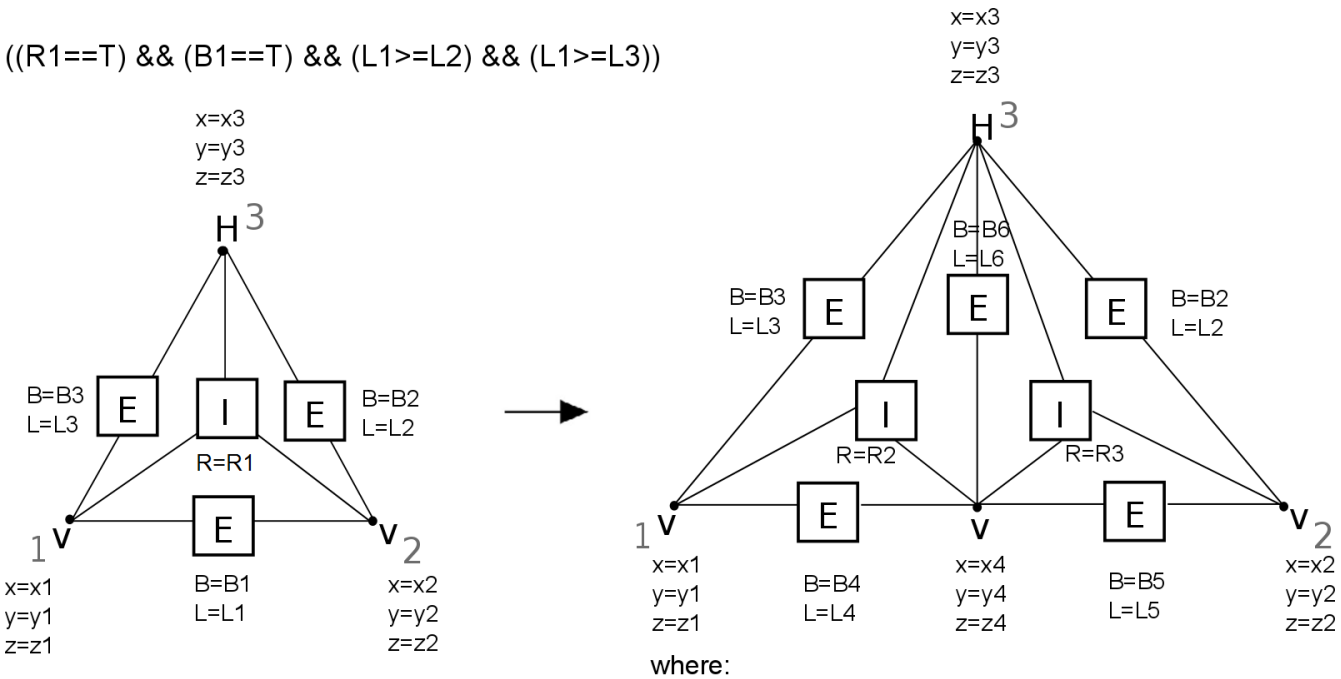
Graph transformations modeling terrain topography for CAD/CAE Systems
Anna Paszynska & Maciej Paszynski & Krzysztof Podsiadło (c)

1. Please implement graph transformation (P1)

((R1==T) && (B1==T) && (L1>=L2) && (L1>=L3))

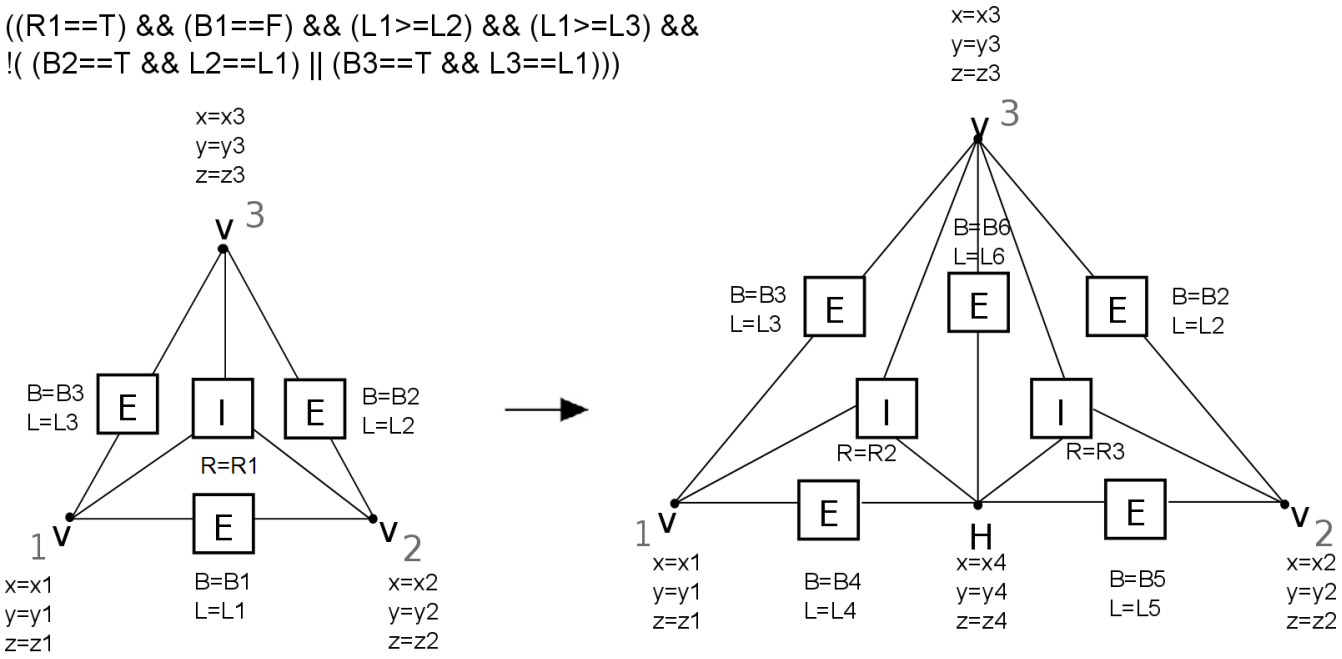


where:
L4=L1/2
B4=B1
L5=L1/2
B5=B1
x4=(x1+x2)/2
y4=(y1+y2)/2
z4=(z1+z2)/2
$L6=\sqrt{(x3-x4)^2+(y3-y4)^2+(z3-z4)^2}$
R2=F
R3=F
B6=F

2. Please implement unit test for graph transformation described in 1, concerning rule (P1).

3. Please implement graph transformation (P2)

((R1==T) && (B1==F) && (L1>=L2) && (L1>=L3) &&
!((B2==T && L2==L1) || (B3==T && L3==L1)))

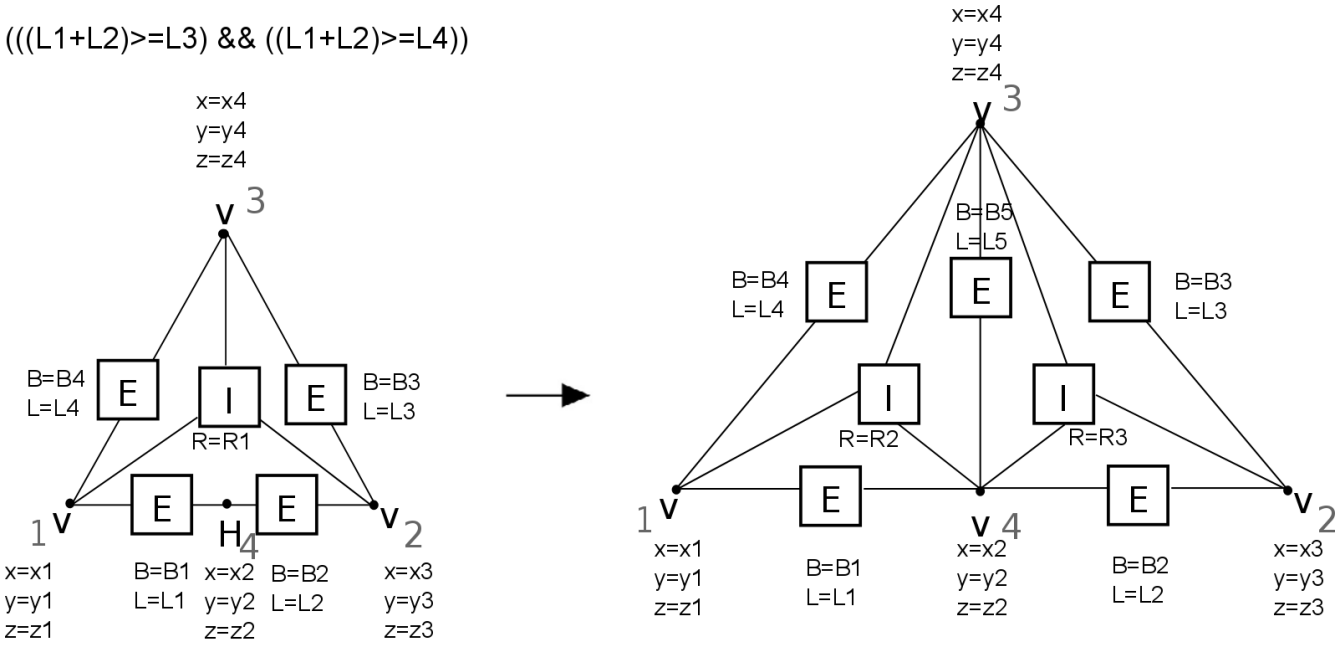


where:
L4=L1/2
B4=B1
L5=L1/2
B5=B1
x4=(x1+x2)/2
y4=(y1+y2)/2
z4=(z1+z2)/2
$L6=sqrt((x3-x4)^2+(y3-y4)^2+(z3-z4)^2))$
B6=F
R2=F
R3=F

4. Please implement unit test for graph transformation described in 3, concerning rule (P2).

5. Please implement graph transformation (P3)

(((L1+L2)>=L3) && ((L1+L2)>=L4))

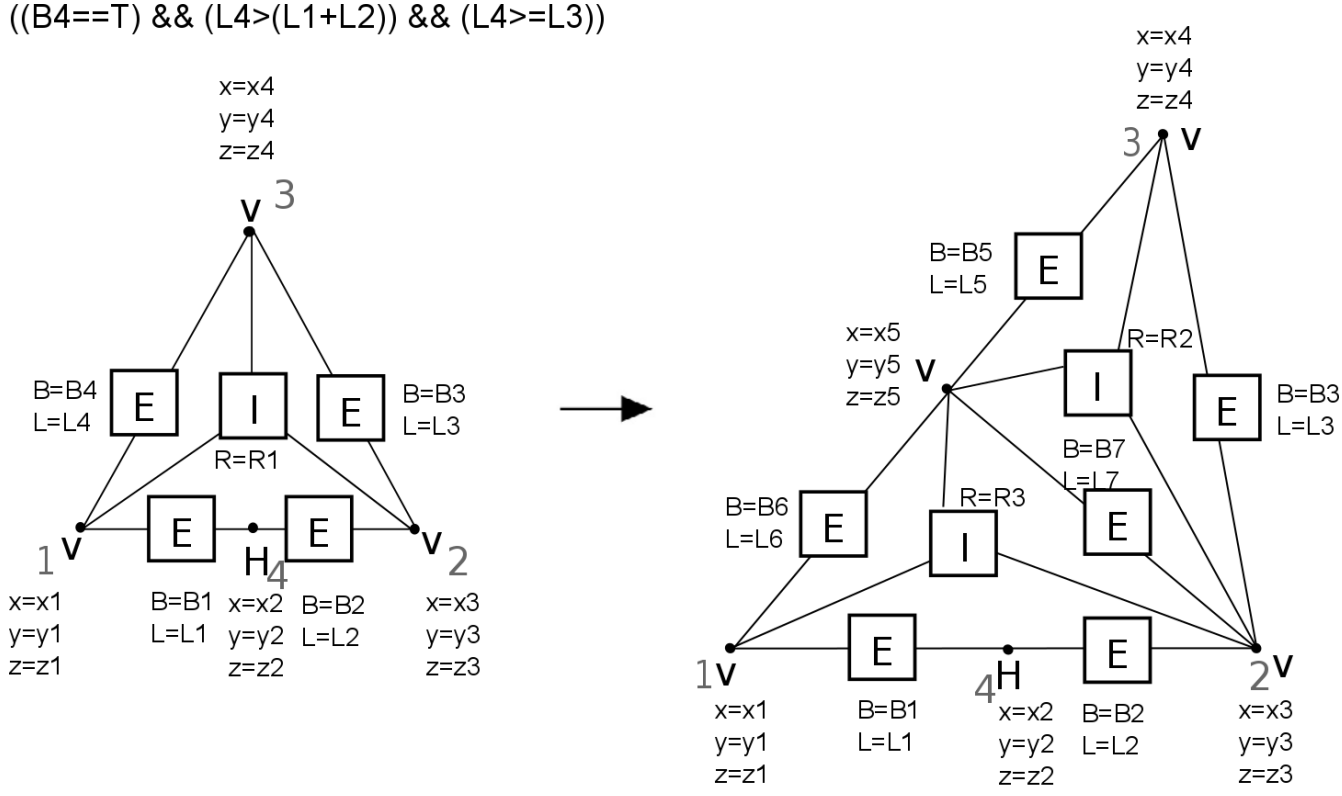


where:
R2=F
R3=F
L6=sqrt((x2-x4)^2+(y2-y4)^2+(z2-z4)^2))
B6=F

6. Please implement unit test for graph transformation described in 5, concerning rule (P3).

7. Please implement graph transformation (P4)

((B4==T) && (L4>(L1+L2)) && (L4>=L3))

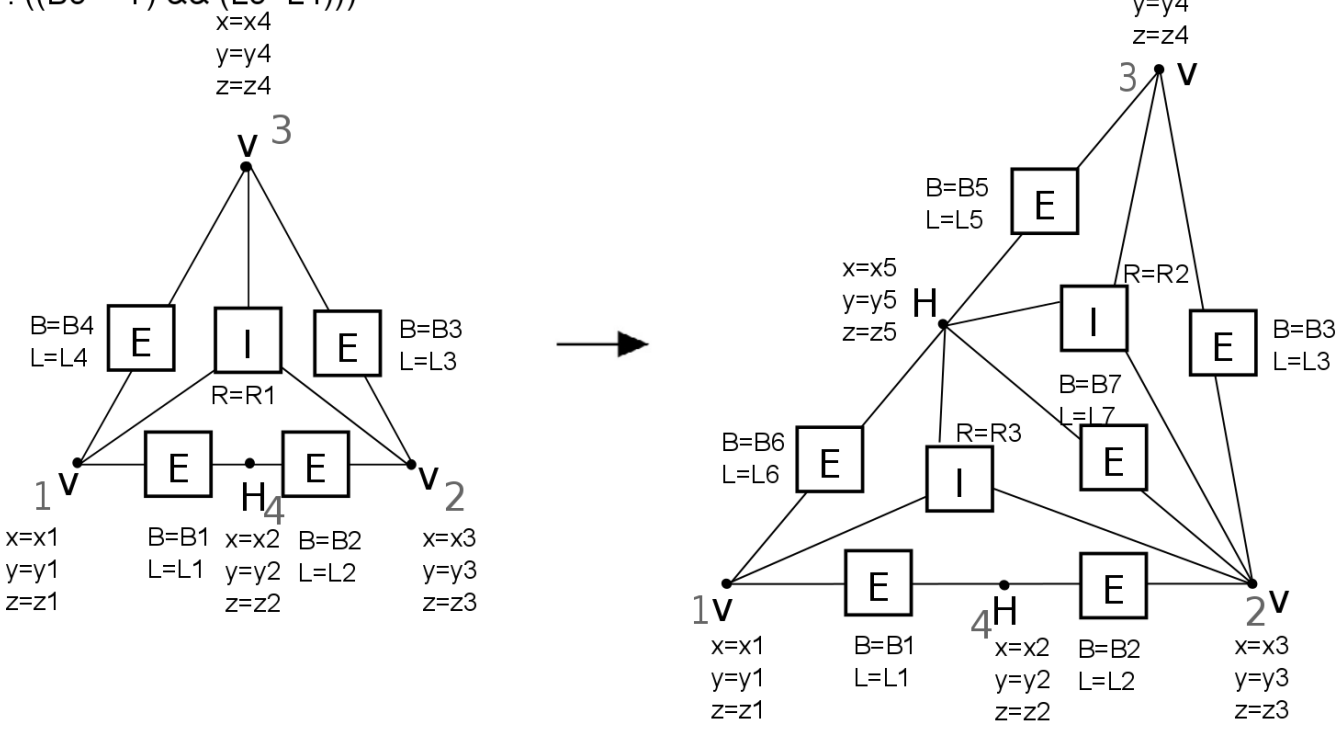


where:
L5=L4/2; B5=B4
L6=L4/2; B6=B4
x5=(x1+x4)/2
y5=(y1+y4)/2
z5=(z1+z4)/2
$L7=sqrt((x3-x5)^2+(y3-y5)^2+(z3-z5)^2))$
B7=F
R2=F
R3=F

8. Please implement unit test for graph transformation described in 7, concerning rule (P4).

9. Please implement graph transformation (P5)

((B4==F) && (L4>(L1+L2)) && (L4>=L3) &&
! ((B3==T) && (L3=L4)))

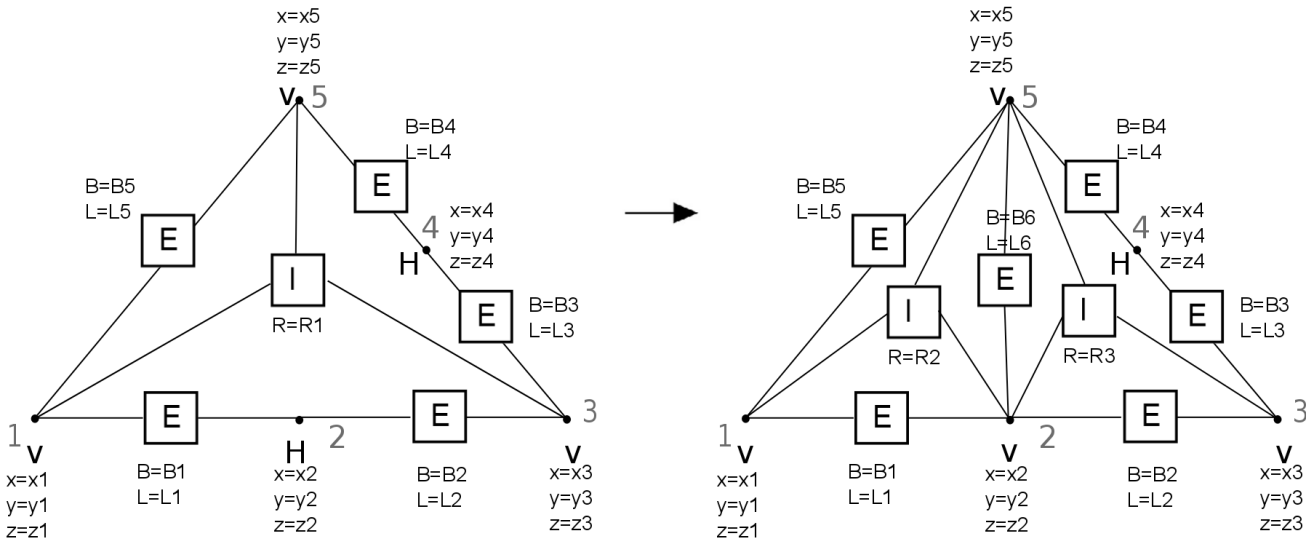


where:
L5=L4/2; B5=B4
L6=L4/2; B6=B4
x5=(x1+x4)/2
y5=(y1+y4)/2
z5=(z1+z4)/2
$L7=sqrt((x3-x5)^2+(y3-y5)^2+(z3-z5)^2))$
B7=F
R2=F
R3=F

10. Please implement unit test for graph transformation described in 9, concerning rule (P5).

11. Please implement graph transformation (P6)

((L1+L2)>=(L3+L4)) && ((L1+L2)>=L5))

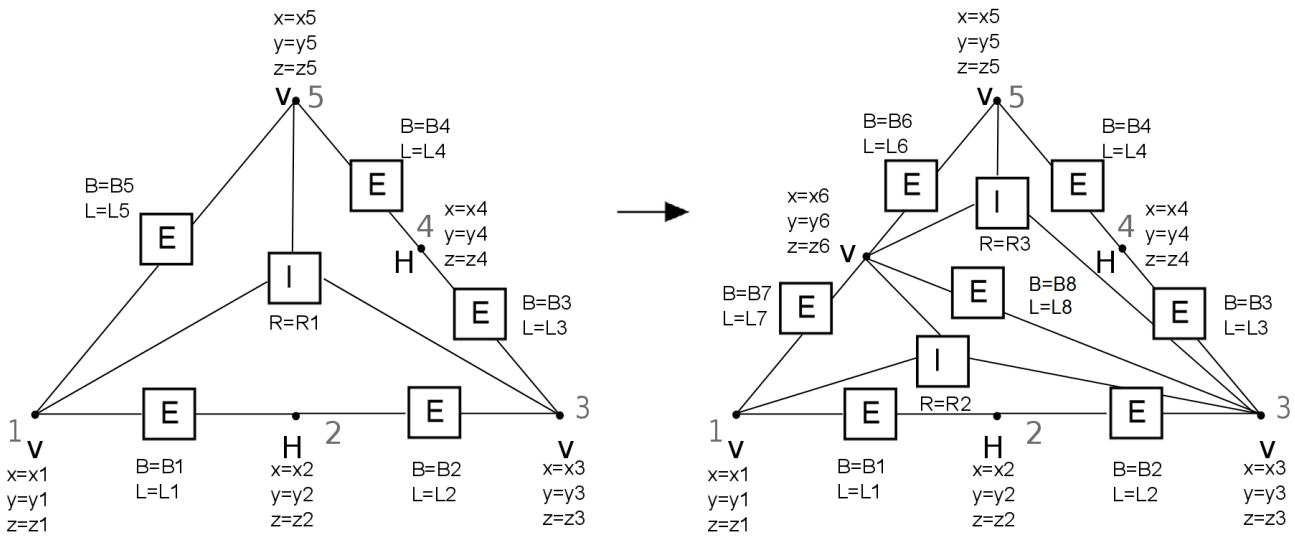


where:
R2=F
R3=F
L6=sqrt((x2-x5)^2+(y2-y5)^2+(z2-z5)^2))
B6=F

12. Please implement unit test for graph transformation described in 11, concerning rule (P6).

13. Please implement graph transformation (P7)

((B5==T) && (L5>(L1+L2)) && (L5>(L3+L4)))

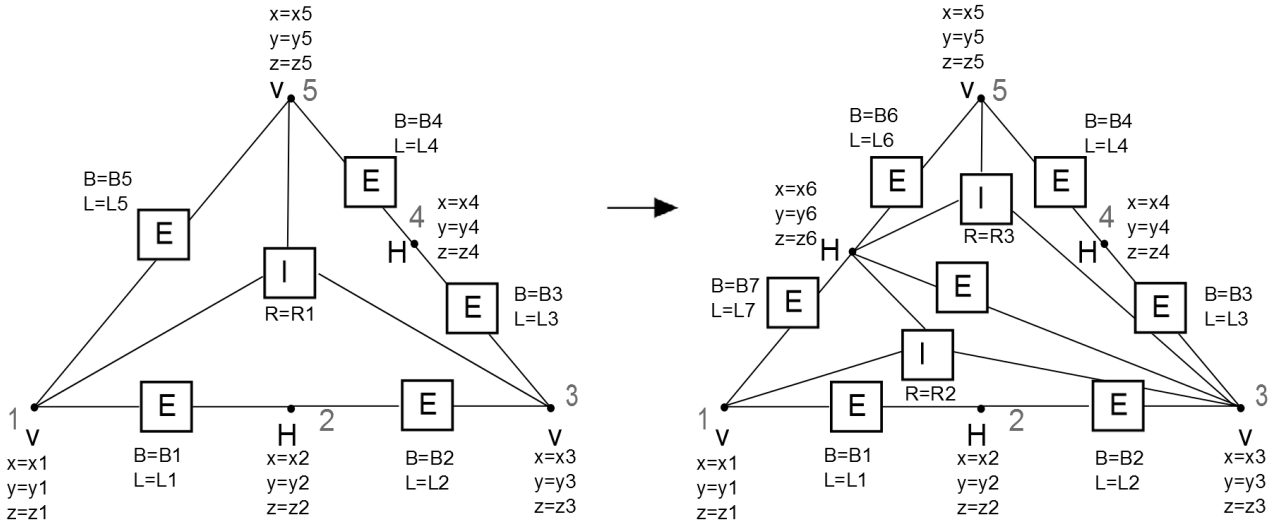


where:
L6=L5/2; B6=B5
L7=L5/2; B7=B5
x6=(x1+x5)/2
y6=(y1+y5)/2
z6=(z1+z5)/2
$L8=\sqrt{(x3-x6)^2+(y3-y6)^2+(z3-z6)^2}$
B8=F
R2=F
R3=F

14. Please implement unit test for graph transformation described in 13, concerning rule (P7).

15. Please implement graph transformation (P8)

((B5==F) && (L5>(L1+L2)) && (L5>(L3+L4)))

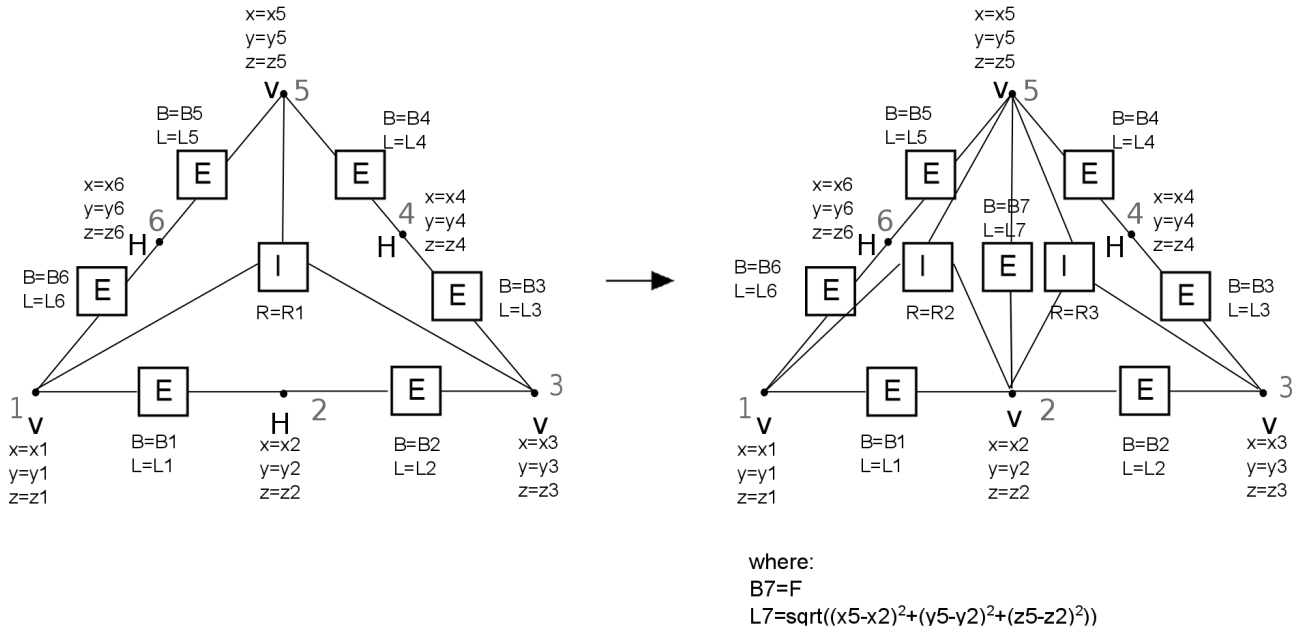


where:
L6=L5/2; B6=B5
L7=L5/2; B7=B5
x6=(x1+x5)/2
y6=(y1+y5)/2
z6=(z1+z5)/2
$L7=\sqrt{(x3-x6)^2+(y3-y6)^2+(z3-z6)^2}$
B8=F
R2=F
R3=F

16. Please implement unit test for graph transformation described in 15, concerning rule (P8).

17. Please implement graph transformation (P9)

(((L1+L2)>=(L3+L4)) && ((L1+L2)>=(L5+L6)))



18. Please implement unit test for graph transformation described in 17, concerning rule (P9).

19. Please download, compile and run
https://github.com/Podsiadlo/terrain

For example, for Europe

```
tergen -o out -Y -d Data/SRTM3 -x 100 -y 100 -z 100 -i 5 -j 5 -k 1 -N 68 -S 35 -E 50 -W -10
```

based on NASA database it generates files with terrain data.
-N -S range (-90,90) along the meridian, -W -E range (-180,180) along the equator
-i -j -k number of output files
-x, -y, -z number of terrain points, sorted from bottom to top layers, from W to E, from N to S, =0 if air, =1 if terrain
Data/SRTM3 NASA database directory (to be downloaded)
-o out name of the input file
Please generate data for Poland with 1 file and 3000 x 3000 x 3000 points (around 300 meters grid)

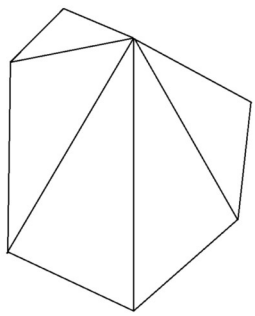
20. Procedure estimating the terrain approximation error for a given triangle
a) Find all terrain points located in the triangle area
b) Compute terrain points resulting from triangular approximation
c) Compute sum of squares of differences between real terrain points and triangle approximation points
d) Compute sum of squares of real terrain points located in the triangle area
e) Divide c) by d)

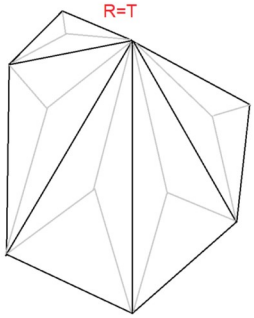
21. Algorithm marking triangles for refinement
a) Start from a mesh constructed from two triangles forming a rectangle, span over the terrain
b) Loop over triangles T
b.1) Estimate approximation error over T
b.2) If error is larger than given epsilon, mark triangle for refinement

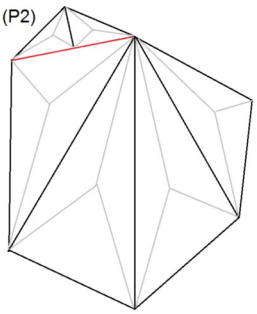
22. Please implement reasonable integration test for algorithm from point 21

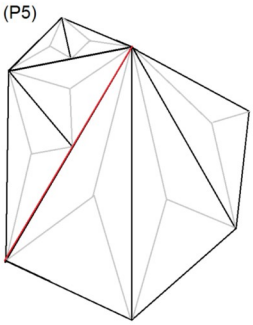
23. Algorithm for mesh adaptation
a) Execute all possible graph transformations P1
b) Execute all possible graph transformations P2
c) Execute all possible graph transformations P3
d) Execute all possible graph transformations P4
e) Execute all possible graph transformations P5
f) Execute all possible graph transformations P6
g) Execute all possible graph transformations P7
h) Execute all possible graph transformations P8
i) Execute all possible graph transformations P9
j) If no graph transformations P3,P4,P5,P6,P7,P8,P9 were executed in points c)-i) then STOP
k) Go to c)

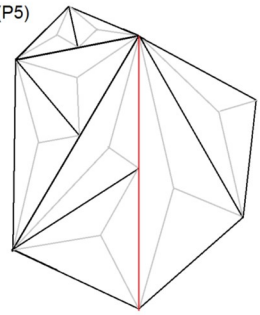
24. Please implement integration test for algorithm from point 23, using the sketch below

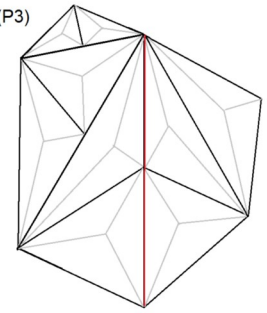


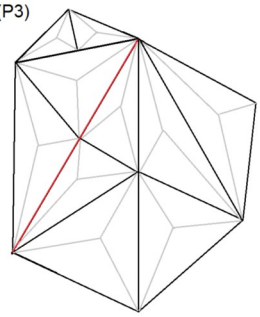

R=T

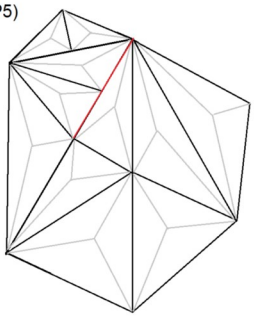

(P2)

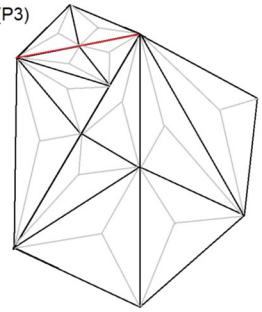

(P5)

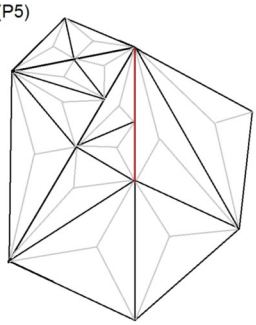

(P5)

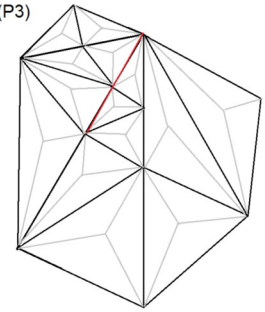

(P3)

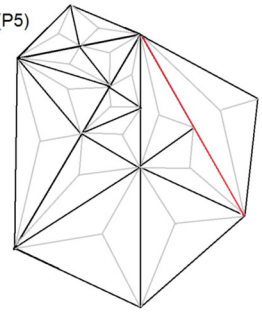

(P3)

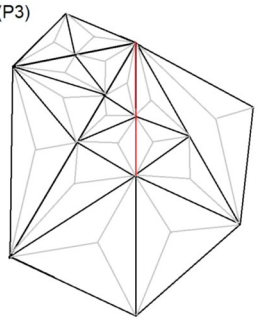

(P5)

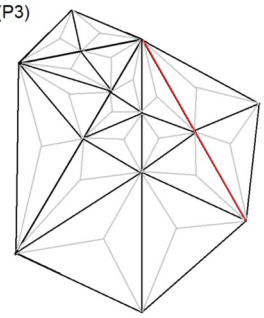

(P3)

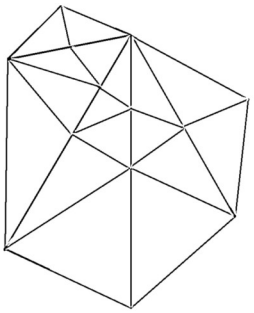

(P5)


(P3)


(P5)


(P3)


(P3)



25. Proszę stworzyć następującą siatkę reprezentowaną przez naszą strukturę grafu
Proszę uruchomić algorytm z punktu 23 na tej siatce, I zwizualizować zmiany (adaptacje siatki)