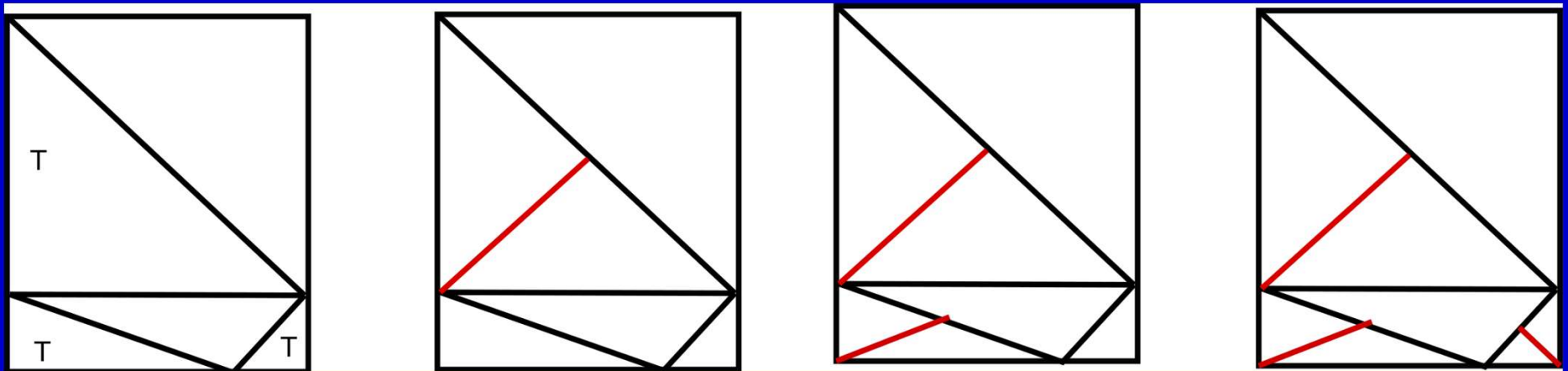


PROBLEM 1:

W podanej gramatyce nie rozróżnialiśmy hanging nodes i zwykłych wierzchołków.

Możemy wygenerować graf odpowiadający siatce, gdzie na jednej krawędzi są dwa hanging nodes

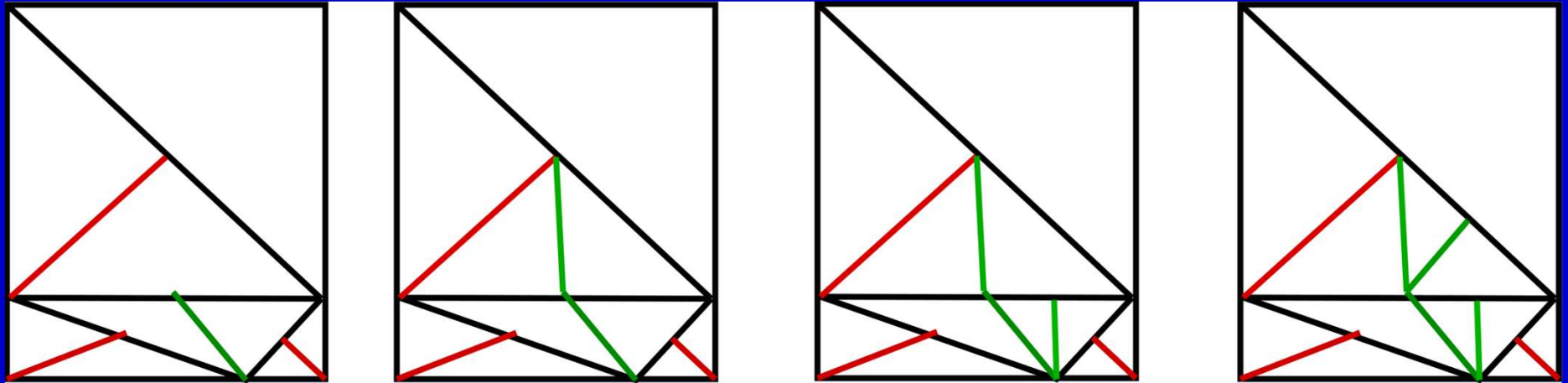
Łamanie elementów zaznaczonych do złamania



PROBLEM 1:

W podanej gramatyce nie rozróżnialiśmy hanging nodes i zwykłych wierzchołków.

Naprawianie siatki – łamanie elementów gdzie są hanging nodes



Problem – nie sprawdzamy w lewej stronie produkcji, czy wierzchołek jest hanging node, jeśli jest, to w niektórych przypadkach nie powinniśmy łamać

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA- podejście pierwsze

PROBLEM 1:

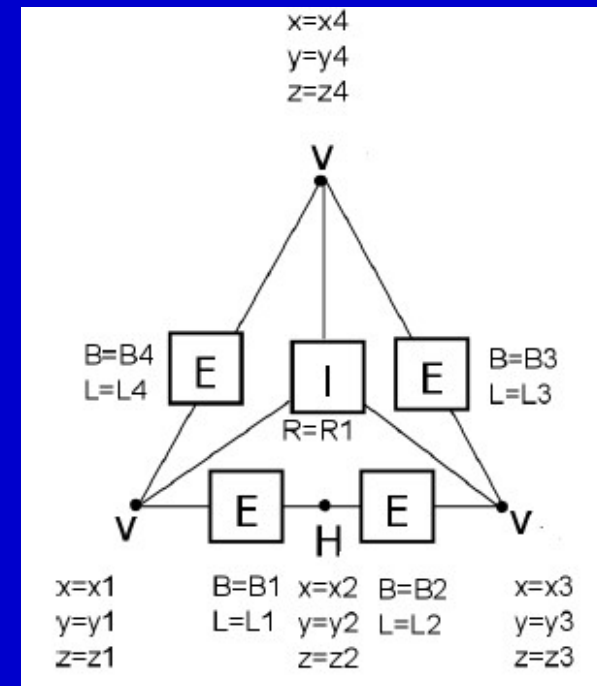
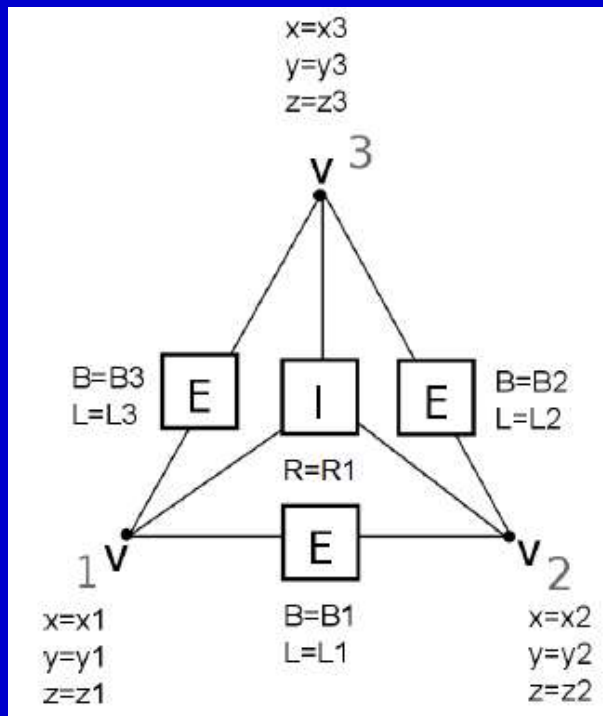
W podanej gramatyce nie rozróżnialiśmy hanging nodes i zwykłych wierzchołków.

Rozwiązanie:

- rozróżnić wierzchołki i hanging nodes,

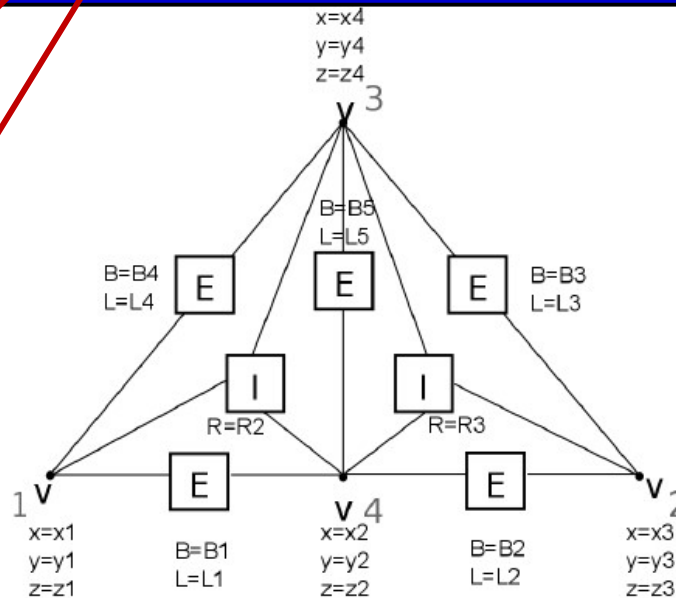
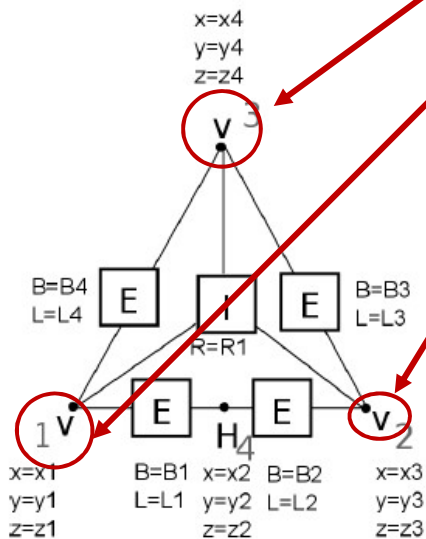
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA- podejście pierwsze

- **etykieta v** – wierzchołek grafu reprezentujący wierzchołek elementu siatki
- **etykieta H** – wierzchołek grafu reprezentujący hanging node



Produkcje naprawiające siatkę: zawsze po lewej stronie produkcji 3 wierzchołki z etykietą v (odpowiadają wierzchołkom trójkąta)

$((L1+L2) \geq L3) \ \&\& \ ((L1+L2) \geq L4)$



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

Naprawianie siatki

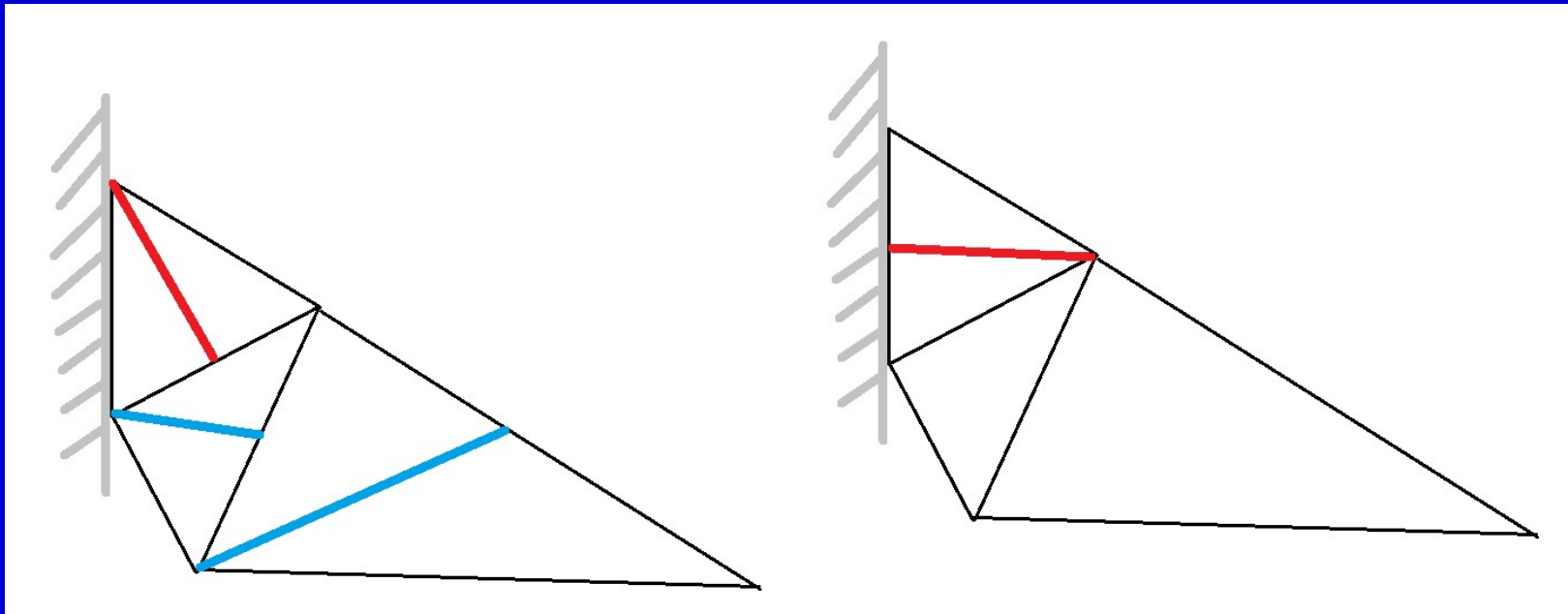
- element ma już hanging node
- hanging node jest na najdłuższej krawędzi (lub inne krawędzie są tak samo długie)

Wtedy łamiemy do najdłuższej krawędzi – mamy 2 nowe trójkąty, hanging node zamienia się w zwykły wierzchołek

Problem 2:

Nie rozróżnialiśmy krawędzi na brzegu i krawędzi współdzielonej.

Trójkąt równoboczny leżący przy brzegu musi być złamany do brzegu żeby uniknąć niepotrzebnej propagacji adaptacji



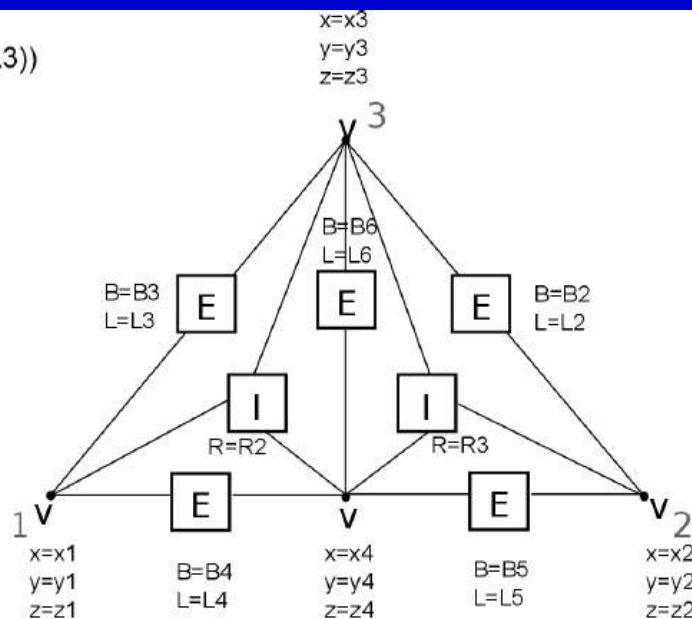
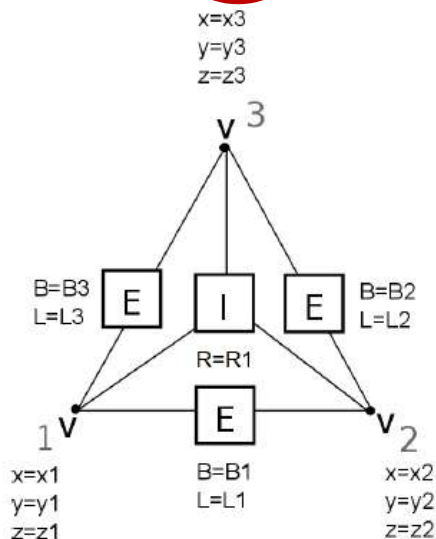
PROBLEM 2 - Trójkąt równoboczny leżący przy brzegu musi być złamany do brzegu żeby uniknąć niepotrzebnej propagacji adaptacji

Rozwiązanie:

- rozróżnić krawędzie współdzielone i krawędzie na brzegu,
(krawędzie z etykietą **E** mają atrybut **B**; **B=TRUE** dla brzegu, **B=FALSE** wpp)
- dodać produkcję łamiącą krawędź na brzegu
- wykonywać najpierw produkcję łamiącą na brzegu
- potem łamiącą krawędź współdzieloną

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P1

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



where:

$L4=L1/2$
 $B4=B1$
 $L5=L1/2$
 $B5=B1$

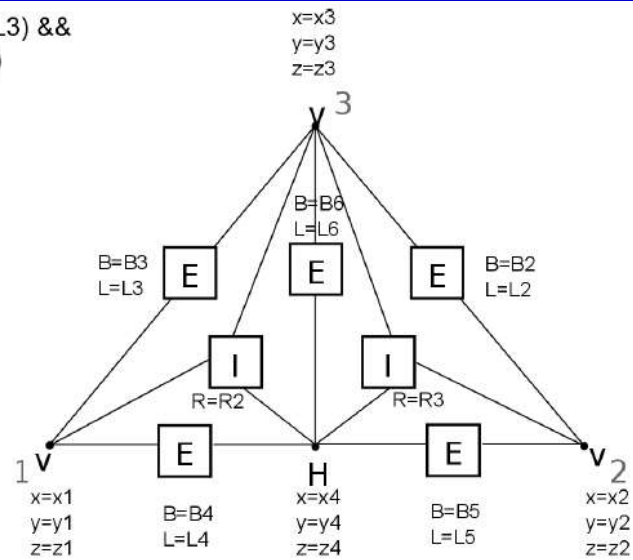
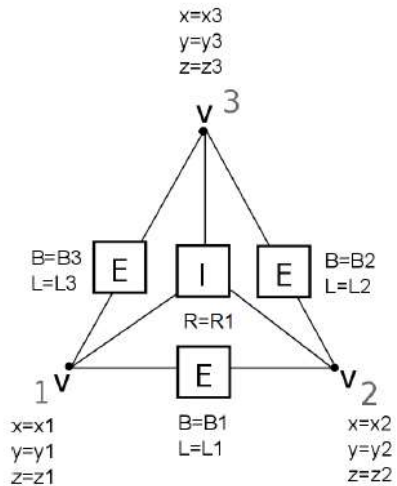
$x4=(x1+x2)/2$
 $y4=(y1+y2)/2$
 $z4=(z1+z2)/2$
 $L6=\text{sqrt}((x3-x4)^2+(y3-y4)^2+(z3-z4)^2)$
 $R2=F$
 $R3=F$
 $B6=F$

Łamanie elementu:
 -zaznaczonego do złamania,
 -gdy żadna krawędź nie jest złamana oraz
-najdłuższa krawędź jest brzegiem

łamiemy do najdłuższej krawędzi –brzegu i mamy 2 nowe trójkąty,
 na krawędzi pojawia się wierzchołek

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P2

$((R1==T) \&\& (B1==F) \&\& (L1>=L2) \&\& (L1>=L3) \&\& !((B2==T \&\& L2==L1) \parallel (B3==T \&\& L3==L1)))$



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

Łamanie elementu:

- Zaznaczonego do złamania,
- gdy żadna krawędź nie jest złamana oraz
- najdłuższa krawędź nie jest brzegiem**

łamiemy do
 najdłuższej krawędzi
 i mamy 2 nowe
 trójkąty,
 na krawędzi pojawia
 się hanging node

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA

DIAGRAM STERUJĄCY

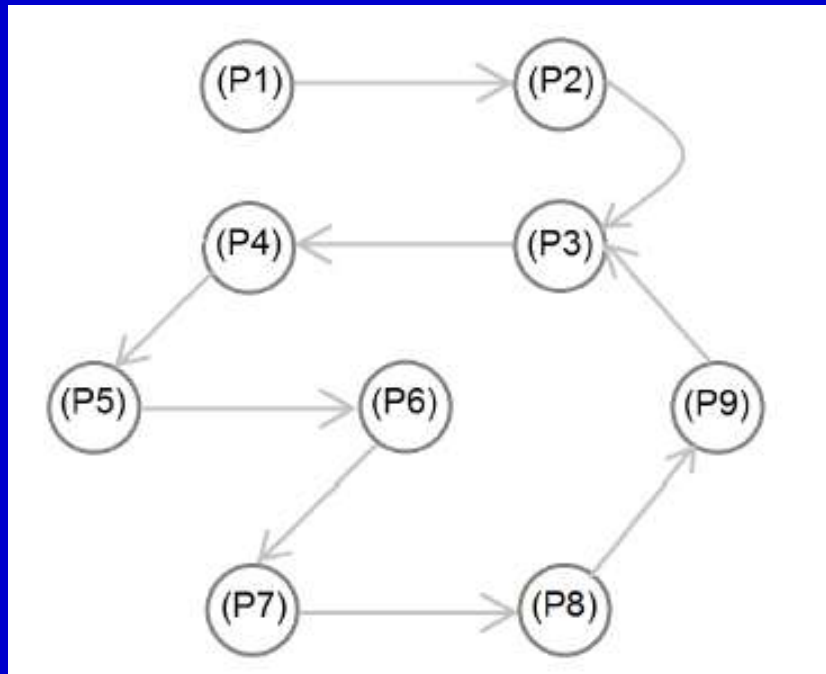


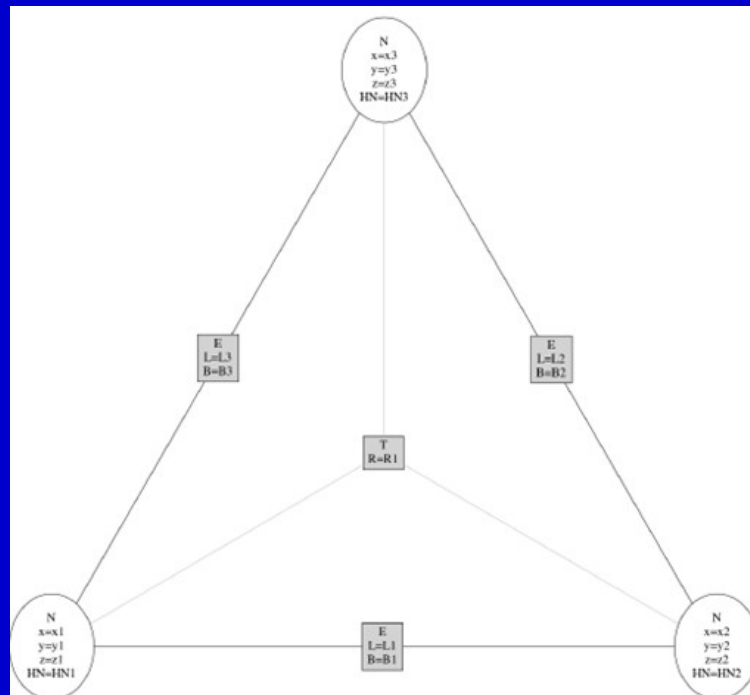
Diagram sterujący wymusza łamanie do brzegu, tzn. jeśli mamy trójkąt równoboczny, i mamy krawędź na brzegu to ją złamiemy (stosując produkcję P1) a nie złamiemy krawędzi współdzielonej, gdzie powstaje hanging node (produkcja P2)

- W PAMIĘCI GLOBALNEJ MAMY LICZNIK,
- DLA KAŻDEJ PRODUKCJI DODAJĄCEJ NOWY HANGING NODE ZWIĘKSZAMY WARTOŚĆ LICZNIKA,
- DLA KAŻDEJ PRODUKCJI USUWAJĄCEJ HANGING NODE ZMNIJSZAMY WARTOŚĆ LICZNIKA,
- KOŃCZYMY, GDY LICZNIK=0

NOWE PODEJŚCIE
GRAMATYKA GRAFOWA
DLA ALGORYTMU RIVARA

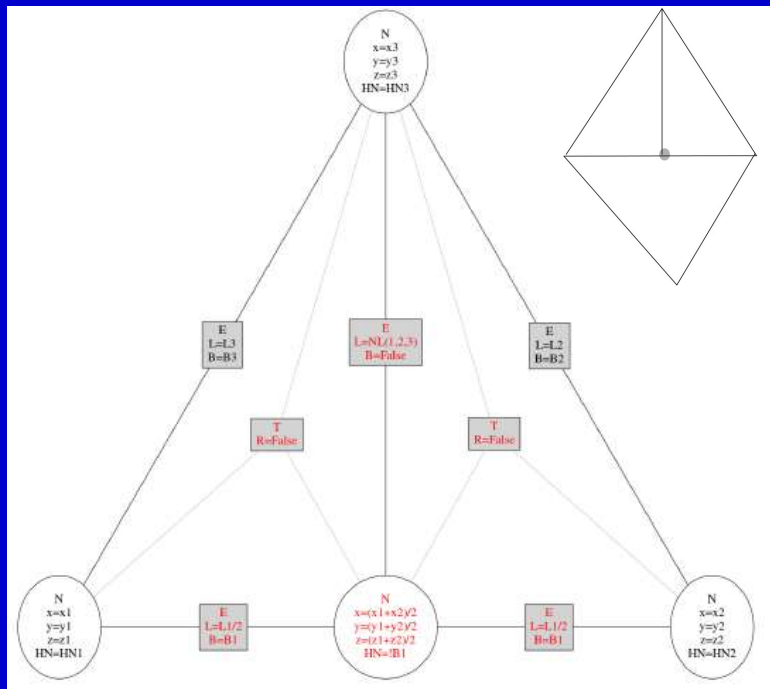
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA

- **etykieta N** – wierzchołek grafu reprezentujący wierzchołek elementu siatki (obojętnie, czy hanging node, czy nie)
- **etykieta T** – hiperkrawędź grafu reprezentująca wnętrze elementu trójkątnego
- **etykieta E** – hiperkrawędź grafu reprezentująca krawędź elementu trójkątnego (obojętnie, czy brzeg, czy nie)

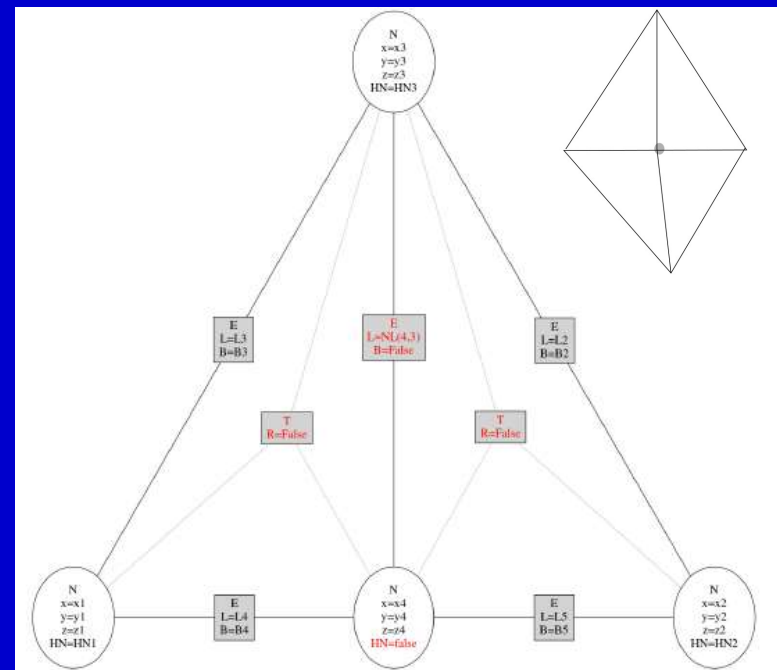


GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA

- **Atrybuty x,y,z** – dla wierzchołków grafu reprezentujących wierzchołki elementów siatki (współrzędne wierzchołków)
- **Atrybut HN** – dla wierzchołków grafu reprezentujących wierzchołki siatki ($HN=TRUE$ dla hanging node, $FALSE$ wpp)



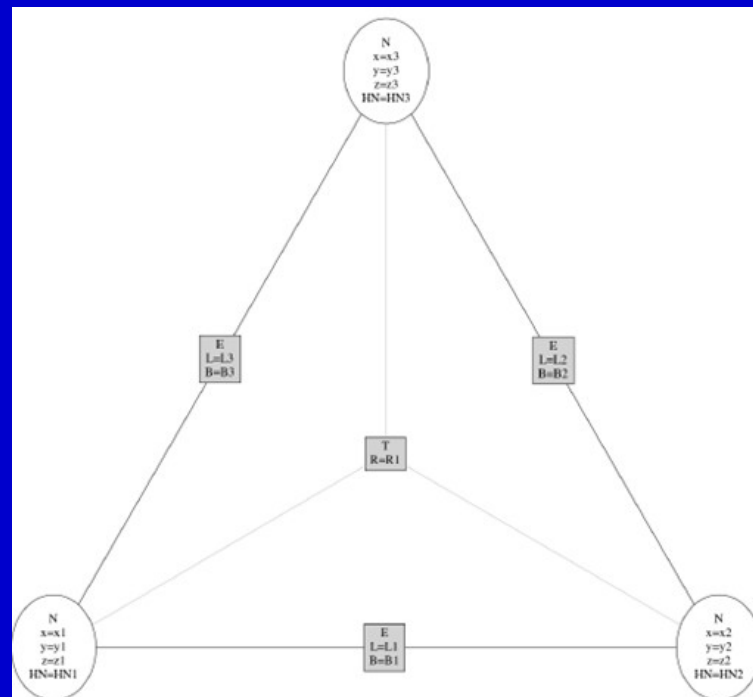
HN=TRUE



HN=FALSE

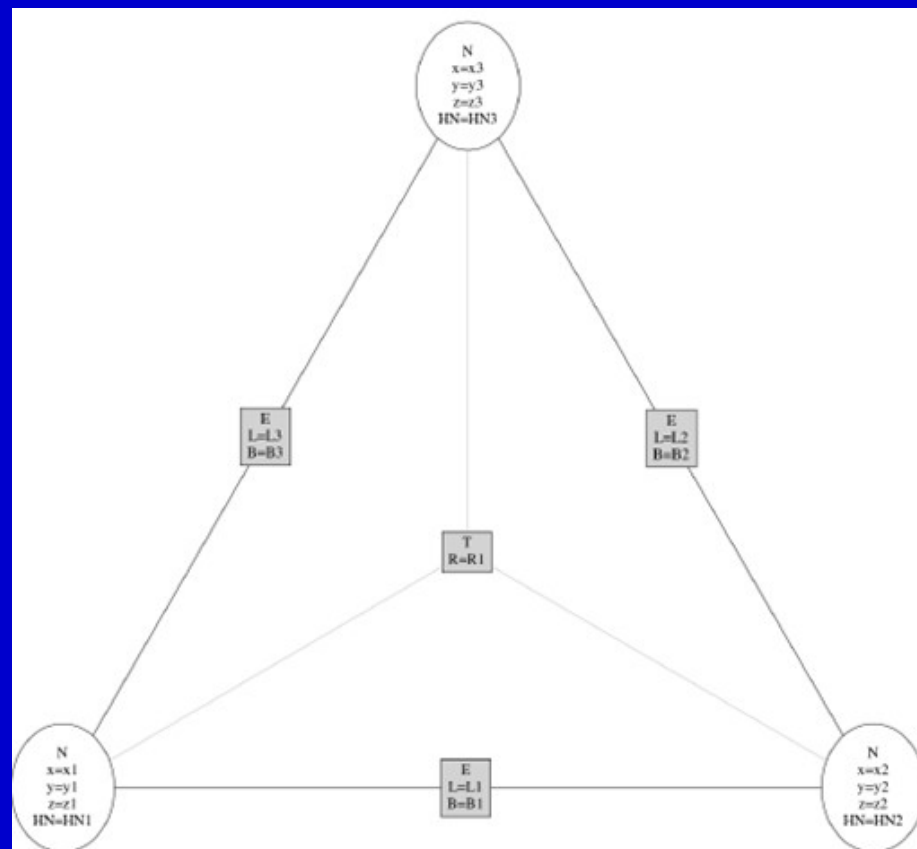
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA

- **Atrybut B** – dla hiperkrawędzi grafu reprezentujących krawędzie elementów siatki (B=TRUE, gdy krawędź jest na brzegu, FALSE wpp)
- **Atrybut L** – dla hiperkrawędzi grafu reprezentujących krawędzie elementów siatki (oznacza długość krawędzi)

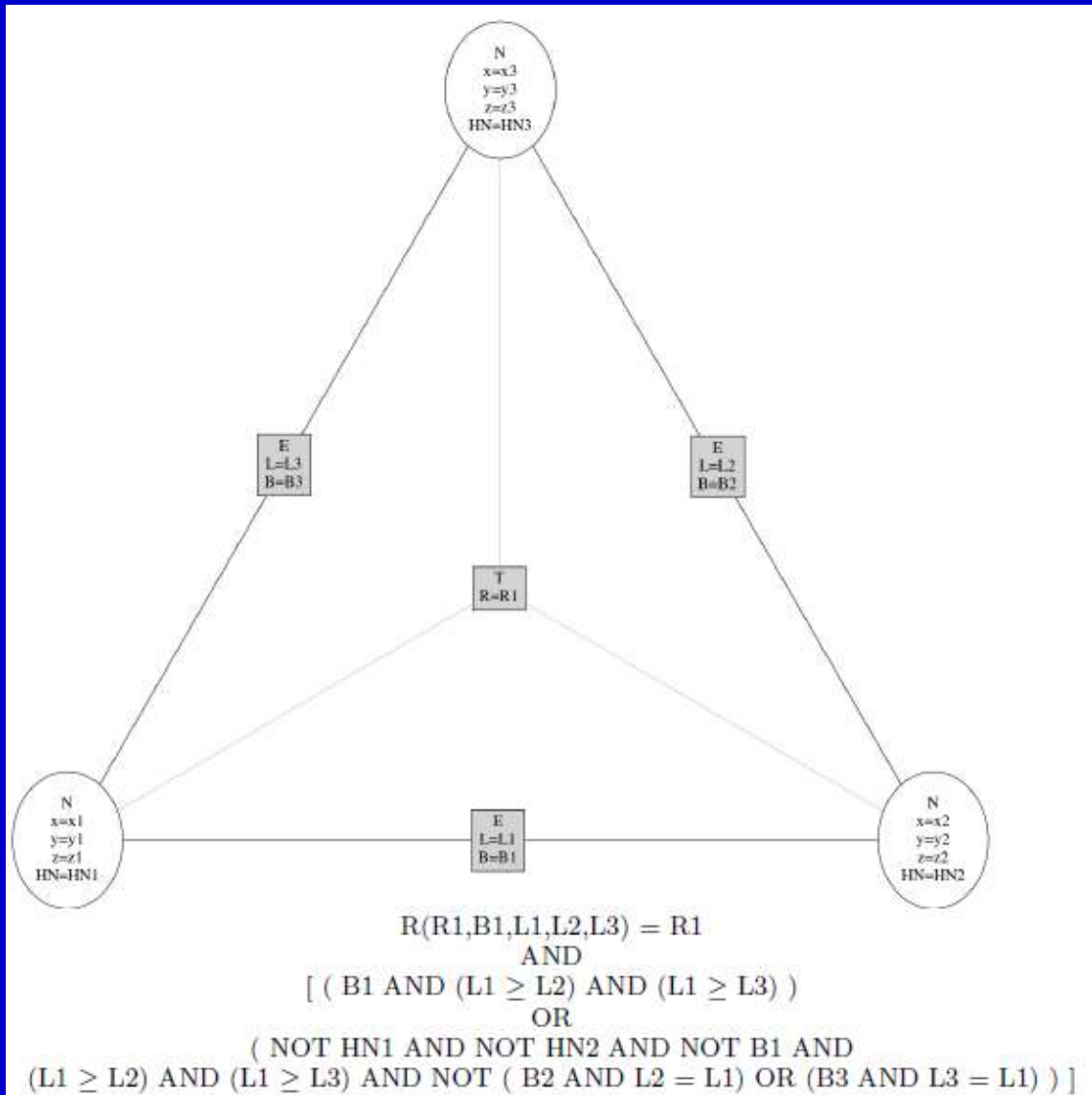


GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA

- **Atrybut R** – dla hiperkrawędzi grafu reprezentujących wnętrza elementów siatki
(R=TRUE, gdy element ma zostać złamany, FALSE wpp)



GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P1 (L)



Łamanie elementu:

-Zaznaczonego do złamania,
 -gdy żadna krawędź nie jest
 złamana oraz

-najdłuższa krawędź jest
 brzegiem

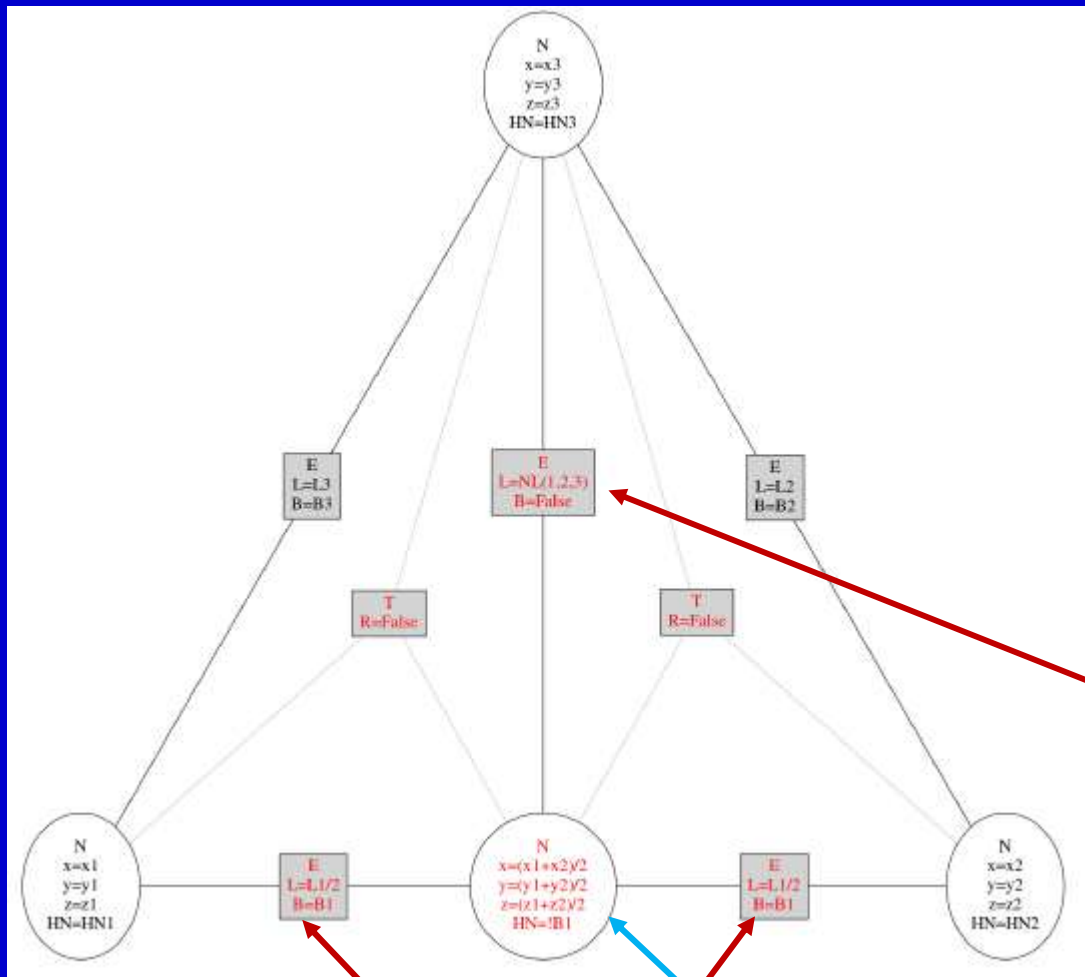
lub

-najdłuższa krawędź nie jest
 brzegiem i

wierzchołki należące do
 najdłuższej krawędzi nie są
 hanging nodes

I żadna z innych krawędzi nie
 jest tej samej długości i nie
 jest jednocześnie brzegiem

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P1 (R)



Łamanie elementu
przeznaczonego do złamania:

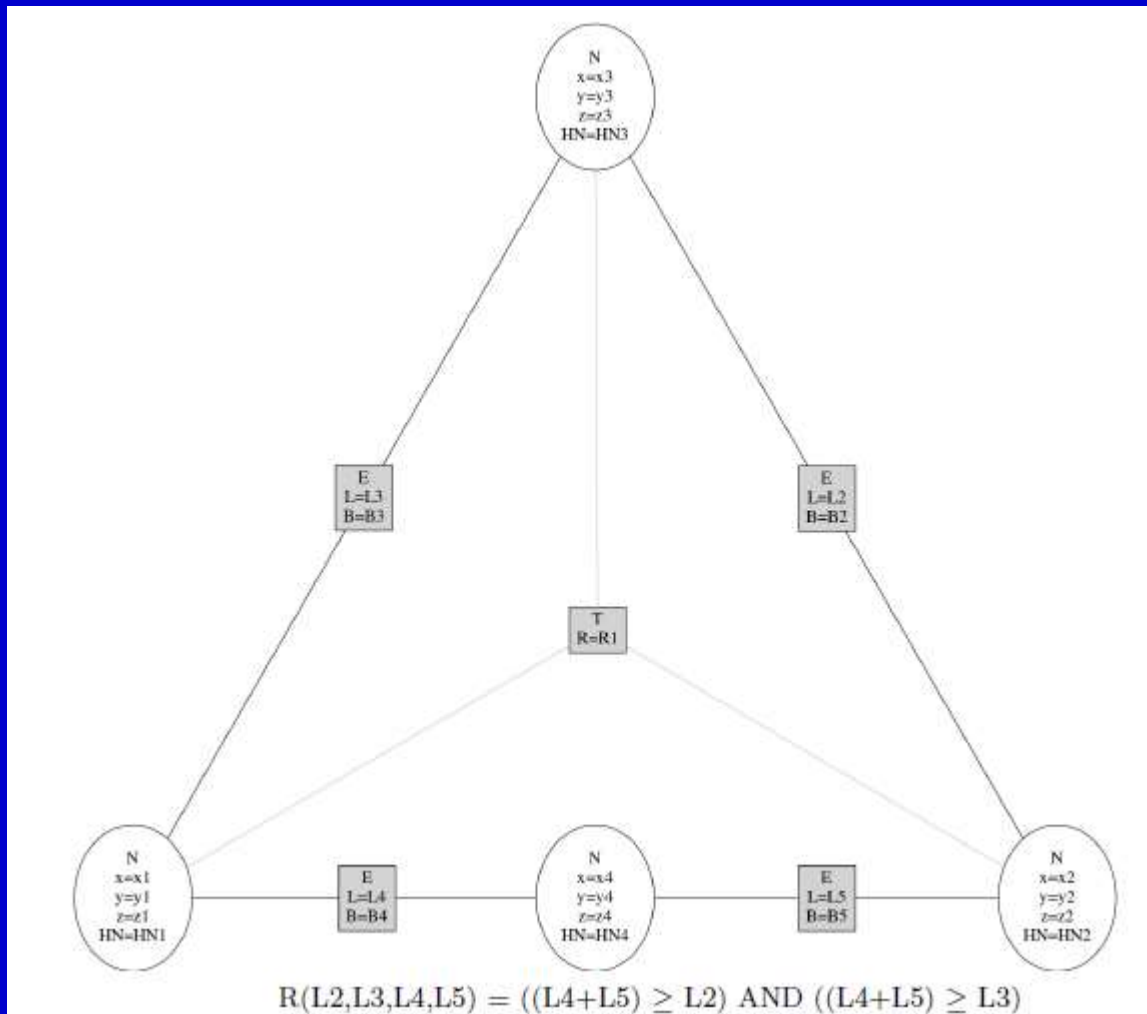
łamiemy do najdłuższej
krawędzi – mamy 2 nowe
trójkąty, na krawędzi pojawia się
hanging node

Nowopowstałe
hiperkrawędzie z etykietą
E „w środku” trójkąta
zawsze mają **B=false**

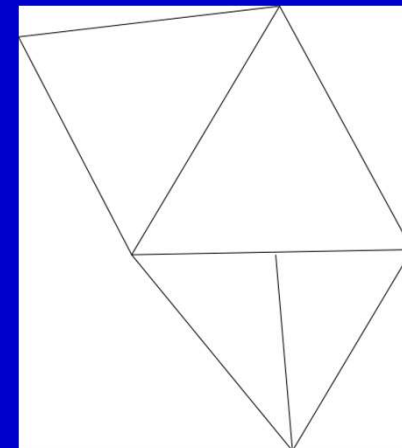
B=B1 – dziedziczymy po
krawędzi, którą podzieliliśmy

HN=!B1

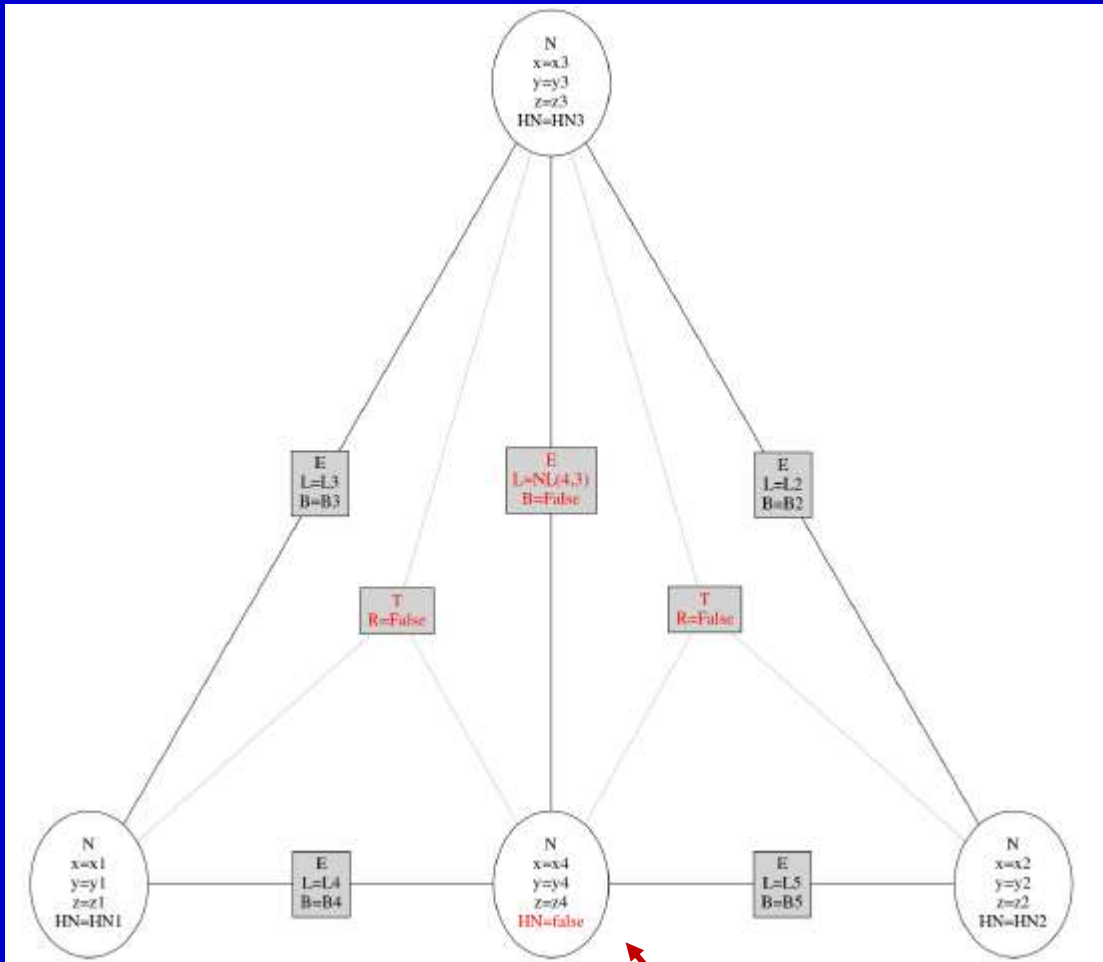
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P2 (L)



„naprawianie siatki”
 -element ma już hanging node
 -hanging node jest na
 najdłuższej krawędzi
 (lub inne krawędzie są tak samo
 długie)



GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P2 (R)

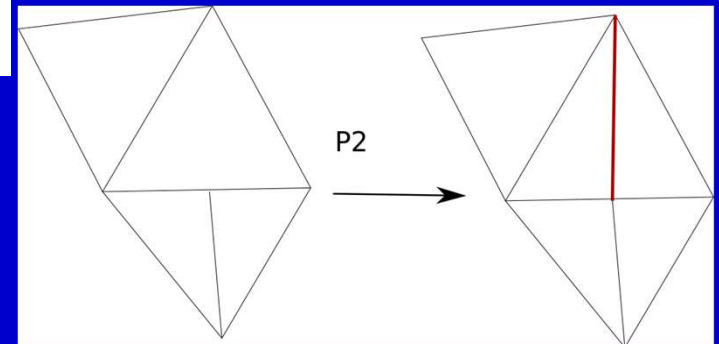


HN=FALSE

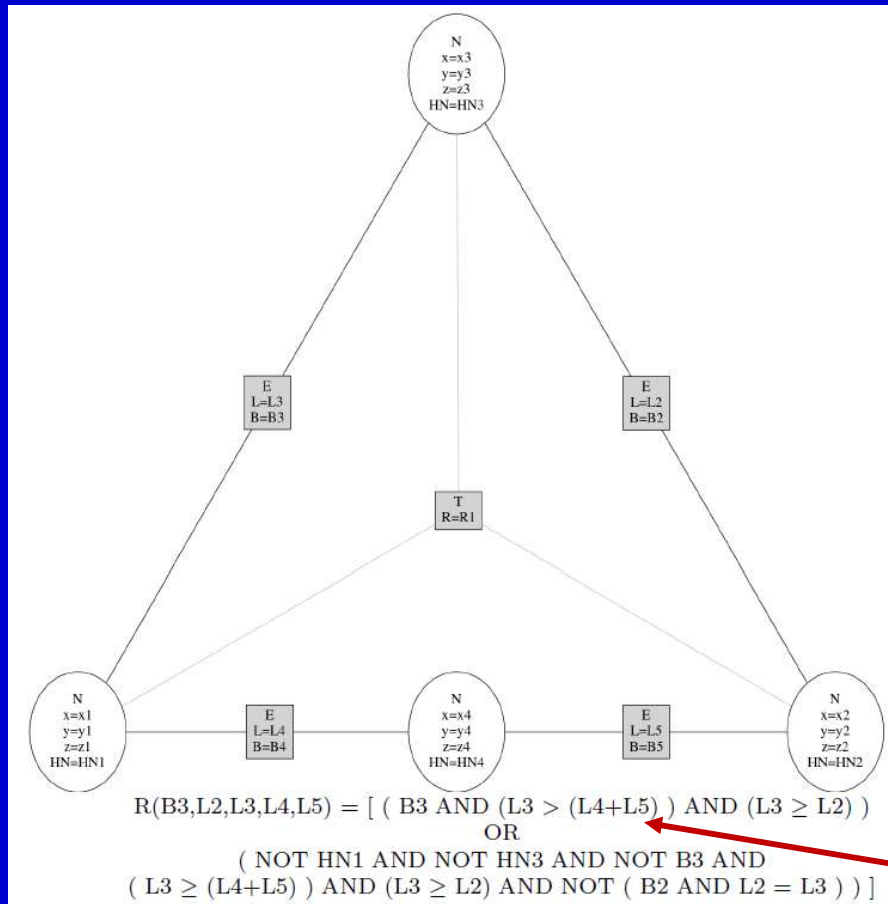
„naprawianie siatki”

-element ma już hanging node
-hanging node jest na
najdłuższej krawędzi
(lub inne krawędzie są tak samo
długie)

Wtedy łamiemy do najdłuższej
krawędzi – mamy 2 nowe
trójkąty, hanging node zmienia
się w zwykły wierzchołek



GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P3 (L)



„naprawianie siatki”

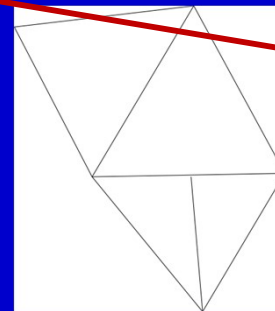
-element ma już hanging node
 -hanging node NIE jest na najdłuższej krawędzi

ORAZ

-najdłuższa krawędź jest brzegiem LUB

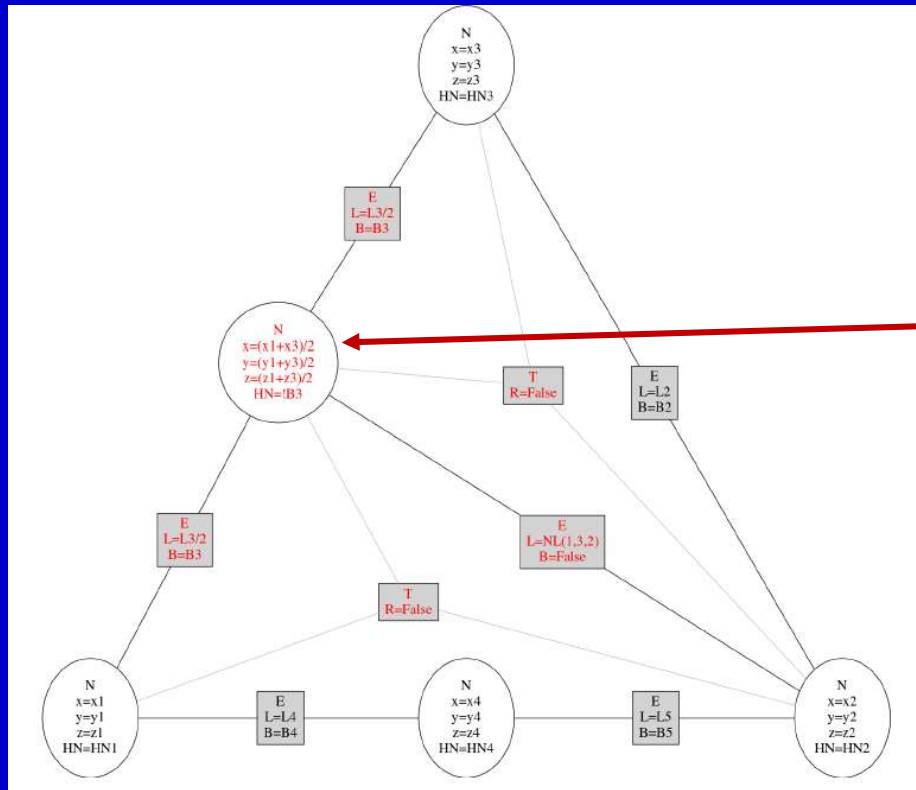
(najdłuższa krawędź nie jest brzegiem i należące do niej wierzchołki nie są hanging nodes i druga nie złamana krawędź nie jest tej samej długości będąc brzegiem)

Uwaga: jeśli mamy 2 krawędzie tej samej długości i jedna jest na brzegu, to łamiemy tą na brzegu!
 (żeby nie trzeba był propagować złamania)



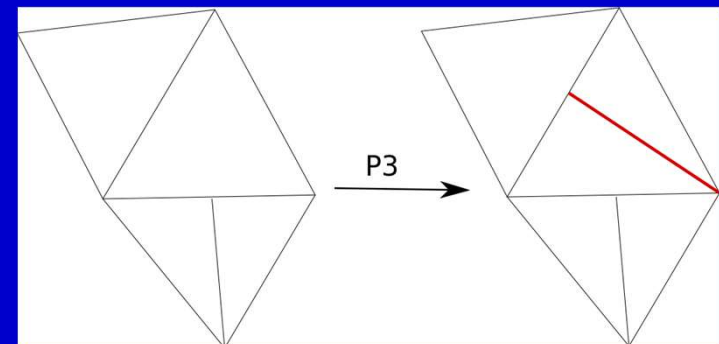
Najpierw sprawdzamy predykat dla krawędzi brzegowej

GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P3 (R)

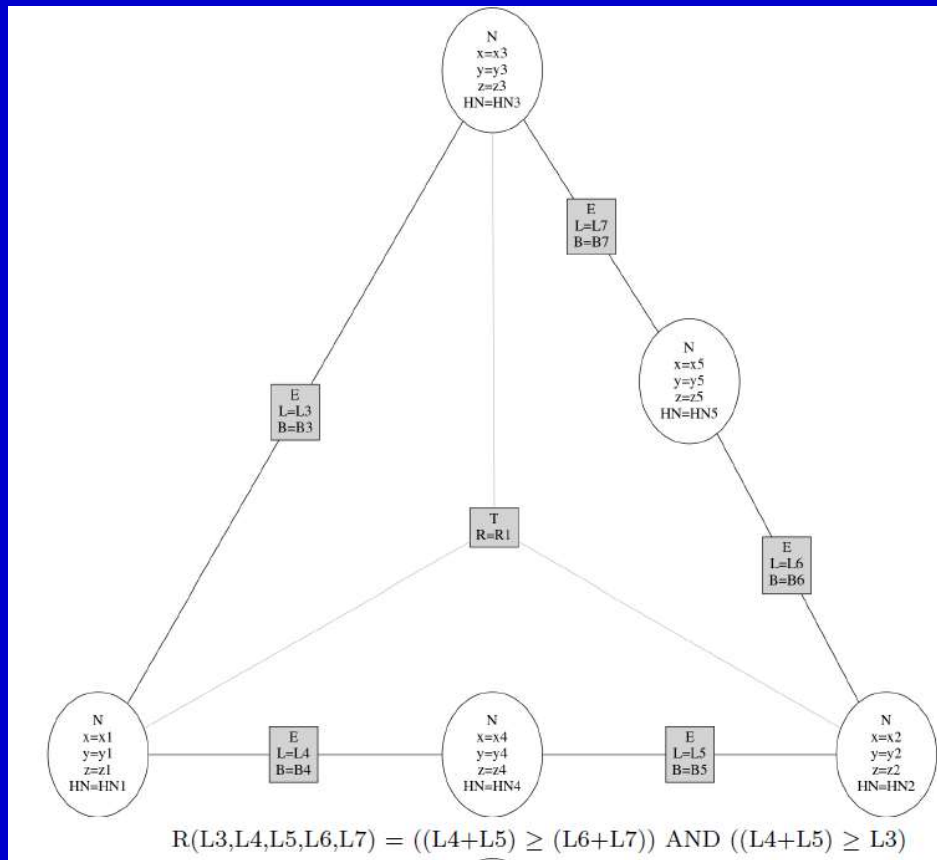


Wtedy łamiemy element i dodajemy nowy hanging node

HN=IB3

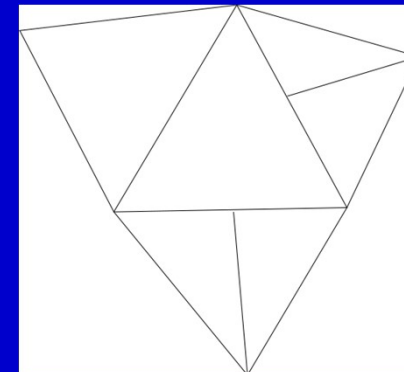


GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P4 (L)

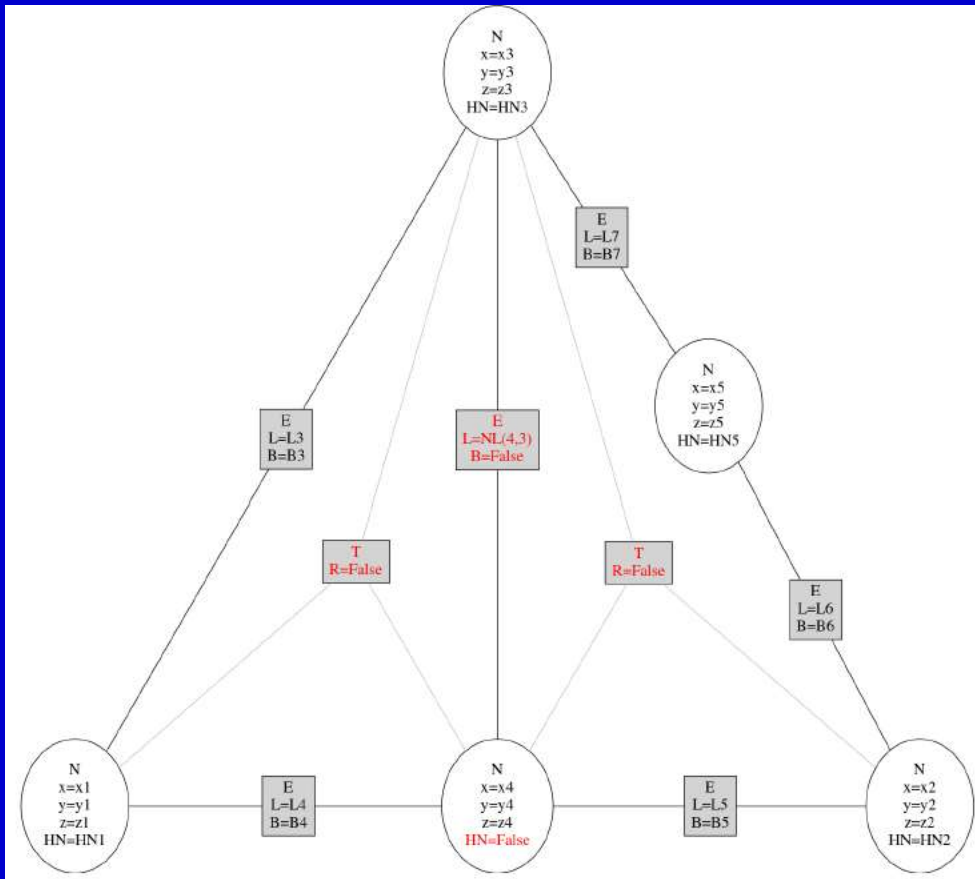


„naprawianie siatki”

- element ma już 2 hanging nodes
- hanging node jest na najdłuższej krawędzi



GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P4 (R)

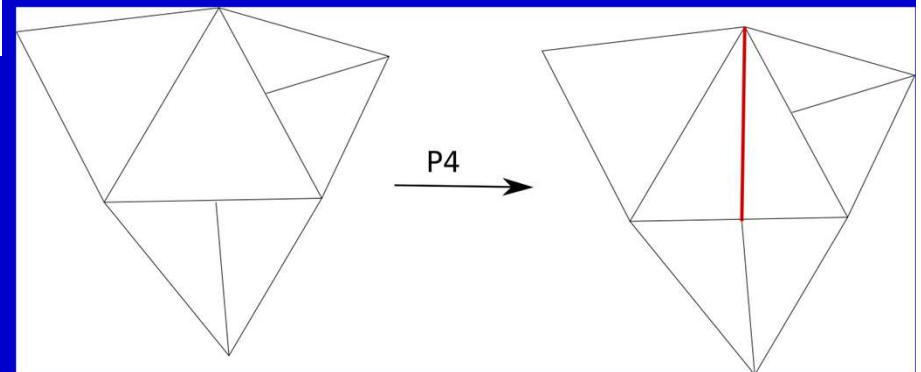


HN=FALSE

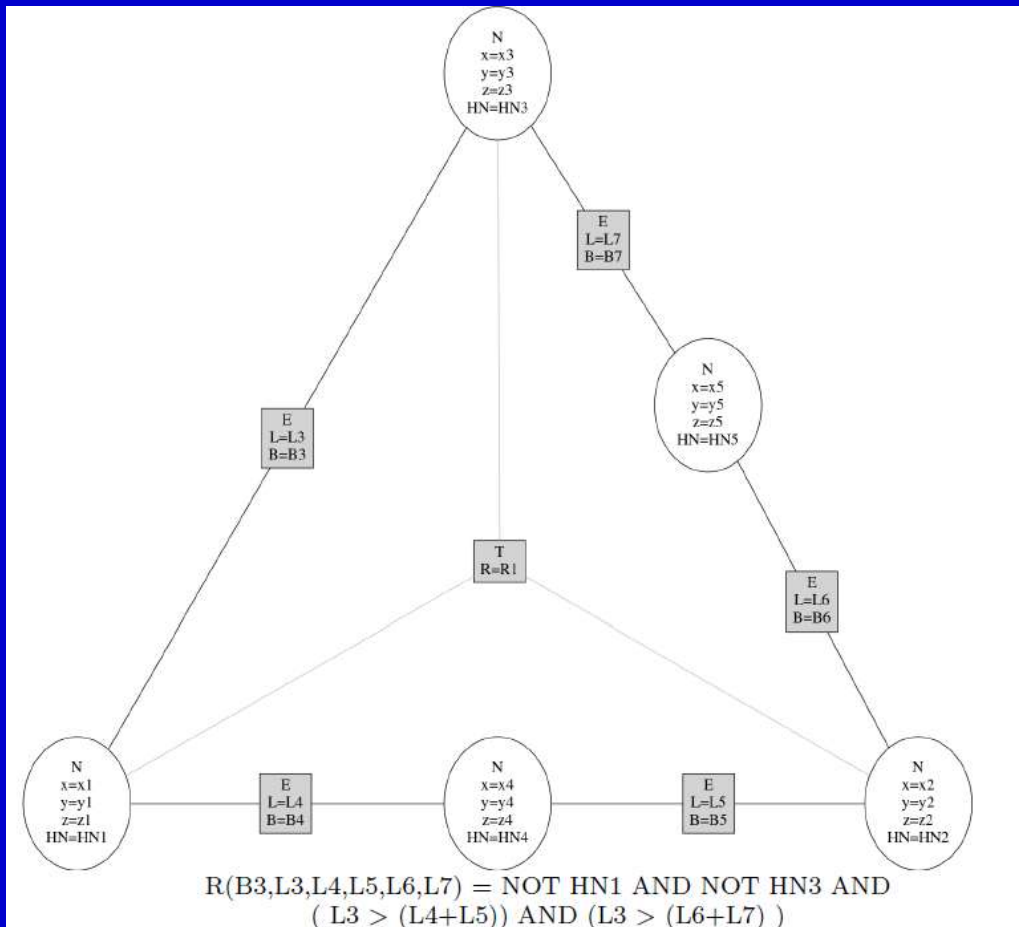
„naprawianie siatki”

- element ma już 2 hanging nodes
- hanging node jest na najdłuższej krawędzi

Wtedy łamiemy element i hanging node zamienia się na node

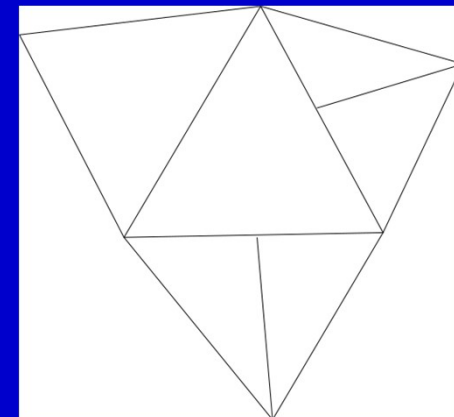


GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P5 (L)



„naprawianie siatki”

- element ma już 2 hanging nodes
- hanging node NIE jest na najdłuższej krawędzi
- wierzchołki należące do najdłuższej krawędzi nie są hanging nodes



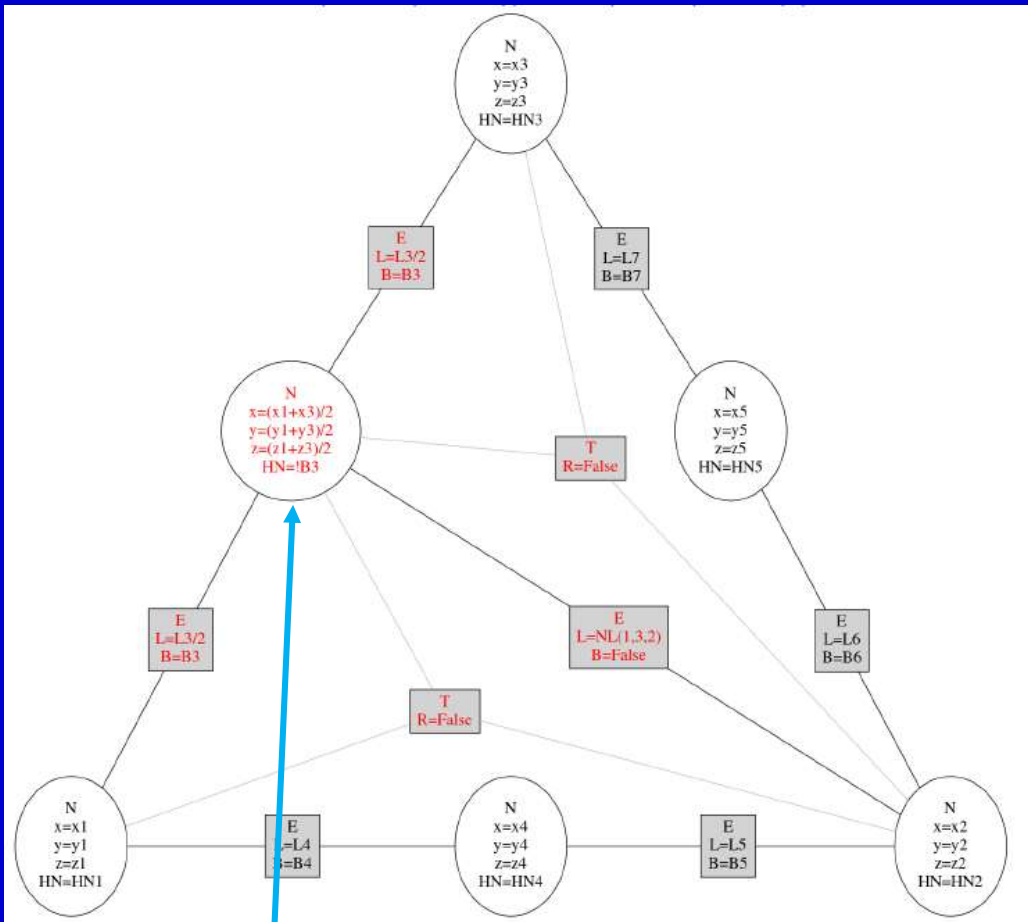
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P5 (R)

„naprawianie siatki”

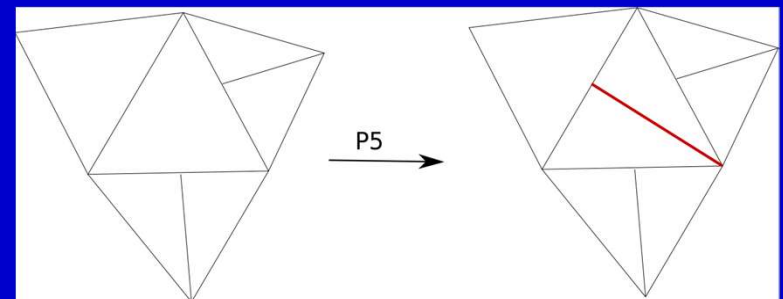
- element ma już 2 hanging nodes
- hanging node NIE jest na najdłuższej krawędzi
- wierzchołki należące do najdłuższej krawędzi nie są hanging nodes

Wtedy:

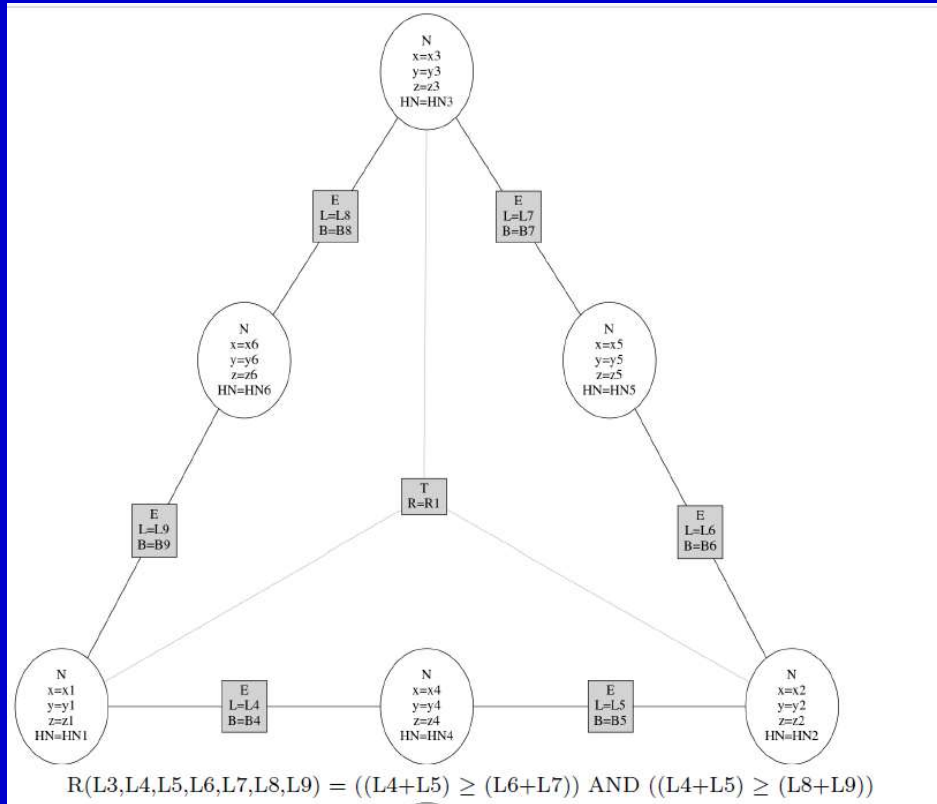
- łamiemy element
- nowoutworzony wierzchołek nie będzie hanging node jeśli był na brzegu,
Będzie hanging node jeśli nie był na brzegu



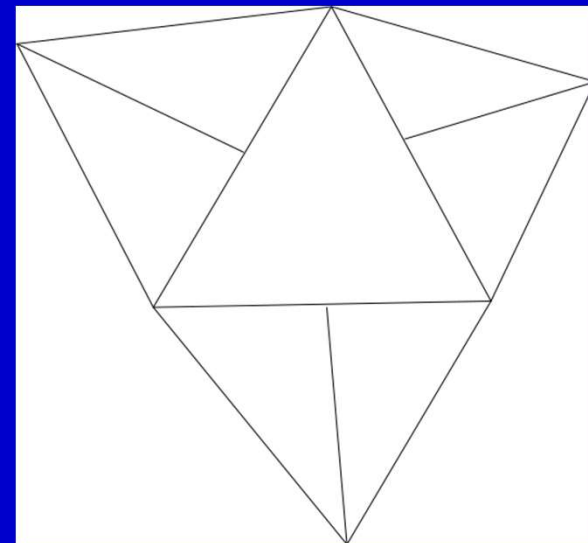
HN=!B3



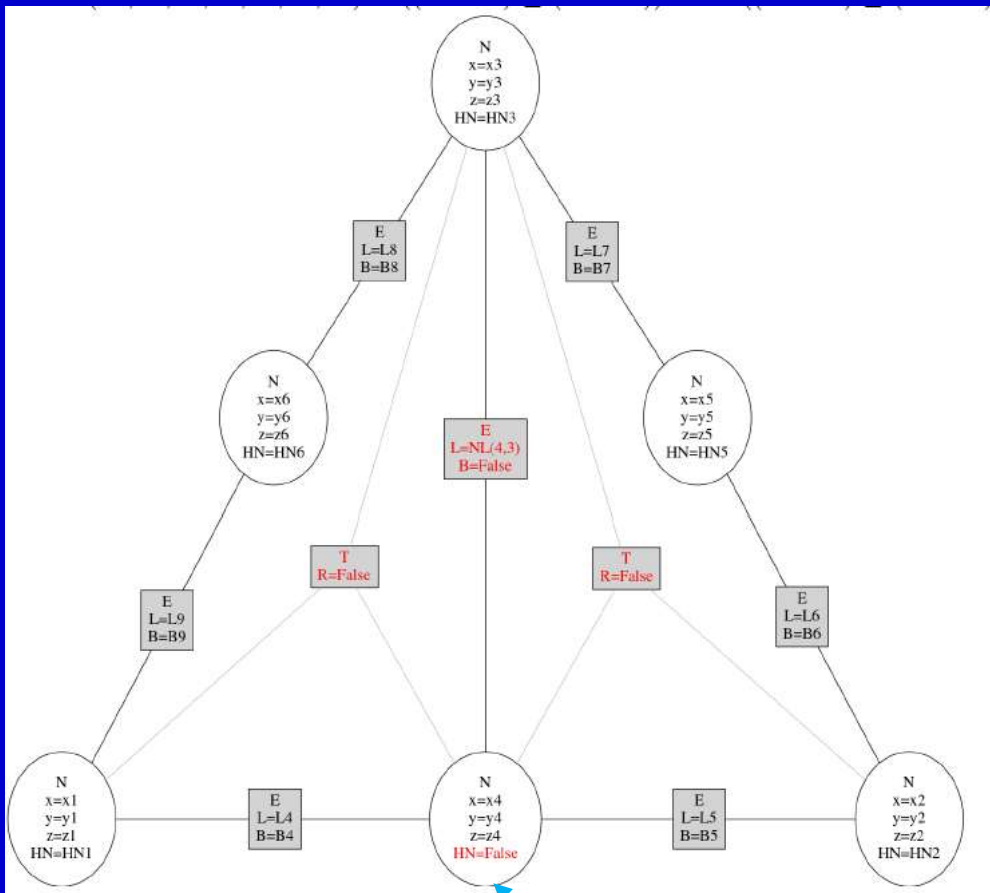
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P6 (L)



„naprawianie siatki”
 -element ma już 3 hanging nodes
 -łamiemy najdłuższą krawędź



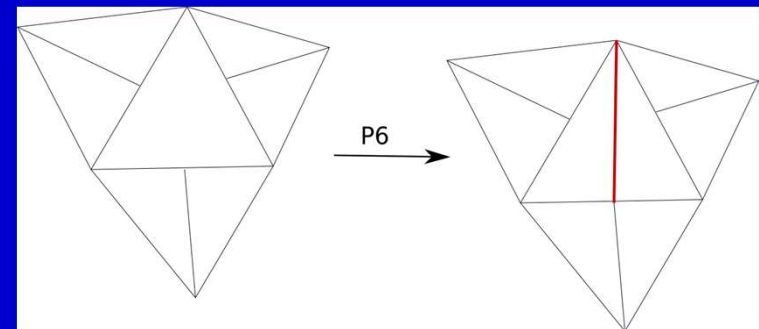
GRAMATYKA GRAFOWA DLA ALGORYTMU RIVARA P6 (R)



„naprawianie siatki”

- element ma już 3 hanging nodes
- łamiemy najdłuższą krawędź

Wtedy wierzchołek, który był hanging node staje się node



HN=FALSE