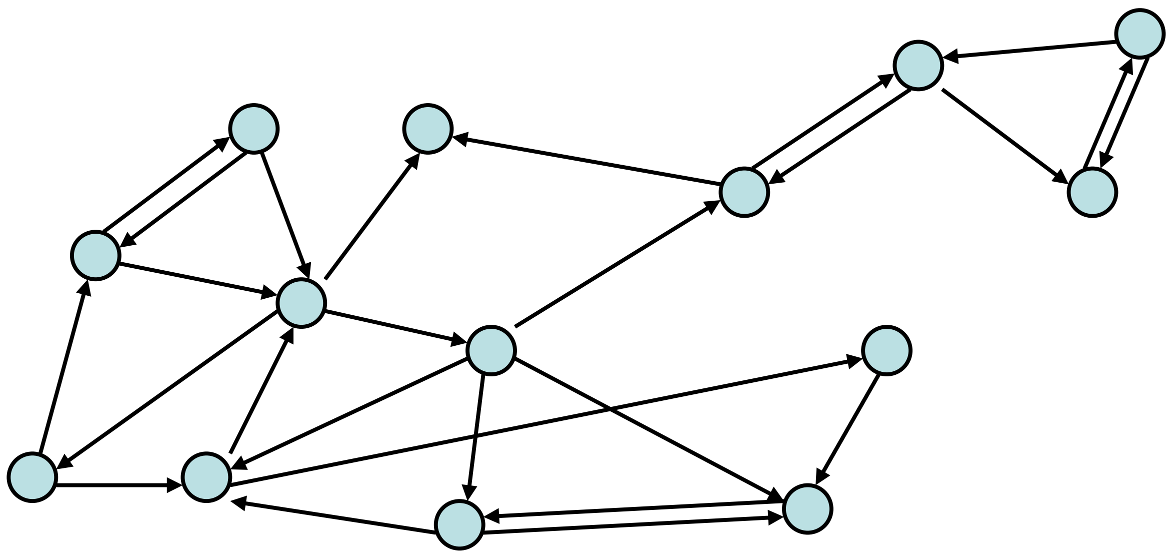
CSE 311: Foundations of Computing

Lecture 22: Relations and Directed Graphs



Last time: Languages — REs and CFGs

Saw two new ways of defining languages

- Regular Expressions $(0 \cup 1)^*\ 0110\ (0 \cup 1)^*$
  - easy to understand (declarative)
- Context-free Grammars $S \rightarrow SS \mid 0S1 \mid 1S0 \mid \varepsilon$
  - more expressive
  - (≈ recursively-defined sets)

We will connect these to machines shortly.

But first, we need some new math terminology....

Relations

Let A and B be sets,
A **binary relation from** A **to** B is a subset of $A \times B$

Let A be a set,
A **binary relation on** A is a subset of $A \times A$

Relations You Already Know

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

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

$<$ on $\mathbb{R}$

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

$=$ on Σ^*

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

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

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

More Relation Examples

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

$\mathbf{R_2}$ = $\{(x, y) \mid x \equiv_5 y\}$

$\mathbf{R_3}$ = $\{(c_1, c_2) \mid c_1$ is a prerequisite of $c_2\}$

$\mathbf{R_4}$ = {(s, c) | student s has taken course c }

Properties of Relations

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** $\sum^*$ **:**

$\subseteq$ **on** $\mathcal{P}$**(U):**

R$_2$ = $\{(x, y) \mid x \equiv_5 y\}$:

R$_3$ = $\{(c_1, c_2) \mid c_1$ is a prerequisite of $c_2\}$:

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

$\subseteq$ on $\mathcal{P}(U)$: Reflexive, Antisymmetric, Transitive

$R_2 = \{(x, y) \mid x \equiv_5 y\}$: Reflexive, Symmetric, Transitive

$R_3 = \{(c_1, c_2) \mid c_1$ is a prerequisite of $c_2\}$: Antisymmetric

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 $(a, c) \in R$

Combining Relations

Let R be a relation from A to B.
Let S be a relation from 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 such that (a,b)$\in$ R and (b,c)$\in$ $S\}$

Intuitively, a pair is in the composition if there is a "connection" from the first to the second.

Examples

$(a,b) \in$ **Parent** iff b is a parent of a

$(a,b) \in$ **Sister** iff b is a sister of a

When is $(x,y) \in$ Parent $\circ$ Sister?

When is $(x,y) \in$ Sister $\circ$ Parent?

$R \circ S = \{(a, c) \mid \exists\, b$ such that $(a,b) \in R$ and $(b,c) \in S\}$

Examples

Using only the relations Parent, Child, Father, Son, Brother, Sibling, Husband

and *composition*, express the following:

Uncle: b is an uncle of a

Cousin: b is a cousin of a

Powers of a Relation

$$\begin{aligned} R^2 &= R \circ R \\ &= \{(a, c) \mid \exists b \text{ such that } (a, b) \in R \text{ and } (b, c) \in R\} \end{aligned}$$

$R^0 = \{(a, a) \mid a \in A\}$ "the equality relation on A"

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

e.g., $R^1 = R^0 \circ R = R$

$R^2 = R^1 \circ R = R \circ R$

Non-constructive Definitions

Recursively defined sets and functions describe these objects by explaining how to construct / compute them

But sets can also be defined non-constructively:

$$\boxed{S = \{x : P(x)\}}$$

How can we define functions non-constructively?

– (useful for writing a function specification)

Functions

A function $f : A \rightarrow B$ (A as input and B as output) is a special type of relation.

A **function** f **from** A **to** B is a relation from A to B such that: for every $a \in A$, there is *exactly one* $b \in B$ with $(a, b) \in f$

I.e., for every input $a \in A$, there is one output $b \in B$.
We denote this b by $f(a)$.

(When attempting to define a function this way, we sometimes say the function is "well defined" if the *exactly one* part holds)

Functions

A function $f : A \to B$ (A as input and B as output) is a special type of relation.

A **function** f **from** A **to** B is a relation from A to B such that: for every $a \in A$, there is *exactly one* $b \in B$ with $(a, b) \in f$

Ex: {((a, b), d) : d is the largest integer dividing a and b}

- **gcd :** $\mathbb{N} \times \mathbb{N} \to \mathbb{N}$
- **defined without knowing how to compute it**

Matrix Representation

Relation $\boldsymbol{R}$ on $\boldsymbol{A} = \{a_1, \dots, a_p\}$

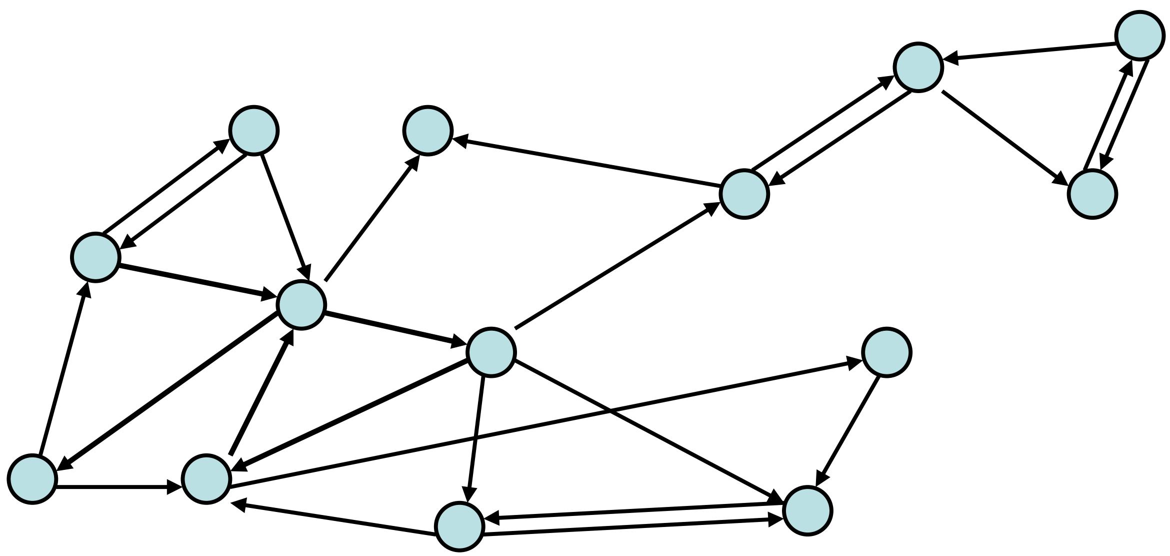
$$\boldsymbol{m_{ij}} = \begin{cases} 1 & \text{if } (a_i, a_j) \in \boldsymbol{R} \\ 0 & \text{if } (a_i, a_j) \notin \boldsymbol{R} \end{cases}$$

{ (1, 1), (1, 2), (1, 4), (2, 1), (2, 3), (3, 2), (3, 3), (4, 2), (4, 3) }

	1	2	3	4
1	1	1	0	1
2	1	0	1	0
3	0	1	1	0
4	0	1	1	0

Directed Graphs

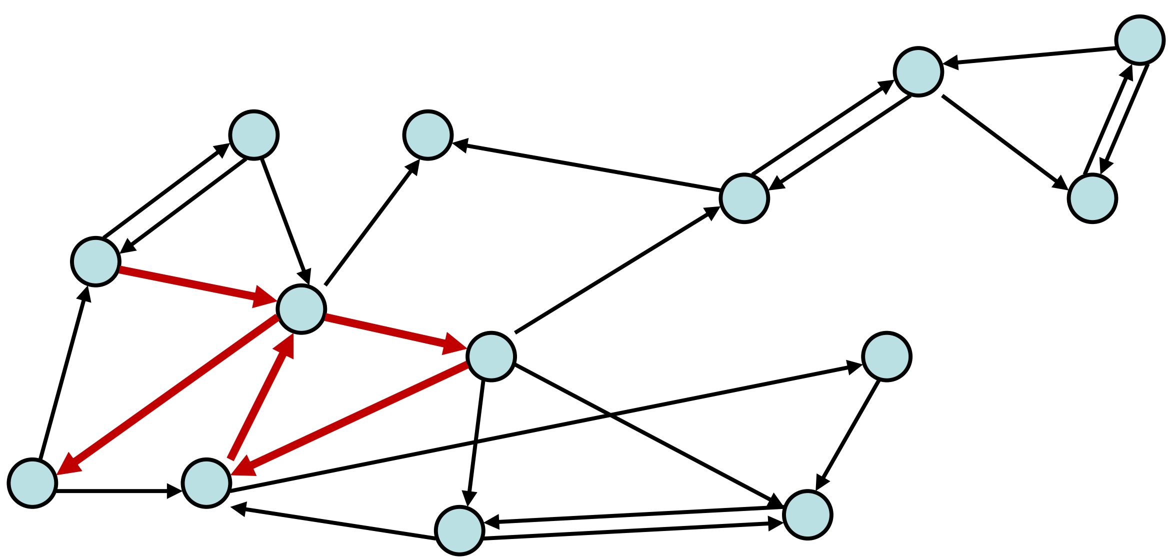
G = (V, E)　　V – vertices
　　　　　　E – edges　　(relation on vertices)



Directed Graphs

G = (V, E) V – vertices
E – edges (relation on vertices)

Path: $v_0, v_1, \ldots, v_k$ with each (v_i, v_{i+1}) in E

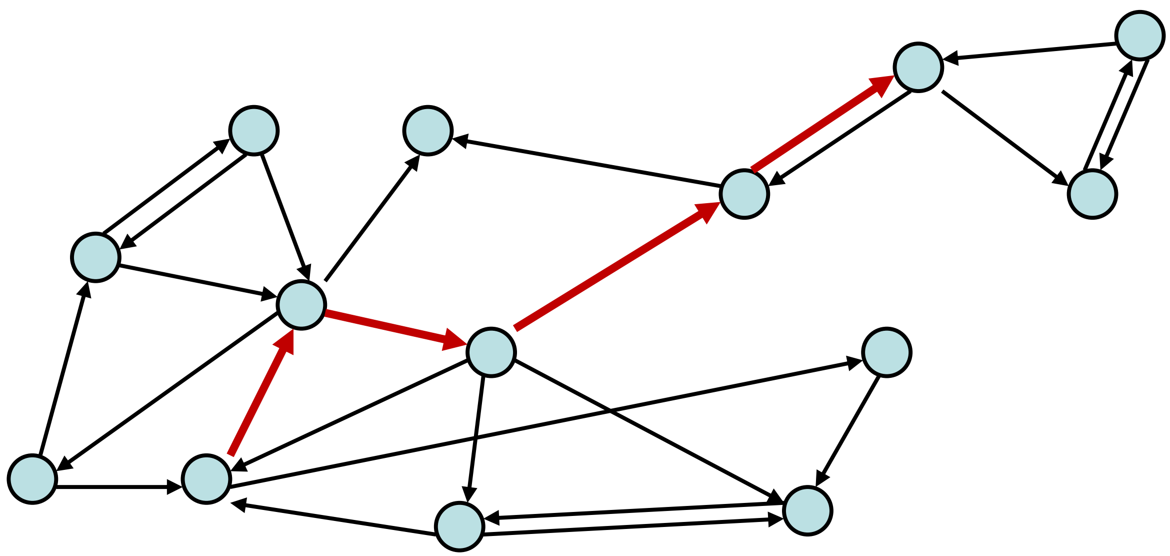


Directed Graphs

$G = (V, E)$ V – vertices
E – edges (relation on vertices)

Path: $v_0, v_1, \ldots, v_k$ with each (v_i, v_{i+1}) in E

Simple Path: none of $\mathbf{v_0}, \ldots, \mathbf{v_k}$ repeated
Cycle: $\mathbf{v_0} = \mathbf{v_k}$
Simple Cycle: $\mathbf{v_0} = \mathbf{v_k}$, none of $\mathbf{v_1}, \ldots, \mathbf{v_k}$ repeated

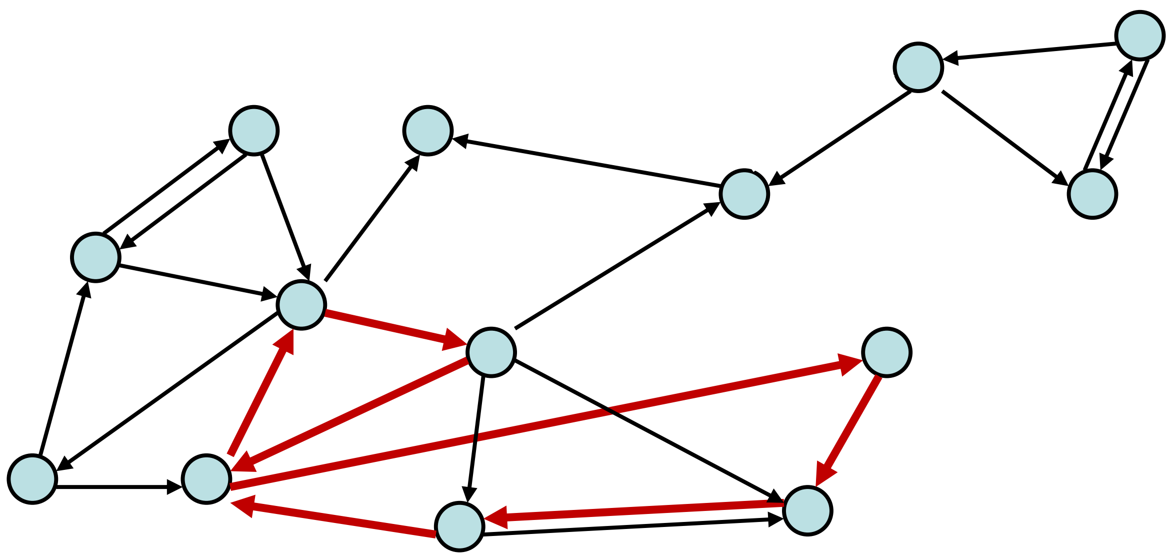


Directed Graphs

$G = (V, E)$ V – vertices
E – edges (relation on vertices)

Path: $v_0, v_1, \ldots, v_k$ with each (v_i, v_{i+1}) in E

Simple Path: none of $\mathbf{v_0}, \ldots, \mathbf{v_k}$ repeated
Cycle: $\mathbf{v_0} = \mathbf{v_k}$
Simple Cycle: $\mathbf{v_0} = \mathbf{v_k}$, none of $\mathbf{v_1}, \ldots, \mathbf{v_k}$ repeated

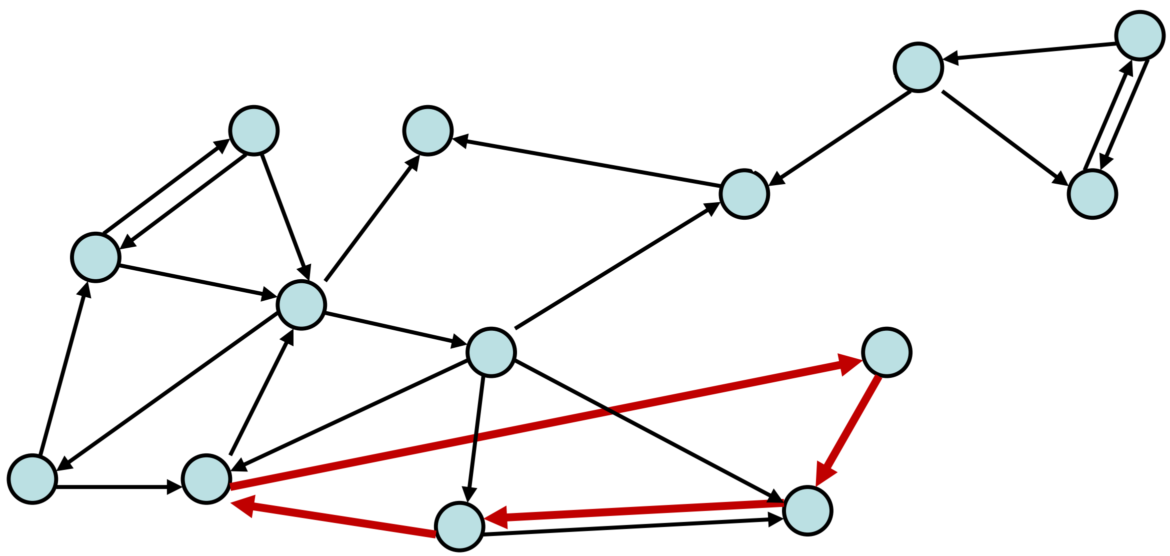


Directed Graphs

$G = (V, E)$ V – vertices
E – edges (relation on vertices)

Path: $v_0, v_1, \ldots, v_k$ with each (v_i, v_{i+1}) in E

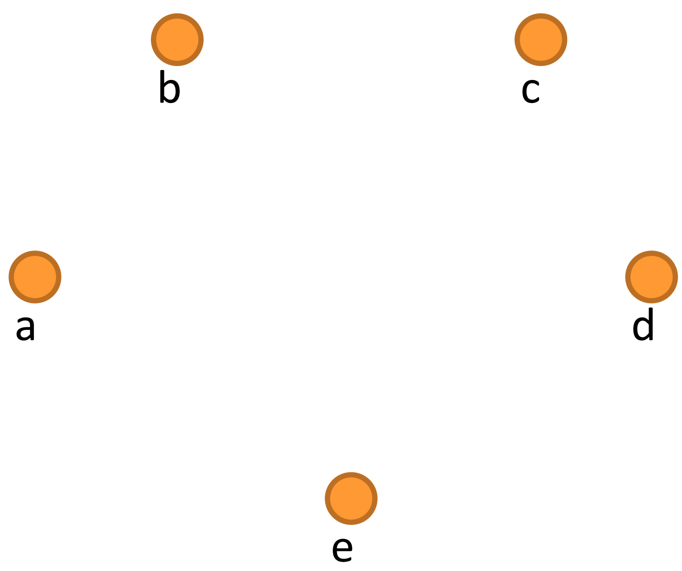
Simple Path: none of $\mathbf{v_0}, \ldots, \mathbf{v_k}$ repeated
Cycle: $\mathbf{v_0} = \mathbf{v_k}$
Simple Cycle: $\mathbf{v_0} = \mathbf{v_k}$, none of $\mathbf{v_1}, \ldots, \mathbf{v_k}$ repeated



Representation of Relations

Directed Graph Representation (Digraph)

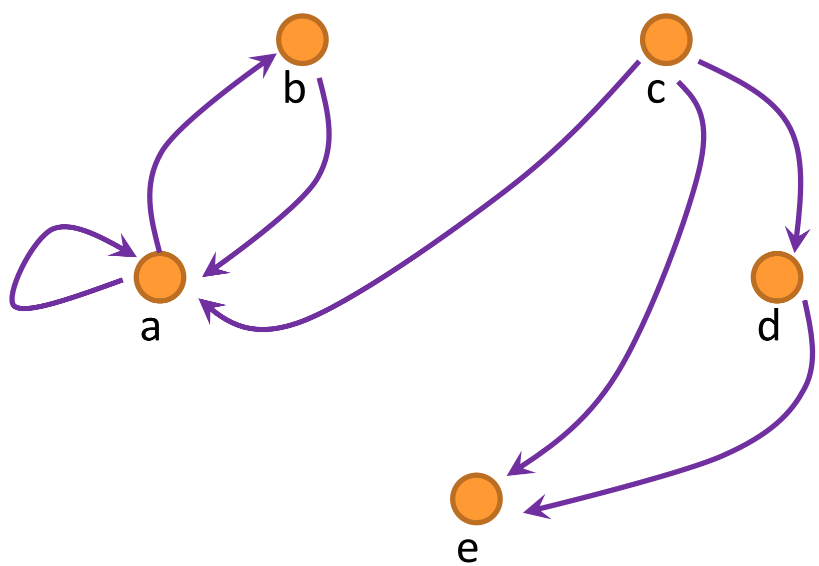
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }



Representation of Relations

Directed Graph Representation (Digraph)

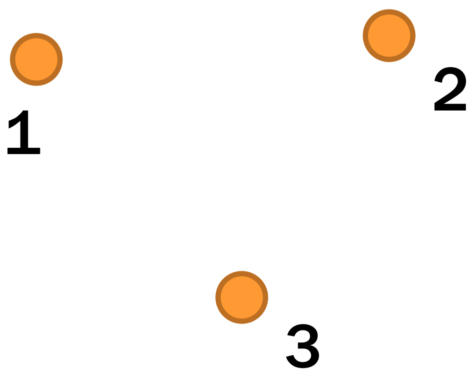
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }

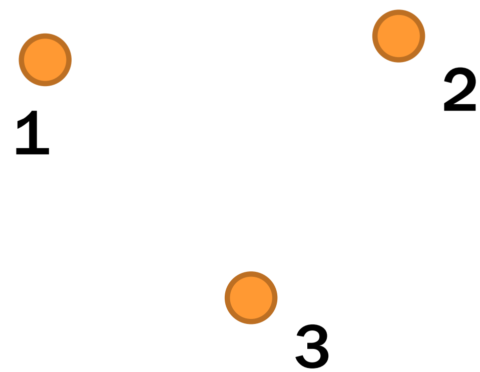


Relational Composition using Digraphs

If $\boldsymbol{S} = \{(\mathbf{2}, \mathbf{2}), (\mathbf{2}, \mathbf{3}), (\mathbf{3}, \mathbf{1})\}$ and $\boldsymbol{R} = \{(\mathbf{1}, \mathbf{2}), (\mathbf{2}, \mathbf{1}), (\mathbf{1}, \mathbf{3})\}$

Compute $\boldsymbol{R} \circ \boldsymbol{S}$

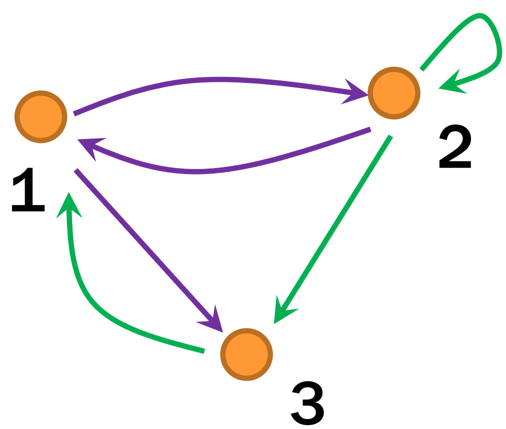


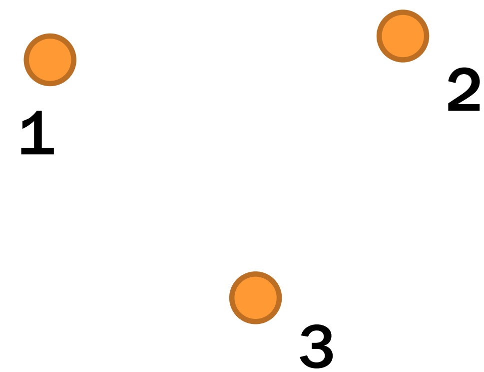


Relational Composition using Digraphs

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

Compute $R \circ S$

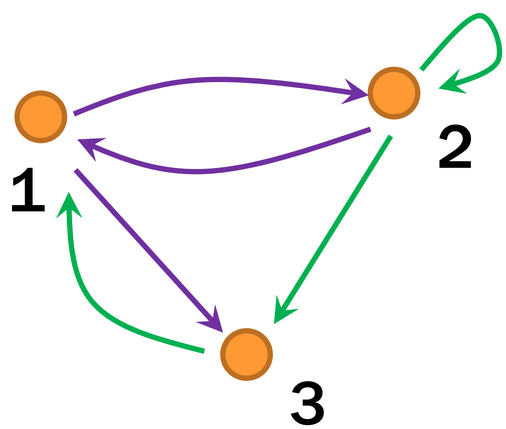


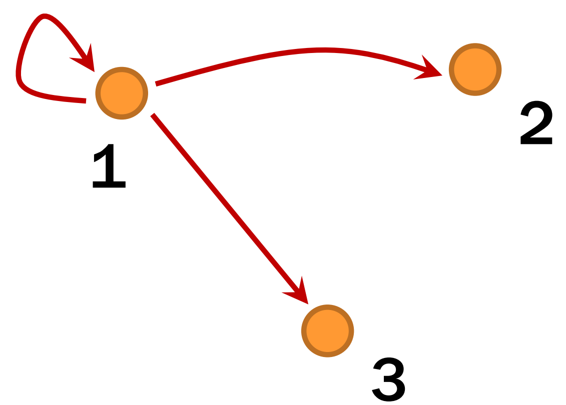


Relational Composition using Digraphs

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

Compute $R \circ S$





Special case: $R \circ R$ is paths of length 2.

- R is paths of length 1
- R^0 is paths of length 0 (can't go anywhere)
- $R^3 = R^2 \circ R$ etc, so is R^n paths of length n

Paths in Relations and Graphs

Def: The **length** of a path in a graph is the number of edges in it (counting repetitions if edge used > once).

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

Connectivity In Graphs

Def: Two vertices in a graph are **connected** iff there is a path between them.

Let $\boldsymbol{R}$ be a relation on a set $\boldsymbol{A}$. The **connectivity** relation $\boldsymbol{R}^*$ consists of the pairs (a, b) such that there is a path from a to b in $\boldsymbol{R}$.

$$R^* = \bigcup_{k=0}^{\infty} R^k$$

Note: The text uses the wrong definition of this quantity. What the text defines (ignoring k=0) is usually called R^+

How Properties of Relations show up in Graphs

Let R be a relation on A.

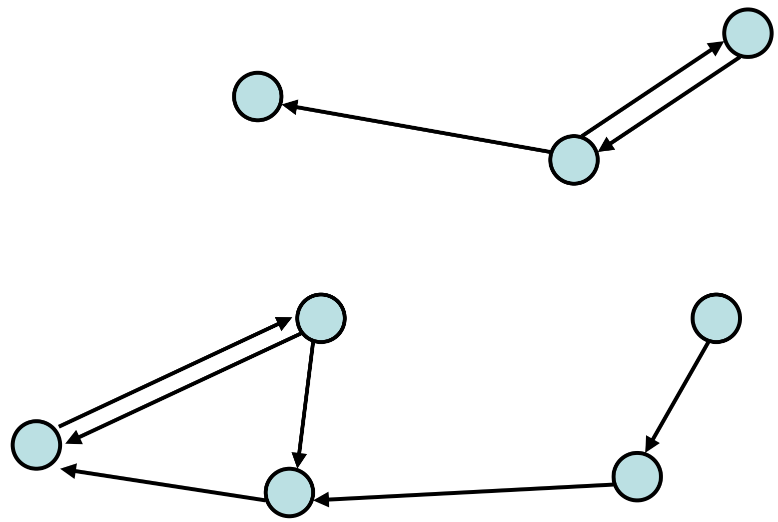
R is **reflexive** iff $(a,a) \in R$ for every $a \in A$

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 $(b,a) \notin R$

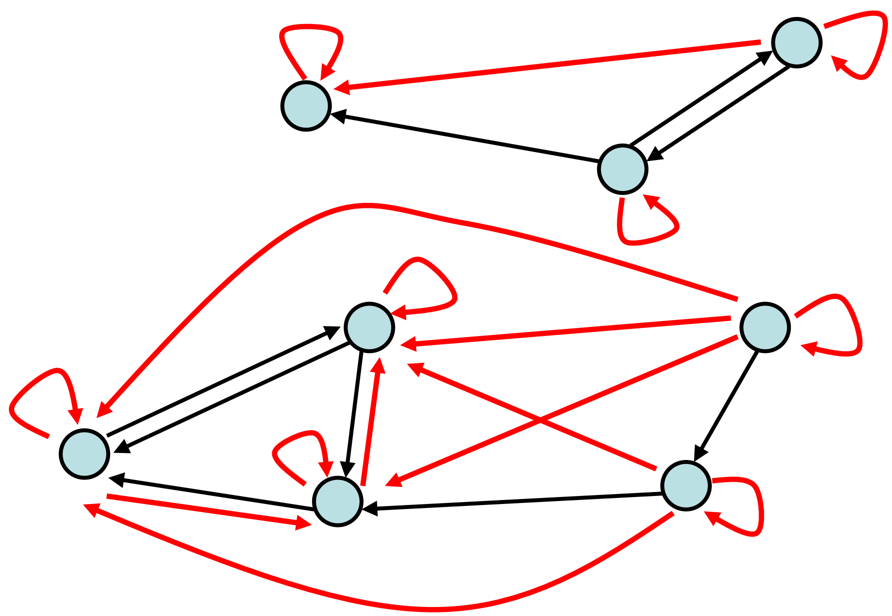
R is **transitive** iff $(a,b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$

Transitive-Reflexive Closure



Add the **minimum possible** number of edges to make the relation transitive and reflexive.

Transitive-Reflexive Closure



Relation with the **minimum possible** number of **extra edges** to make the relation both transitive and reflexive.

The **transitive-reflexive closure** of a relation $\boldsymbol{R}$ is the connectivity relation $\boldsymbol{R}^*$

n-ary Relations

Let $\boldsymbol{A_1, A_2, \ldots, An}$ be sets. An $\boldsymbol{n}$**-ary** relation on these sets is a subset of $\boldsymbol{A_1 \times A_2 \times \cdots \times A_n}$.

Relational Databases

STUDENT

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Von Neuman	481080220	555	3.78
Russell	238082388	022	3.85
Einstein	238001920	022	2.11
Newton	1727017	333	3.61
Karp	348882811	022	3.98
Bernoulli	2921938	022	3.21

Database Operations: Projection

Find all offices: $\Pi_{\text{Office}}(\text{STUDENT})$

Office
022
555
333

Find offices and GPAs: $\Pi_{\text{Office,GPA}}(\text{STUDENT})$

Office	GPA
022	4.00
555	3.78
022	3.85
022	2.11
333	3.61
022	3.98
022	3.21

Database Operations: Selection

Find students with GPA > 3.9 : $\boldsymbol{\sigma}_{\mathbf{GPA>3.9}}$(STUDENT)

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Karp	348882811	022	3.98

Retrieve the name and GPA for students with GPA > 3.9:

$\boldsymbol{\Pi}_{\mathbf{Student\_Name,GPA}}(\boldsymbol{\sigma}_{\mathbf{GPA>3.9}}$(STUDENT))

Student_Name	GPA
Knuth	4.00
Karp	3.98

Relational Databases

STUDENT

Student_Name	ID_Number	Office	GPA	Course
Knuth	328012098	022	4.00	CSE311
Knuth	328012098	022	4.00	CSE351
Von Neuman	481080220	555	3.78	CSE311
Russell	238082388	022	3.85	CSE312
Russell	238082388	022	3.85	CSE344
Russell	238082388	022	3.85	CSE351
Newton	1727017	333	3.61	CSE312
Karp	348882811	022	3.98	CSE311
Karp	348882811	022	3.98	CSE312
Karp	348882811	022	3.98	CSE344
Karp	348882811	022	3.98	CSE351
Bernoulli	2921938	022	3.21	CSE351

What's not so nice?

Relational Databases

STUDENT

Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Von Neuman	481080220	555	3.78
Russell	238082388	022	3.85
Einstein	238001920	022	2.11
Newton	1727017	333	3.61
Karp	348882811	022	3.98
Bernoulli	2921938	022	3.21

TAKES

ID_Number	Course
328012098	CSE311
328012098	CSE351
481080220	CSE311
238082388	CSE312
238082388	CSE344
238082388	CSE351
1727017	CSE312
348882811	CSE311
348882811	CSE312
348882811	CSE344
348882811	CSE351
2921938	CSE351

Better

Database Operations: Natural Join

Student $\bowtie$ Takes

Student_Name	ID_Number	Office	GPA	Course
Knuth	328012098	022	4.00	CSE311
Knuth	328012098	022	4.00	CSE351
Von Neuman	481080220	555	3.78	CSE311
Russell	238082388	022	3.85	CSE312
Russell	238082388	022	3.85	CSE344
Russell	238082388	022	3.85	CSE351
Newton	1727017	333	3.61	CSE312
Karp	348882811	022	3.98	CSE311
Karp	348882811	022	3.98	CSE312
Karp	348882811	022	3.98	CSE344
Karp	348882811	022	3.98	CSE351
Bernoulli	2921938	022	3.21	CSE351