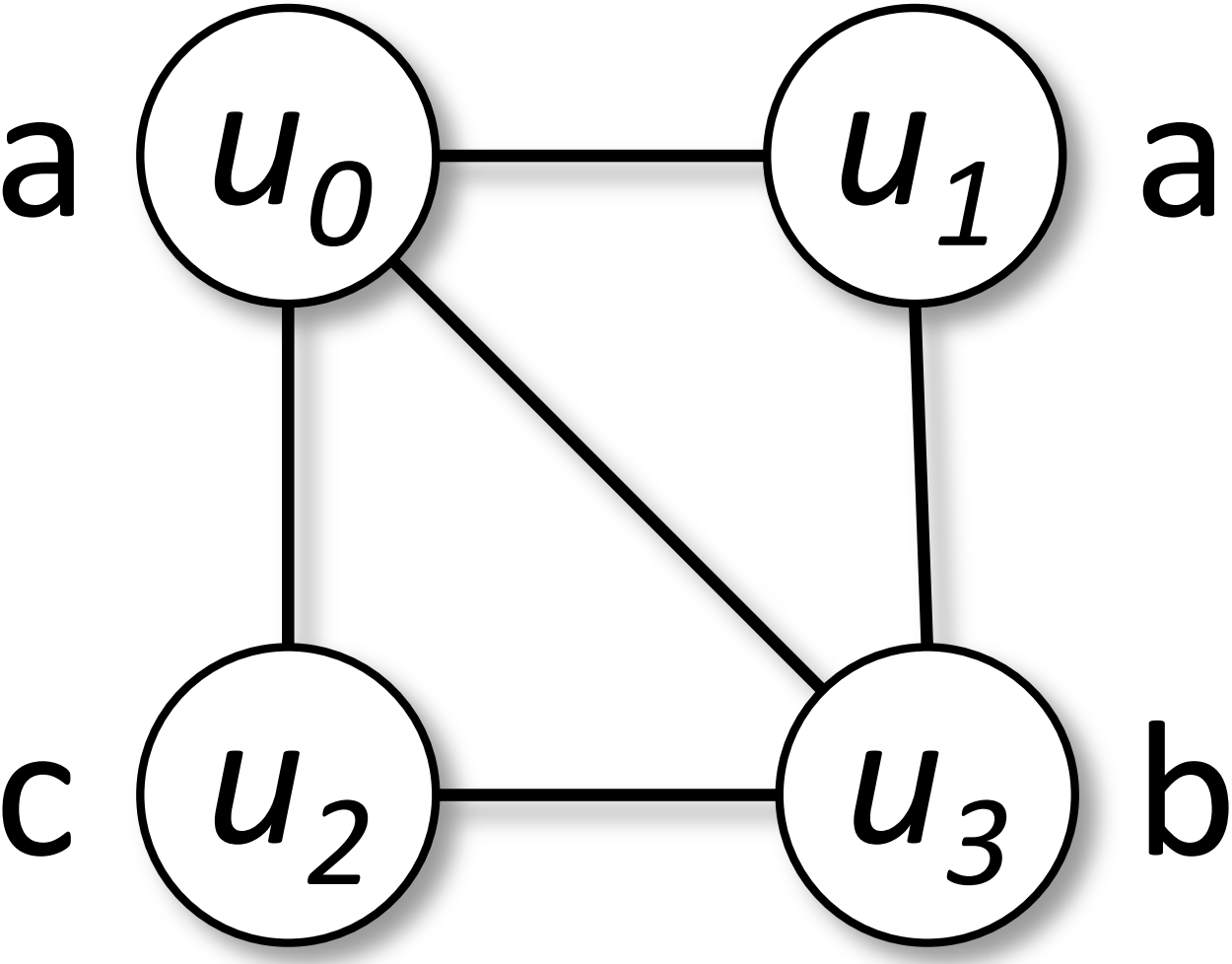
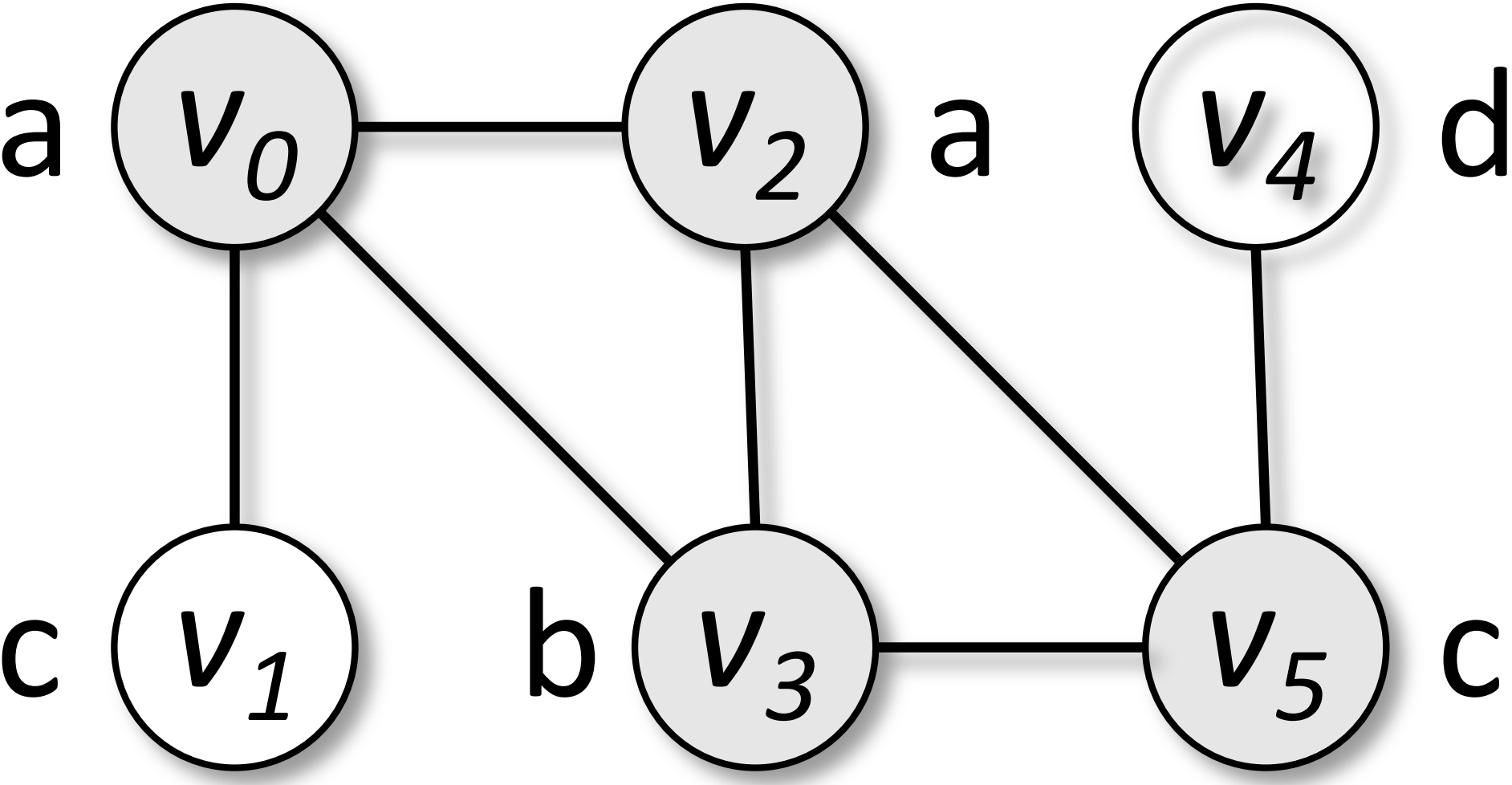
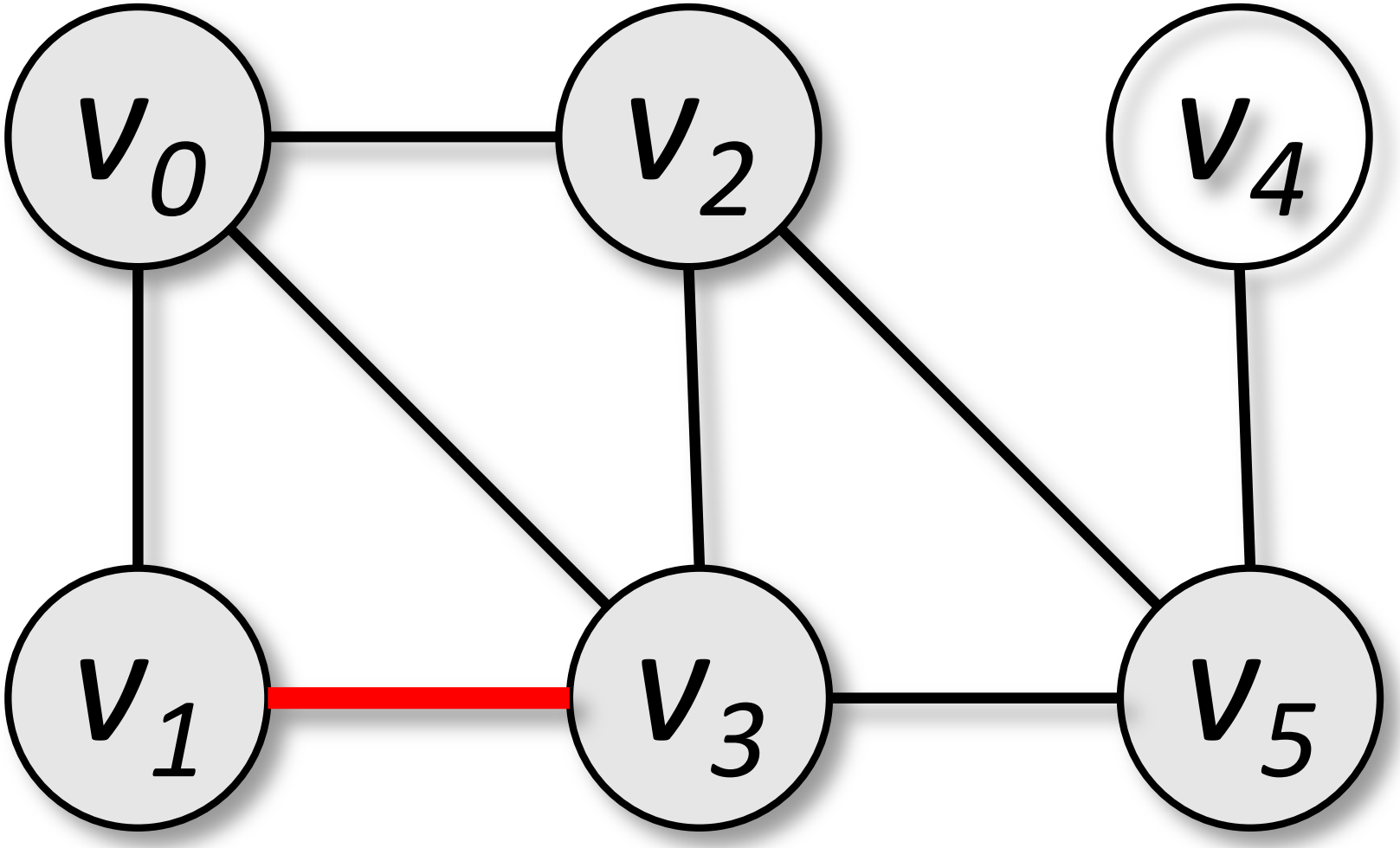
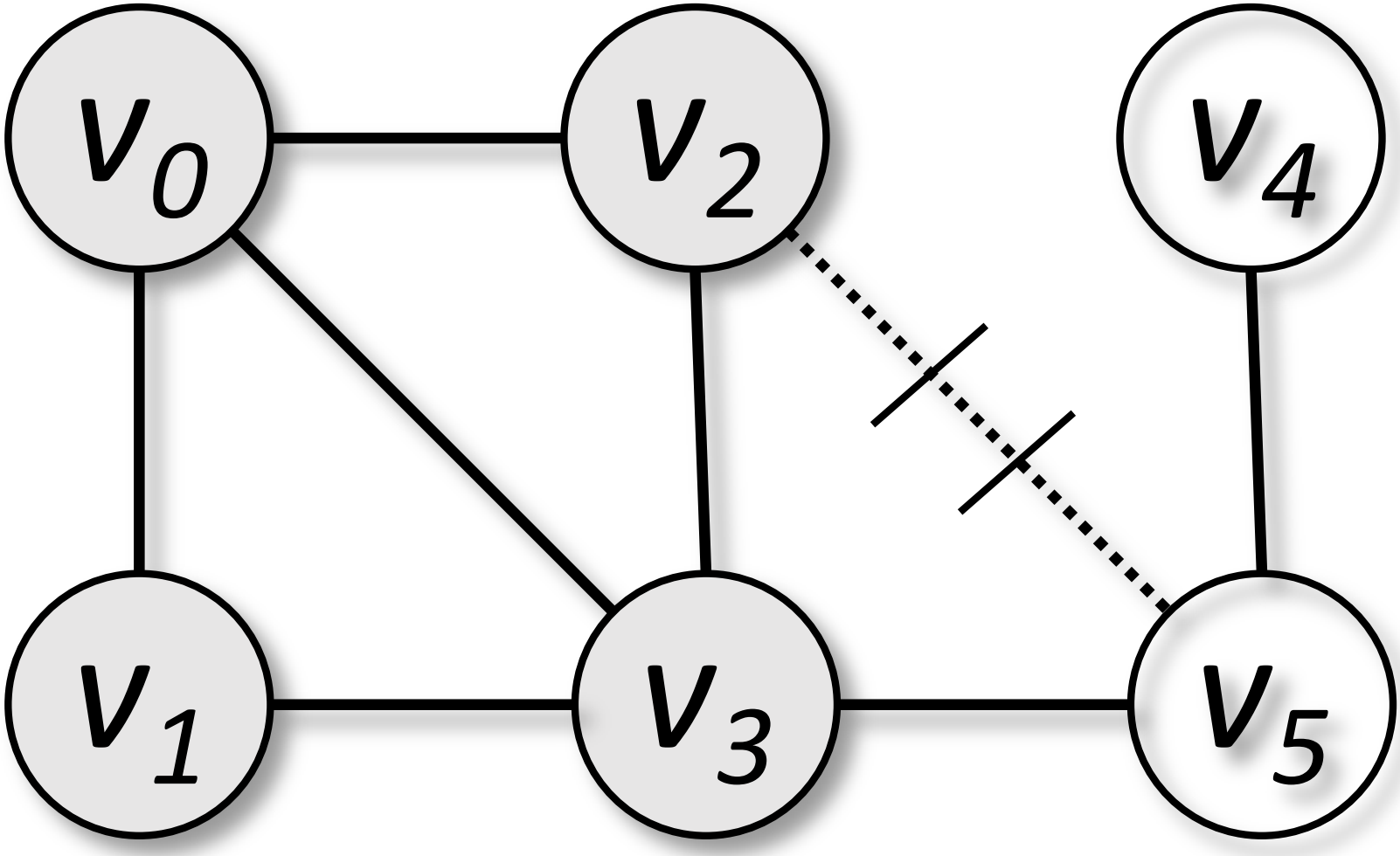

a
$u_0$
$u_1$
a
c
$u_2$
$u_3$
b

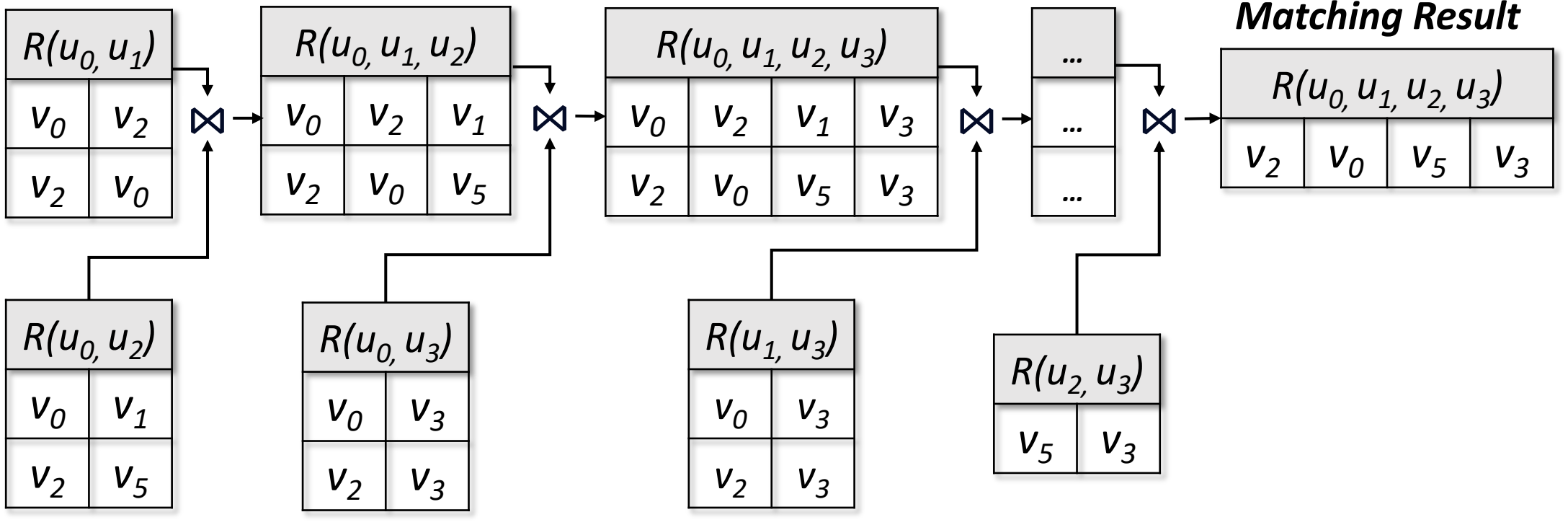

a
$v_0$
$v_2$
a
$v_4$
d
c
$v_1$
b
$v_3$
$v_5$
c

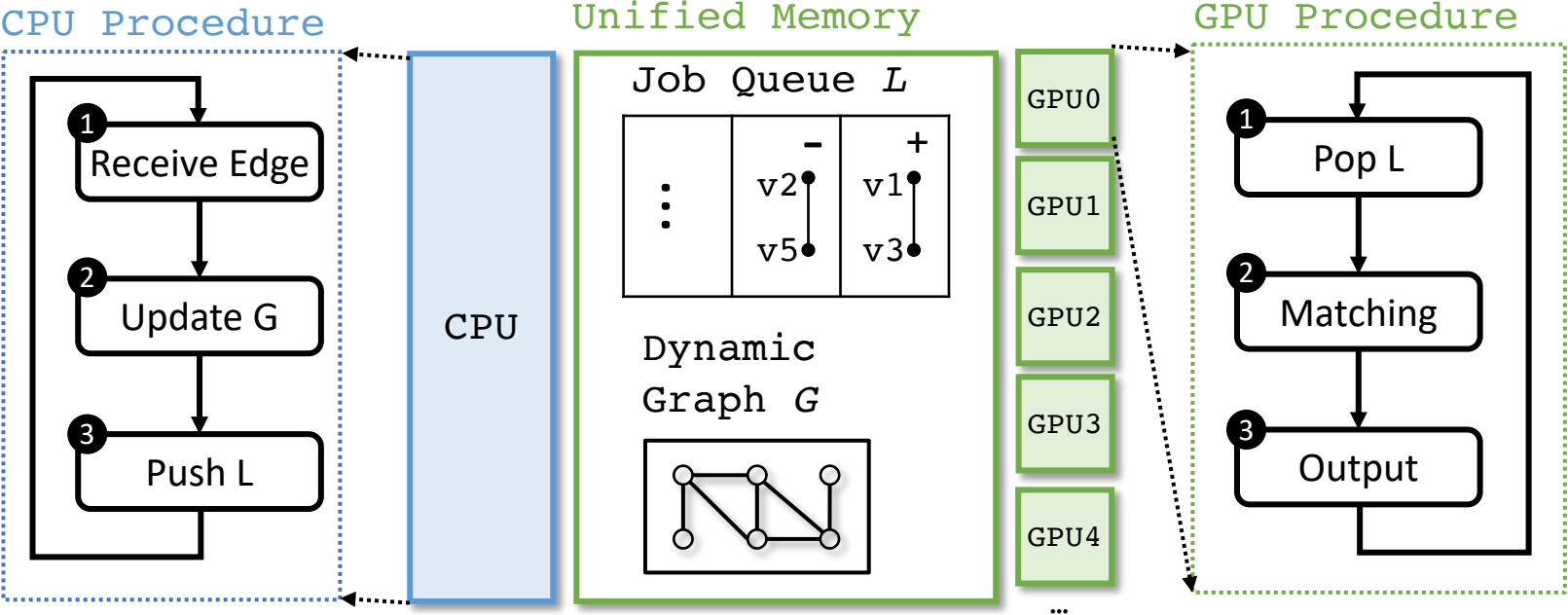


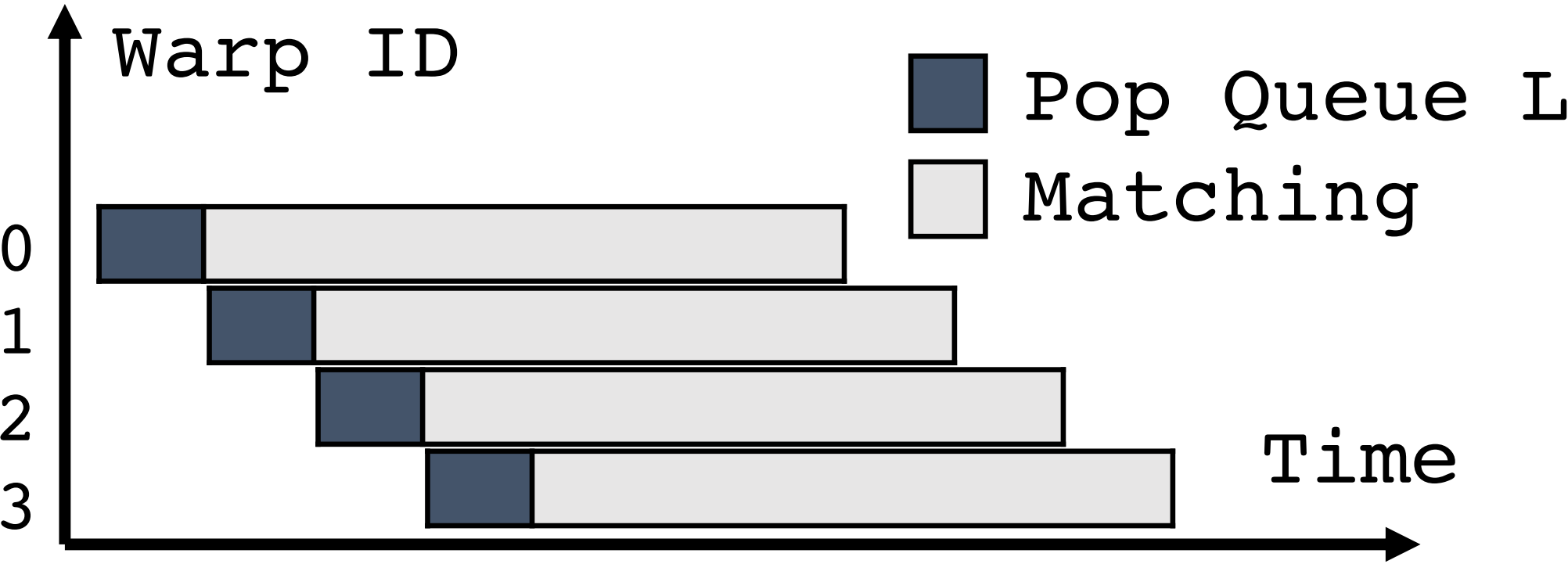

$+ e( v_1 , v_3 )$

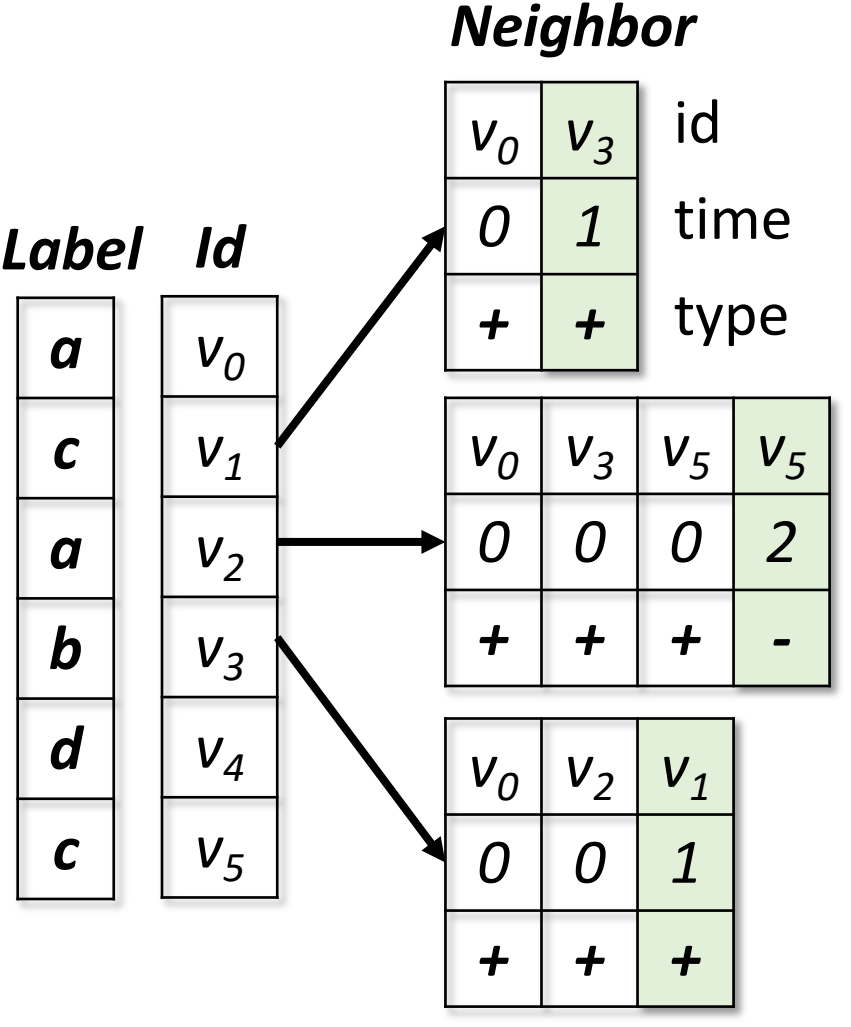


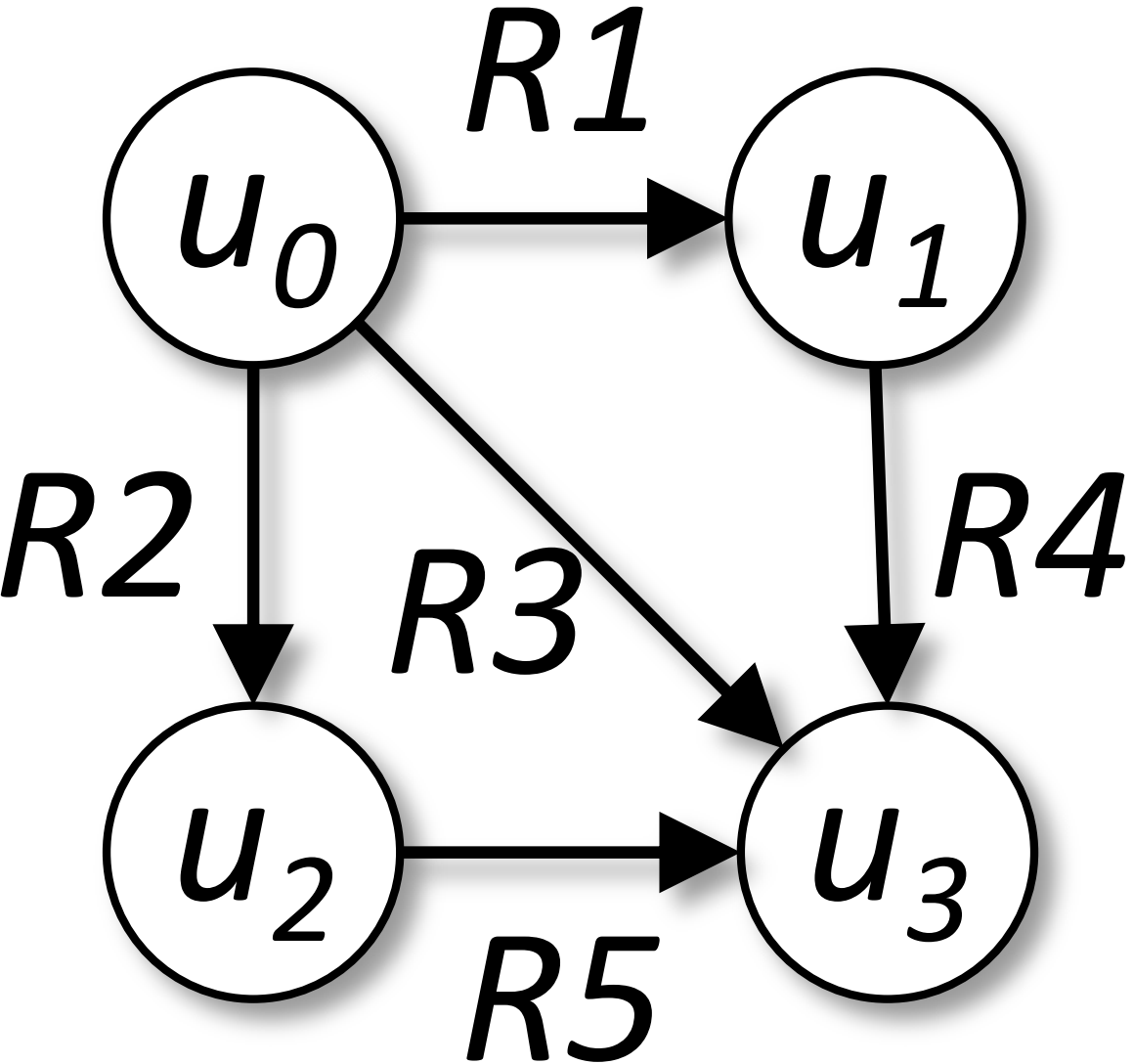

$- e( v_2 , v_5 )$

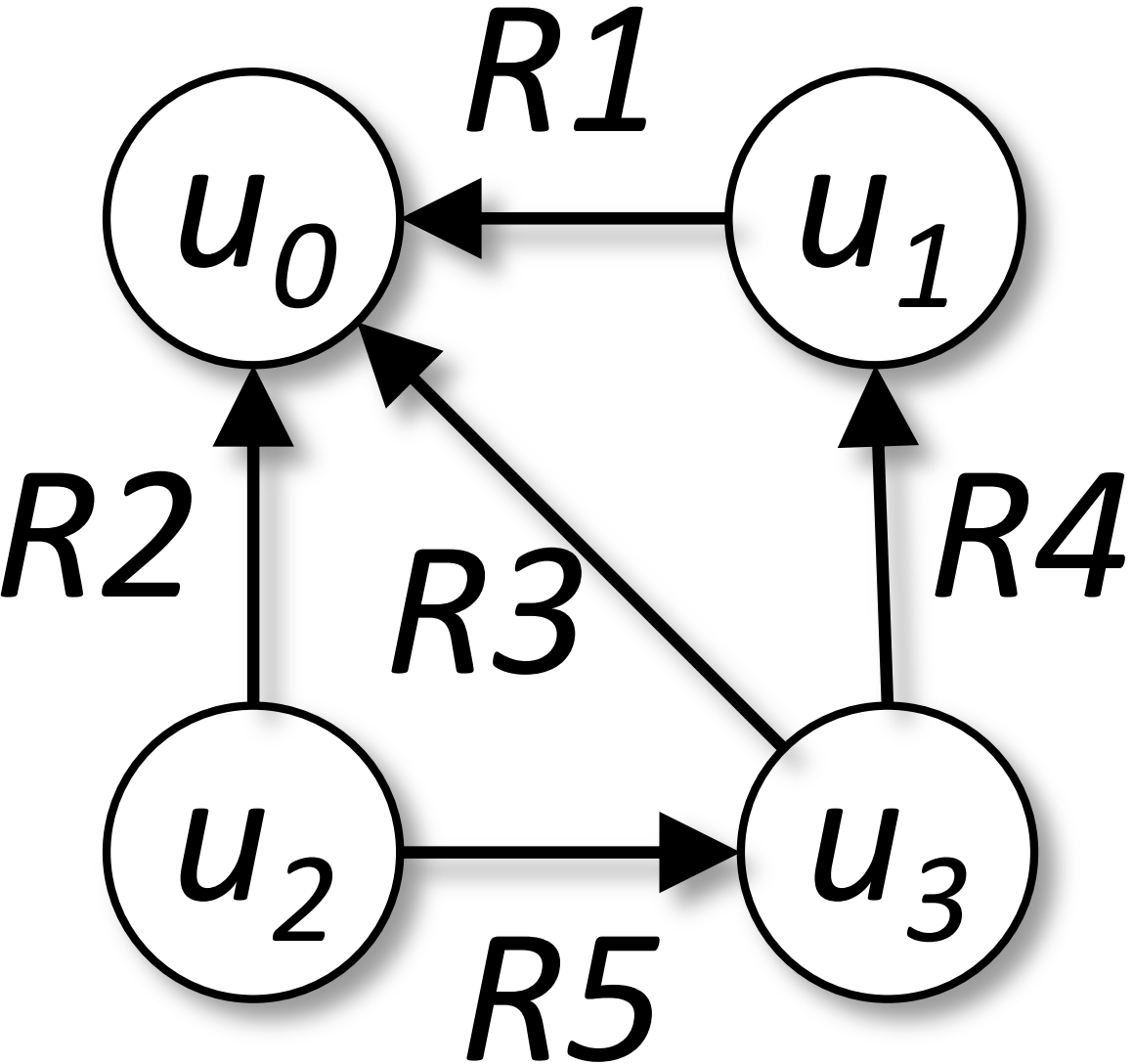

$R(u_0, u_1)$
$v_0$ $v_2$
$v_2$ $v_0$
$R(u_0, u_2)$
$v_0$ $v_1$
$v_2$ $v_5$
$R(u_0, u_1, u_2)$
$v_0$ $v_2$ $v_1$
$v_2$ $v_0$ $v_5$
$R(u_0, u_3)$
$v_0$ $v_3$
$v_2$ $v_3$
$R(u_0, u_1, u_2, u_3)$
$v_0$ $v_2$ $v_1$ $v_3$
$v_2$ $v_0$ $v_5$ $v_3$
$R(u_1, u_3)$
$v_0$ $v_3$
$v_2$ $v_3$
...
...
...
$R(u_2, u_3)$
$v_5$ $v_3$
Matching Result
$R(u_0, u_1, u_2, u_3)$
$v_2$ $v_0$ $v_5$ $v_3$


CPU Procedure
1
Receive Edge
2
Update G
3
Push L
CPU
Unified Memory
Job Queue L
-
+
v2
v1
v5
v3
Dynamic
Graph G
GPU0
GPU1
GPU2
GPU3
GPU4
...
GPU Procedure
1
Pop L
2
Matching
3
Output


Warp ID
Pop Queue L
Matching
0
1
2
3
Time


Neighbor
Label
Id
a
c
a
b
d
c
$v_0$
$v_1$
$v_2$
$v_3$
$v_4$
$v_5$
$v_0$ $v_3$
0 1
+ +
id
time
type
$v_0$ $v_3$ $v_5$ $v_5$
0 0 0 2
+ + + -
$v_0$ $v_2$ $v_1$
0 0 1
+ + +


R1
$u_0$
$u_1$
R2
R3
R4
$u_2$
$u_3$
R5


R1
$u_0$
$u_1$
R2
R3
R4
$u_2$
$u_3$
R5

```
1) void Matching(G, Label, x2, x3){
2)  assert(x2.time = x3.time);
3)  assert(x2.type = x3.type);
4)  uint64_t time  = x2.time;
5)  uint32_t type  = x2.type;
6)  bool     valid = false;
7)  for(x0 ∈ G[x2.id] && Label[x0.id]=='a'){
8)   if(x0.id == x3.id)  continue;
9)   if(x0.time >= time) break;
10)
11)   for(x0' ∈ G[x3.id] && Label[x0'.id]=='a'){
12)    if(x0'.id != x0.id)  continue;
13)    if(x0'.time >= time) break;
14)
15)    for(x1 ∈ G[x3.id] && Label[x1.id]=='a'){
16)     if(x1.id == x2.id) continue;
17)     if(x1.id == x0.id) continue;
18)     if(x1.time >= time) break;
19)
20)     for(x0'' ∈ G[x1.id] && Label[x0''.id]=='a'){
21)      if(x0''.id != x0.id)  continue;
22)      if(x0''.time >= time) break;
23)
24)      if(x0.type  =='+' &&
25)         x0'.type =='+' &&
26)         x1.type  =='+' &&
27)         x0''.type=='+'){ valid = true;}
28)
29)      Output(x0, x1, x2, x3, type, valid);
30)}}}}}
```