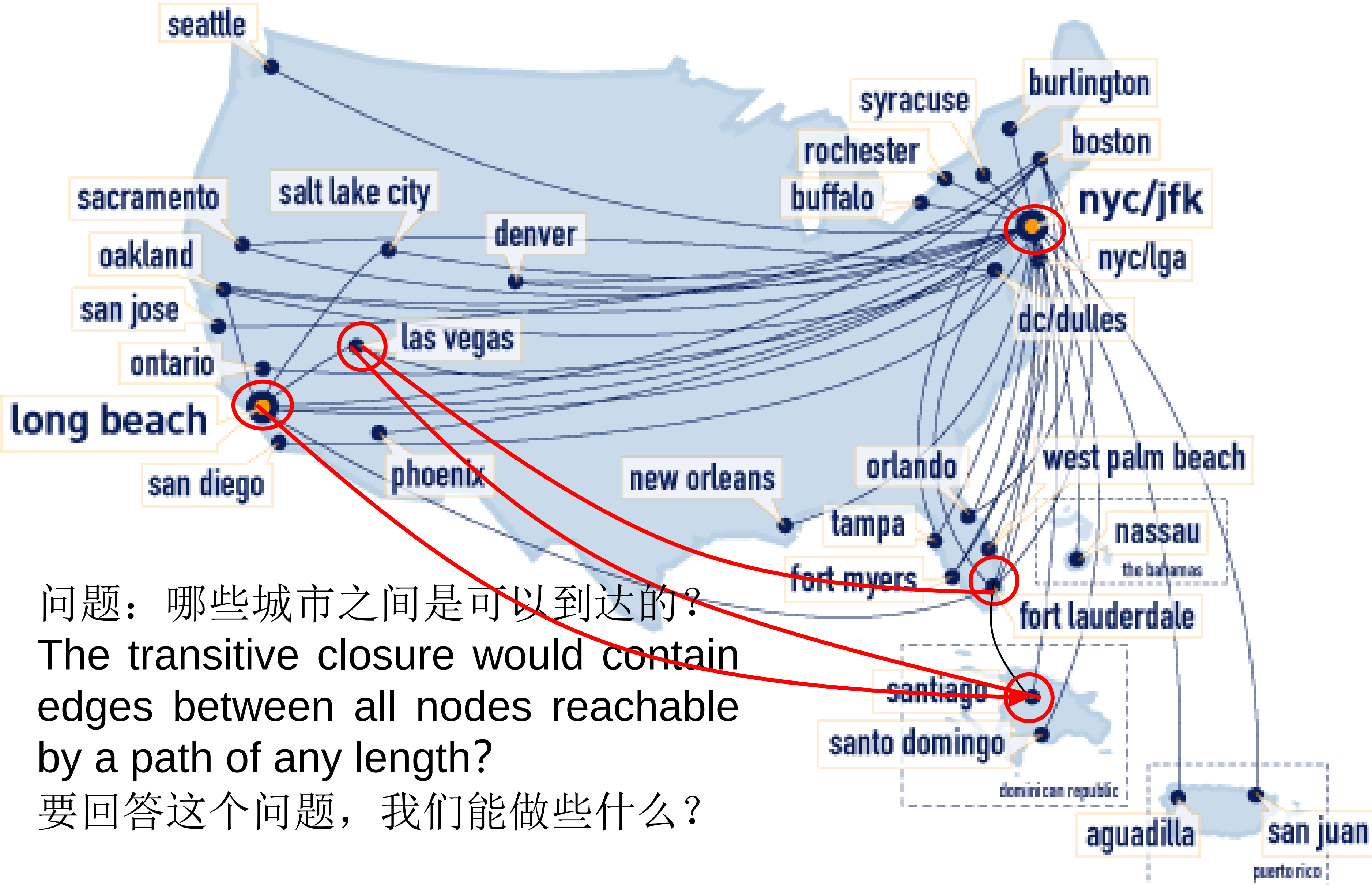
Closures of Relations

关系闭包

Transitive closure传递闭包



问题：哪些城市之间是可以到达的？

The transitive closure would contain edges between all nodes reachable by a path of any length?

要回答这个问题，我们能做些什么？

- 再思考：一个通讯网络，在已知哪些点之间有直接的连接的前提下， 哪些终端之间是可以通讯的？ 我们能否对于给定的网络图，构造一个新的结构，这个结构能明确告知我们那些点之间是可以通讯的？

Relational closures关系闭包

- We know that:
- a relation R on a non-empty set A may or may not have some special property such as “Transitive”.

- But for some purpose, we need a relation which is transitive and contains R as its subset.

- Question: how can we add some pairs to R to make R “a little bit bigger”, then it is transitive? What is that relation? Is it possible? Easy or hard?
- The same idea for other properties: Reflexive and Symmetric
- Introduction of the concept of relational closure…

Relational closures

- Three types of Closures we will study
  - Reflexive 自反闭包
    - Easy
  - Symmetric 对称闭包
    - Easy
  - Transitive 传递闭包
    - Hard

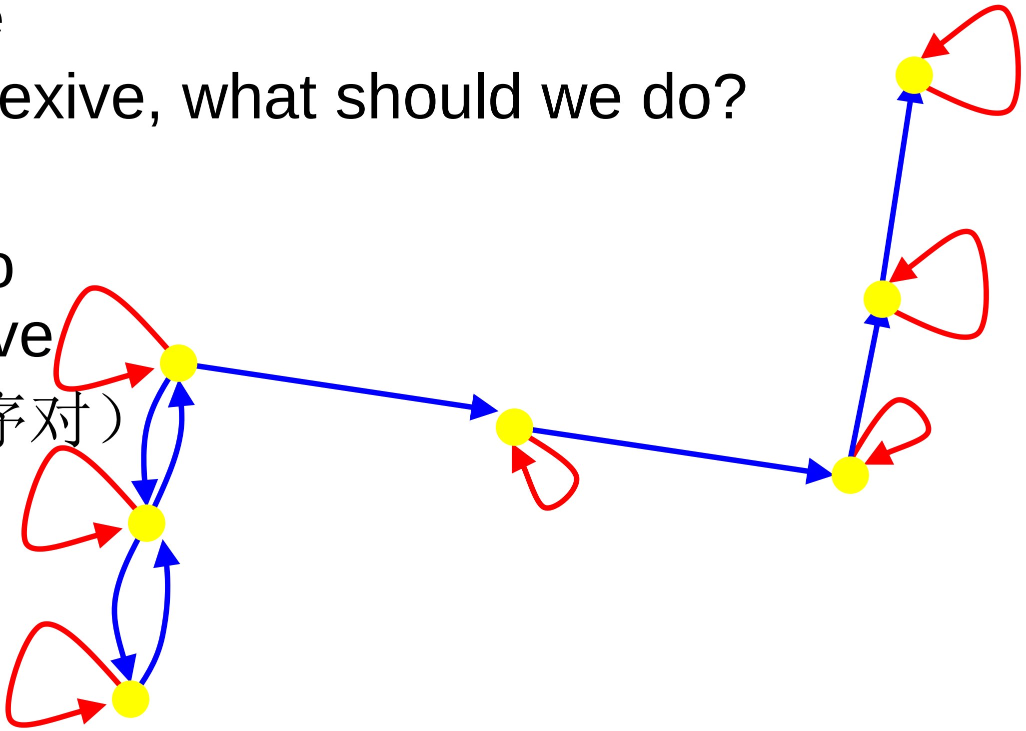
Definition闭包定义

- R is a relation on a non-empty set A, R 可能不满足某种特性 “P” such as reflexive/transitive/symmetric.
- 用$C_P(R)$ 表示关系R的关于特性P的闭包，则闭包定义如下：
- **定义**：如果$C_P(R)$ 是A上的包含R的（$R \subseteq C_P(R)$ ）满足特性 “P” 关系，而且如果还有其它的二元关系S也满足包含R且具有特性 ”P”的话， 就有 $C_P(R) \subseteq S$.
- $C_P(R)$ is called as the closure of R with respect to the property “P”. 这样的$C_P(R)$称作R关于性质P的关系闭包

- Actually, the closure $C_P(R)$ of R with respect to property “P” is the minimum relation containing R as a subset and satisfy the property “P”．闭包也即最小的包含R且满足性质”P”的关系， 或者说R的满足性质 “P”最小的超集。
- 如何求闭包也就是如何寻找这个所谓的“最小超集”

Reflexive closure 自反闭包

- Consider a relation *R*:
  - Note that it is not reflexive
- Question: to make R reflexive, what should we do?



- We want to add edges to make the relation reflexive
- 添加边 （实际上是添加序对）
- By adding those edges, we have made a non-reflexive relation *R* into a reflexive relation
- This new relation is called the **reflexive closure** of *R*
- *注意在添加内容的过程中，所有添加的都是为了满足自反性所必须添加的。既不能少添加，也不能多添加。*

Reflexive closure自反闭包公式

- 给每一个没有loop的节点添加loop，构造自反闭包（添加必须加的）

- The reflexive closure自反闭包 of R is

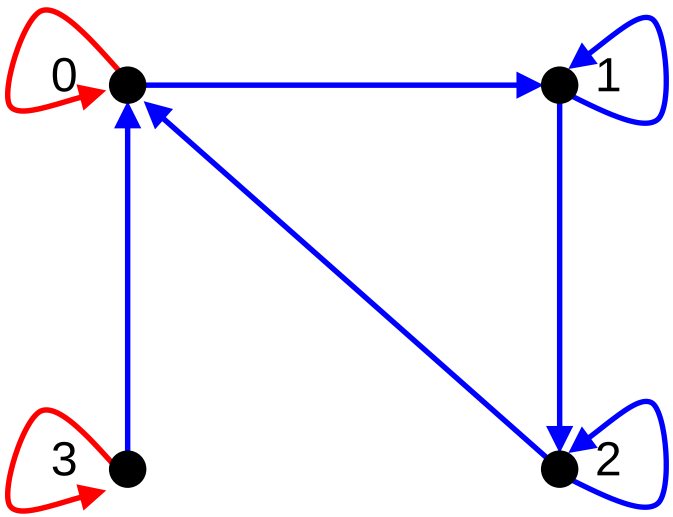
  $R \cup I_A$， Where $I_A = \{ (a,a) \mid a \in A \}$

    - Called the "diagonal relation"

  – With matrices, we set the diagonal to all 1's

Closure--Example

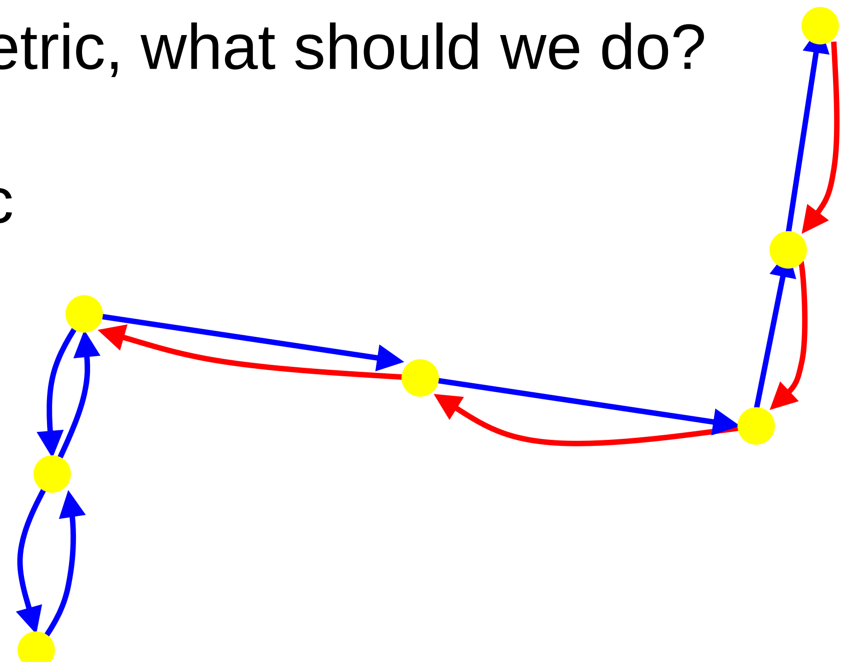
- Let *R* be a relation on the set { 0, 1, 2, 3 } containing the ordered pairs (0,1), (1,1), (1,2), (2,0), (2,2), and (3,0)
- What is the reflexive closure of *R*?
- We add all pairs of edges (*a*,*a*) that do not already exist



We add edges:
(0,0), (3,3)

Symmetric closure对称闭包

- Consider a relation *R*:
  - Note that it is not symmetric (why?)
- Question: to make R symmetric, what should we do?
- We want to add edges to make the relation symmetric
- 添加 对称的边，
- By adding those edges, we have made a non-symmetric relation *R* into a symmetric relation



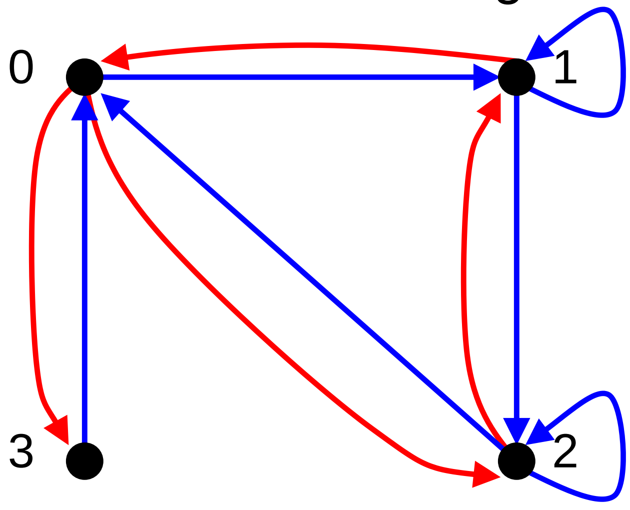
- This new relation is called the **symmetric closure** of *R*

Symmetric closure对称闭包公式

- 添加双向边到所有存在单向边的地方
- The symmetric closure of R is $R \cup R^{-1}$
  - If $R = \{ (a,b) \mid \ldots \}$
  - Then $R^{-1} = \{ (b,a) \mid \ldots \}$

对称Closure--Example

- Let *R* be a relation on the set { 0, 1, 2, 3 } containing the ordered pairs (0,1), (1,1), (1,2), (2,0), (2,2), and (3,0)
- What is the symmetric closure of *R*?
- We add all pairs of edges (*a,b*) where (*b,a*) exists
  - We make all "single" edges into anti-parallel pairs



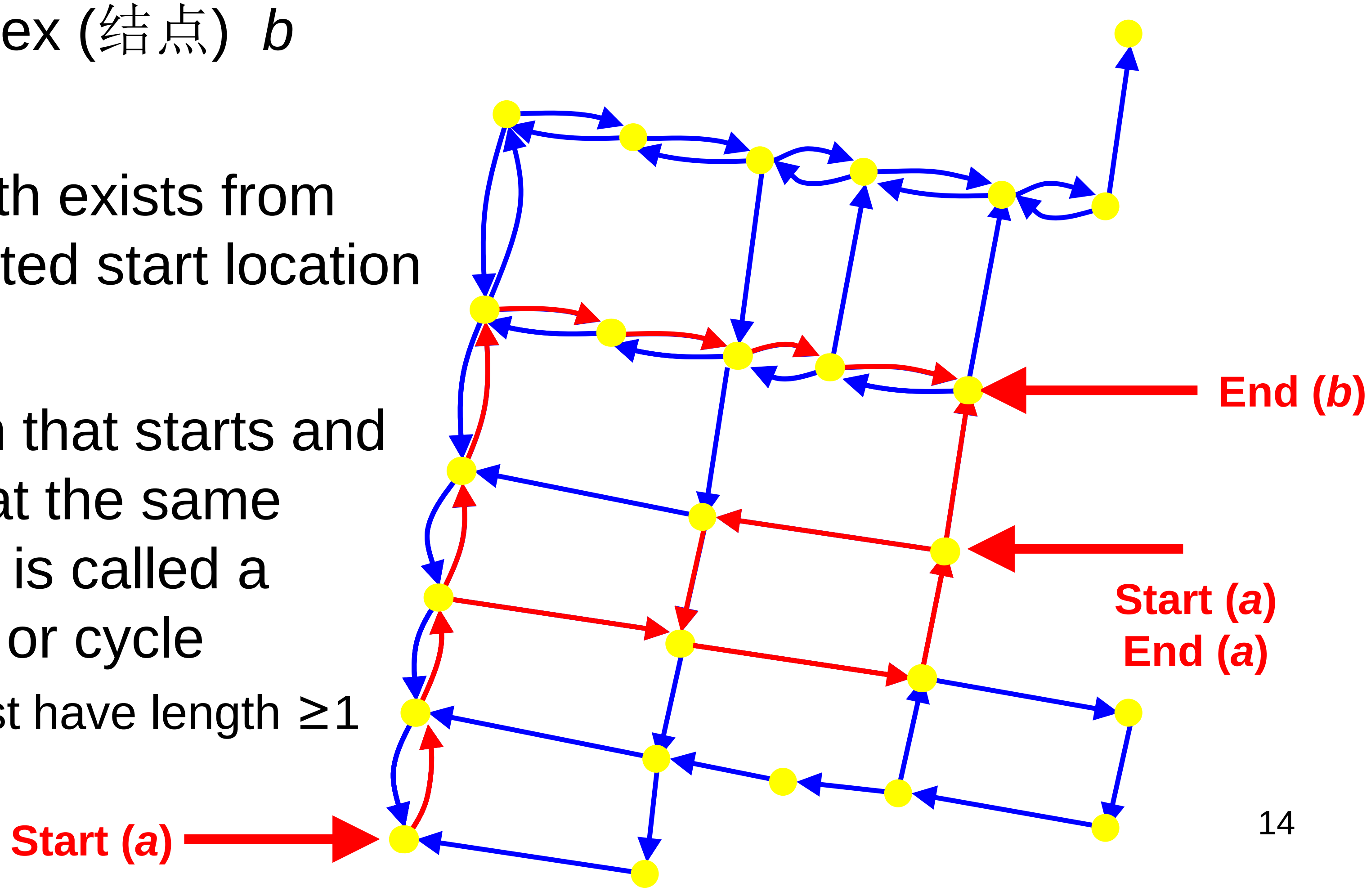
We add edges:
(0,2), (0,3)
(1,0), (2,1)

传递闭包问题

如何在给定的二元关系R的基础上，“适当扩大”直到获得其传递闭包?

Paths in directed graphs有向图中的路

- A *path* is a sequences of connected edges from vertex *a* to vertex (结点) *b*
- No path exists from the noted start location
- A path that starts and ends at the same vertex is called a circuit or cycle
  - Must have length ≥1



More on paths…

- The length of a path is the number of **edges** in the path, not the number of nodes （路长概念）

- *Note: "path" is a concept of graph theory, we will show much more detail in the chapter GRAPH*

Transitive closure传递闭包

- Formal definition: R is a binary relation on a set A, the transitive closure of R is a new relation R* which contains R, transitive, and for any transitive relation on A containing R is a superset of R*.

- $R \subseteq R^*$, R^* is transitive; If S is a transitive relation such that $R \subseteq S$, then $R^* \subseteq S$

- The transitive closure of R is the smallest transitive relation on A which contains R as its subset.

传递闭包是包含R的可传递的最小的二元关系；是R的满足可传递性的最小超集。

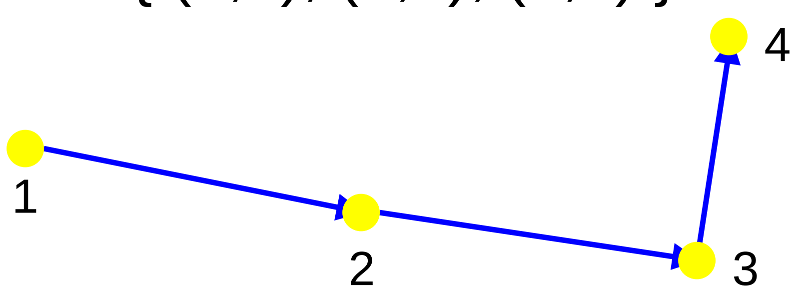
Finding Transitive closure 寻找传递闭包

- Informal definition: If there is a path from *a* to *b*, then there should be an edge from *a* to *b* in the transitive closure

如果有一条从a到b的路，则传递闭包里应该有从a到b的边（思考为什么？）

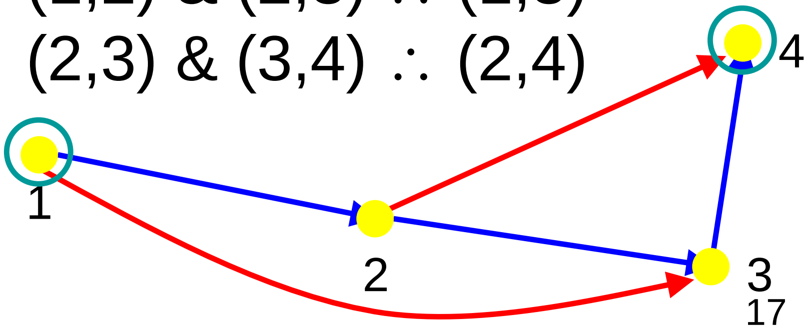
- First take of a definition:
  - In order to find the transitive closure of a relation *R*, we add an edge from *a* to *c*, whenever there are edges from a to *b* and *b* to *c*
- But there is a path from 1 to 4 with no edge!

$R = \{ (1,2), (2,3), (3,4) \}$



$(1,2)\ \&\ (2,3)\ \therefore\ (1,3)$

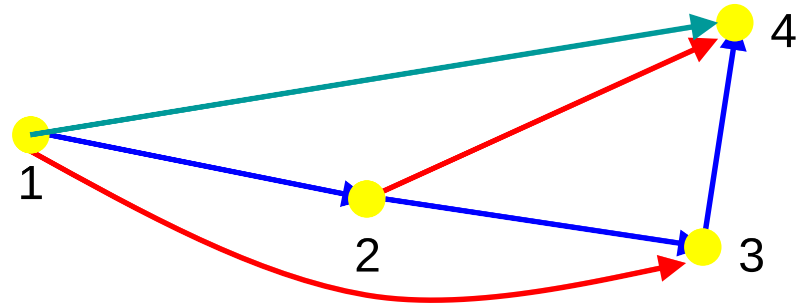
$(2,3)\ \&\ (3,4)\ \therefore\ (2,4)$



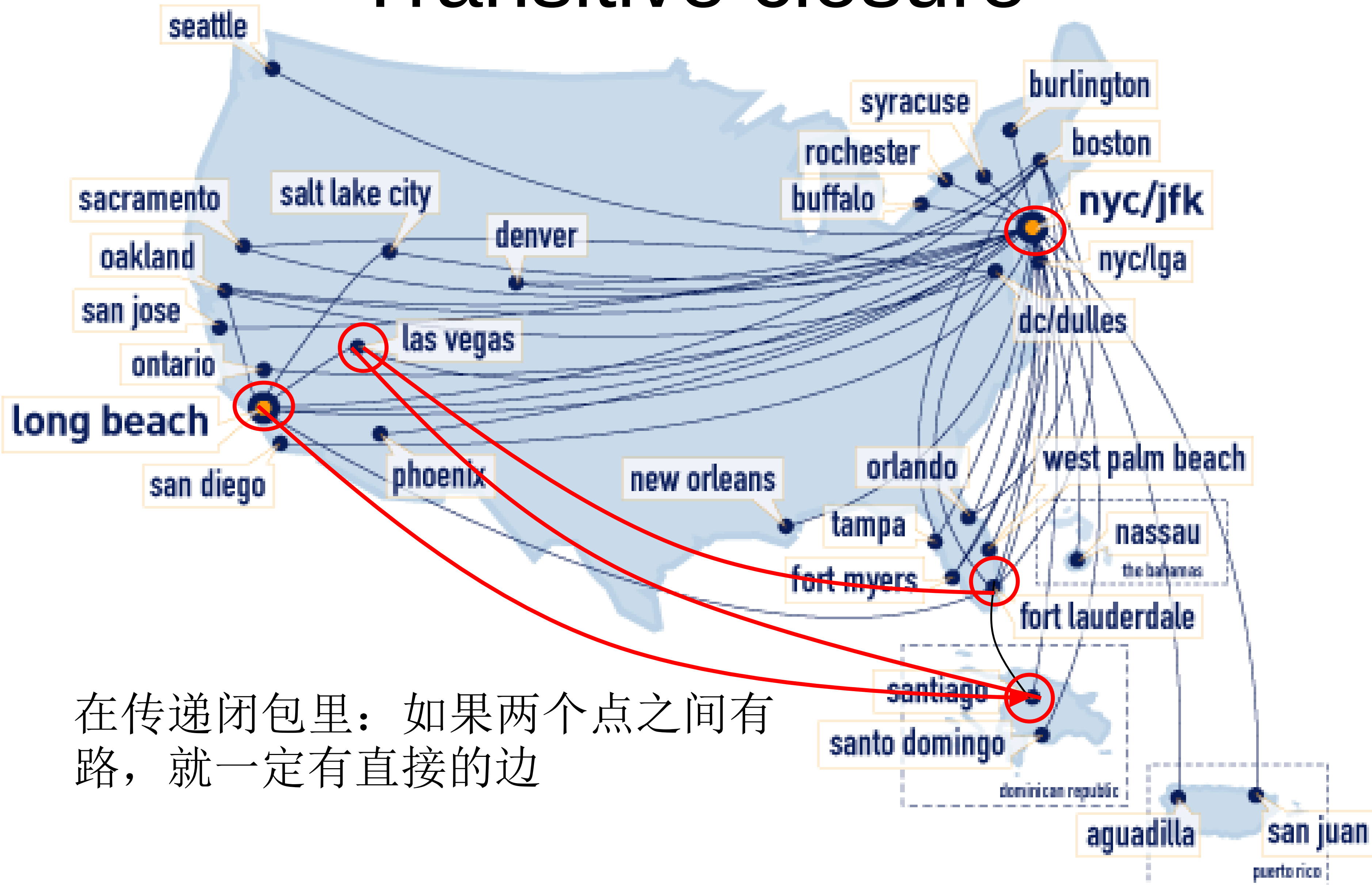
Transitive closure传递闭包

- 在传递闭包里面，如果有从点a到b的路，那么就一定有a到b的边
- Second take of a definition:
  - In order to find the transitive closure of a relation *R*, we add an edge from *a* to *c*, when there are edges from a to *b* and *b* to *c*
  - Repeat this step until no new edges are added to the relation

- red means added on the first repeat
- teal means added on the second repeat



Transitive closure



在传递闭包里：如果两个点之间有路，就一定有直接的边

Question 传递闭包存在性问题

- For any binary relation R on set A, is there transitive relation containing R?

- Is there transitive closure for binary relation on set A? Unique?

transitive closure传递闭包计算公式

- If R is a binary relation on non-empty set A, then the transitive closure of R is

$$R^* = \bigcup_{i=1}^{\infty} R^i$$

- If A is a finite set with n elements, then

$$R^* = \bigcup_{i=1}^{n} R^i$$

Think about why? Can you prove it is the minimum transitive relation including R as subset?

注意到:$R^* = \bigcup_{i=1}^{\infty} R^i = \bigcup_{i=1}^{n} R^i \cup \bigcup_{i=n+1}^{\infty} R^i$

Finding the transitive closure
利用关系矩阵计算传递闭包

- **Theorem**: Let $\mathbf{M}_R$ be the zero-one matrix of the relation R on a set A with n elements. Then the zero-one matrix of the transitive closure R^* is:

$$\mathbf{M}_{R^*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]} \vee \cdots \vee \mathbf{M}_R^{[n]}$$

Nodes reachable with one application of the relation

Nodes reachable with two applications of the relation

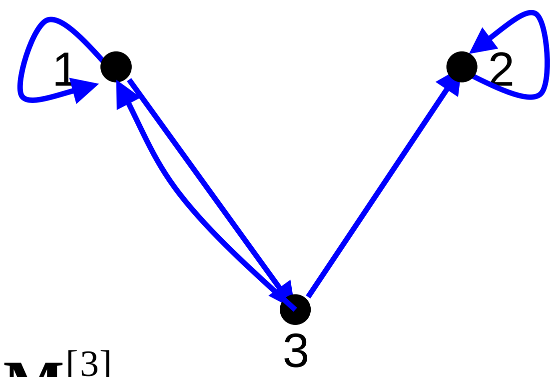
Nodes reachable with n applications of the relation

Please think about the amount of computing of finding transitive closure..

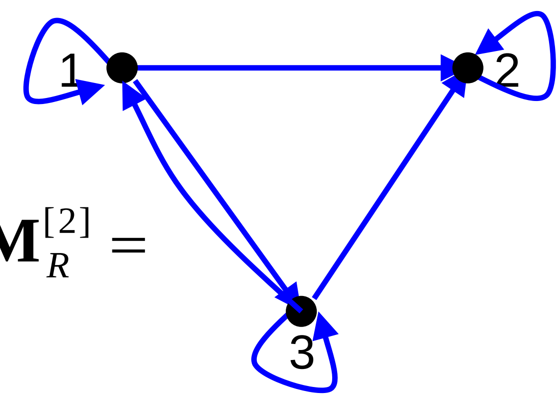
Close--example

- Find the zero-one matrix of the transitive closure of the relation R given by:

$$\mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$



$$\mathbf{M}_R^{[2]} =$$

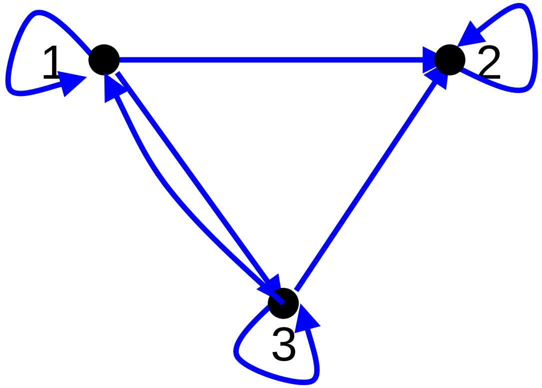


$$\mathbf{M}_{R*} = \mathbf{M}_R \vee \mathbf{M}_R^{[2]} \vee \mathbf{M}_R^{[3]}$$

$$\mathbf{M}_R^{[2]} = \mathbf{M}_R \odot \mathbf{M}_R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \odot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Close--example

$$\mathbf{M}_R^{[3]} = \mathbf{M}_R^{[2]} \odot \mathbf{M}_R = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \odot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\mathbf{M}_{R^*} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \vee \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Transitive closure algorithm 传递闭包算法

- What we did (or rather, could have done):
  - Compute the next matrix $\mathbf{M}_R^{[i]}$, where $1 \le i \le n$
  - Do a Boolean join with the previously computed matrix
- For our example:
  - Compute $\mathbf{M}_R^{[2]} = \mathbf{M}_R \circ \mathbf{M}_R$
  - Join that with $\mathbf{M}_R$ to yield $\mathbf{M}_R \vee \mathbf{M}_R^{[2]}$
  - Compute $\mathbf{M}_R^{[3]} = \mathbf{M}_R^{[2]} \circ \mathbf{M}_R$
  - Join that with $\mathbf{M}_R \vee \mathbf{M}_R^{[2]}$ from above

思考问题

- 计算有限集合上的关系R的传递闭包时，基于上面的公式，以及可传递的一些性质，能否将上面的算法在编程实现时改进如下，思考是否正确，为什么？
- 记M_1=M
- (1）在计算$(M_1)^2$时，只计算M中为0的那些项的位置对应的M_1^2 的项的值，然后用M_1^2与M_1求并，得到M_2;
- (2）再求$(M_2)^2$，类似于(1)， 只计算M_2中为0对应的位置的项的值；然后类似于(1)，求其于M_2的并，得到M_3;
- (3）在上面的计算过程中，如果不再有新的位置的值1的出现，则中止循环（why?），得到的就是传递闭包。
- (4）**当算到第k步 $(M_K)^2$时，该矩阵对应的关系已经是$R^1, R^2, R^3, \cdots R^{2^K}$,的超集了。所以，只要是$2^k$>=n,计算一定可以中止了。也即最多计算k=[$\log_2(n)$]步。为什么？**

思考问题续

- (5) 另外，假设关系R对应的最长的有向简单路的长度位t(t≤n)，那么上面的计算最多只需要到第k=[$\log_2$(t)]步。 当然不用去寻找这个t， 实际计算时，最多计算到[$\log_2$(t)]+1步后，一定能判断出不会再有1的增加。
- 在上面的矩阵乘积的计算中，由于只计算那些0项对应位置的项的值，节省了很多的计算量；
- （6）如果关系矩阵有全为0的行，则删除掉所有全为0的行和列,降低计算量。
-
- **评估计算复杂度，对比分析**沃舍尔算法的计算复杂度；
- 可以试着编写伪代码

无限集合上的有限二元关系的传递闭包

- 鼓励学生探索无限集合上的有限二元关系的传递闭包的计算。

- 假设S为无限集，R为S上的有限二元关系。 求R的传递闭包

思路：记A为关系R涉及到的所有元素的集合，则A必然是S的有限子集。R也就相当于A上的二元关系，在求其传递闭包时，不会涉及到A以外的元素。

Transitive closure algorithm
传递闭包算法伪代码

procedure *transitive_closure* (**M**$_R$: zero-one $n \times n$ matrix)

A := **M**$_R$

B := **A**

for i := 2 **to** n

begin

  A := **A** $\odot$ **M**$_R$

  B := **B** $\vee$ **A**

end { **B** is the zero-one matrix for R^* }

Warshall’s algorithm经典的沃舍尔算法*

- 也学习图论中的路的知识后再回头学习这个算法

- More efficient algorithms exist, such as Warshall’s algorithm, which is famous
  - not going to teach it in this class

  - To learn it please see the textbook, and think about why Warshall’s algorithm is good.

  - 实现时可以改进这个算法的计算：当计算第k个矩阵的项v_{ij}时，如果第k-1个矩阵的这个对应项为1，则不用再算，就直接等于第k-1个矩阵这个项；否则，就只需要计算出第k-1个矩阵的$v_{ik} \vee v_{kj}$就可以了。

思考问题

- 假设R是有限集合A上的一个二元关系，如何寻找一个同时满足自反性、对称性与可传递性的闭包？或者说寻找一个包含A的最小的同时满足自反性、对称性与可传递性的A上的二元关系？ （曾经的考题）

作业

- 5.4节
- T1
- T11 （在T5的基础上）
- T21， T23