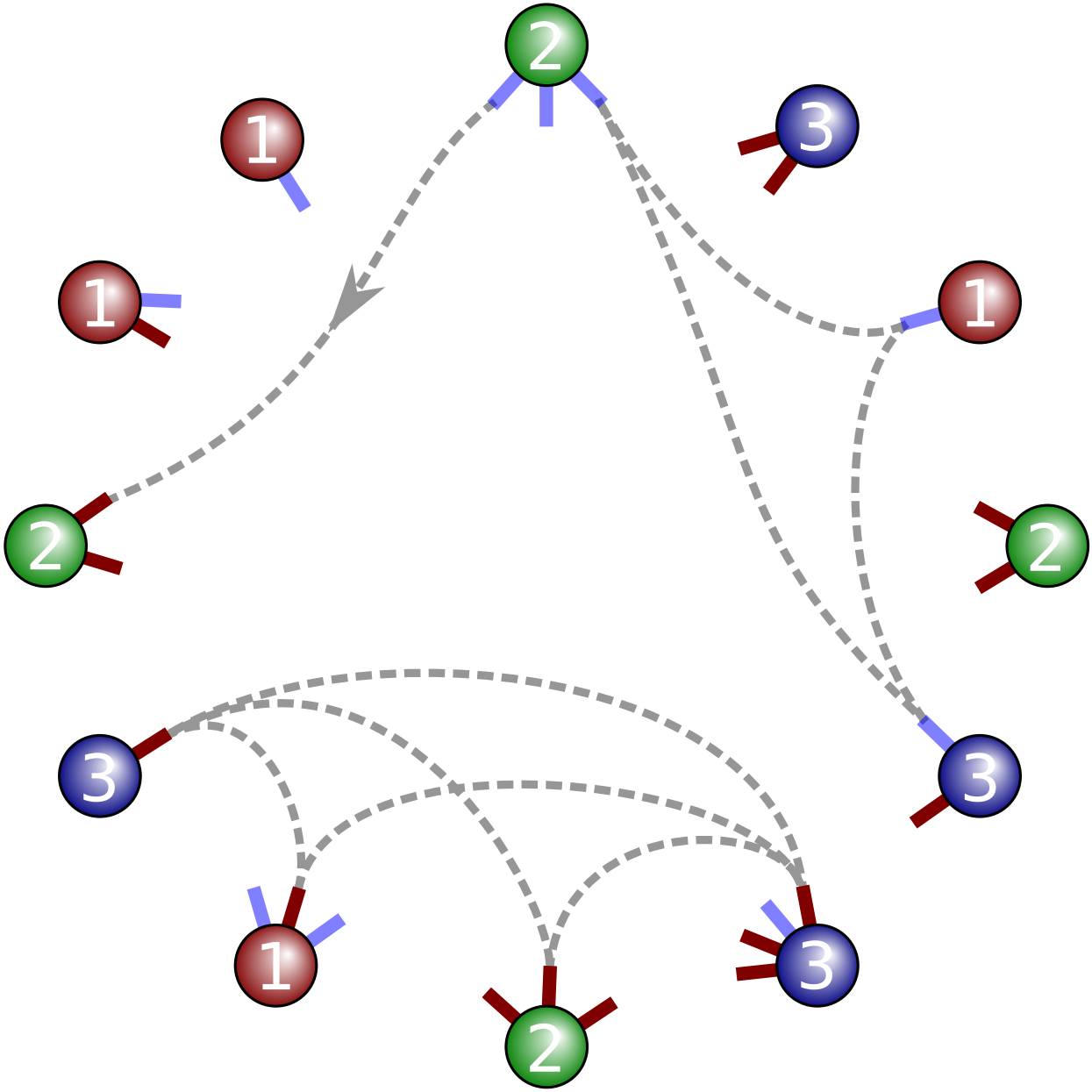
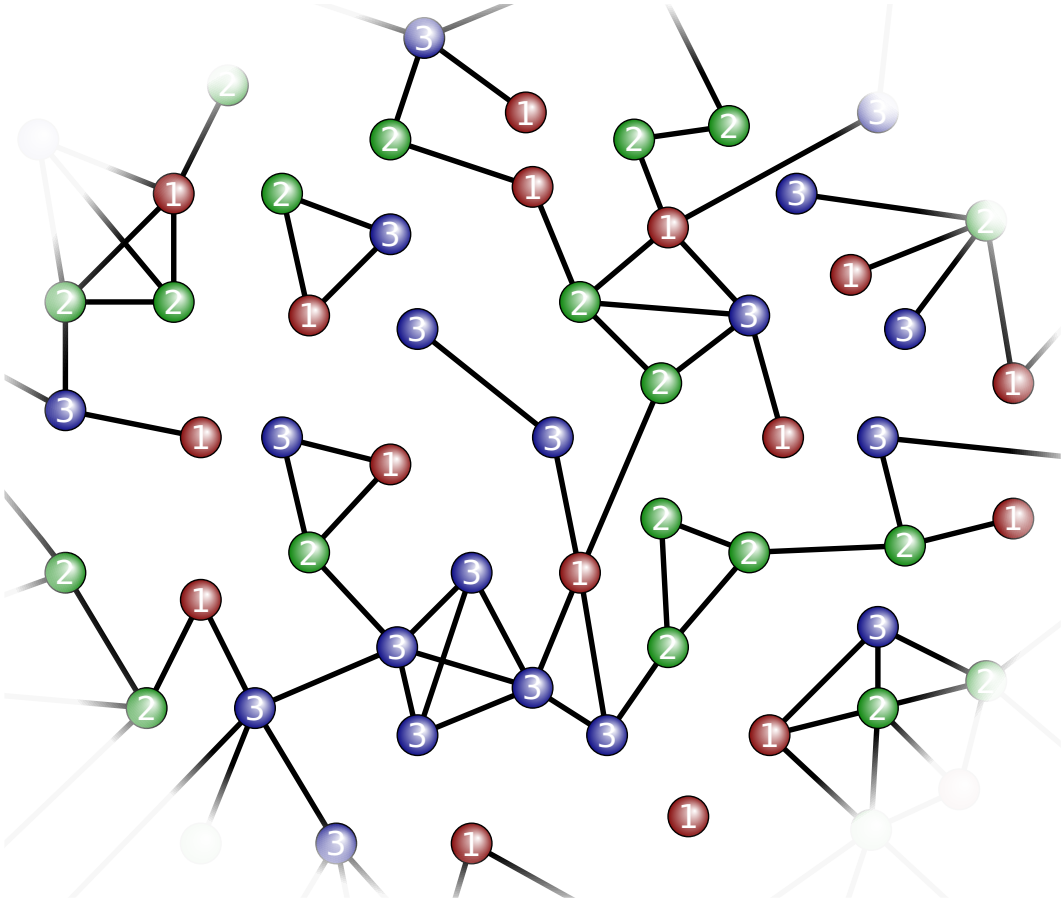
24


2
1
3
1
1
2
2
3
3
1
2
3



General *stub matching* scheme

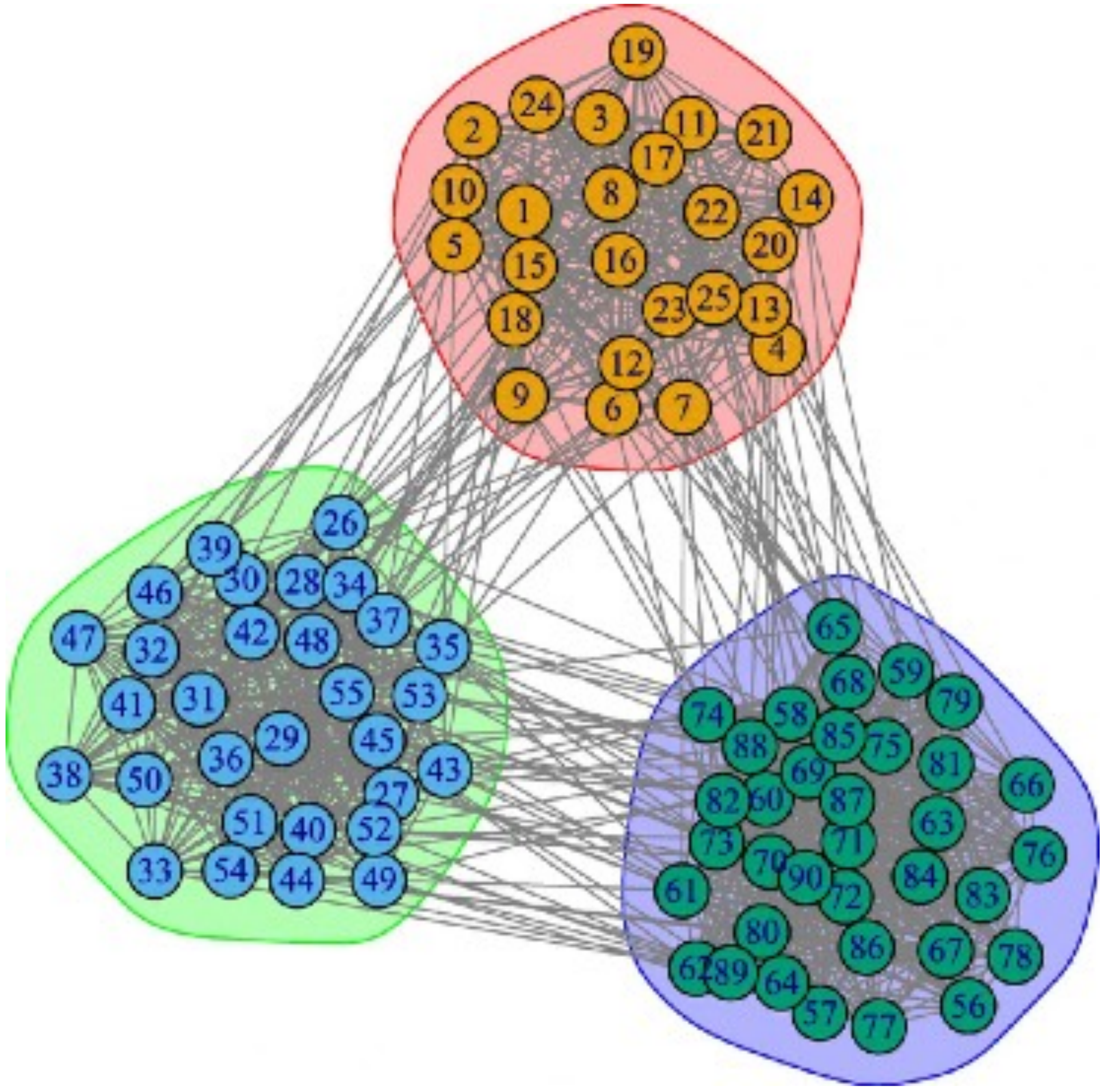
Many network models can be seen as a *stub matching* scheme with

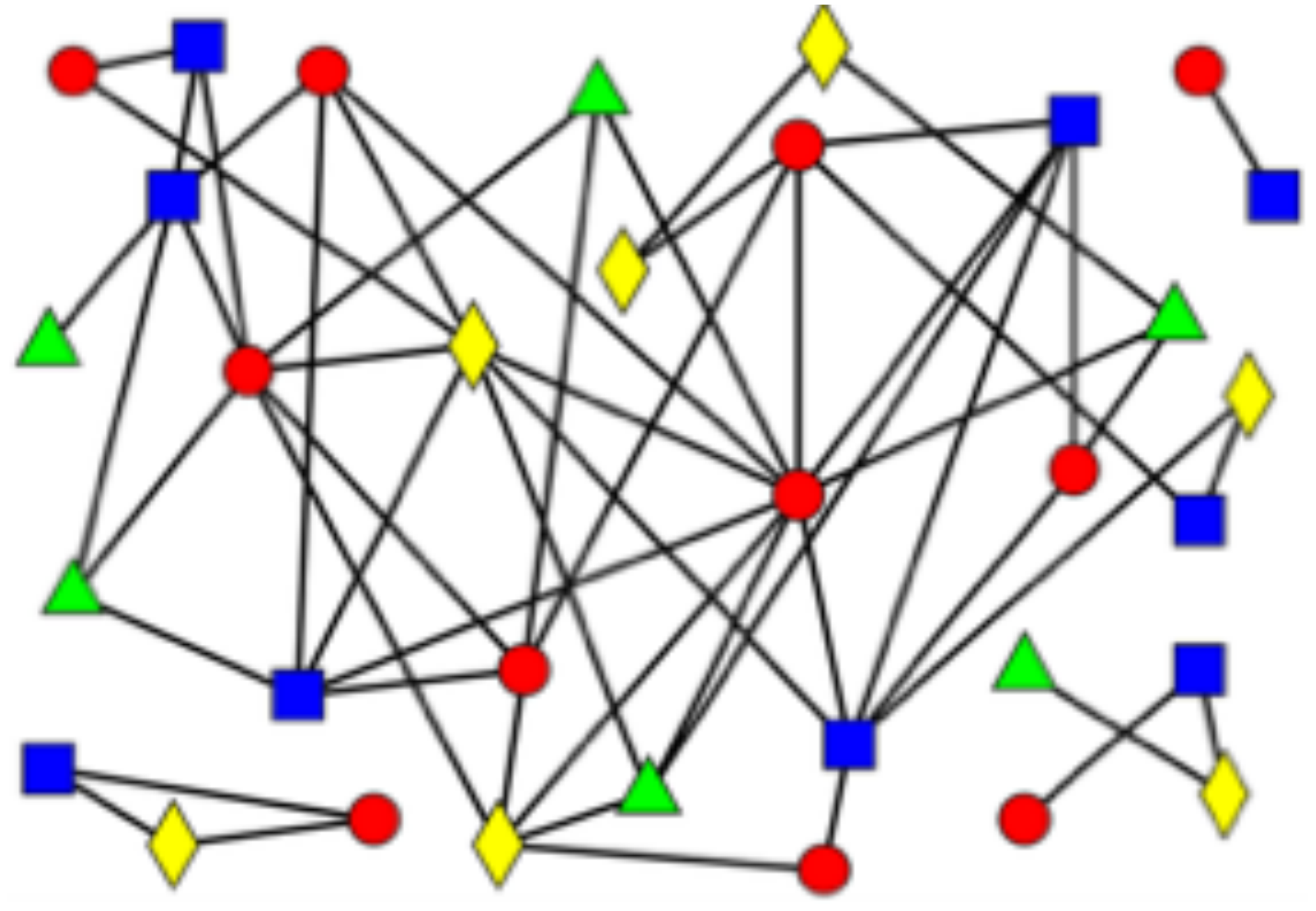
- node types
- stub types
- rules governing how stubs are matched

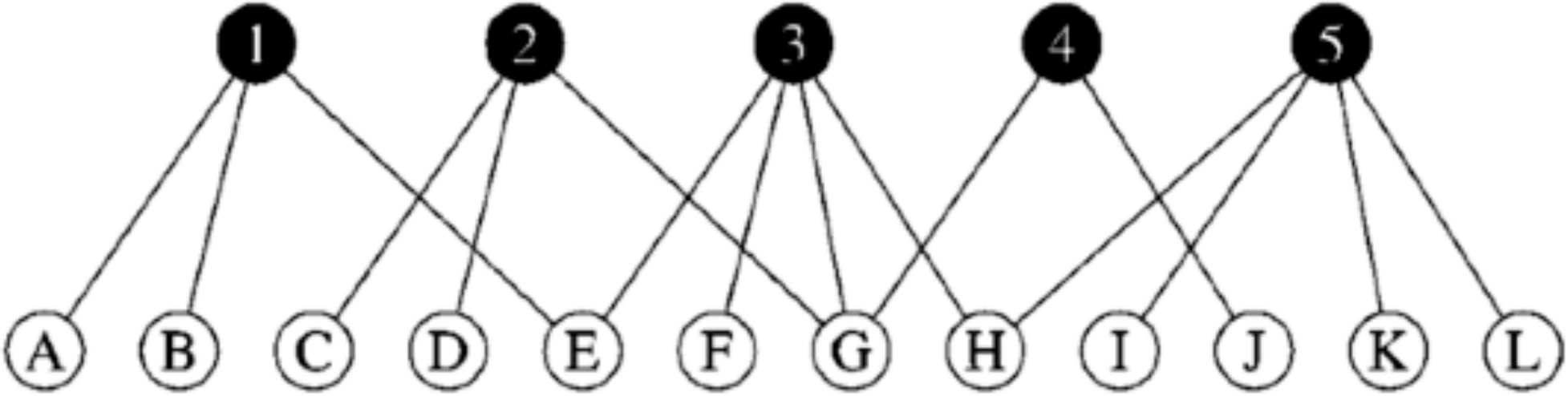
This perspective facilitates the mathematical description of the dynamical processes on networks

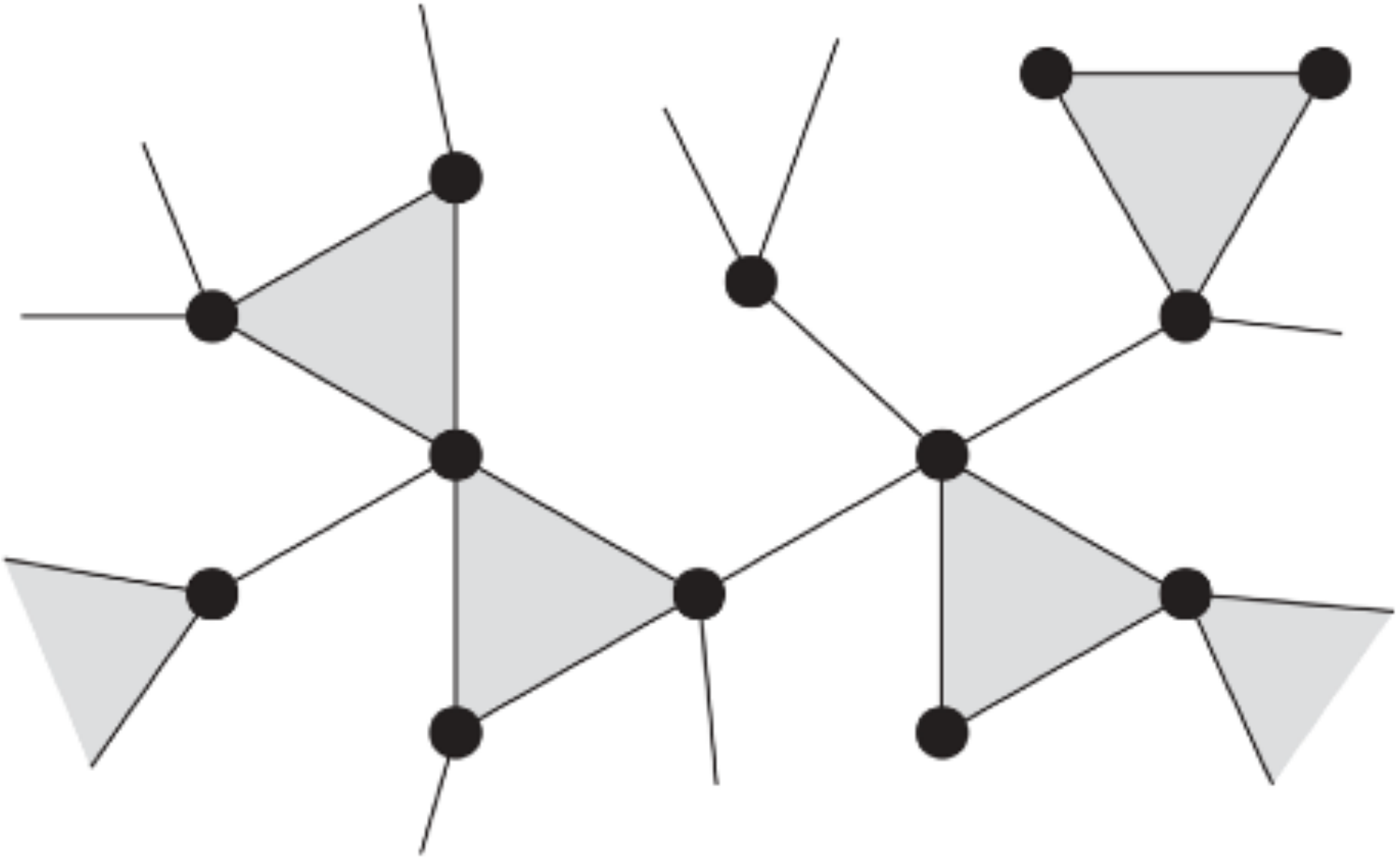
- probability generating functions (ex.: percolation, robustness)
- ordinary differential equations (ex.: epidemic spreading, opinion dynamics)

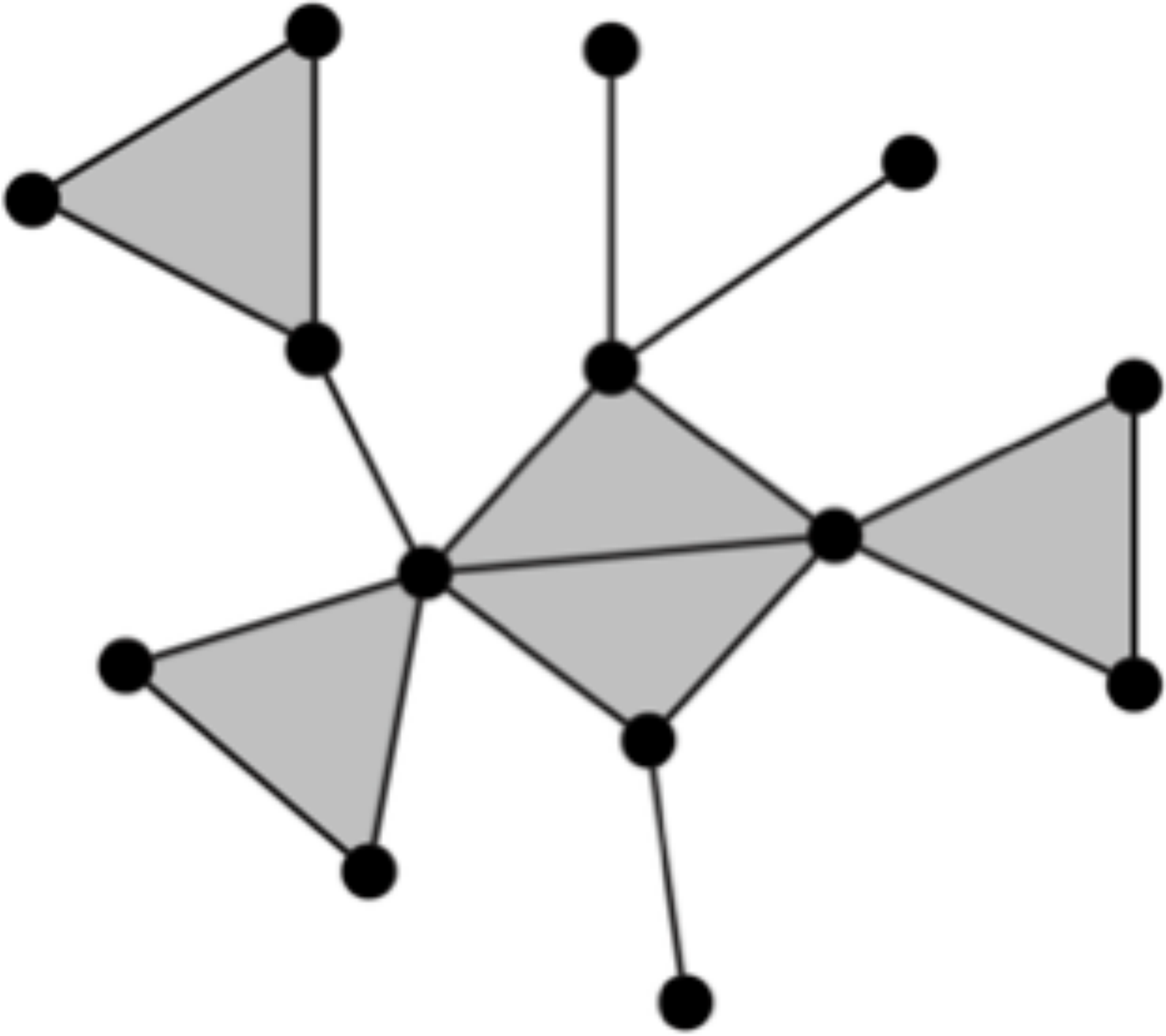
Most of these approaches use stub types to enforce *local* connection patterns.


19
24
3
11
21
2
17
10
8
22
14
1
5
20
15
16
25
23
13
18
4
12
9
6
7
26
39
30
28
34
46
37
47
42
48
35
32
55
53
41
31
29
45
36
43
38
50
27
51
40
52
33
54
44
49
65
68
59
79
74
58
85
75
88
81
66
69
82
60
87
63
73
71
76
70
90
84
61
72
83
80
86
67
78
62
89
64
57
56
77




1
2
3
4
5
A
B
C
D
E
F
G
H
I
J
K
L





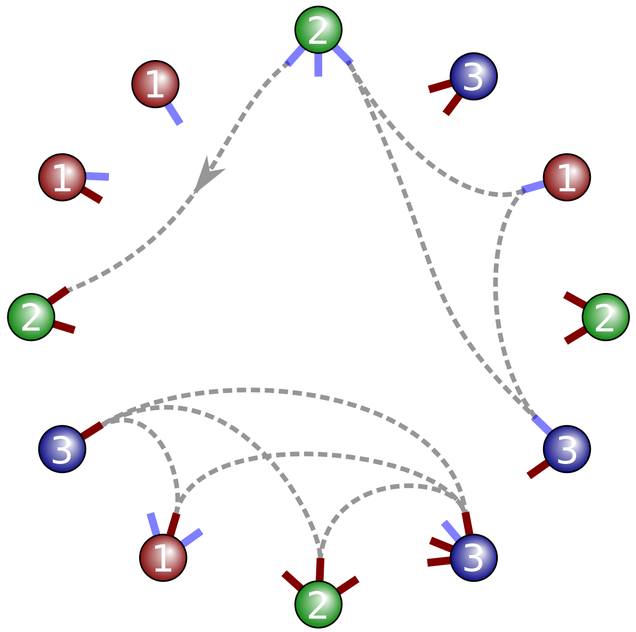
General *stub matching* scheme

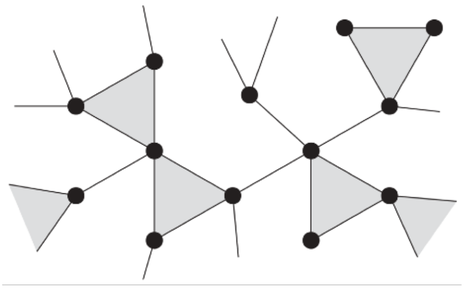
Many network models can be seen as a *stub matching* scheme with

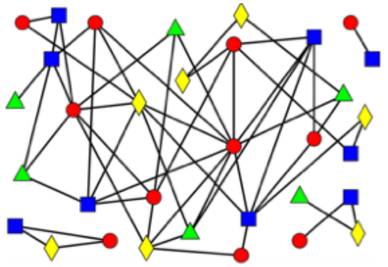
- node types
- stub types
- rules governing how stubs are matched

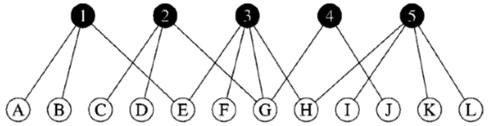
This perspective facilitates the mathematical description of the dynamical processes on networks

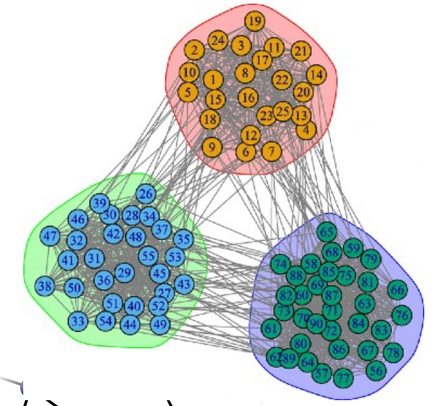
- probability generating functions (ex.: percolation, robustness)
- ordinary differential equations (ex.: epidemic spreading, opinion dynamics)

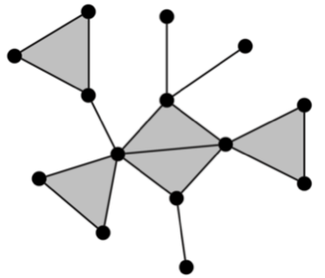


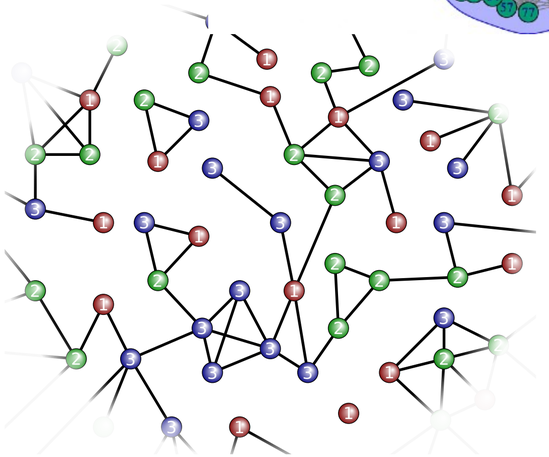












Most of these approaches use stub types to enforce *local* connection patterns.

Mesoscopic level: The k-core/onion decomposition

Onion decomposition: k-core decomposition with additional information about the positions of nodes within every k-shell (layers).

Information about layers is obtained from the k-core decomposition with minimal additional computational cost.