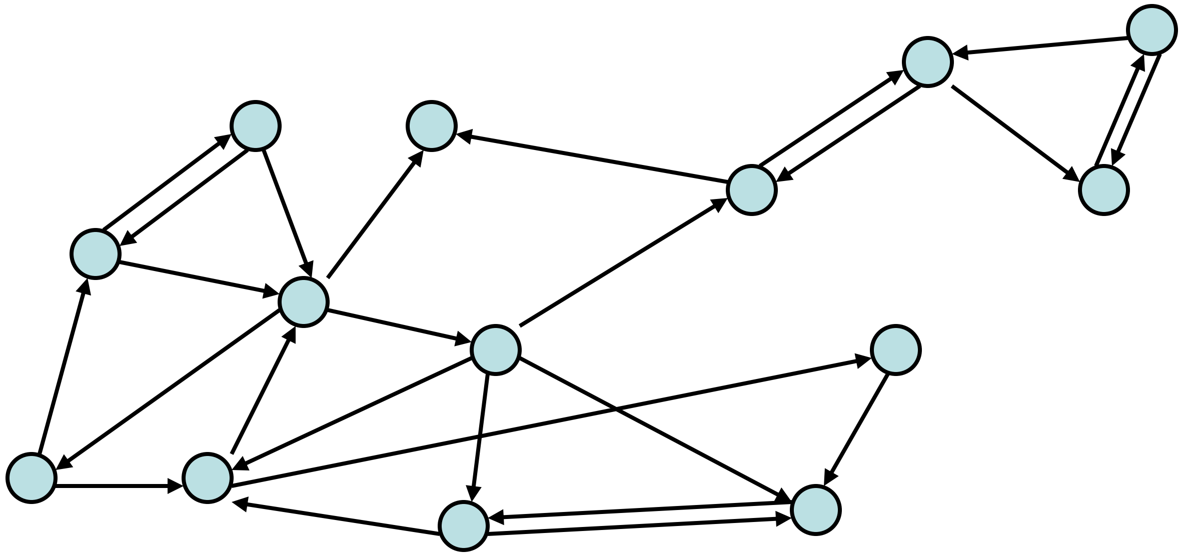
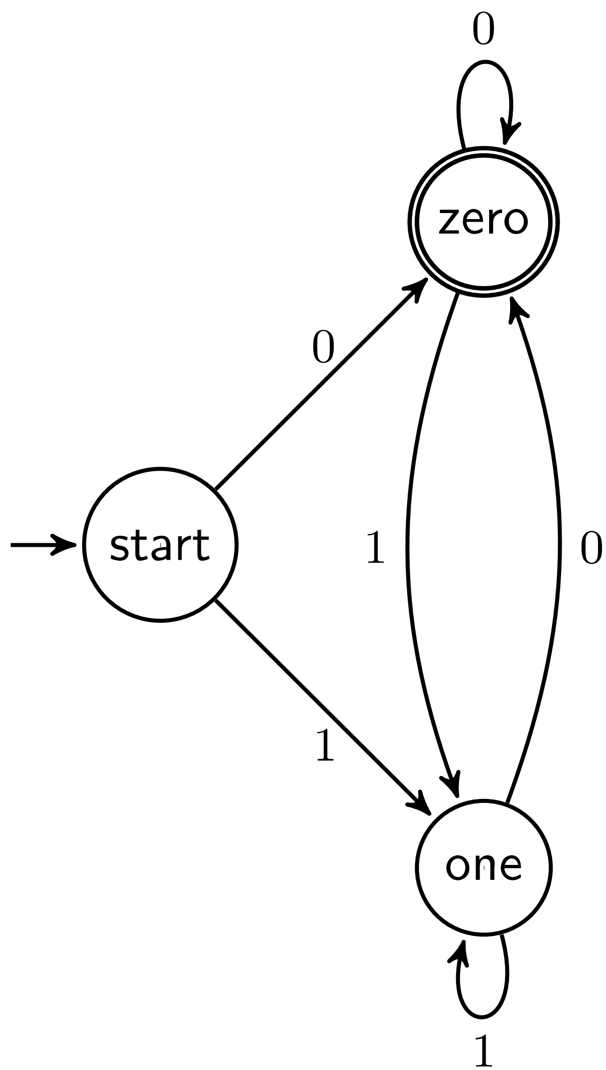
CSE 311: Foundations of Computing

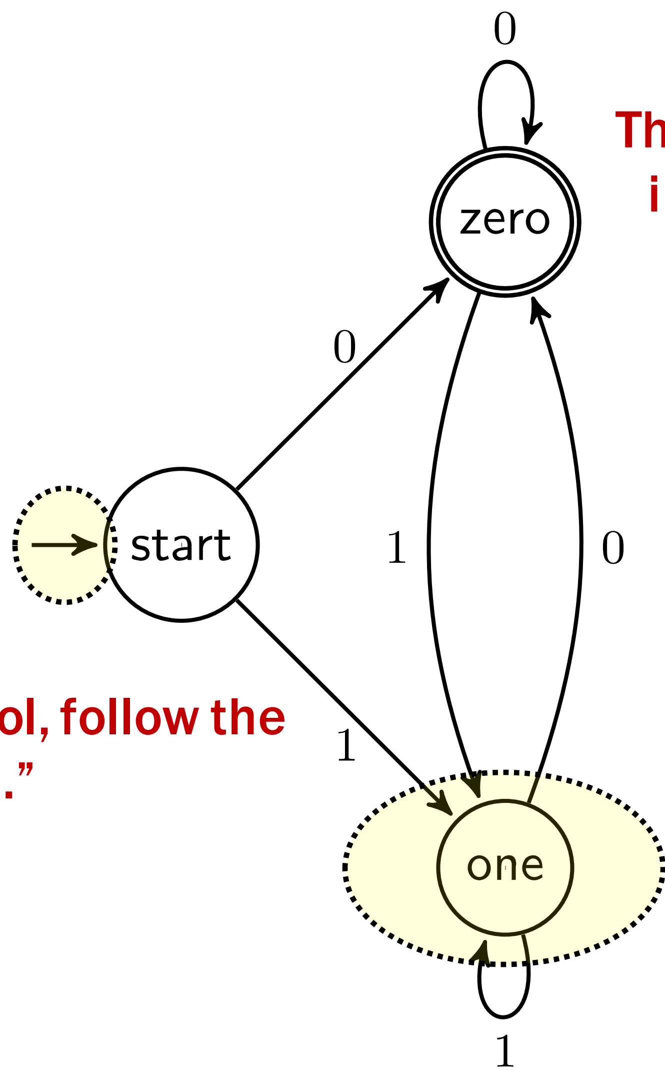
Lecture 23:   DFAs and Directed Graphs



Selecting strings using labeled graphs as “machines”



Deterministic Finite Automata (Finite State Machines)



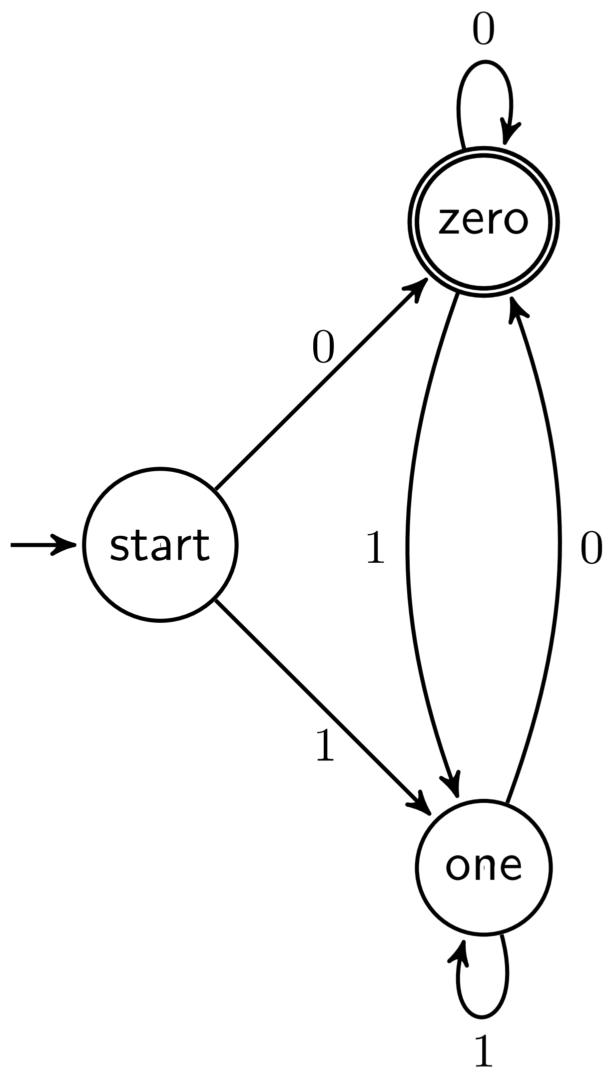
The "double circle" means "the input is good if it ends here"

"Start here"

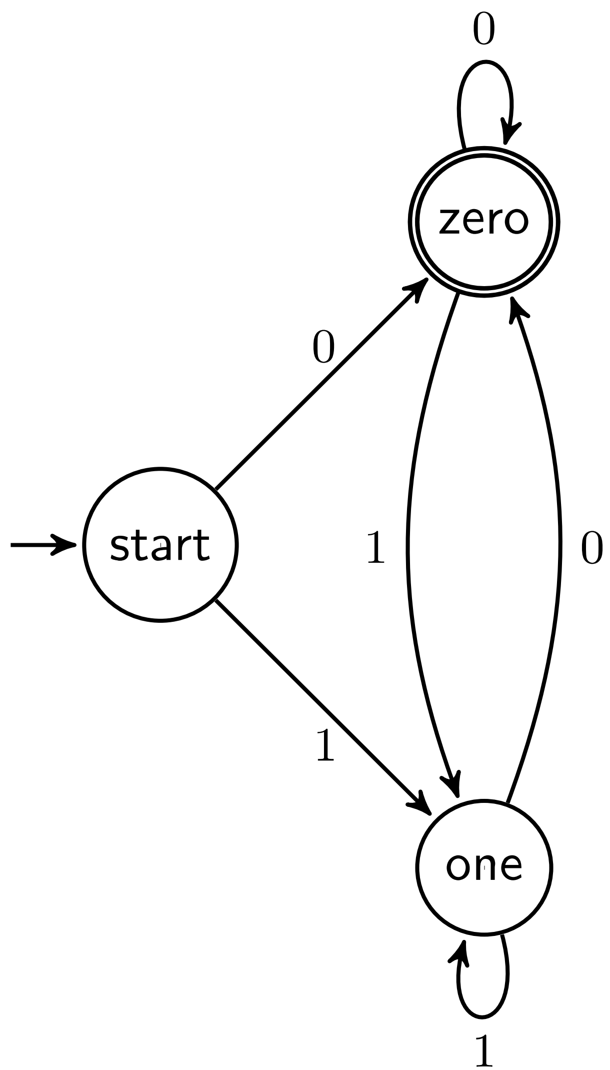
"If I get this symbol, follow the arrow..."

The circles are called "states"
We're only in a single state at any point in time...

Which strings does this machine say are OK?



Which strings does this machine say are OK?

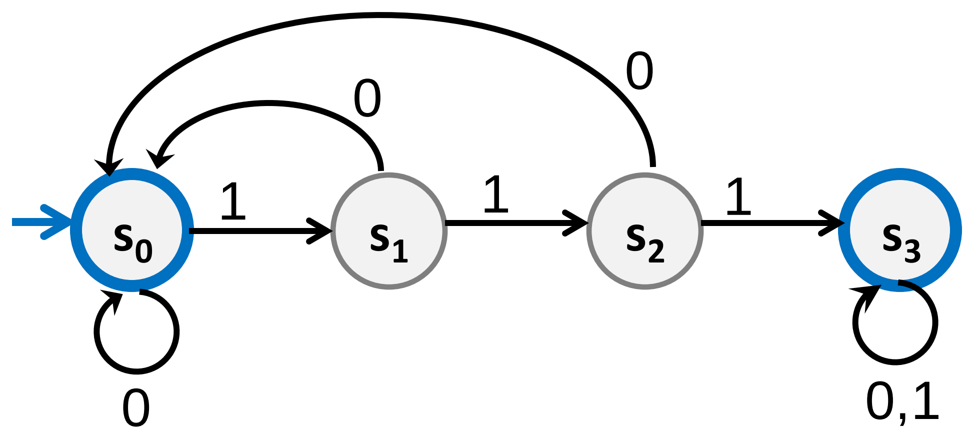


The set of all binary strings that end in 0

Deterministic Finite Automata (DFAs)

- States
- Transitions on input symbols
- Start state and final states
- The “language recognized” by the machine is the set of strings that reach a final state from the start

Old State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3

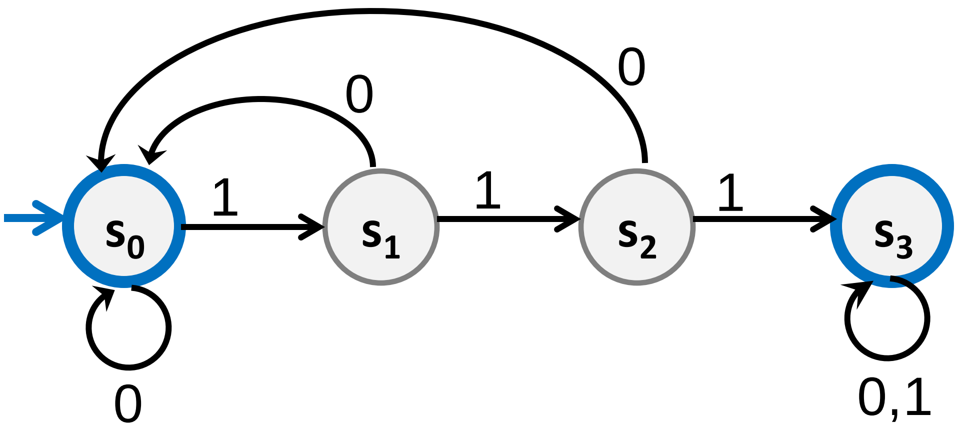


Deterministic Finite Automata (DFAs)

- **Each machine designed for strings over some fixed alphabet Σ.**

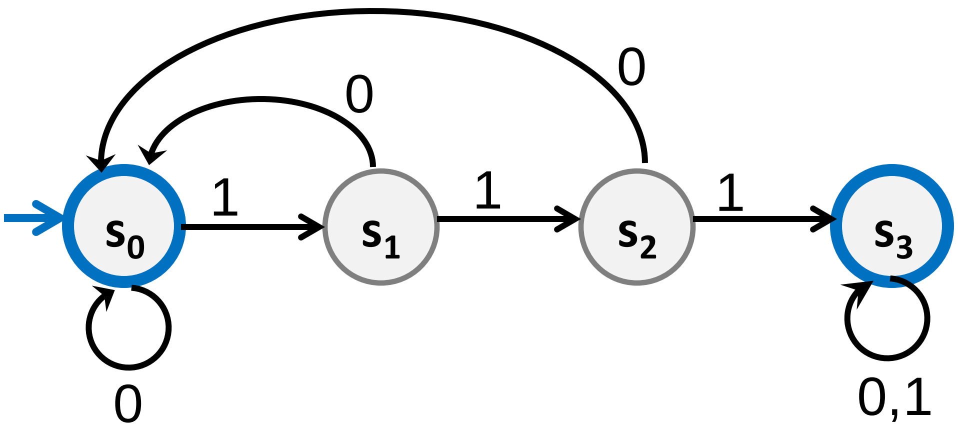
- **Must have a transition defined from each state for *every* symbol in Σ.**

Old State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3



What language does this machine recognize?

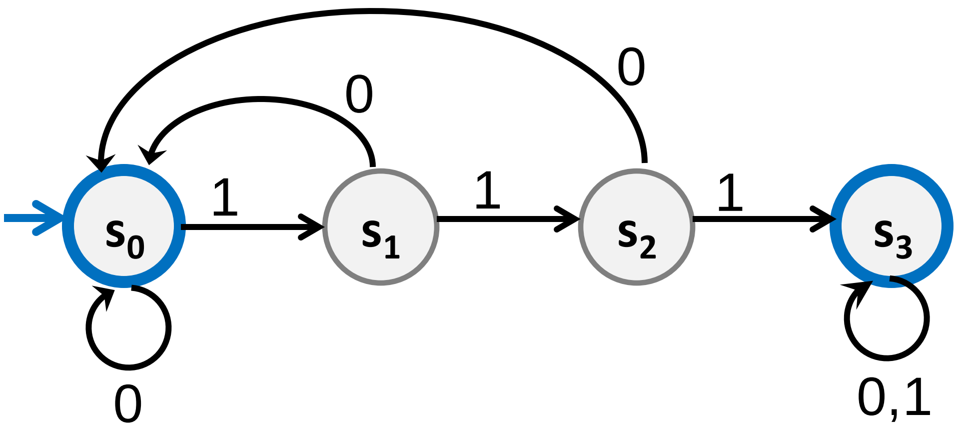
Old State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3



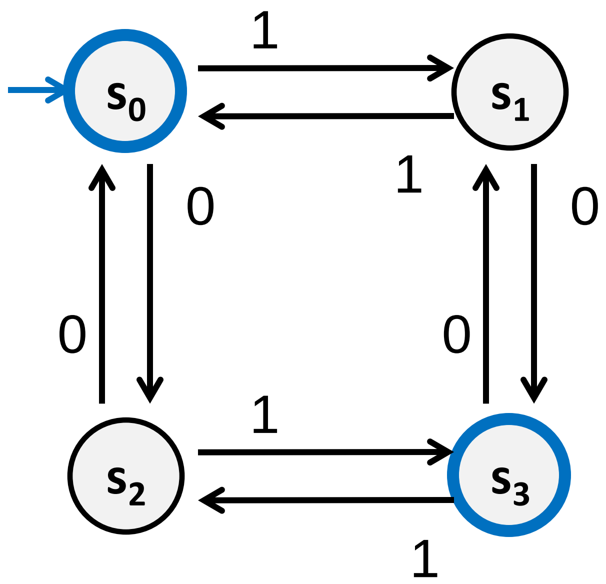
What language does this machine recognize?

The set of all binary strings that contain 111 or don't end in 1

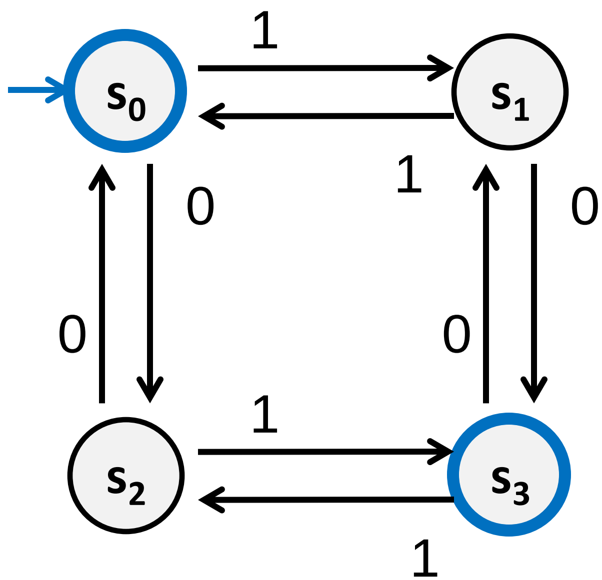
Old State	0	1
s_0	s_0	s_1
s_1	s_0	s_2
s_2	s_0	s_3
s_3	s_3	s_3



What language does this machine recognize?



What language does this machine recognize?



The set of all binary strings with # of 1's ≡ # of 0's (mod 2) (both are even or both are odd).

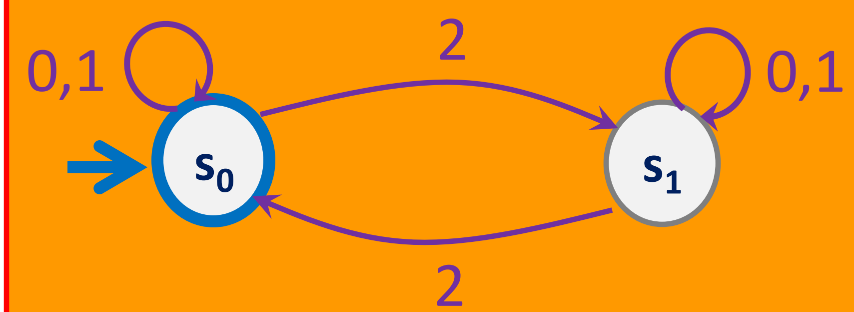
Lecture 23 Activity

You will be assigned to breakout rooms. Please:

- Introduce yourself
- Choose someone to share their screen, showing this PDF
- Design a DFA with input alphabet $\Sigma = \{0,1,2\}$ that recognizes precisely the strings where the sum of the digits is congruent 0 mod 3.

(Hint: Try it first with just the alphabet {0,1})

Example M_1: Strings with an even number of 2's



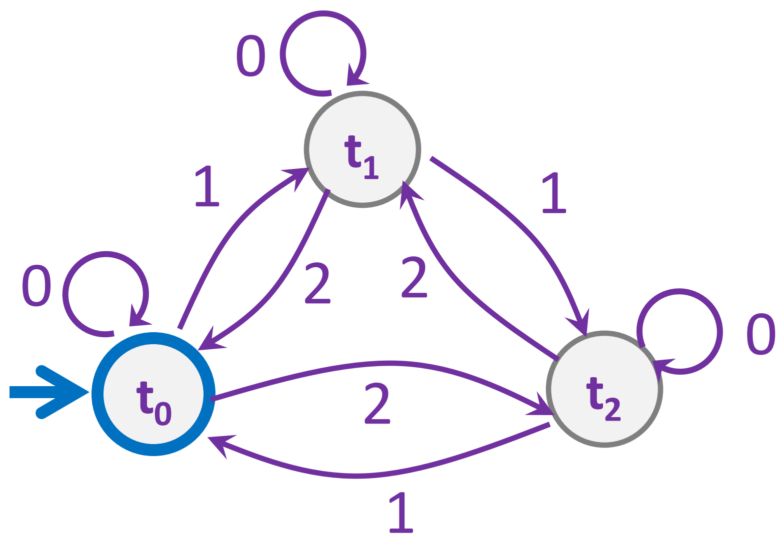
Fill out the poll everywhere for Activity Credit!
Go to pollev.com/philipmg and login with your UW identity

Lecture 23 Activity

You will be assigned to breakout rooms. Please:

- Introduce yourself
- Choose someone to share their screen, showing this PDF
- Design a DFA with input alphabet $\Sigma = \{0,1,2\}$ that recognizes precisely the strings where the sum of the digits is congruent 0 mod 3.

One possible solution:



Recap: Relations and their properties?

A relation on A is a set $R \subseteq A \times A$

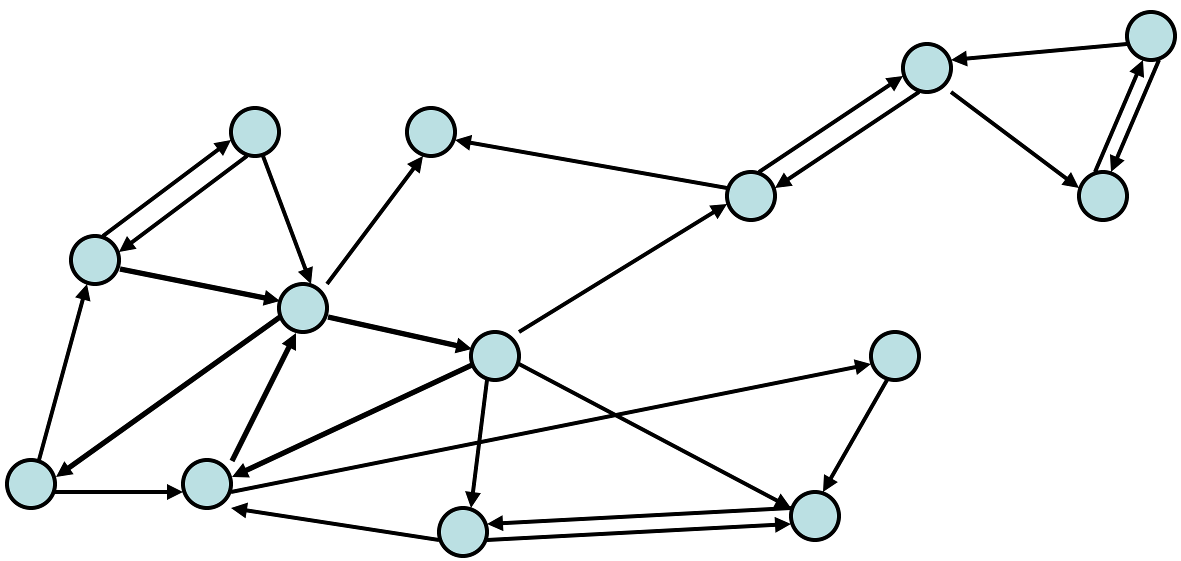
R is **reflexive** iff (a,a) $\in$ R for every a $\in$ A
R is **symmetric** iff (a,b) $\in$ R implies (b, a) $\in$ R
R is **antisymmetric** iff (a,b) $\in$ R and a $\neq$ b implies (b,a) $\notin$ R
R is **transitive** iff (a,b) $\in$ R and (b, c) $\in$ R implies (a, c) $\in$ R

Examples:

- $R_1 = \{ (x, y): x, y \in \mathbb{R} \text{ with } x \geq y\}$:
  reflexive, antisymmetric, transitive
- $R_2 = \{ (x, y): x, y \in \mathbb{Z} \text{ with } x \equiv y \bmod 5\}$:
  reflexive, symmetric, transitive

Directed Graphs

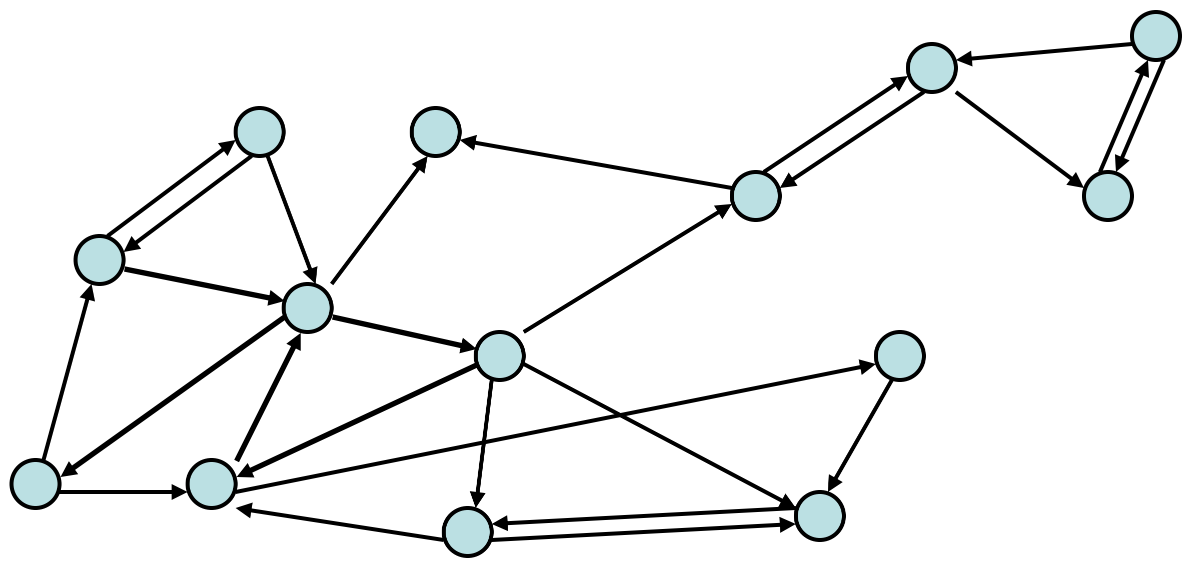
Definition. A *directed graph* is a pair (V, E), where E is a relation on V. The *vertex set* V can be any set; E is the *edge set.*



Directed Graphs

Definition. A *directed graph* is a pair (V, E), where E is a relation on V. The *vertex set* V can be any set; E is the *edge set.*

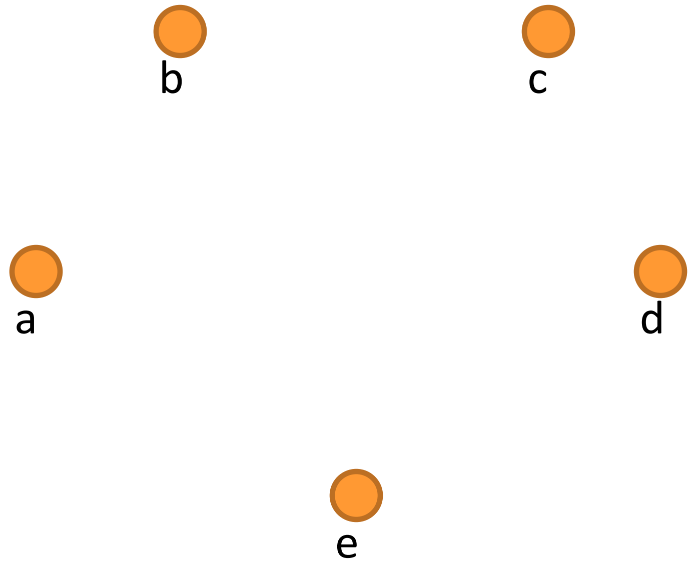
"Let G = (V, E) be a graph"
V – vertices
E – edges, ordered pairs of vertices



Representation of Relations

Directed Graph Representation (Digraph)

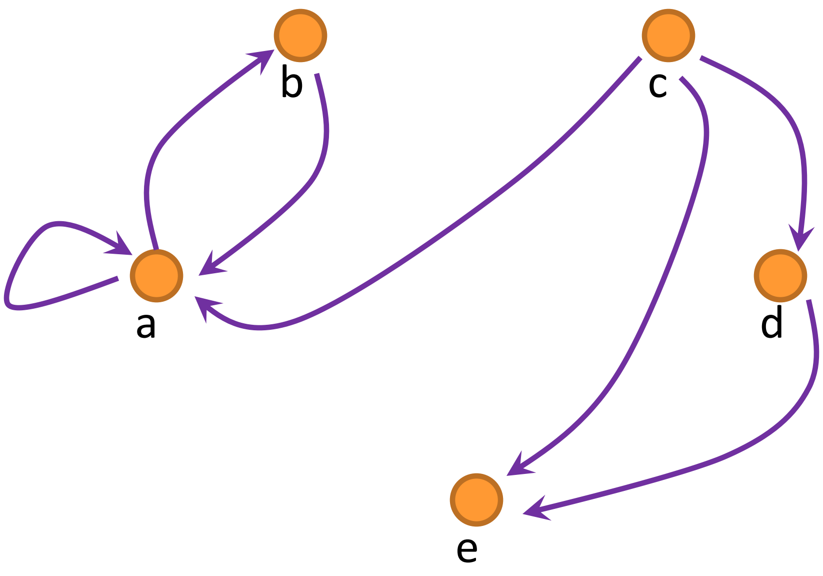
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }



Representation of Relations

Directed Graph Representation (Digraph)

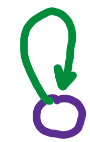
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }



How Properties of Relations show up in Graphs

Let R be a relation on A.

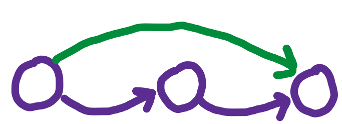
R is **reflexive** iff $(a,a) \in R$ for every $a \in A$



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 $(b,a) \notin R$

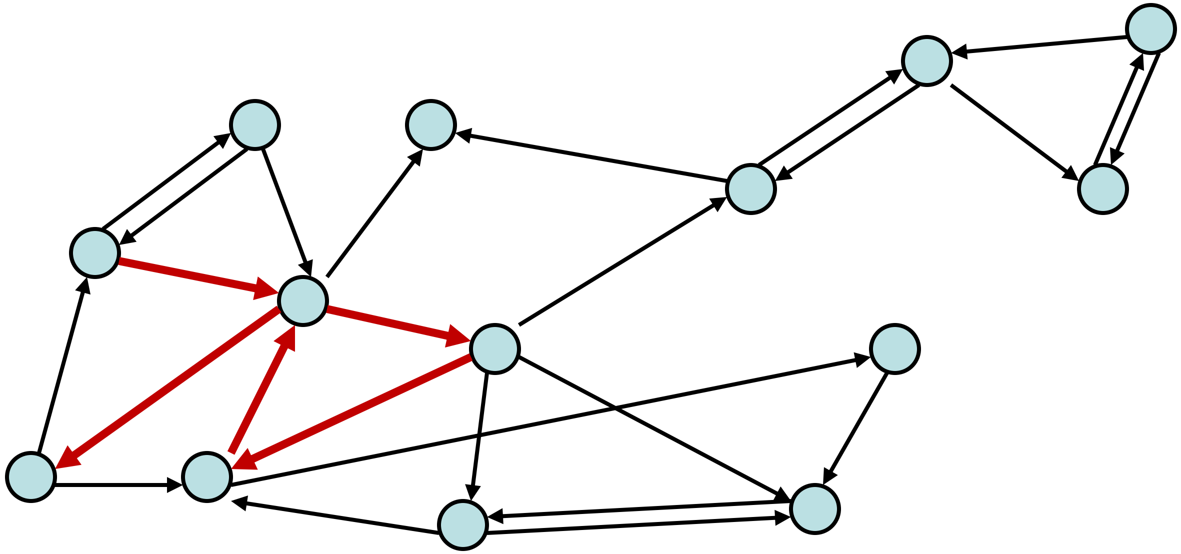
R is **transitive** iff $(a,b) \in R$ and $(b, c) \in R$ implies $(a, c) \in R$



Paths in Relations and Graphs

Definition: Let G = (V, E) be a graph. A *path* in G is a sequence $v_0, v_1, \ldots, v_k$ where each (v_i, v_{i+1}) is in E. $(0 \leq i < k)$

Definition: The **length** of a path $v_0, v_1, \ldots, v_k$ is k, the number of edges in it (counting repetitions if edge used > once).



Paths in Relations and Graphs

Definition: Let G = (V, E) be a graph. A *path* in G is a sequence $v_0, v_1, \ldots, v_k$ where each (v_i, v_{i+1}) is in E. $(0 \leq i < k)$

Definition: The **length** of a path $v_0, v_1, \ldots, v_k$ is k, 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** in the graph (A, R) if and only if $(\mathbf{a},\mathbf{b}) \in \boldsymbol{R}^{\boldsymbol{n}}$

Connectivity In Graphs

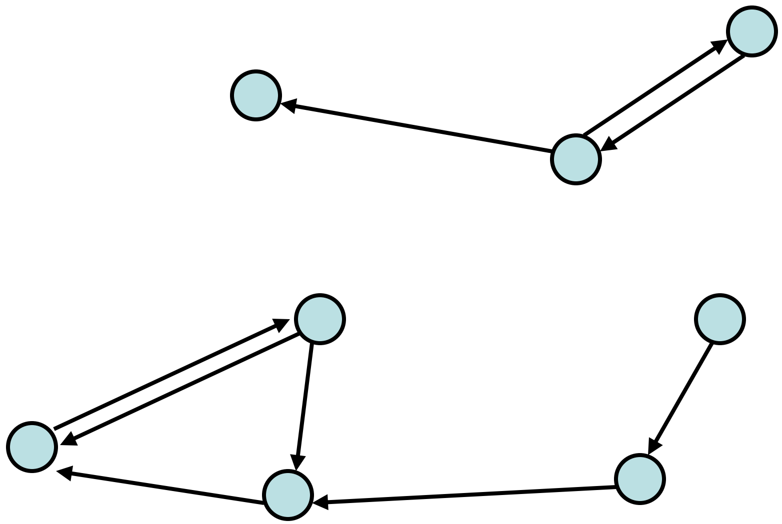
Defn: Two vertices in a graph are **connected** iff there is a path between them.

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^+

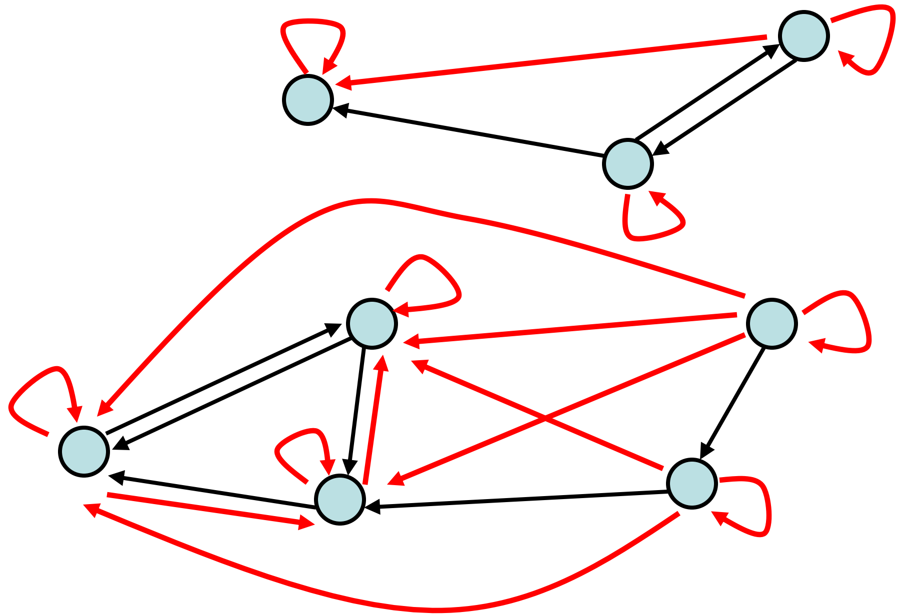
Transitive-Reflexive Closure



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

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

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

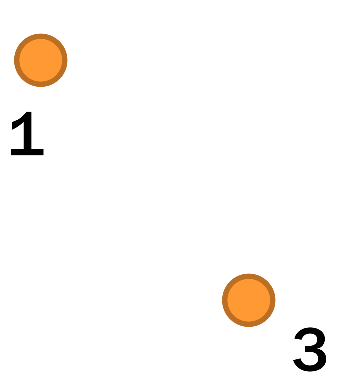
Example application: Database theory

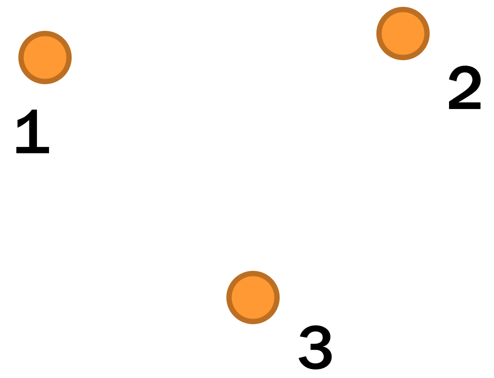
Student_Name	ID_Number	Office	GPA
Knuth	328012098	022	4.00
Von Neuman	481080220	555	3.78
Russell	238082388	022	3.85
Einstein	238001920	022	2.11
Newton	1727017	333	3.61
Karp	348882811	022	3.98
Bernoulli	2921938	022	3.21

Relational Composition using Digraphs

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

Compute $R \circ S$

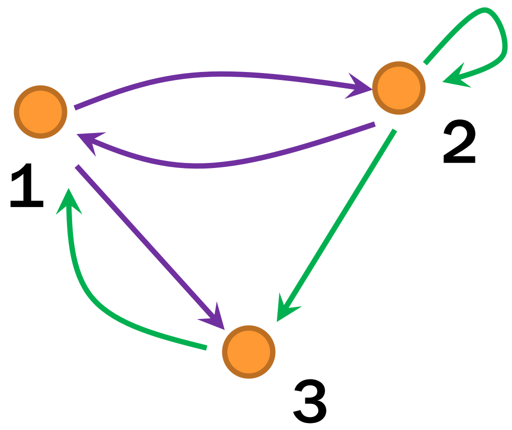


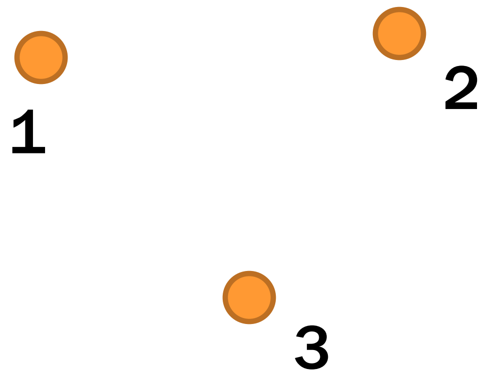


Relational Composition using Digraphs

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

Compute $R \circ S$





Relational Composition using Digraphs

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

Compute $R \circ S$