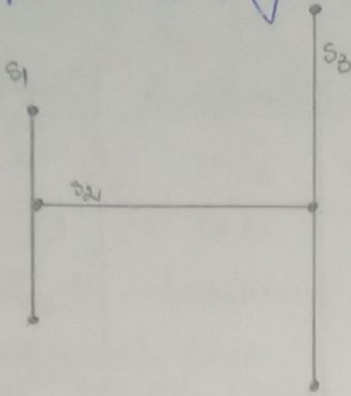
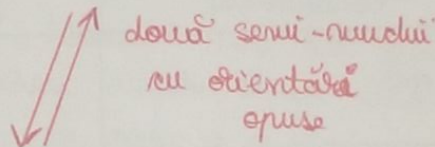
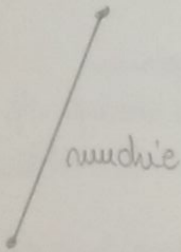


GEOMETRIE COMPUTAȚIONALĂ

caz particular (degenerat)



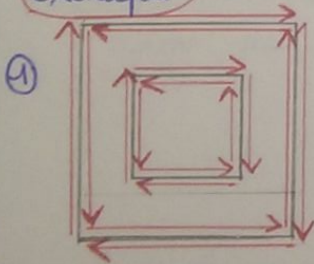
Eveniment	Statut	Evenimente noi
u_{s_3}	$\{s_3\}$	—
u_{s_1}	$\{s_1, s_3\}$	—
u_{s_2}	$\{s_1, s_2, s_3\}$	$s_1 \cap s_2$ și $s_2 \cap s_3$
$s_1 \cap s_2$	$\{s_1, s_2, s_3\}$	—
$s_2 \cap s_3$	$\{s_1, s_2, s_3\}$	—
d_{s_2}	$\{s_1, s_3\}$	—

- SUPRAPUNEREA STRATURILOR TEMATICE -- (OVERLAY) -CONCEPTUL CHEIE este de semi-muchie (muchie orientată) - half-edge

Dat un poligon (eventual cu goluri):

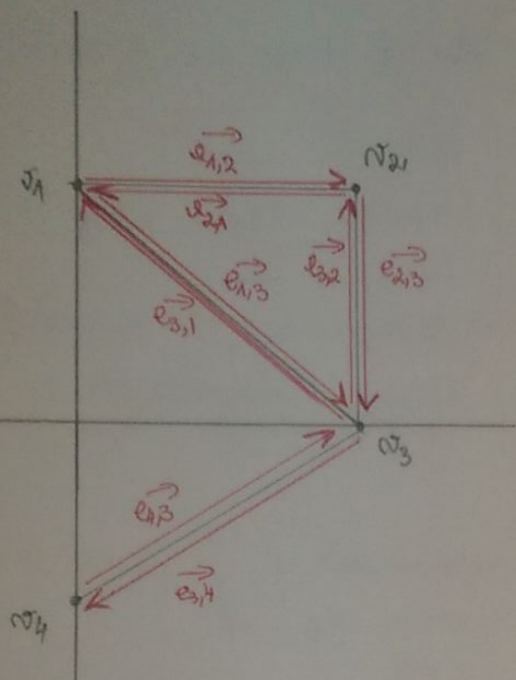
→ frontiera exterioară poate fi parcursă cu ajutorul semi-muchilor astfel încât poligonul să fie la stânga frontierei și virajele să fie la stânga.

→ dacă componența a frontierei interioare: poligonul este la stânga, dar frontiera e parcursă cu viraje la dreapta.

Exemple

8 vârfuri
16 semi-muchii
3 fețe

② Exemplu de subdiviziune planară și lista dublu înălțuită (DCEL = ~~double~~ doubly connected edge list) asociată:



Vârful	Coordonate	Incident Edge
v_1	(0,4)	$\vec{e_{1,2}}$
v_2	(4,4)	$\vec{e_{2,3}}$
v_3	(4,0)	$\vec{e_{3,4}}$
v_4	(0,0)	$\vec{e_{4,1}}$

Față	outer Component	Inner Component
f_1 (exterior)	nil	$\vec{e_{1,2}}$
$f_2 (\Delta)$	$\vec{e_{2,1}}$	nil

semii-nmuchie	Origine	Termin	Incident Face	Next	Previous
$\vec{e_{1,2}}$	v_1	$\vec{e_{2,1}}$	f_1	$\vec{e_{2,3}}$	$\vec{e_{3,1}}$
$\vec{e_{2,3}}$	v_2	$\vec{e_{3,2}}$	f_1	$\vec{e_{3,4}}$	$\vec{e_{4,2}}$

Exercițiu: Explica cum, folosind pointerii de mai sus:

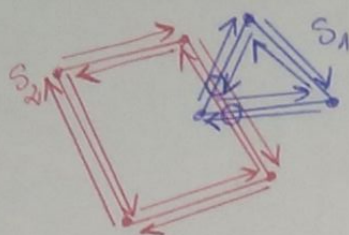
- poate fi parcursă frontiera unei fețe (exteriore/interioare)
- pot fi găsite toate semii-nmuchii incidente cu un vârf (cu un sens se face parcurgerea)

CONCEPTE

- Conceptul de INCIDENȚĂ $\begin{cases} \rightarrow \text{vârfuri / semii-nmuchii} \\ \rightarrow \text{vârfuri / fețe} \end{cases}$
- Conceptul de COMPLEXITATE a unei subdiviziuni - nr. de vârfuri + nr. de fețe + nr. de muchii

OVERLAY-UL (suprapunerea) dintre două subdiviziuni

S_1, S_2 subdiviziuni \rightarrow "overlay" $O(S_1, S_2)$



! Modul de actualizare al semi-nuchiilor (algoritmul)

~~Algoritmul~~
 \rightarrow după copiere (PAS 1):

- nu apar noile vârfuri
- există semi-nuchi care trebuie legate de noile vârfuri
- există fețe nou apărute

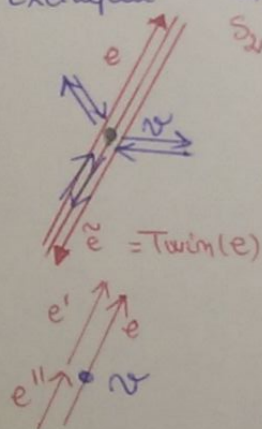
\rightarrow PAS 2:

- detectarea intersecțiilor de segmente trebuie completată cu
 - \rightarrow actualizarea listei de semi-nuchi (local; tot un PAS 2)
 - \rightarrow actualizarea fetelor (se face mai târziu în PAS 3-PAS 6)

Esential pentru actualizarea "locală" este că semi-nuchiile existente pot fi folosite la introducerea unor semi-nuchi noi, bazată pe următorul principiu: fata pe care o nădărmese este la stânga lor

Actualizarea listei de nuchi semi-nuchi

Exemplu S_1 | prim vârful v al lui S_1 trece o nuchie
 S_2 | a lui S_2 (exercițiu: imaginați și alte situații!)



În acest caz:

\rightarrow semi-nuchiile lui S ~~trebuie~~ pot fi "actualizate"
 \rightarrow cele două semi-nuchi din S_2 (e și e') sunt înlocuite cu 4 semi-nuchi

e' $origin = v$
 $Twin = \rightarrow e$
 $Next = Next(e)$
 $Prev = \Delta$

e'' $origin = origin(e)$
 $Twin = \rightarrow e$
 $Next = \Delta$
 $Prev = Prev(e)$

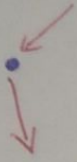
Similar pentru e

Next pentru e' și $Prev$.
pentru e' se găsește parcurgând nuchiile lui S_1 din v (pentru $Next(e')$ este luată cea mai apropiată în sensul acelor de ceasornic;) similar ptr. $Prev(e'')$

⇒ PAS 3

4) Pentru un vâlc de frontieră cum stabilim dacă este interioară (i.e. frontiera unui gol) sau exterioară?

R: Cu ajutorul vârfului celui mai din stânga și a tipului de viraj.



dacă e viraj la stânga
e frontieră exterioară



dacă e viraj la dreapta
e frontieră interioară

În acest moment știu câte fețe are area în overlay: nr de frontiere exterioare + 1.