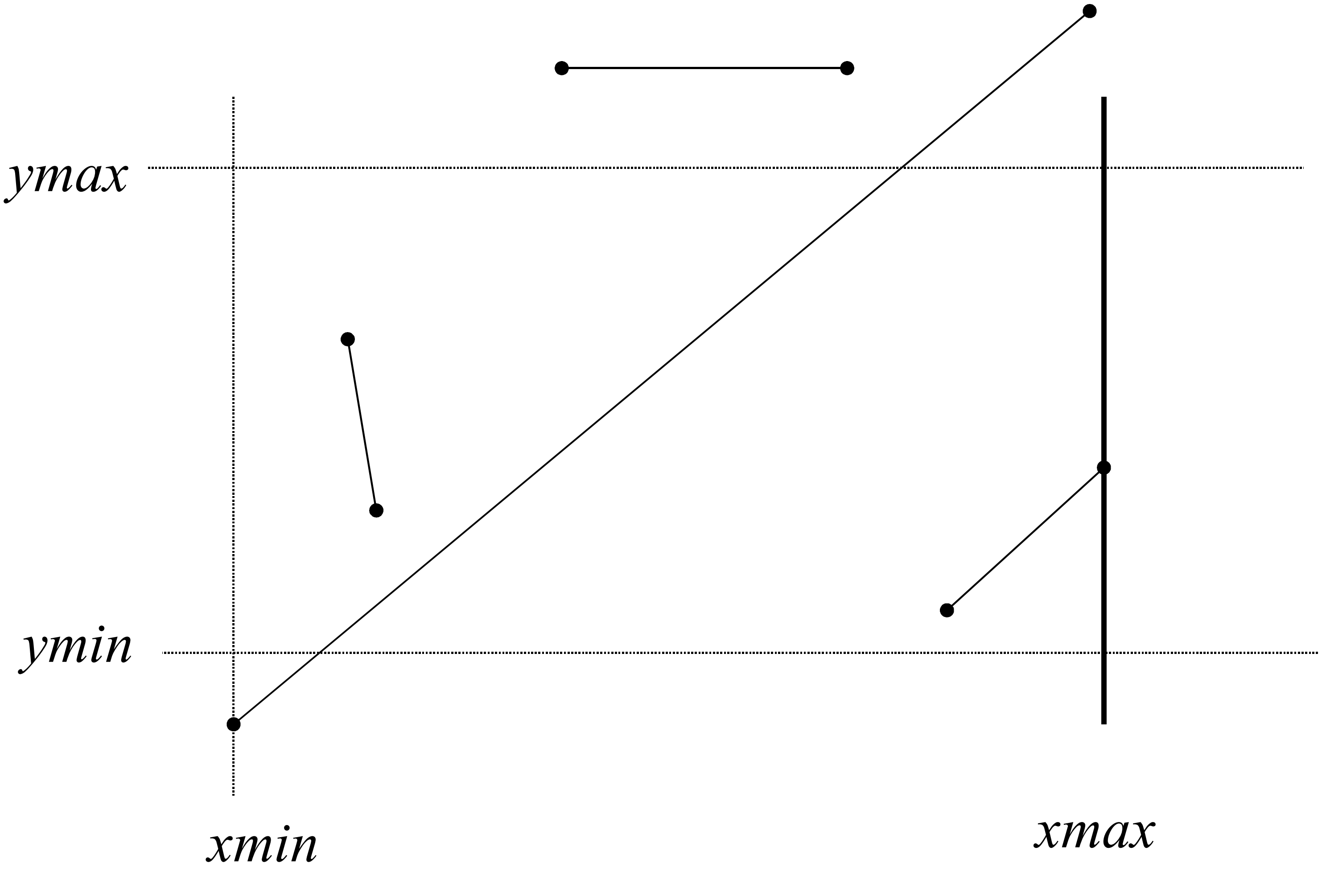
Cohen-Sutherland



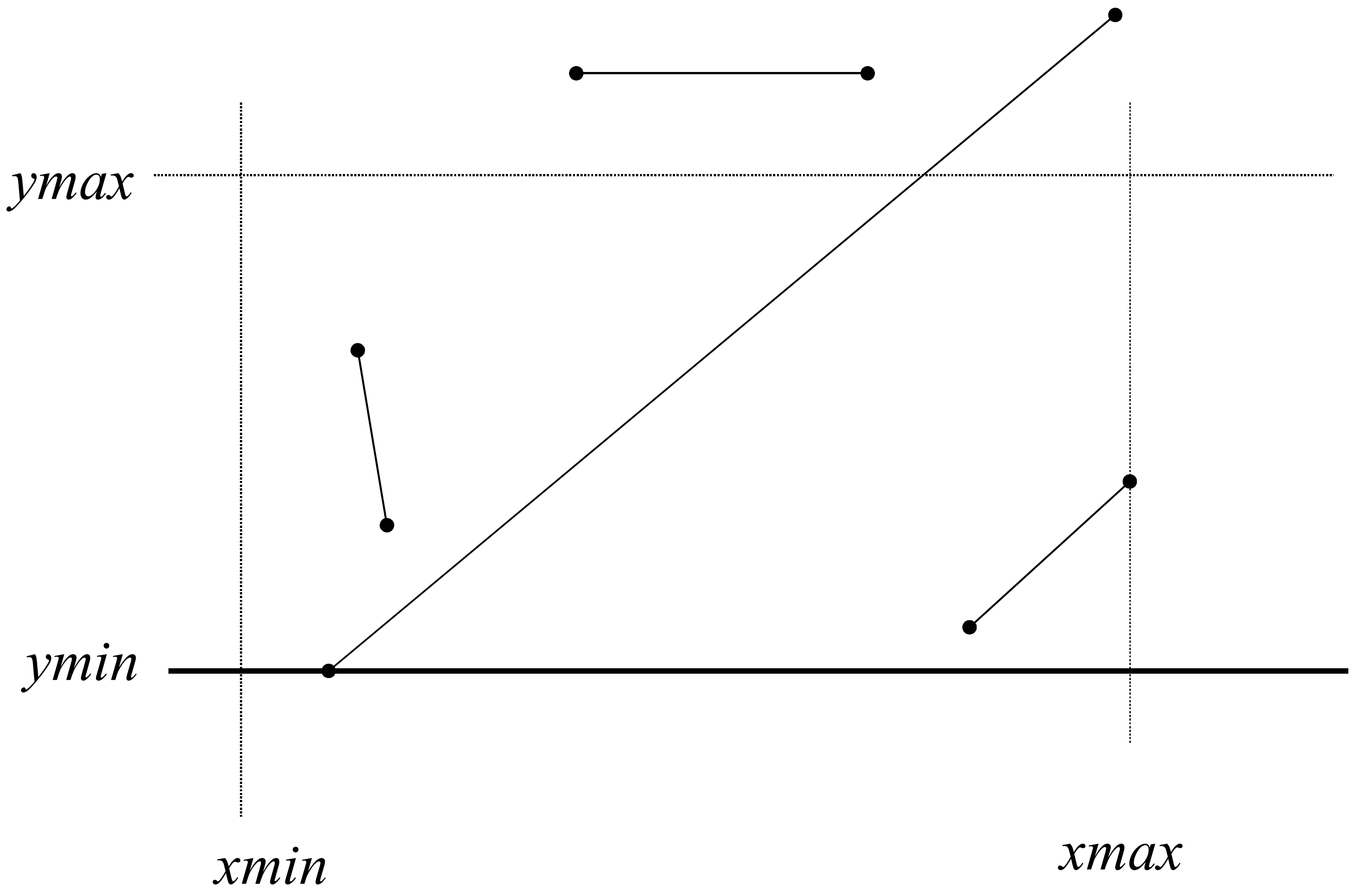
Cohen-Sutherland



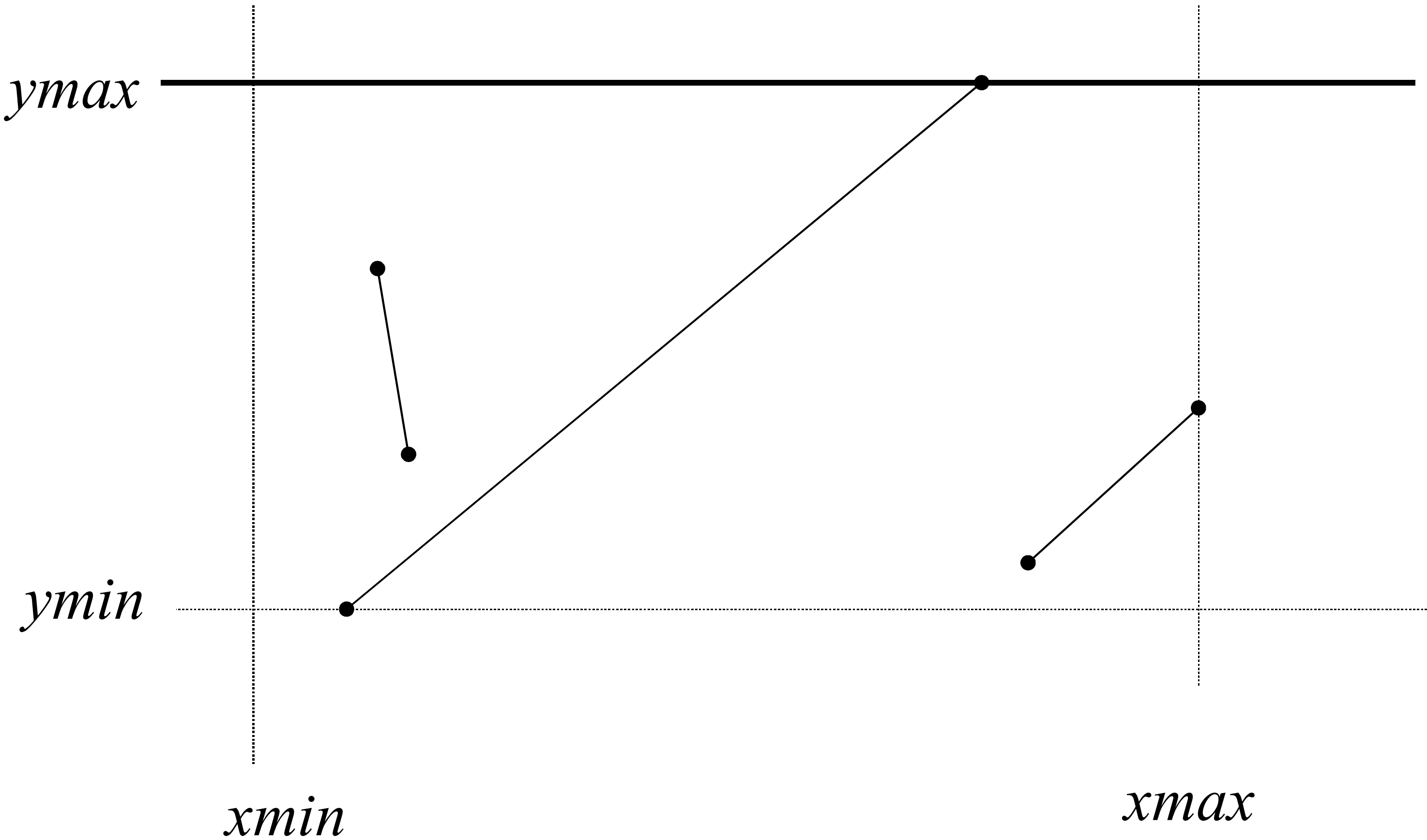
Cohen-Sutherland



Cohen-Sutherland

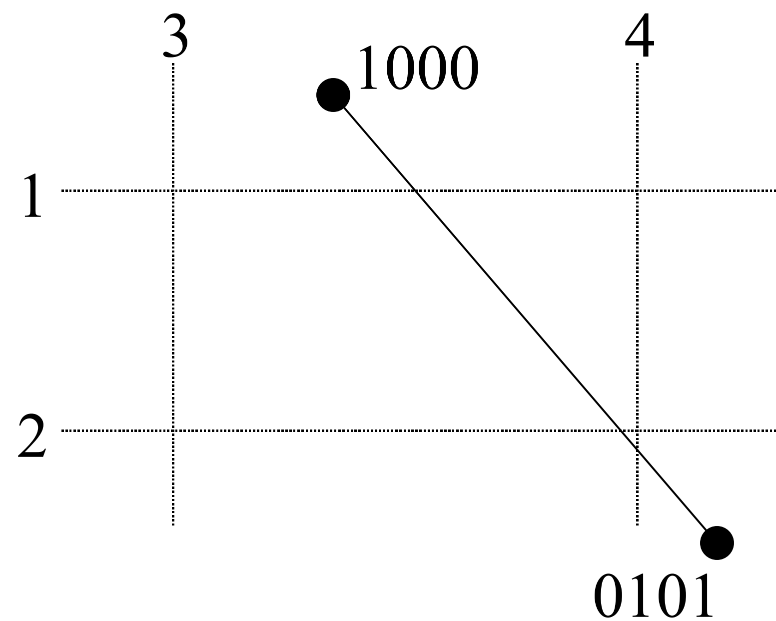


Cohen-Sutherland



Cohen-Sutherland - Detalhes

- Recorte só é necessário se um vértice estiver dentro e outro estiver fora
- Classificação de cada vértice pode ser codificada em 4 bits, um para cada semi-plano
  - Dentro = 0 e Fora = 1
- Rejeição trivial:
  - Classif(P_1) & Classif(P_2) $\neq 0$
- Aceitação trivial:
  - Classif(P_1) | Classif(P_2) = 0
- Interseção com quais semi-planos?
  - Classif(P_1) ^ Classif(P_2)



Algoritmo de Liang-Barsky

- Refinamento que consiste em representar a reta em forma paramétrica
- É mais eficiente visto que não precisamos computar pontos de interseção irrelevantes
- Porção da reta não recortada deve satisfazer

$$x_{\min} \leq x_1 + t\,\Delta\,\mathrm{x} \leq x_{\max} \qquad \Delta\, x = x_2 - x_1$$

$$y_{\min} \leq y_1 + t\,\Delta\,\mathrm{y} \leq y_{\max} \qquad \Delta\, y = y_2 - y_1$$

Algoritmo de Liang-Barsky

- Linha infinita intercepta semi-espaços planos para os seguintes valores do parâmetro *t*:

$$t_k = \frac{q_k}{p_k}$$

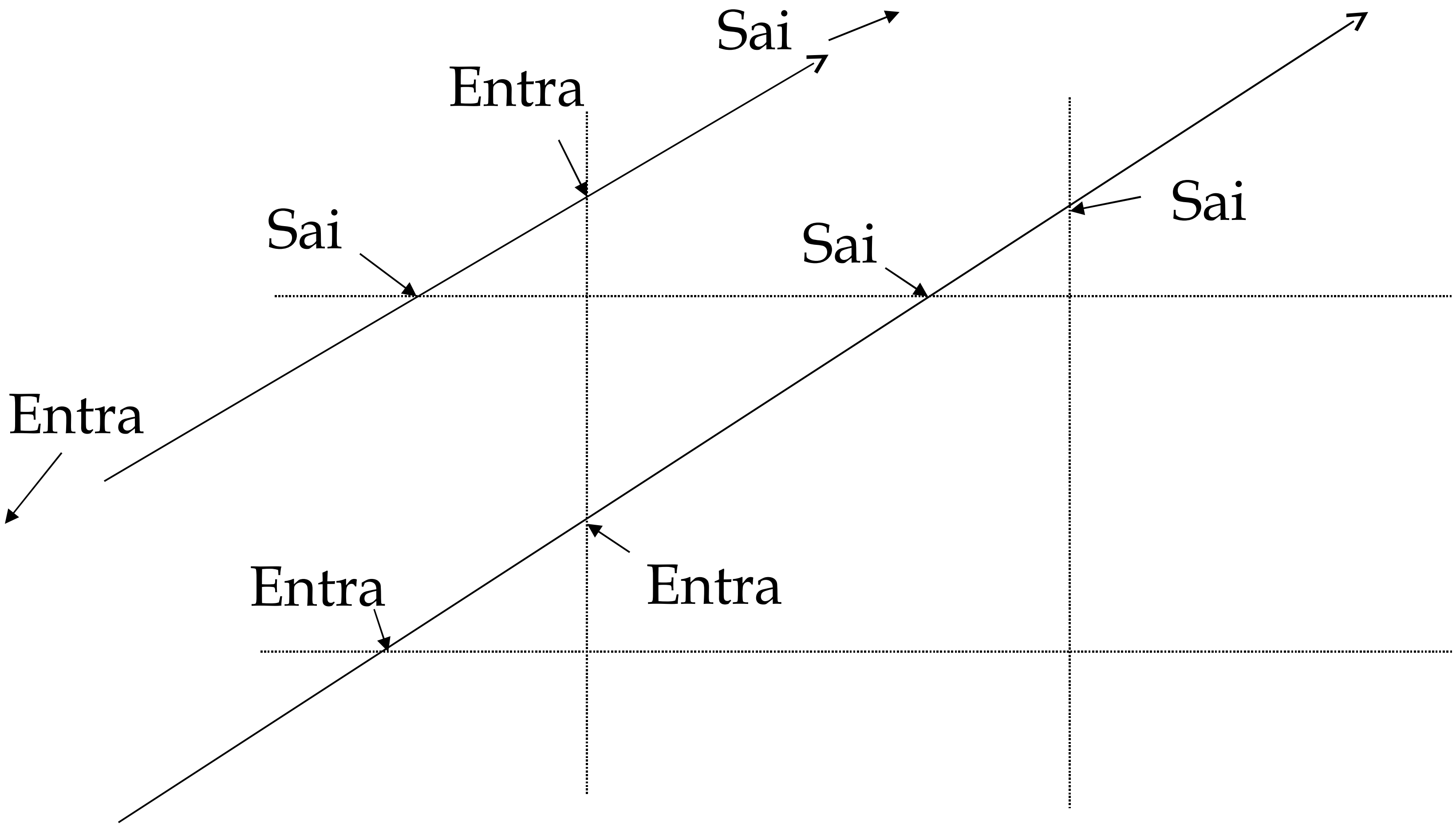
onde

$$\begin{aligned}
p_1 &= -\Delta x & q_1 &= x_1 - x_{\min} \\
p_2 &= \Delta x & q_2 &= x_{\max} - x_1 \\
p_3 &= -\Delta y & q_3 &= y_1 - y_{\min} \\
p_4 &= \Delta y & q_4 &= y_{\max} - y_1
\end{aligned}$$

Algoritmo de Liang-Barsky

- Se $p_k < 0$, à medida que t aumenta, reta **entra** no semi-espaço plano
- Se $p_k > 0$, à medida que t aumenta, reta **sai** do semi-espaço plano
- Se $p_k = 0$, reta é paralela ao semi-espaço plano (recorte é trivial)
- Se existe um segmento da reta dentro do retângulo, classificação dos pontos de interseção deve ser **entra, entra, sai, sai**

Algoritmo de Liang-Barsky



Liang-Barsky – Pseudo-código

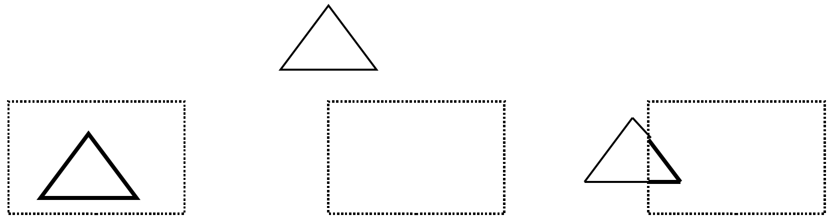
- Computar valores de *t* para os pontos de interseção
- Classificar pontos em **entra** ou **sai**
- Vértices do segmento recortado devem corresponder a dois valores de *t:*
  - t_{min}= max (0*, t*'s do tipo **entra**)
  - t_{max}= min (1*, t*'s do tipo **sai**)
- Se $t_{min}<t_{max}$, segmento recortado é não nulo
  - Computar vértices substituindo os valores de *t*
- Na verdade, o algoritmo calcula e classifica valores de *t* um a um
  - Rejeição precoce
    - Ponto é do tipo **entra** mas $t > 1$
    - Ponto é do tipo **sai** mas $t < 0$

Recorte de Polígono contra Retângulo

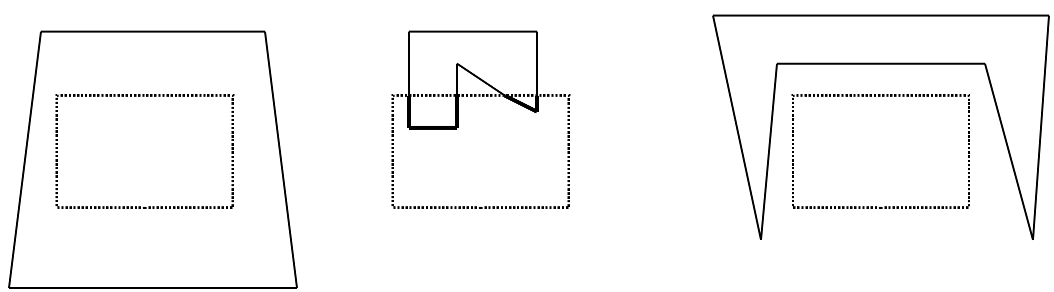
- Inclui o problema de recorte de segmentos de reta
  - Polígono resultante tem vértices que são
    - Vértices da janela,
    - Vértices do polígono original, ou
    - Pontos de interseção aresta do polígono/aresta da janela
- Dois algoritmos clássicos
  - Sutherland-Hodgman
    - Figura de recorte pode ser qualquer polígono convexo
  - Weiler-Atherton
    - Figura de recorte pode ser qualquer polígono

Recorte de Polígono contra Retângulo

- Casos Simples

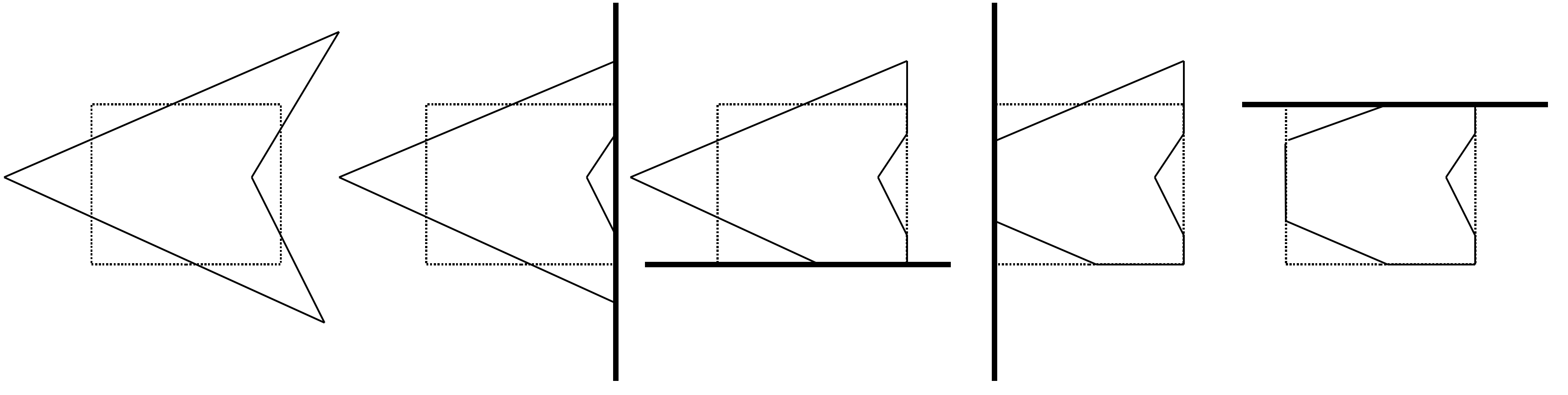


- Casos Complicados



Algoritmo de Sutherland-Hodgman

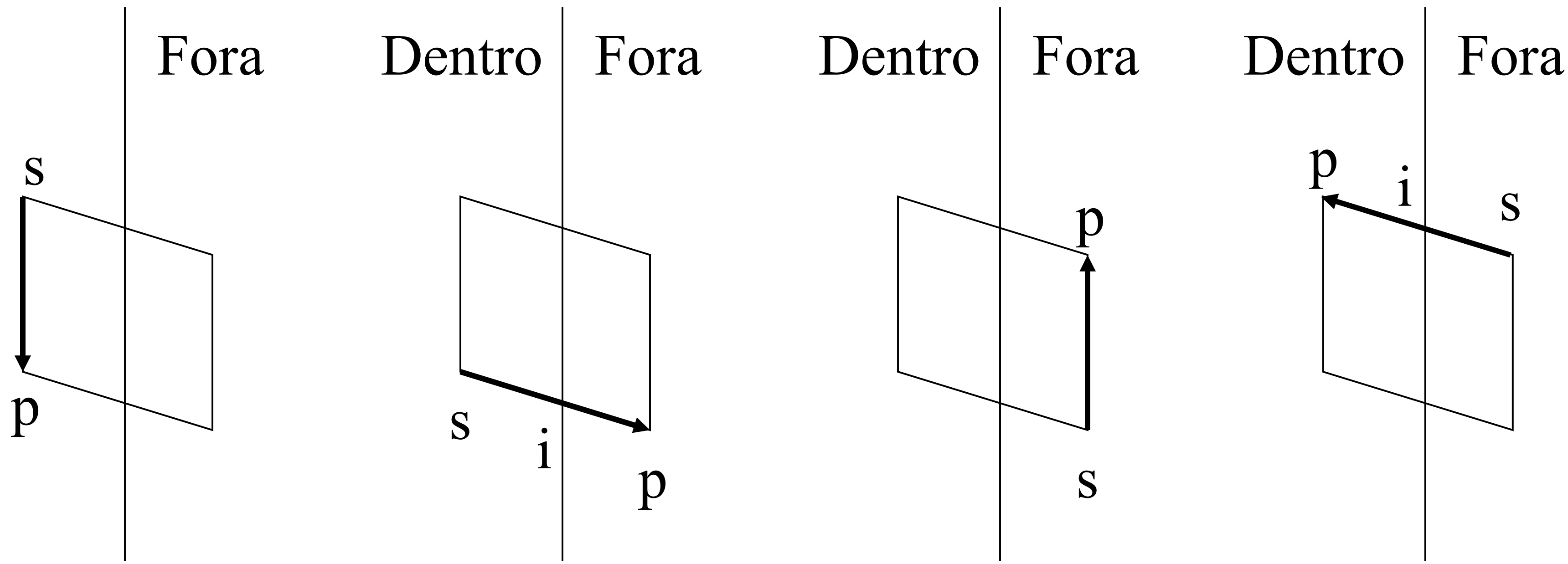
- Idéia é semelhante à do algoritmo de Sutherland-Cohen
  - Recortar o polígono sucessivamente contra todos os semi-espaços planos da figura de recorte



Algoritmo de Sutherland-Hodgman

- Polígono é dado como uma lista circular de vértices
- Vértices e arestas são processados em seqüência e classificados contra o semi-espaço plano corrente
  - Vértice:
    - Dentro: copiar para a saída
    - Fora: ignorar
  - Aresta
    - Intercepta semi-espaço plano (vértice anterior e posterior têm classificações diferentes) : Copiar ponto de interseção para a saída
    - Não intercepta: ignorar

Algoritmo de Sutherland-Hodgman



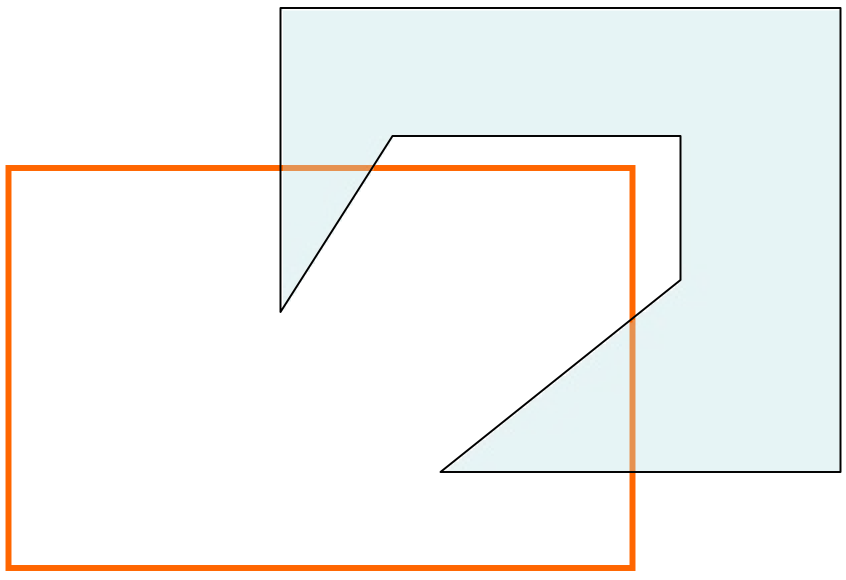
Copiar p

Copiar i

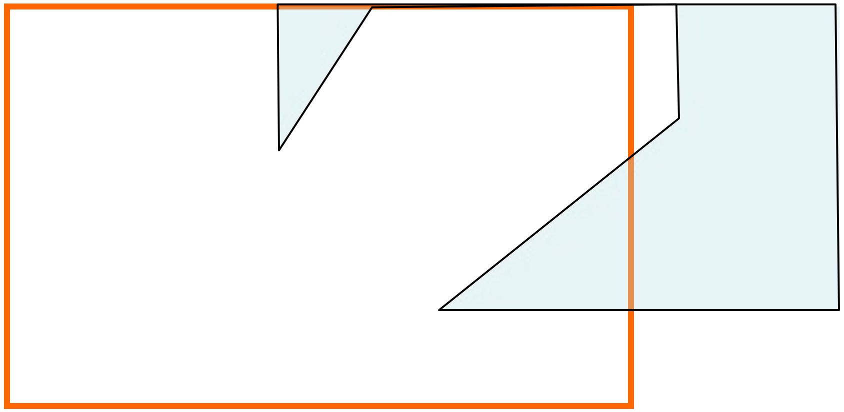
Ignorar

Copiar i,p

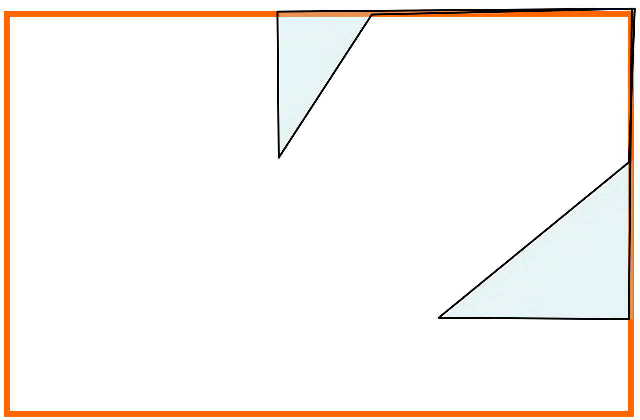
Sutherland-Hodgman – Exemplo



Sutherland-Hodgman – Exemplo

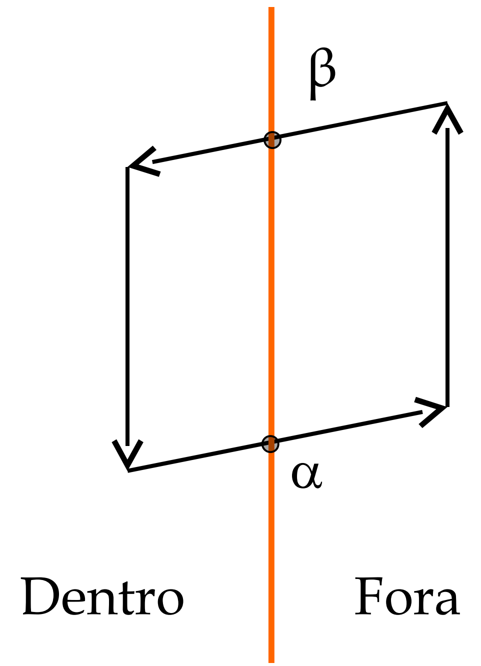


Sutherland-Hodgman – Exemplo

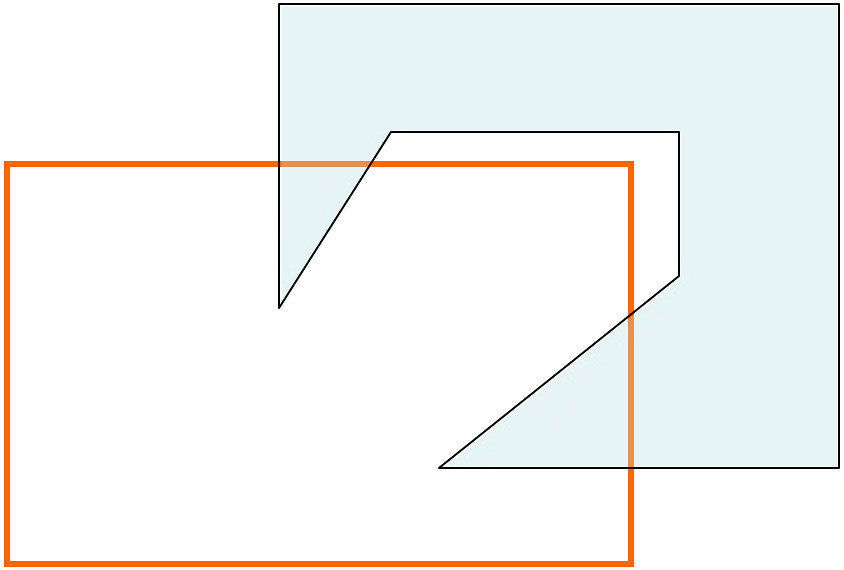


Sutherland Hodgman – Eliminando Arestas Fantasmas

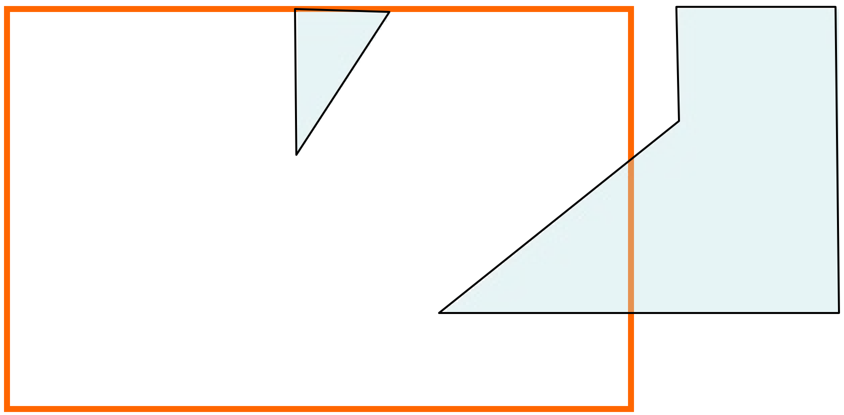
- Distinguir os pontos de interseção gerados
  - De dentro para fora: rotular como do tipo α
  - De fora para dentro: rotular como do tipo β
- Iniciar o percurso de algum vértice "fora"
- Ao encontrar um ponto de interseção α, ligar com o último β visto
- Resultado pode ter mais de uma componente conexa



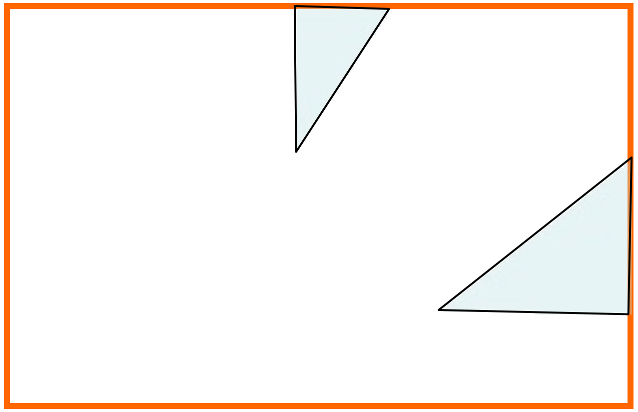
Sutherland Hodgman – Eliminando Arestas Fantasmas – Exemplo



Sutherland Hodgman – Eliminando Arestas Fantasmas – Exemplo



Sutherland Hodgman – Eliminando Arestas Fantasmas – Exemplo



Sutherland-Hodgman - Resumo

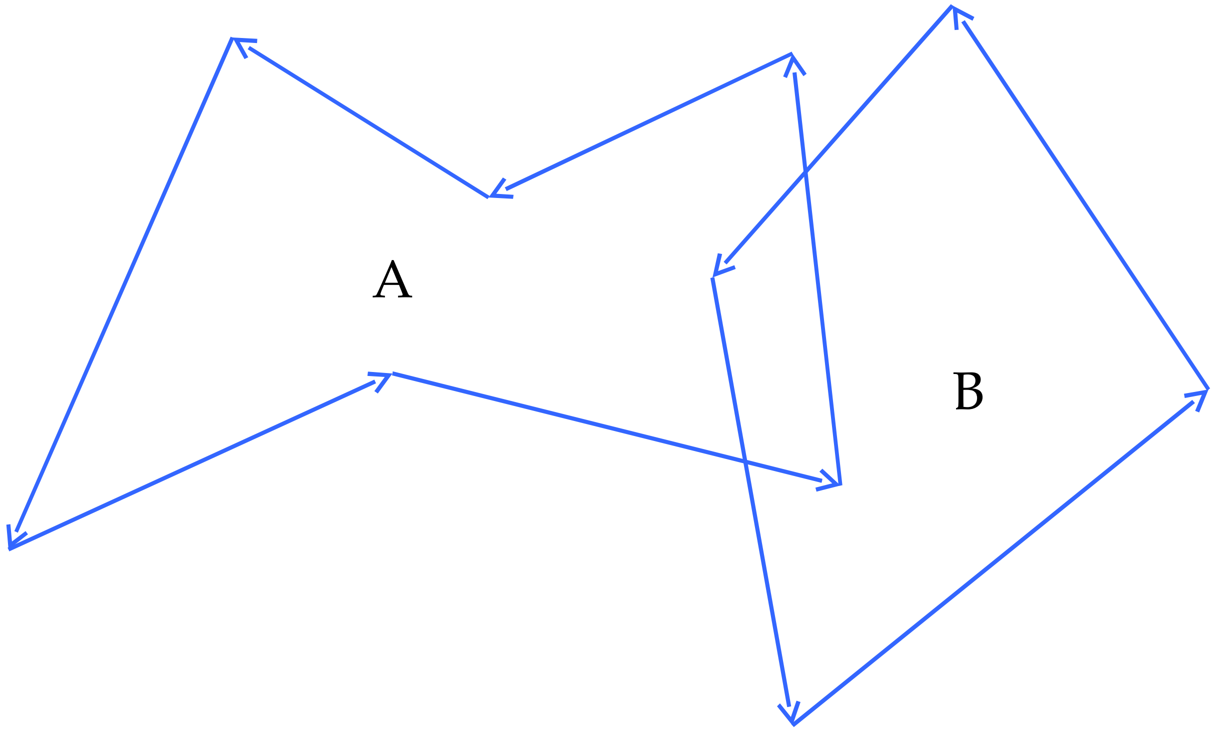
- Facilmente generalizável para 3D
- Pode ser adaptado para implementação em hardware
  - Cada vértice gerado pode ser passado pelo pipeline para o recorte contra o próximo semi-espaço plano
- Pode gerar arestas "fantasma"
  - Irrelevante para propósitos de desenho
  - Podem ser eliminadas com um pouco mais de trabalho

Algoritmo de Weiler-Atherton

- Recorta qualquer polígono contra qualquer outro polígono
- Pode ser usado para computar operações de conjunto com polígonos
  - União, Interseção, Diferença
- Mais complexo que o algoritmo de Sutherland-Hodgman
- Idéia:
  - Cada polígono divide o espaço em 3 conjuntos
    - Dentro, fora, borda
  - Borda de cada polígono é “duplicada”
    - Uma circulação corresponde ao lado de dentro e outra ao lado de fora
  - Nos pontos de interseção, é preciso “costurar” as 4 circulações de forma coerente

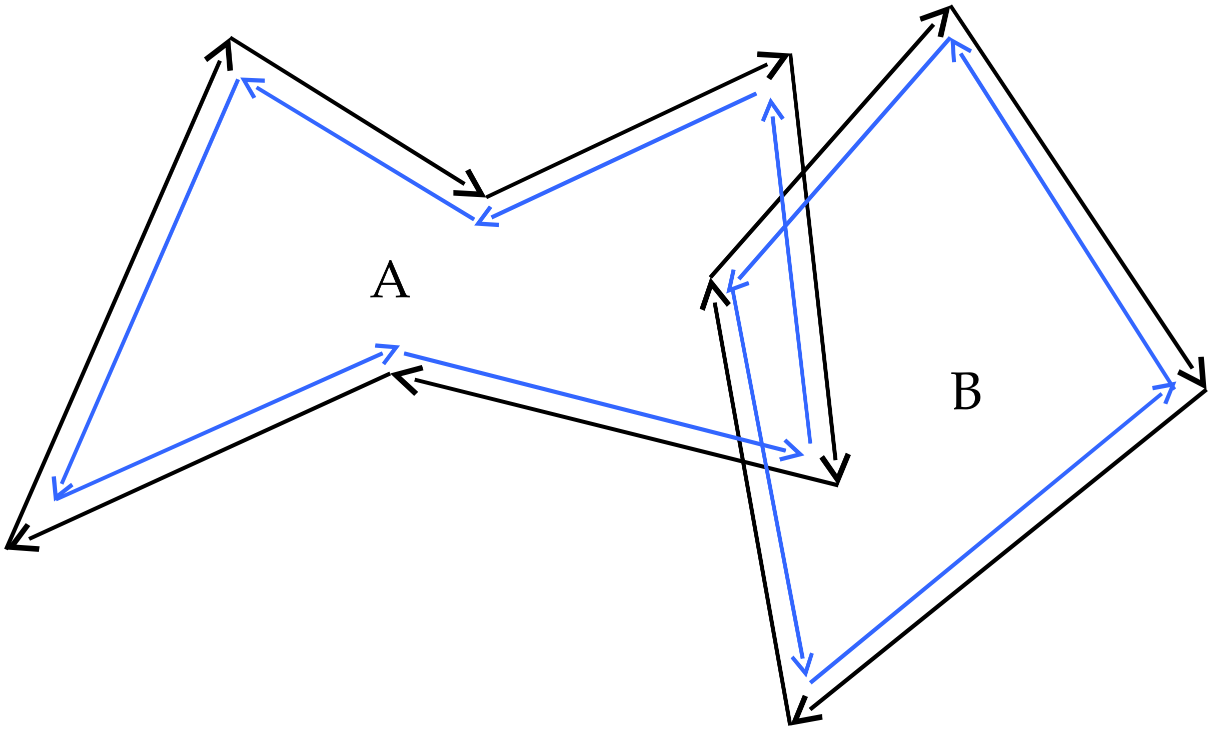
Algoritmo de Weiler-Atherton

Interior do polígono à esquerda da seta
(circulação anti-horária)



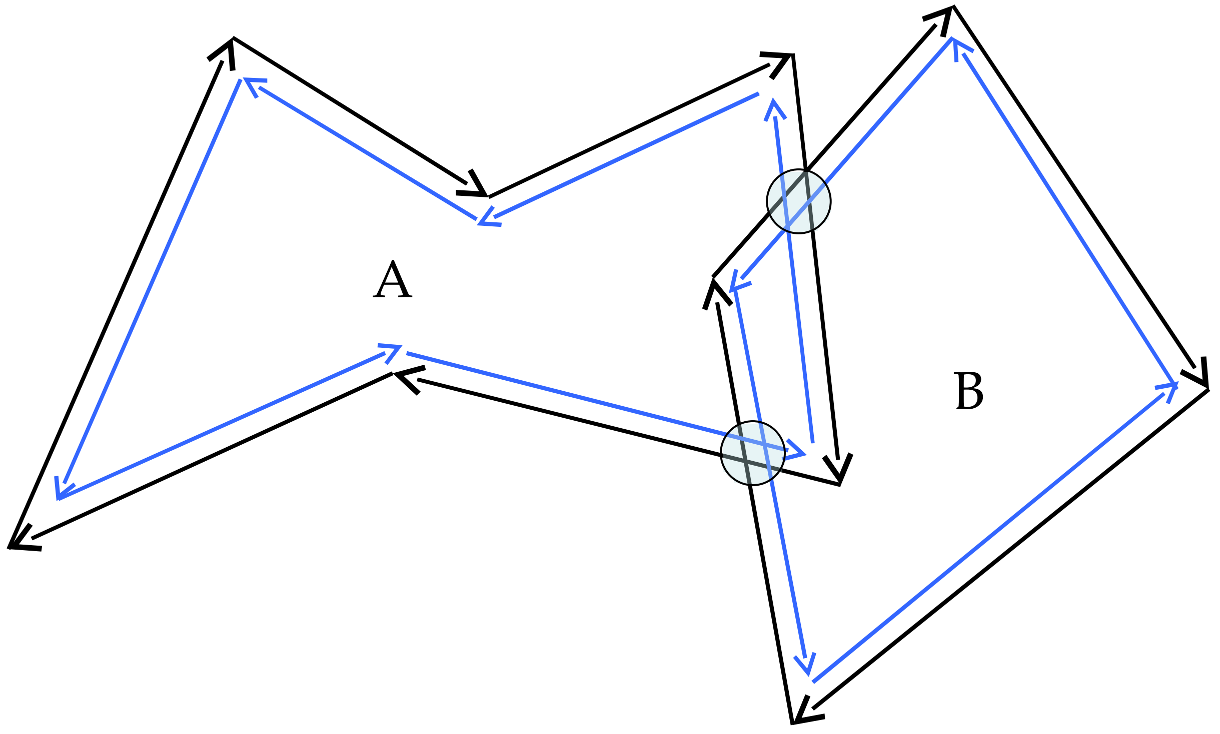
Algoritmo de Weiler-Atherton

Exterior do polígono à direita da seta
(circulação horária)



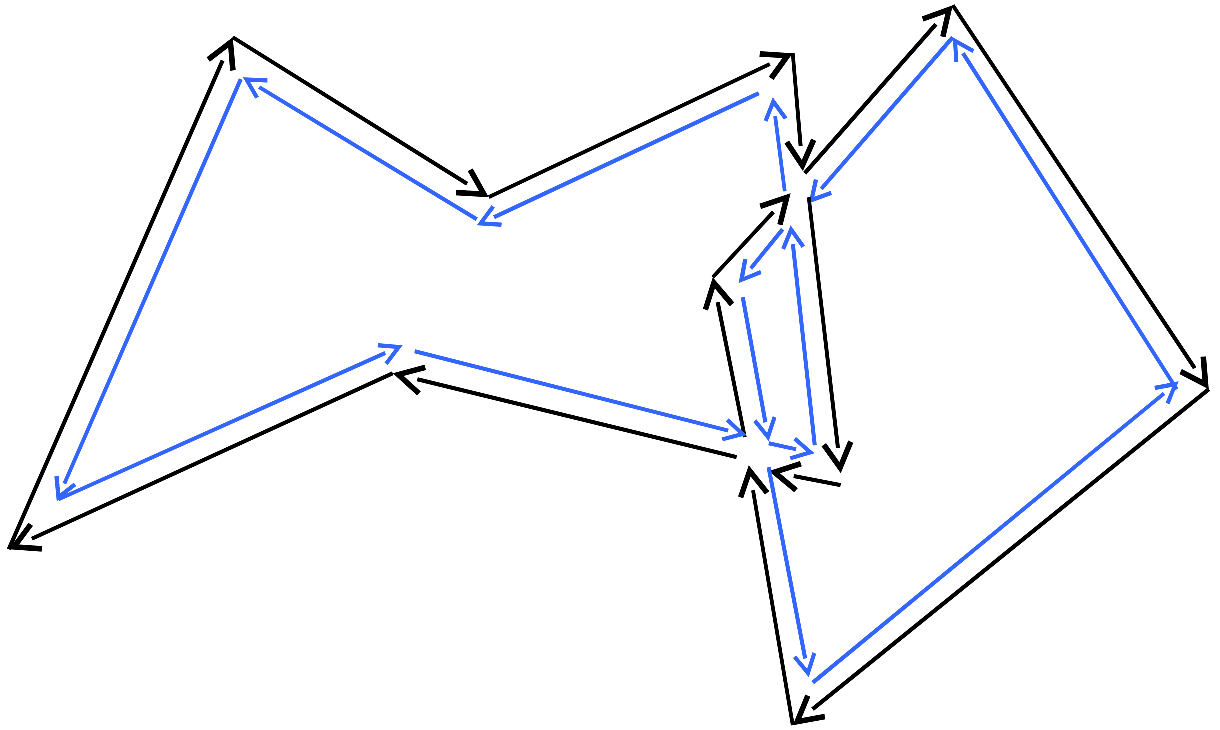
Algoritmo de Weiler-Atherton

Pontos de interseção são calculados



Algoritmo de Weiler-Atherton

Circulações são costuradas



Algoritmo de Weiler-Atherton

Circulações são classificadas