Topics

Relations

Relations, properties, operations, and applications.

Directed graphs

Directed graphs and representing relations as directed graphs.

Relations

Relations, properties, operations, and applications.

What are relations?

Binary relation

Let A and B be sets.

A *binary relation* from A to B is a subset of $A \times B$.

What are relations?

Binary relation

Let A and B be sets.

A *binary relation* from A to B is a subset of $A \times B$.

A binary relation R is a set of tuples. If R is a relation from a set A to itself, we say that R is a relation **on** A.

Examples of relations we've seen before

$\geq$ on $\mathbb{N}$

$$\{(x, y) : x \geq y \text{ and } x, y \in \mathbb{N}\}$$

$<$ on $\mathbb{R}$

$$\{(x, y) : x < y \text{ and } x, y \in \mathbb{R}\}$$

$=$ on Σ^*

$$\{(x, y) : x = y \text{ and } x, y \in \Sigma^*\}$$

$\subseteq$ on $\mathcal{P}(U)$ for a universe U

$$\{(A, B) : A \subseteq B \text{ and } A, B \in \mathcal{P}(U)\}$$

More examples of relations

$$R_1 = \{(a,1),(a,2),(b,1),(b,3),(c,3)\}$$

$$R_2 = \{(x,y) : x \equiv y \pmod 5\}$$

$$R_3 = \{(c_1,c_2) : c_1 \text{ is a prerequisite for } c_2\}$$

$$R_4 = \{(s,c) : \text{student } s \text{ has taken course } c\}$$

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$

$<$ on $\mathbb{R}$

$=$ on Σ^*

$\{(x, y) : x \equiv y \pmod 5\}$

$\{(1, 2), (2, 1), (1, 3)\}$

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$ reflexive, antisymmetric, transitive

$<$ on $\mathbb{R}$

$=$ on Σ^*

$\{(x, y) : x \equiv y \pmod 5\}$

$\{(1, 2), (2, 1), (1, 3)\}$

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$	reflexive, antisymmetric, transitive
$<$ on $\mathbb{R}$	antisymmetric, transitive
$=$ on Σ^*	
$\{(x, y) : x \equiv y \pmod 5\}$	
$\{(1, 2), (2, 1), (1, 3)\}$	

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$ — reflexive, antisymmetric, transitive

$<$ on $\mathbb{R}$ — antisymmetric, transitive

$=$ on Σ^* — reflexive, symmetric, antisymmetric, transitive

$\{(x, y) : x \equiv y \pmod 5\}$

$\{(1, 2), (2, 1), (1, 3)\}$

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$	reflexive, antisymmetric, transitive
$<$ on $\mathbb{R}$	antisymmetric, transitive
$=$ on Σ^*	reflexive, symmetric, antisymmetric, transitive
$\{(x, y) : x \equiv y \pmod 5\}$	reflexive, symmetric, transitive
$\{(1, 2), (2, 1), (1, 3)\}$	

Important properties of relations

Relation properties

Let R be a relation on A.

R is *reflexive* iff $(a, a) \in R$ for every $a \in A$.

R is *symmetric* iff $(a, b) \in R$ implies $(b, a) \in R$.

R is *antisymmetric* iff $(a, b) \in R$ and $a \neq b$ implies $(b, a) \notin R$.

R is *transitive* iff $(a, b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$.

Which relations have which properties?

$\geq$ on $\mathbb{N}$	reflexive, antisymmetric, transitive
$<$ on $\mathbb{R}$	antisymmetric, transitive
$=$ on Σ^*	reflexive, symmetric, antisymmetric, transitive
$\{(x, y) : x \equiv y \pmod 5\}$	reflexive, symmetric, transitive
$\{(1, 2), (2, 1), (1, 3)\}$	none!

Operations on relations

Relation composition

Let R be a relation on A to B.

Let S be a relation on B to C.

The *composition* of R and S, $R \circ S$, is the relation from A to C defined by:

$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$

Operations on relations

Relation composition

Let R be a relation on A to B.

Let S be a relation on B to C.

The *composition* of R and S, $R \circ S$, is the relation from A to C defined by:

$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$

Since relations are just sets, we can also combine them with the standard set theoretic operators ($\cup$, $\cap$, $\setminus$).

Examples of relational composition

$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$

Consider the relations Parent **and** Sister.

$(a, b) \in \mathrm{Parent}$ iff b is a parent of a

$(a, b) \in \mathrm{Sister}$ iff b is a sister of a

When is $(x, y) \in \mathrm{Parent} \circ \mathrm{Sister}$**?**

When is $(x, y) \in \mathrm{Sister} \circ \mathrm{Parent}$**?**

Examples of relational composition

$$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$$

Consider the relations Parent **and** Sister.

$(a, b) \in \mathrm{Parent}$ iff b is a parent of a

$(a, b) \in \mathrm{Sister}$ iff b is a sister of a

When is $(x, y) \in \mathrm{Parent} \circ \mathrm{Sister}$**?**

y is an aunt of x.

When is $(x, y) \in \mathrm{Sister} \circ \mathrm{Parent}$**?**

Examples of relational composition

$$R \circ S = \{(a, c) : \exists b.\, (a, b) \in R \text{ and } (b, c) \in S\}$$

Consider the relations Parent **and** Sister.

$(a, b) \in \mathrm{Parent}$ iff b is a parent of a

$(a, b) \in \mathrm{Sister}$ iff b is a sister of a

When is $(x, y) \in \mathrm{Parent} \circ \mathrm{Sister}$**?**

y is an aunt of x.

When is $(x, y) \in \mathrm{Sister} \circ \mathrm{Parent}$**?**

y is a parent of x and x has a sister.

More examples of relational composition

$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$

Use the relations $\text{Parent}, \text{Child}, \text{Sister}, \text{Brother}, \text{Sibling}$ **to express**

$(a, b) \in \text{Uncle}$ iff b is an uncle of a

$(a, b) \in \text{Cousin}$ iff b is a cousin of a

More examples of relational composition

$$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$$

Use the relations $\text{Parent}, \text{Child}, \text{Sister}, \text{Brother}, \text{Sibling}$ **to express**

$(a, b) \in \text{Uncle}$ iff b is an uncle of a $\quad\quad$ $\text{Parent} \circ \text{Brother}$

$(a, b) \in \text{Cousin}$ iff b is a cousin of a

More examples of relational composition

$R \circ S = \{(a, c) : \exists b. (a, b) \in R \text{ and } (b, c) \in S\}$

Use the relations $\text{Parent}, \text{Child}, \text{Sister}, \text{Brother}, \text{Sibling}$ **to express**

$(a, b) \in \text{Uncle}$ iff b is an uncle of a $\quad \text{Parent} \circ \text{Brother}$

$(a, b) \in \text{Cousin}$ iff b is a cousin of a $\quad \text{Parent} \circ \text{Sibling} \circ \text{Child}$

Powers of a relation

$R^0 = \{(a, a) : a \in A\}$

The *identity* relation on A.

$R^1 = R^0 \circ R = R$

$R^2 = R^1 \circ R = R \circ R = \{(a, c) : \exists b.\, (a, b) \in R \text{ and } (b, c) \in R\}$

$\vdots$

$R^{n+1} = R^n \circ R$ for $n \geq 0$

Powers of a relation

$R^0 = \{(a, a) : a \in A\}$

The *identity* relation on A.

$R^1 = R^0 \circ R = R$

$R^2 = R^1 \circ R = R \circ R = \{(a, c) : \exists b.\, (a, b) \in R \text{ and } (b, c) \in R\}$

$\vdots$

$R^{n+1} = R^n \circ R$ for $n \geq 0$

Reflexive transitive closure

Let R be a relation on A.

The *reflexive transitive closure*, R^*, of R is defined by $R^* = \bigcup_{n=0}^{\infty} R^n = R^0 \cup R^1 \cup R^2 \cup \ldots$

Applications of relations

Databases use relations to store and organize data.

Relations are also used to reason about software systems.

Applications of relations

Databases use relations to store and organize data.

So far, we've only talked about binary relations.

But the same concepts extend to *n-ary relations*, which contain tuples of length $n \geq 1$.

A (relational) database table is an n-ary relation, e.g., $R \subseteq \text{Student} \times \text{ID} \times \text{GPA}$.

Student	ID	GPA
Einstein	299792458	2.11
Newton	667408310	3.42
Turing	110101011	4.00

Relations are also used to reason about software systems.

Applications of relations

Databases use relations to store and organize data.

So far, we've only talked about binary relations.

But the same concepts extend to *n-ary relations*, which contain tuples of length $n \geq 1$.

A (relational) database table is an n-ary relation, e.g., $R \subseteq \text{Student} \times \text{ID} \times \text{GPA}$.

Student	ID	GPA
Einstein	299792458	2.11
Newton	667408310	3.42
Turing	110101011	4.00

Relations are also used to reason about software systems.

Relational logic (relations + predicate logic) is a language for specifying and automatically checking properties of software systems.

Many applications, including verifying safety critical software, synthesizing memory consistency models, and verifying security and privacy policies.

A filesystem spec

$\text{Root} = \{(\texttt{root})\}$

$\text{Root} \subseteq \text{Dir}$

$\text{contents} \subseteq \text{Dir} \times (\text{File} \cup \text{Dir})$

$(\text{File} \cup \text{Dir}) \subseteq \text{Root} \circ \text{contents}^*$

Directed graphs

Directed graphs and representing relations as directed graphs.

What is a directed graph?

Directed graphs

A *directed graph* $G = (V, E)$ consists of a set of *vertices* V and a set of *edges* $E \subseteq V \times V$, which are ordered pairs of vertices.

What is a directed graph?

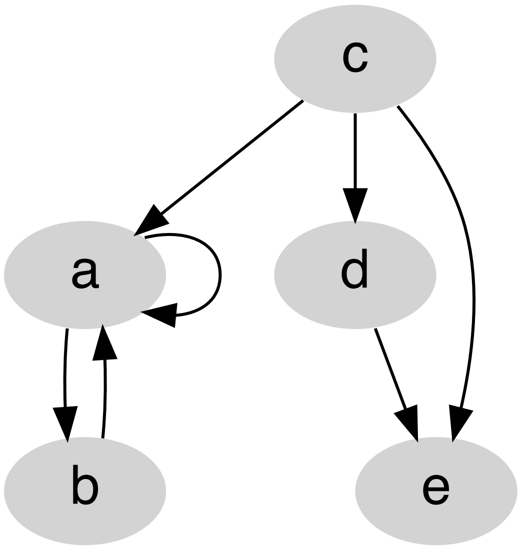
Directed graphs

A *directed graph* $G = (V, E)$ consists of a set of *vertices* V and a set of *edges* $E \subseteq V \times V$, which are ordered pairs of vertices.

Example directed graph $G = (V, E)$

$V = \{a, b, c, d, e\}$

$E = \{(a, a), (a, b), (b, a), (c, a), (c, d), (c, e), (d, e)\}$



What is a directed graph?

Directed graphs

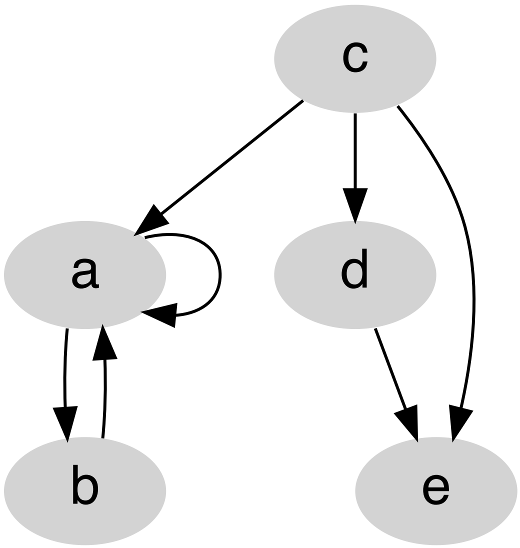
A *directed graph* $G = (V, E)$ consists of a set of *vertices* V and a set of *edges* $E \subseteq V \times V$, which are ordered pairs of vertices.

Example directed graph $G = (V, E)$

$V = \{a, b, c, d, e\}$

$E = \{(a, a), (a, b), (b, a), (c, a), (c, d), (c, e), (d, e)\}$

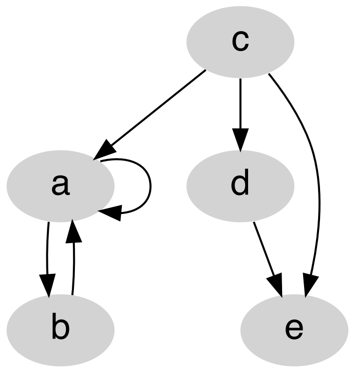
E is just a relation on V!



Representing relations as directed graphs

A relation R on A corresponds to the graph $G = (A, R)$.

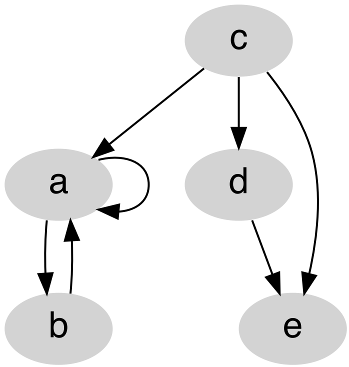
$$R = \{(a,a),(a,b),(b,a),(c,a),(c,d),(c,e),(d,e)\}$$
$$A = \{a,b,c,d,e\}$$



Representing relations as directed graphs

A relation R on A corresponds to the graph $G = (A, R)$.

$$R = \{(a,a),(a,b),(b,a),(c,a),(c,d),(c,e),(d,e)\}$$
$$A = \{a,b,c,d,e\}$$



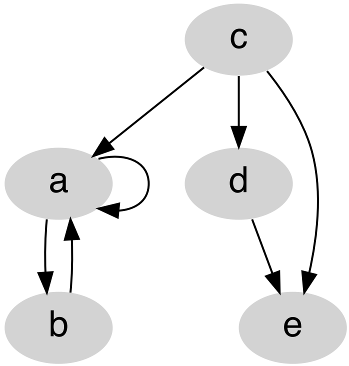
How about a relation R from A to B?

$$R = \{(a,1),(a,2),(b,1),(b,3),(c,3)\}$$
$$A = \{a,b,c\}$$
$$B = \{1,2,3\}$$

Representing relations as directed graphs

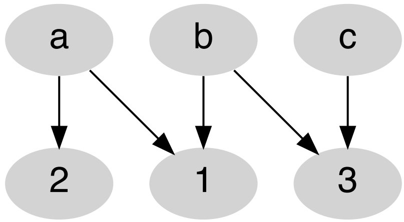
A relation R on A corresponds to the graph $G = (A, R)$.

$$R = \{(a, a), (a, b), (b, a), (c, a), (c, d), (c, e), (d, e)\}$$
$$A = \{a, b, c, d, e\}$$



How about a relation R from A to B?

$$R = \{(a, 1), (a, 2), (b, 1), (b, 3), (c, 3)\}$$
$$A = \{a, b, c\}$$
$$B = \{1, 2, 3\}$$
$$G = (A \cup B, R)$$



Paths in graphs

Path in a directed graph

Let $G = (V, E)$ be a directed graph.

A *path* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where $(v_i, v_{i+1}) \in E$ for each $0 \leq i < k$.

Paths in graphs

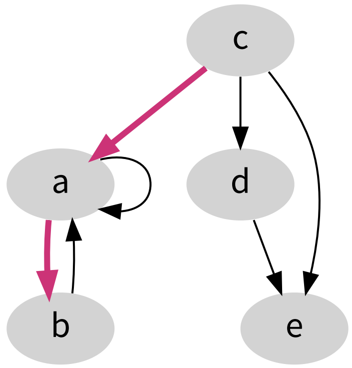
Path in a directed graph

Let $G = (V, E)$ be a directed graph.

A *path* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where $(v_i, v_{i+1}) \in E$ for each $0 \leq i < k$.

Example paths

Simple path (no v_i repeated): c, a, b



Paths in graphs

Path in a directed graph

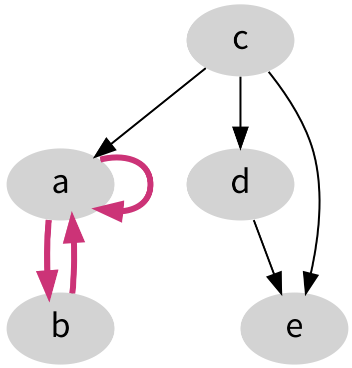
Let $G = (V, E)$ be a directed graph.

A *path* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where $(v_i, v_{i+1}) \in E$ for each $0 \leq i < k$.

Example paths

Simple path (no v_i repeated): c, a, b

Cycle ($v_0 = v_k$): b, a, a, a, b



Paths in graphs

Path in a directed graph

Let $G = (V, E)$ be a directed graph.

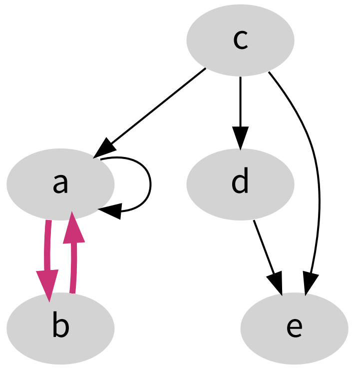
A *path* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where $(v_i, v_{i+1}) \in E$ for each $0 \leq i < k$.

Example paths

Simple path (no v_i repeated): c, a, b

Cycle ($v_0 = v_k$): b, a, a, a, b

Simple cycle ($v_0 = v_k$ and no other v_i repeated): b, a, b



Paths in graphs and relations

Path in a directed graph

Let $G = (V, E)$ be a directed graph.

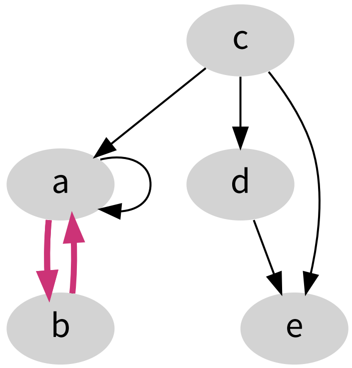
A *path* in G is a sequence of vertices $v_0, v_1, \ldots, v_k$ where $(v_i, v_{i+1}) \in E$ for each $0 \leq i < k$.

Example paths

Simple path (no v_i repeated): c, a, b

Cycle ($v_0 = v_k$): b, a, a, a, b

Simple cycle ($v_0 = v_k$ and no other v_i repeated): b, a, b



We've defined paths on graphs but the same definition applies to relations, since a relation and its graph representation are interchangeable.

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

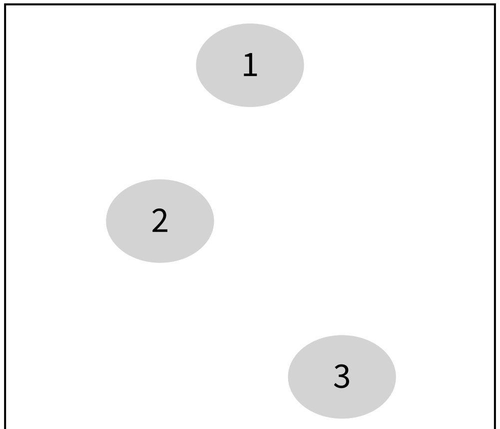
Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

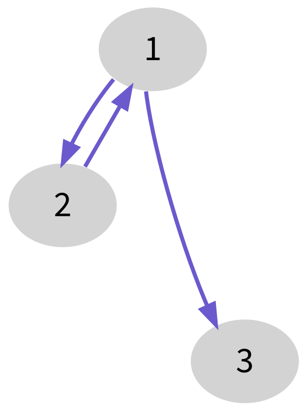
Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

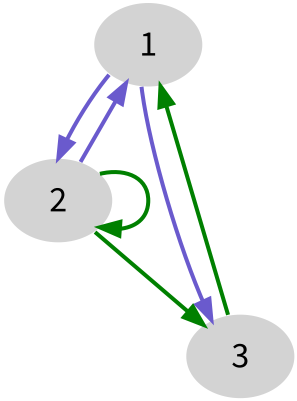
Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$

Relational composition using directed graphs

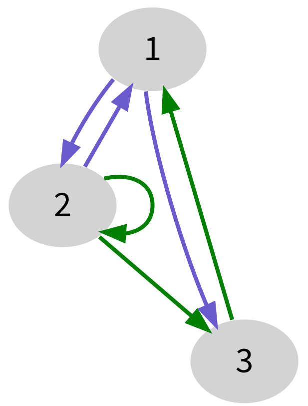
If R and S are relations on A, compute $R \circ S$ as follows:

Create the digraph $G = (A, R \cup S)$.

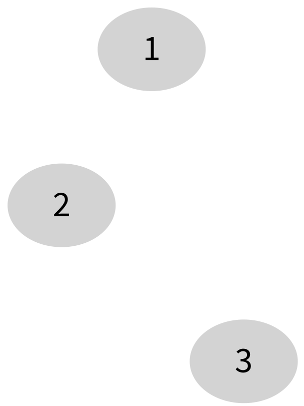
Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$
$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$



$R \circ S$

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

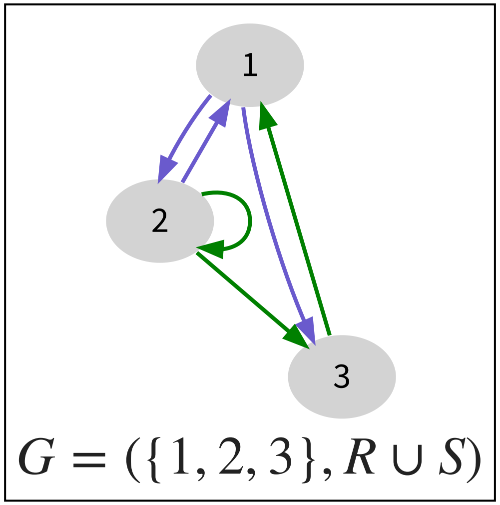
Create the digraph $G = (A, R \cup S)$.

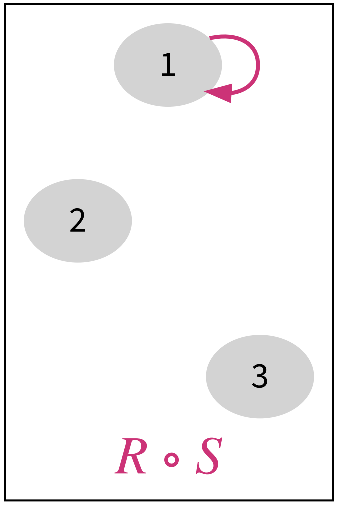
Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

Example:

$R = \{(1,2), (2,1), (1,3)\}$

$S = \{(2,2), (2,3), (3,1)\}$





Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

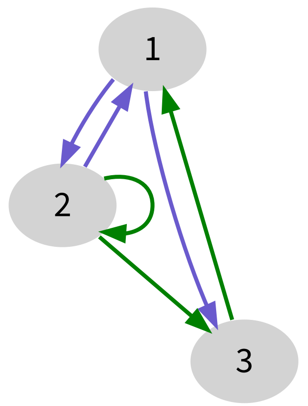
Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

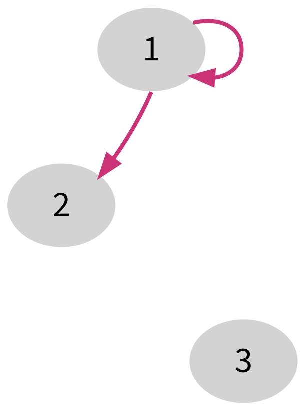
Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$



$R \circ S$

Relational composition using directed graphs

If R and S are relations on A, compute $R \circ S$ as follows:

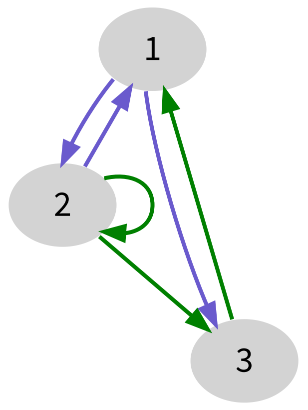
Create the digraph $G = (A, R \cup S)$.

Add an edge (a, b) to $R \circ S$ iff there is a path a, v, b in G such that $(a, v) \in R$ and $(v, b) \in S$.

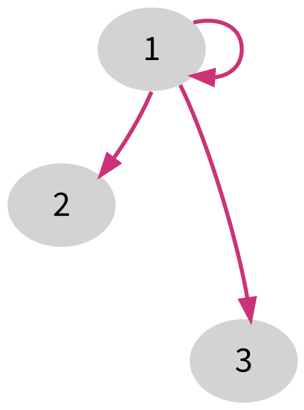
Example:

$R = \{(1, 2), (2, 1), (1, 3)\}$

$S = \{(2, 2), (2, 3), (3, 1)\}$



$G = (\{1, 2, 3\}, R \cup S)$



$R \circ S$

Powers of a relation using directed graphs

Length of a path in a graph

The *length* of a path $v_0, \ldots, v_k$ is the number of edges in it, i.e., k.

Powers and paths

Let R be a relation on a set A. There is a path of length n from a to b in R if and only if $(a, b) \in R^n$.

Reflexive transitive closure using directed graphs

Connectivity relation

Let R be a relation on a set A. The *connectivity* relation $R^\star$ consists of the tuples (a, b) such that there is a path (of any length) from a to b in R.

Connectivity and reflexive transitive closure

The reflexive transitive closure R^* of a relation R is its connectivity relation $R^\star$, i.e., $R^\star = R^* = \bigcup_{n=0}^{\infty} R^n$.

Summary

Relations are a fundamental structure in computer science.

- A relation is a set of tuples.
- Relations can be reflexive, (anti)symmetric, transitive.
- We can combine binary relations with the composition $\circ$ operator.

Directed graphs consist of nodes and edges (ordered pairs of nodes).

- Relations can be represented as directed graphs.
- The two representations are interchangeable: relational operations have corresponding graph operations and vice versa.