A First Course in Network Theory
Reminders and Basics about Graphs and Networks

Luce le Gorrec, Philip Knight, Francesca Arrigo

University of Strathclyde, Glasgow

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

[1] https://www.naxys.be/

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- **Social networks**,
- Biological networks,
- Transports,
- Language,
- Technological,
- Etc.

[1] `https://www.naxys.be/`

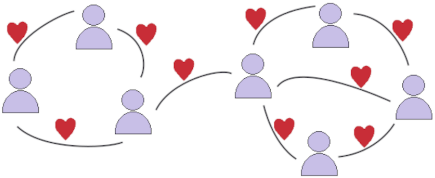
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- **Social networks**,
- Biological networks,
- Transports,
- Language,
- Technological,
- Etc.



[1] https://www.naxys.be/

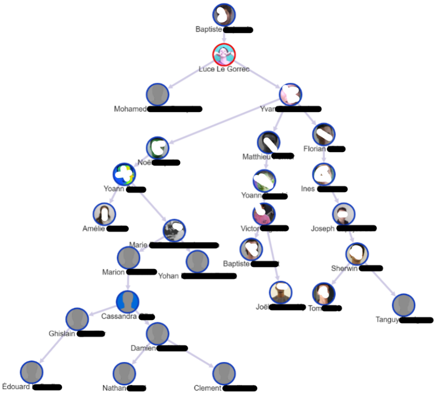
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- **Social networks**,
- Biological networks,
- Transports,
- Language,
- Technological,
- Etc.



[1] https://www.naxys.be/

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- **Biological networks**,
- Transports,
- Language,
- Technological,
- Etc.

[1] `https://www.naxys.be/`

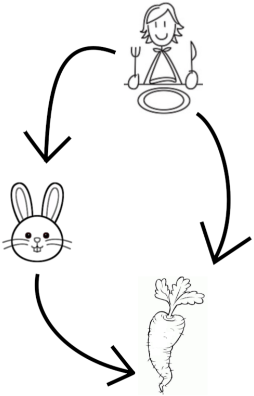
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- **Biological networks**,
- Transports,
- Language,
- Technological,
- Etc.



[1] https://www.naxys.be/

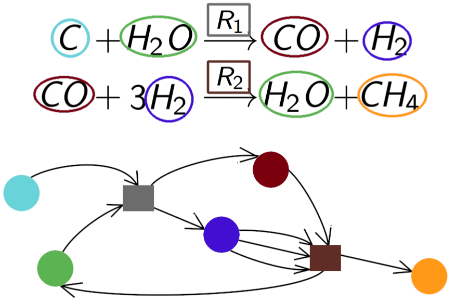
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- **Biological networks**,
- Transports,
- Language,
- Technological,
- Etc.



[1] https://www.naxys.be/

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- Biological networks,
- **Transports**,
- Language,
- Technological,
- Etc.

[1] https://www.naxys.be/

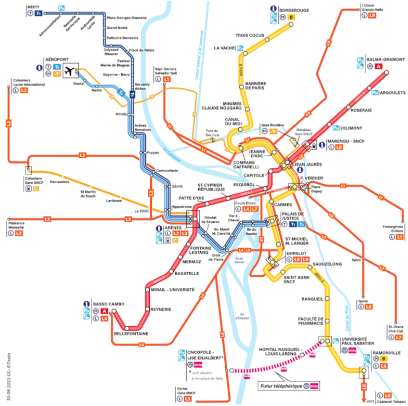
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- Biological networks,
- **Transports**,
- Language,
- Technological,
- Etc.



[1] https://www.naxys.be/

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- Biological networks,
- Transports,
- **Language**,
- Technological,
- Etc.

[1] https://www.naxys.be/

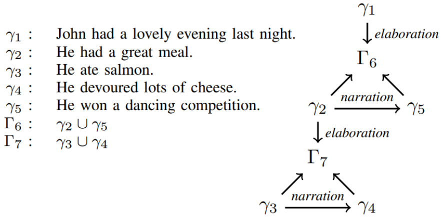
Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- Biological networks,
- Transports,
- **Language**,
- Technological,
- Etc.



[1] https://www.naxys.be/

Generalities and Interest

Complex Networks are Graphs used to represent **complex systems**.

Complex Systems *Systems composed of a* **large number of simple elements in interaction** *and exhibiting emerging phenomena*[1].

Examples of complex networks

- Social networks,
- Biological networks,
- Transports,
- Language,
- **Technological**,
- Etc.

[1] `https://www.naxys.be/`

Basics – Definitions

Definition A **Graph or Network** $G = (V, E, \omega)$ is a tuple of:

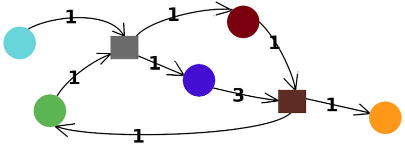
- a set V, called the vertex set (or node set). Elements of V are called **vertices or nodes**.
- a set $E \subset V \times V$, called the edge set (or link set). Elements in E are called **edges or links**.
- an application $\omega : E \to \Omega$, called the **edge weight function**.

Basics – Definitions

Definition A **Graph or Network** $G = (V, E, \omega)$ is a tuple of:

- a set V, called the vertex set (or node set). Elements of V are called **vertices or nodes**.
- a set $E \subset V \times V$, called the edge set (or link set). Elements in E are called **edges or links**.
- an application $\omega : E \to \Omega$, called the **edge weight function**.

$$C + H_2O \xrightarrow{R_1} CO + H_2$$
$$CO + 3H_2 \xrightarrow{R_2} H_2O + CH_4$$

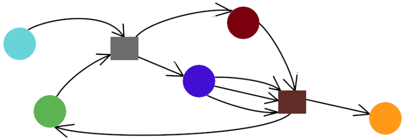


Basics – Definitions

Definition A **Graph or Network** $G = (V, E, \omega)$ is a tuple of:

- a set V, called the vertex set (or node set). Elements of V are called **vertices or nodes**.
- a set $E \subset V \times V$, called the edge set (or link set). Elements in E are called **edges or links**.
- an application $\omega : E \to \Omega$, called the **edge weight function**.

$$C + H_2O \xrightarrow{R_1} CO + H_2$$
$$CO + 3H_2 \xrightarrow{R_2} H_2O + CH_4$$



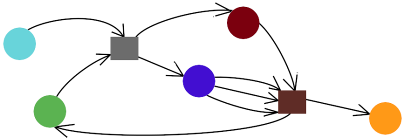
Remark If $\Omega \subset \mathbb{N}$, possibility to define multi-graph instead.

Basics – Definitions

Definition A **Graph or Network** $G = (V, E, \omega)$ is a tuple of:

- a set V, called the vertex set (or node set). Elements of V are called **vertices or nodes**.
- a set $E \subset V \times V$, called the edge set (or link set). Elements in E are called **edges or links**.
- an application $\omega : E \to \Omega$, called the **edge weight function**.

$$C + H_2O \xrightarrow{R_1} CO + H_2$$
$$CO + 3H_2 \xrightarrow{R_2} H_2O + CH_4$$



Remark If $\Omega \subset \mathbb{N}$, possibility to define multi-graph instead.

Remark With this definition, information about nodes is lost.

Basics – Definitions

Definitions Given a graph $G = (V, E, \omega)$, and an edge $e \in E$:

- If $e(1) = e(2)$, then e is called a **(self-)loop**.

NB If $\forall e \in E$, e is **not** a self-loop, then G is called **anti-reflexive**.

- When $\omega : E \to \Omega = \{\omega_0\}$, then G is called an **unweighted graph** and ω may be omitted: $G = (V, E)$.

- If $\exists f \in E : \begin{cases} f(1) = e(2) \\ f(2) = e(1) \\ \omega(f) = \omega(e) \end{cases}$, then e is called a **bi-directed edge**.

NB If all edges are bi-directed, then G is said to be **undirected or non-directed**. One can choose to count each edge only once.

Definition A graph is called **simple** if it is i) non-directed; ii) unweighted; iii) anti-reflexive.

Basics – Definitions

Definitions Given a graph $G = (V, E, \omega)$, and an edge $e \in E$:

- If $e(1) = e(2)$, then e is called a **(self-)loop**.

NB If $\forall e \in E$, e is **not** a self-loop, then G is called **anti-reflexive**.

- When $\omega : E \to \Omega = \{\omega_0\}$, then G is called an **unweighted graph** and ω may be omitted: $G = (V, E)$.

- If $\exists f \in E : \begin{cases} f(1) = e(2) \\ f(2) = e(1) \\ \omega(f) = \omega(e) \end{cases}$, then e is called a **bi-directed edge**.

NB If all edges are bi-directed, then G is said to be **undirected or non-directed**. One can choose to count each edge only once.

Definition A graph is called **simple** if it is i) non-directed; ii) unweighted; iii) anti-reflexive.

Basics – Definitions

Definitions Given a graph $G = (V, E, \omega)$, and an edge $e \in E$:

- If $e(1) = e(2)$, then e is called a **(self-)loop**.

NB If $\forall e \in E$, e is **not** a self-loop, then G is called **anti-reflexive**.

- When $\omega : E \to \Omega = \{\omega_0\}$, then G is called an **unweighted graph** and ω may be omitted: $G = (V, E)$.

- If $\exists f \in E : \begin{cases} f(1) = e(2) \\ f(2) = e(1) \\ \omega(f) = \omega(e) \end{cases}$, then e is called a **bi-directed edge**.

NB If all edges are bi-directed, then G is said to be **undirected or non-directed**. One can choose to count each edge only once.

Definition A graph is called **simple** if it is i) non-directed; ii) unweighted; iii) anti-reflexive.

Basics – Definitions

Definitions Given a graph $G = (V, E, \omega)$, and an edge $e \in E$:

- If $e(1) = e(2)$, then e is called a **(self-)loop**.

NB If $\forall e \in E$, e is **not** a self-loop, then G is called **anti-reflexive**.

- When $\omega : E \to \Omega = \{\omega_0\}$, then G is called an **unweighted graph** and ω may be omitted: $G = (V, E)$.

- If $\exists f \in E : \begin{cases} f(1) = e(2) \\ f(2) = e(1) \\ \omega(f) = \omega(e) \end{cases}$, then e is called a **bi-directed edge**.

NB If all edges are bi-directed, then G is said to be **undirected or non-directed**. One can choose to count each edge only once.

Definition A graph is called **simple** if it is i) non-directed; ii) unweighted; iii) anti-reflexive.

Basics – Definitions

Definitions Given a graph $G = (V, E, \omega)$, and an edge $e \in E$:

- If $e(1) = e(2)$, then e is called a **(self-)loop**.

NB If $\forall e \in E$, e is **not** a self-loop, then G is called **anti-reflexive**.

- When $\omega : E \to \Omega = \{\omega_0\}$, then G is called an **unweighted graph** and ω may be omitted: $G = (V, E)$.

- If $\exists f \in E : \begin{cases} f(1) = e(2) \\ f(2) = e(1) \\ \omega(f) = \omega(e) \end{cases}$, then e is called a **bi-directed edge**.

NB If all edges are bi-directed, then G is said to be **undirected or non-directed**. One can choose to count each edge only once.

Definition A graph is called **simple** if it is i) non-directed; ii) unweighted; iii) anti-reflexive.

Representing Graphs – Sets, Drawings and Matrices

Sets and Drawings

Representing Graphs – Sets, Drawings and Matrices

Sets and Drawings

Matrix representations We enforce $V = \{1, ..., n\}$ with $n = |V|$.

- **Incidence**: Given $G = (V, E)$ anti-reflexive and unweighted, with $m = |E = \{e_1, ..., e_m\}|$, the so-called **incidence matrix of** G is a matrix $\mathbf{B} \in \{-1, 0, 1\}^{n \times m}$ s.t. $b(i, k) = \begin{cases} -1 & \text{if } e_k(1) = i \\ 1 & \text{if } e_k(2) = i \\ 0 & \text{otherwise} \end{cases}$

NB For undirected graphs: two definitions.

- **Adjacency**: Given $G = (V, E, \omega)$, the so-called **adjacency matrix of** G is a matrix $\mathbf{A} \in \{\{0\} \cup \Omega\}^{n \times n}$ s.t. $a(i, j) = \begin{cases} \omega((i, j)) & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$

NB Undirected graphs have symmetric adjacency matrices. Adjacency matrices of anti-reflexive graphs have 0 diagonal.

Representing Graphs – Sets, Drawings and Matrices

Sets and Drawings

Matrix representations We enforce $V = \{1, ..., n\}$ with $n = |V|$.

- **Incidence**: Given $G = (V, E)$ anti-reflexive and unweighted, with $m = |E = \{e_1, ..., e_m\}|$, the so-called **incidence matrix of** G is a matrix $\mathbf{B} \in \{-1, 0, 1\}^{n \times m}$ s.t. $b(i,k) = \begin{cases} -1 & \text{if } e_k(1) = i \\ 1 & \text{if } e_k(2) = i \\ 0 & \text{otherwise} \end{cases}$

NB For undirected graphs: two definitions.

- **Adjacency**: Given $G = (V, E, \omega)$, the so-called **adjacency matrix of** G is a matrix $\mathbf{A} \in \{\{0\} \cup \Omega\}^{n \times n}$ s.t. $a(i,j) = \begin{cases} \omega((i,j)) & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$

NB Undirected graphs have symmetric adjacency matrices. Adjacency matrices of anti-reflexive graphs have 0 diagonal.

Representing Graphs – Sets, Drawings and Matrices

Sets and Drawings

Matrix representations We enforce $V = \{1, \dots, n\}$ with $n = |V|$.

- **Incidence**: Given $G = (V, E)$ anti-reflexive and unweighted, with $m = |E = \{e_1, \dots, e_m\}|$, the so-called **incidence matrix of** G is a matrix $\mathbf{B} \in \{-1, 0, 1\}^{n \times m}$ s.t. $b(i,k) = \begin{cases} -1 & \text{if } e_k(1) = i \\ 1 & \text{if } e_k(2) = i \\ 0 & \text{otherwise} \end{cases}$

NB For undirected graphs: two definitions.

- **Adjacency**: Given $G = (V, E, \omega)$, the so-called **adjacency matrix of** G is a matrix $\mathbf{A} \in \{\{0\} \cup \Omega\}^{n \times n}$ s.t. $a(i,j) = \begin{cases} \omega((i,j)) & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$

NB Undirected graphs have symmetric adjacency matrices. Adjacency matrices of anti-reflexive graphs have 0 diagonal.

Question: How many adjacency matrices/incidence matrices possible for one graph ?

Representing Graphs – Isomorphisms

Definition Two graphs $G_1 = (V_1, E_1, \omega_1)$ and $G_2 = (V_2, E_2, \omega_2)$ are called **isomorphic** if

- $|V_1| = |V_2|$.
- $|E_1| = |E_2|$.
- $\exists s : V_1 \to V_2$ a bijection s.t. $(i, j) \in E_1 \iff (s(i), s(j)) \in E_2$.
- $\forall (i, j) \in E_1, \omega_1((i, j)) = \omega_2((s(i), s(j)))$.

Representing Graphs – Isomorphisms

Definition Two graphs $G_1 = (V_1, E_1, \omega_1)$ and $G_2 = (V_2, E_2, \omega_2)$ are called **isomorphic** if

- $|V_1| = |V_2|$.
- $|E_1| = |E_2|$.
- $\exists s : V_1 \to V_2$ a bijection s.t. $(i, j) \in E_1 \iff (s(i), s(j)) \in E_2$.
- $\forall (i, j) \in E_1, \omega_1((i, j)) = \omega_2((s(i), s(j)))$.

Remark One can say that G_2 is the graph G_1 in which nodes have been **relabelled** by s, which is thus called a **relabelling**.

Representing Graphs – Isomorphisms

Definition Two graphs $G_1 = (V_1, E_1, \omega_1)$ and $G_2 = (V_2, E_2, \omega_2)$ are called **isomorphic** if

- $|V_1| = |V_2|$.
- $|E_1| = |E_2|$.
- $\exists s : V_1 \to V_2$ a bijection s.t. $(i,j) \in E_1 \iff (s(i), s(j)) \in E_2$.
- $\forall (i,j) \in E_1, \omega_1((i,j)) = \omega_2((s(i), s(j)))$.

Remark One can say that G_2 is the graph G_1 in which nodes have been **relabelled** by s, which is thus called a **relabelling**.

Characterisation Two graphs of adjacency matrices respectively $\mathbf{A}_1$ and $\mathbf{A}_2$ are isomorphic iff $\mathbf{A}_1$ can be obtained from **simultaneous permutations of rows and columns** of $\mathbf{A}_2$.

Representing Graphs – Permutation matrices

Definition A matrix $\mathbf{M} \in \mathbb{R}_+^{n \times n}$ is called **bi-stochastic** if $\mathbf{M}\mathbf{1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}$.

Representing Graphs – Permutation matrices

Definition A matrix $\mathbf{M} \in \mathbb{R}_+^{n \times n}$ is called **bi-stochastic** if $\mathbf{M1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}$.

Definition A bi-stochastic matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ is a **permutation matrix**.

Representing Graphs – Permutation matrices

Definition A matrix $\mathbf{M} \in \mathbb{R}_+^{n \times n}$ is called **bi-stochastic** if $\mathbf{M}\mathbf{1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}$.

Definition A bi-stochastic matrix $\mathbf{P} \in \{0, 1\}^{n \times n}$ is a **permutation matrix**.
Remark This means that $\mathbf{P}$ has exactly one 1 per row and per column.
Exercise Prove that $\mathbf{P}^T = \mathbf{P}^{-1}$.

Representing Graphs – Permutation matrices

Definition A matrix $\mathbf{M} \in \mathbb{R}_+^{n \times n}$ is called **bi-stochastic** if $\mathbf{M1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}$.

Definition A bi-stochastic matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ is a **permutation matrix**.
Remark This means that $\mathbf{P}$ has exactly one 1 per row and per column.
Exercise Prove that $\mathbf{P}^T = \mathbf{P}^{-1}$.

Property Given $\sigma : \{1, ..., n\} \to \{1, ..., n\}$ a permutation. The matrix $\mathbf{P} \in \{0,1\}^{n \times n}$ s.t. $p(i,j) = 1 \iff j = \sigma(i)$ is a permutation matrix, and

$$\forall \mathbf{M} \in \mathbb{R}^{n \times n} \begin{cases} \mathbf{PM} = \mathbf{M}([\sigma(1), ..., \sigma(n)], :) \\ \mathbf{MP}^T = \mathbf{M}(:, [\sigma(1), ..., \sigma(n)]) \end{cases}$$

Proof Exercise

Representing Graphs – Permutation matrices

Definition A matrix $\mathbf{M} \in \mathbb{R}_+^{n \times n}$ is called **bi-stochastic** if $\mathbf{M1} = \mathbf{M}^T\mathbf{1} = \mathbf{1}$.

Definition A bi-stochastic matrix $\mathbf{P} \in \{0, 1\}^{n \times n}$ is a **permutation matrix**.
Remark This means that $\mathbf{P}$ has exactly one 1 per row and per column.
Exercise Prove that $\mathbf{P}^T = \mathbf{P}^{-1}$.

Property Given $\sigma : \{1, ..., n\} \to \{1, ..., n\}$ a permutation. The matrix $\mathbf{P} \in \{0, 1\}^{n \times n}$ s.t. $p(i, j) = 1 \iff j = \sigma(i)$ is a permutation matrix, and

$$\forall \mathbf{M} \in \mathbb{R}^{n \times n} \begin{cases} \mathbf{PM} = \mathbf{M}([\sigma(1), ..., \sigma(n)], :) \\ \mathbf{MP}^T = \mathbf{M}(:, [\sigma(1), ..., \sigma(n)]) \end{cases}$$

Proof Exercise

Characterisation Two graphs of adjacency matrices respectively $\mathbf{A}_1$ and $\mathbf{A}_2$ are isomorphic iff $\exists \mathbf{P} \in \{0, 1\}^{n \times n}$ a permutation matrix s.t. $\mathbf{A}_1 = \mathbf{PA}_2\mathbf{P}^T$.

Connectivity – Neighbourhood and degrees

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, we call:

- The **out-neighbourhood of** v the set $\mathcal{N}_{out}(v) = \{u : (v, u) \in E\}$,
  Its cardinal is called **the out-degree of** v: $d_{out}(v) = |\mathcal{N}_{out}(v)|$.
- The **in-neighbourhood of** v the set $\mathcal{N}_{in}(v) = \{u : (u, v) \in E\}$.
  Its cardinal is called **the in-degree of** v: $d_{in}(v) = |\mathcal{N}_{in}(v)|$.
- The **neighbourhood of** v the set $\mathcal{N}(v) = \mathcal{N}_{in}(v) \cup \mathcal{N}_{out}(v)$.
  Its cardinal is called **the degree of** v: $d(v) = |\mathcal{N}(v)|$

Exercise Given $G = (V, E)$ an unweighted graph, express the in- and out-degrees of a node $v \in V$ using G's adjacency matrix, then G's incidence matrix.

Connectivity – Neighbourhood and degrees

<u>Definition</u> Given a graph $G = (V, E, \omega)$ and a node $v \in V$, we call:

- The **out-neighbourhood of** v the set $\mathcal{N}_{out}(v) = \{u : (v, u) \in E\}$, Its cardinal is called **the out-degree of** v: $d_{out}(v) = |\mathcal{N}_{out}(v)|$.
- The **in-neighbourhood of** v the set $\mathcal{N}_{in}(v) = \{u : (u, v) \in E\}$. Its cardinal is called **the in-degree of** v: $d_{in}(v) = |\mathcal{N}_{in}(v)|$.
- The **neighbourhood of** v the set $\mathcal{N}(v) = \mathcal{N}_{in}(v) \cup \mathcal{N}_{out}(v)$. Its cardinal is called **the degree of** v: $d(v) = |\mathcal{N}(v)|$

<u>Exercise</u> Given $G = (V, E)$ an unweighted graph, express the in- and out-degrees of a node $v \in V$ using G's adjacency matrix, then G's incidence matrix.

Connectivity – Neighbourhood and degrees

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, we call:

- The **out-neighbourhood of** v the set $\mathcal{N}_{out}(v) = \{u : (v, u) \in E\}$,
  Its cardinal is called **the out-degree of** v: $d_{out}(v) = |\mathcal{N}_{out}(v)|$.
- The **in-neighbourhood of** v the set $\mathcal{N}_{in}(v) = \{u : (u, v) \in E\}$.
  Its cardinal is called **the in-degree of** v: $d_{in}(v) = |\mathcal{N}_{in}(v)|$.
- The **neighbourhood of** v the set $\mathcal{N}(v) = \mathcal{N}_{in}(v) \cup \mathcal{N}_{out}(v)$.
  Its cardinal is called **the degree of** v: $d(v) = |\mathcal{N}(v)|$

Exercise Given $G = (V, E)$ an unweighted graph, express the in- and out-degrees of a node $v \in V$ using G's adjacency matrix, then G's incidence matrix.

Connectivity – Neighbourhood and degrees

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, we call:

- The **out-neighbourhood of** v the set $\mathcal{N}_{out}(v) = \{u : (v, u) \in E\}$,
  Its cardinal is called **the out-degree of** v: $d_{out}(v) = |\mathcal{N}_{out}(v)|$.

- The **in-neighbourhood of** v the set $\mathcal{N}_{in}(v) = \{u : (u, v) \in E\}$.
  Its cardinal is called **the in-degree of** v: $d_{in}(v) = |\mathcal{N}_{in}(v)|$.

- The **neighbourhood of** v the set $\mathcal{N}(v) = \mathcal{N}_{in}(v) \cup \mathcal{N}_{out}(v)$.
  Its cardinal is called **the degree of** v: $d(v) = |\mathcal{N}(v)|$

Exercise Given $G = (V, E)$ an unweighted graph, express the in- and out-degrees of a node $v \in V$ using G's adjacency matrix, then G's incidence matrix.

Connectivity – Interlude on weighted degrees

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, thus:

- The **weighted out-degree of** v is

$$d^{\omega}_{out}(v) = \sum_{u:(v,u)\in E} \omega((v, u)).$$

- The **weighted in-degree of** v is

$$d^{\omega}_{in}(v) = \sum_{u:(u,v)\in E} \omega((u, v)).$$

- If G is **non-directed**, the **weighted degree of** v is

$$d^{\omega}(v) = \sum_{u:\{u,v\}\in E} \omega((u, v)).$$

Exercise Explain the limitation of weighted degree to non-directed graph. Any idea for extending this notion to directed graphs?

Connectivity – Paths

Definition Given a graph $G = (V, E, \omega)$, and $u, v \in V$, a **path from u to v** is a sequence of edges $e_1, ..., e_k \in E$ s.t.

$$i) : e_1(1) = u, \quad ii) : e_k(2) = v, \quad iii) : \forall i \in \{1, ..., k-1\}, e_i(2) = e_{i+1}(1).$$

The size of the sequence is called the **length of the path**. A path of length k can be called a **k-path**.

Connectivity – Paths

Definition Given a graph $G = (V, E, \omega)$, and $u, v \in V$, a **path from u to v** is a sequence of edges $e_1, ..., e_k \in E$ s.t.

$$i) : e_1(1) = u, \quad ii) : e_k(2) = v, \quad iii) : \forall i \in \{1, ..., k-1\}, e_i(2) = e_{i+1}(1).$$

The size of the sequence is called the **length of the path**. A path of length k can be called a **k-path**.

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, thus:

- The k-**hop out-neighbourhood of** v is the set $\mathcal{N}_{out}^k(v) = \{u : \exists \text{k-path from } u \text{ to } v\}$.
- The k-**hop in-neighbourhood of** v is the set $\mathcal{N}_{in}^k(v) = \{u : \exists \text{k-path from } v \text{ to } u\}$.
- The k-**hop neighbourhood of** v is $\mathcal{N}^k(v) = \mathcal{N}_{out}^k(v) \cup \mathcal{N}_{in}^k(v)$.

Connectivity – Paths

Definition Given a graph $G = (V, E, \omega)$, and $u, v \in V$, a **path from u to v** is a sequence of edges $e_1, ..., e_k \in E$ s.t.

$$i) : e_1(1) = u, \quad ii) : e_k(2) = v, \quad iii) : \forall i \in \{1, ..., k-1\}, e_i(2) = e_{i+1}(1).$$

The size of the sequence is called the **length of the path**. A path of length k can be called a **k-path**.

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, thus:

- The k-**hop out-neighbourhood of** v is the set $\mathcal{N}^k_{out}(v) = \{u : \exists \text{k-path from } u \text{ to } v\}$.
- The k-**hop in-neighbourhood of** v is the set $\mathcal{N}^k_{in}(v) = \{u : \exists \text{k-path from } v \text{ to } u\}$.
- The k-**hop neighbourhood of** v is $\mathcal{N}^k(v) = \mathcal{N}^k_{out}(v) \cup \mathcal{N}^k_{in}(v)$.

Connectivity – Paths

<u>Definition</u> Given a graph $G = (V, E, \omega)$, and $u, v \in V$, a **path from u to v** is a sequence of edges $e_1, ..., e_k \in E$ s.t.

$$i) : e_1(1) = u, \quad ii) : e_k(2) = v, \quad iii) : \forall i \in \{1, ..., k-1\}, e_i(2) = e_{i+1}(1).$$

The size of the sequence is called the **length of the path**. A path of length k can be called a **k-path**.

<u>Definition</u> Given a graph $G = (V, E, \omega)$ and a node $v \in V$, thus:

- The k-**hop out-neighbourhood of** v is the set $\mathcal{N}^k_{out}(v) = \{u : \exists\text{k-path from } u \text{ to } v\}$.
- The k-**hop in-neighbourhood of** v is the set $\mathcal{N}^k_{in}(v) = \{u : \exists\text{k-path from } v \text{ to } u\}$.
- The k-**hop neighbourhood of** v is $\mathcal{N}^k(v) = \mathcal{N}^k_{out}(v) \cup \mathcal{N}^k_{in}(v)$.

Connectivity – Paths

Definition Given a graph $G = (V, E, \omega)$, and $u, v \in V$, a **path from u to v** is a sequence of edges $e_1, ..., e_k \in E$ s.t.

$$i) : e_1(1) = u, \quad ii) : e_k(2) = v, \quad iii) : \forall i \in \{1, ..., k-1\}, e_i(2) = e_{i+1}(1).$$

The size of the sequence is called the **length of the path**. A path of length k can be called a **k-path**.

Definition Given a graph $G = (V, E, \omega)$ and a node $v \in V$, thus:

- The k-**hop out-neighbourhood of** v is the set $\mathcal{N}^k_{out}(v) = \{u : \exists \text{k-path from } u \text{ to } v\}$.
- The k-**hop in-neighbourhood of** v is the set $\mathcal{N}^k_{in}(v) = \{u : \exists \text{k-path from } v \text{ to } u\}$.
- The k-**hop neighbourhood of** v is $\mathcal{N}^k(v) = \mathcal{N}^k_{out}(v) \cup \mathcal{N}^k_{in}(v)$.

Property Given $\mathbf{A} \in \{0,1\}^{n \times n}$ the adjacency matrix of an unweighted graph, **the value of $a^k(i,j)$ is the number of k-paths** from node i to node j.
Proof Exercise.

Connectivity – Irreducibility

Definition A graph is said to be **(strongly) connected** if $\forall(u, v) \in V^2$, it exists a path from u to v. It is said to be **weakly connected** if the **underlying undirected** graph is connected.

Connectivity – Irreducibility

Definition A graph is said to be **(strongly) connected** if $\forall (u, v) \in V^2$, it exists a path from u to v. It is said to be **weakly connected** if the **underlying undirected** graph is connected.

Remark Every strongly connected graph is weakly connected. The opposite is false.

Connectivity – Irreducibility

Definition A graph is said to be **(strongly) connected** if $\forall (u, v) \in V^2$, it exists a path from u to v. It is said to be **weakly connected** if the **underlying undirected** graph is connected.

Remark Every strongly connected graph is weakly connected. The opposite is false.

Definition A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called **irreducible** if it **cannot** be written as

$$\mathbf{A} = \mathbf{P} \begin{bmatrix} \mathbf{A}_1 & * \\ 0 & \mathbf{A}_2 \end{bmatrix} \mathbf{P}^T, \text{ with } \mathbf{P} \text{ a permutation matrix and } \mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_1}$$

Connectivity – Irreducibility

Definition A graph is said to be **(strongly) connected** if $\forall (u, v) \in V^2$, it exists a path from u to v. It is said to be **weakly connected** if the **underlying undirected** graph is connected.

Remark Every strongly connected graph is weakly connected. The opposite is false.

Definition A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is called **irreducible** if it **cannot** be written as

$$\mathbf{A} = \mathbf{P} \begin{bmatrix} \mathbf{A}_1 & * \\ 0 & \mathbf{A}_2 \end{bmatrix} \mathbf{P}^T, \text{ with } \mathbf{P} \text{ a permutation matrix and } \mathbf{A}_1 \in \mathbb{R}^{n_1 \times n_1}$$

Characterisation **Irreducible matrices** are adjacency matrices of **strongly connected graphs**.
Proof Exercise.