More on Relations 3

Wolfgang Schreiner
Research Institute for Symbolic Computation (RISC-Linz)
Johannes Kepler University, Linz, Austria

Wolfgang.Schreiner@risc.uni-linz.ac.at
http://www.risc.uni-linz.ac.at/people/schreine

Directed Graphs

Overview

- Directed Graphs
- Paths and Reachability
- Trees

Directed Graphs

Definition: A directed graph is a pair $\langle V, E\rangle$ of a set V of vertices/nodes and a set of E of edges/arcs where E is binary relation on V:

$$
\begin{aligned}
&G \text{ is directed graph} :\Leftrightarrow \\
&\quad \exists V, E : \\
&\qquad G = \langle V, E\rangle \wedge \\
&\qquad E \subseteq V \times V.
\end{aligned}
$$

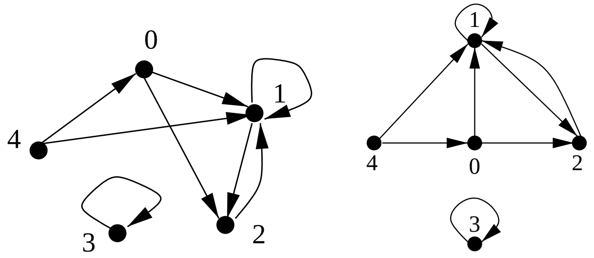
Interpretation: $\langle x, y\rangle \in E$

- x is connected to y in G,
- x is initial node of edge,
- y is terminal node of edge.

Example

Graph $\langle \mathbb{N}_5, E \rangle$

$$E = \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 3,3\rangle, \langle 4,0\rangle, \langle 4,1\rangle\}$$



The visual representation of a graph is not unique.

Example

Graph $\langle \mathbb{N}_5, E \rangle$

$$E = \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \langle 2,1\rangle, \langle 3,3\rangle, \langle 4,0\rangle, \langle 4,1\rangle\}$$

	0	1	2	3	4
0	false	true	true	false	false
1	false	true	true	false	false
2	false	true	false	false	false
3	false	false	false	true	false
4	true	true	false	false	false

Matrix representation is used for computing.

Adjacency Matrix

Definition: Let $G = \langle V, E\rangle$ be a directed graph with $|V| = n$. The adjacency matrix of G is the boolean $n*n$ matrix M where $M(x,y) =$ true if and only if $\langle x,y\rangle \in E$:

$$\begin{array}{l}
\text{adjacency}(G) := \\
\quad \textbf{let } V = G_0, E = G_1 : \\
\quad\quad \textbf{such } M \in V \times V \to \{\text{true}, \text{false}\} : \\
\quad\quad\quad (\forall x \in V, y \in V : M(x,y) = \text{true} \Leftrightarrow \langle x,y\rangle \in E).
\end{array}$$

Matrix representation from graph.

Undirected Graph

Definition: An undirected graph is a directed graph whose edge relation is symmetric:

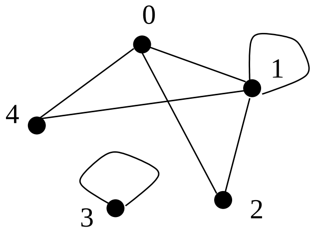
$$\begin{array}{l}
G \text{ is undirected graph} :\Leftrightarrow \\
\quad \exists V, E : \\
\quad\quad G = \langle V, E\rangle, \\
\quad\quad E \subseteq V \times V, \\
\quad\quad E \text{ is symmetric on } V.
\end{array}$$

Special kind of directed graph.

Example

Graph $\langle \mathbb{N}_5, E \rangle$

$$E = \{\langle 0,1\rangle, \langle 1,0\rangle, \langle 0,2\rangle, \langle 2,0\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \\ \langle 2,1\rangle, \langle 3,3\rangle, \langle 0,4\rangle, \langle 4,0\rangle, \langle 1,4\rangle, \langle 4,1\rangle\}$$



Draw single undirected edge instead of pair of directed edges.

Example

Graph $\langle \mathbb{N}_5, E \rangle$

$$E = \{\langle 0,1\rangle, \langle 1,0\rangle, \langle 0,2\rangle, \langle 2,0\rangle, \langle 1,1\rangle, \langle 1,2\rangle, \\ \langle 2,1\rangle, \langle 3,3\rangle, \langle 0,4\rangle, \langle 4,0\rangle, \langle 1,4\rangle, \langle 4,1\rangle\}$$

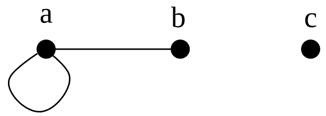
	0	1	2	3	4
0	false	true	true	false	true
1		true	true	false	true
2			false	false	false
3				true	false
4					false

Missing matrix elements are determined by symmetry.

Example

Graph $\langle \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}, E \rangle$

$$\mathsf{E} = \{\langle \mathsf{a}, \mathsf{a}\rangle, \langle \mathsf{a}, \mathsf{b}\rangle, \langle \mathsf{b}, \mathsf{a}\rangle\}$$



	a	b	c
a	true	true	false
b		false	false
c			false

Degree

Definition: In a directed graph, the indegree of x is the number of edges whose terminal node is x:

$$\mathrm{indeg}_G(x) := |\{y \in V : \langle y, x\rangle \in E\}|$$
$$\textbf{where } V = G_0, E = G_1.$$

The outdegree of x is the number of edges whose initial node is x:

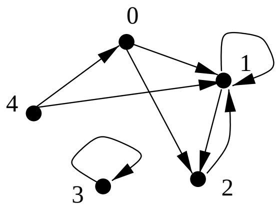
$$\mathrm{outdeg}_G(x) := |\{y \in V : \langle x, y\rangle \in E\}|$$
$$\textbf{where } V = G_0, E = G_1.$$

The total degree of x is the sum of its indegree and its outdegree:

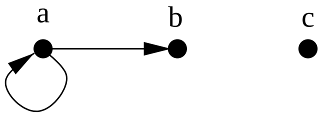
$$\mathrm{deg}_G(x) := \mathrm{indeg}_G(x) + \mathrm{outdeg}_G(x).$$

Example

- Indegree of node 1 is 4 and its outdegree is 2:



- Indegree and the outdegree of node c are both 0.



Graph Isomorphisms

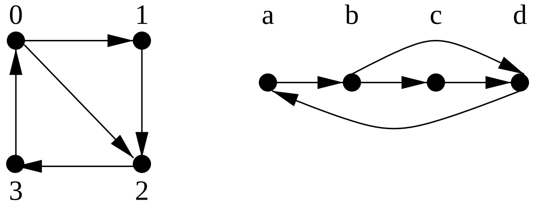
Definition: Two graphs are isomorphic if there exists a bijection between the nodes of the two graphs that preserves the edge structure:

$$\begin{array}{l} G \text{ and } G' \text{ are isomorphic} :\Leftrightarrow \\ \quad G \text{ is directed graph} \wedge G' \text{ is directed graph} \wedge \\ \quad \exists f : f : V \xrightarrow{\text{iso}(E,E')} V' \\ \qquad \textbf{where } V = G_0, E = G_1, V' = G'_0, E' = G'_1. \end{array}$$

Different graphs may have same structure.

Example

The graphs

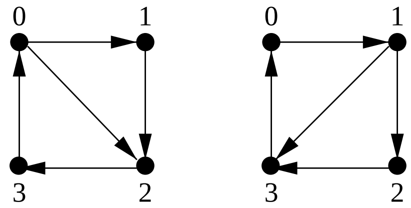


are isomorphic with isomorphism

$$f = \{\langle 0, \mathsf{b}\rangle, \langle 1, \mathsf{c}\rangle, \langle 2, \mathsf{d}\rangle, \langle 3, \mathsf{a}\rangle\}.$$

Example

The graphs

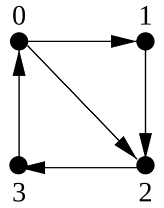


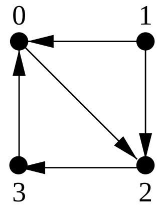
are isomorphic with isomorphism

$$f = \{\langle 0, 1\rangle, \langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 0\rangle\}.$$

Example

The graphs





are not isomorphic.

Paths and Reachability

Path

Definition: A path is a sequence of nodes connected by edges:

$$
\begin{array}{l}
p \text{ is path in } G :\Leftrightarrow \\
\quad (\exists n \in \mathbb{N}_{>0} : p : \mathbb{N}_n \to V \wedge \\
\qquad \forall i \in \mathbb{N}_{n-1} : \langle p_i, p_{i+1} \rangle \in E) \textbf{ where } V = G_0, E = G_1.
\end{array}
$$

The length of a path is the number of edges it contains:

$$\text{length}(p) := \textbf{such } n \in \mathbb{N} : \exists V : p : \mathbb{N}_{n+1} \to V.$$

A path from x to y has initial node x and terminal node y

$$
\begin{array}{l}
p \text{ is path from } x \text{ to } y :\Leftrightarrow \\
\quad p_0 = x \wedge p_n = y \textbf{ where } n = \text{length}(p).
\end{array}
$$

Path Properties

Definition: A path is simple if it does not contain any edge twice:

$$
\begin{array}{l}
p \text{ is simple} :\Leftrightarrow \\
\quad \forall i \in \mathbb{N}_n, j \in \mathbb{N}_n : \langle p_i, p_{i+1} \rangle = \langle p_j, p_{j+1} \rangle \Rightarrow i = j \\
\qquad \textbf{where } n = \text{length}(p).
\end{array}
$$

A path is elementary if it does not contain any node twice:

$$
\begin{array}{l}
p \text{ is elementary} :\Leftrightarrow \\
\quad (\forall i \in \mathbb{N}_n, j \in \mathbb{N}_n : p_i = p_j \Rightarrow i = j) \textbf{ where } n = 1 + \text{length}(p).
\end{array}
$$

A path is a cycle or circuit if it terminates in its initial node:

$$p \text{ is cycle} :\Leftrightarrow \exists x : p \text{ is path from } x \text{ to } x.$$

Reachability

Definition: A node y is reachable from a node x in a graph G if there is a path in G from x to y:

$$\begin{aligned}&y \text{ is reachable from } x \text{ in } G :\Leftrightarrow\\ &\quad \exists p : p \text{ is path in } G \wedge p \text{ is path from } x \text{ to } y.\end{aligned}$$

- For fixed G, "is reachable" is a binary relation on V.
- E is a binary relation on V.

We are going to construct the reachability relation from E.

Reflexive Closure

Definition: Let R be a binary relation on S. The reflexive closure of R on S is the smallest relation that contains R and is reflexive on S:

$$\begin{aligned}&\text{reflexive}_S(R) :=\\ &\quad \textbf{such } R' \subseteq S \times S :\\ &\quad\quad R \subseteq R' \wedge R' \text{ is reflexive on } S \wedge\\ &\quad\quad \forall R'' : (R \subseteq R'' \wedge R'' \text{ is reflexive on } S) \Rightarrow R' \subseteq R''.\end{aligned}$$

Proposition:

$$\forall S, R : R \subseteq S \times S \;\Rightarrow\; \text{reflexive}_S(R) = R \cup \{\langle x, x\rangle : x \in S\}.$$

Add reflexivity to relation.

Transitive Closure

Definition: The transitive closure of R on S is the smallest relation that contains R and is transitive on S:

$$\begin{aligned}&\text{transitive}_S(R) :=\\ &\quad \textbf{such } R' \subseteq S \times S :\\ &\quad\quad R \subseteq R' \wedge R' \text{ is transitive on } S \wedge\\ &\quad\quad \forall R'' : (R \subseteq R'' \wedge R'' \text{ is transitive on } S) \Rightarrow R' \subseteq R''.\end{aligned}$$

Add transitivity to relation (how?).

Reachability and Edge Relation

Proposition: We define the reachability relation

$$R_G := \{\langle x, y\rangle \in G_0 \times G_0 : y \text{ is reachable from } x \text{ in } G\}.$$

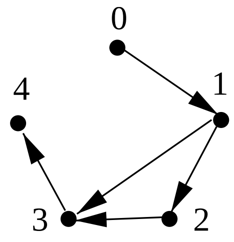
Then, for any directed graph $\langle V, E\rangle$, $R_{\langle V,E\rangle}$ is the reflexive and transitive closure of E on V:

$$\begin{aligned}&\forall V, E : \langle V, E\rangle \text{ is directed graph } \Rightarrow\\ &\quad R_{\langle V,E\rangle} = \text{reflexive}_V(\text{transitive}_V(E)).\end{aligned}$$

Problem reduced to computing the transitive closure of edge relation.

Example

Graph $\langle \mathbb{N}_5, E \rangle$ where $E = \{\langle 0,1\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 2,3\rangle, \langle 3,4\rangle\}$



$$\begin{aligned}&\mathrm{reflexive}_{\mathbb{N}_5}(E) = E \cup \{\langle 0,0\rangle, \langle 1,1\rangle, \langle 2,2\rangle, \langle 3,3\rangle, \langle 4,4\rangle\}\\ &\mathrm{transitive}_{\mathbb{N}_5}(E) =\\ &\quad \{\langle 0,1\rangle, \langle 0,2\rangle, \langle 0,3\rangle, \langle 0,4\rangle, \langle 1,2\rangle, \langle 1,3\rangle, \langle 1,4\rangle, \langle 2,3\rangle, \langle 2,4\rangle\}\end{aligned}$$

Exponentiation of Relations

Definition:

$$\begin{aligned}R^0_S &:= \{\langle x, x\rangle : x \in S\}\\ R^{i+1}_S &:= R^i_S \circ R.\end{aligned}$$

- R^0_S is the identity relation.
- R^1_S is R.
- $R^3_S = R \circ R \circ R$.

Repeated composition of relation.

Transitive Closure

Proposition:

$$\begin{aligned}&\forall S, R : R \subseteq S \times S \Rightarrow\\ &\quad \mathrm{transitive}_S(R) = \bigcup\{R^i_S : i \in \mathbb{N}_{>0}\}\end{aligned}$$

Interpretation: Transitive closure is union of

$$[R, R \cup (R \circ R), R \cup (R \circ R) \cup (R \circ R \circ R), \ldots]$$

Problem: cannot compute infinite sequence!

Transitive Closure

Proposition: Let R be a binary relation on S where S has n elements. Then $\bigcup_{1 \le i \le n} R^i_S$ is the transitive closure of R:

$$\begin{aligned}&\forall S, R : R \subseteq S \times S \;\Rightarrow\\ &\quad \mathrm{transitive}_S(R) = \textstyle\bigcup_{1 \le i \le n} R^i_S\\ &\quad \textbf{where } n = |S|.\end{aligned}$$

Interpretation: Transitive closure is union of

$$[R, R \cup (R \circ R), \ldots, R \cup (R \circ R) \cup \ldots \cup R^n]$$

Constructive method for computing transitive closure.

Reachability

Algorithm:

$$\begin{array}{l} \text{reachability}(V, E): \\ \quad n = |V| \\ \quad R^0 = \{\langle x, x\rangle : x \in V\} \\ \quad \textbf{for}(i = 0; i < n; i{+}{+}) \\ \qquad R^{i+1} = R^i \cup (R^i \circ E) \\ \quad \textbf{return } R^n \end{array}$$

Can compute reachability relation from edge relation.

Composition of Relations

Let R and S be binary relations on $\mathbb{N}_n$ for some $n \in \mathbb{N}$.

The composition of R and S is

$$R \circ S = \{\langle a, c\rangle : a \in \mathbb{N}_n \wedge c \in \mathbb{N}_n \wedge (\exists b : \langle a, b\rangle \in R \wedge \langle b, c\rangle \in S)\}.$$

For the corresponding adjacency matrix, we thus have

$$\begin{array}{l} \forall i \in \mathbb{N}_n, j \in \mathbb{N}_n : \\ \quad \text{adjacency}(\langle \mathbb{N}_n, R \circ S\rangle)_{i,j} = \mathsf{true} \Leftrightarrow \\ \qquad \exists k \in \mathbb{N}_n : A_{i,k} = \mathsf{true} \wedge B_{k,j} = \mathsf{true} \\ \qquad \textbf{where } A = \text{adjacency}(\langle \mathbb{N}_n, R\rangle), B = \text{adjacency}(\langle \mathbb{N}_n, S\rangle). \end{array}$$

Composition of Relations

Written as a Java method, the composition of two adjacency matrices A and B giving a result matrix C resembles matrix multiplication:

```
void compose(int n, boolean[][] A, boolean[][] B, boolean[][] C)
{
  for (int i=0; i<n; i++)
    for (int j=0; j<n; j++)
    {
      C[i][j] = false;
      for (int k=0; k<n; k++)
        C[i][j] = C[i][j] || (A[i][k] && B[k][j]);
    }
}
```


Trees

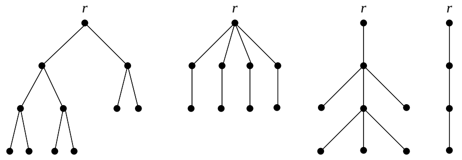
Tree

Definition: A tree is a directed graph such that there is exactly one node, the root, that has indegree zero, every other node has indegree one, and every node can be reached from the root.

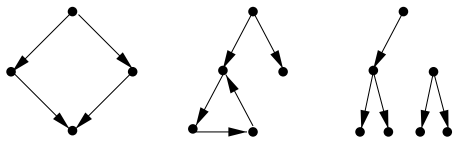
$$
\begin{array}{l}
T \text{ is tree} :\Leftrightarrow \\
\quad T \text{ is directed graph} \land \\
\quad (\exists r \in V : \text{indeg}(r) = 0 \land \\
\quad\quad \forall x \in V - \{r\} : \\
\quad\quad\quad \text{indeg}(x) = 1 \land \\
\quad\quad\quad x \text{ is reachable from } r \text{ in } T) \textbf{ where } V = T_0. \\
\text{root}(T) := (\textbf{such } r \in V : \text{indeg}(r) = 0) \textbf{ where } V = T_0.
\end{array}
$$

Example

- The following diagrams depict trees with root r:

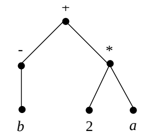


- The following directed graphs are not trees:



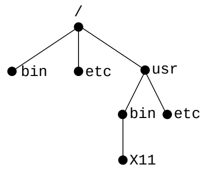
Example

- The term $-b + 2a$:



- A file system with directories

`/`, `/bin`, `/etc`, `/usr`, `/usr/bin`, `/usr/bin/X11`, `/usr/etc`



Trees and Cycles

Proposition: A tree has only cycles of length 0:

$$
\begin{array}{l}
\forall T : T \text{ is tree } \Rightarrow \\
\quad \neg(\exists p : p \text{ is path in } T \land \text{length}(p) > 0 \ \land p \text{ is cycle}).
\end{array}
$$

No (non-trivial) path in a tree is a cycle.

Parents and Children

Definition: Let T be a tree. A node y is called a child of x if there is an edge from x to y in T:

$$\begin{aligned}&y \text{ is child of } x \text{ in } T :\Leftrightarrow\\ &\quad \langle x, y\rangle \in E \textbf{ where } E = T_1.\end{aligned}$$

x is then called the parent of y:

$$\begin{aligned}&\text{parent}_T(y) := \textbf{such } x \in V : \langle x, y\rangle \in E\\ &\quad \textbf{where } V = T_0, E = T_1.\end{aligned}$$

Every node (apart from the root) has a unique parent.

Other Tree Relations

Definition: A node x is a leaf, if it does not have children:

$$\begin{aligned}&x \text{ is leaf in } T :\Leftrightarrow x \in V \land \neg\exists y : y \text{ is child of } x \text{ in } T\\ &\quad \textbf{where } V = T_0.\end{aligned}$$

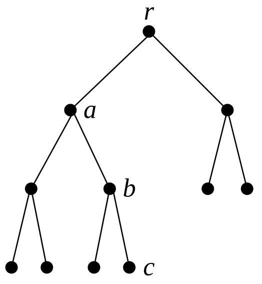
A node x is an ancestor of y if there is a path from x to y in T:

$$\begin{aligned}&x \text{ is ancestor of } y \text{ in } T :\Leftrightarrow\\ &\quad \exists p : p \text{ is path in } T \land p \text{ is path from } x \text{ to } y.\end{aligned}$$

y is then called a descendant of x:

$$y \text{ is descendant of } x \text{ in } T :\Leftrightarrow x \text{ is ancestor of } y \text{ in } T.$$

Example



a is the parent of b and an ancestor of leaf c.

Levels and Heights

Definition: The level of a node x in a tree is the length of the path from the root of the tree to x:

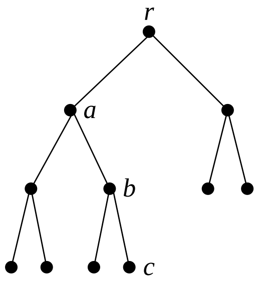
$$\begin{aligned}&\text{level}_T(x) := \text{length}(p) \textbf{ where } p =\\ &\quad \textbf{such } p : p \text{ is path in } T \land p \text{ is path from root}(T) \text{ to } x.\end{aligned}$$

The height of a tree is the maximum level of its nodes:

$$\text{height}(T) := \max\{\text{level}_T(x) : x \in V\} \textbf{ where } V = T_0.$$

Root has level 0; level of every other node is one plus the parent level.

Example



a has level 1, b has level 2, c has level 3. The height of the tree is 3.

Binary Trees

Proposition: Let T be a tree of height h where every node has an outdegree of at most 2. The number of tree nodes is less than 2^{h+1}:

$$\forall T : (T \text{ is tree} \wedge \forall x \in V : \text{outdeg}(x) \leq 2) \Rightarrow |V| < 2^{h+1}$$
$$\textbf{where } V = T_0, h = \text{height}(T).$$

Proof: Let T be a such a tree. We proceed by complete induction on the height of T.

1. Assume the height is $h = 0$. Then $|V| = 1 < 2 = 2^{h+1}$.
2. Assume the height is $h > 0$. Consequently the root of T has a child that is the root of a tree of height $h - 1$ and possibly a second child that is the root of a tree of height less than or equal $h - 1$. By the induction hypothesis, we thus have

$$|V| \leq 1 + (2^h - 1) + (2^h - 1) = 2^{h+1} - 1 < 2^{h+1}.$$

Summary

- Directed graphs
  - Pair of node set and edge relation.
  - Adjacency matrix.
  - Degree.
- Paths and reachability
  - Closure of relations.
  - Reachability is closure of edge relation.
- Trees
  - Root, parent, children.
  - Levels and heights.