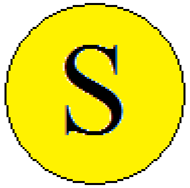
CP-graph grammar on the example of 2D mesh generation and direct solver

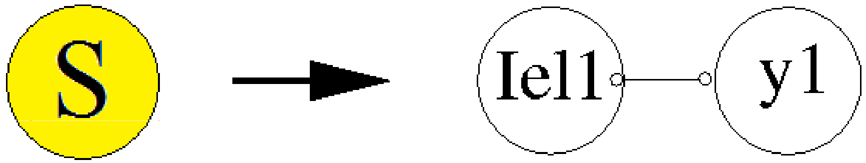
Maciej Paszynski

AGH University, Krakow, Poland

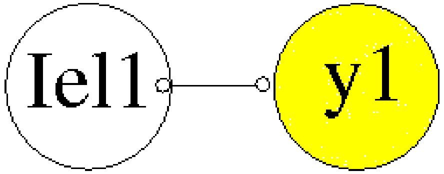
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (1/9)



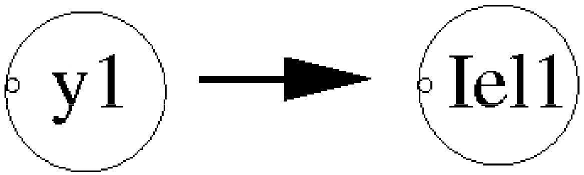
(P1)



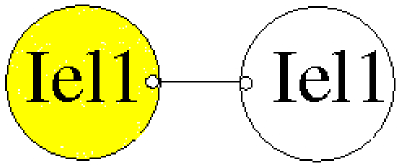
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (2/9)



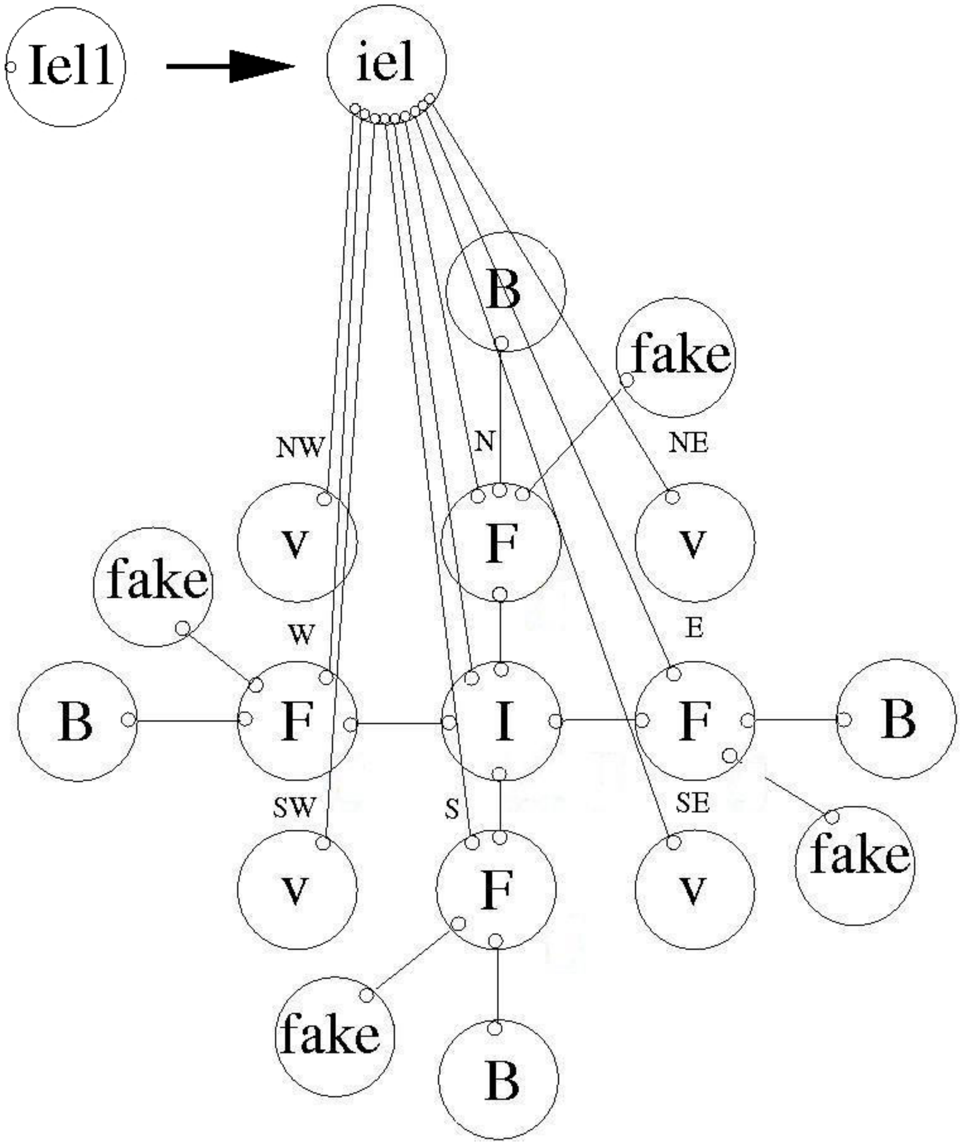
(P2)



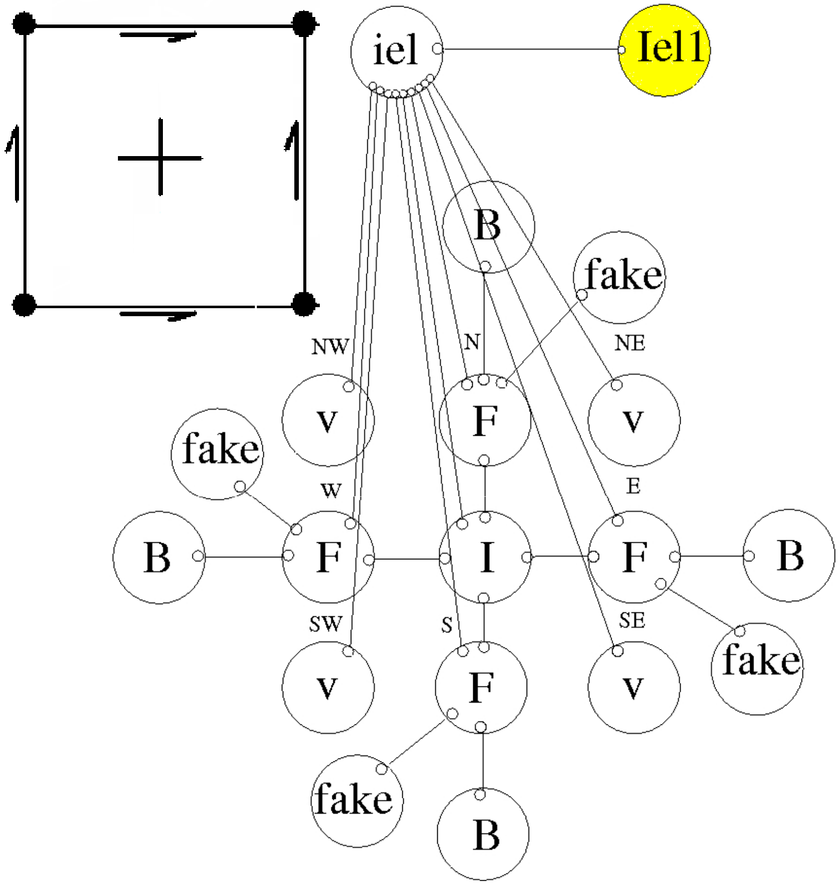
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (3/9)



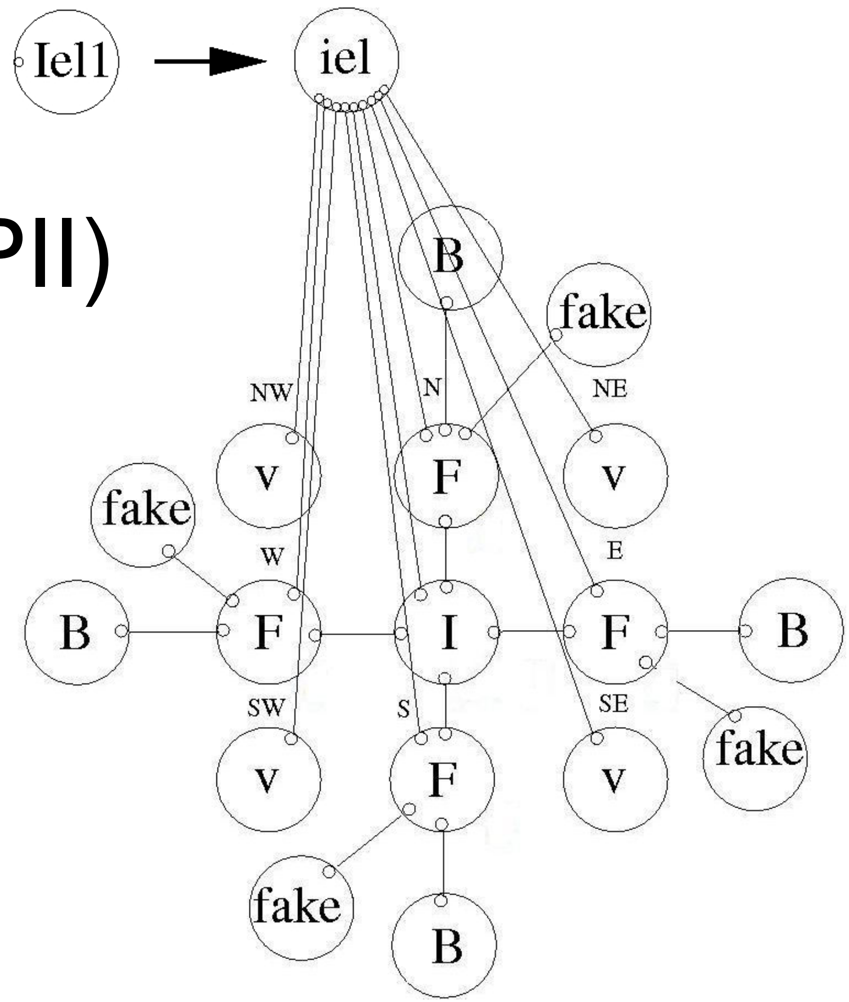
(PII)



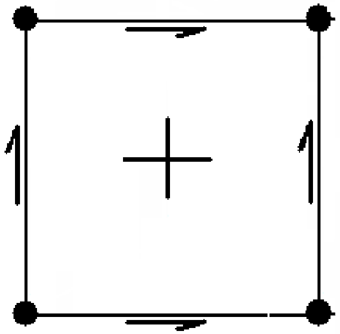
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (4/9)

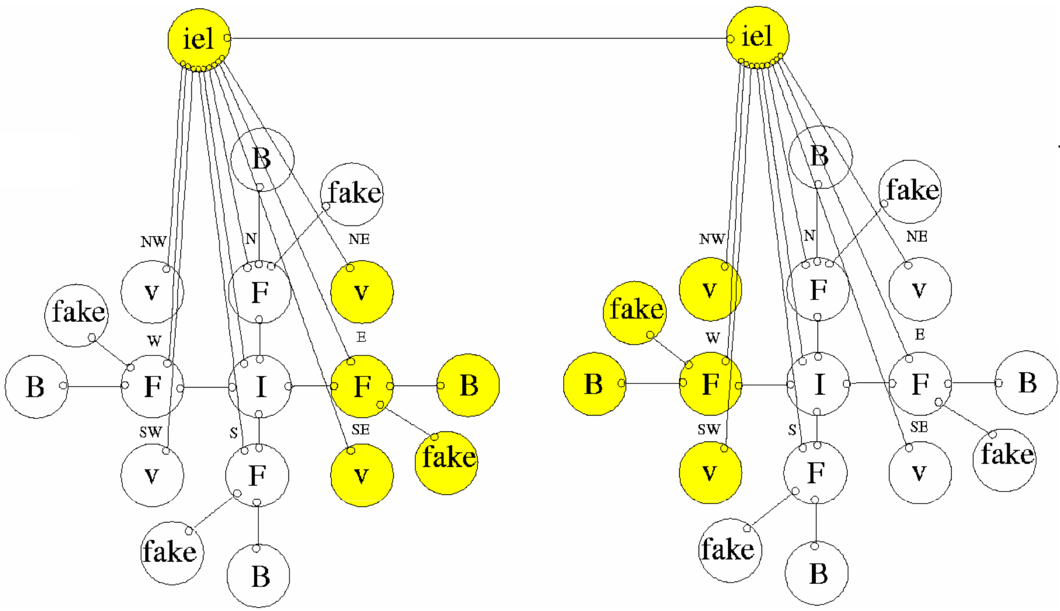


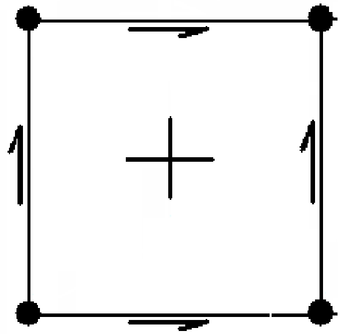
(PII)



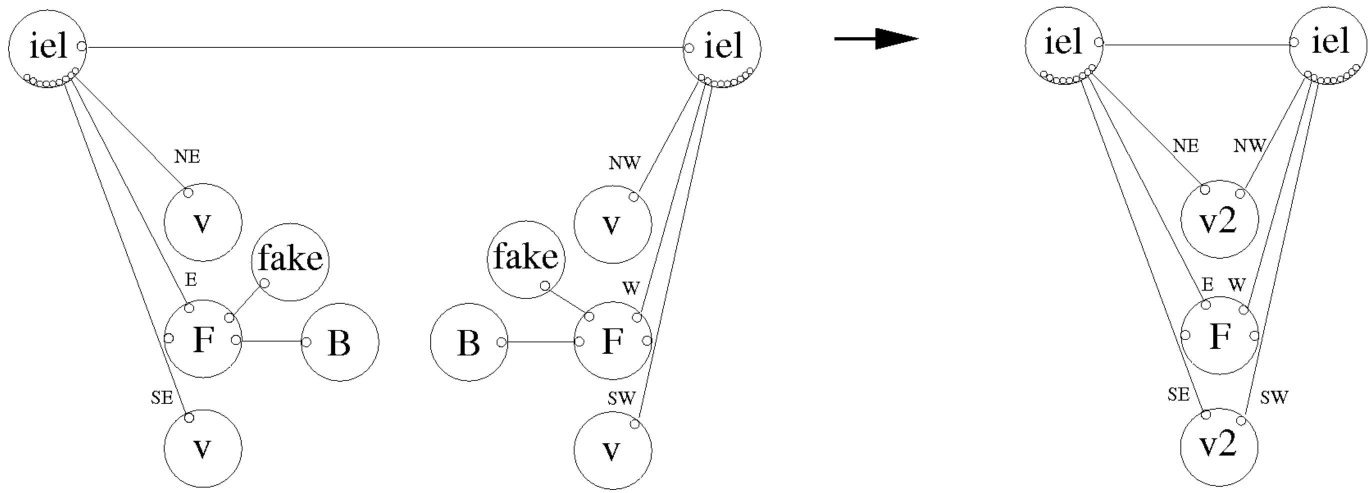
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (5/9)



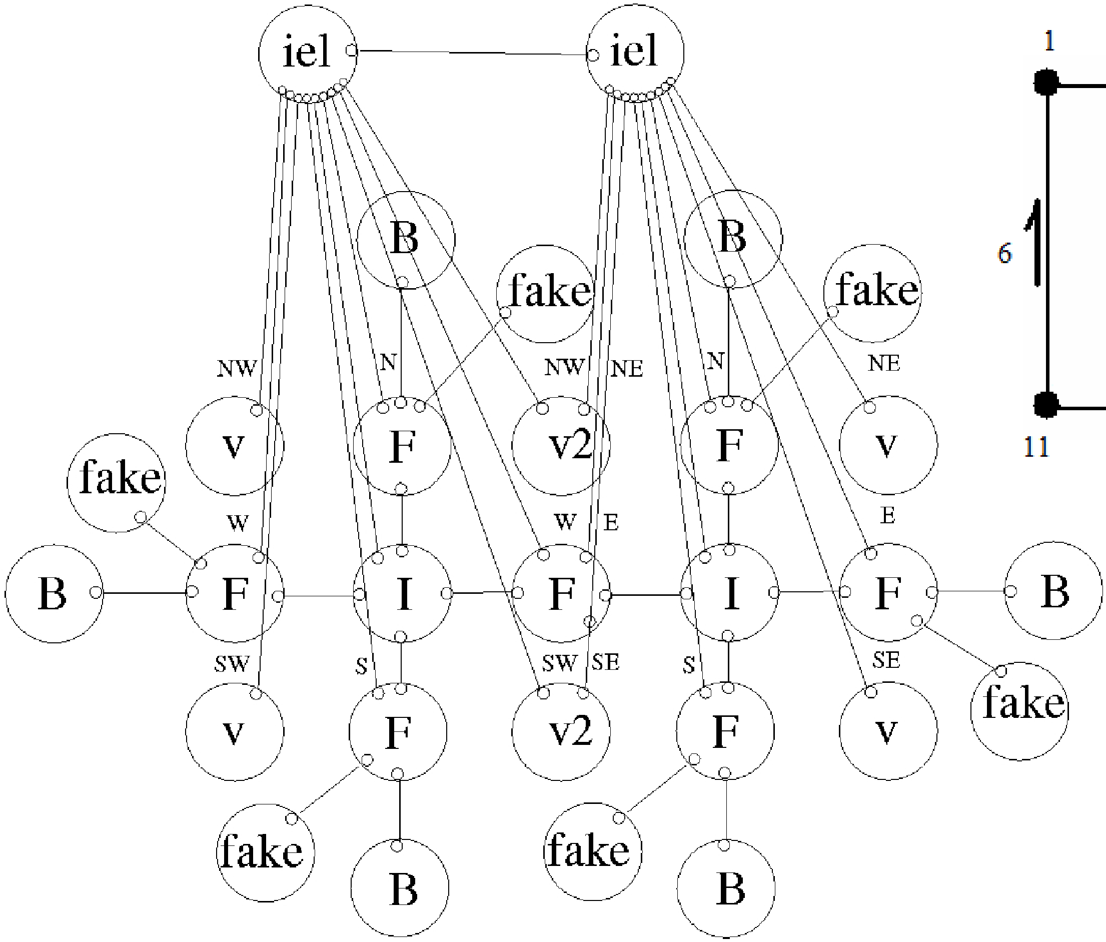


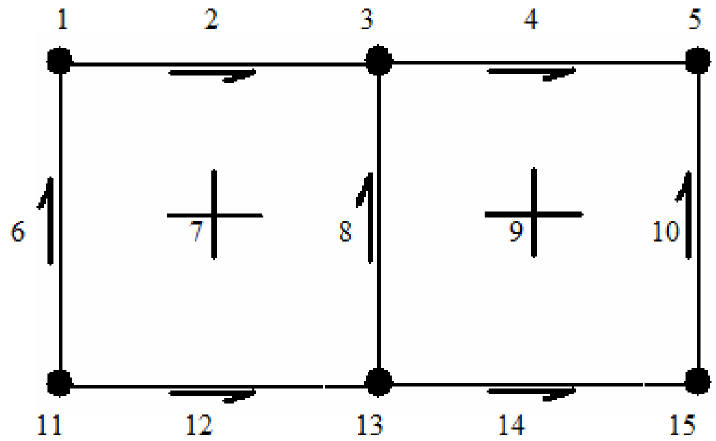


(PIC)



APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (6/9)





(P1)-(P2)-$(PII)^2$-(PIC)

DEFINITION OF
COMPOSITE PROGRAMMABLE GRAPH (CP-GRAPH)

Composite programmable graph (CP-graph) over $W = A_V \times [i]_N \times [i]_N$ and A_E

$c = ((V, E, s, t), \xi_V, \xi_E, att_V, att_E)$

where

V set of graph vertices

$\xi_V : V \rightarrow A_V \times [i]_N \times [i]_N$ (label, in-bound index, out-bound index)

A_V set of vertex labels

$[i]_N = \{\{1\}, \{1,2\}, \{1,2,3\}, \ldots, \{1,2,\ldots,i\}\}$ numbering of bounds

$\beta_I(\xi_V(v))$ projection onto in-bound index

$\beta_o(\xi_V(v))$ projection onto out-bound index

$B_0(V) = \bigcup_{v \in V} \beta_o(\xi_V(v)) \times \{v\}$ (out-bound index, vertex)

$B_I(V) = \bigcup_{v \in V} \beta_I(\xi_V(v)) \times \{v\}$ (in-bound index, vertex)

$E \subseteq B_0(V) \times B_I(V)$ graph edges, such that

- $\forall ((v, j), (i, u)) \in E, v \neq u$ (no edges to itself)
- $\forall (v, j) \in B_0(V)$ there exist at most one bound $(i, u) \in B_i(V)$ such that $((v, j), (i, u)) \in E$ (no multiple edges to single out-bound)
- $\forall (i, u) \in B_i(V)$ there exist at most one bound $(v, j) \in B_0(V)$ such that $((v, j), (i, u)) \in E$ (no multiple edges from single in-bound)

$s : E \rightarrow V$, $t : E \rightarrow V$ source, target vertices

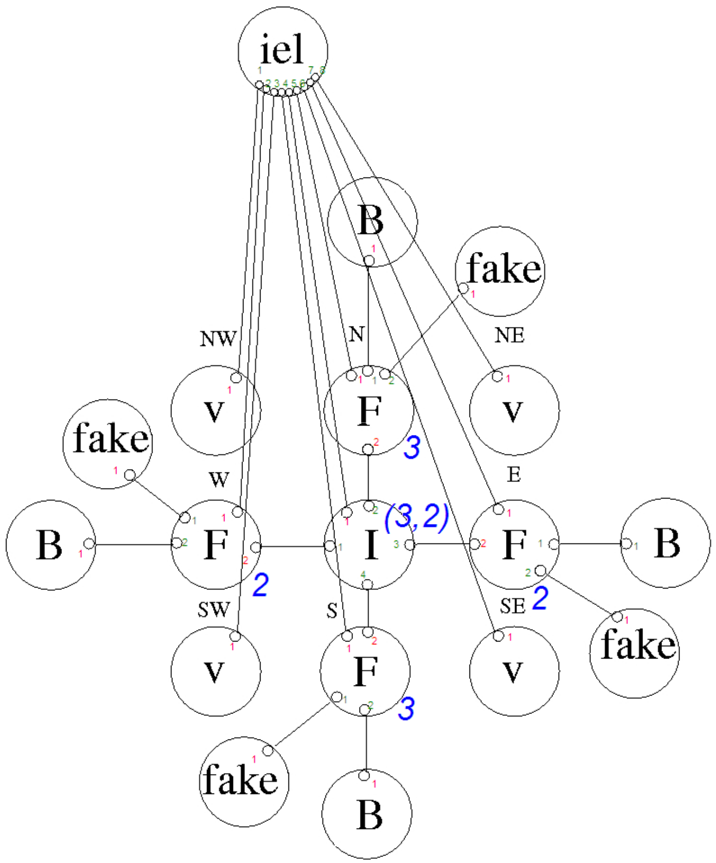
$s(e) = v$ and $t(e) = u$, $\forall e = ((v, j), (i, u)) \in E$

A_E set of edge labels

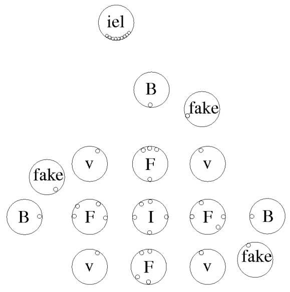
$\xi_E : E \rightarrow A_E$ edge labeling function

$att_V : V \rightarrow P(A_T)$ vertex label attributing function

$att_E : E \rightarrow P(A_R)$ edge label attributing function



EXAMPLE OF COMPOSITE PROGRAMMABLE GRAPH (CP-GRAPH) REPRESENTING SINGLE *hp* FINITE ELEMENT



V set of graph vertices

vertex labeling function:

$\xi_V : V \to A_V \times [i]_N \times [i]_N$

(label, in-bound index, out-bound index)

vertex labels:

$A_V = \{iel, F, I, v, B, fake\}$

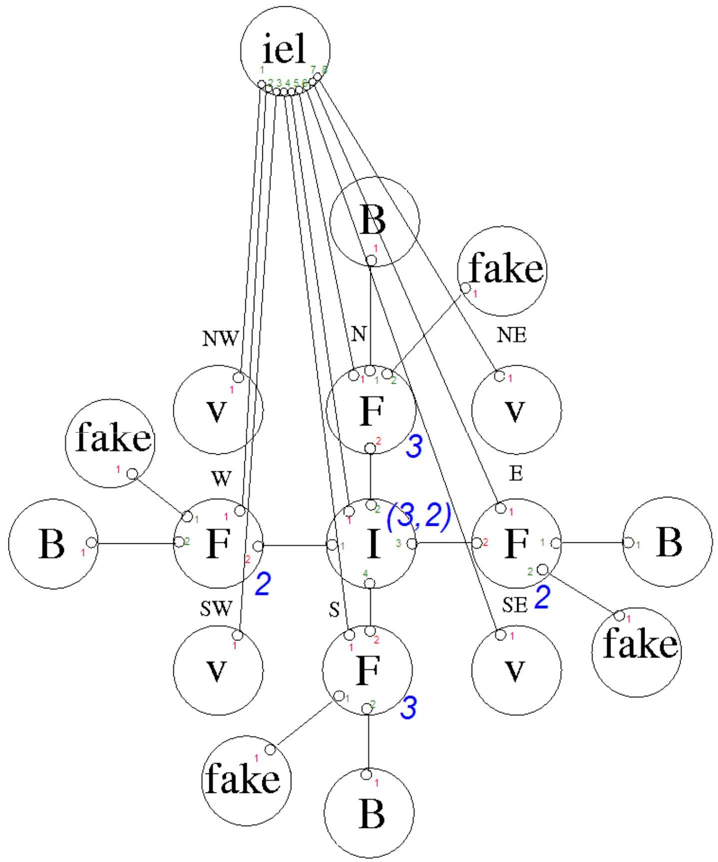
vertex attributing function:

$att_V : V \to P(A_T)$

vertex attributes:

$A_T = \{(1,1),(1,2),\ldots,(8,8)\} \cup \{1,\ldots,8\}$

(orders of approximation in middle nodes, in horizontal and vertical directions, orders of approximation for edges)



EXAMPLE OF COMPOSITE PROGRAMMABLE GRAPH (CP-GRAPH) REPRESENTING SINGLE *hp* FINITE ELEMENT

vertex labeling function:

$\xi_V : V \to A_V \times [i]_N \times [i]_N$

(label, in-bound index, out-bound index)

$\beta_I(\xi_V(v))$ projection onto in-bound index

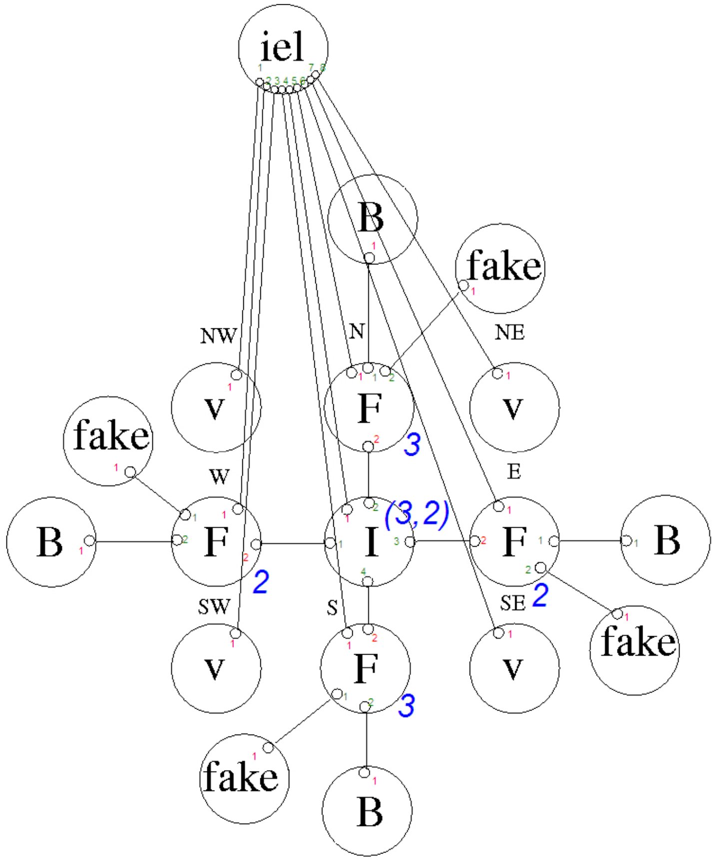
$\beta_o(\xi_V(v))$ projection onto out-bound index

$B_0(V) = \bigcup_{v \in V} \beta_o(\xi_V(v)) \times \{v\}$ (out-bound index, vertex)

$B_I(V) = \bigcup_{v \in V} \beta_I(\xi_V(v)) \times \{v\}$ (in-bound index, vertex)

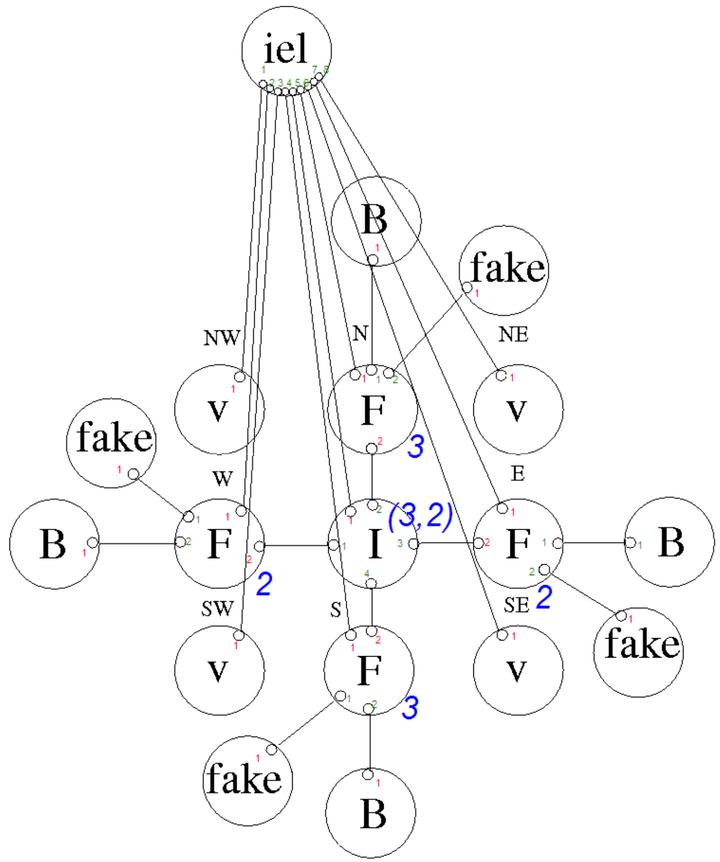
$E \subseteq B_0(V) \times B_I(V)$ graph edges, such that

- $\forall ((v,j),(i,u)) \in E,\ v \neq u$ (no edges to itself)
- $\forall (v,j) \in B_0(V)$ there exist at most one bound $(i,u) \in B_I(V)$ such that $((v,j),(i,u)) \in E$ (no multiple edges to single out-bound)
- $\forall (i,u) \in B_I(V)$ there exist at most one bound $(v,j) \in B_0(V)$ such that $((v,j),(i,u)) \in E$ (no multiple edges from single in-bound)



EXAMPLE OF
COMPOSITE PROGRAMMABLE GRAPH (CP-GRAPH)
REPRESENTING SINGLE *hp* FINITE ELEMENT

$E \subseteq B_0(V) \times B_I(V)$ graph edges



edge labeling function:

$\xi_E : E \rightarrow A_E$

A_E edge labels:

$A_E = \{N, S, W, E, NW, NE, SW, SE\}$

edge label attributing function

$att_E : E \rightarrow P(A_R)$

(void in this example)

edge attribute

$A_R = \emptyset$

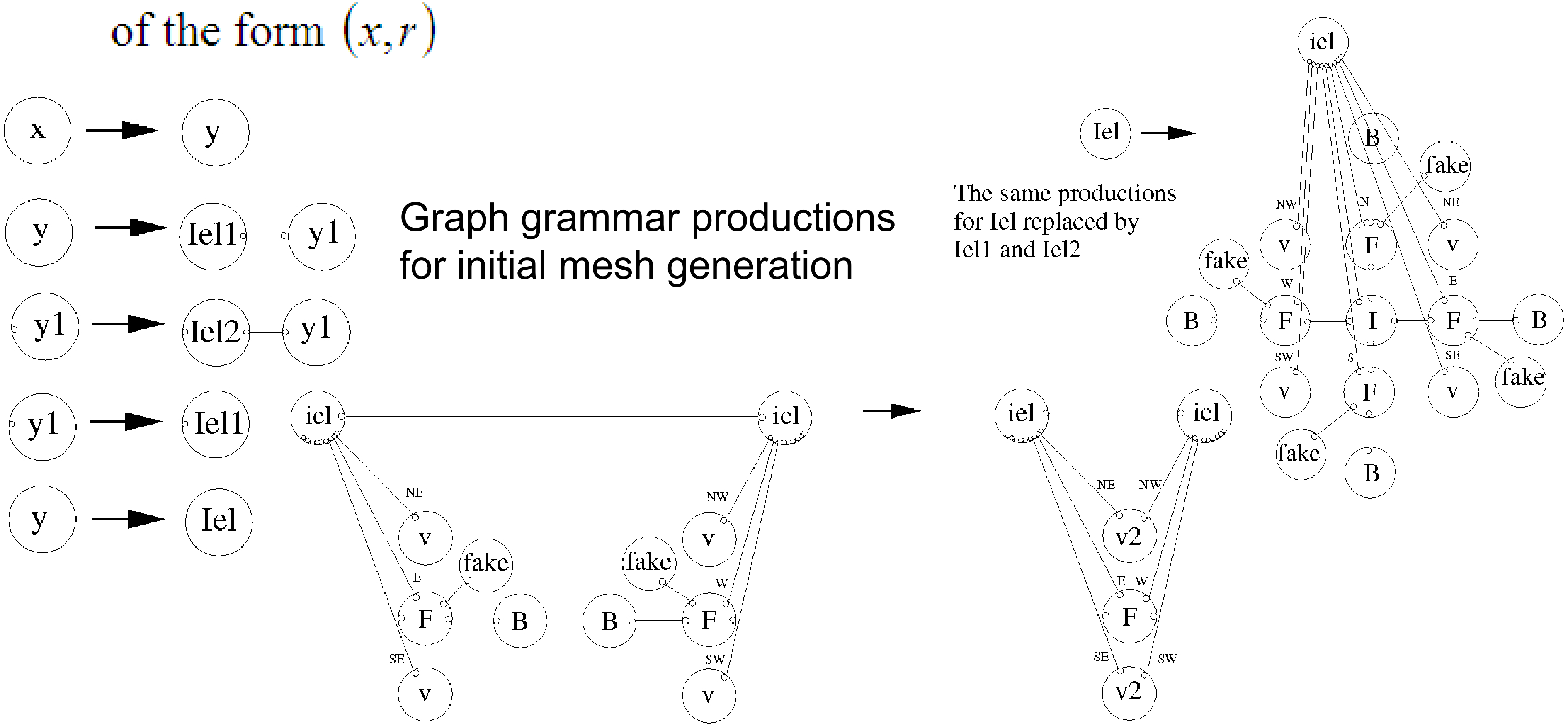
(empty in this example)

DEFINITION OF GRAPH GRAMMAR PRODUCTIONS

Composite programmable graph grammar (CP-graph grammar) over $W = A_V \times [i]_N \times [i]_N$ and A_E

$G = (P, x)$

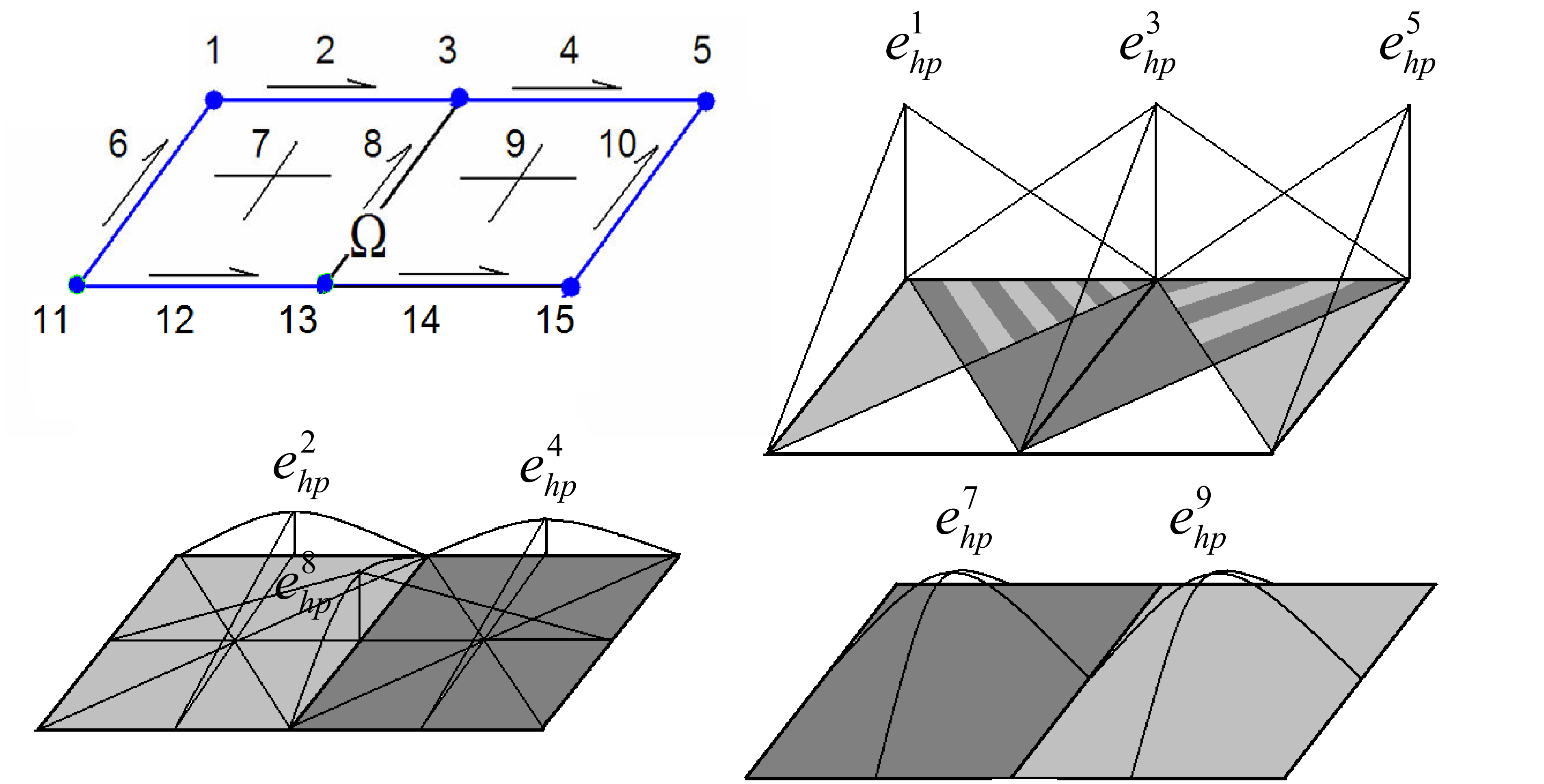
1. P is finite set of productions of the form (l, r) where l and r are CP-graphs with the same number of in- and out-bounds
2. x called the axiom symbol, such that there is at least one production of the form (x, r)



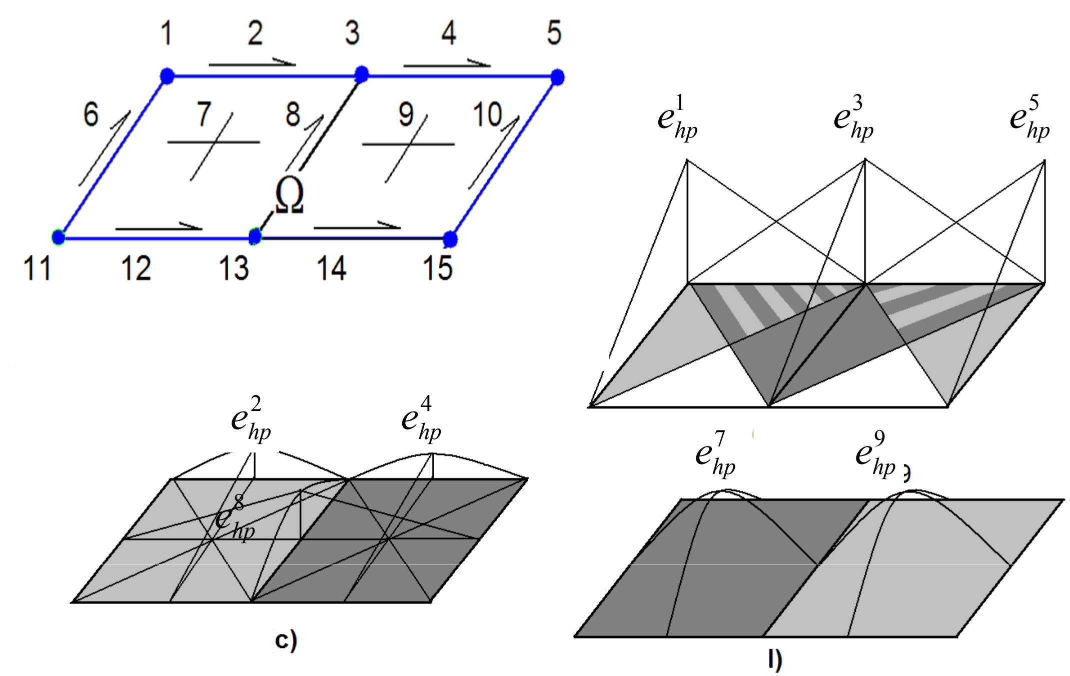
2D *hp* FINITE ELEMENT METHOD

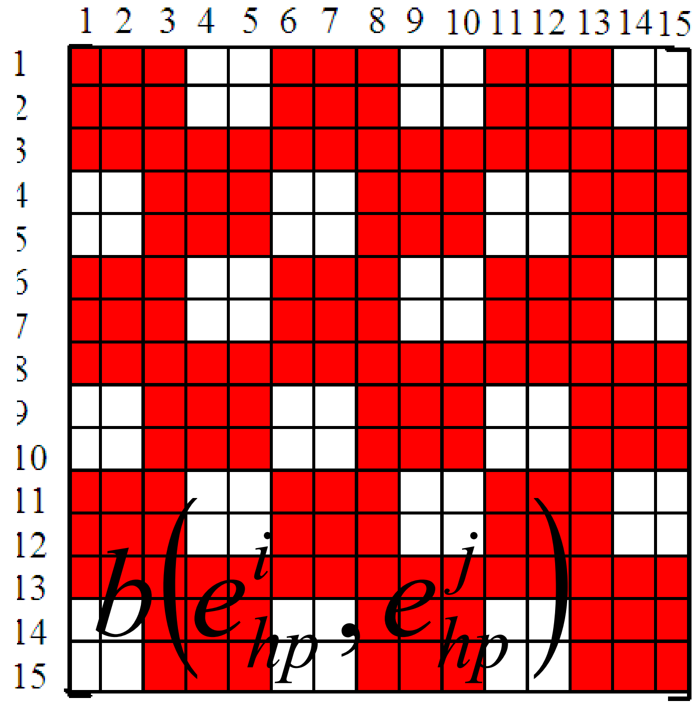
$$u = \sum_{i=1}^{15} u_{hp}^{i} e_{hp}^{i}$$

We seek the solution ***u*** of some weak form of PDE as a linear combination of shape functions e_{hp}^{i} spread over finite element mesh



2D *hp* FINITE ELEMENT METHOD





The coefficients u^i_{hp}

(called „degrees of freedom" **d.o.f.**)

are obtained by solving

system of linear equations –

finite element disecretization

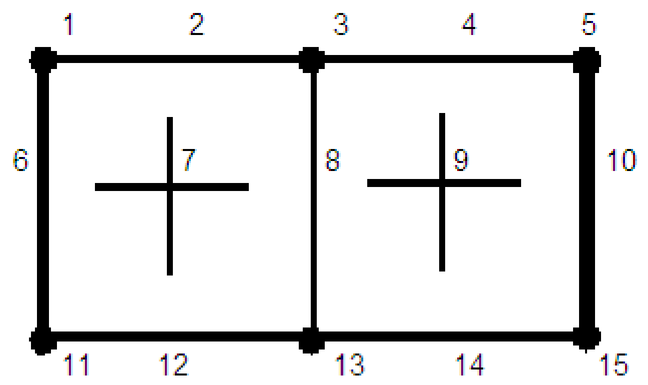
of a weak (variational) form of PDE

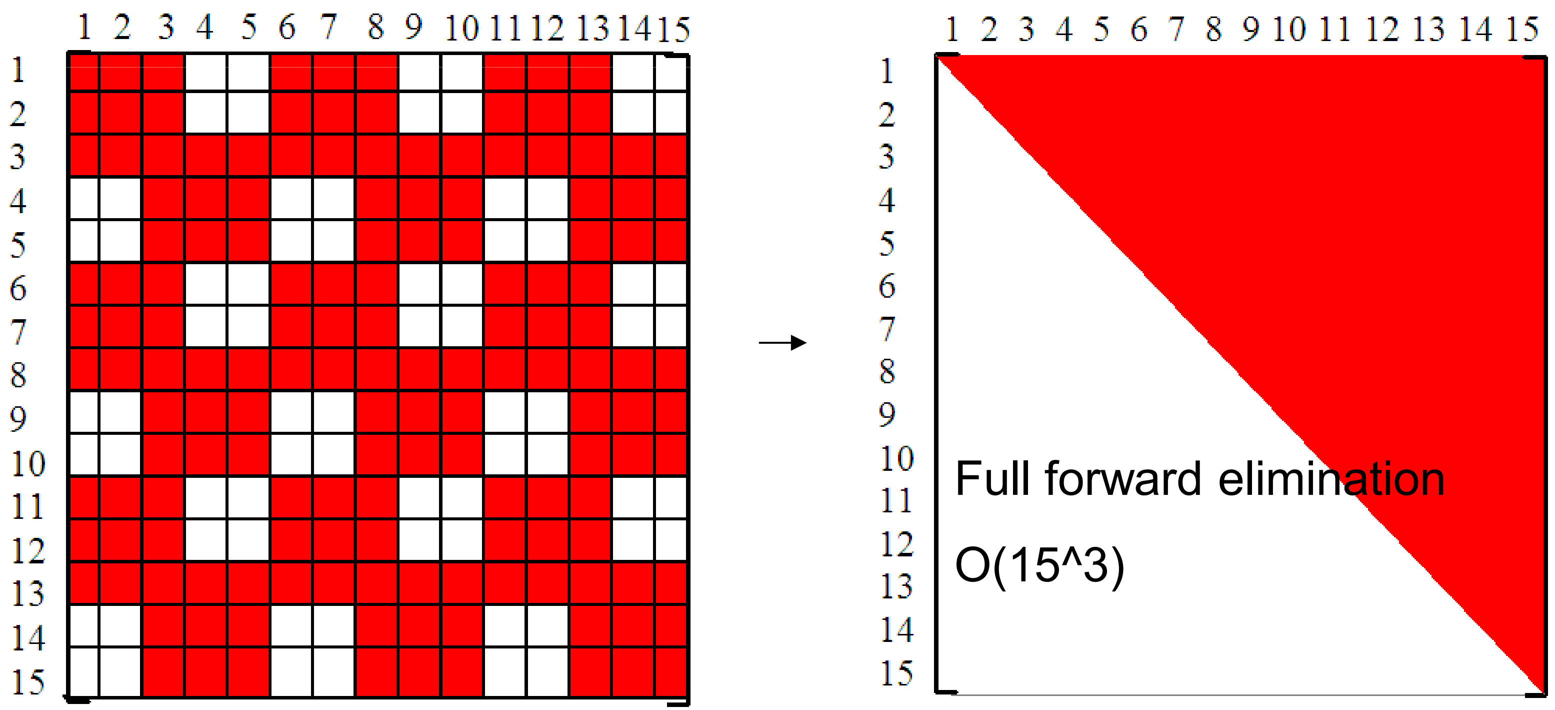
$$\sum_{i=1}^{15} u^i_{hp} b\left(e^i_{hp}, e^j_{hp}\right) = l\left(e^j_{hp}\right) \quad j = 1,\ldots,15$$

where $b\left(e^i_{hp}, e^j_{hp}\right)$ and $l\left(e^j_{hp}\right)$

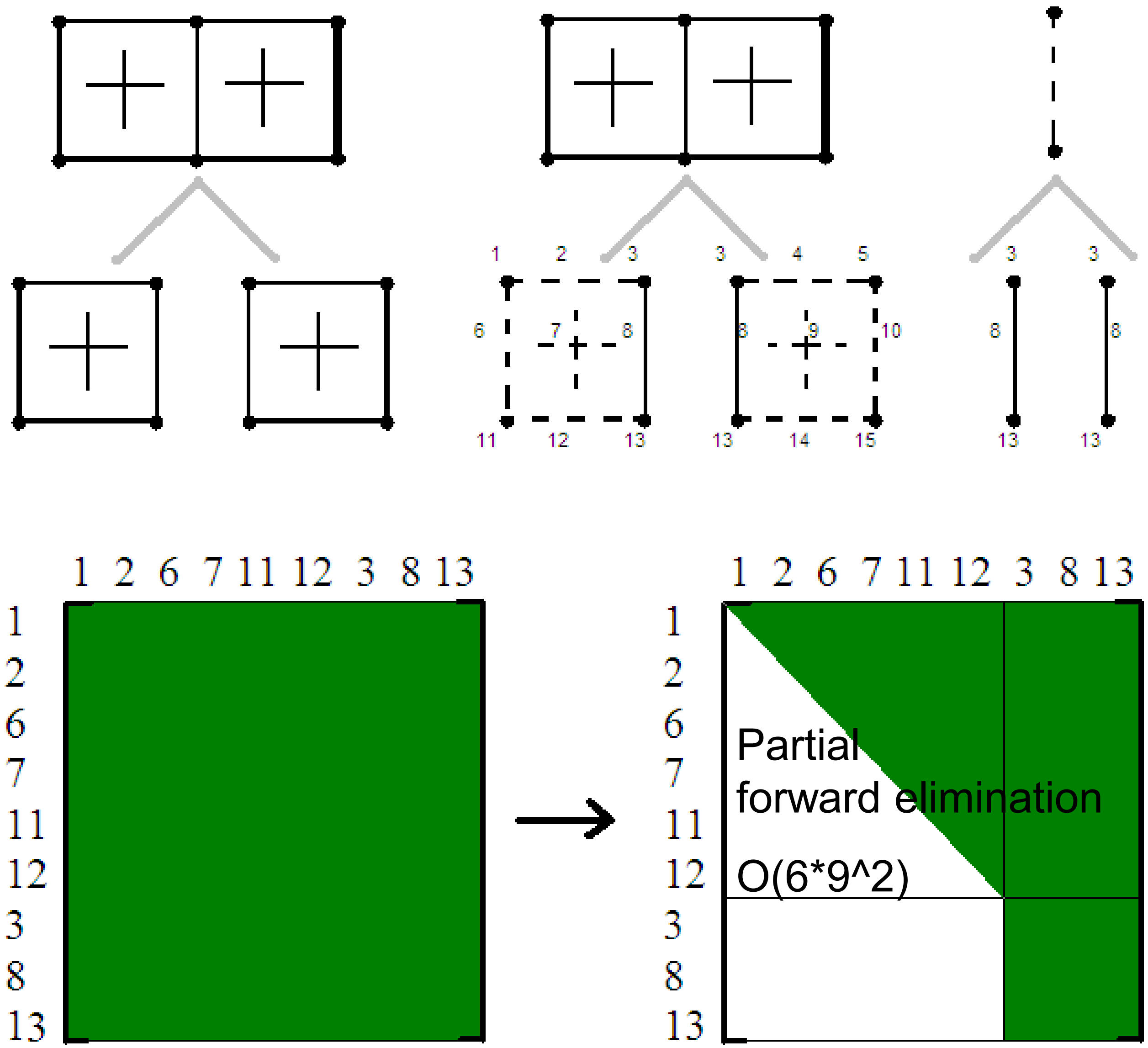
are some integrals of shape functions e^i_{hp}, e^j_{hp}

ORDERING ISSUES

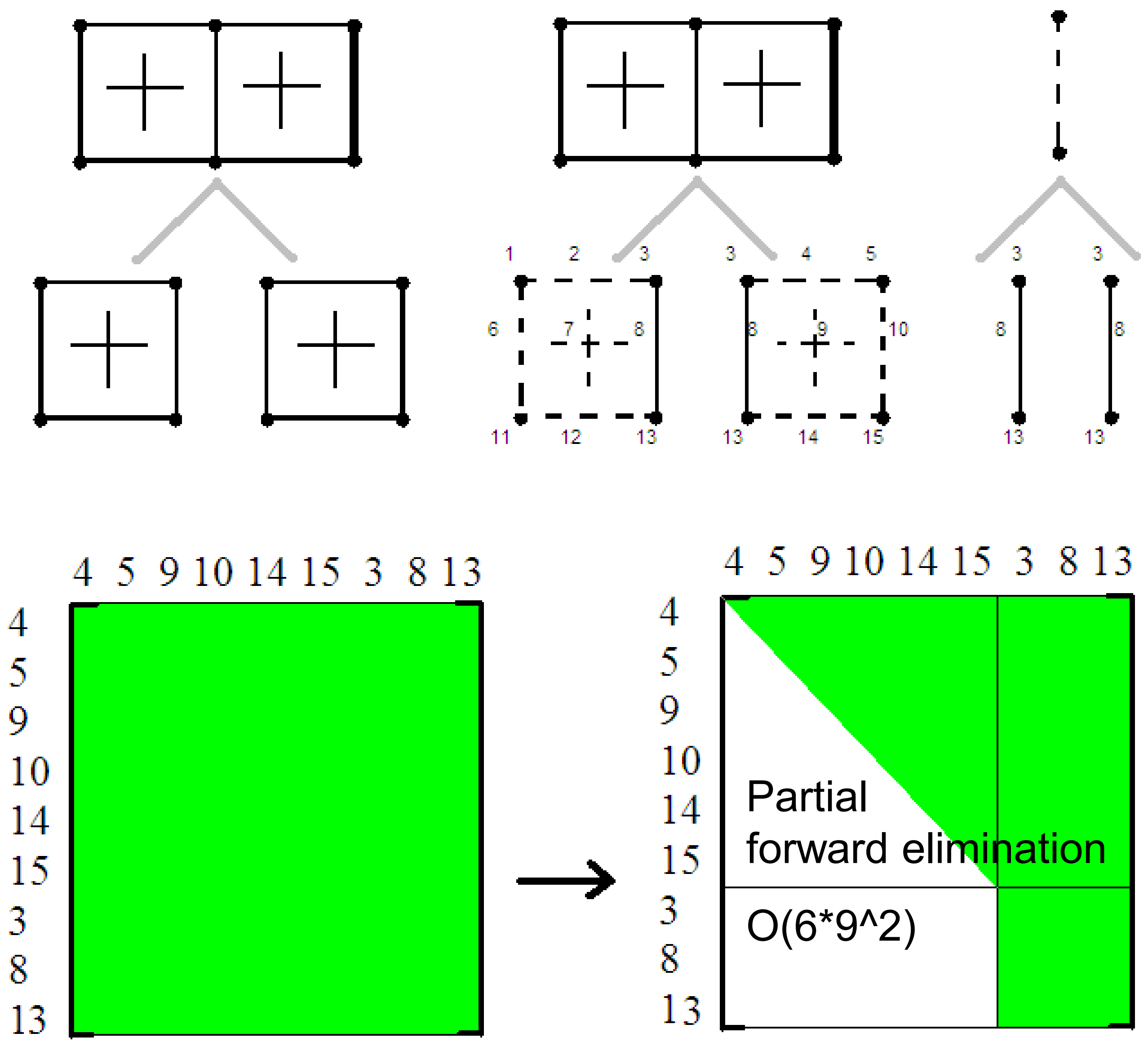




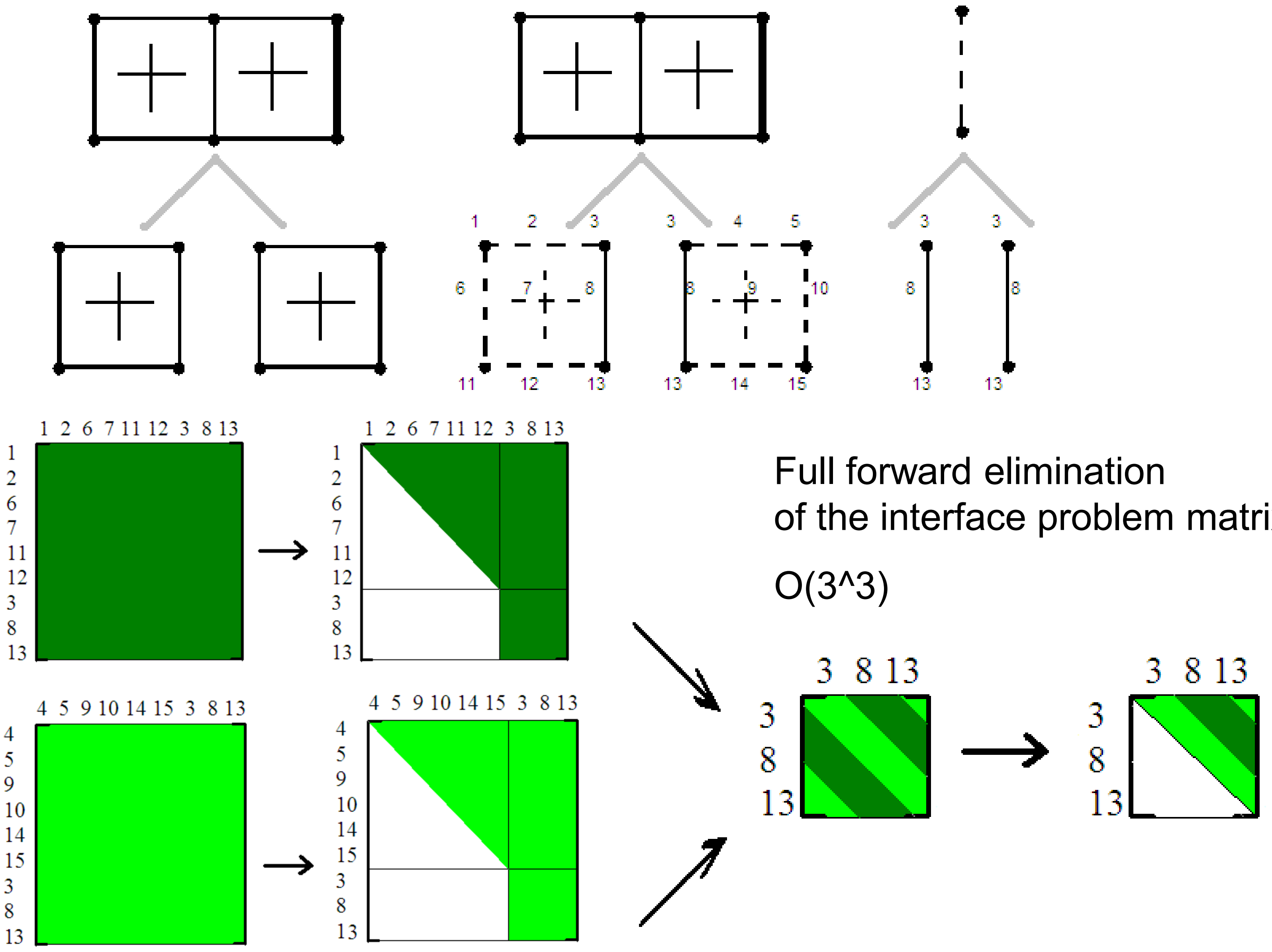
ORDERING ISSUES



ORDERING ISSUES



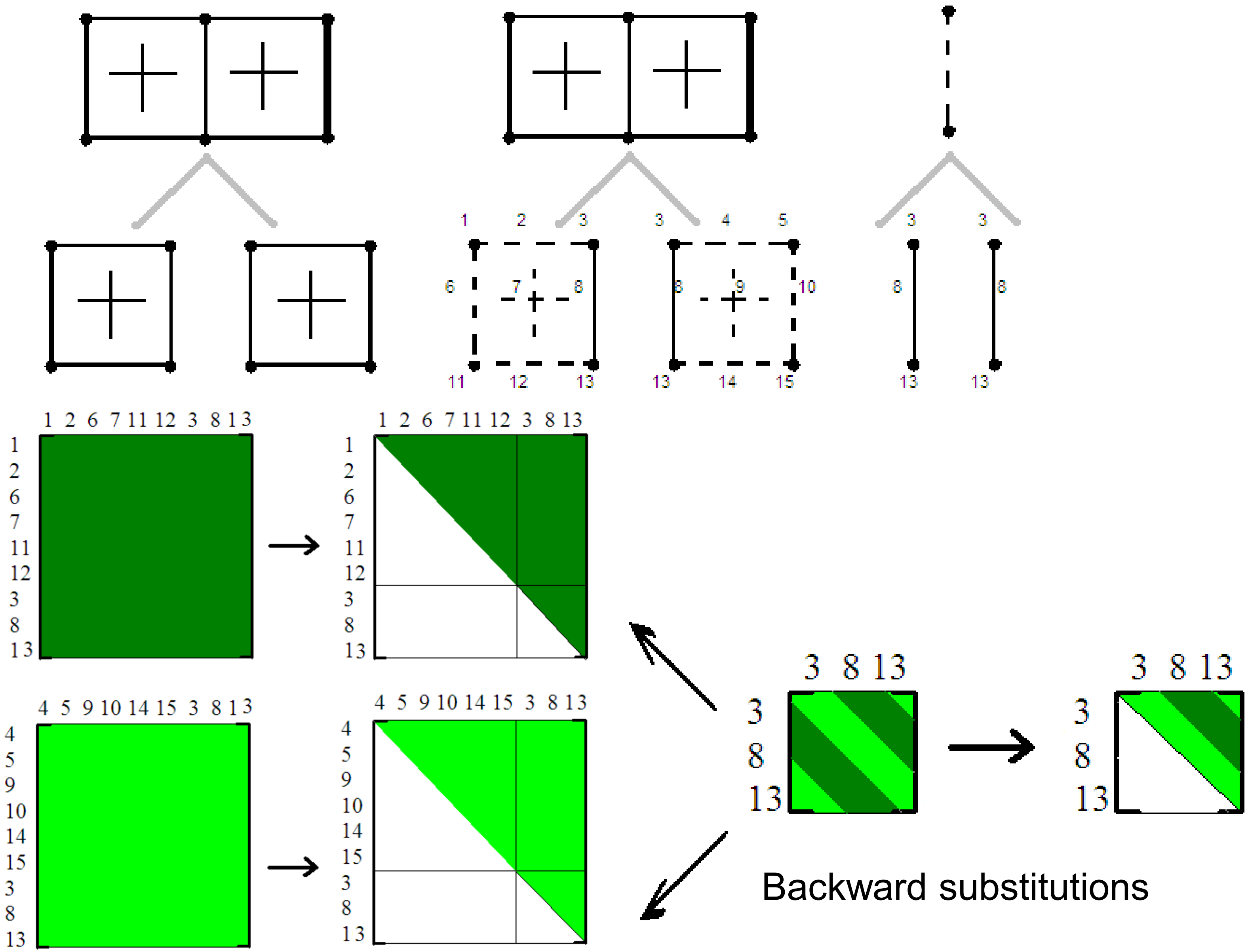
ORDERING ISSUES



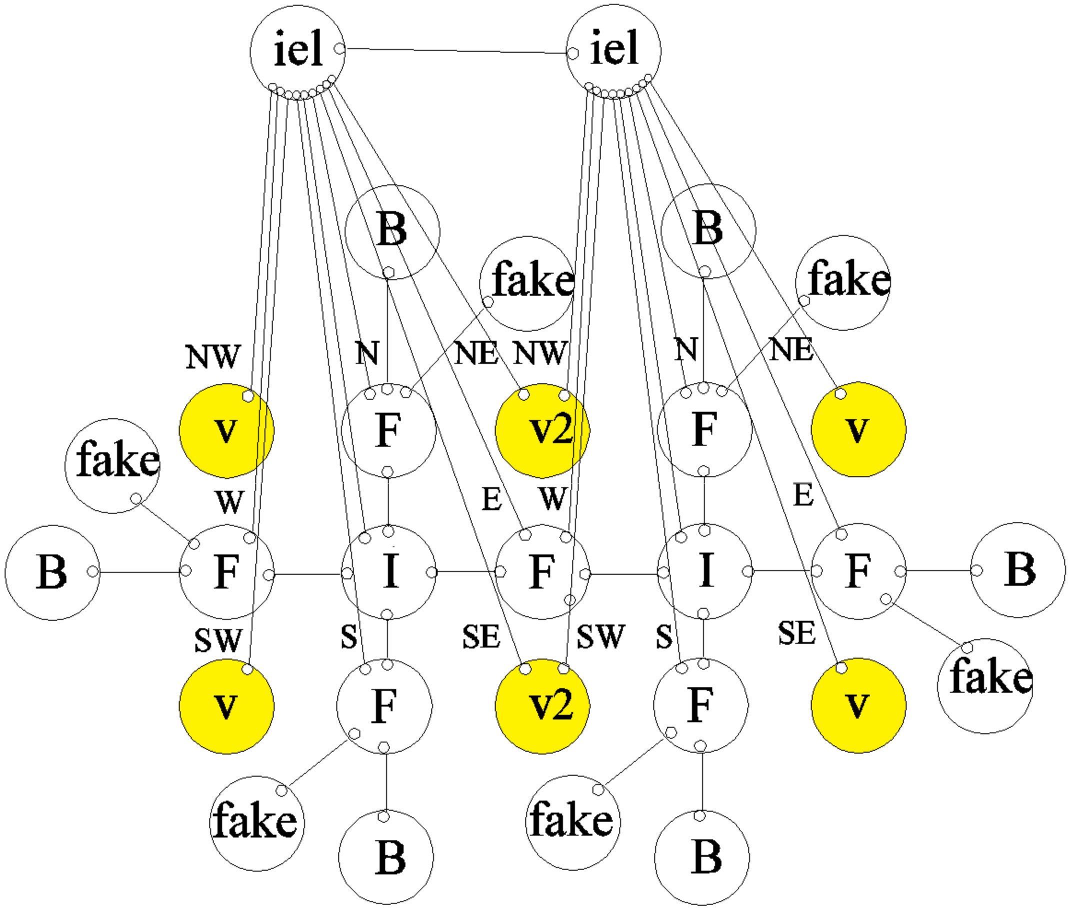
Full forward elimination
of the interface problem matrix

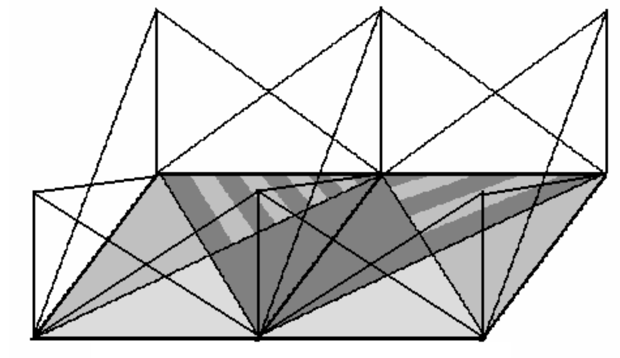
O(3^3)

ORDERING ISSUES

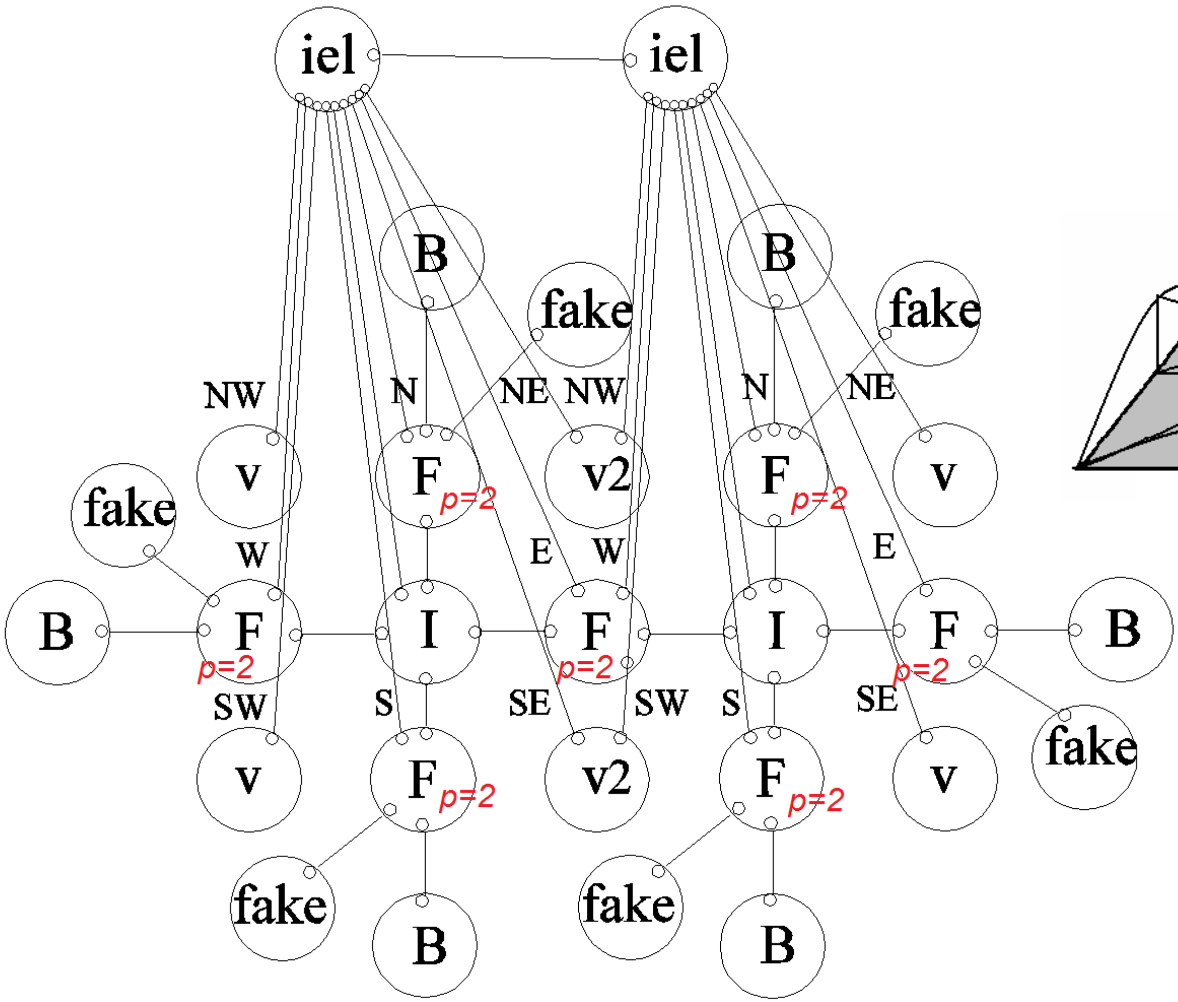


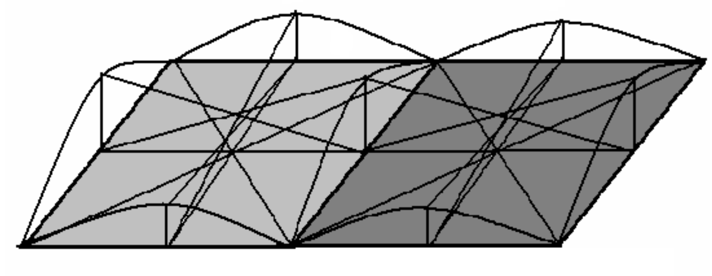
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (7/9)



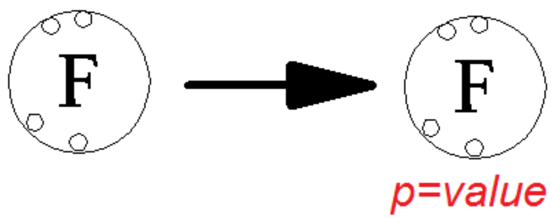


APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (8/9)

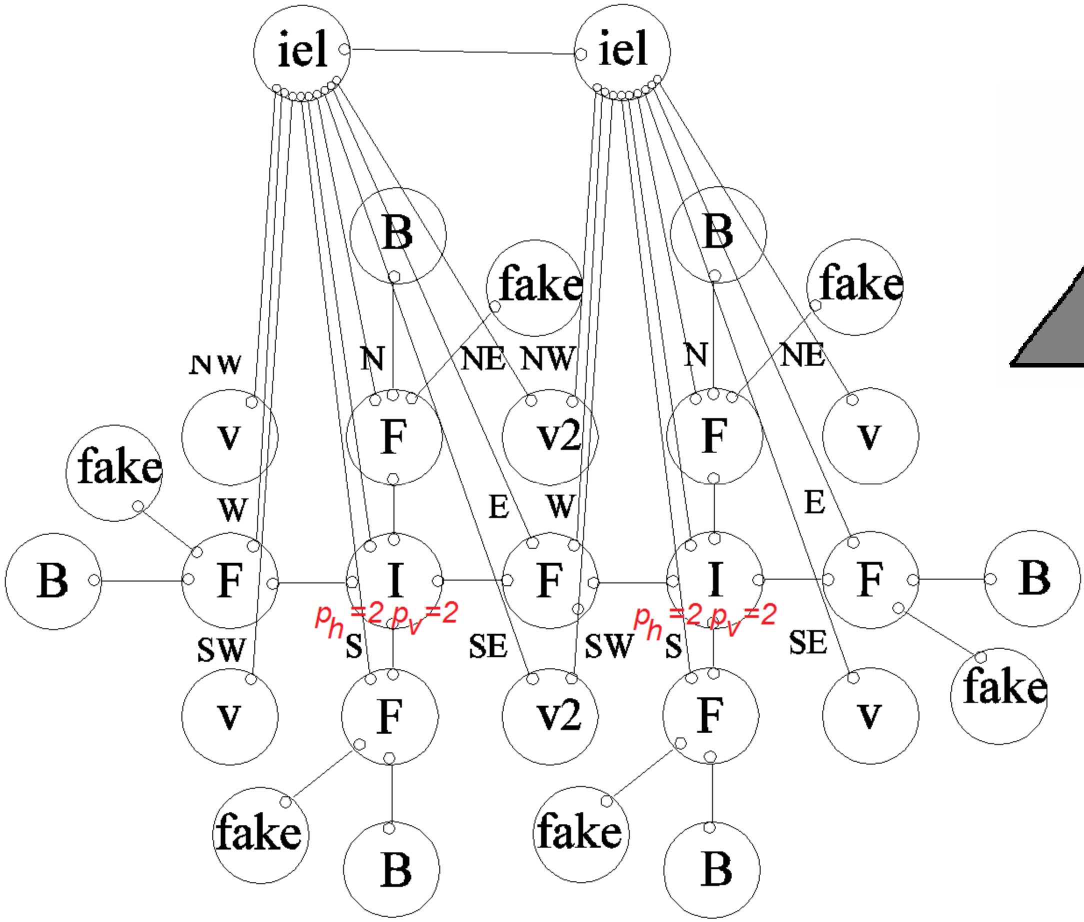


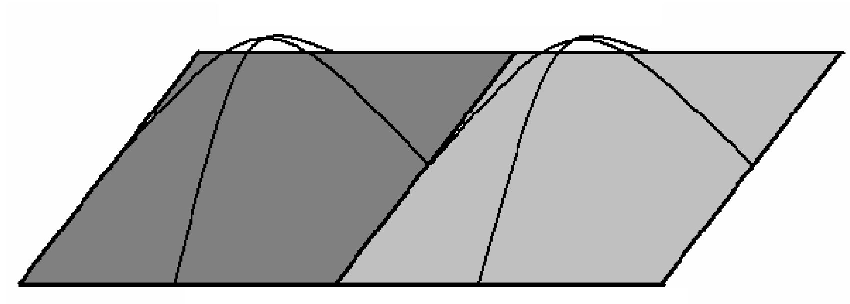


(Patt edge)

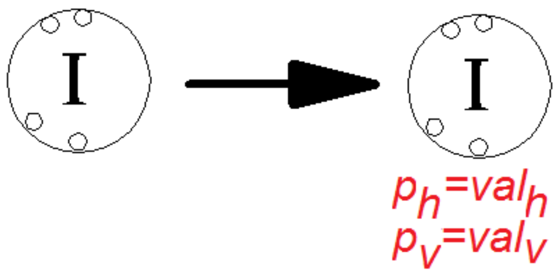


APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE STRUCTURE OF 2 ELEMENTS (9/9)

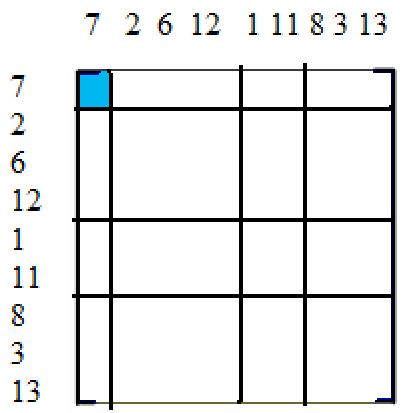


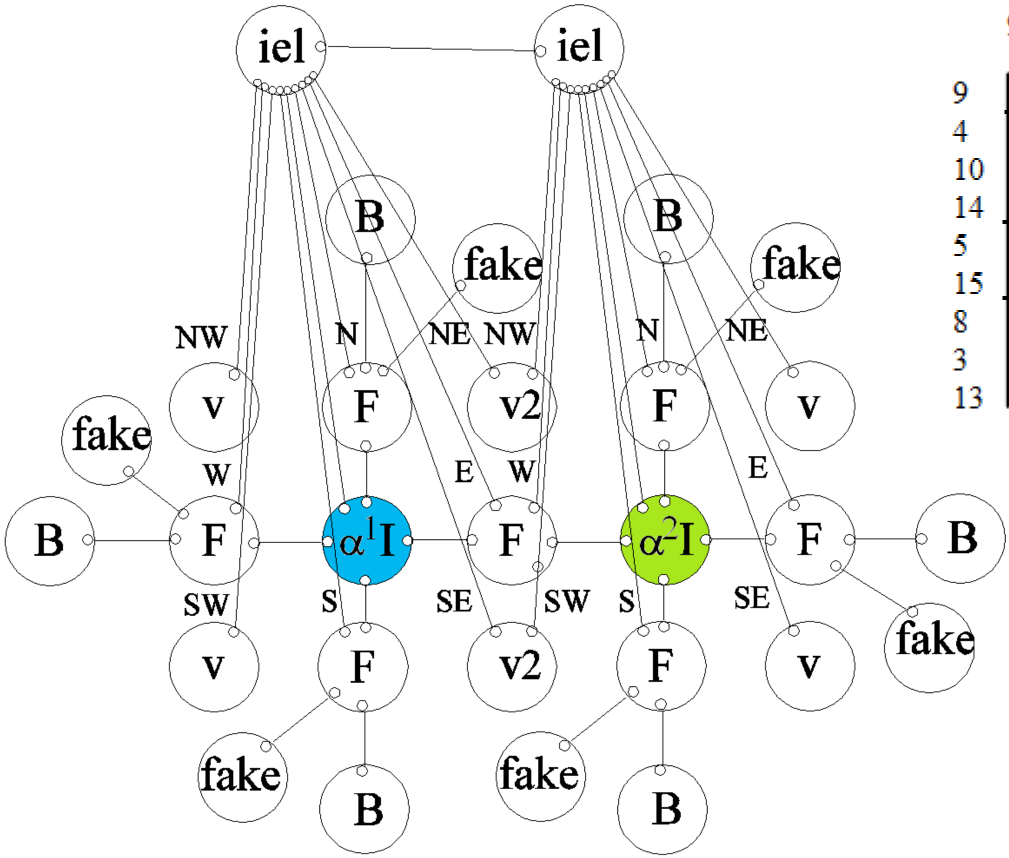


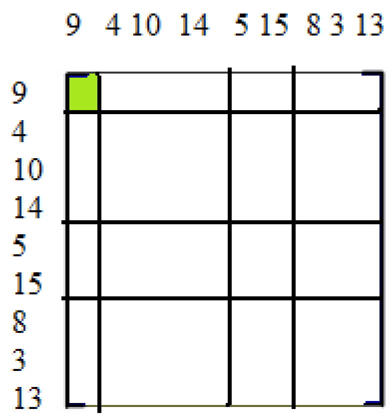
(Patt interior)



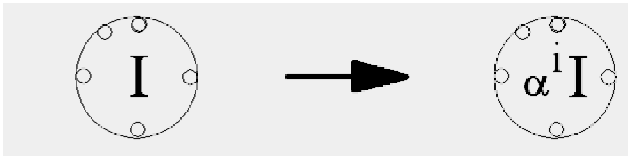
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE FRONTAL MATRICES (1/3)



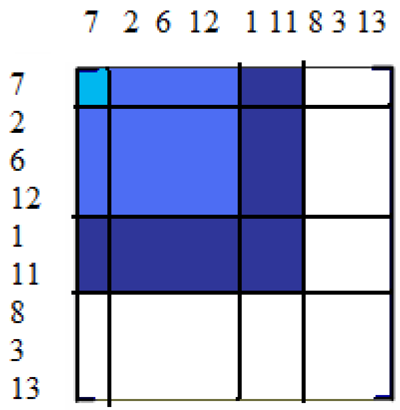


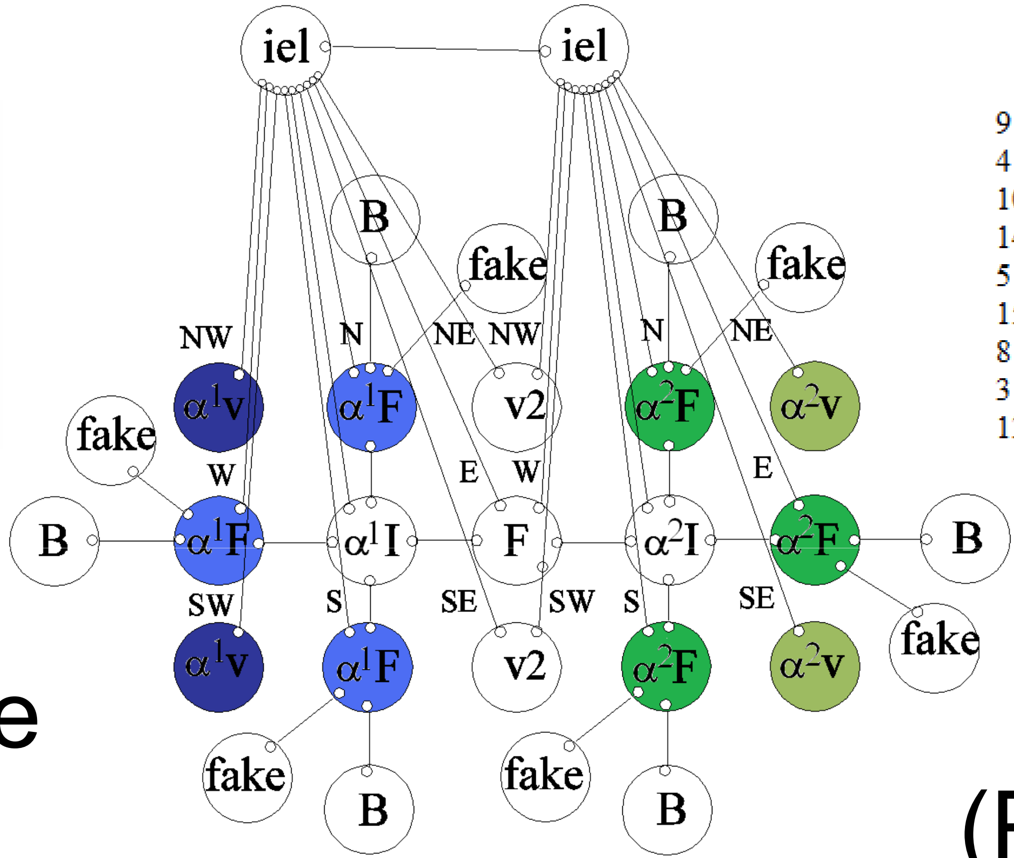


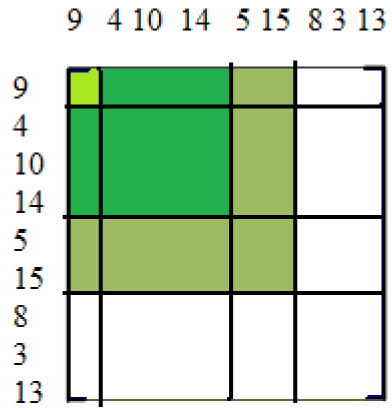
(Pagregate interior)



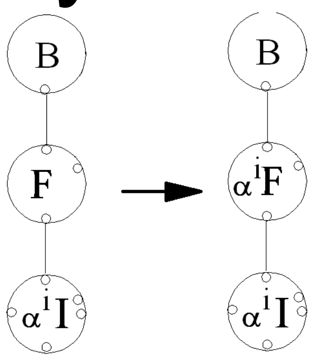
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE FRONTAL MATRICES (2/3)

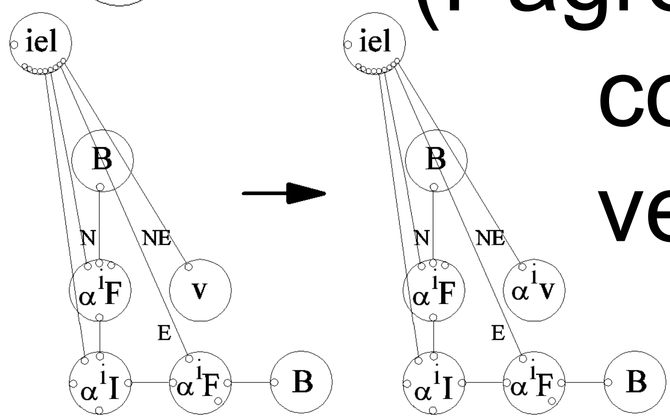






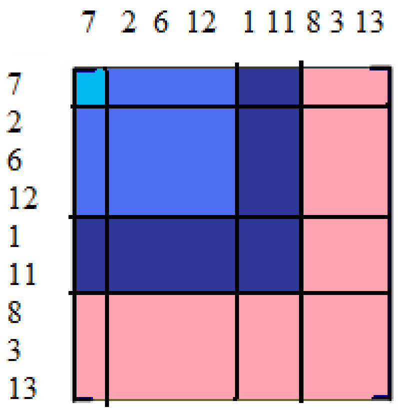
(Pagregate boundary edge)

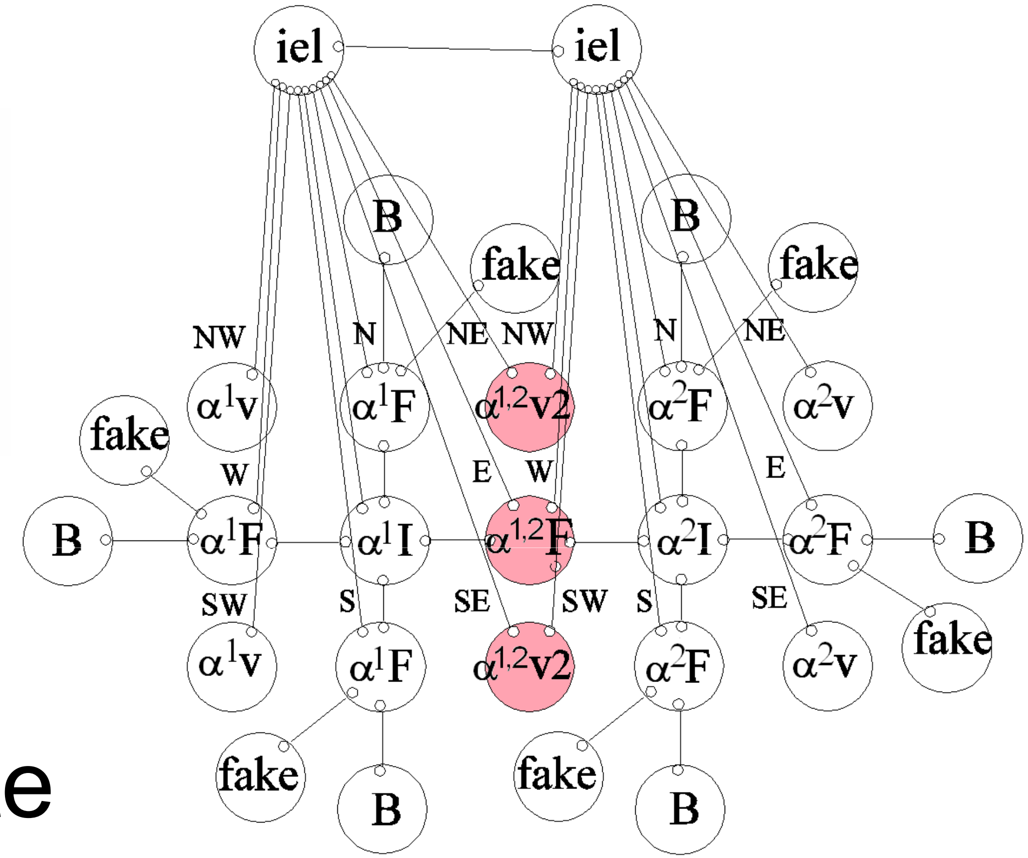


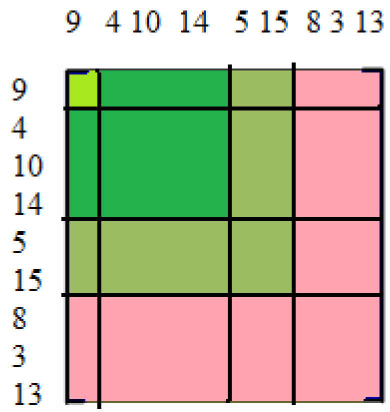


(Pagregate corner vertex)

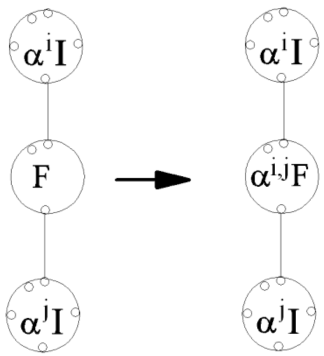
APPLICATION OF THE GRAPH GRAMMAR
GENERATION OF THE FRONTAL MATRICES (3/3)



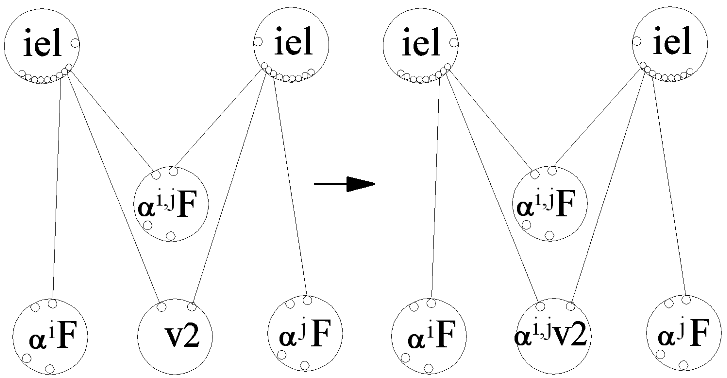




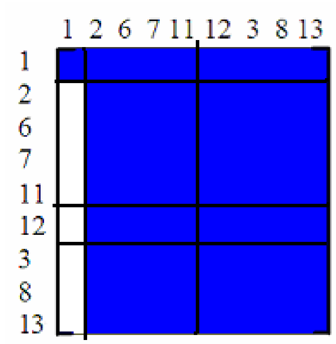
(Pagregate edge)

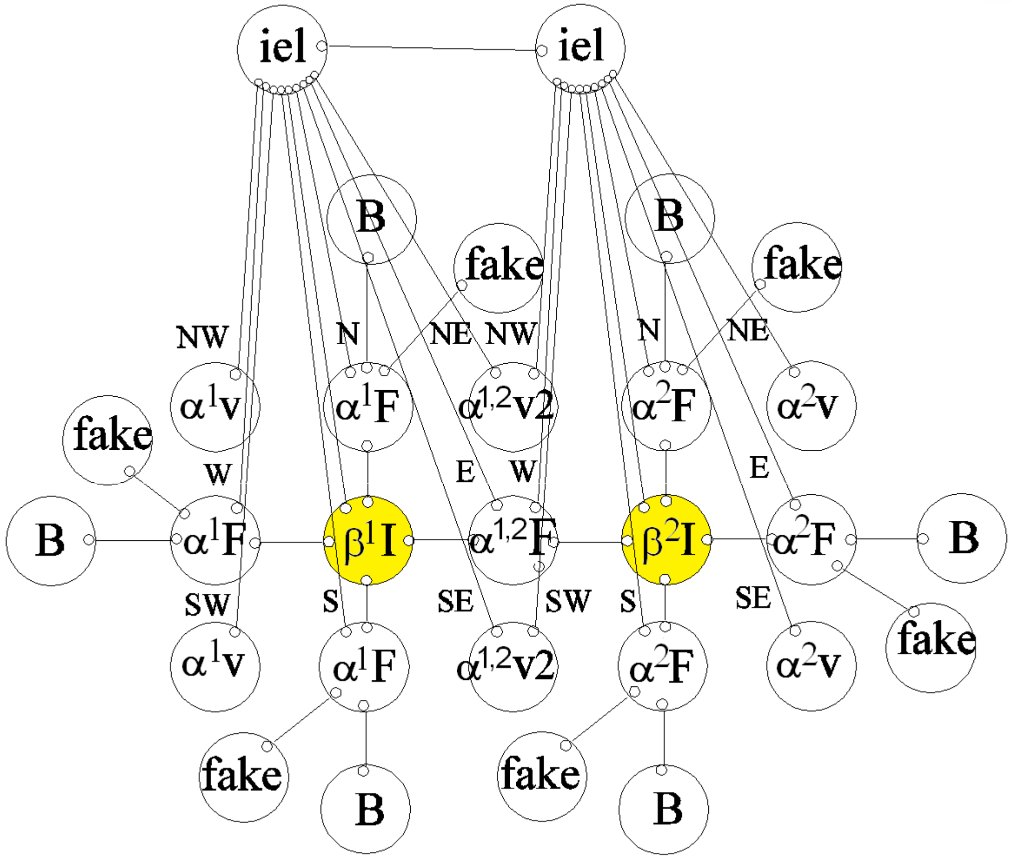


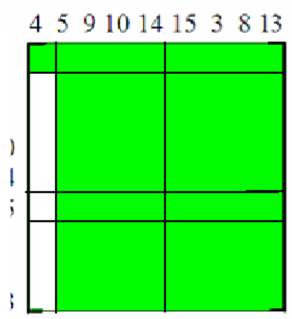
(Pagregate shared vertex)



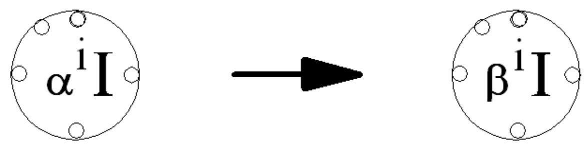
APPLICATION OF THE GRAPH GRAMMAR
LU FACTORIZATION (1/3)



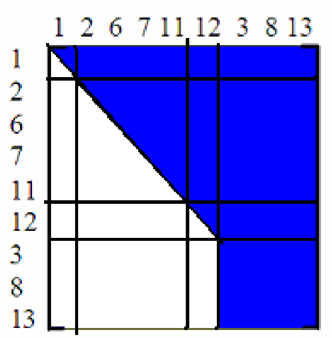


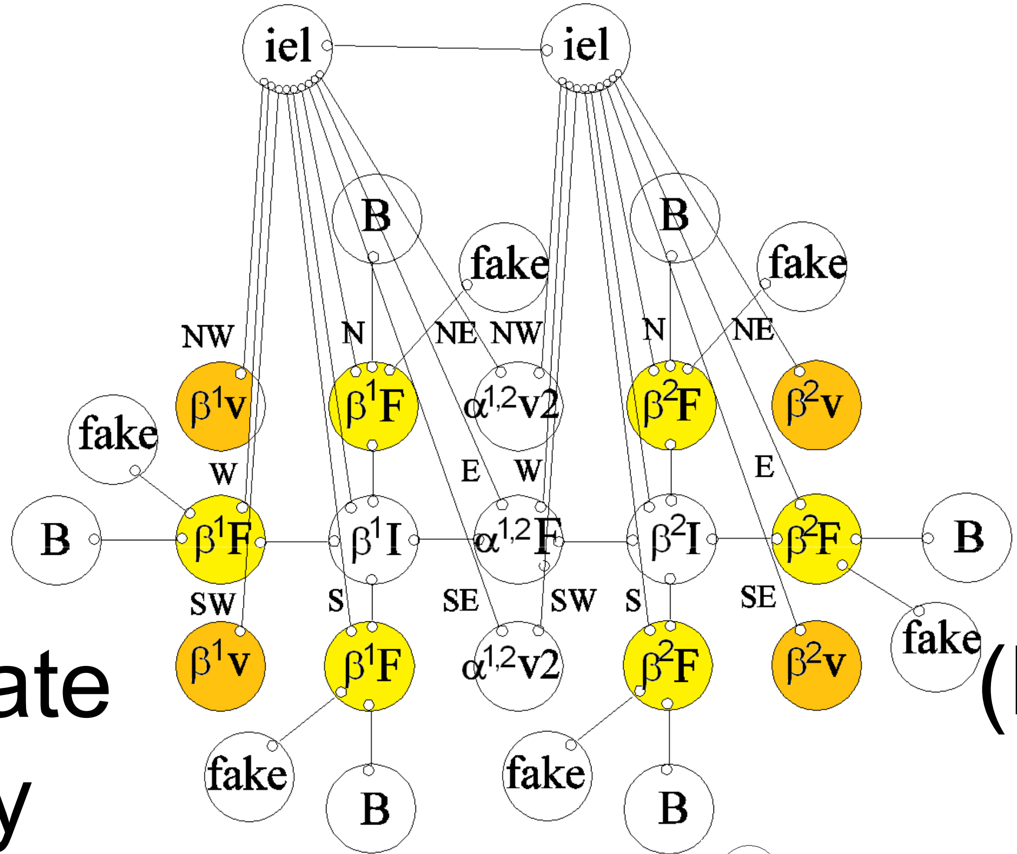


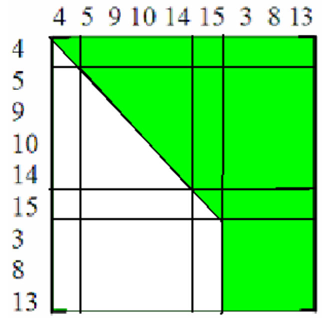
(Peliminate interior)



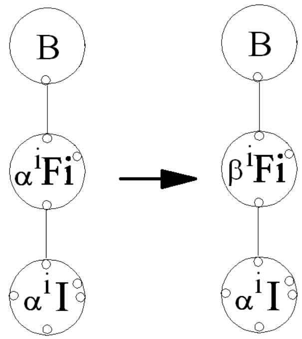
APPLICATION OF THE GRAPH GRAMMAR LU FACTORIZATION (2/3)



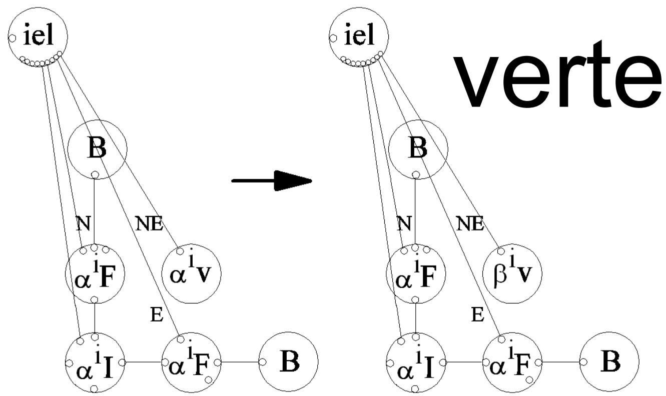




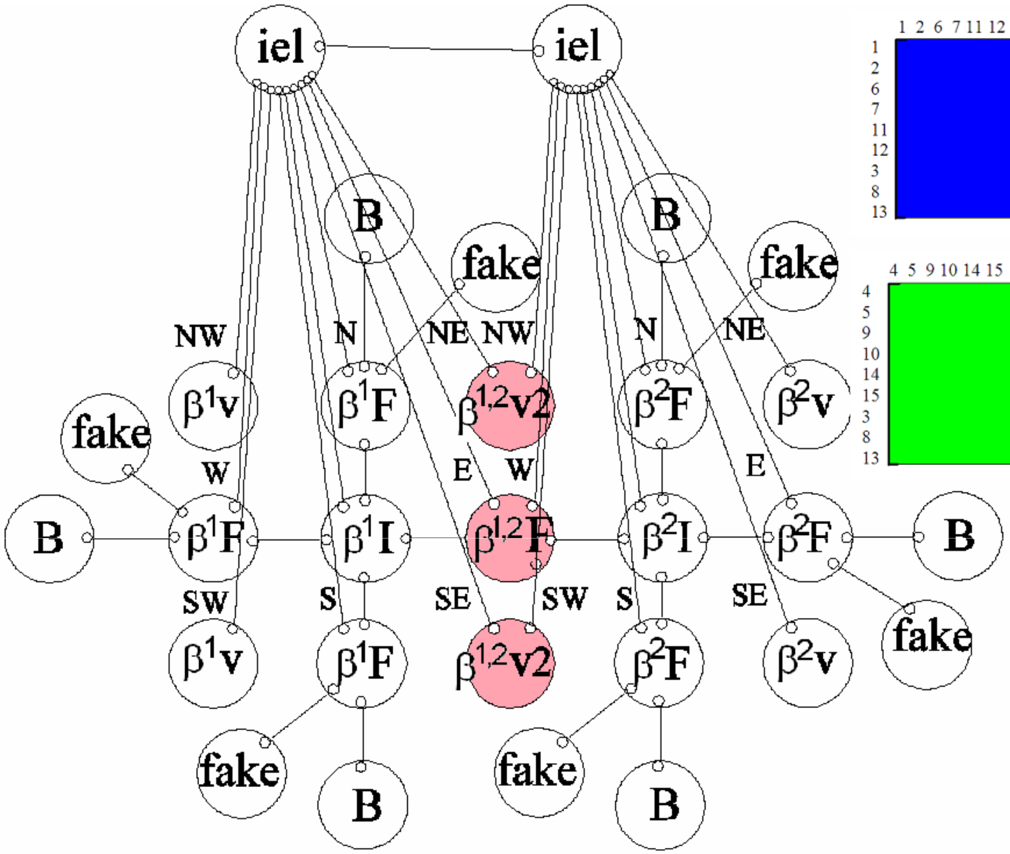
(Peliminate boundary edge)

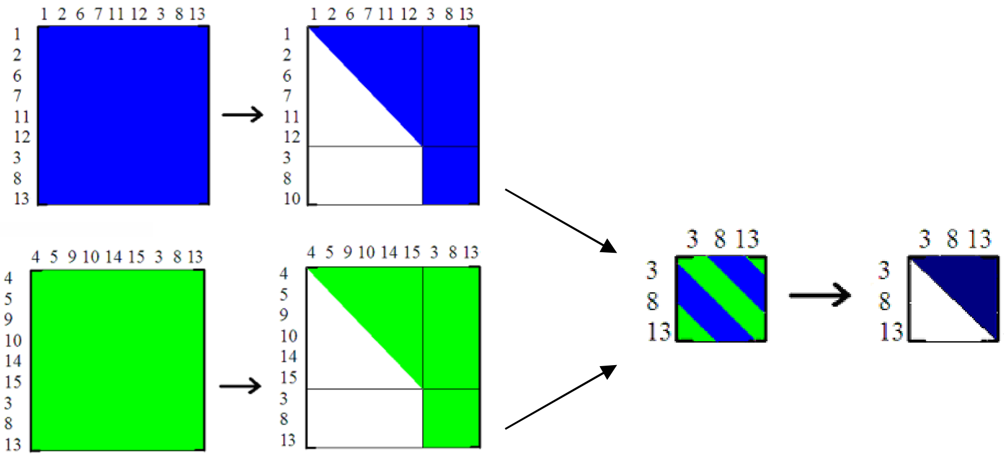


(Peliminate corner vertex)

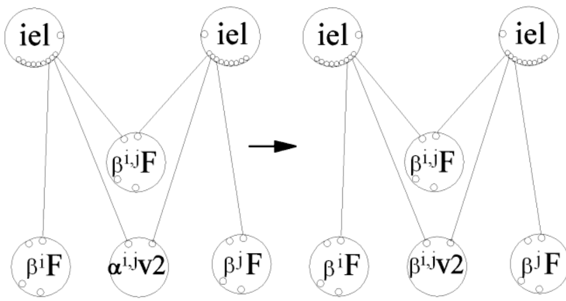


APPLICATION OF THE GRAPH GRAMMAR LU FACTORIZATION (3/3)

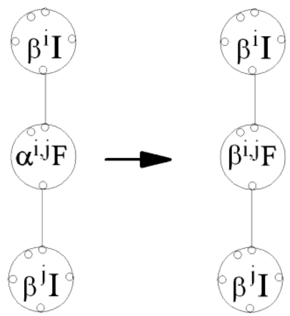




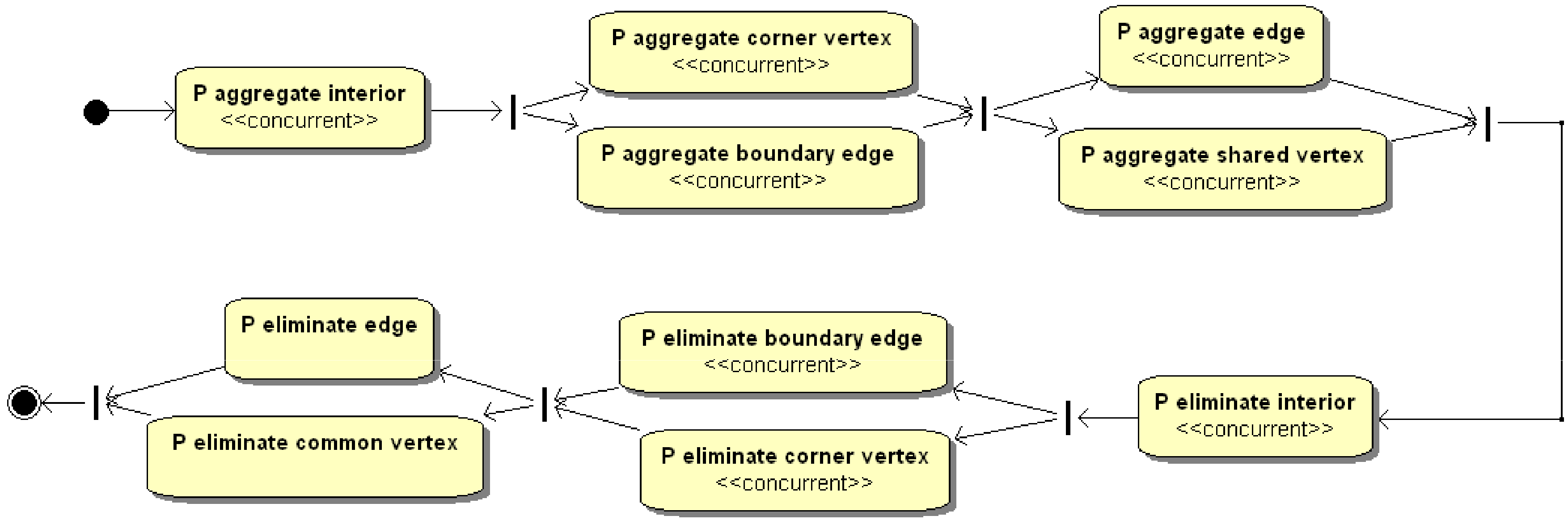
(Peliminate common vertex)



(Peliminate edge)



MODEL OF CONCURRENCY
ON THE LEVEL OF ATOMIC TASKS
(GRAPH GRAMMAR PRODUCTIONS)



The atomic tasks (graph grammar productions) can be executed concurrently on sepparate parts of the graph representing the computational mesh

(Paggregate interior)^{2}-(Paggregate boundary edge)^{6}-(Paggregate edge)
(Paggregate corner vertex)^{4} (Paggregate shared vertex)^{2}

-(Peliminate interior)^{2}-(Peliminate boundary edge)^{6}- (Peliminate edge)
(Peliminate corner edge)^{6} (eliminate common vertex)^{2}