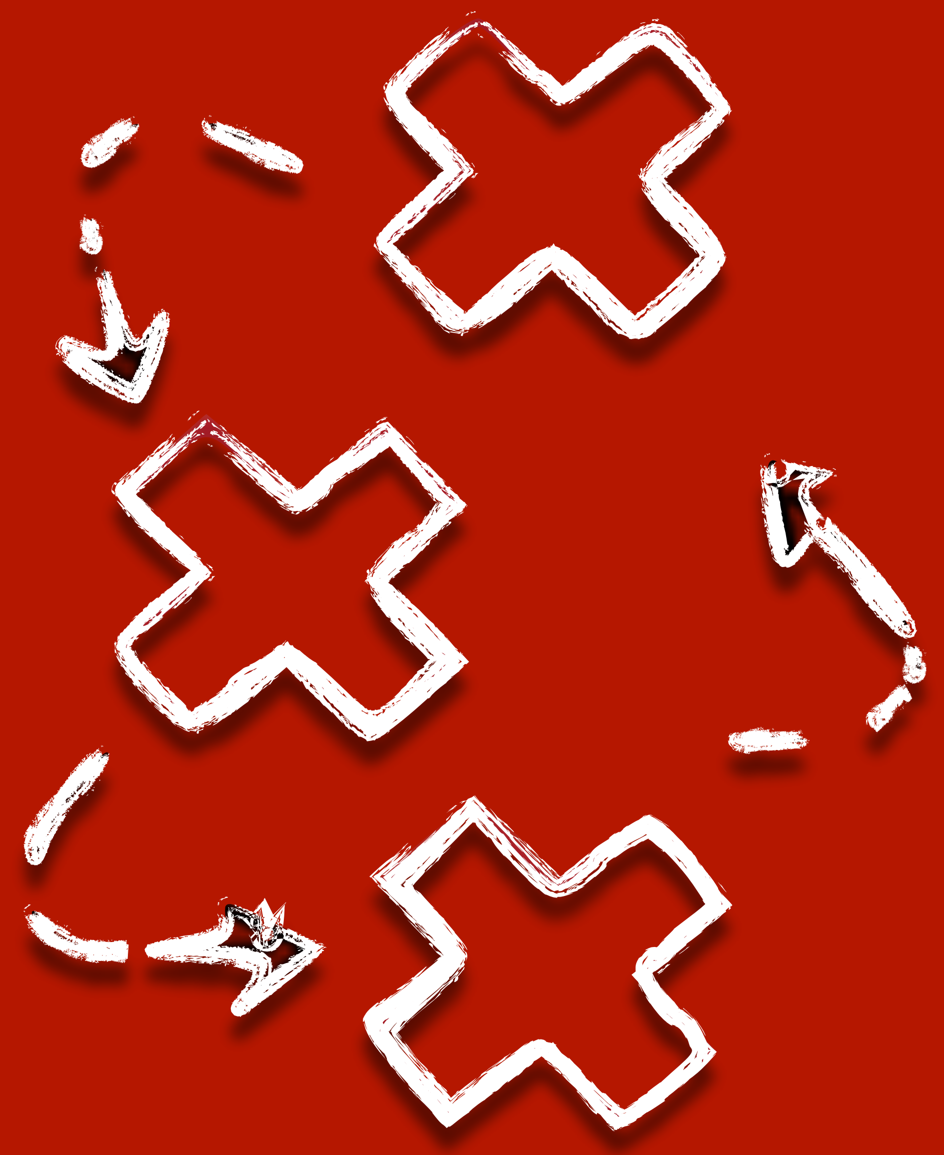


Deep learning 2: Causality & DL

1.2: Graphical models

Lecturer: Sara Magliacane

UvA - Spring 2022

Why should we care about Bayesian networks?

- We have a set of random variables $X_1, \ldots, X_p$ with joint $p(X_1, \ldots, X_p)$
- We have a DAG G, s.t. **each random variable** X_i is represented by **node** i

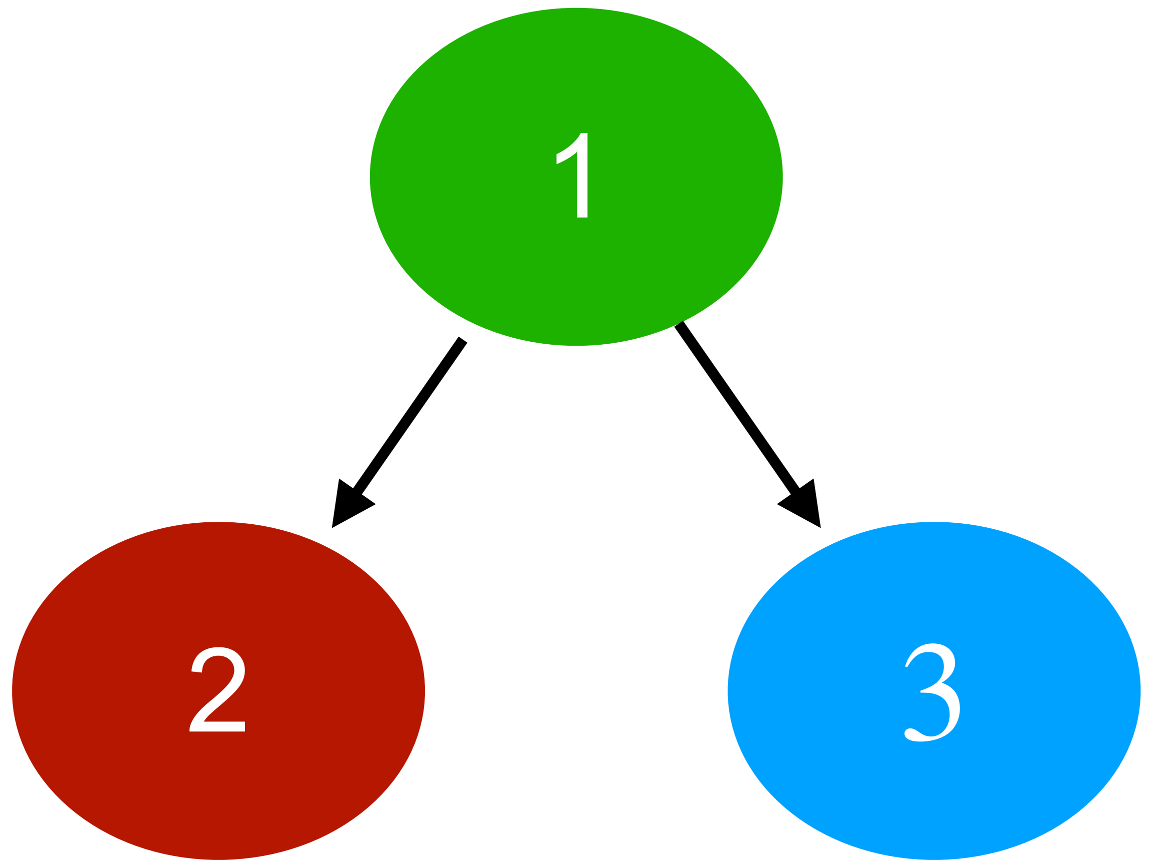
- We then say $P(X_1, \ldots, X_p)$ **factorizes over G** if

$$P(X_1, \ldots, X_p) = \prod_{i \in V} P(X_i \,|\, \mathbf{X}_{\mathrm{pa}(i)})$$

They can help simplify the factorisation

We can easily read conditional independences

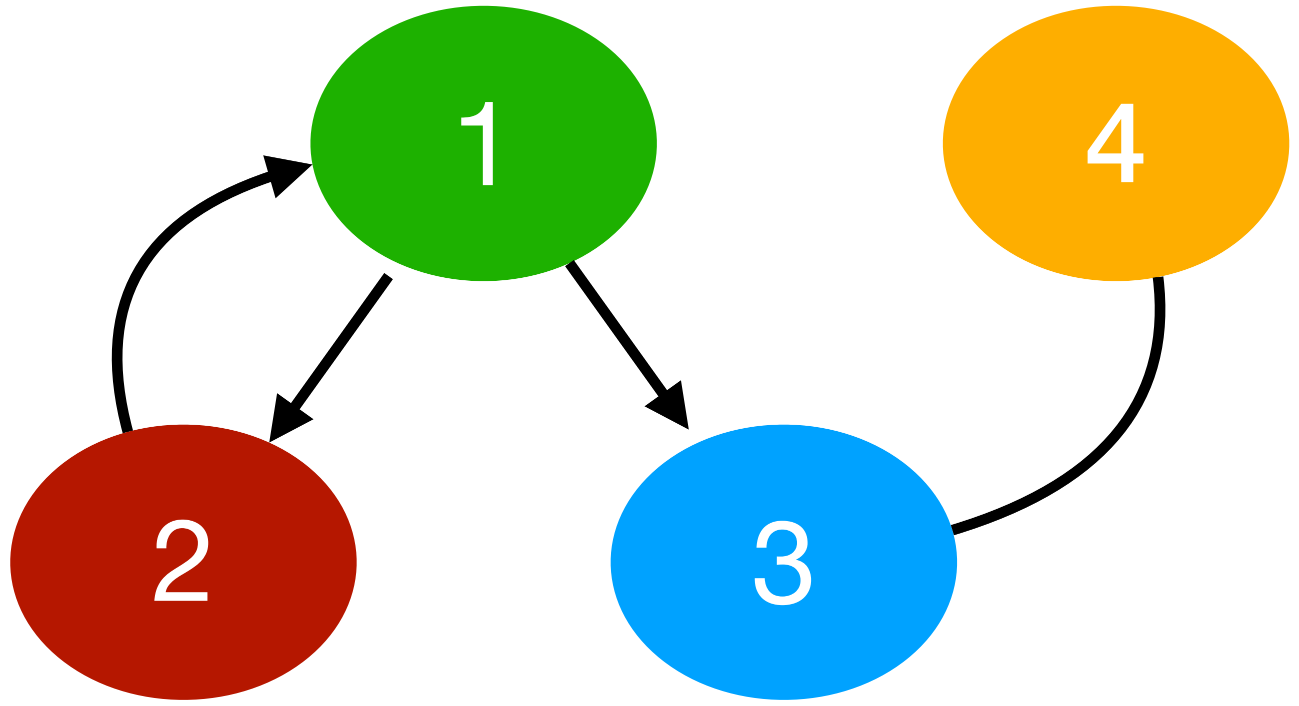
They can represent causal models



Graph terminology

- A graph G is a tuple $G = (\mathbf{V}, \mathbf{E})$:
  - $\mathbf{V}$ is the set of **nodes** (vertices)
  - $\mathbf{E}$ is the set of **edges** between two nodes, i.e. $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$

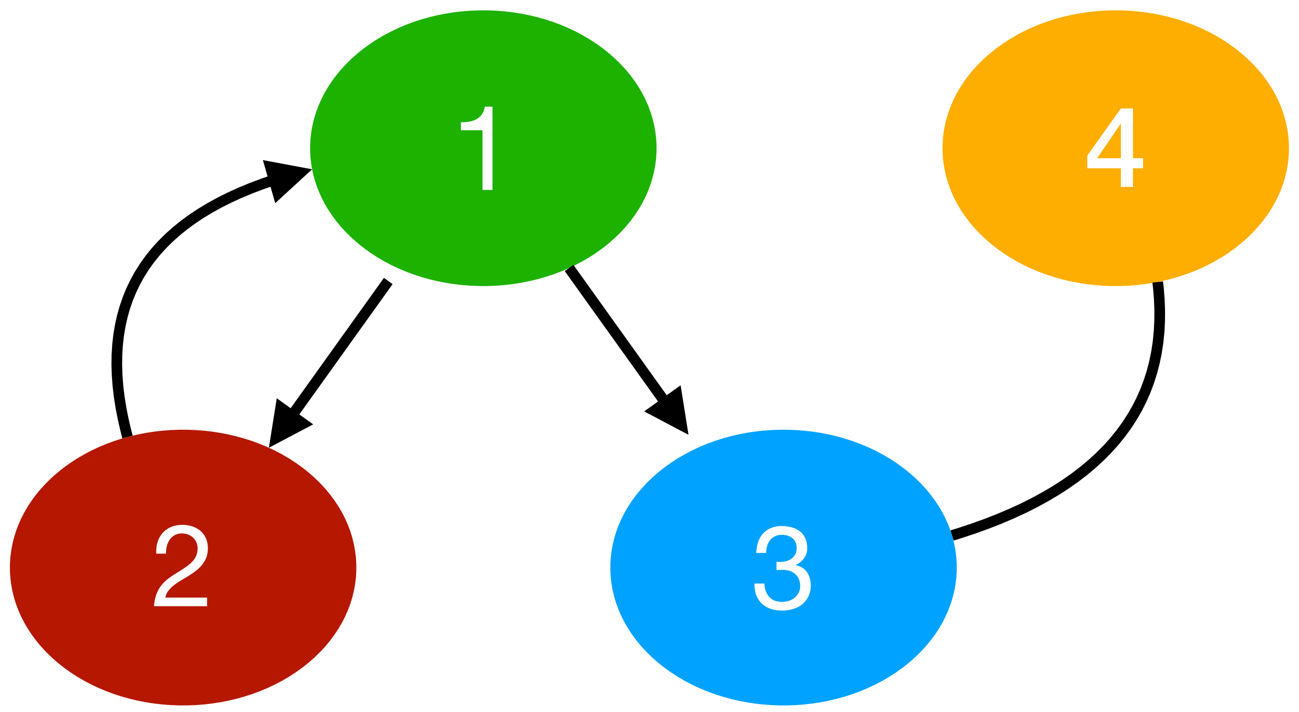
    $\Longrightarrow$ only one edge between an ordered pair of nodes

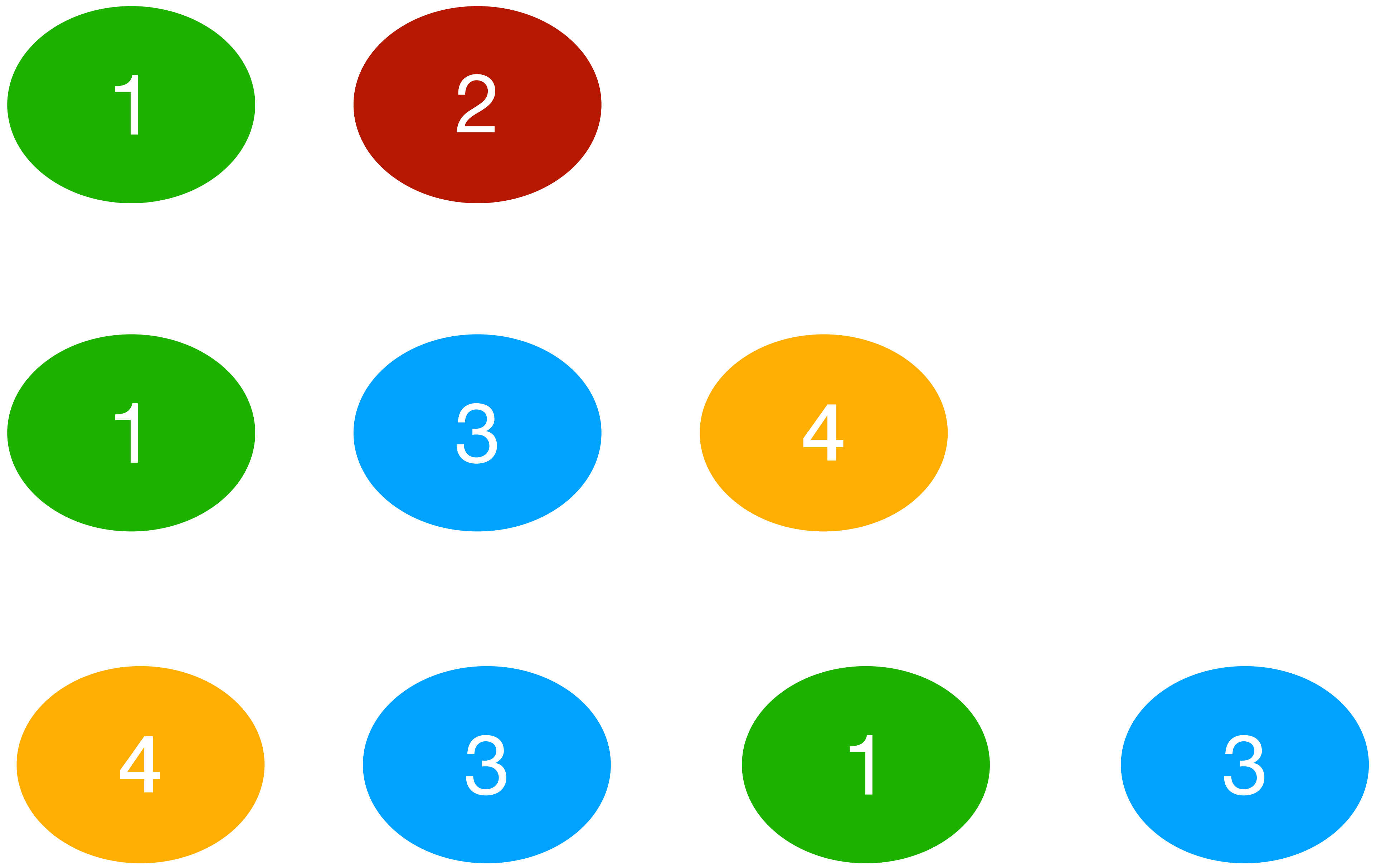


- Two nodes connected by an edge are **adjacent**

Graph terminology: paths

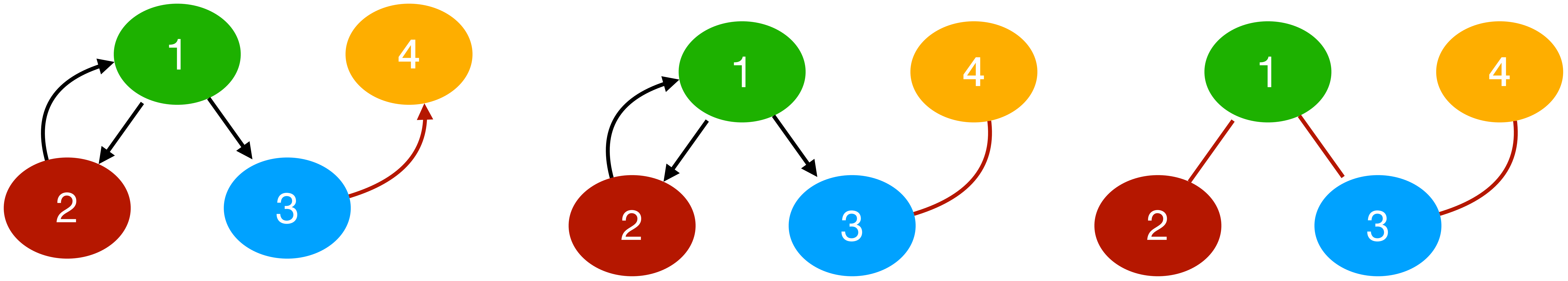
- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**





Directed graphs vs mixed graphs

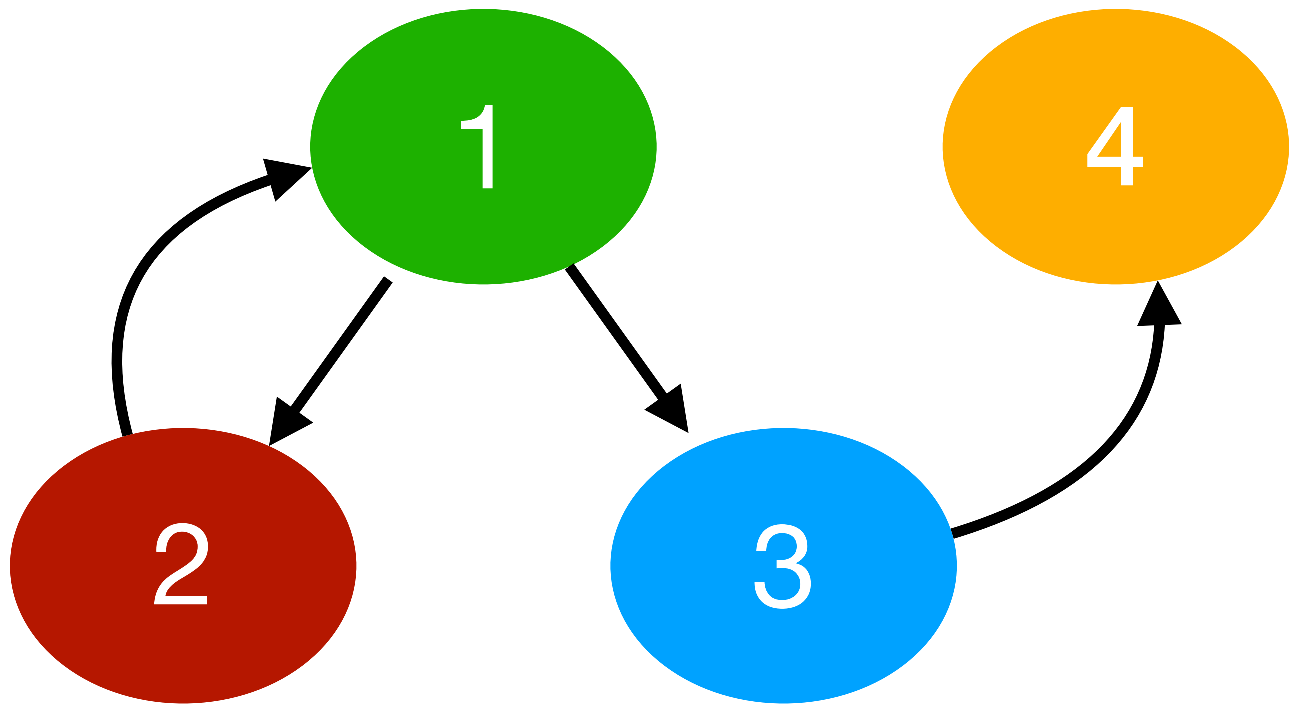
- A graph G is a tuple $G = (\mathbf{V}, \mathbf{E})$:
  - $\mathbf{V}$ is the set of **nodes** (vertices)
  - $\mathbf{E}$ is the set of **edges** between two nodes, i.e. $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$
  - If all edges are **directed** $\rightarrow$ then the graph is **directed**



Directed graph **Mixed graph** **Undirected graph**

Directed graphs: paths vs directed paths

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**
- A **directed path** between **node i and node j** is a path where **all edges point towards j**, i.e. $i \to \ldots \to j$

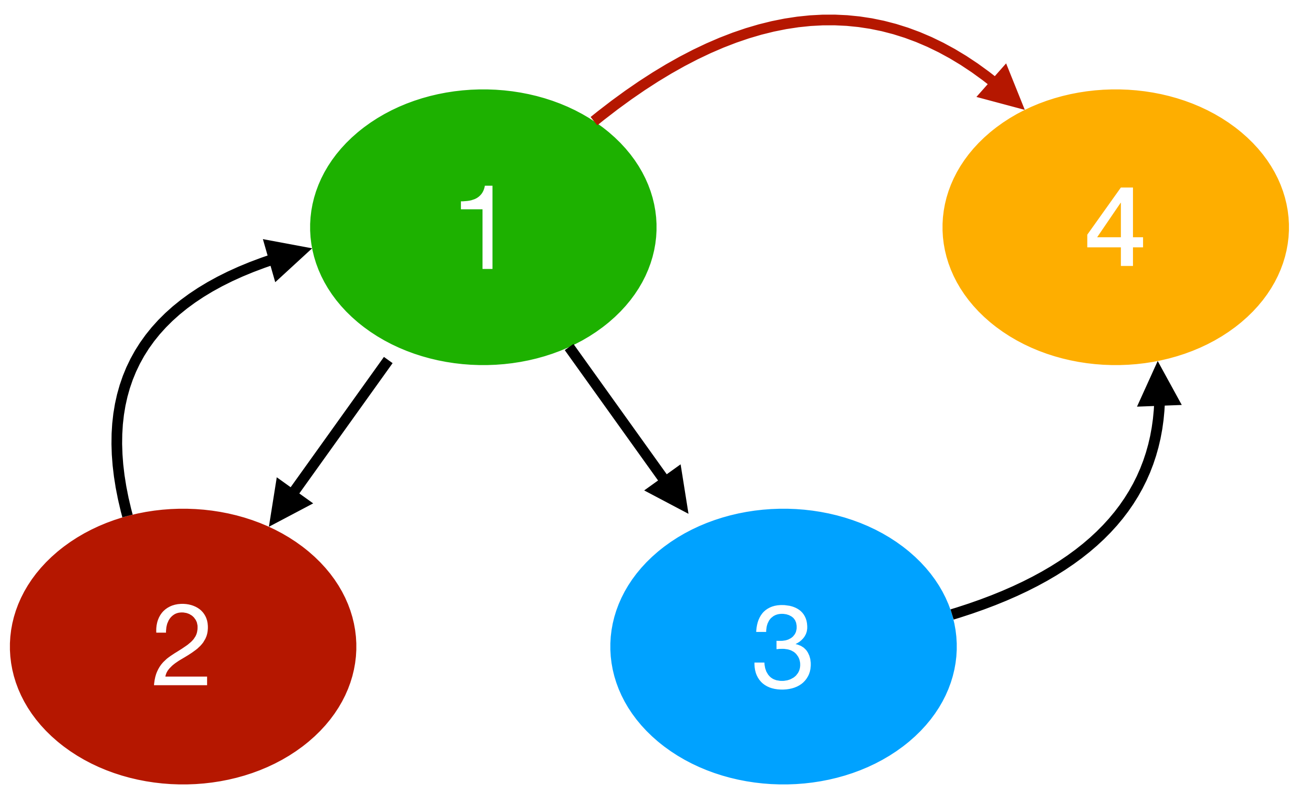


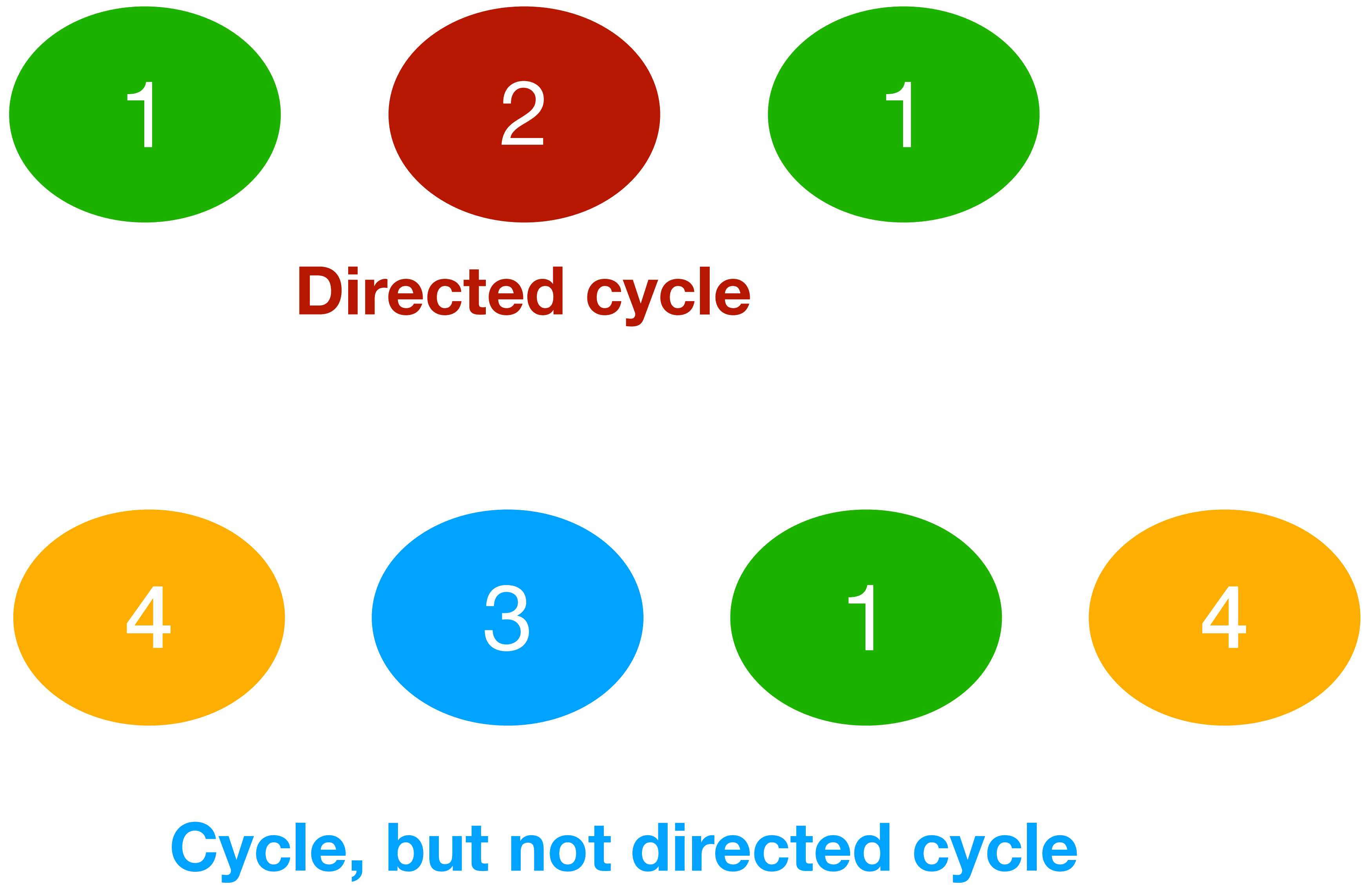
Directed path from 1 to 4

A path from 4 to 1, but not a directed path

Cycles and directed cycles

- A **cycle** is a path $(i, \ldots, i)$ (could also be a **self-cycle**)*
- A **directed cycle** is a directed path $(i, \ldots, i)$ *



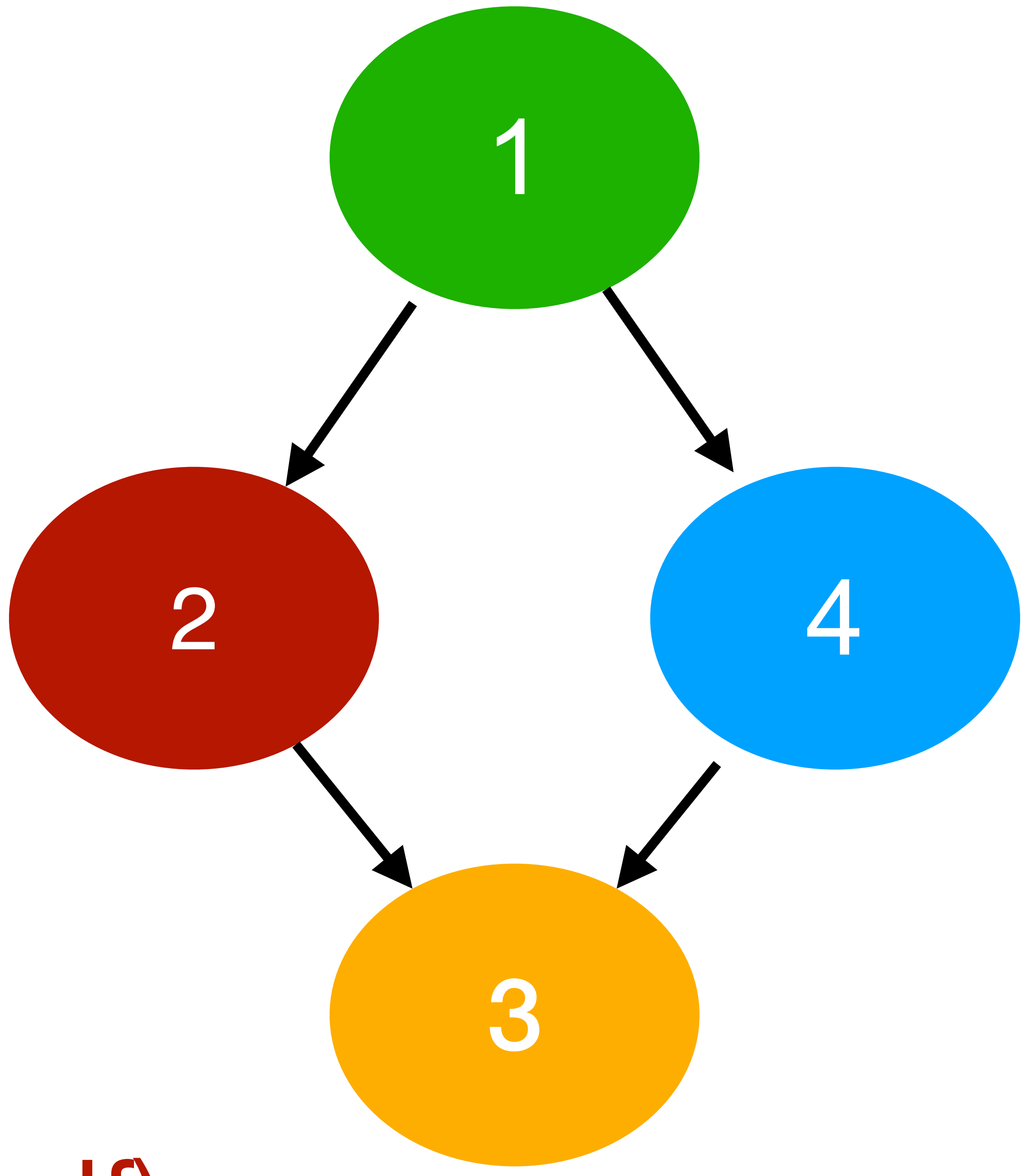


Directed Acyclic Graphs (DAGs)

- A DAG is a directed graph $G = (V, E)$:
  - V is the set of **nodes** (vertices)
  - E is the set of **directed edges** between the nodes
  - There are **no directed cycles**

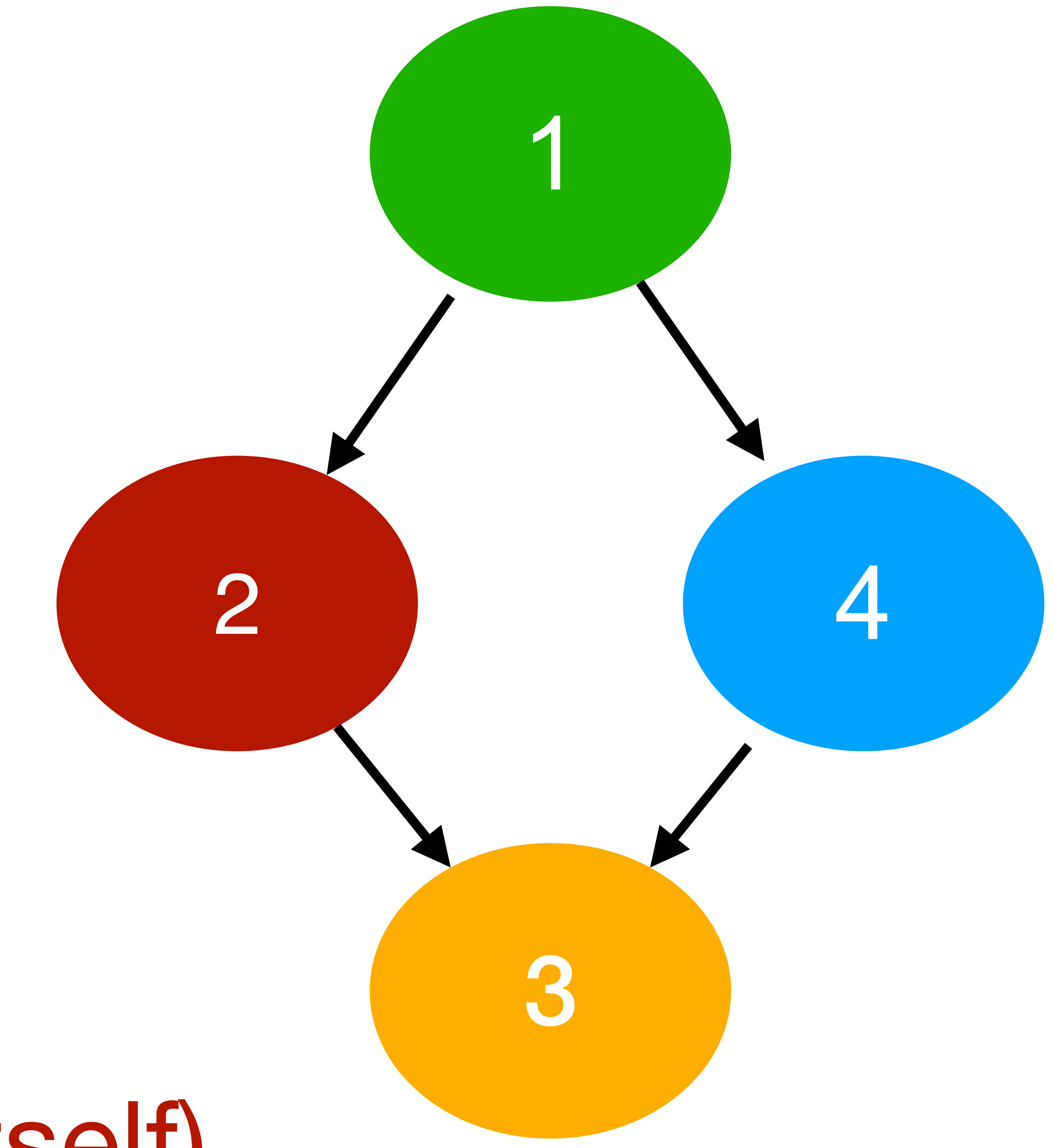
Relationships between nodes in a DAG

- **Parents** of a node $\mathrm{Pa}_G(V)$
  - Nodes that have an edge pointing to V
- **Children** of a node $\mathrm{Ch}_G(V)$
  - Nodes that have an edge pointing from V
- **Ancestors** of a node $\mathrm{An}_G(V)$
  - Nodes that have a **directed path** to V (including V itself)
- **Descendants** of a node $\mathrm{Desc}_G(V)$
  - Nodes that are reached from V via **directed paths** (including V itself)



Relationships between nodes in a DAG

[We omit G, when it is clear from the context]



- **Parents** of a node $\mathrm{Pa}(V)$
  - Nodes that have an edge pointing to V
- **Children** of a node $\mathrm{Ch}(V)$
  - Nodes that have an edge pointing from V
- **Ancestors** of a node $\mathrm{An}(V)$
  - Nodes that have a **directed path** to V (including V itself)
- **Descendants** of a node $\mathrm{Desc}(V)$
  - Nodes that are reached from V via **directed paths** (including V itself)

Kinship relationships for sets of nodes

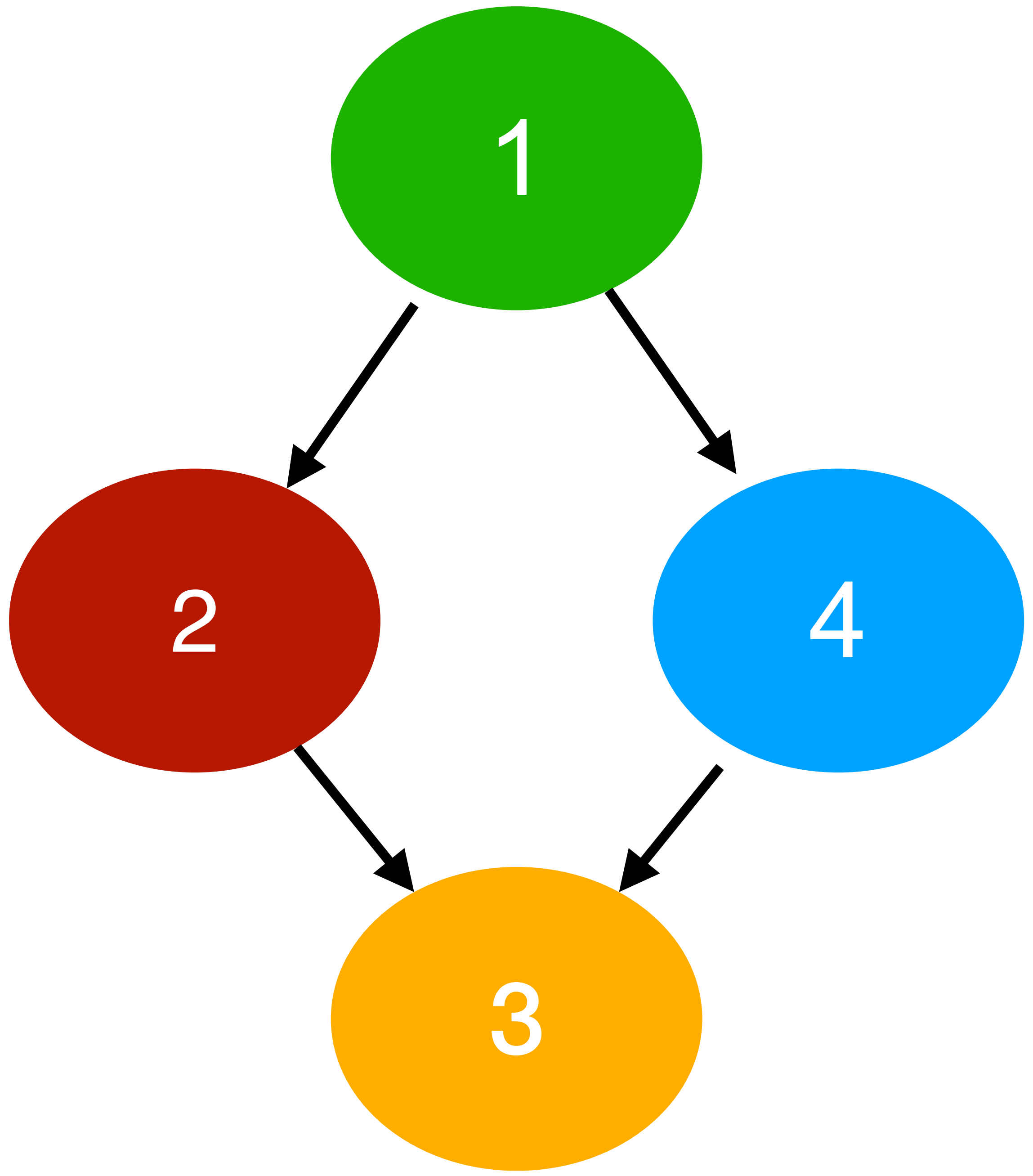
- We will use **bold** for sets (including sets of nodes)

- Parents of a set of nodes $\mathbf{A} \subseteq \mathbf{V}$:

$$\mathrm{Pa}(\mathbf{A}) := \bigcup_{V \in \mathbf{A}} \mathrm{Pa}(V)$$

$$\mathrm{Pa}(\{2,3\}) = \mathrm{Pa}(2) \cup \mathrm{Pa}(3) = \{1\} \cup \{2,4\} = \{1,2,4\}$$

- Similarly for children, ancestors and descendants

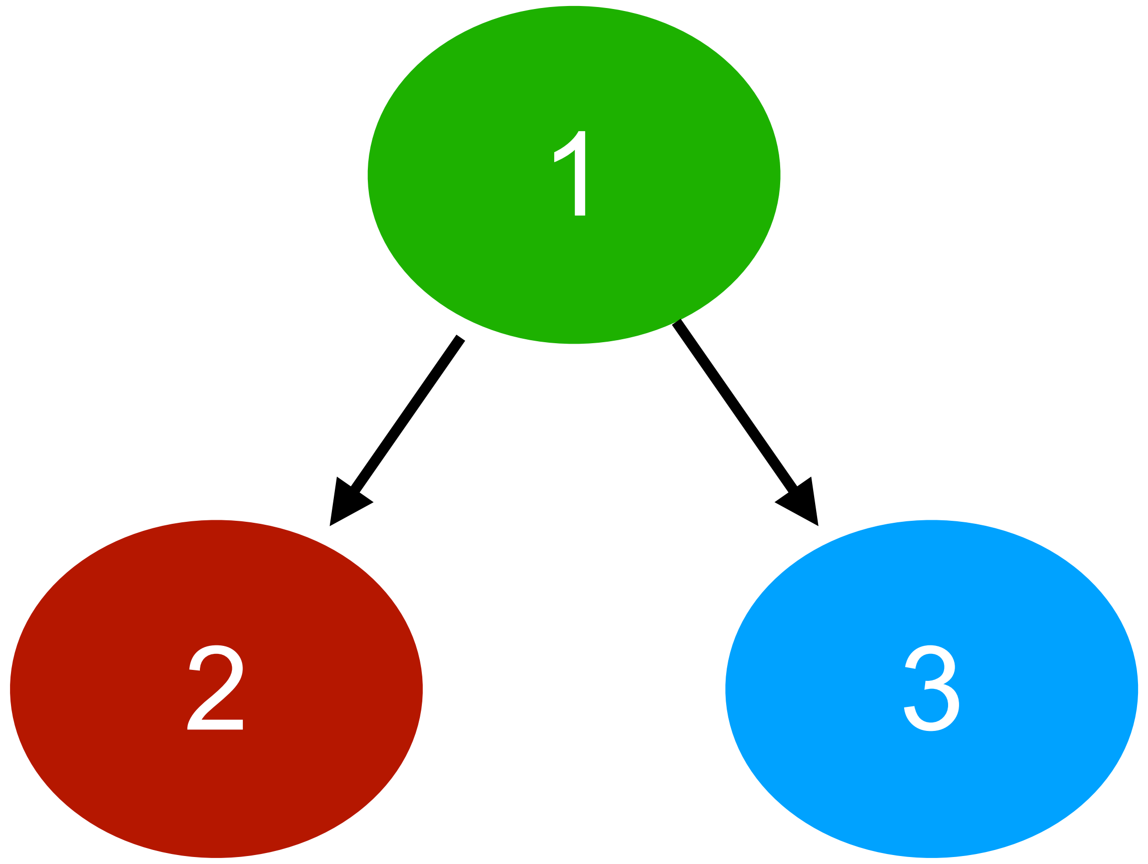


DAGs and random variables

- We can represent a factorisation of joint probability as a **DAG**

- **Each node $i \in \mathbf{V}$** represents a **random variable X_i**
  - For $\mathbf{A} \subseteq \mathbf{V}$, we can define $X_\mathbf{A} := \{X_i : i \in \mathbf{A}\}$

- **Edges** represent relationships between variables (*it will be clearer later*)



Factorizing joint distributions

- A joint distribution can always be factorized in several ways by iterating the **chain rule**

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X} \mid \mathbf{Y})P(\mathbf{Y})$$

Factorizing joint distributions

- A joint distribution can always be factorized in several ways by iterating the **chain rule**

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{X} \mid \mathbf{Y})P(\mathbf{Y})$$

- In general, given any **ordering** of the variables $(X_1, \ldots, X_p)$, we can write:

$$P(X_1, \ldots, X_p) = P(X_1)P(X_2 \mid X_1)\ldots P(X_p \mid X_1, \ldots, X_{p-1})$$

Factorizing joint distributions

- Given any **ordering** of the variables $(X_1, \dots, X_p)$ we can write:

$$P(X_1, \dots, X_p) = P(X_1)P(X_2 \mid X_1)\dots P(X_p \mid X_1, \dots, X_{p-1})$$

- For example $P(X, Y, Z)$ can be equivalently factorized as:
  - $P(X, Y, Z) =$
  - $P(X, Z, Y) =$
  - $P(Z, Y, X) =$
  - …

Exploiting conditional independences

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$P(X_1, \ldots, X_p) = P(X_1)P(X_2 \mid X_1)\ldots P(X_p \mid X_1, \ldots, X_{p-1})$$

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \mid X_{\mathbf{Z}} \implies P(X_i \mid X_j, X_{\mathbf{Z}}) = P(X_i \mid X_{\mathbf{Z}})$$

(special case $X_i \perp\!\!\!\perp X_j \implies P(X_i \mid X_j) = P(X_i)$)

Quick recap: Independent random variables

- **Definition:** Two random variables X and Y are **independent** iff:

$$\forall x, y : P(X = x, Y = y) = P(X = x)P(Y = y)$$

- We then write $X \perp\!\!\!\perp Y$

- This is equivalent to $P(X = x \mid Y = y) = P(X = x)$ (and vice versa for Y)

- Intuitively, this means that knowing the value of Y **will not tell us anything** about the distribution of X.

Quick recap: conditional independence

- X is independent of Y **conditioned/given** Z iff

$$\forall x, y, z : P(X = x \mid Y = y, Z = z) = P(X = x \mid Z = z) \qquad (\text{for } P(Z = z) > 0)$$

- We then write $X \perp\!\!\!\perp Y \mid Z$, otherwise $X \not\perp\!\!\!\perp Y \mid Z$
- Intuitively this means that **Y does not add any information** to predict X that isn't already offered by Z
- Z can be a **set of variables,** e.g. $X \perp\!\!\!\perp Y \mid Z_1, Z_2$

Exploiting conditional independences

- Given any **ordering** of the variables $(X_1, \ldots, X_p)$ we can write:

$$P(X_1, \ldots, X_p) = P(X_1)P(X_2 | X_1) \ldots P(X_p | X_1, \ldots, X_{p-1})$$

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j | X_{\mathbf{Z}} \implies P(X_i | X_j, X_{\mathbf{Z}}) = P(X_i | X_{\mathbf{Z}})$$

(special case $X_i \perp\!\!\!\perp X_j \implies P(X_i | X_j) = P(X_i)$)

Simpler factorisations mean less parameters to learn (less data)

Exploiting conditional independences

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \mid X_{\mathbf{Z}} \implies P(X_i \mid X_j, X_{\mathbf{Z}}) = P(X_i \mid X_{\mathbf{Z}})$$

- For example: $X \perp\!\!\!\perp Y \mid Z$:
  - $P(X, Y, Z) = P(X)P(Y \mid X)P(Z \mid X, Y)$
  - $P(X, Z, Y) = P(X)P(Z \mid X)P(Y \mid X, Z)$
  - $P(Z, Y, X) = P(Z)P(Y \mid Z)P(X \mid Y, Z)$ ...

Exploiting conditional independences

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \mid X_{\mathbf{Z}} \implies P(X_i \mid X_j, X_{\mathbf{Z}}) = P(X_i \mid X_{\mathbf{Z}})$$

- For example: $X \perp\!\!\!\perp Y \mid Z$:
  - $P(X, Y, Z) = P(X)P(Y \mid X)P(Z \mid X, Y)$
  - $P(X, Z, Y) = P(X)P(Z \mid X)P(Y \mid \cancel{X,} Z) = P(X)P(Z \mid X)P(Y \mid Z)$
  - $P(Z, Y, X) = P(Z)P(Y \mid Z)P(X \mid \cancel{Y,} Z) = P(Z)P(Y \mid Z)P(X \mid Z)$

Exploiting conditional independences

- We can **simplify** the factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j \mid X_{\mathbf{Z}} \implies P(X_i \mid X_j, X_{\mathbf{Z}}) = P(X_i \mid X_{\mathbf{Z}})$$

- For example: if $X \perp\!\!\!\perp Y \mid Z$ and $X \perp\!\!\!\perp Z$

  - $P(X, Y, Z) = P(X)P(Y \mid X)P(Z \mid X, Y)$

[Spoiler: each factorisation is a different DAG]

  - $P(X, Z, Y) = P(X)P(Z \mid \cancel{X})P(Y \mid \cancel{X}, Z) = P(X)P(Z)P(Y \mid Z)$
  - $P(Z, Y, X) = P(Z)P(Y \mid Z)P(X \mid \cancel{Y}, \cancel{Z}) = P(Z)P(Y \mid Z)P(X)$

Bayesian networks

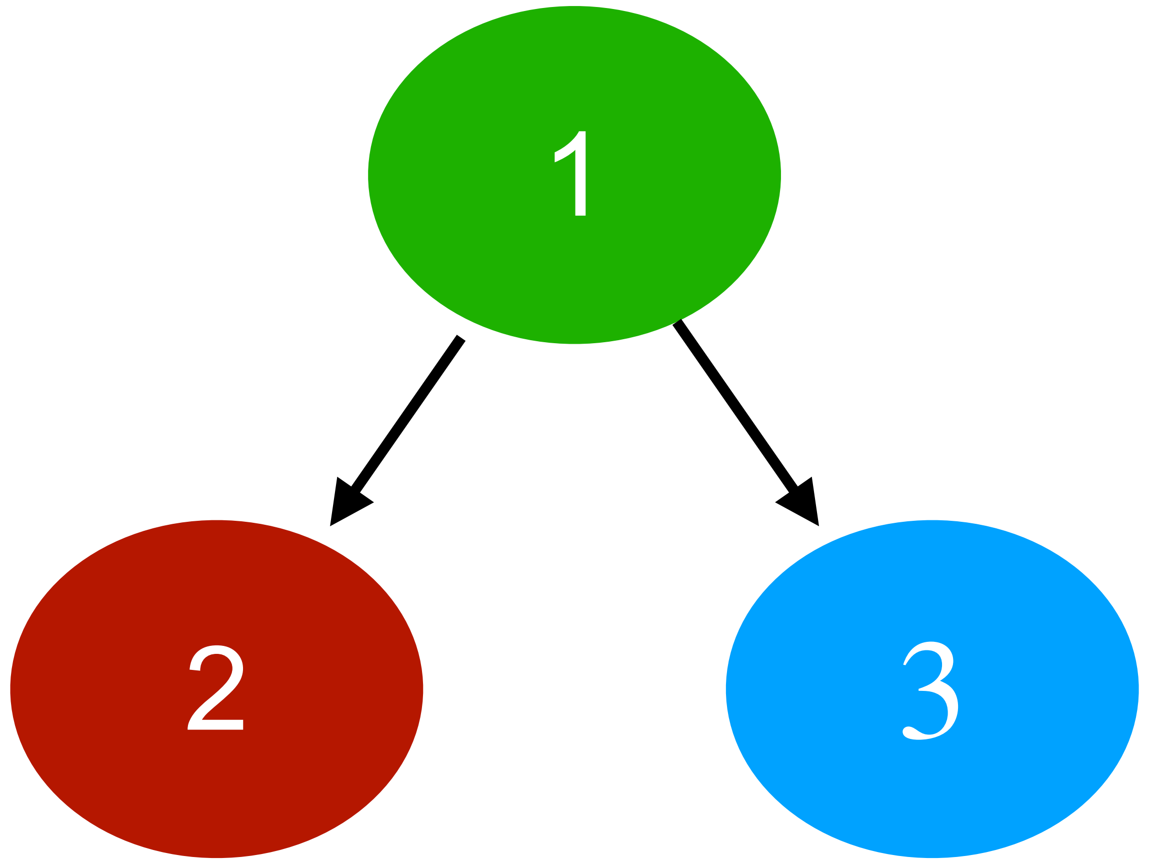
- We have a set of random variables $X_1, \ldots, X_p$ with joint $p(X_1, \ldots, X_p)$
- We have a DAG G, s.t. **each random variable** X_i is represented by **node** i

- We then say $P(X_1, \ldots, X_p)$ **factorizes over G** if

$$P(X_1, \ldots, X_p) = \prod_{i \in V} P(X_i \mid \mathbf{X}_{\text{pa}(i)})$$

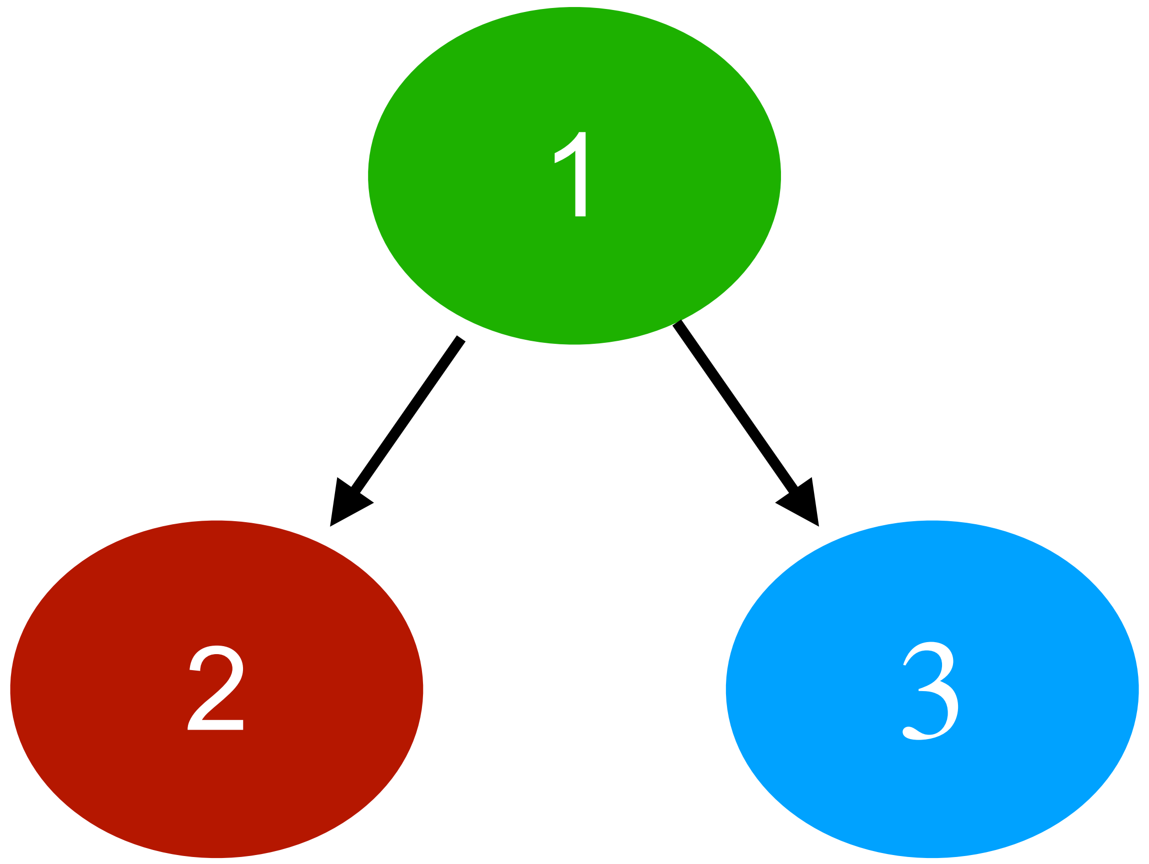
- A **Bayesian network** (BN) is the tuple (G, p) s.t. p **factorizes over** G

Example Bayesian networks



$$P(X_1, X_2, X_3) = P(X_1 \mid X_{\mathrm{Pa}(1)}) P(X_2 \mid X_{\mathrm{Pa}(2)}) P(X_3 \mid X_{\mathrm{Pa}(3)})$$

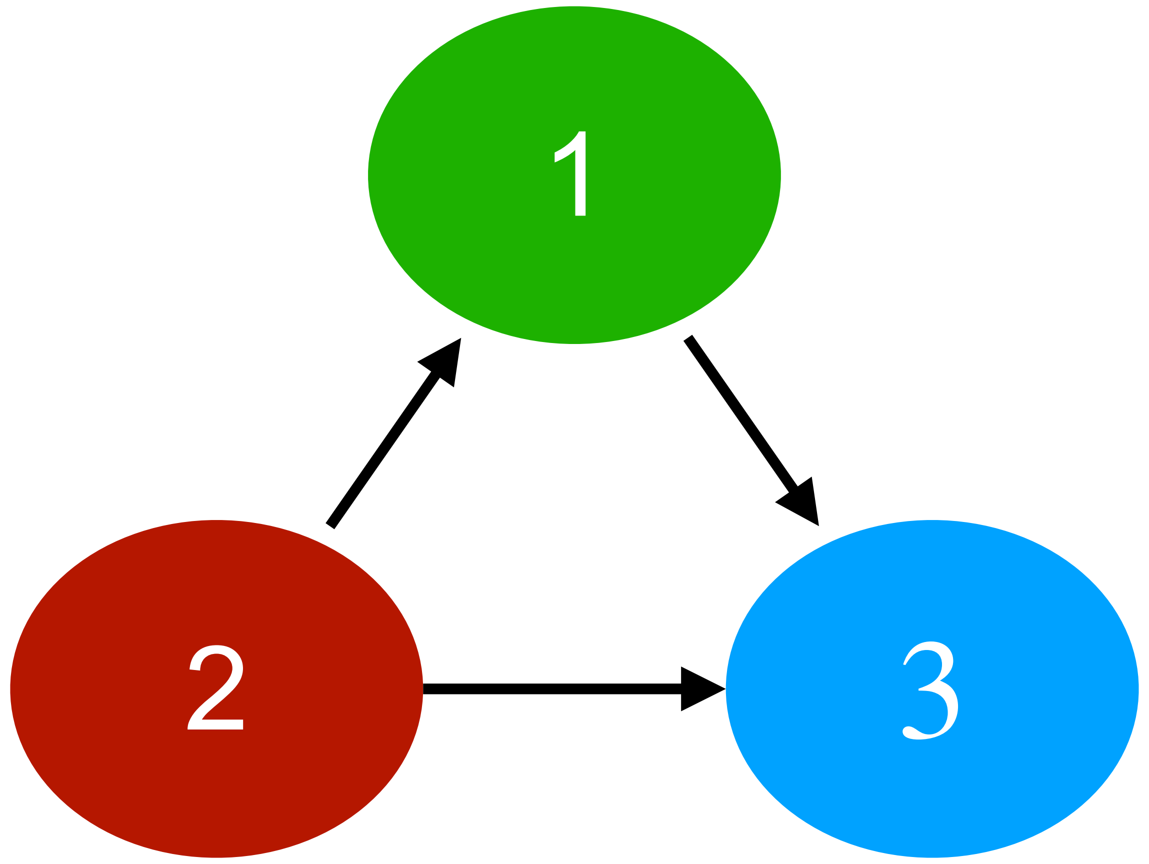
Example Bayesian networks



$$P(X_1, X_2, X_3) = P(X_1 \mid X_{\text{Pa}(1)}) P(X_2 \mid X_{\text{Pa}(2)}) P(X_3 \mid X_{\text{Pa}(3)})$$

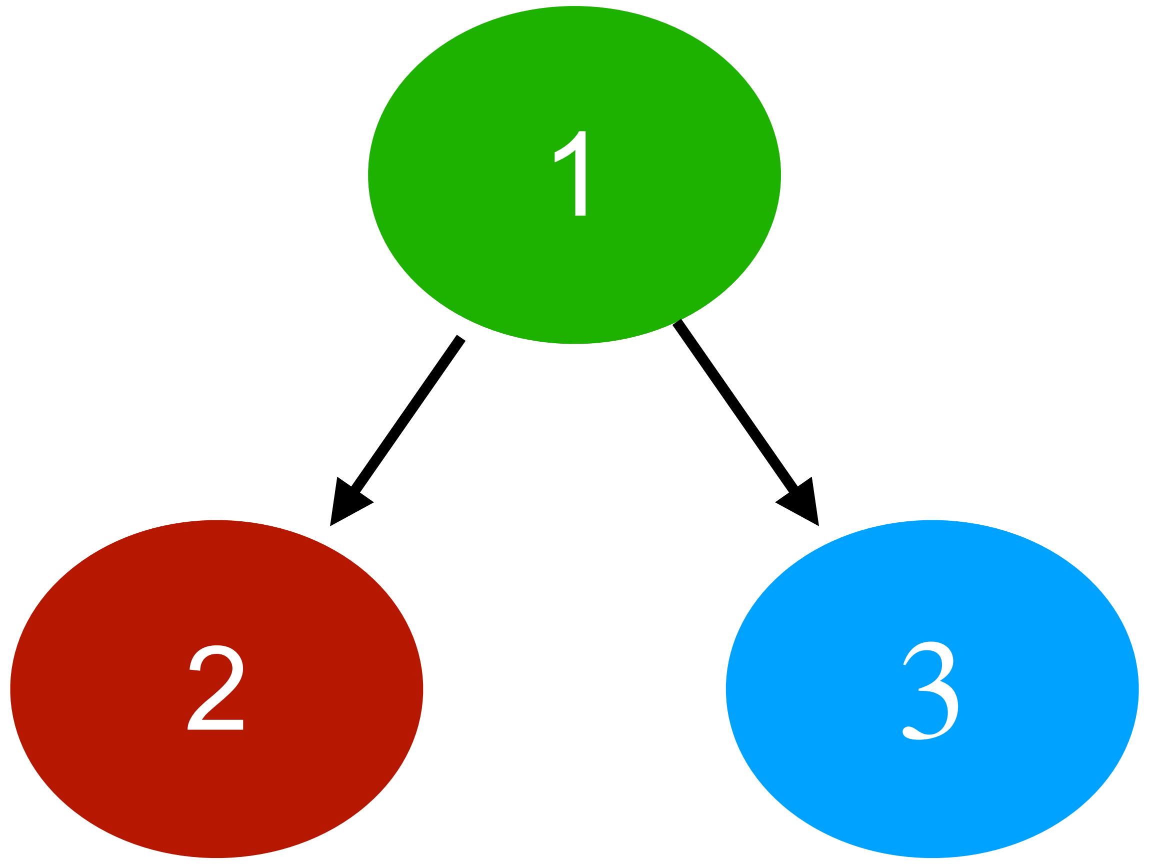
$$P(X_1) \quad P(X_2 \mid X_1) \quad P(X_3 \mid X_1)$$

The DAG/factorization is not unique for the joint distribution:



$$P(X_1, X_2, X_3) = P(X_2) P(X_1 \mid X_2) P(X_3 \mid X_1, X_2)$$

Example Bayesian networks

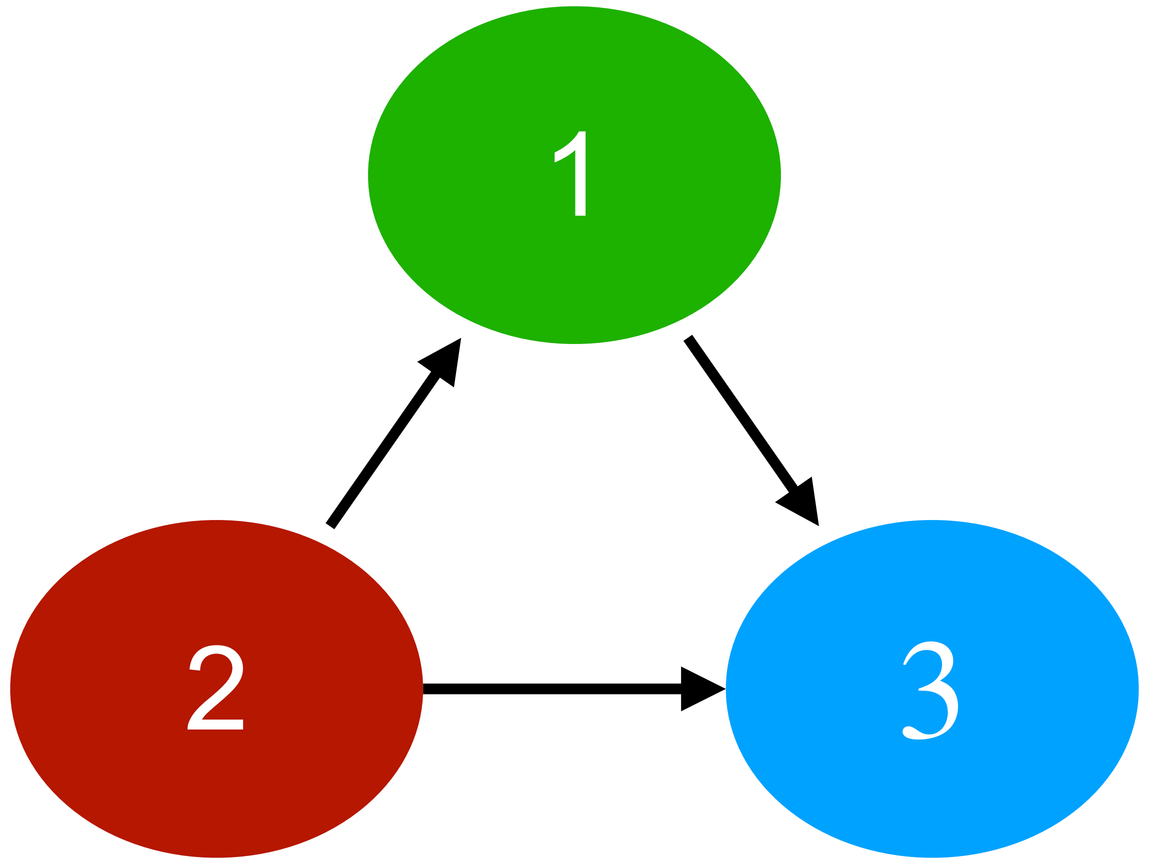


$$P(X_1, X_2, X_3) = P(X_1 \mid X_{\mathrm{Pa}(1)}) P(X_2 \mid X_{\mathrm{Pa}(2)}) P(X_3 \mid X_{\mathrm{Pa}(3)})$$

$$P(X_1) \quad P(X_2 \mid X_1) \quad P(X_3 \mid X_1)$$

The DAG/factorization is not unique for the joint distribution:

(for example any fully connected graph factorizes p)



$$P(X_1, X_2, X_3) = P(X_2) P(X_1 \mid X_2) P(X_3 \mid X_1, X_2)$$

Multiple BN can represent a distribution

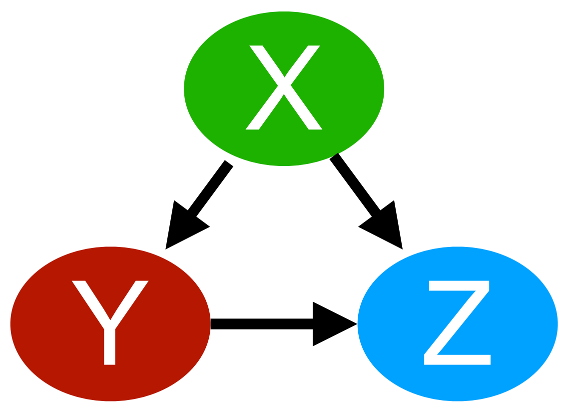
- We can **simplify** a factorisation by using **conditional independences:**

$$X_i \perp\!\!\!\perp X_j | X_{\mathbf{Z}} \implies P(X_i | X_j, X_{\mathbf{Z}}) = P(X_i | X_{\mathbf{Z}})$$

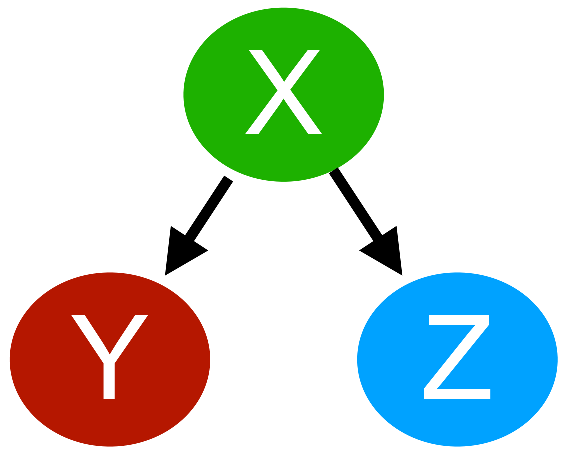
- For example: $X \perp\!\!\!\perp Y | Z$:

[Each factorisation can be represented with a DAG]

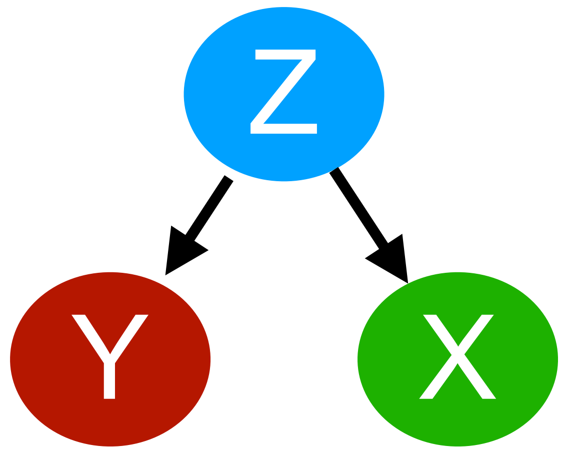
  - $P(X, Y, Z) = P(X)P(Y|X)P(Z|X, Y)$



  - $P(X, Z, Y) = P(X)P(Z|X)P(Y|\cancel{X}, Z) = P(X)P(Z|X)P(Y|Z)$



  - $P(Z, Y, X) = P(Z)P(Y|Z)P(X|\cancel{Y}, Z) = P(Z)P(Y|Z)P(X|Z)$



Why should we care about Bayesian networks?

- We have a set of random variables $X_1, \ldots, X_p$ with joint $p(X_1, \ldots, X_p)$
- We have a DAG G, s.t. **each random variable** X_i is represented by **node** i

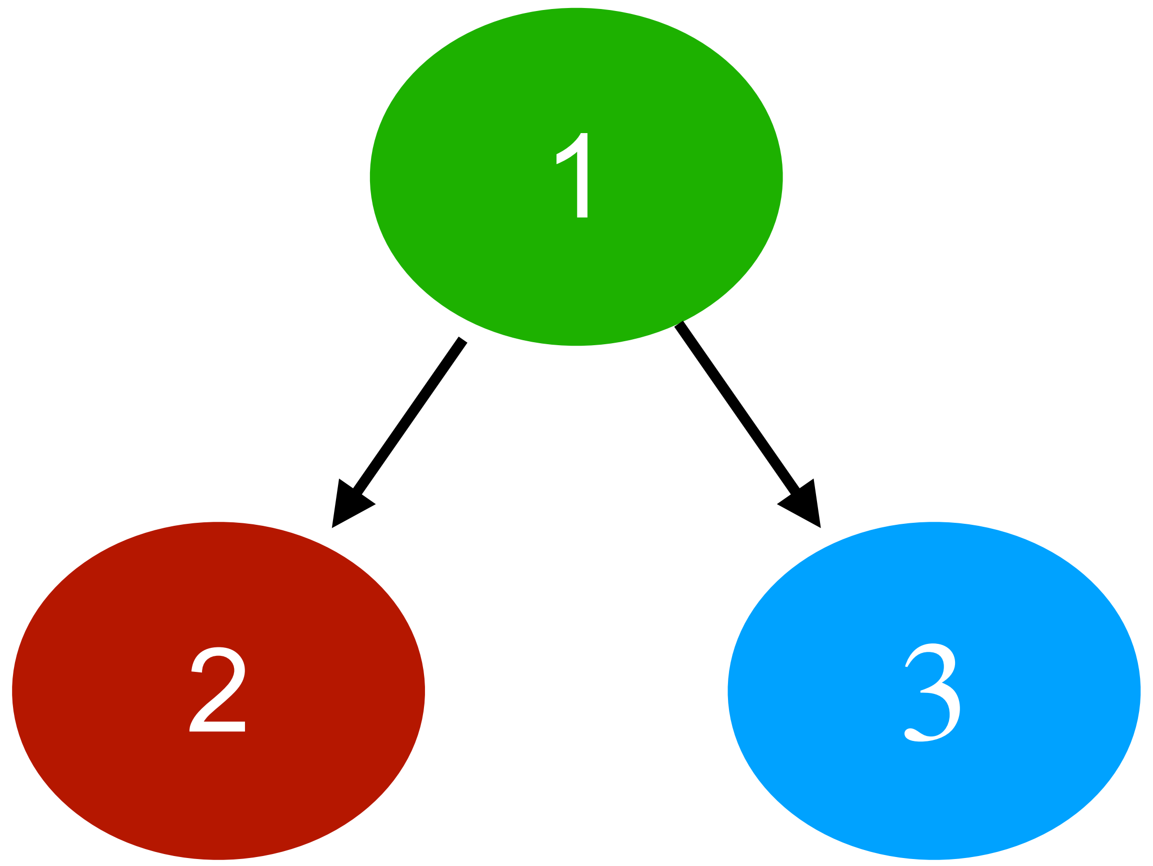
- We then say $P(X_1, \ldots, X_p)$ **factorizes over G** if

$$P(X_1, \ldots, X_p) = \prod_{i \in V} P(X_i \,|\, \mathbf{X}_{\mathrm{pa}(i)})$$

They can help simplify the factorisation

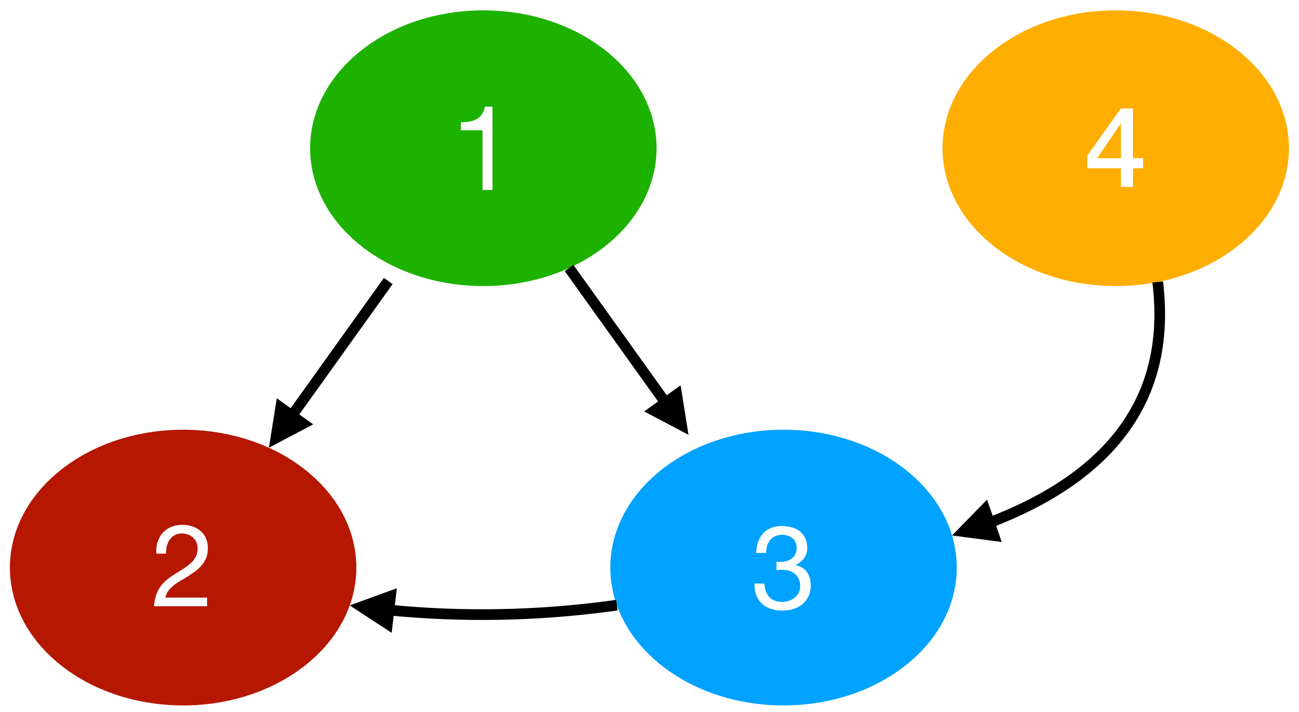
We can easily read conditional independences

They can represent causal models



Graph terminology: collider on a path

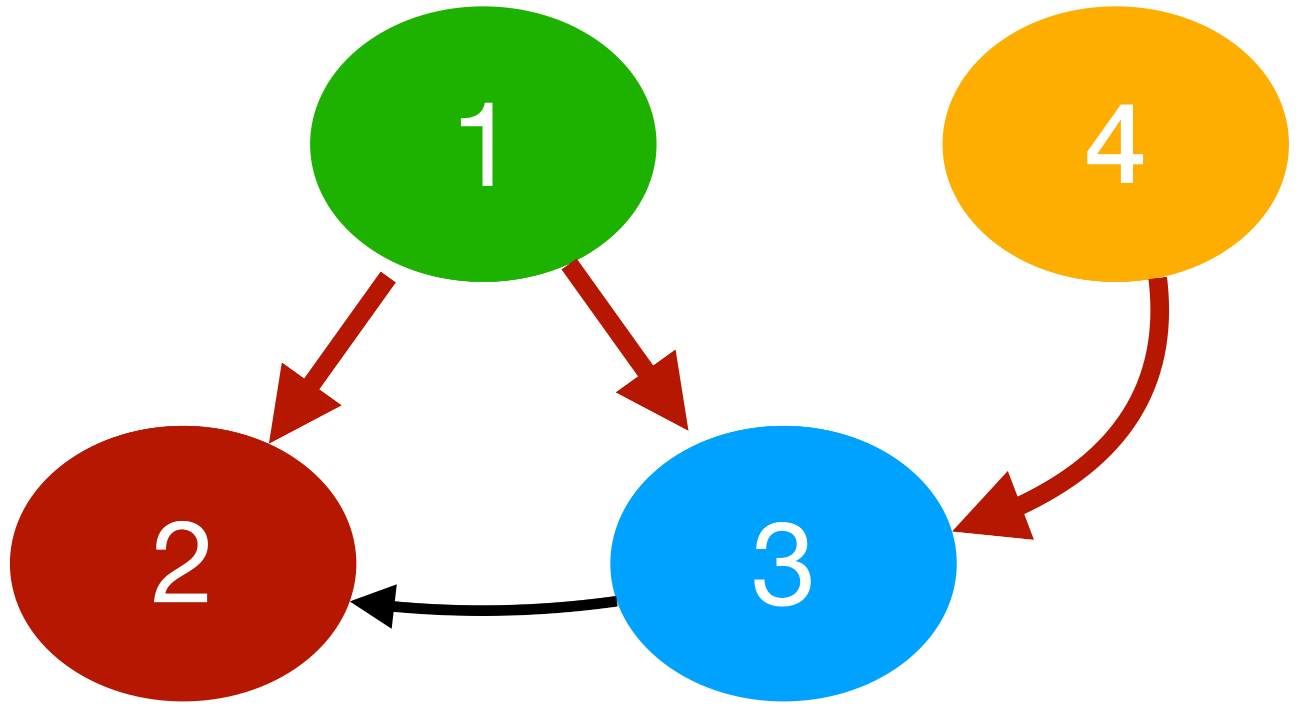
- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**

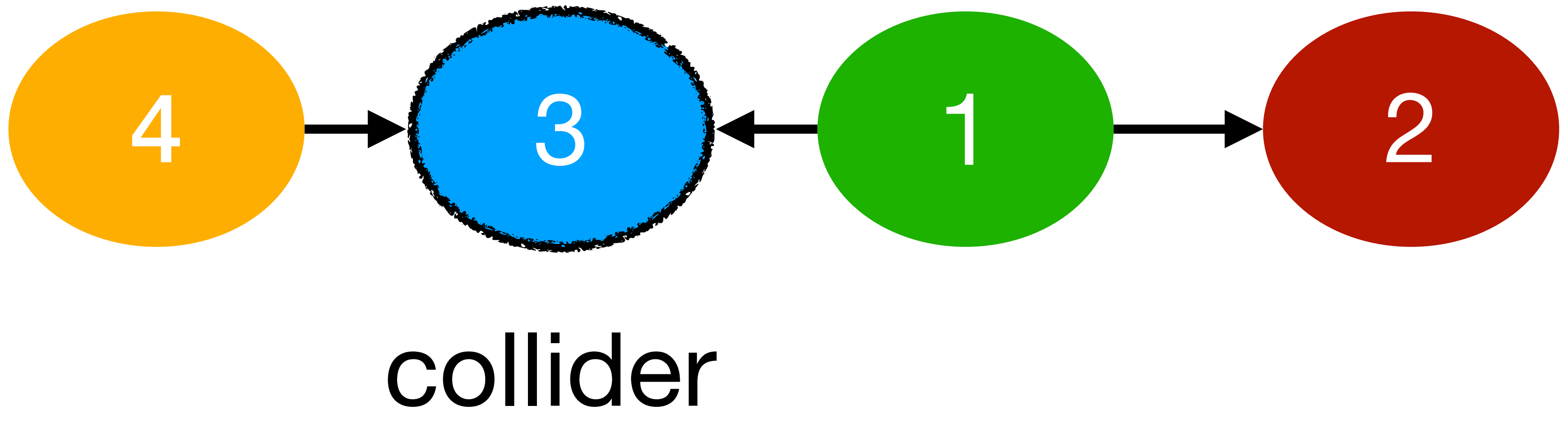


- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. the path π contains $\rightarrow k \leftarrow$

Graph terminology: collider on a path

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**

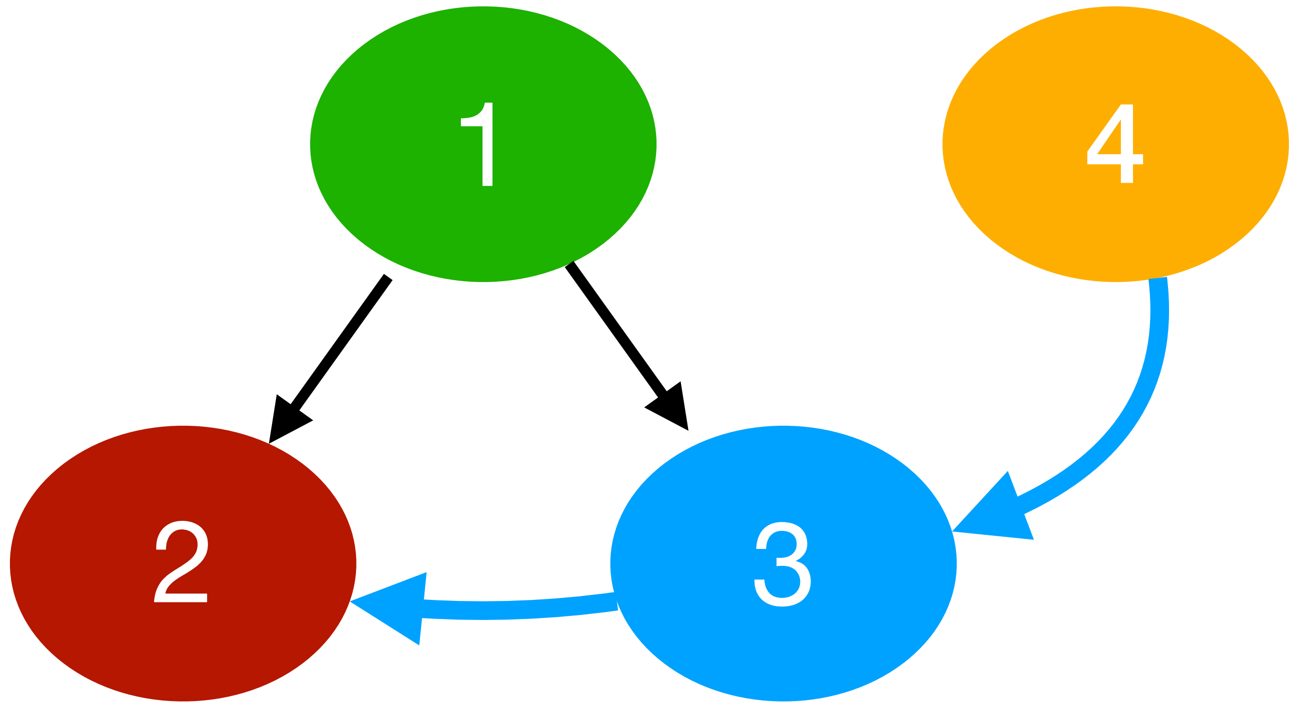




- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. the path π contains $\rightarrow k \leftarrow$

Graph terminology: collider on a path

- A **path** between **node i and node j** is a sequence of **distinct nodes** $(i, \ldots, j)$ such that each two **consecutive nodes** are **adjacent**



non-collider (chain)

- A **collider** k on a **path** $\pi = (i, \ldots, j)$ is a non-endpoint node $(k \neq i, j)$ s.t. the path π contains $\rightarrow k \leftarrow$

d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V}$ at least one condition holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \emptyset$

d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V}$ at least one condition holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \emptyset$
- Otherwise it is **active**

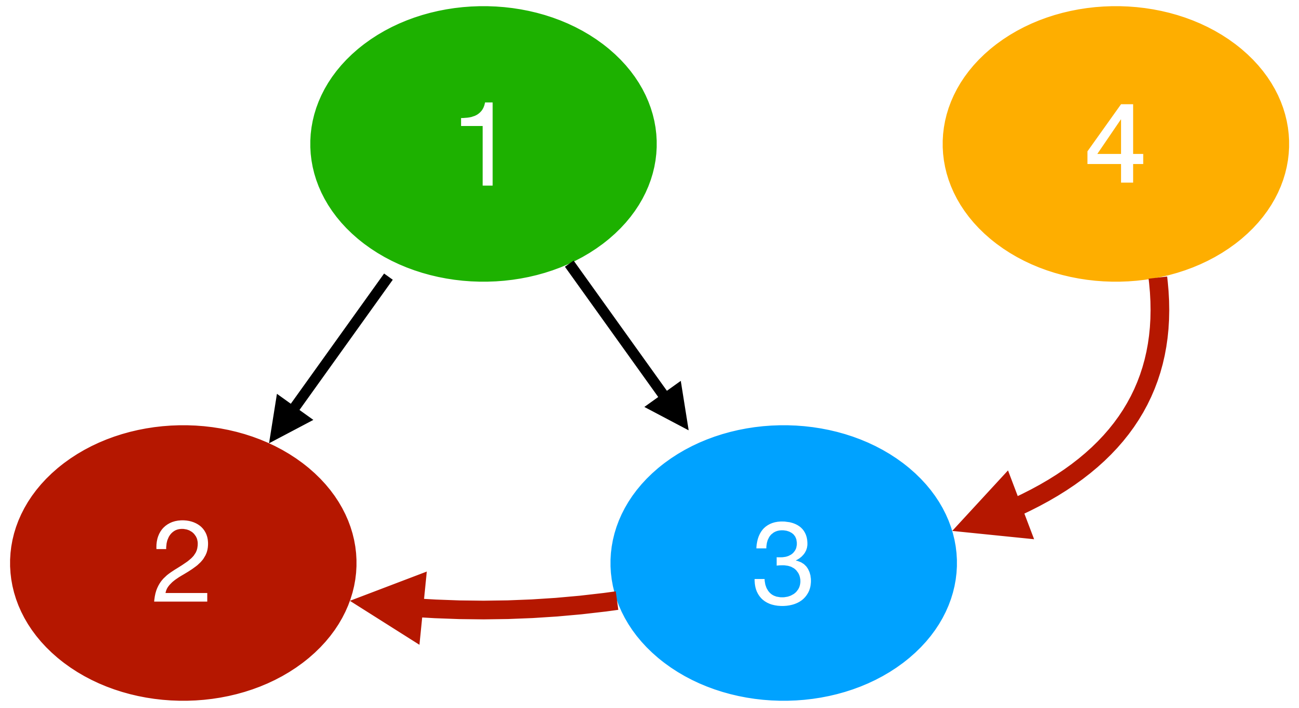
d-separation: blocked paths

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V}$ at least one condition holds:
  - There is a non-collider on the path that is in $\mathbf{A}$**, or**
  - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \emptyset$
- Otherwise it is **active**

Note: descendants w.r.t. the **whole graph**

d-separation: blocked paths - example 1

- A **path** between **i and j** is **blocked by $\mathbf{A} \subseteq \mathbf{V}$** at least one condition holds:
    - There is a non-collider on the path that is in $\mathbf{A}$**, or**
    - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \emptyset$
- Otherwise it is **active**

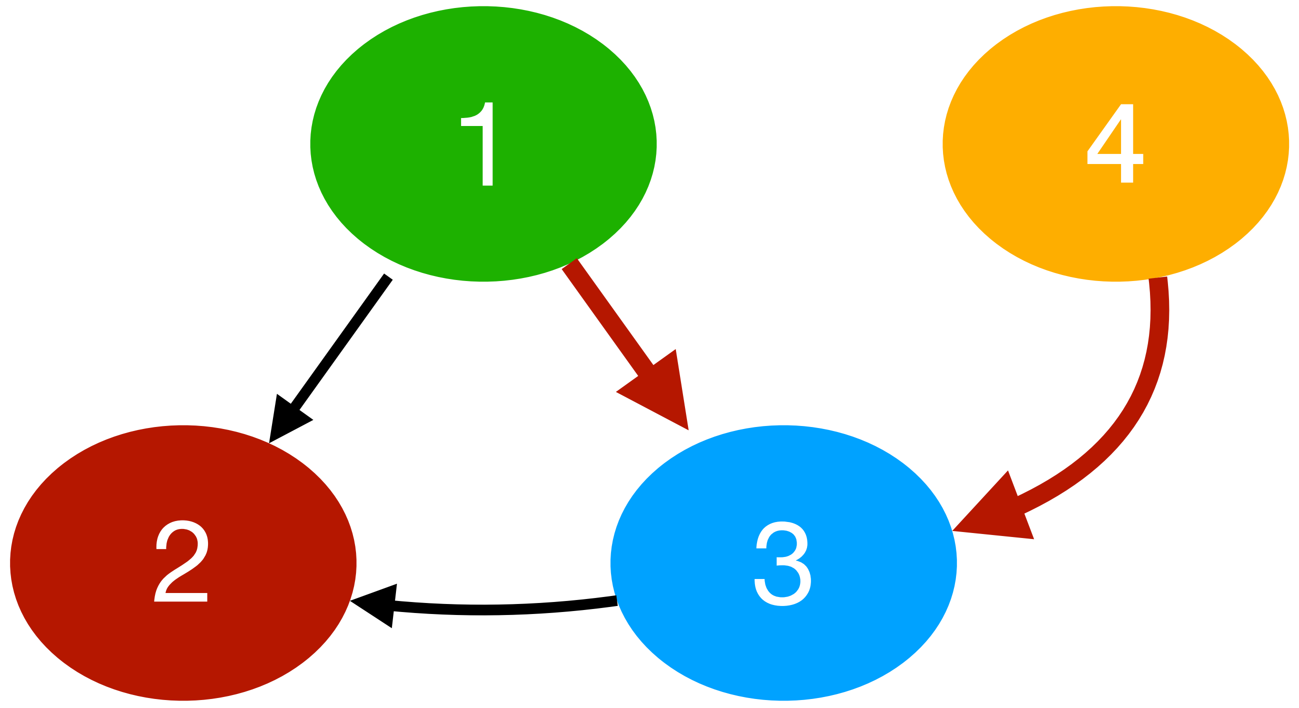


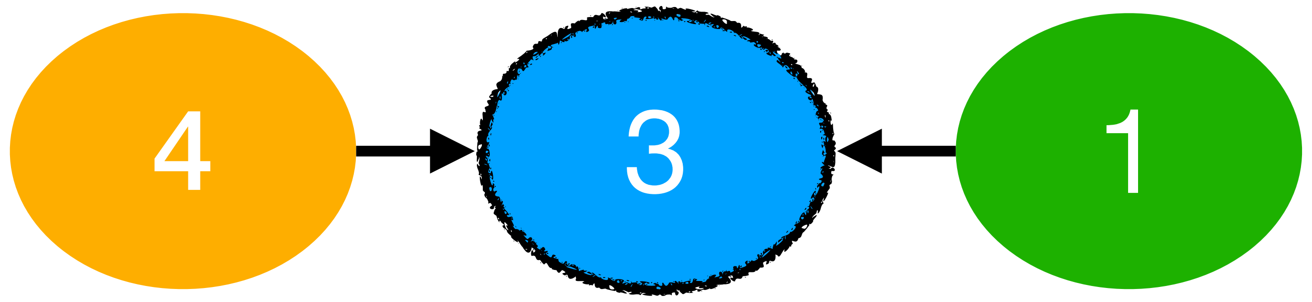
non-collider

If $3 \in \mathbf{A}$, the path is **blocked,** otherwise it is **active**

d-separation: blocked paths - example 2

- A **path** between **i and j** is **blocked by $\mathbf{A} \subseteq \mathbf{V}$** at least one condition holds:
  - There is a non-collider on the path that is in $\mathbf{A}$, **or**
  - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \emptyset$
- Otherwise it is **active**



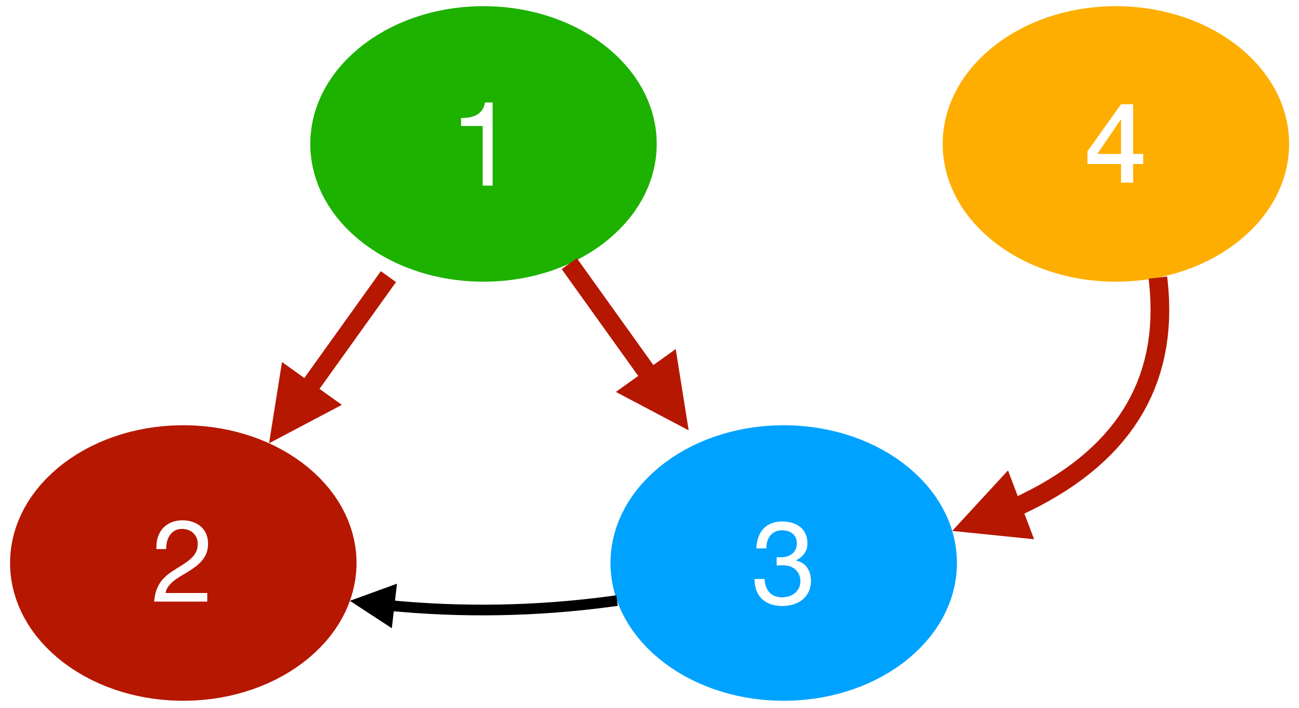


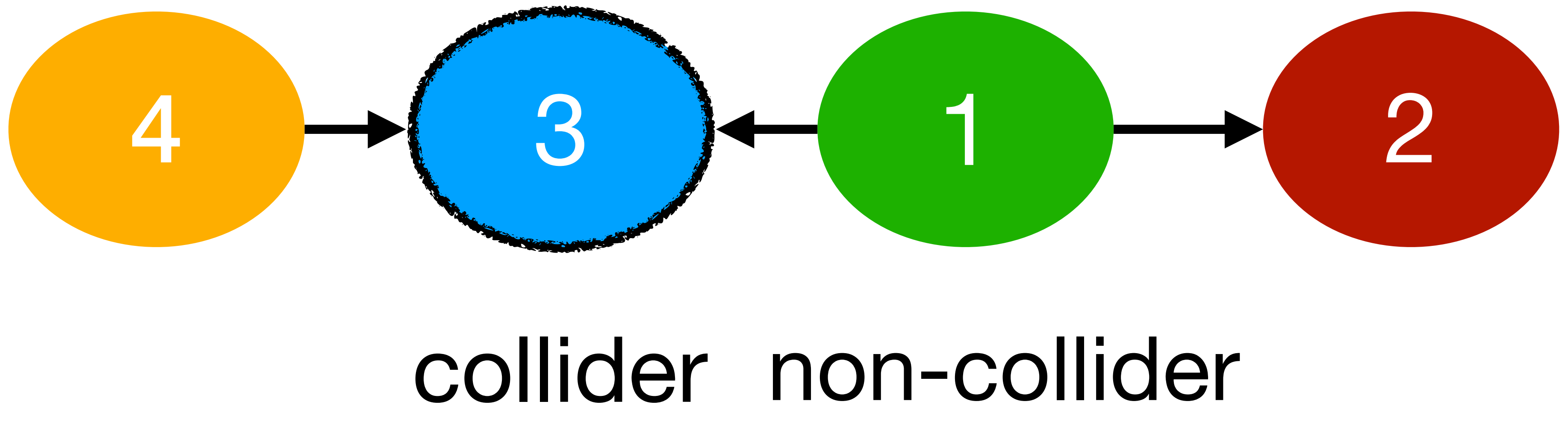
collider

If $3 \notin \mathbf{A}$ and $2 \notin \mathbf{A}$, the path is **blocked,** otherwise it is **active**

d-separation: blocked paths - example 3

- A **path** between **i and j** is **blocked by** $\mathbf{A} \subseteq \mathbf{V}$ at least one condition holds:
  - There is a non-collider on the path that is in $\mathbf{A}$, **or**
  - There is a collider k on the path, but $k \notin \mathbf{A}$ and $\mathrm{Desc}(k) \cap \mathbf{A} = \varnothing$
- Otherwise it is **active**





If $1 \in \mathbf{A}$, the path is **blocked**

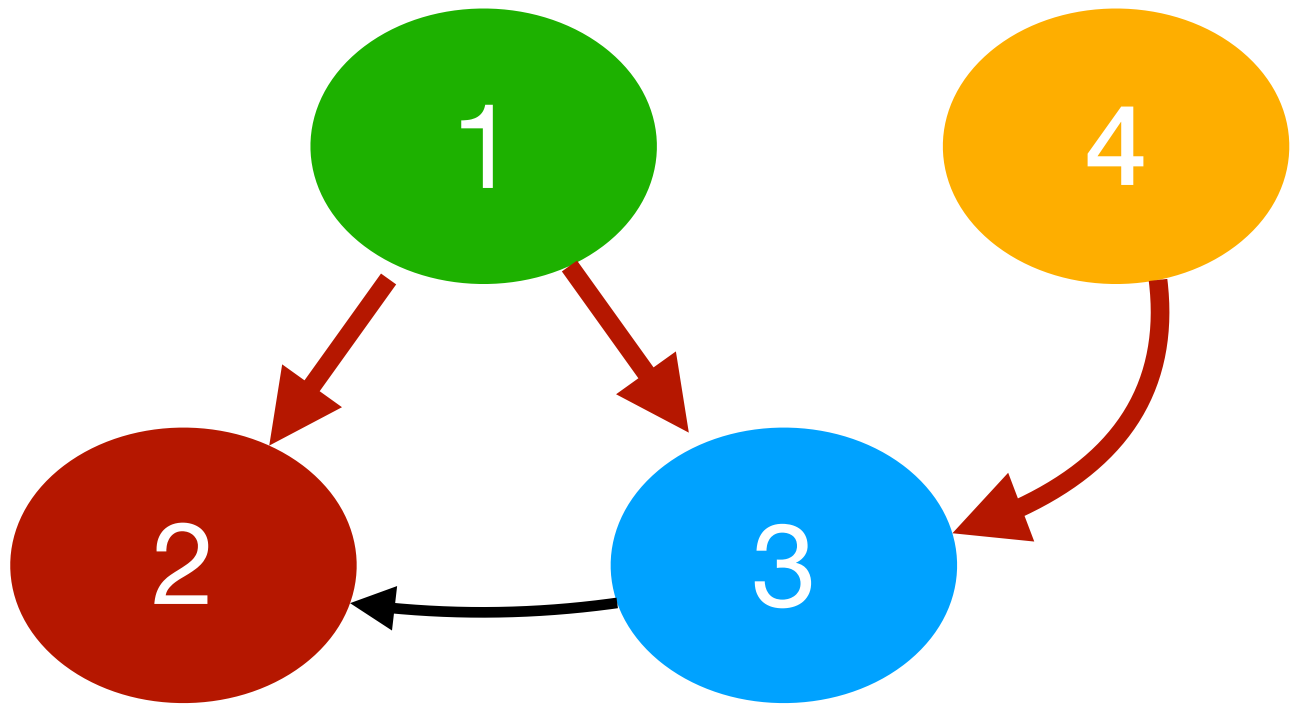
OR

If $3 \notin \mathbf{A}$ and $2 \notin \mathbf{A}$, the path is **blocked**

d-separation

- Nodes **i and j** is **d-separated by** $\mathbf{A} \subseteq \mathbf{V}$ if all paths between i, j are **blocked**
  - We denote d-separation as $i \perp j \,|\, \mathbf{A}$
- Otherwise we say they are **d-connected**
  - We denote d-connection as $i \not\perp j \,|\, \mathbf{A}$

Note: d-separation is **symmetric**



Global Markov Property and faithfulness

- If (G, P) is a Bayesian network with a DAG $G = (\mathbf{V}, \mathbf{E})$, i.e. **P factorizes according to G**, then for any disjoint $\mathbf{A}, \mathbf{B}, \mathbf{C} \subseteq \mathbf{V}$:

$$\mathbf{A} \perp \mathbf{B} \,|\, \mathbf{C} \implies X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

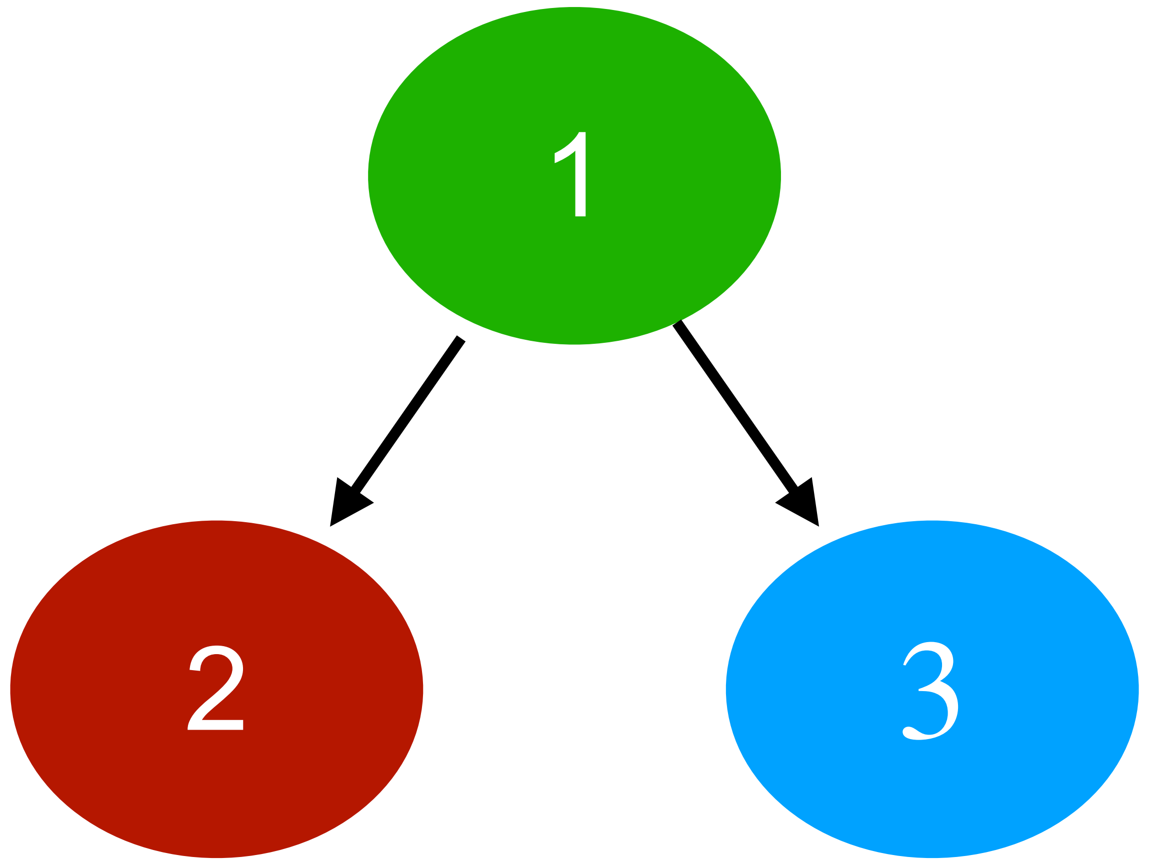
- **d-separations** that can be read purely from a graph imply **conditional independences** in the random variables and data generated by the graph

- The reverse implication is not true in general, but if it is **P is faithful to G**

$$\mathbf{A} \perp \mathbf{B} \,|\, \mathbf{C} \iff X_{\mathbf{A}} \perp\!\!\!\perp X_{\mathbf{B}} \,|\, X_{\mathbf{C}}$$

We usually assume both assumptions hold

Example Bayesian networks

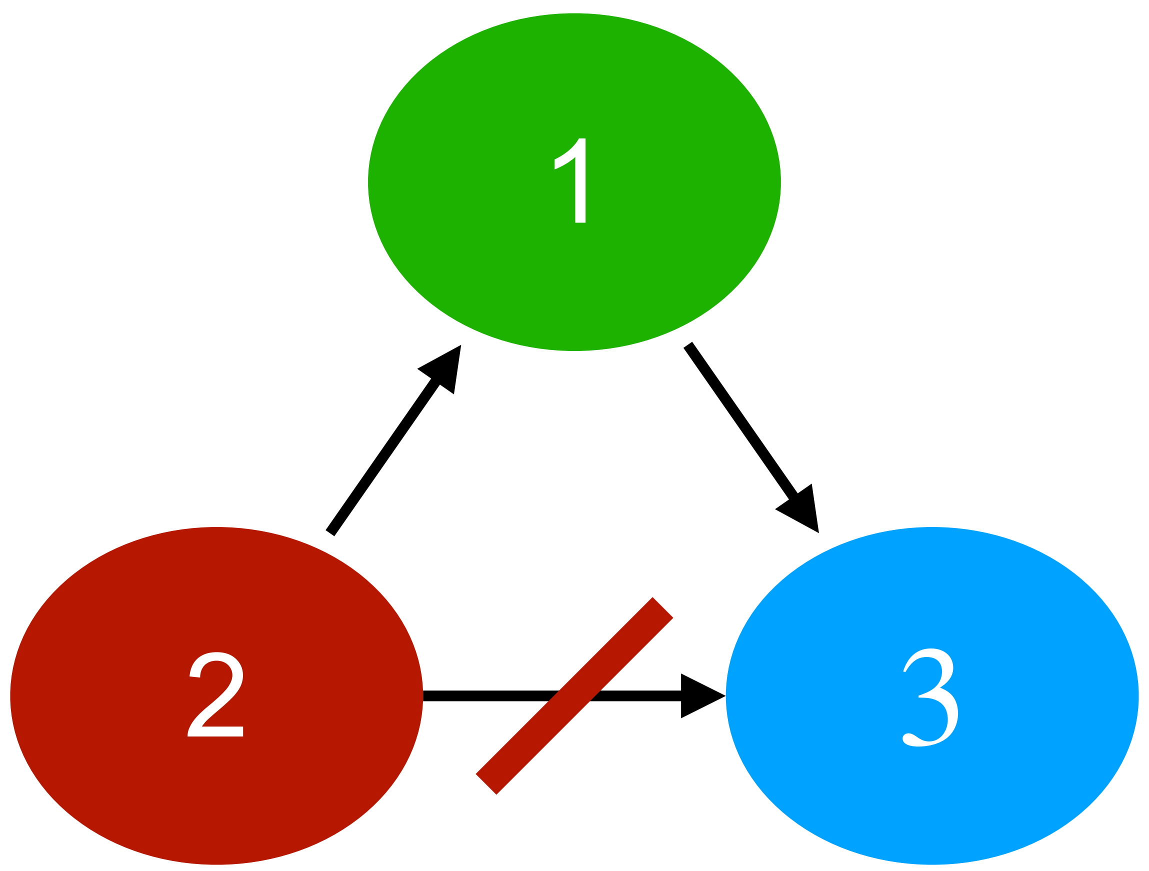


$$P(X_1, X_2, X_3) = P(X_1 \mid X_{\mathrm{Pa}(1)})P(X_2 \mid X_{\mathrm{Pa}(2)})P(X_3 \mid X_{\mathrm{Pa}(3)})$$

$$P(X_1) \quad P(X_2 \mid X_1) \quad P(X_3 \mid X_1)$$

The DAG/factorization is not unique for the joint distribution:

(for example any fully connected graph factorizes p)



$$P(X_1, X_2, X_3) = P(X_2)P(X_1 \mid X_2)P(X_3 \mid X_1, \cancel{X_2})$$

Since $X_3 \perp_d X_2 \mid X_1 \implies X_3 \perp\!\!\!\perp X_2 \mid X_1$