informatyka+

Wszechnica Popołudniowa: Algorytmika i programowanie

O relacjach i algorytmach

Zenon Gniazdowski

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

O relacjach i algorytmach

Relacja jest podstawowym pojęciem matematycznym, również użytecznym w informatyce:

- ✓ Operatory relacji $=$, $\neq$, $<$, $\leq$, $>$, $\geq$ w językach programowania.
- ✓ Relacyjne bazy danych.
- ✓ W eksploracji danych:
  - skale pomiarowe.
  - relacja nierozróżnialności w teorii zbiorów przybliżonych

O relacjach i algorytmach

- *W wykładzie zostaną omówione **relacje dwuczłonowe**, a także sposoby ich reprezentacji w postaci **macierzy** lub **grafu**.*
- *Grafy zostaną użyte do pokazania relacji w postaci przemawiającej do wyobraźni. Tymczasem postać macierzowa umożliwi konstrukcję algorytmów do badania własności relacji.*
- *Zostaną także pokazane przykłady relacji wraz z przedstawieniem ich własności i określeniem ich typów.*

Iloczyn kartezjański zbiorów

Rozważa się dwa zbiory $\boldsymbol{X}$ i $\boldsymbol{Y}$.

Zbiór wszystkich uporządkowanych par elementów należących odpowiednio do tych zbiorów, nazywa się **iloczynem** (albo **produktem**) **kartezjańskim** zbiorów $\boldsymbol{X}$ i Y.

Iloczyn kartezjański oznacza się jako $\boldsymbol{X}\times\boldsymbol{Y}$. Można to zapisać w następujący sposób:

$$\boldsymbol{X\times Y=\{(x,y):\ x\in X\ i\ y\in Y\}}$$

Jeżeli dodatkowo zachodzi równość $\boldsymbol{X=Y}$, to zamiast $\boldsymbol{X\times X}$ można napisać $\boldsymbol{X^2}$.

Iloczyn kartezjański zbiorów

Przykład: Iloczyn kartezjański zbiorów ***X={1,2,3,4}***, ***Y={a,b,c}***:

$$X \times Y = \{1, 2, 3, 4\} \times \{a, b, c\} = \begin{Bmatrix} (1, a) & (1, b) & (1, c) \\ (2, a) & (2, b) & (2, c) \\ (3, a) & (3, b) & (3, c) \\ (4, a) & (4, b) & (4, c) \end{Bmatrix}$$

Jest to zbiór wszystkich par (cyfra, litera).

Operacja iloczynu kartezjańskiego nie jest przemienna:

$$X \times Y \neq Y \times X$$

Iloczyn kartezjański zbiorów

Przykład ten można także zinterpretować graficznie.

Jeżeli w tabeli wiersze oznaczymy elementami jednego zbioru, a kolumny – elementami drugiego zbioru, to punkt na przecięciu odpowiedniego wiersza i kolumny reprezentuje parę ***(cyfra, litera).***

	a	b	c
1	■	■	■
2	■	■	■
3	■	■	■
4	■	■	■

Iloczyn kartezjański zbiorów

Innym przykładem iloczynu kartezjańskiego jest zbiór punktów na płaszczyźnie ***OXY***, oznaczany jako $\boldsymbol{R}^2$.

Pojęcie iloczynu kartezjańskiego można rozszerzyć na większą liczbę wymiarów.

Dla przykładu, iloczyn $\boldsymbol{R \times R \times R = R^3}$ oznacza zbiór punktów w przestrzeni trójwymiarowej.

Macierz

Kolejnym przykładem iloczynu kartezjańskiego jest zbiór indeksów elementów macierzy.

Macierz jest skończonym zbiorem elementów, zapisywanym w postaci prostokątnej tablicy o ***m*** wierszach i ***n*** kolumnach:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Macierz

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

Adres elementu w macierzy składa się z dwóch liczb, z których pierwsza wskazuje na numer wiersza, a druga na numer kolumny, w których ten element się znajduje.

Na przykład $\boldsymbol{a_{24}=1}$ oznacza, że element znajdujący się w drugim wierszu i czwartej kolumnie macierzy jest równy **1**.

Macierz

Macierz kwadratowa, macierz prostokątna

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 0 & 1 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 4 & 7 & 0 \\ 2 & 5 & 8 & 1 \\ 3 & 6 & 9 & 2 \end{bmatrix}$$

Macierz

Macierz transponowana

Transpozycja nie zmienia elementów na głównej przekątnej

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, A^T = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

$$B = C^T, C = B^T :$$

$$B = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 0 & 1 & 2 \end{bmatrix}, C = \begin{bmatrix} 1 & 4 & 7 & 0 \\ 2 & 5 & 8 & 1 \\ 3 & 6 & 9 & 2 \end{bmatrix}$$

Macierz

Macierz symetryczna

$$S = S^T = \begin{bmatrix} 1 & 6 & 5 \\ 6 & 2 & 4 \\ 5 & 4 & 3 \end{bmatrix}$$

Relacja dwuczłonowa

Dla danych zbiorów ***X*** i ***Y*** relacją dwuczłonową na iloczynie kartezjańskim ***X×Y*** jest **dowolny podzbiór** tego iloczynu.

Relacja dwuczłonowa

Przykład: Szkoła uczy komputerowych metod kroju i szycia.

Zbiór nazwisk nauczycieli:

***X*={*Kowalski, Nowak, Jankowski, Kaszubski, Góralski, Kurpiowski*}**

Zbiór przedmiotów nauczanych w pierwszym semestrze:

***Y*={*Krój, szycie, fastrygowanie, krój komputerowy, wyszywanie komputerowe*}**

Relacja dwuczłonowa

Oto przydział nauczycieli do przedmiotów:

R={(***Kowalski***, ***krój***),

(***Kowalski***, ***fastrygowanie***),

(***Jankowski***, ***wyszywanie komputerowe***),

(***Kaszubski***, ***szycie***),

(***Góralski***, ***krój komputerowy)***,

(***Kurpiowski***, ***krój***),

(***Kurpiowski***, ***fastrygowanie***),

(***Kurpiowski***, ***wyszywanie komputerowe***)}

Relacja dwuczłonowa

Przydział ten jest przykładem relacji. Iloczyn kartezjański zawierałby **30** par (**nazwisko**, **przedmiot**), tymczasem przedstawiona relacja zawiera tylko **8** par.

Reprezentacja relacji - macierz

Rozważa się dwa zbiory: zbiór $\boldsymbol{X=\{x_1,x_2,\ldots,x_m\}}$ składający się z ***m*** elementów oraz zbiór $\boldsymbol{Y=\{y_1,y_2,\ldots,y_n\}}$ zawierający ***n*** elementów.

Niech na iloczynie kartezjańskim $\boldsymbol{X \times Y}$ będzie określona pewna relacja **ρ**.

Relacja ta może być reprezentowana w postaci **macierzy**. Wiersze macierzy odpowiadają kolejnym elementom zbioru ***X***, zaś kolumny – elementom zbioru ***Y***.

Reprezentacja relacji - macierz

Macierzą relacji ρ jest zerojedynkowa macierz [$\boldsymbol{R_{ij}}$], zawierająca ***m*** wierszy i ***n*** kolumn. Jej elementy określone są następującym wzorem:

$$R_{ij} = \begin{cases} 1, & gdy \quad x_i \rho\, y_j \\ 0, & gdy \quad \neg\left(x_i \rho\, y_j\right) \end{cases}$$

gdzie ***i*=1, 2, …, *m***, ***j*=1, 2, …, *n***.

Reprezentacja relacji - macierz

Jeżeli $\boldsymbol{X}=\boldsymbol{Y}$, to relacją w zbiorze $\boldsymbol{X}$ jest pewien podzbiór iloczynu kartezjańskiego $\boldsymbol{X}\times\boldsymbol{X}$.

W dalszej części wykładu będą rozważane wyłącznie relacje w $\boldsymbol{n}$-elementowym zbiorze $\boldsymbol{X}$, reprezentowane przez kwadratową macierz o rozmiarze $\boldsymbol{n}\times\boldsymbol{n}$.

Reprezentacja relacji - macierz

Przykład 1: Relacja **ρ** jest określona w zbiorze ***X={1,2,3,4,5,6}***.

$\forall x, y \in X \quad x \, \rho \, y \Leftrightarrow x+y$ ***jest liczbą złożoną***

m \ n	1	2	3	4	5	6
1	0	0	1	0	1	0
2	0	1	0	1	0	1
3	1	0	1	0	1	1
4	0	1	0	1	1	1
5	1	0	1	1	1	0
6	0	1	1	1	0	1

Reprezentacja relacji - graf

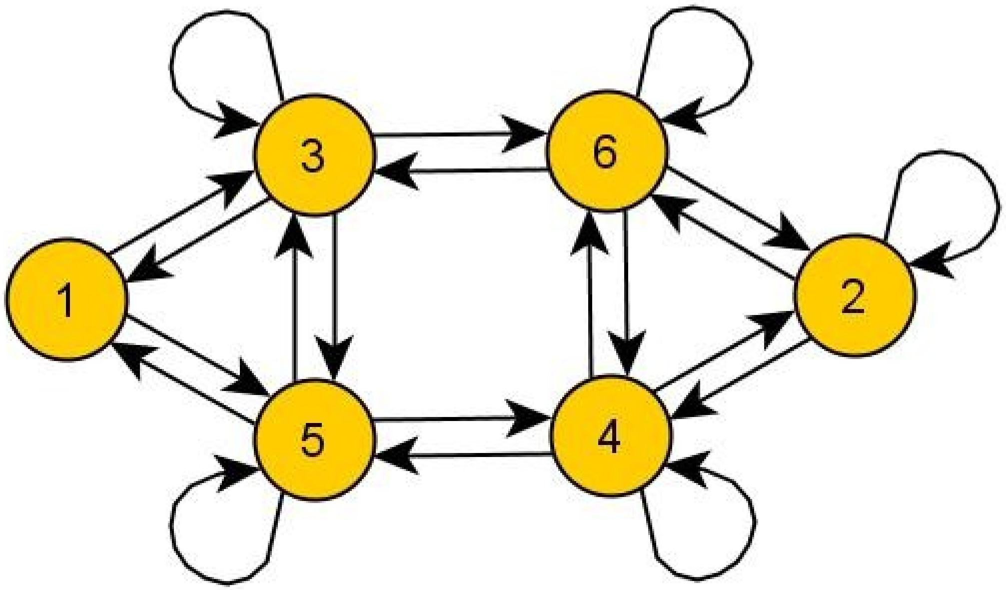
Jeżeli elementy zbioru potraktować, jako węzły grafu, to zdarzenie, iż element ***i*-ty** jest w relacji z elementem ***j*-tym**, można przedstawić przy pomocy łuku skierowanego od węzła ***i*** do węzła ***j***.

Otrzymany graf jest **grafem skierowanym**, zaś macierz relacji jest **macierzą sąsiedztwa grafu**.

Reprezentacja relacji - graf

X={1,2,3,4,5,6}, ∀x, y∈X   x ρ y ⇔ x+y jest liczbą złożoną

$$R=\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$



Własności relacji

- Relacja dwuczłonowa może być **zwrotna**, **przeciwzwrotna**, **symetryczna**, **antysymetryczna**, **spójna**, **przechodnia**.
- Własności relacji ujawnią się **w macierzy** lub **w grafie**.
- Korzystając z reprezentacji macierzowej można **algorytmizować** badanie relacji.
- Reprezentacja grafowa wpływa na wyobraźnię i ułatwia proces tworzenia algorytmu.

Własności relacji

Do tworzenia algorytmów zostanie użyty język C++.

- Elementy macierzy **1** albo **0** można traktować, jako logiczne wartości ***prawda*** albo ***fałsz***.
- Dla elementów macierzy mogą być stosowne operacje arytmetyczne lub logiczne.
- W C++ *n* wierszy i *n* kolumn tablicy dwuwymiarowej są indeksowane liczbami od 0 do *n* - 1.

Własności relacji - zwrotność

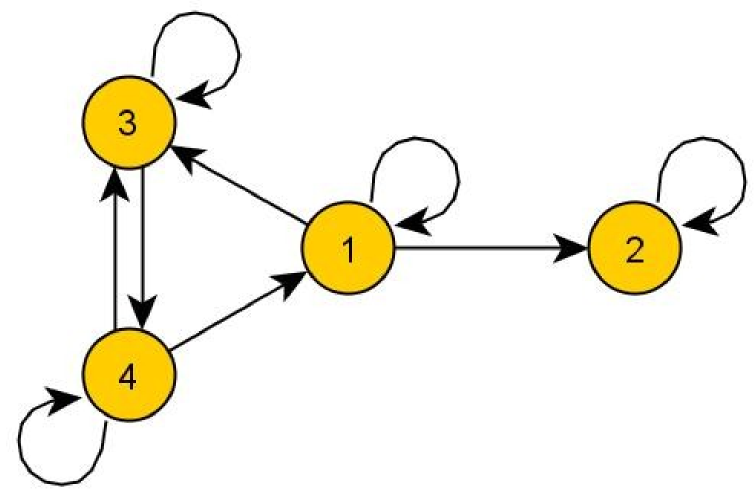
Relacja ρ określona w zbiorze ***X*** jest zwrotna, gdy dla każdego elementu $\boldsymbol{x \in X}$ element ten pozostaje w relacji z samym sobą.

$$\forall \boldsymbol{x} \in \boldsymbol{X}\boldsymbol{:}\ \boldsymbol{x} \rho \boldsymbol{x}$$

Własności relacji - zwrotność

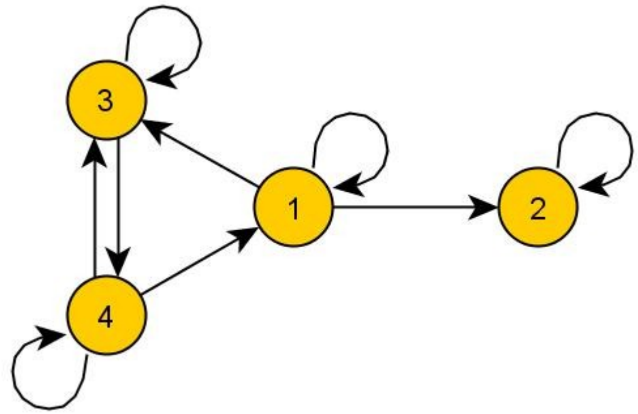
W zapisie macierzowym zwrotność przejawia się tym, że wszystkie elementy znajdujące się na przekątnej macierzy ***R*** są równe ***1***. Oznacza to, że w każdym węźle grafu relacji znajduje się pętla.

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$



Własności relacji - zwrotność

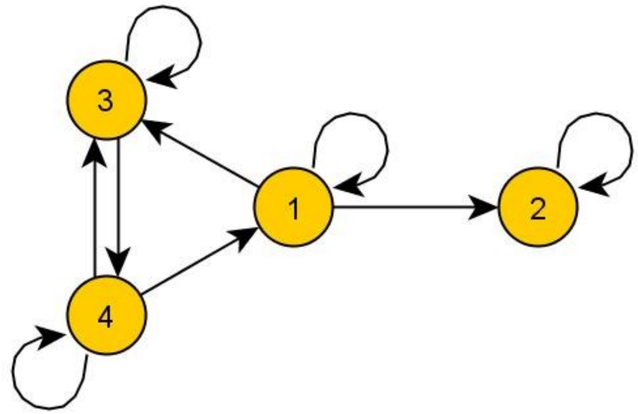
$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$



Algorytm badania zwrotności powinien sprawdzać, **czy na przekątnej macierzy znajdują się same jedynki**. **Jeżeli** pojawi się **co najmniej jedno zero, to relacja nie jest** relacją **zwrotną**.

Własności relacji - zwrotność

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$



Algorytm badania zwrotności:

```
int zwrotna(int R[SIZE][SIZE],int n)
{  int i;
   for (i=0; i<n; i++)
      if (R[i][i] == 0) return 0;
   return 1;
}
```

Własności relacji - przeciwzwrotność

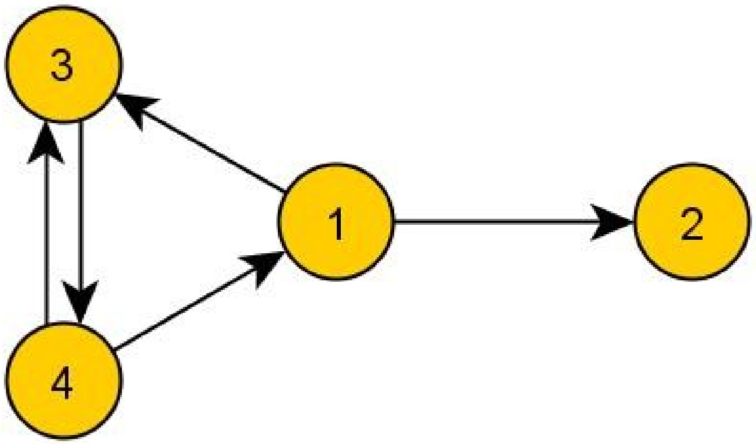
Relacja ρ określona w zbiorze ***X*** jest przeciwzwrotna, gdy żaden element $\boldsymbol{x} \in \boldsymbol{X}$ nie jest w relacji z samym sobą.

$$\forall \boldsymbol{x} \in \boldsymbol{X}: \neg(\boldsymbol{x} \rho \boldsymbol{x})$$

Własności relacji - przeciwzwrotność

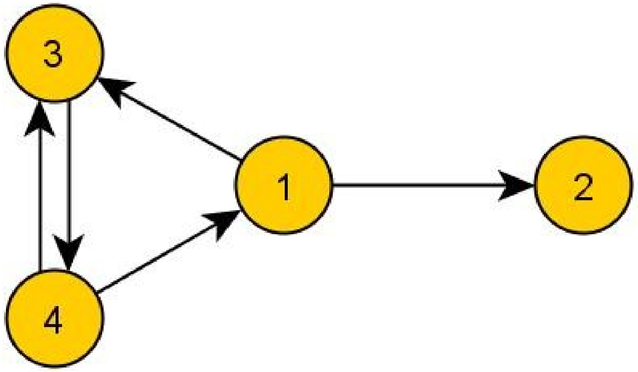
W zapisie macierzowym przeciwzwrotność przejawia się tym, że wszystkie elementy znajdujące się na przekątnej macierzy ***R*** są równe **0**. Oznacza to, że w żadnym węźle grafu relacji nie ma pętli.

$$R = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Własności relacji - przeciwzwrotność

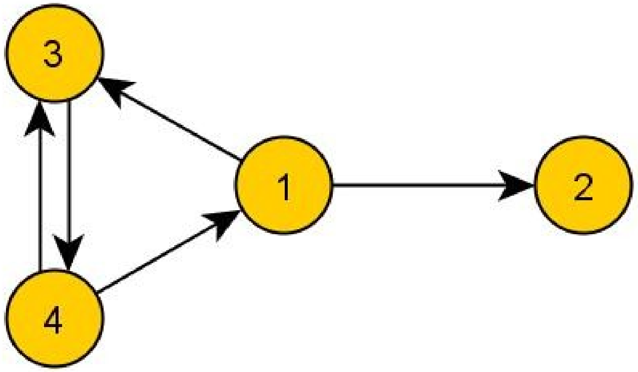
$$R = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania zwrotności powinien sprawdzać, czy **na przekątnej macierzy** znajdują się **same zera**. Jeżeli pojawi się **co najmniej jedna jedynka**, to relacja **nie jest** relacją **przeciwzwrotną**.

Własności relacji - przeciwzwrotność

$$R = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania przeciwzwrotności:

```
int przeciwzwrotna(int R[SIZE][SIZE],int n)
{  int i;
   for (i=0; i<n; i++)
      if (R[i][i]) return 0;
   return 1;
}
```

Własności relacji - symetria

Relacja ρ określona w zbiorze ***X*** jest symetryczna, gdy dla każdych dwóch elementów $x,y \in X$, z faktu, że ***x*** jest w relacji z ***y*** wynika, że ***y*** jest w relacji z ***x***.

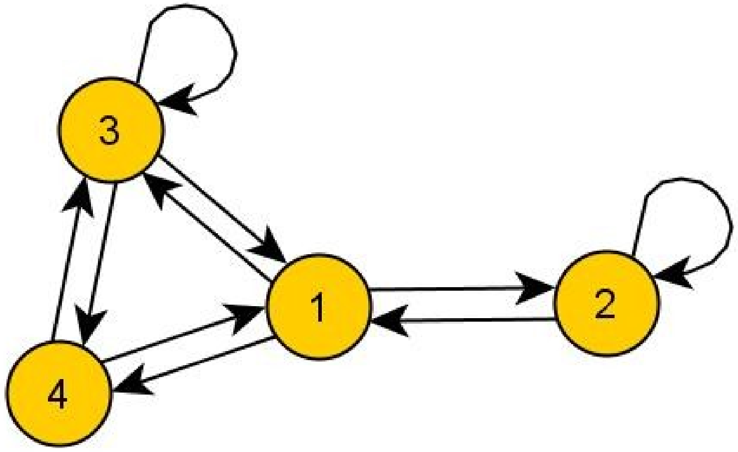
$$\forall x,y \in X:\ x \rho y \Rightarrow y \rho x$$

Własności relacji - symetria

W zapisie macierzowym symetria przejawia się w symetrii macierzy ***R***.

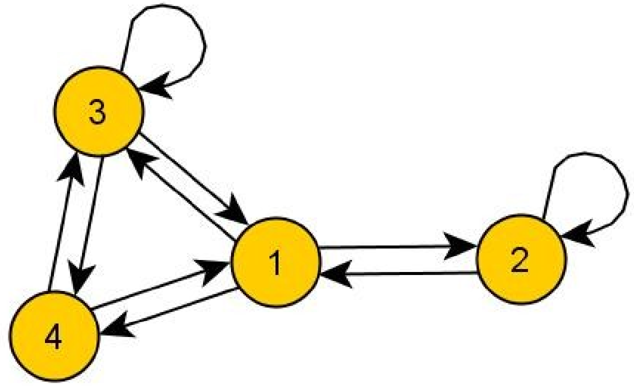
W grafie **wszystkie łuki** między dwoma różnymi węzłami **biegną w dwóch kierunkach**.

$$R = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Własności relacji - symetria

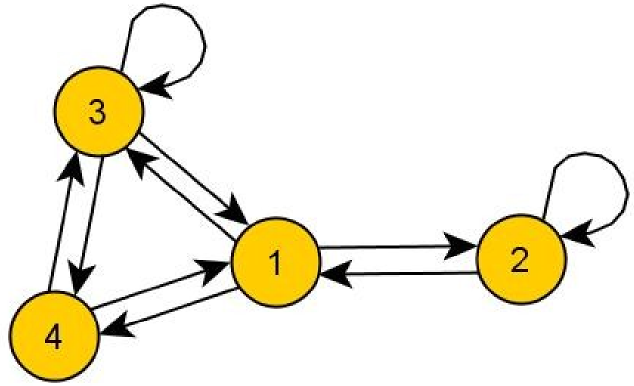
$$R = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania symetrii polega na przechodzeniu wszystkich elementów $\boldsymbol{R_{ij}}$ znajdujących się ponad główną przekątną macierzy i sprawdzaniu, czy są im równe elementy $\boldsymbol{R_{ji}}$ leżące pod przekątną. Jeżeli pojawi się $\boldsymbol{R_{ij} \neq R_{ji}}$, to relacja nie jest symetryczna.

Własności relacji - symetria

$$R = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania symetrii:

```
int symetryczna(int R[SIZE][SIZE],int n)
{  int i,j;
   for (i=0; i<n–1; i++)
      for (j=i+1; j<n; j++)
         if (R[i][j] != R[j][i]) return 0;
   return 1;
}
```

Własności relacji - antysymetria

Relacja ρ określona w zbiorze ***X*** jest antysymetryczna, jeżeli dla każdych dwóch elementów ***x*, *y*** $\in$ ***X*** z faktu, że ***x*** jest w relacji z ***y*** i ***y*** jest w relacji z ***x***, wynika, że elementy ***x*** i ***y*** są identyczne:

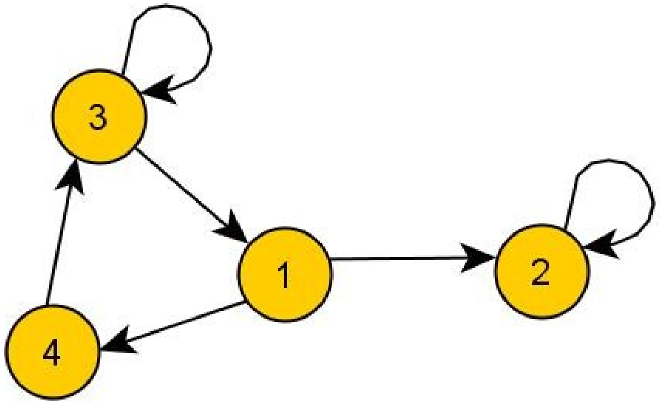
$$\forall x,y \in X:\ x\rho y \wedge y\rho x \Rightarrow x=y$$

Własności relacji - antysymetria

W macierzy relacji antysymetrycznej każdemu elementowi $\boldsymbol{R_{ij}=1}$ spoza przekątnej, towarzyszy element $\boldsymbol{R_{ji}=0}$: $\forall(\boldsymbol{i,j})_{i\neq j}\, \boldsymbol{R_{ij}} \wedge \boldsymbol{R_{ji}=0}$.

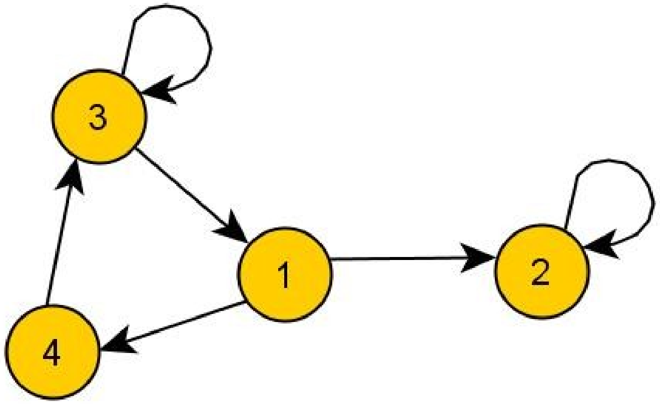
Jeżeli między dwoma różnymi węzłami istnieje łuk w grafie, to **nie towarzyszy** mu łuk w przeciwną stronę.

$$R = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Własności relacji - antysymetria

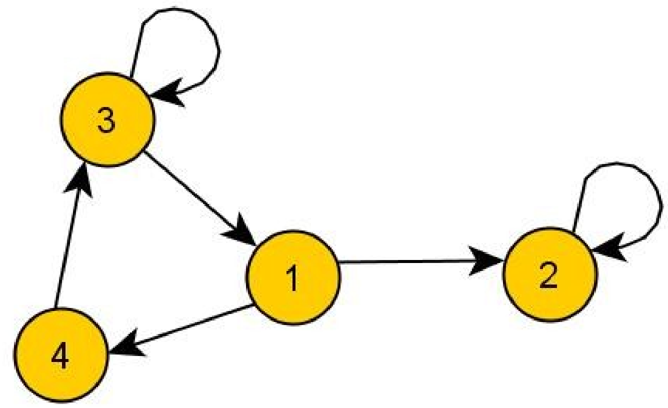
$$R = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania antysymetrii polega na przechodzeniu przez wszystkie elementy macierzy znajdujące się ponad przekątną i sprawdzaniu, czy $\boldsymbol{R_{ij} \wedge R_{ji} = 0}$. Jeżeli ten warunek nie jest spełniony co najmniej jeden raz, to relacja nie jest antysymetryczna.

Własności relacji - antysymetria

$$R = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Algorytm badania antysymetrii:

```
int antysymetryczna(int R[SIZE][SIZE],int n)
{  int i,j;
   for (i=0; i<n–1; i++)
      for (j=i+1; j<n; j++)
         if (R[i][j] && R[j][i]) return 0;
   return 1;
}
```

Własności relacji - przeciwsymetria

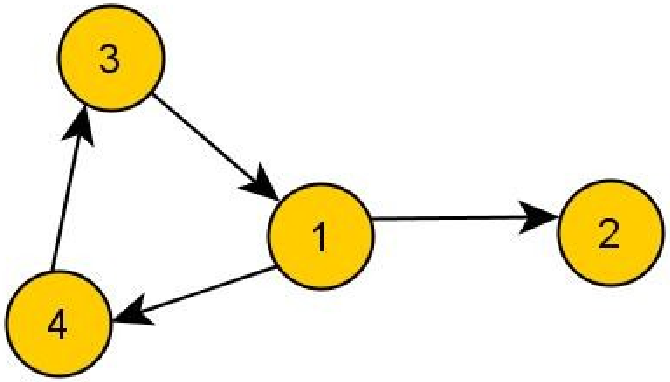
Relacja ρ określona w zbiorze ***X*** jest przeciwsymetryczna, gdy dla każdych dwóch elementów $\boldsymbol{x,y \in X}$, z faktu, że ***x*** jest w relacji z ***y***, wynika, że ***y*** nie jest w relacji z ***x***:

$$\forall x,y \in X:\ x\rho y \Rightarrow \neg(y\rho x).$$

Własności relacji - przeciwsymetria

Przeciwsymetria: jeżeli relacja jest **przeciwzwrotna i antysymetryczna**, to jest **przeciwsymetryczna (asymetryczna)**.

$$R = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$



Własności relacji - spójność

Relacja ρ określona w zbiorze ***X*** jest spójna, jeżeli dla dowolnych dwóch elementów ***x,y*** ∈ ***X***, ***x*** pozostaje w relacji z ***y*** lub ***y*** pozostaje w relacji z ***x***:

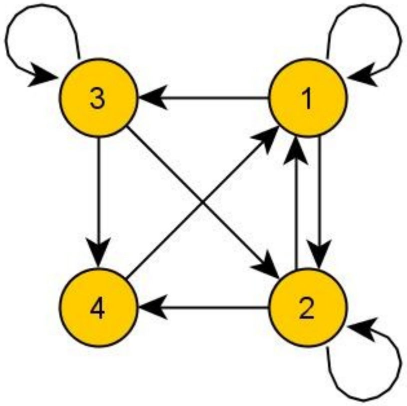
$$\forall x,y \in X:\ x\rho y \lor y\rho x \lor x=y$$

Własności relacji - spójność

W zapisie macierzowym spójność przejawia się tym, że jeżeli poza przekątną w macierzy relacji zachodzi $\boldsymbol{R_{ij}=0}$**,** to odpowiednio $\boldsymbol{R_{ji}=1}$. Oznacza to, ze dla relacji spójnej zawsze zachodzi następujący warunek:

$$\boldsymbol{\forall (i,j)_{i \neq j}\ R_{ij} \vee R_{ji} = 1.}$$

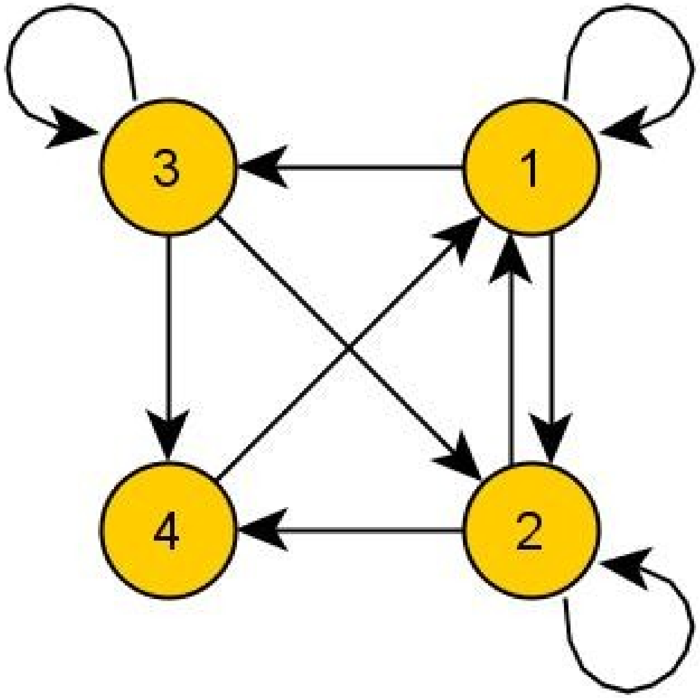
$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



Własności relacji - spójność

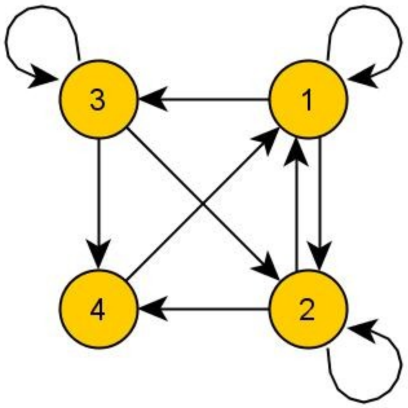
W grafie relacji spójnej pomiędzy dwoma różnymi węzłami istnieje łuk co najmniej w jednym kierunku.

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



Własności relacji - spójność

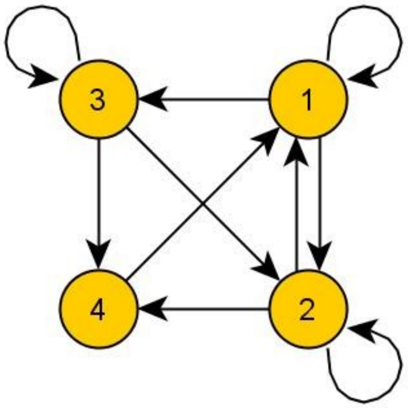
$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



Algorytm badania spójności polega na przechodzeniu przez wszystkie elementy macierzy znajdujące się nad przekątną i sprawdzaniu, czy $\boldsymbol{R_{ij} \vee R_{ji} = 1}$. Jeżeli ten warunek nie jest spełniony co najmniej jeden raz, to relacja nie jest spójna.

Własności relacji - spójność

$$R = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$



Algorytm badania spójności:

```
int spojna(int R[SIZE][SIZE],int n)
{  int i,j;
   for (i=0; i<n–1; i++)
      for (j=i+1; j<n; j++)
         if (!(R[i][j] || R[j][i])) return 0;
   return 1;
}
```

Własności relacji - przechodniość

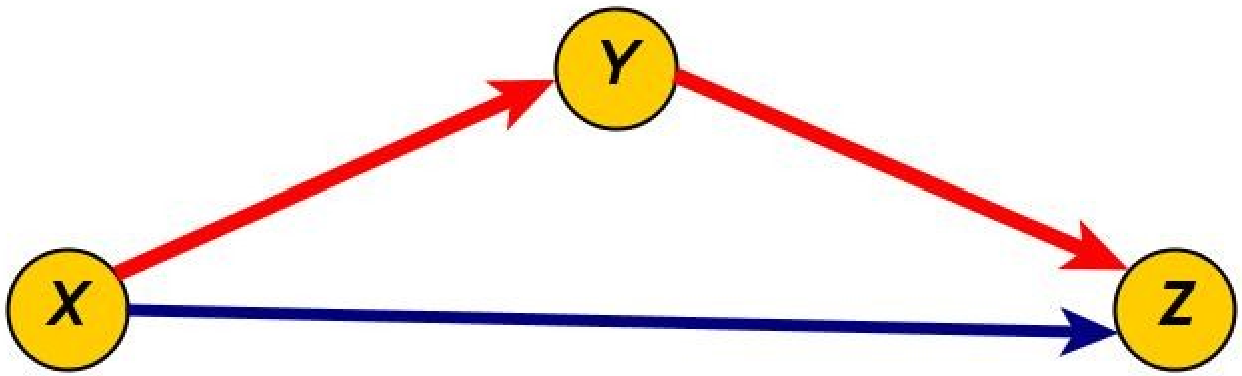
Przechodniość: **R**elacja ρ określona w zbiorze ***X*** jest przechodnia, jeżeli dla dowolnych elementów ***x,y,z*∈*X***, , z faktu, że ***x*** jest w relacji z ***y*** i ***y*** jest w relacji z ***z***, wynika, że ***x*** jest w relacji z ***z***:

$$\forall x,y,z \in X:\ x\rho y \land y\rho z \Rightarrow x\rho z.$$

Własności relacji - przechodniość

$$\forall x,y,z \in X\colon x\rho y \wedge y\rho z \Rightarrow x\rho z.$$

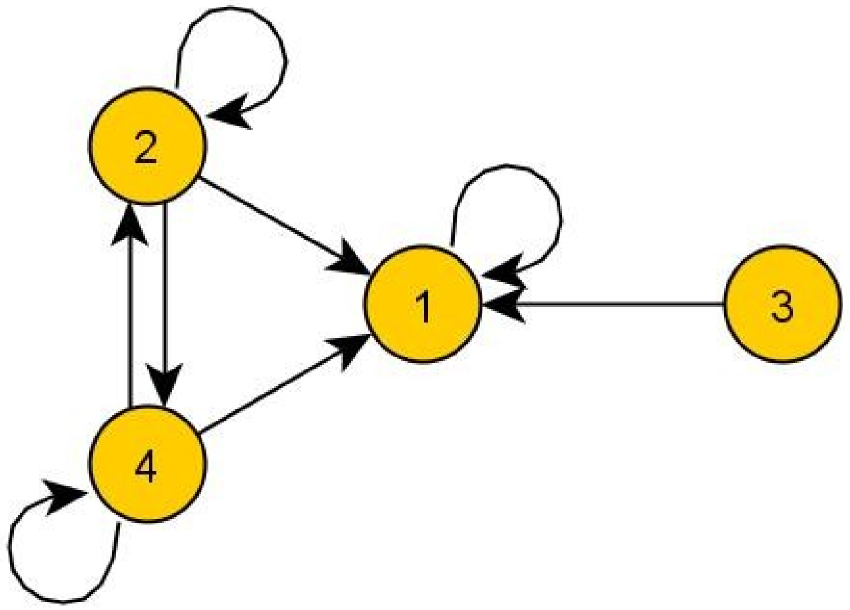
Graf relacji przechodniej charakteryzuje się tym, że jeżeli istnieje łuk od węzła ***x*** do węzła ***y*** i od węzła ***y*** do węzła ***z***, to istnieje także łuk (na skróty) idący od węzła ***x*** do węzła ***z***.



Własności relacji - przechodniość

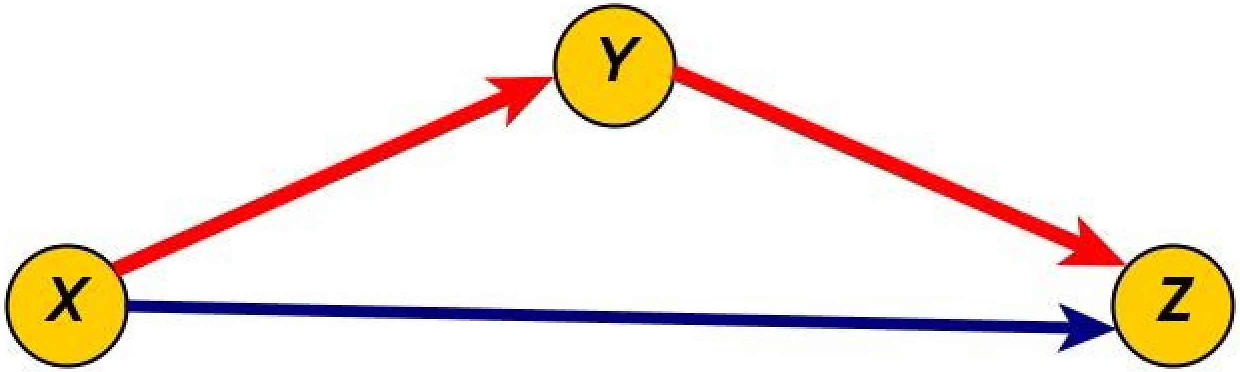
Przykład relacji przechodniej:

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



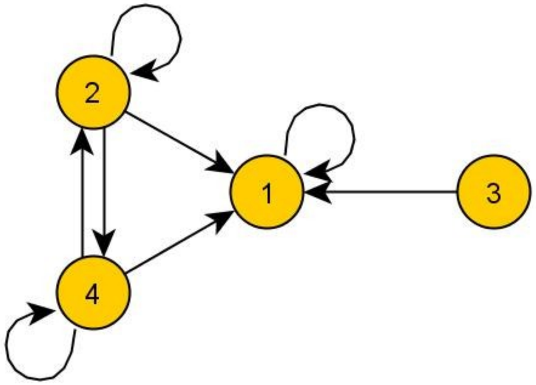
Własności relacji - przechodniość

Warunek przechodniości: *jeśli pomiędzy dwoma różnymi węzłami grafu relacji istnieje ścieżka o długości dwóch łuków, to w grafie relacji przechodniej będzie między nimi istniała ścieżka (**na skróty**) o długości jednego łuku.*



Własności relacji - przechodniość

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



Element $\boldsymbol{R_{ij}}$ macierzy grafu jest równy liczbie ścieżek o długości jednego łuku, biegnących od ***i*** do ***j***.

Kwadrat macierzy sąsiedztwa zlicza wszystkie ścieżki o długości dwóch łuków. Warunek przechodniości relacji można teraz wyrazić następująco:

jeżeli $\boldsymbol{(R^2)_{ij} > 0}$**, to** $\boldsymbol{R_{ij} = 1}$.

Własności relacji - przechodniość

Jak znaleźć kwadrat macierzy relacji?

Należy umieć mnożyć macierze: *C*:=*A*·*B*

$$C=\begin{bmatrix} * & * & \vdots & * & * \\ * & * & \vdots & * & * \\ \dots & \dots & c_{ij} & \dots & \dots \\ * & * & \vdots & * & * \\ * & * & \vdots & * & * \end{bmatrix},\ A=\begin{bmatrix} a_{11} & * & * & * & a_{1n} \\ * & * & * & * & * \\ a_{i1} & a_{i2} & \dots & \dots & a_{in} \\ * & * & * & * & * \\ a_{n1} & * & * & * & a_{nn} \end{bmatrix},\ B=\begin{bmatrix} * & * & b_{1j} & * & * \\ * & * & b_{2j} & * & * \\ \dots & \dots & \vdots & \dots & \dots \\ * & * & \vdots & * & * \\ * & * & b_{nj} & * & * \end{bmatrix}$$

Własności relacji - przechodniość

Mnożenie macierzy *C*:=*A*·*B*

Element C_{ij} macierzy ***C*** znajduje się na przecięciu *i*-tego wiersza macierzy ***A*** i *j*-tej kolumny macierzy ***B***.

$$\begin{bmatrix} * & * & b_{1j} & * & * \\ * & * & b_{2j} & * & * \\ \cdots & \cdots & \vdots & \cdots & \cdots \\ * & * & \vdots & * & * \\ * & * & b_{nj} & * & * \end{bmatrix}$$

$$\begin{bmatrix} * & * & * & * & * \\ * & * & * & * & * \\ a_{i1} & a_{i2} & \cdots & \cdots & a_{in} \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix} \begin{bmatrix} * & * & \vdots & * & * \\ * & * & \vdots & * & * \\ \cdots & \cdots & c_{ij} & \cdots & \cdots \\ * & * & \vdots & * & * \\ * & * & \vdots & * & * \end{bmatrix}$$

Własności relacji - przechodniość

Warunek przechodniości relacji:

***jeżeli* $(R^2)_{ij}>0$, to $R_{ij}=1$.**

Mnożenie macierzy

- Znalezienie elementu w i-tym wierszu i j-tej kolumnie iloczynu macierzy $\boldsymbol{R^2:=R\cdot R}$ wyraża się wzorem:

$$\left(R^2\right)_{ij}=\sum_{k=1}^{n}R_{ik}\cdot R_{kj}$$

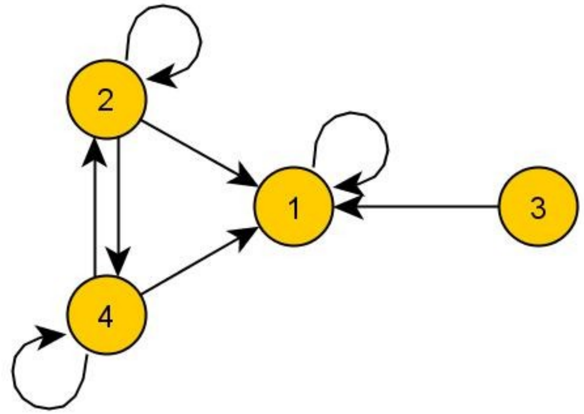
Własności relacji - przechodniość

Warunek przechodniości relacji:

jeżeli **$(R^2)_{ij}>0$, to $R_{ij}=1$.**

Porównanie Macierzy ***R*** oraz ***R²:***

$$R=\begin{bmatrix}1&0&0&0\\1&1&0&1\\1&0&0&0\\1&1&0&1\end{bmatrix},\quad R^2=\begin{bmatrix}1&0&0&0\\3&2&0&2\\1&0&0&0\\3&2&0&2\end{bmatrix}$$



Własności relacji - przechodniość

Macierz relacji jest macierzą zerojedynkową. Jej kwadrat może zawierać zera lub liczby większe od zera.

Dla badania przechodniości istotne jest to, czy ścieżki o długości dwóch łuków istnieją. Dlatego, w odpowiednich miejscach wynikowej macierzy R^2 wystarczy wstawić jedynkę informującą, że takie ścieżki istnieją, a następnie sprawdzać, czy jedynkom w wynikowej macierzy R^2 towarzyszą jedynki w macierzy R.

Własności relacji - przechodniość

Operację podnoszenia macierzy ***R*** do kwadratu, można zastąpić operacją boolowskiego mnożenia macierzy. Wtedy warunek przechodniości relacji będzie spełniony, gdy:

$$\boldsymbol{R * R \leq R}.$$

Operacja * oznacza boolowskie mnożenie macierzy.

Własności relacji - przechodniość

Mnożenie boolowskie macierzy

Mnożenie boolowskie macierzy jest analogiczne ze zwykłym mnożeniem macierzy. Różnica: w mnożeniu boolowskim iloczyn i suma liczb są zastępowane iloczynem i sumą boolowską:

a	b	$a \wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$a \vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Dlatego, we wzorach znak mnożenia „·” jest zastępowany znakiem koniunkcji „$\wedge$”, zaś znak sumowania po indeksach „**Σ**” jest zastępowany znakiem uogólnionej alternatywy „**U**”.

Własności relacji - przechodniość

Mnożenie boolowskie macierzy

- Znalezienie elementu w i-tym wierszu i j-tej kolumnie iloczynu macierzy $\boldsymbol{R^2:=R\cdot R}$ wyraża się wzorem:

$$\left(R^2\right)_{ij} = \sum_{k=1}^{n} R_{ik} \cdot R_{kj}$$

- Znalezienie elementu w i-tym wierszu i j-tej kolumnie iloczynu boolowskiego macierzy $\boldsymbol{B:=R*R}$ wyraża się wzorem:

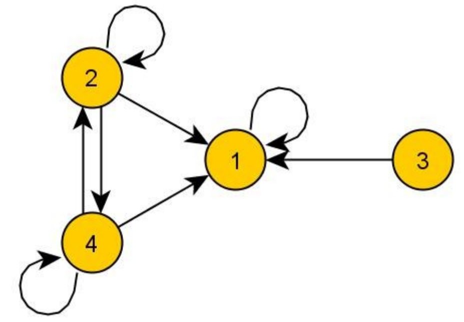
$$B_{ij} = \bigcup_{k=1}^{n} R_{ik} \wedge R_{kj}$$

Własności relacji - przechodniość

Algorytm badania przechodniości:

```
int przechodnia(int R[SIZE][SIZE], int n)
{  int i, j, k, B[SIZE][SIZE];
//mnożenie boolowskie: B = R*R
   for (i=0; i<n; i++)  for (j=0; j<n; j++)
   {  B[i][j] = 0;
      for (k=0; k<n; k++) B[i][j] = B[i][j] || (R[i][k]&&R[k][j]);
   }
//sprawdzenie warunku przechodniości
   for (i=0; i<n; i++) for (j=0; j<n; j++) if (R[i][j] <B[i][j]) return 0;
   return 1;
}
```

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



Własności relacji - przechodniość

Przedstawiony algorytm można dalej uprościć:

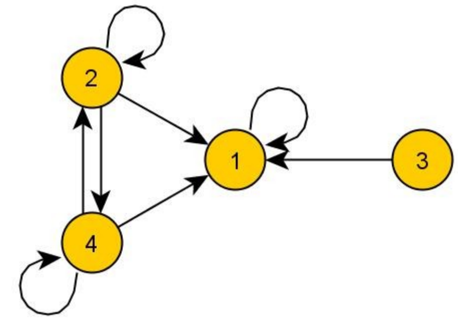
- Ponieważ znajdowanie kolejnych elementów macierzy ***B*** odbywa się niezależnie od innych elementów tej macierzy, dlatego nie ma potrzeby wyliczać wcześniej całej macierzy.
- Zamiast tego można wyliczać kolejny element **macierzy *B*** i od razu sprawdzić, czy jest równy **1**. Jeżeli tak, to trzeba sprawdzić, czy odpowiadający mu element macierzy ***R*** jest równy **1**. Jeżeli równość nie zachodzi, to można zakończyć działanie algorytmu, gdyż relacja nie jest relacją przechodnią.

Własności relacji - przechodniość

Algorytm badania przechodniości:

```
int przechodnia(int R[SIZE][SIZE], int n)
{  int i, j, k, B[SIZE][SIZE];
//mnożenie boolowskie: B = R*R
   for (i=0; i<n; i++)  for (j=0; j<n; j++)
   {  B[i][j] = 0;
      for (k=0; k<n; k++) B[i][j] = B[i][j] || (R[i][k]&&R[k][j]);
      if (R[i][j] <B[i][j]) return 0;
   }
   return 1;
}
```

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



Własności relacji - przechodniość

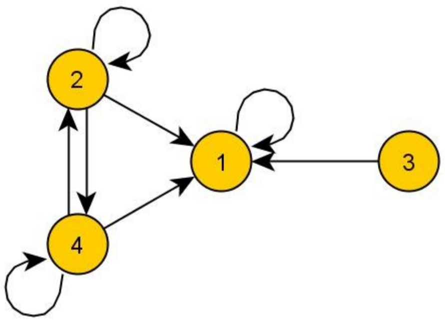
- Można zrezygnować z sumowania. Jeżeli kolejny iloczyn wynosi zero, sumowanie nic nie zmieni. Jeżeli iloczyn jest równy jeden, należy przerwać pętlę liczącą iloczyny.
- Ponieważ cała macierz ***B*** nigdy nie jest potrzebna, a tylko jej kolejne elementy są lokalnie wyliczane, dlatego można także zrezygnować z macierzy ***B*** na rzecz zmiennej lokalnej ***B***, która przechowuje boolowską sumę iloczynów.

Własności relacji - przechodniość

Po uproszczeniach algorytm ma postać:

```
int przechodnia(int n, int R[SIZE][SIZE])
{  int i,j,k;
   int B;
   for (i=0; i<n; i++) for (j=0; j<n; j++)
     {  B = 0;
        for (k=0; (!B) && (k<n); k++) B = R[i][k] && R[k][j];
        if (R[i][j] < B) return 0;
     }
   return 1;
}
```

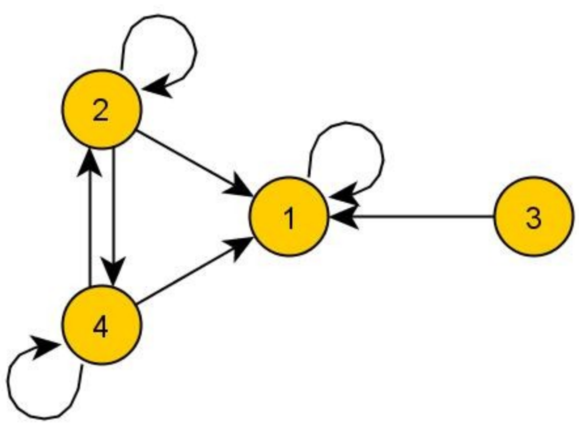
$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



Własności relacji - przechodniość

Porównanie Macierzy *R*, *R*² oraz *R*R:*

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}, \quad R^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 2 \\ 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 2 \end{bmatrix}, \quad R * R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$



Typy relacji

Relacja równoważności: jeżeli relacja jest **zwrotna**, **symetryczna** i **przechodnia**, to jest relacją **równoważności. Relacja równoważności** dzieli zbiór na rozłączne **klasy abstrakcji** (*klasy równoważności*).

Przykład: dwa samochody na pobliskim parkingu są ze sobą w relacji, gdy mają ten sam kolor. Jest to relacja jest **zwrotna**, **symetryczna** i **przechodnia**.

Klas abstrakcji jest tyle, na ile różnych kolorów są pomalowane samochody. Jedną z klas abstrakcji stanowią samochody **czarne**, inną **czerwone**, jeszcze inną **srebrne**, itd.

Typy relacji

Relacja porządku częściowego: jeżeli relacja jest **zwrotna**, **antysymetryczna** i **przechodnia**, to jest relacją **porządku częściowego.**

Zbiór z relacją częściowego porządku jest zbiorem częściowo uporządkowanym. W zbiorze tym porządkowanie (sortowanie w sensie danej relacji) jest możliwe w ramach pewnych podzbiorów.

Przykład: **relacja podzielności** w zbiorze liczb całkowitych dodatnich nie większych niż 100.

Typy relacji

Relacja porządku częściowego: Obok grafu, zbiór z relacją porządku można przedstawić w postaci tzw. diagramu Hassego. Diagram ten powstaje przez zredukowanie grafu relacji:

1. Zredukować wszystkie **pętle charakteryzujące zwrotność**.
2. Zredukować wszystkie **łuki charakteryzujące przechodniość**.
3. Graf narysować tak, aby wszystkie jego **strzałki** były **skierowane do góry**.
4. Zredukować **strzałki** na końcach łuków.

Typy relacji

Relacja porządku liniowego: jeżeli relacja jest relacją **porządku częściowego** i jest **spójna**, to jest to **relacja porządku liniowego**.

Porządek liniowy (silniejszy niż porządek częściowy) **umożliwia** porządkowanie (**sortowanie** w sensie danej relacji) **całego zbioru**.

Przykład: relacja $\geq$ w zbiorze liczb całkowitych dodatnich nie większych niż 100.

Relację porządku liniowego można również przedstawić w postaci **diagramu Hassego**, który tym razem ma **postać linii**.

Typy relacji

Uwaga: Relacja **równoważności umożliwia badanie**, czy dwa elementy w zbiorze **są równe** (należą do tej samej klasy abstrakcji), czy też **są różne** (należą do różnych klas abstrakcji).

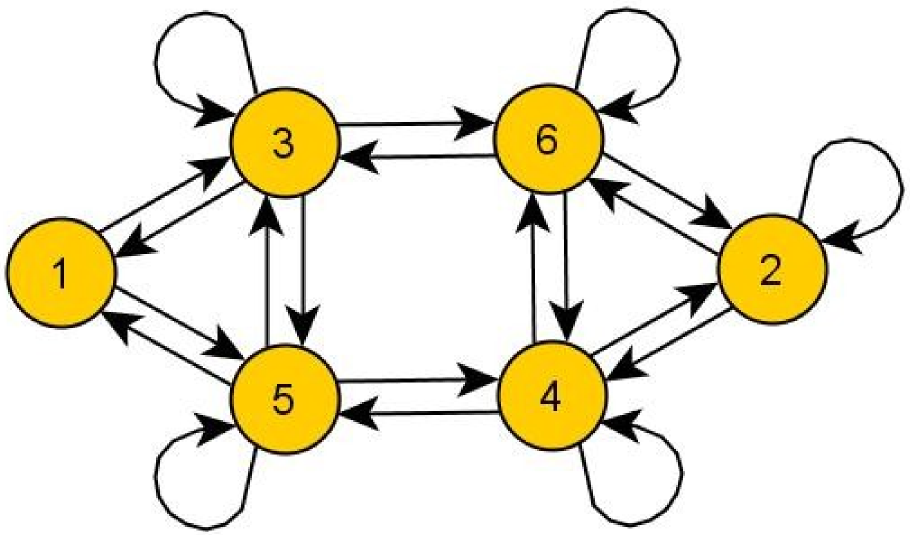
W ramach tej relacji **nie ma możliwości porządkowania** elementów np. w sensie sprawdzania, czy jeden element poprzedza drugi element (w sensie rozważanej relacji).

Tymczasem **relacje porządku** dają takie możliwości.

Przykłady relacji

Przykład 1: Dany jest zbiór ***X={1,2,3,4,5,6}***, Relacja w zbiorze ***X*** jest zdefiniowana w następujący sposób ***xρy*** $\Leftrightarrow$ ***x+y jest liczbą złożoną***, ***x,y***$\in$***X***

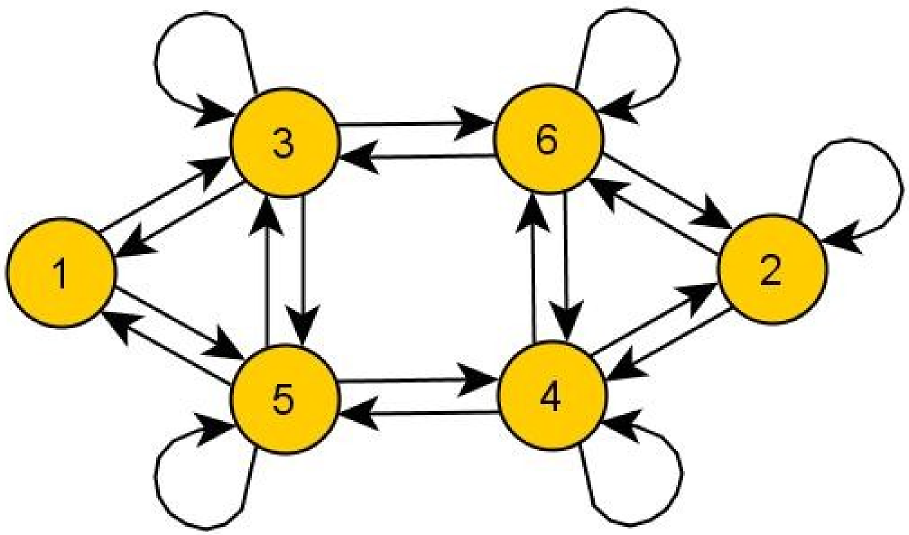
$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$



Przykłady relacji

Przykład 1: Relacja jest **relacją symetryczną**: ***R=R^T***. W grafie łuki biegną w obie strony. Symetria jest jedyną własnością tej relacji. Relacja nie jest **zwrotna**, **przeciwzwrotna**, **antysymetryczna**, **przechodnia** i **spójna**.

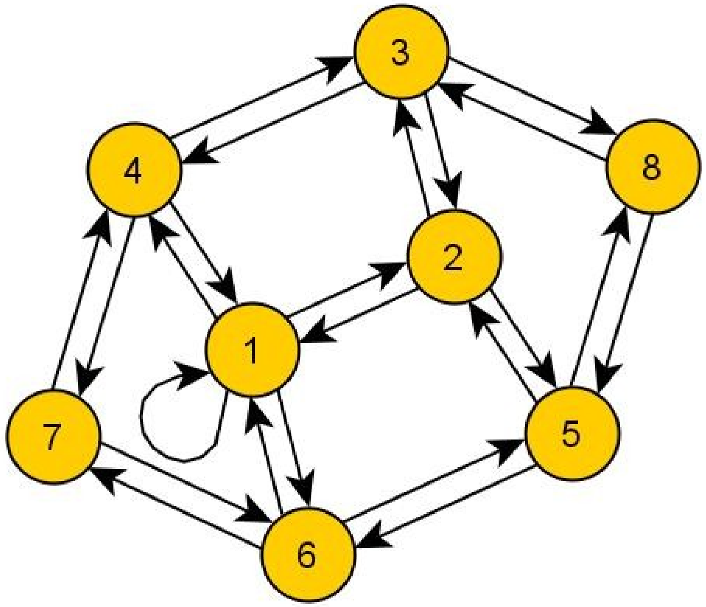
$$R = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$



Przykłady relacji

Przykład 2: Dany jest zbiór ***X*={1,2,3,4,5,6,7,8}**. Relacja w zbiorze ***X*** jest zdefiniowana w następujący sposób: ***xρy* ⇔ *x+y jest liczbą pierwszą***, ***x,y*∈*X***.

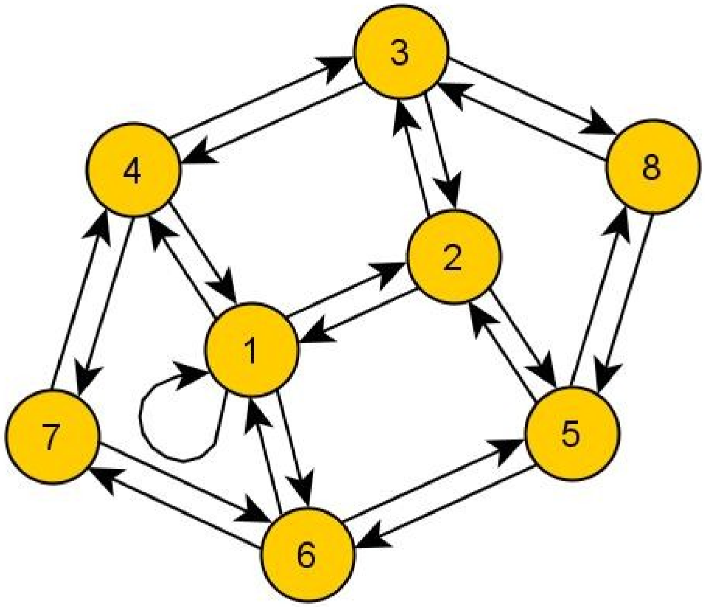
$$R = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$



Przykłady relacji

Przykład 2: Relacja jest **relacją symetryczną**: $R=R^T$. W grafie łuki biegną w obie strony. Symetria jest jedyną własnością tej relacji. Relacja nie jest **zwrotna**, **przeciwzwrotna**, **antysymetryczna**, **przechodnia** i **spójna**.

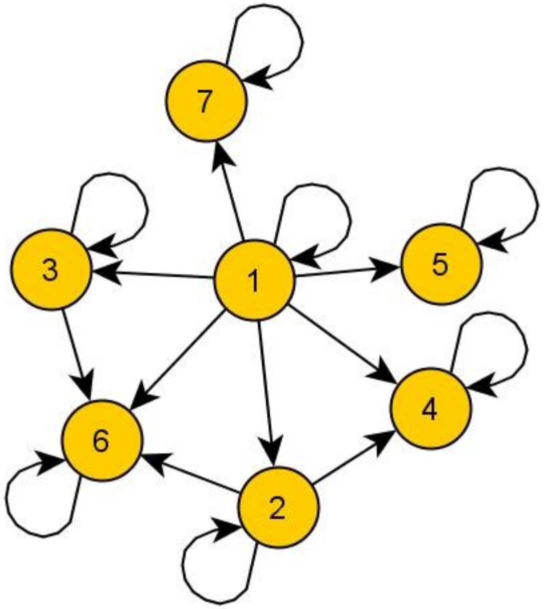
$$R = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$



Przykłady relacji

Przykład 3: Dany jest zbiór ***X*={1,2,3,4,5,6,7}**. Relacja jest zdefiniowana w następujący sposób: ***xρy*** $\Leftrightarrow$ ***x jest dzielnikiem liczby y***, ***x,y***$\in$***X***.

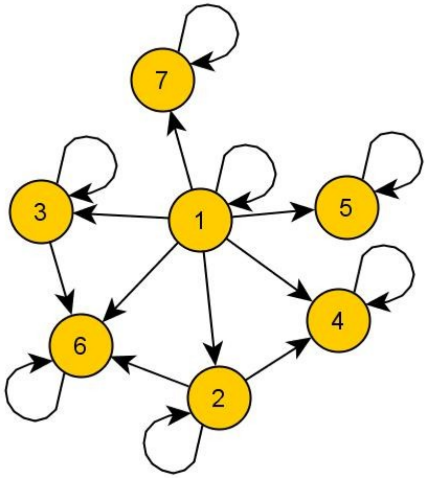
$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

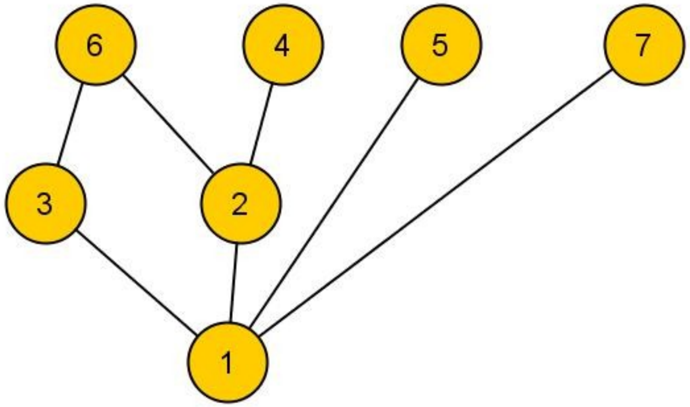


Przykłady relacji

Przykład 3: Relacja jest **zwrotna** – każda liczba jest swoim własnym dzielnikiem. Jest **antysymetryczna** – jeżeli liczba ***x*** jest dzielnikiem różnej od siebie liczby ***y***, to ***y*** nie jest dzielnikiem liczby ***x***. W grafie pomiędzy węzłami biegną łuki tylko w jedną stronę. Relacja jest **przechodnia**.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



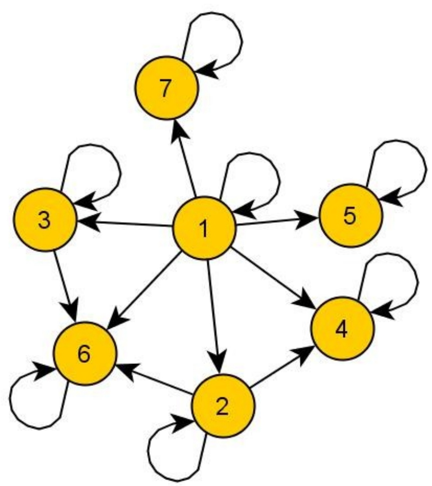


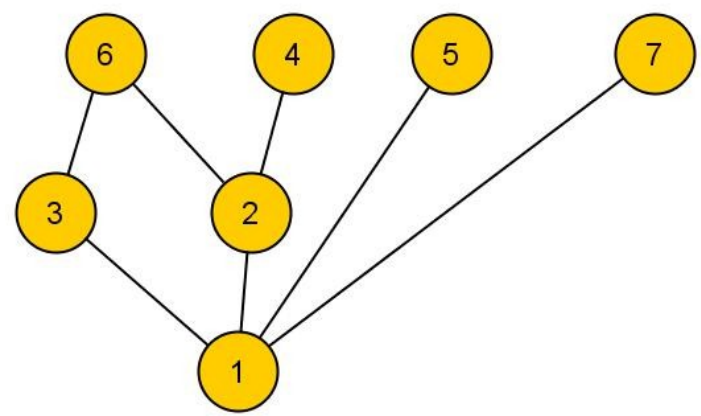
Przykłady relacji

Przykład 3: **Spełnione są warunki definiujące relację częściowego porządku**.

.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

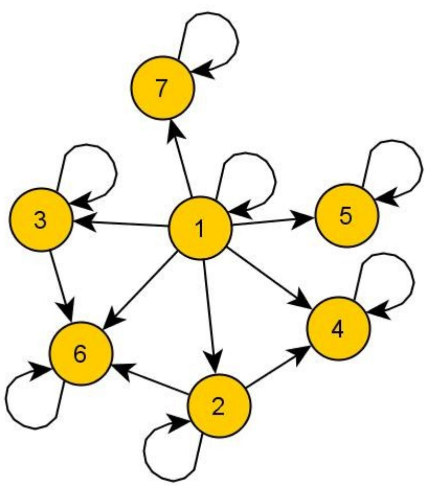


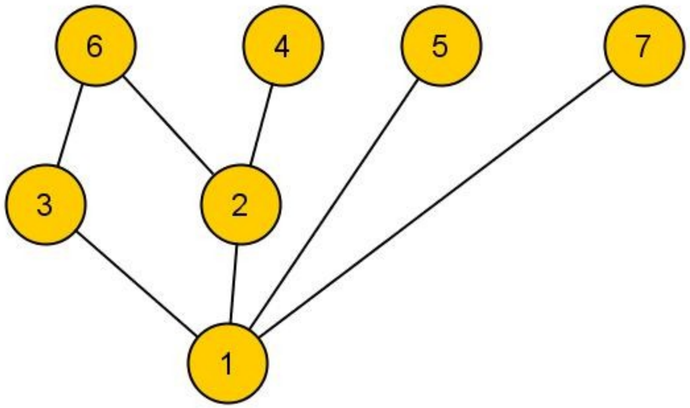


Przykłady relacji

Przykład 3: Gdy relacja jest relacją częściowego porządku, jedne elementy poprzedzają inne w ramach podzbiorów. Oznacza to, że pewne podzbiory danego zbioru mogą być porządkowane (sortowane) w sensie tej relacji.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$



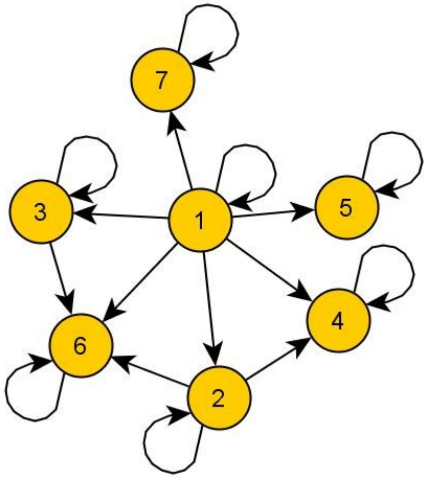


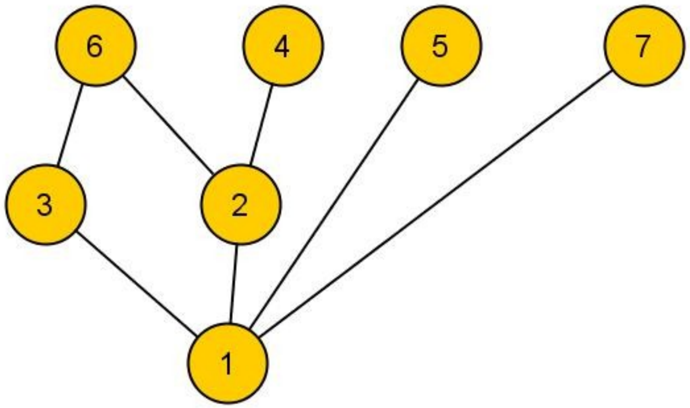
Przykłady relacji

Przykład 3: Z diagramu Hassego można odczytać, które elementy poprzedzają się wzajemnie, a więc które podzbiory można porządkować w sensie omawianej relacji: **{1,3,6}**, **{1,2,6}**, **{1,2,4}**, **{1,5}** oraz **{1,7}**.

.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

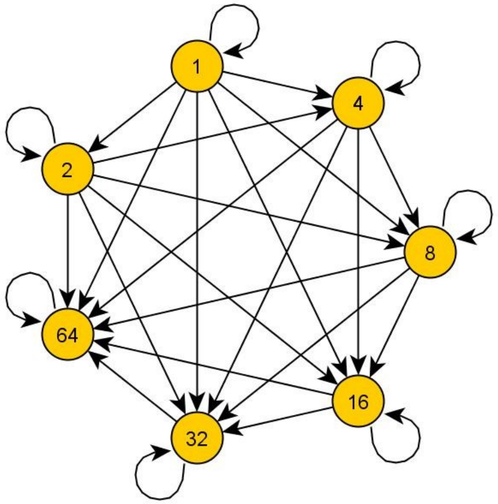




Przykłady relacji

Przykład 4: ***X*={1,2,4,8,16,32,64}**. Relacja jest zdefiniowana w następujący sposób: ***x*ρ*y* ⇔ *x* jest dzielnikiem liczby *y***, ***x*,*y*∈*X***. *Założono*, że kolejnym elementom zbioru ***X*** odpowiadają kolejne numery wierszy i kolumn w macierzy relacji

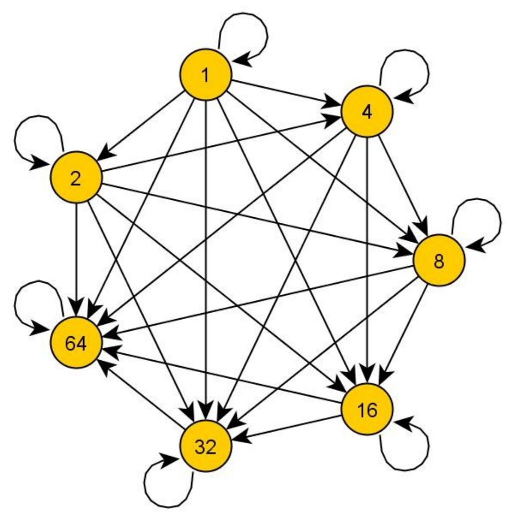
$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

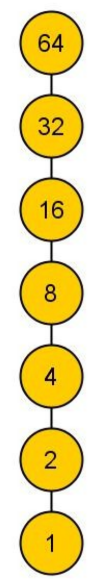


Przykłady relacji

Przykład 4: relacja jest **zwrotna**, **antysymetryczna**, **przechodnia** i **spójna**. Jest to relacja **porządku liniowego**. Oznacza to, że cały zbiór można uporządkować (posortować) w sensie danej relacji, co widać na **diagramie Hassego**.

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

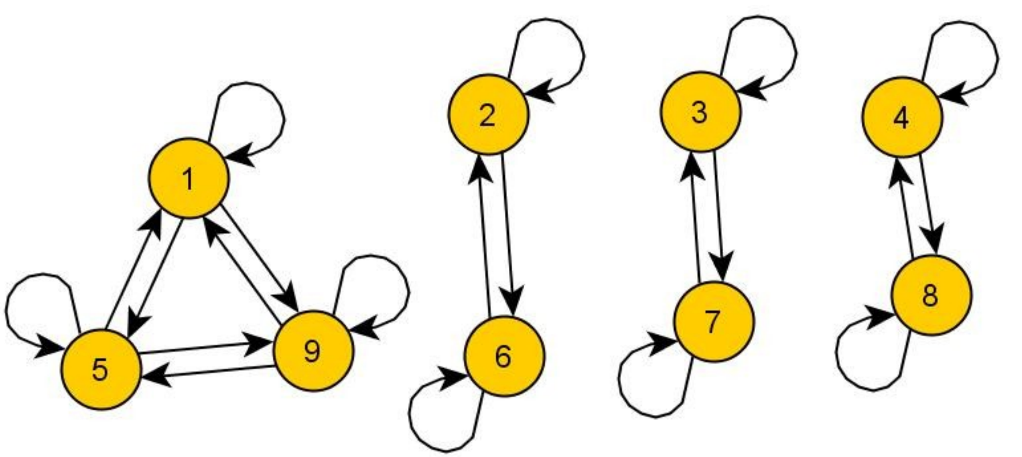




Przykłady relacji

Przykład 5:Dla zbioru ***X*={1,2,3,4,5,6,7,8,9}**, relacja jest zdefiniowana w następujący sposób: ***xρy* ⇔ (*x*≡*y*) *mod* 4**, ***x,y*∈*X***. Inaczej: zmienne ***x*** i ***y*** są ze sobą w relacji, gdy mają **jednakowe reszty z dzielenia przez 4**.

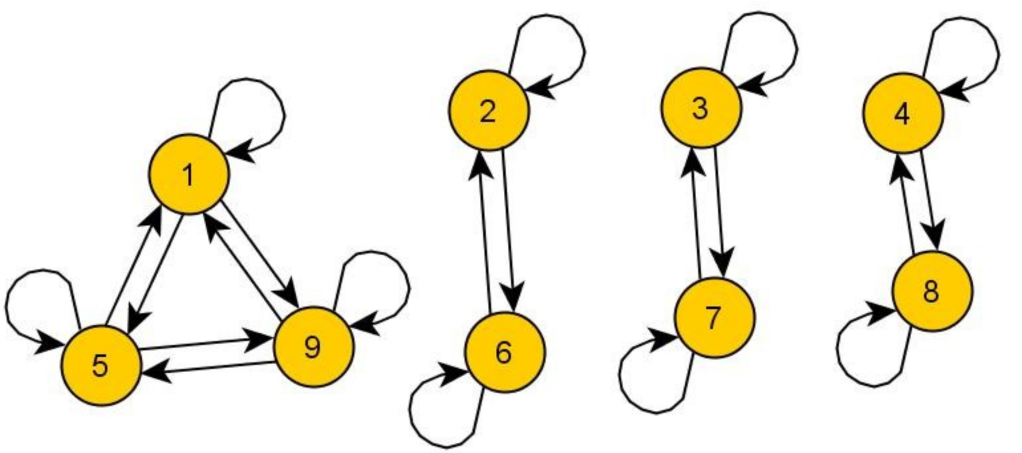
$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Przykłady relacji

Przykład 5: Relacja **zwrotna** – każdy element jest w relacji z samym sobą, **na przekątnej** macierzy znajdują się **same jedynki**, a w grafie **każdy węzeł** ma **pętlę**.

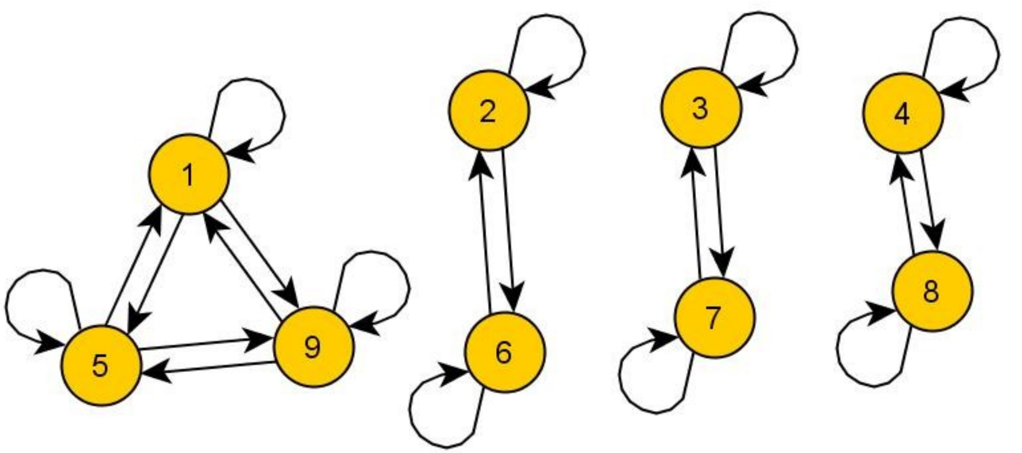
$$R=\begin{bmatrix}1&0&0&0&1&0&0&0&1\\0&1&0&0&0&1&0&0&0\\0&0&1&0&0&0&1&0&0\\0&0&0&1&0&0&0&1&0\\1&0&0&0&1&0&0&0&1\\0&1&0&0&0&1&0&0&0\\0&0&1&0&0&0&1&0&0\\0&0&0&1&0&0&0&1&0\\1&0&0&0&1&0&0&0&1\end{bmatrix}$$



Przykłady relacji

Przykład 5: Relacja **symetryczna** – jeżeli ***x*** ma **resztę z dzielenia przez 4** identyczną z resztą z dzielenia ***y*** przez **4**, to także ***y*** ma **resztę z dzielenia przez 4** identyczną z resztą z dzielenia ***x*** przez **4**. W grafie **łuki biegną parami** i są skierowane w przeciwnych kierunkach.

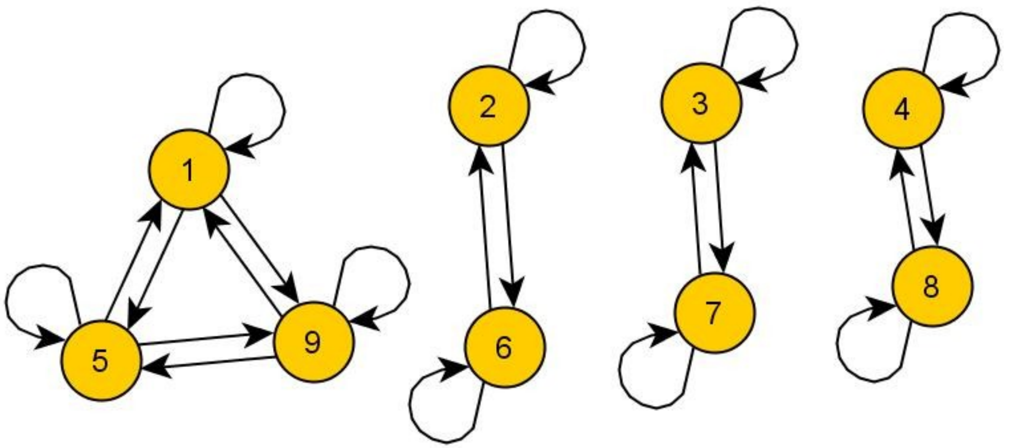
$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Przykłady relacji

Przykład 5: Relacja jest także **przechodnia** – w grafie każda **ścieżka o długości dwóch łuków** może być zastąpiona przez **ścieżkę „na skróty”**. Jest to więc **relacja równoważności**.

$$R=\begin{bmatrix}1&0&0&0&1&0&0&0&1\\0&1&0&0&0&1&0&0&0\\0&0&1&0&0&0&1&0&0\\0&0&0&1&0&0&0&1&0\\1&0&0&0&1&0&0&0&1\\0&1&0&0&0&1&0&0&0\\0&0&1&0&0&0&1&0&0\\0&0&0&1&0&0&0&1&0\\1&0&0&0&1&0&0&0&1\end{bmatrix}$$

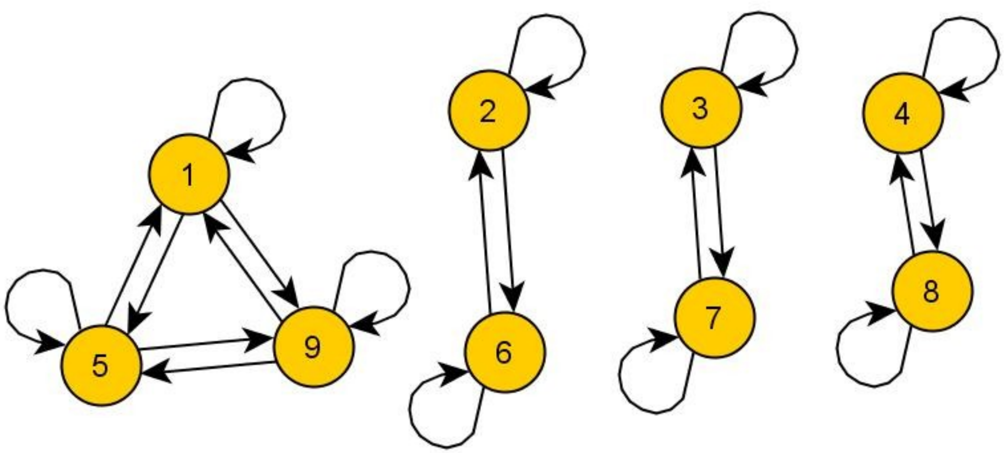


Przykłady relacji

Przykład 5: Relacja ta dzieli zbiór na cztery podzbiory będące klasami abstrakcji. Każda z klas zawiera elementy będące ze sobą w relacji, czyli takie, które maja identyczne reszty z dzielenia przez **4**: **{1,5,9}**, **{2,6}**, **{3,7}**, **{4,8}**. Klasy abstrakcji są widoczne jako cztery podgrafy spójne.

.

$$R = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$



Program do badania relacji

Macierzowa reprezentacja relacji umożliwia algorytmizację badania własności relacji, a w konsekwencji umożliwia utworzenie programu do badania własności i typów relacji.

Taki program mógłby być dobrym treningiem programistycznym, przygotowującym do tworzenia w przyszłości bardziej złożonych programów.

Sformułowanie odpowiedniego zadania programistycznego przedstawiono w postaci zbliżonej do formy zadań występujących na zawodach algorytmicznych.

Program do badania relacji

Koledzy Jasia postanowili zagrać w grę planszową, która składa się z wielu etapów. W każdym z etapów odbywa się rozgrywka, po zakończeniu której, aby przejść na wyższy poziom trzeba rozwiązać pewne dziwne zadanie matematyczne.

Jaś chętnie by wziął udział w grze, ale nie potrafi rozwiązać zadania, gdyż nie uważał w czasie spotkań kółka matematycznego. Potrzebuje pomocy. Pomóż mu, pisząc odpowiedni program.

Program do badania relacji

Jasio powinien odpowiedzieć na pytanie, jakie własności i jaki typ ma pewna relacja dwuczłonowa w zbiorze składającym się z nie więcej niż stu elementów.

Relacja jest zdefiniowana w następujący sposób: jeżeli *i*-ty element jest w relacji z elementem *j*-tym, to na wejściu w osobnym wierszu pojawi się rozdzielona spacjami para liczb *i* oraz *j*. Liczba wierszy w zestawie danych jest równa liczbie par (*i*,*j*) dla których zachodzi relacja.

O wielkości zbioru, na którym określona jest ta relacja można wnioskować na podstawie największej z liczb *i* albo *j* w parach.

Program do badania relacji

Zadanie: Napisz program, którego wynikiem działania będzie wiersz opisujący przy pomocy odpowiednich skrótów oddzielonych spacjami własności relacji (o ile występują) oraz typ relacji (również, o ile występuje), w następującej kolejności:

- *Z* – zwrotna;
- *PZ* – przeciwzwrotna;
- *S* – symetryczna;
- *AS* – antysymetryczna;
- *PS* – przeciwsymetryczna;
- *P* – przechodnia;
- *SP* – spójna;
- *RR* – relacja równoważności;
- *RCP* – relacja częściowego porządku;
- *RLP* – relacja liniowego porządku;
- *X* – żadna z powyższych.

Program do badania relacji

Przykład:

Wejście:

2 3

2 9

3 4

5 7

5 9

6 7

7 8

8 3

Wyjście:

PZ AS

Podsumowanie

Relacja dwuczłonowa w zbiorze może być przedstawiona przy pomocy **macierzy zerojedynkowej**, która jest jednocześnie macierzą sąsiedztwa grafu relacji.

Graf relacji przez odwoływanie do wyobraźni, daje możliwość badania relacji w prosty i intuicyjny sposób.

Tymczasem **reprezentacja macierzowa** umożliwia algorytmizację badania własności relacji, a w konsekwencji pozwala na napisanie programu do badania własności i typów relacji.

Podsumowanie

Przedstawione podejście daje możliwość przejście od abstrakcyjnego modelu matematycznego do konkretnego modelu w postaci algorytmu komputerowego.

W wykładzie omówiono algorytmy wykorzystujące tę macierz do badania własności relacji takich jak **zwrotność, przeciwzwrotność, symetria, antysymetria, przechodniość** oraz **spójność** relacji.

Podsumowanie

Przedstawiono przykłady różnych relacji, wraz z określeniem ich własności oraz typów.

Zaproponowano zadanie programistyczne w postaci zbliżonej do zadań występujących na zawodach algorytmicznych. Rozwiązaniem tego zadania byłby program wykorzystujący omówione w wykładzie algorytmy badania relacji.

Podsumowanie

Jeszcze raz o przydatności pojęci a relacji w informatyce:

- ✓ **Relacyjne bazy danych** – tabela jest uogólnieniem relacji na wiele wymiarów.
- ✓ W eksploracji danych i statystyce przy zbieraniu danych używa się **skal pomiarowych** konstytuowanych przez relacje:
  - ✓ Skala nominalna – relacja równoważności.
  - ✓ Skala porządkowa – relacja porządku częściowego.
  - ✓ Skala ilorazowa lub interwałowa – relacja porządku liniowego.

Informacja

W trakcie przygotowywania niniejszego wykładu, do rysowania grafów zastosowano program yEd Graph Editor w wersji 3.6.1.1, dostępny na stronie internetowej http://www.yworks.com, należącej do firmy yWorks GmbH.

informatyka+

Algorytmika i programowanie

Bazy danych

Multimedia, grafika i technologie internetowe

Sieci komputerowe

Tendencje w rozwoju informatyki i jej zastosowań

Człowiek – najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Warszawska Wyższa Szkoła Informatyki

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.