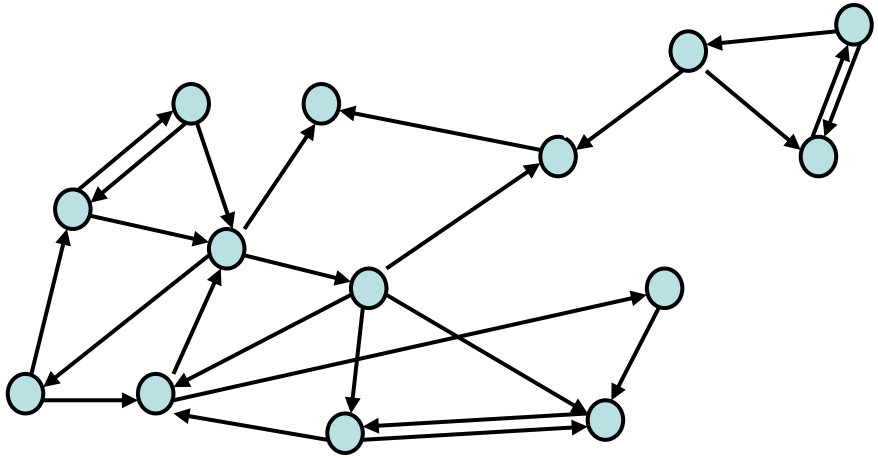
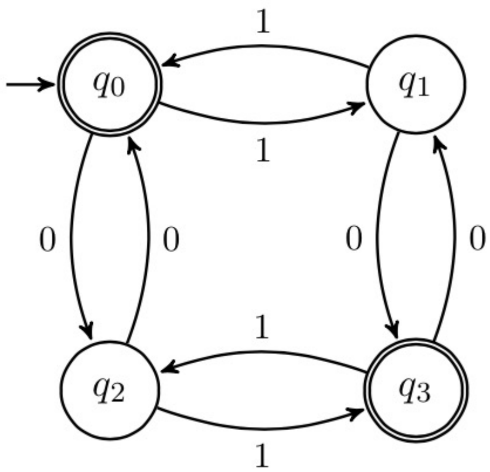
CSE 311: Foundations of Computing

Lecture 21: Directed Graphs, Finite State Machines





Last time: 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 $A \times A$

Last time: 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

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 $(a, b) \in f$

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

Function composition: If $f : A \to B$ and $g : B \to C$ then their **composition** $g \circ f : A \to C$ is defined by

$$g \circ f\,(a) = g(f(a))$$

Composing 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, $S \circ R$ is the relation from A to C defined by:

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

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

The order of writing composition generalizes the function case

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) $\in$ Sister $\circ$ Parent?

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

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

Powers of a Relation

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

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

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

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

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

Matrix Representation

Relation $\boldsymbol{R}$ on $\boldsymbol{A} = \{a_1, \ldots, a_n\}$

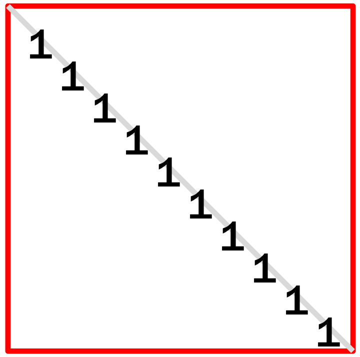
$$\boldsymbol{m_{ij}} = \begin{cases} 1 & \text{if } (a_i, a_j) \in \boldsymbol{R} \\ 0 & \text{if } (a_i, a_j) \notin \boldsymbol{R} \end{cases}$$

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

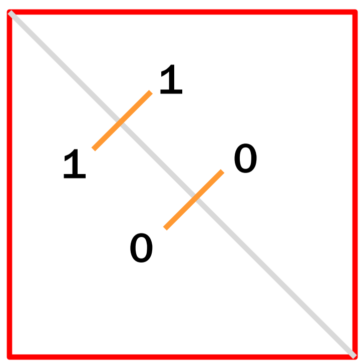
	1	2	3	4
1	1	1	0	1
2	1	0	1	0
3	0	1	1	0
4	0	1	1	0

Properties using matrix representation

reflexive

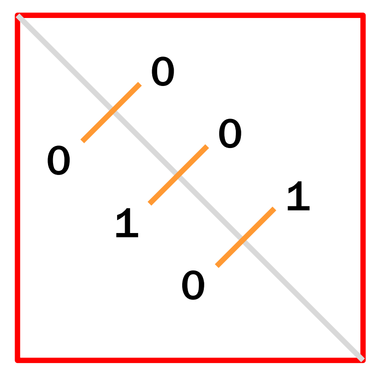


symmetric



Same when
rows & columns
swapped

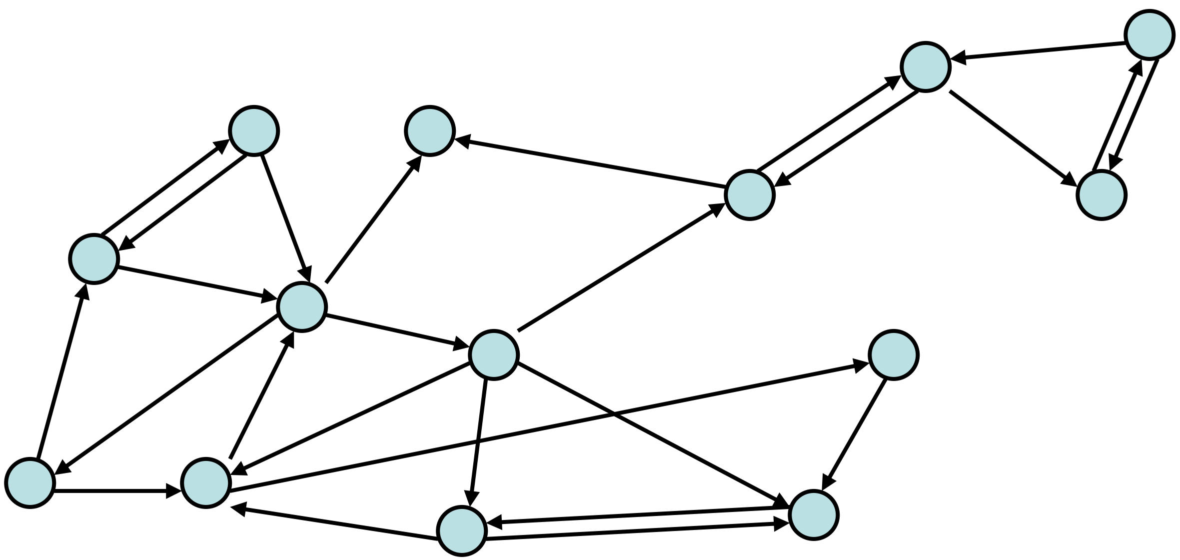
anti-symmetric



No 1-1 pairs

Directed Graphs

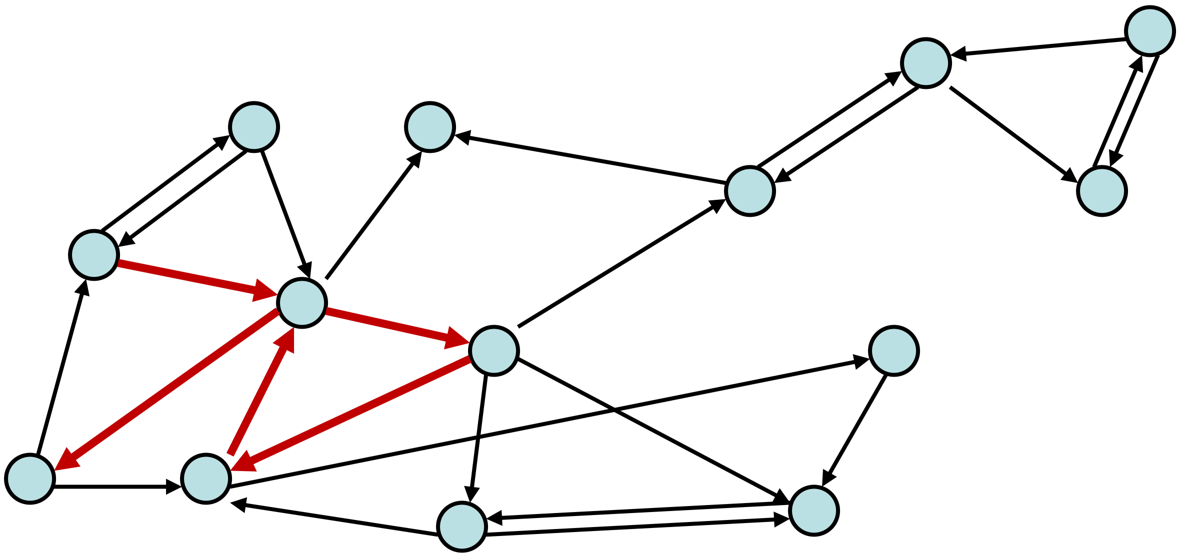
G = (V, E) V – vertices
E – edges, ordered pairs of vertices



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

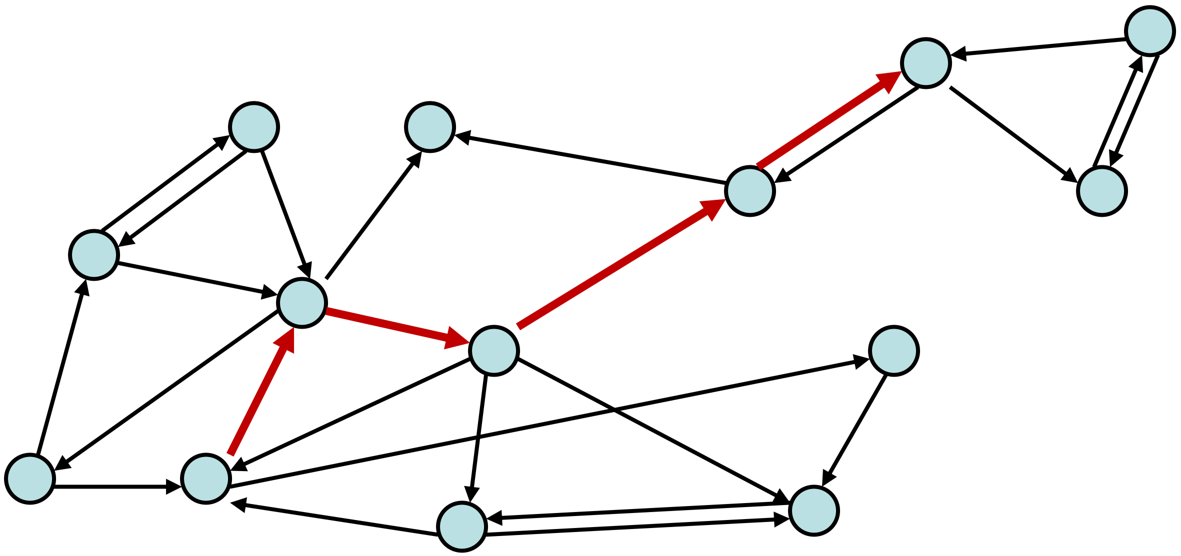


Directed Graphs

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

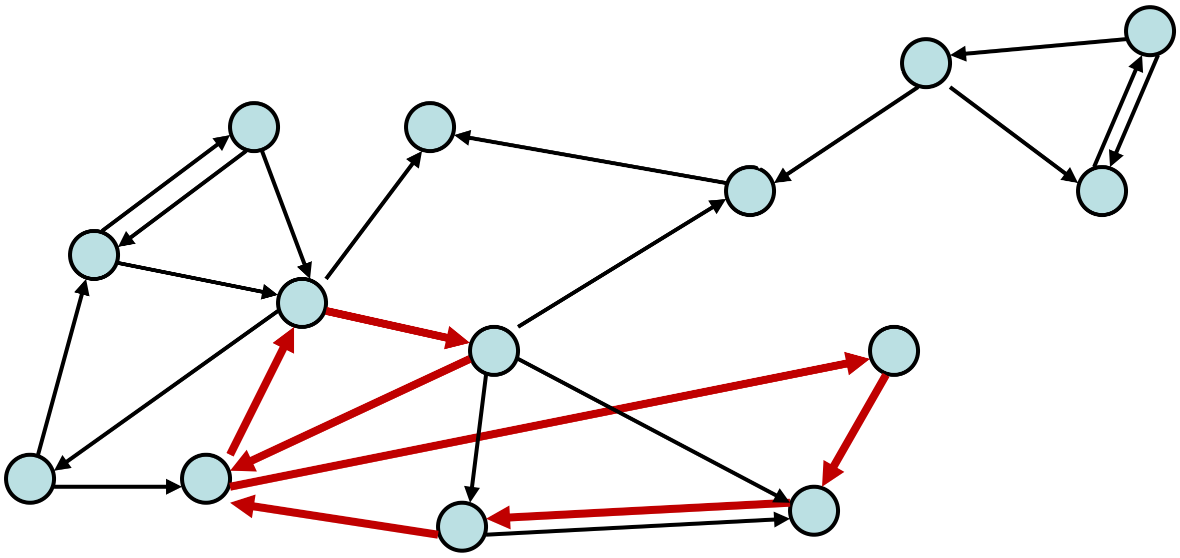


Directed Graphs

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

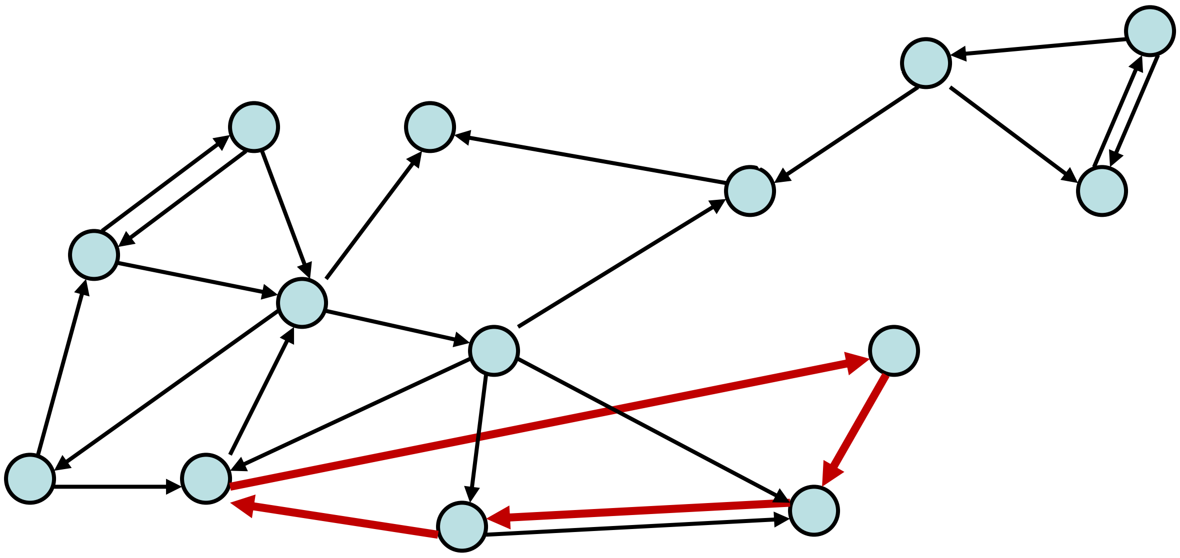


Directed Graphs

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

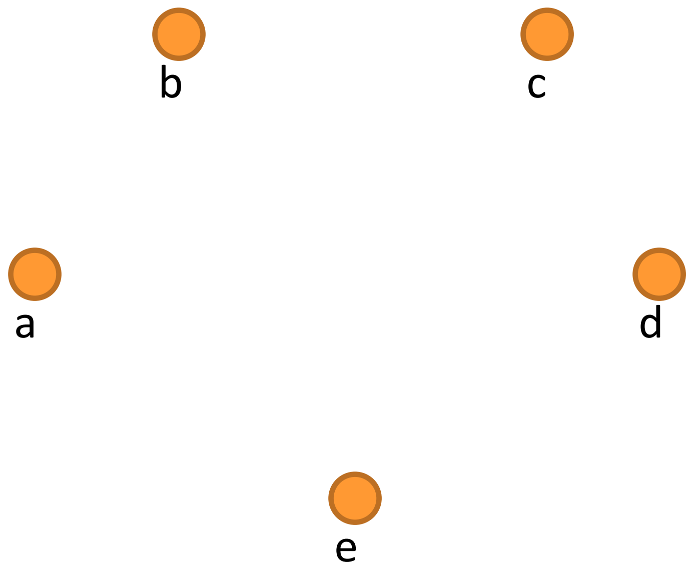
Simple Path: none of $\mathbf{v_0}, \ldots, \mathbf{v_k}$ repeated
Cycle: $\mathbf{v_0} = \mathbf{v_k}$
Simple Cycle: $\mathbf{v_0} = \mathbf{v_k}$, none of $\mathbf{v_1}, \ldots, \mathbf{v_k}$ repeated



Representation of Relations

Directed Graph Representation (Digraph)

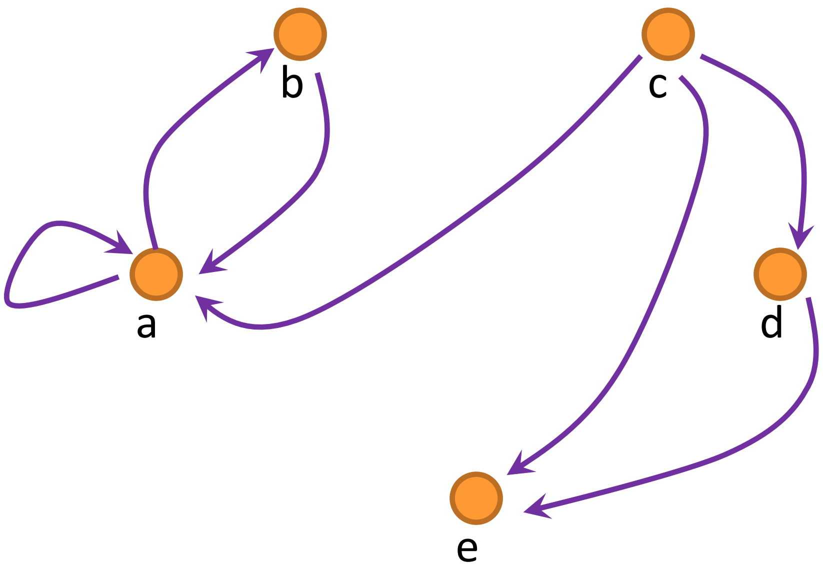
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }



Representation of Relations

Directed Graph Representation (Digraph)

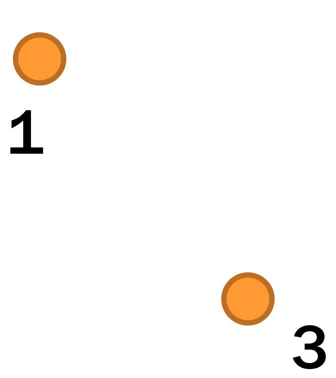
{(a, b), (a, a), (b, a), (c, a), (c, d), (c, e) (d, e) }

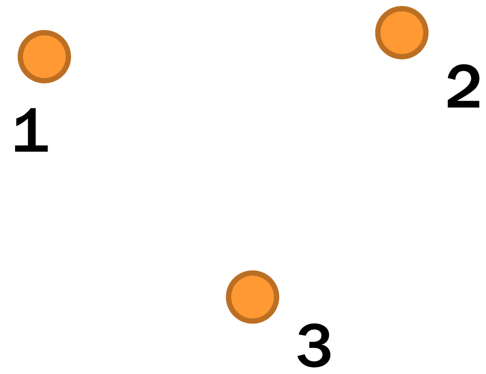


Relational Composition using Digraphs

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

Compute $S \circ R$

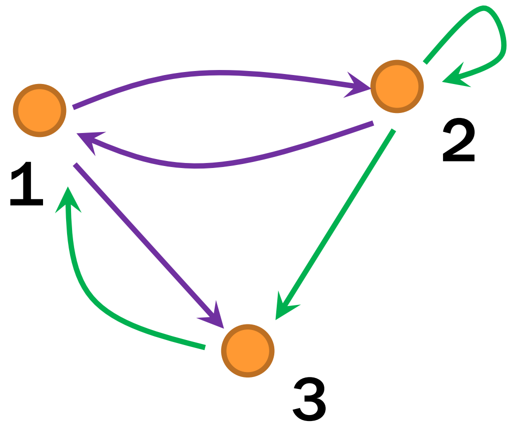


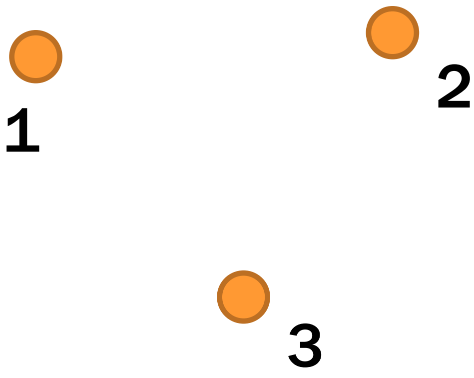


Relational Composition using Digraphs

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

Compute $S \circ R$

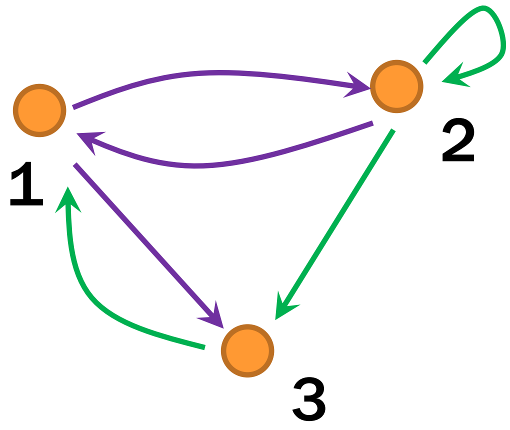


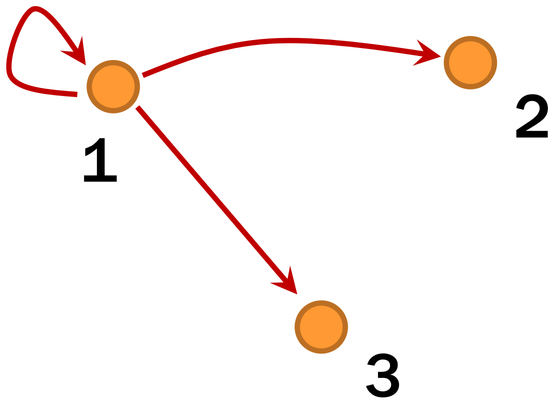


Relational Composition using Digraphs

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

Compute $S \circ R$

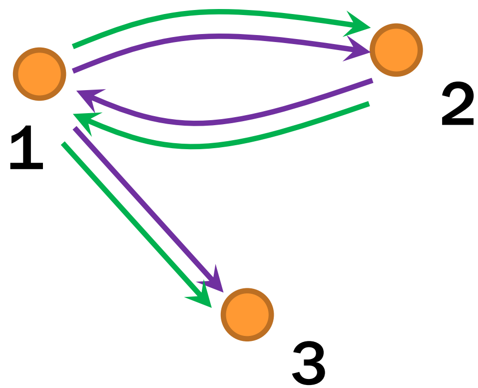


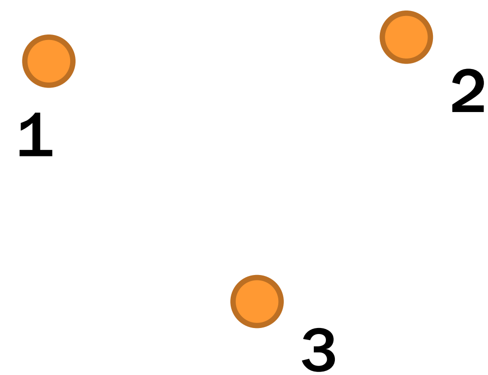


Relational Composition using Digraphs

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

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





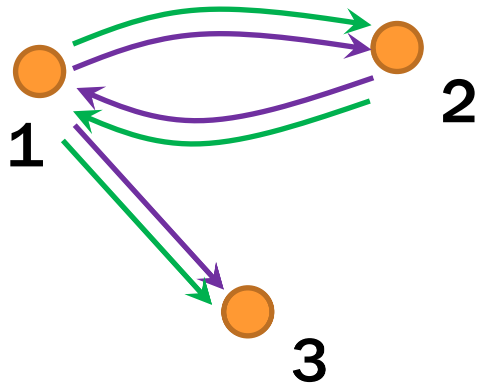
$(a,c) \in R \circ R = R^2$ iff $\exists b\ ((a,b) \in R \land (b,c) \in R)$

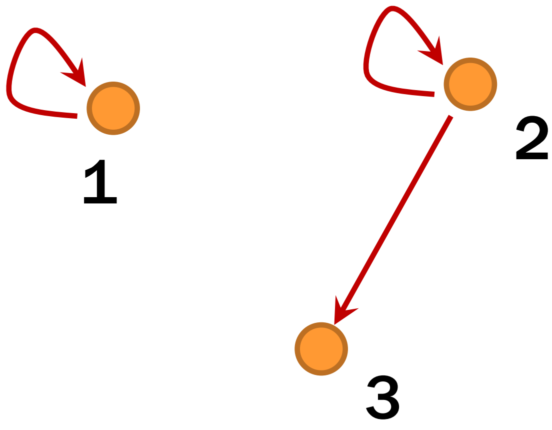
iff $\exists b$ such that a, b, c is a path

Relational Composition using Digraphs

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

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





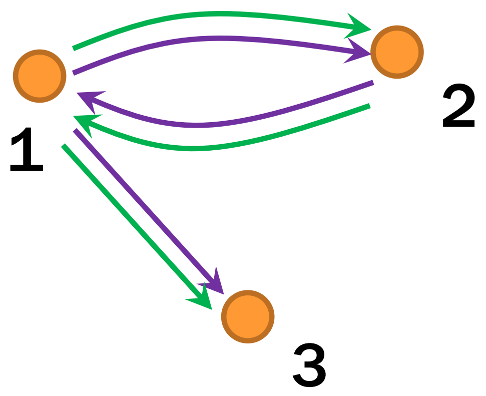
$(a, c) \in R \circ R = R^2$ iff $\exists b\ ((a, b) \in R \land (b, c) \in R)$

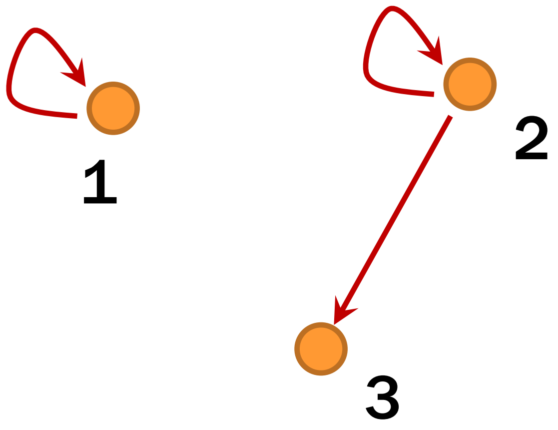
iff $\exists b$ such that a, b, c is a path

Relational Composition using Digraphs

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

Compute $R \circ R$





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 Graphs and Relations

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

Elements of R^0 correspond to paths of length 0.
Elements of $R^1 = R$ are paths of length 1.
Elements of R^2 are paths of length 2.
...

Paths in Graphs and Relations

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** in the digraph for $\boldsymbol{R}$ 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 Rosen book uses the wrong definition of this quantity. What the Rosen 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 $(b, a) \in R$

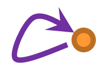
R is **antisymmetric** iff $(a,b) \in R$ and $a \neq b$ implies $(b,a) \notin R$

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

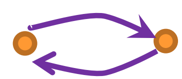
How Properties of Relations show up in Graphs

Let R be a relation on A.

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

 at every node

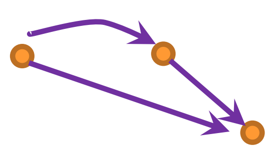
R is **symmetric** iff $(a,b) \in R$ implies $(b, a) \in R$

 or 

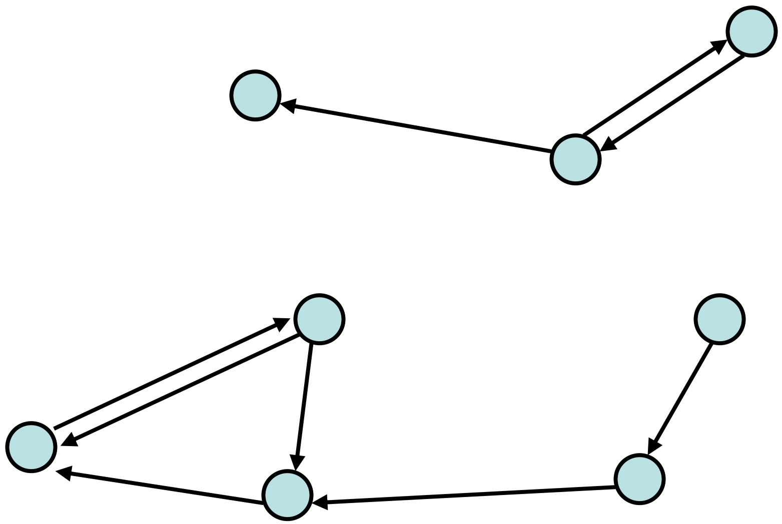
R is **antisymmetric** iff $(a,b) \in R$ and $a \neq b$ implies $(b,a) \notin R$

 or  or 

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

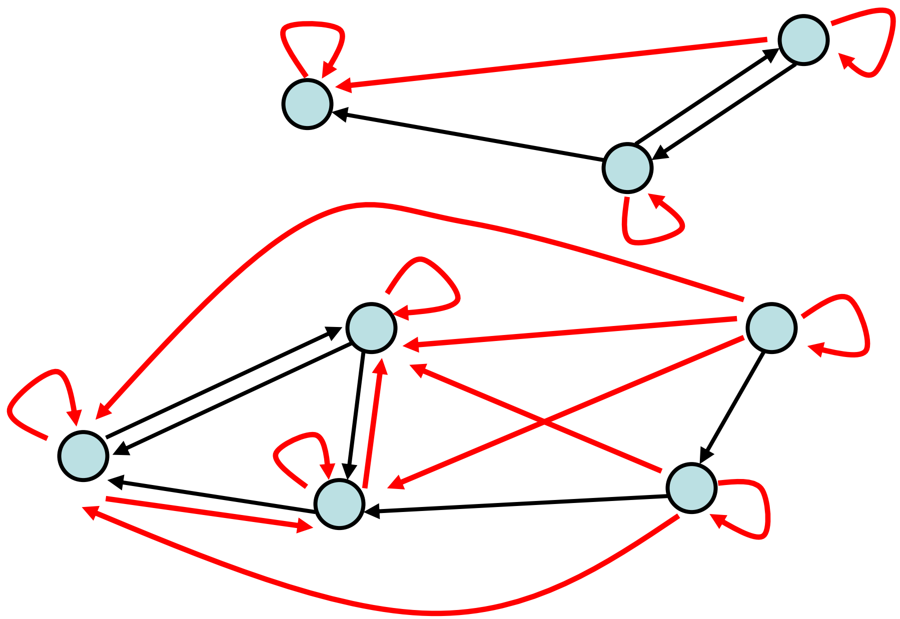


Transitive-Reflexive Closure



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

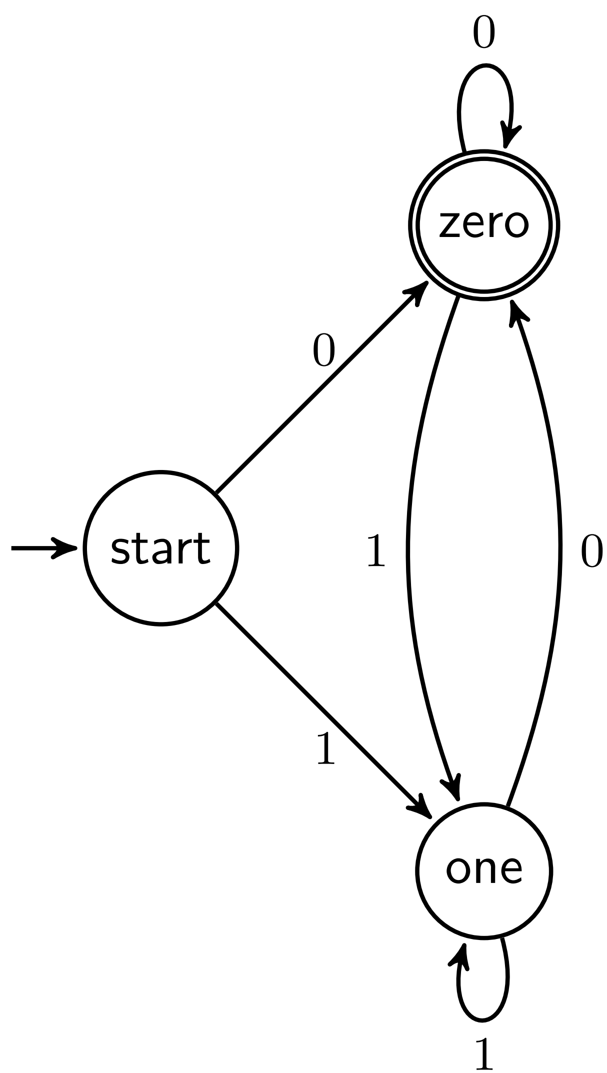
Relational Databases

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

Back to Languages

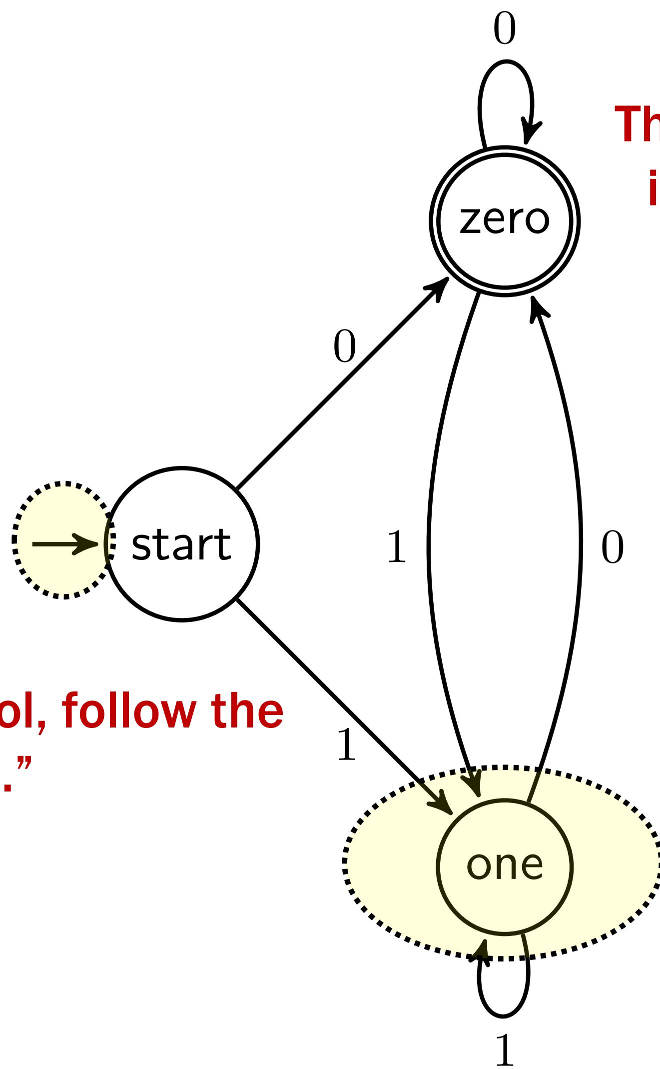
Selecting strings using labeled graphs as “machines”



Finite State Machines

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

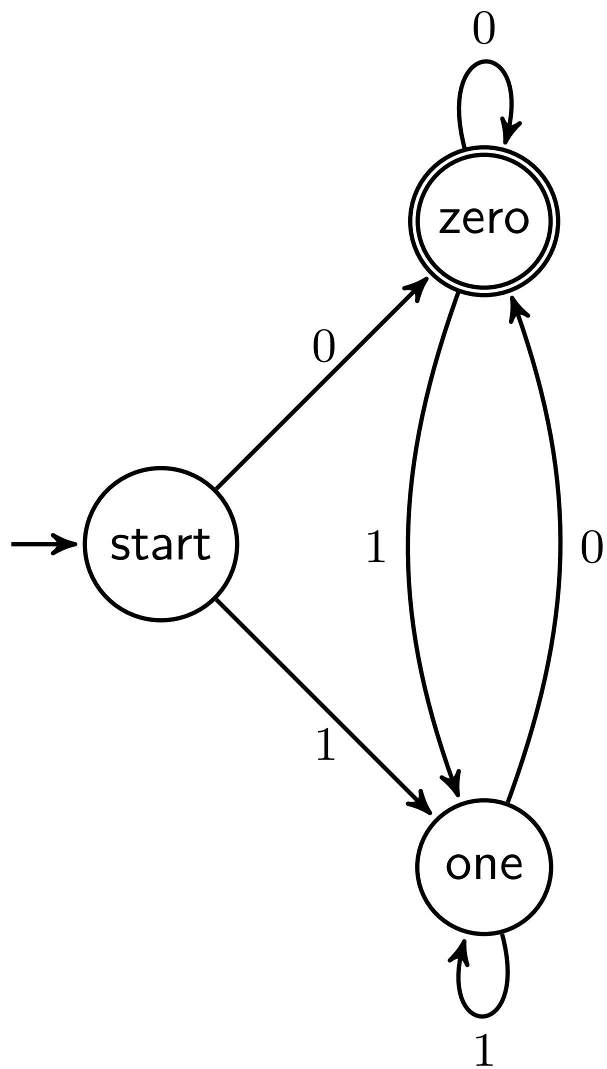
“Start here”



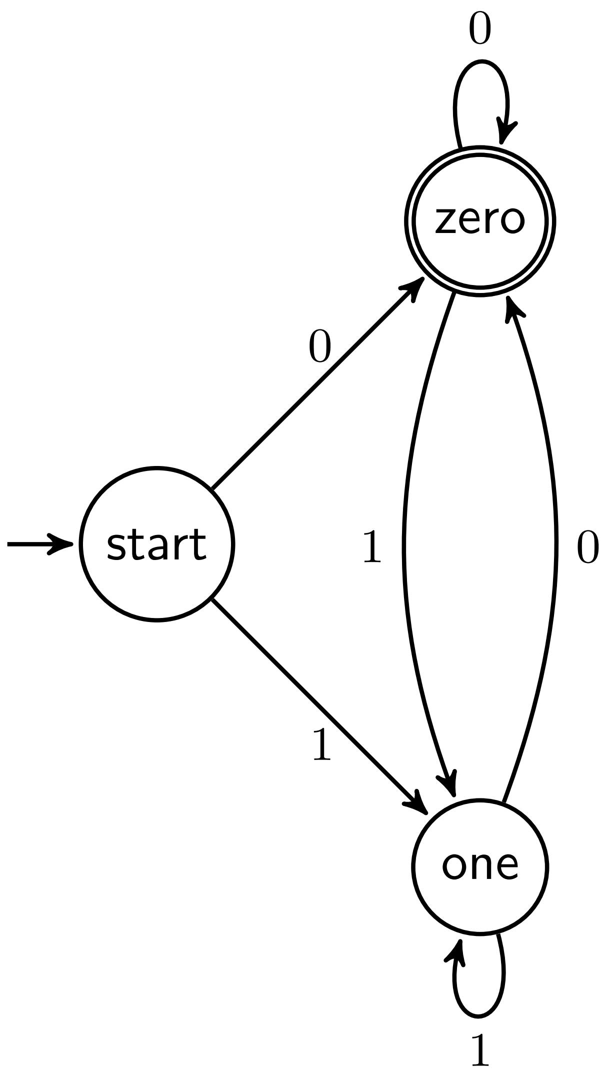
“If I get this symbol, follow the arrow…”

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?

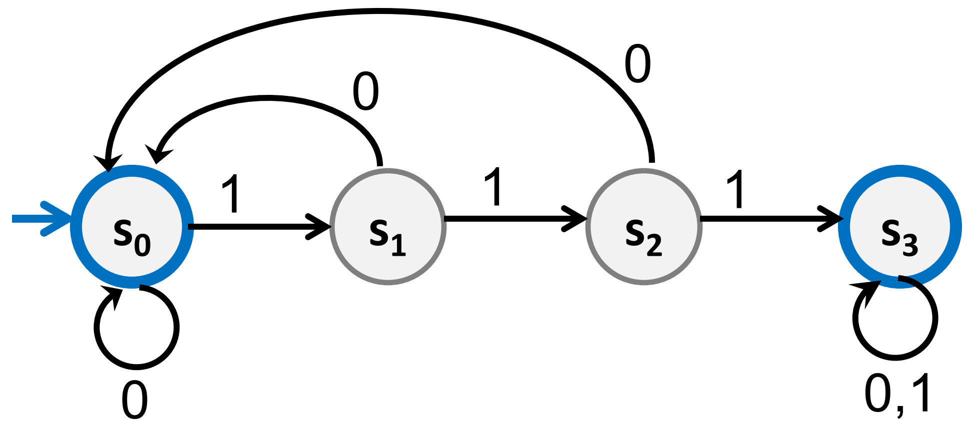


The set of all binary strings that end in 0

Finite State Machines

- 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

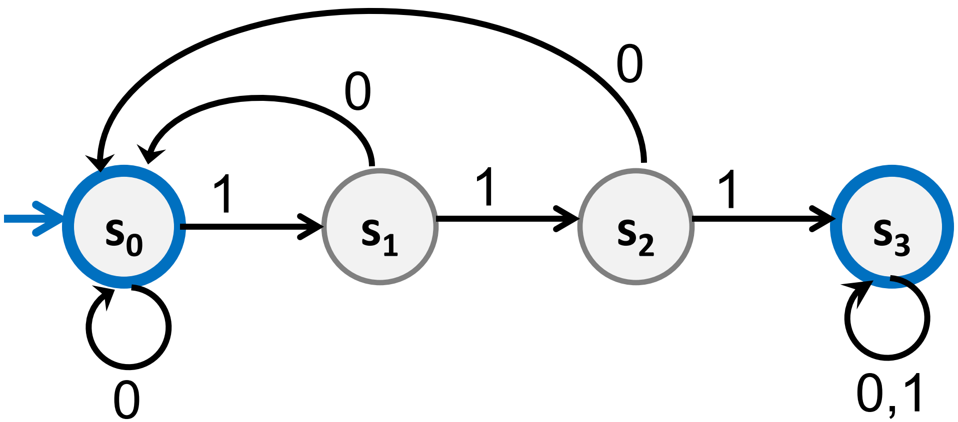


Finite State Machines

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

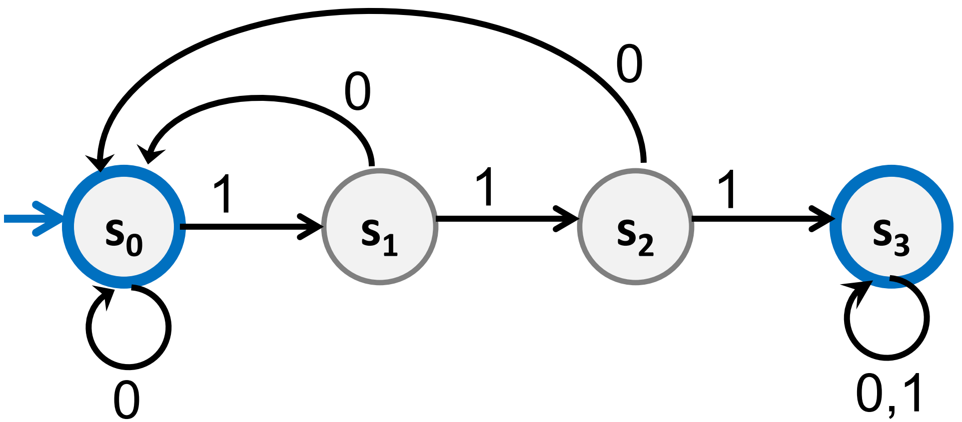
- Must have a transition defined from each state for ***every*** symbol in Σ.

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