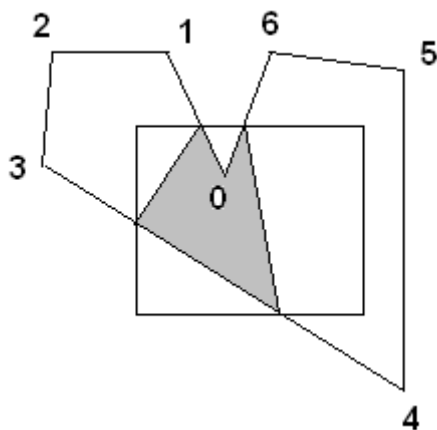
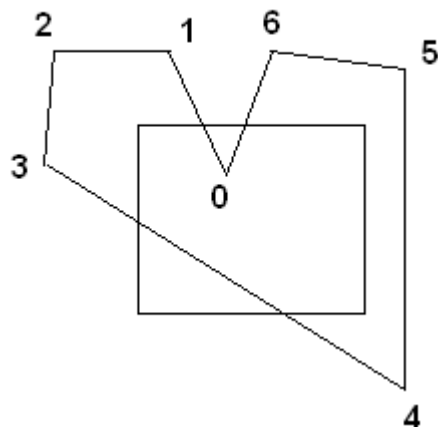


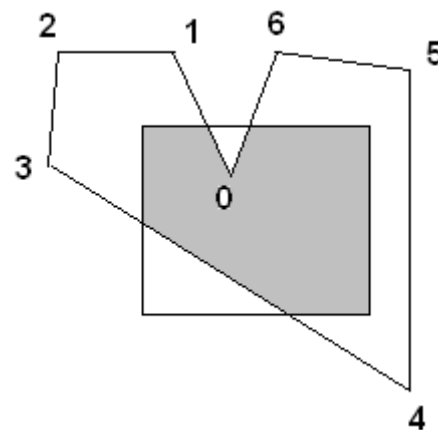
Decuparea poligoanelor

Prof. univ. dr. ing. Florica Moldoveanu

Decuparea poligoanelor 2D

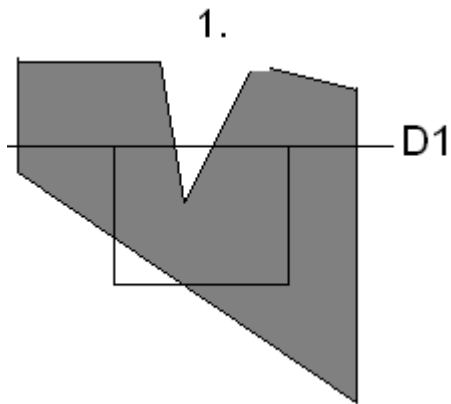


Rezultatul obtinut daca se decupeaza pe rand laturile poligonului folosind algoritmul de decupare vectori.

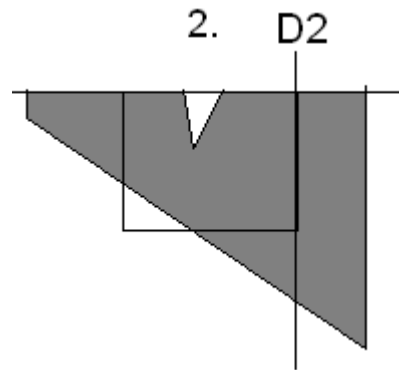


Rezultatul corect.

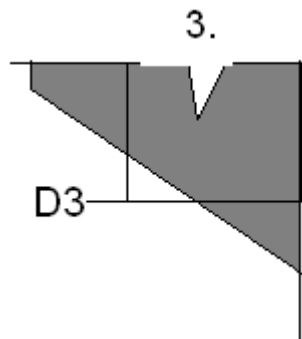
Algoritmul Sutherland-Hodgman(1)



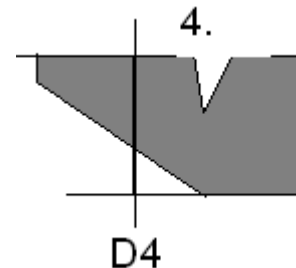
1. Se decupeaza laturile poligonului fata de dreapta pe care se afla o latura a dreptunghiului de decupare, D1
- Se creaza lista varfurilor poligonului rezultat, V1



2. Se decupeaza poligonul cu varfurile V1 fata de dreapta D2
- Se creaza lista varfurilor poligonului rezultat, V2



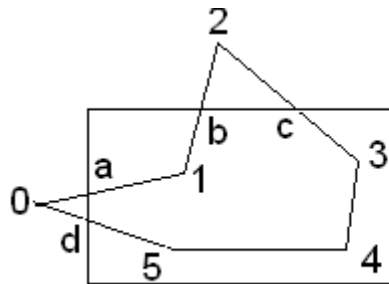
3. Se decupeaza poligonul cu varfurile V2 fata de dreapta D3
- Se creaza lista varfurilor poligonului rezultat, V3



4. Se decupeaza poligonul cu varfurile V3 fata de dreapta D4
- Se creaza lista varfurilor poligonului rezultat, V4

Algoritmul Sutherland-Hodgman(2)

Calculul varfurilor poligonului rezultat intr-o etapa a decuparii



0-1: a, 1 --> a, 1

1-2: 1,b --> a, 1, 1, b

2-3: c,3 --> a, 1, 1, b, c, 3

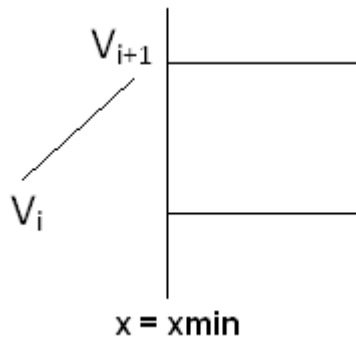
3-4: 3,4 --> a,1, 1, b, c, 3, 3, 4

4-5: 4,5 --> a,1, 1, b, c, 3, 3, 4, 4, 5

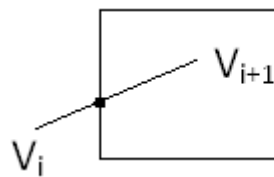
5-0: 5,d --> a,1, 1, b, c, 3, 3, 4, 4, 5, 5, d

Lista corecta: a, 1, b, c, 3, 4, 5, d

- Conventie:**
- Fiecare latura a poligonului se considera un vector: $V_i - V_{i+1}$
 - Numai varful V_i apartine laturii

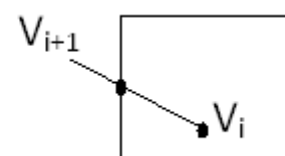


0 varfuri

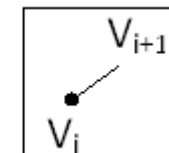


un varf

(punctul de intersectie)



2 varfuri



un varf

Algoritmul Sutherland-Hodgman(3)

```
int DecupLatura(P2D v1, P2D v2, P2d * v1d, P2D *v2d, int latura)
```

```
// decupeaza o latura a poligonului, v1-v2, fata de dreapta pe care se afla o latura a drept. de decupare
```

```
{ // functia intoarce numarul de varfuri rezultate din decupare
```

```
switch(latura)
```

```
{ case 1: // dreapta y = ymax
```

```
    if(v1.y > ymax && v2.y > ymax) return 0; // latura in afara drept. decupare
```

```
    if(v1.y <= ymax && v2.y <= ymax) { *v1d = v1; return 1;} // latura in interiorul drept. decupare
```

```
    if(v1.y <= ymax && v2.y > ymax) / latura iese din drept. de decupare
```

```
        { * v1d = v1; x_intersectie = ...; v2d->x = x_intersectie; v2d->y = ymax; return 2;}
```

```
    else// v1.y > ymax && v2.y <= ymax // latura intra in drept. de decupare
```

```
        { x_intersectie = ....; v1d->x = x_intersectie; v1d->y = ymax; return 1;}
```

```
case 2: // latura x=xmax
```

```
.....
```

```
}
```

```
}
```

Algoritmul Sutherland-Hodgman(4)

```
void DecupLaturi(int nv, P2D *vrf, int *nd, P2D *vrfd, int latura)  
{ // decupeaza toate laturile poligonului fata de dreapta pe care se afla o latura a dreptunghiului  
  // vrf: lista varfurilor poligonului de decupat  
  // nv: numarul de varfuri ale poligonului de decupat  
  // vrfd: lista varfurilor poligonului rezultat din decupare  
  // nd: numarul de varfuri ale poligonului rezultat din decupare  
  
  *nd = 0;  
  
  for(int i=0; i< nv; i++)  
    *nd+= DecupLatura(vrf[i], vrf[i+1], &vrfd[*nd], &vrfd[*nd+1], latura)  
  
  *nd+= DecupLatura(vrf[nv-1], vrf[0], &vrfd[*nd], &vrfd[*nd + 1], latura);  
}
```

Algoritmul Sutherland-Hodgman(5)

```
void DecupPoligon(int nv, P2D * vrf, int *nd, P2D **vrfd)
```

```
{ // vrf: lista varfurilor poligonului de decupat
```

```
  // vrfd: lista varfurilor poligonului rezultat din decupare
```

```
    P2D * v1 = new P2D[2*nv]; P2D * v2 = new P2D[2*nv]; // se folosesc 2 liste de varfuri
```

```
    memcpy(v1, vrf, nv*sizeof(P2D));
```

```
    for(int latura =1 ; latura <= 4; latura+=2)
```

```
    { DecupLaturi(nv, v1, nd, v2, latura);
```

```
      DecupLaturi(*nd, v2, &nv, v1, latura+1);
```

```
    }
```

```
    *vrfd = new P2D[(*nd)];
```

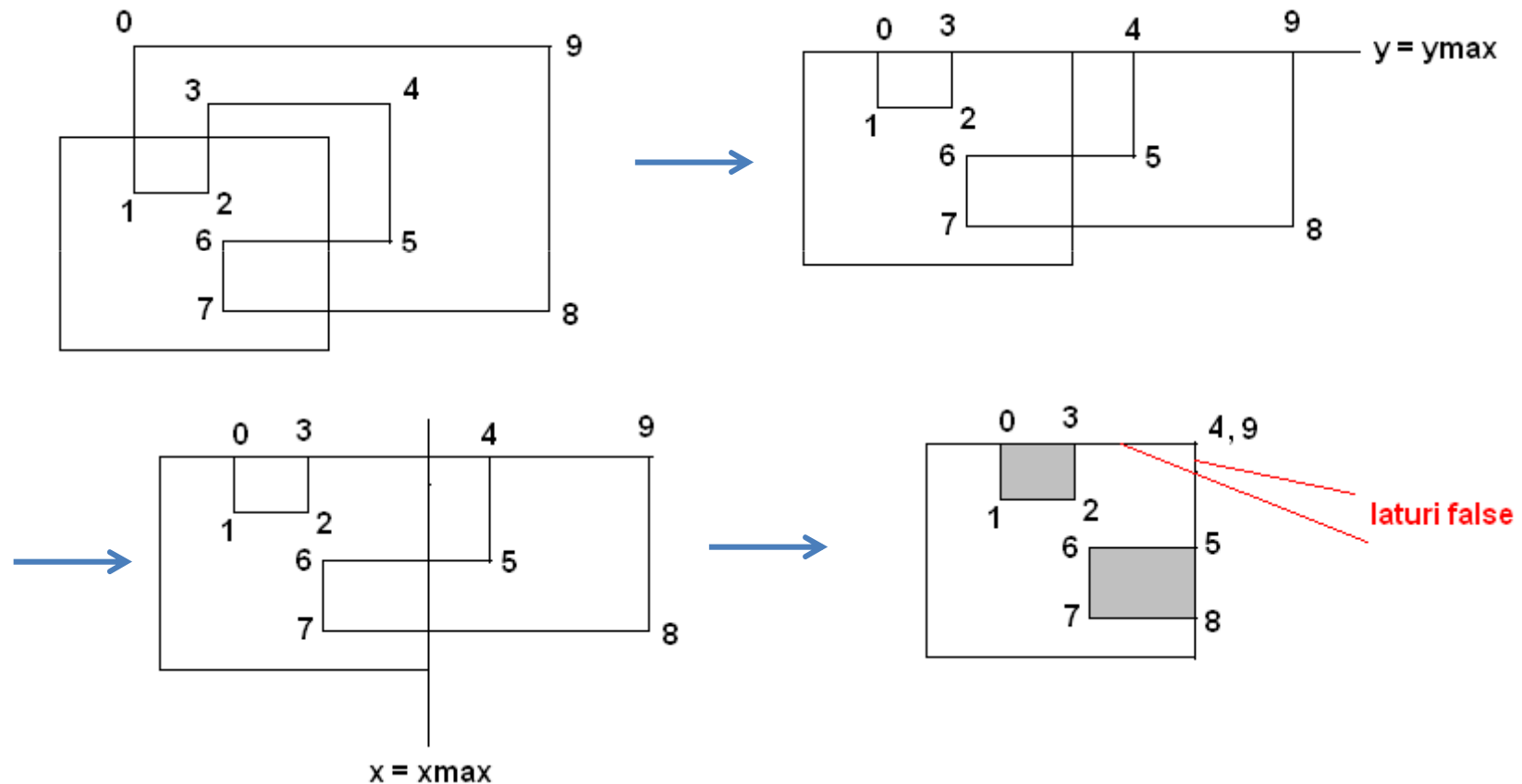
```
    memcpy(*vrfd, v1, (*nd)*sizeof(P2D));
```

```
    delete v1; delete v2;
```

```
}
```

Algoritmul Sutherland-Hodgman(6)

Limitarea algoritmului Sutherland-Hodgman: la decuparea poligoanelor concave, atunci cand din decupare pot rezulta mai multe poligoane. Algoritmul produce un singur poligon, cu posibile laturi false.



Algoritmul Weiler - Atherton(1)

Permite decuparea unui poligon oarecare (poligonul subiect) fata de un poligon oarecare (poligonul de decupare). Notam cu:

PS- poligonul subiect

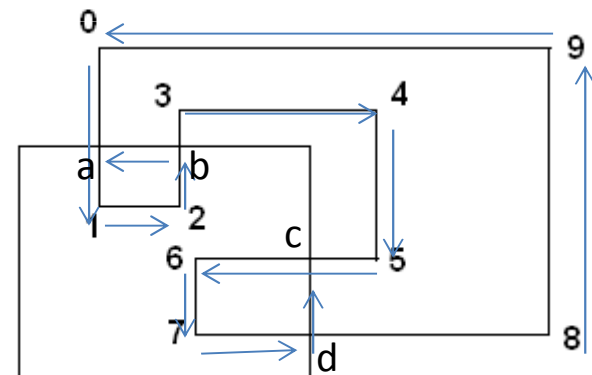
PD-poligonul de decupare

PR- poligonul rezultat – acesta poate fi alcatuit din mai multe cicluri de varfuri(poligoane)

LS – lista de varfuri a PS

LV - lista de varfuri a PR

Cicluri de varfuri: a, 1, 2, b, a; c, 6, 7, d, c



Algoritmul Weiler - Atherton(2)

Se initializeaza LV. Se alege un varf al PS exterior PD. Acesta este Vstart (varful din care incepe parcurgerea conturului poligonului PS).

Vcurent <- Vstart; gata = false.

repetă

*Se parcurge conturul poligonului PS incepand din Vcurent, pana cand se ajunge in Vstart sau intr-un punct Pi de intersectie PS cu PD.

daca s-a ajuns in Vstart

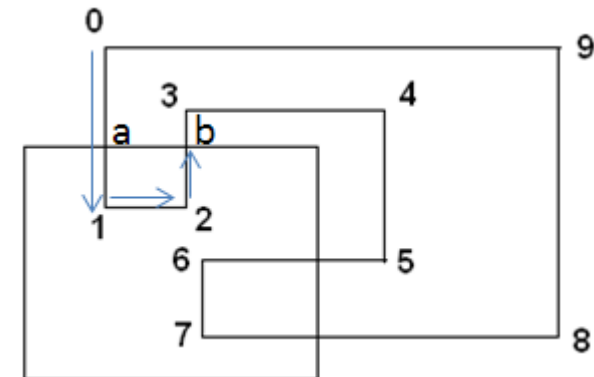
gata = true; break;

altfel

1. Se adauga Pi in LV si se continua

parcurgerea pe conturul PS pana intr-un

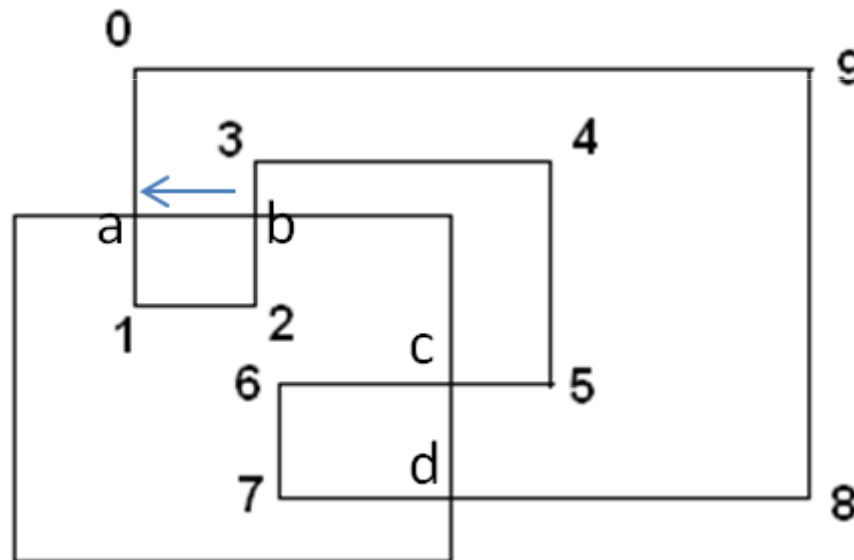
punct Pe de iesire din PD, memorand in LV varfurile PS intalnite pe parcurs si punctul Pe.



LV: a, 1, 2, b

Algoritmul Weiler - Atherton(3)

2. $V_{curent} \leftarrow$ varful exterior al laturii PS care intersecteaza PD in Pe. // vrf. 3
3. Se continua parcurgerea pe conturul PD din Pe pana cand se ajunge intr-un punct, P_i , de intersectie cu PS. Se adauga in LV varfurile PD intalnite pe parcurs si punctul P_i .
4. Daca s-a format un ciclu de varfuri, se memoreaza ciclul de varfuri si se initializeaza LV. altfel, $V_{curent} \leftarrow P_i$ (in inlocuieste in LS vrf. 3 cu P_i).



pana(gata)

In iteratia urmatoare se formeaza ciclul de varfuri
c2: c, 6, 7, d, c

LV: a, 1, 2, b, a

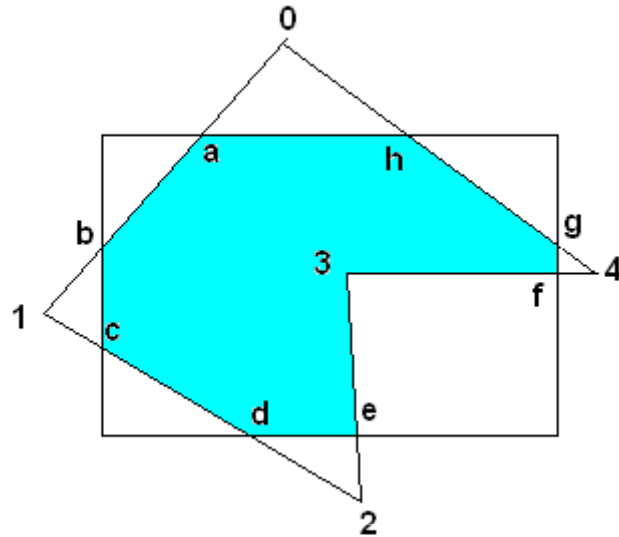


Ciclul de varfuri c1: a, 1, 2, b, a



Lv: vida

Algoritmul Weiler - Atherton(4)



Se porneste din varful 0

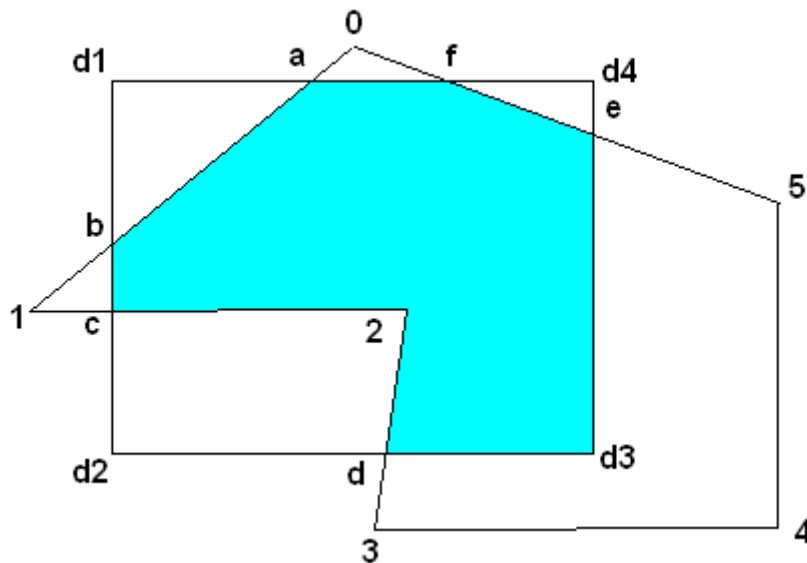
LV: a, b

a, b, c (se inlocuieste in LS vrf. 1 cu c)

a, b, c, d, e (se inlocuieste in LS vrf. 2 cu e)

a, b, c, d, e, 3, f, g (se inlocuieste in LS vrf. 4 cu g)

a, b, c, d, e, 3, f, g, h, a



Se porneste din varful 0

LV: a, b

a, b, c (se inlocuieste in LS vrf. 1 cu c)

a, b, c, 2, d, d3, e (se inlocuieste in LS vrf. 5 cu e)

a, b, c, 2, d, d3, e, f, a

Cazuri particulare

Cazurile particulare:

- dreptunghiul/poligonul de decupare inclus in poligonul decupat (subiect) si
- poligonul decupat inclus in dreptunghiul /poligonul de decupare

se trateaza separat, prin teste efectuate inainte de executia algoritmului de decupare.

1. Daca decuparea se efectueaza față de suprafata unui dreptunghi cu laturile paralele cu axele, atunci pentru a verifica daca poligonul de decupat este inclus in dreptunghiul de decupare, se efectueaza asupra fiecarui varf al sau, $V[i]$, testul:

$xmin_drept \leq V[i].x \leq xmax_drept \ \&\& \ ymin_drept \leq V[i].y \leq ymax_drept$

2. Pentru a verifica daca dreptunghiul/poligonul de decupare este inclus in poligonul de decupat se efectueaza asupra fiecarui varf al dreptunghiului/poligonului de decupare testul de “punct interior unui poligon” (Curs 11: Rasterizarea suprafetelor poligonale).

Analog se verifica daca poligonul de decupat este inclus in poligonul de decupare.