Graph-based image processing

— Graphs: basic notions —

(Professor version)

Silvio Guimarães

Graduate Program in Informatics – PPGINF
Image and Multimedia Data Science Laboratory – IMScience
Pontifical Catholic University of Minas Gerais – PUC Minas

Aug 2022

Outline of the lecture

1. Definitions and first examples of graphs
2. Representation of a graph in a computer memory
3. First algorithms: computing the symmetric of a graph
4. Some remarkable graphs

Graph-based image processing

— Definitions and first examples of graphs —

(Professor version)

Silvio Guimarães

Graduate Program in Informatics – PPGINF
Image and Multimedia Data Science Laboratory – IMScience
Pontifical Catholic University of Minas Gerais – PUC Minas

Aug 2022

Set notations

- Let E be a finite set

Set notations

- Let E be a finite set
- A set S is called a **subset of E** if any element of S is also an element of E

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Example

- *If* $E = \{1, 2, 3\}$
- *Then* $\mathcal{P}(E) =$

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Example

- *If* $E = \{1, 2, 3\}$
- *Then* $\mathcal{P}(E) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1, 2, 3\}\}$

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Example

- *If* $E = \{1, 2, 3\}$
- *Then* $\mathcal{P}(E) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1, 2, 3\}\}$

- *Remark.* $S \in \mathcal{P}(E)$ means that S is a subset of E

Set notations

- Let E be a finite set
- A set S is called a subset of E if any element of S is also an element of E
- If S is a subset of E, we write $S \subseteq E$
- The set of all subsets of E is denoted by $\mathcal{P}(E)$

Example

- *If* $E = \{1, 2, 3\}$
- *Then* $\mathcal{P}(E) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1, 2, 3\}\}$

- *Remark.* $S \in \mathcal{P}(E)$ means that S is a subset of E
- The proposition $S \in \mathcal{P}(E)$ can thus be equivalently written as $S \subseteq E$

Graph

Definition

- *A graph is a pair* $G = (E, \Gamma)$ *where* E *is a finite set and where* Γ *is a map from* E *to* $\mathcal{P}(E)$

Graph

Definition

- *A* **graph** *is a pair* $G = (E, \Gamma)$ *where* E *is a finite set and where* Γ *is a map from* E *to* $\mathcal{P}(E)$

Example

- $G = (E, \Gamma)$
- *with* $E = \{1, 2, 3, 4\}$ *and*
- Γ *defined by*
  - $\Gamma(1) = \{1, 2, 4\}$
  - $\Gamma(2) = \{3, 1\}$
  - $\Gamma(3) = \{4\}$
  - $\Gamma(5) = \emptyset$

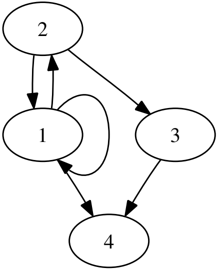
Graph

Definition

- *A* **graph** *is a pair* $G = (E, \Gamma)$ *where* E *is a finite set and where* Γ *is a map from* E *to* $\mathcal{P}(E)$

Example

- $G = (E, \Gamma)$
- *with* $E = \{1, 2, 3, 4\}$ *and*
- Γ *defined by*
  - $\Gamma(1) = \{1, 2, 4\}$
  - $\Gamma(2) = \{3, 1\}$
  - $\Gamma(3) = \{4\}$
  - $\Gamma(5) = \emptyset$



Representation by arrows

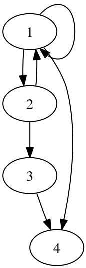
Graph

Definition

- *A* **graph** *is a pair* $G = (E, \Gamma)$ *where* E *is a finite set and where* Γ *is a map from* E *to* $\mathcal{P}(E)$

Example

- $G = (E, \Gamma)$
- *with* $E = \{1, 2, 3, 4\}$ *and*
- Γ *defined by*
  - $\Gamma(1) = \{1, 2, 4\}$
  - $\Gamma(2) = \{3, 1\}$
  - $\Gamma(3) = \{4\}$
  - $\Gamma(5) = \emptyset$



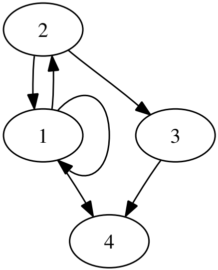
Representation by arrows

Usual terminology

- Any element of E is called a **vertex (of the graph G)**

Example

- *1 is a vertex of G*

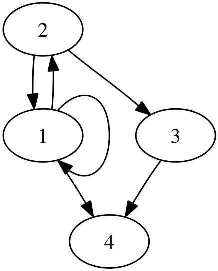


Usual terminology

- Any element of E is called a **vertex (of the graph G)**
- Let x and y be two vertices of E, if $y \in \Gamma(x)$,
  - y is a **successor of x**

Example

- *1 is a vertex of G*
- *4 is a successor of 3*

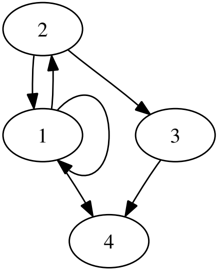


Usual terminology

- Any element of E is called a **vertex (of the graph G)**
- Let x and y be two vertices of E, if $y \in \Gamma(x)$,
  - y is a **successor of x** and x is a **predecessor of y**

Example

- *1 is a vertex of G*
- *4 is a successor of 3*
- *2 is a predecessor of 3*

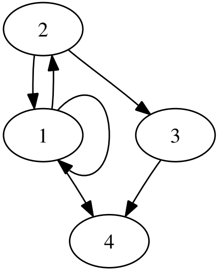


Usual terminology

- Any element of E is called a vertex (of the graph G)
- Let x and y be two vertices of E, if $y \in \Gamma(x)$,
  - y is a successor of x and x is a predecessor of y
  - the ordered pair (x, y) is called an arc (of the graph G)

Example

- *1 is a vertex of G*
- *4 is a successor of 3*
- *2 is a predecessor of 3*
- *Thus, $(3, 4)$ and $(2, 3)$ are two arcs of G*

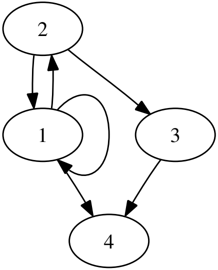


Set of arcs

- The symbol $\overrightarrow{\Gamma}$ denotes the set of all arcs of the graph (E, Γ)

Example

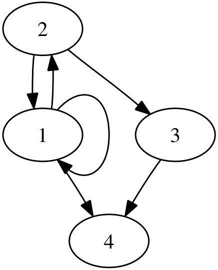
- $\overrightarrow{\Gamma} = \{(1,1),(1,2),(1,4),$
$(2,3),(2,1),(3,4)\}$



- The symbol $\overrightarrow{\Gamma}$ denotes the set of all arcs of the graph (E, Γ)
- $\overrightarrow{\Gamma}$ is a subset of the Cartesian product $E \times E = \{(x, y) \mid x \in E, y \in E\} : \overrightarrow{\Gamma} \in \mathcal{P}(E \times E)$

Example

- $\overrightarrow{\Gamma} = \{(1,1),(1,2),(1,4),(2,3),(2,1),(3,4)\}$

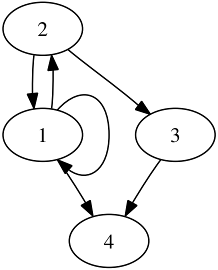


Set of arcs

- The symbol $\overrightarrow{\Gamma}$ denotes the set of all arcs of the graph (E, Γ)
- $\overrightarrow{\Gamma}$ is a subset of the Cartesian product $E \times E = \{(x, y) \mid x \in E, y \in E\} : \overrightarrow{\Gamma} \in \mathcal{P}(E \times E)$
- $\overrightarrow{\Gamma} = \{(x, y) \in E \times E \mid y \in \Gamma(x)\}$

Example

- $\overrightarrow{\Gamma} = \{(1, 1), (1, 2), (1, 4), (2, 3), (2, 1), (3, 4)\}$



Equivalent representation of a graph

Remark

- *The datum of the pair* (E, Γ) *is equivalent to the datum of the pair* $(E, \overrightarrow{\Gamma})$*:*

Equivalent representation of a graph

Remark

- *The datum of the pair* (E, Γ) *is equivalent to the datum of the pair* $(E, \overrightarrow{\Gamma})$*:*
  - *More precisely ,* $\forall x, y \in E, y \in \Gamma(x) \Leftrightarrow (x, y) \in \overrightarrow{\Gamma}$

Equivalent representation of a graph

Remark

- *The datum of the pair* (E, Γ) *is equivalent to the datum of the pair* $(E, \overrightarrow{\Gamma})$*:*
  - *More precisely ,* $\forall x, y \in E, y \in \Gamma(x) \Leftrightarrow (x, y) \in \overrightarrow{\Gamma}$
- *Therefore, the pair* $(E, \overrightarrow{\Gamma})$ *is also called* *a graph*

Equivalent representation of a graph

Remark

- *The datum of the pair* (E, Γ) *is equivalent to the datum of the pair* $(E, \overrightarrow{\Gamma})$*:*
  - *More precisely ,* $\forall x, y \in E, y \in \Gamma(x) \Leftrightarrow (x, y) \in \overrightarrow{\Gamma}$
- *Therefore, the pair* $(E, \overrightarrow{\Gamma})$ *is also called* ***a graph***

Notation

- *We set* $n = |E|$*, and* $m = |\overrightarrow{\Gamma}|$

Equivalent representation of a graph

Remark

- *The datum of the pair* (E, Γ) *is equivalent to the datum of the pair* $(E, \overrightarrow{\Gamma})$*:*
  - *More precisely ,* $\forall x, y \in E, y \in \Gamma(x) \Leftrightarrow (x, y) \in \overrightarrow{\Gamma}$
- *Therefore, the pair* $(E, \overrightarrow{\Gamma})$ *is also called* *a graph*

Notation

- *We set* $n = |E|$*, and* $m = |\overrightarrow{\Gamma}|$
- *The* size *of the graph* $G = (E, \Gamma)$ *is the sum* $n + m$

Exercise: Union of graphs

- Let $E = \{a, b, c, d, e\}$,
- Let $\overrightarrow{\Gamma_1} = \{(a, c), (c, d), (d, e)\}$
- let $\overrightarrow{\Gamma_2} = \{(b, a), (a, d), (d, c), (c, e)\}$
- Let $\overrightarrow{\Gamma_3} = \overrightarrow{\Gamma_1} \cup \overrightarrow{\Gamma_2}$

Exercise: Union of graphs

- Let $E = \{a, b, c, d, e\}$,
- Let $\overrightarrow{\Gamma_1} = \{(a, c), (c, d), (d, e)\}$
- let $\overrightarrow{\Gamma_2} = \{(b, a), (a, d), (d, c), (c, e)\}$
- Let $\overrightarrow{\Gamma_3} = \overrightarrow{\Gamma_1} \cup \overrightarrow{\Gamma_2}$
- Give the sets $\Gamma_1(a)$, $\Gamma_2(a)$, $\Gamma_3(a)$, $\Gamma_1(b)$, $\Gamma_2(b)$, $\Gamma_3(b)$, $\Gamma_1(d)$, $\Gamma_2(d)$, and $\Gamma_3(d)$

Exercise: Union of graphs

- Let $E = \{a, b, c, d, e\}$,
- Let $\overrightarrow{\Gamma_1} = \{(a, c), (c, d), (d, e)\}$
- let $\overrightarrow{\Gamma_2} = \{(b, a), (a, d), (d, c), (c, e)\}$
- Let $\overrightarrow{\Gamma_3} = \overrightarrow{\Gamma_1} \cup \overrightarrow{\Gamma_2}$
- Give the sets $\Gamma_1(a)$, $\Gamma_2(a)$, $\Gamma_3(a)$, $\Gamma_1(b)$, $\Gamma_2(b)$, $\Gamma_3(b)$, $\Gamma_1(d)$, $\Gamma_2(d)$, and $\Gamma_3(d)$

Property. For any $\overrightarrow{\Gamma_1}, \overrightarrow{\Gamma_2}, \overrightarrow{\Gamma_3} \in \mathcal{P}(E \times E)$ such that $\overrightarrow{\Gamma_3} = \overrightarrow{\Gamma_1} \cup \overrightarrow{\Gamma_2}$:
$\forall x \in E$, $\Gamma_3(x) = \Gamma_1(x) \cup \Gamma_2(X)$

Graph-based image processing

— Representation of a graph in a computer memory —

(Professor version)

Silvio Guimarães

Graduate Program in Informatics – PPGINF
Image and Multimedia Data Science Laboratory – IMScience
Pontifical Catholic University of Minas Gerais – PUC Minas

Aug 2022

Outline of the lecture

1 Definitions and first examples of graphs

2 Representation of a graph in a computer memory

3 First algorithms: computing the symmetric of a graph

4 Some remarkable graphs

Representation of a graph in a computer memory

- By a set of vertices and a set of arcs
  (*i.e., representation of* $(E, \overrightarrow{\Gamma})$)
- *By a set of vertices and sets of successors*
  (i.e., representation of (E, Γ))

Representation of a graph in a computer memory

- By a set of vertices and a set of arcs

  (*i.e., representation of* $(E, \overrightarrow{\Gamma})$)
- *By a set of vertices and sets of successors*

  (i.e., representation of (E, Γ))

Problem

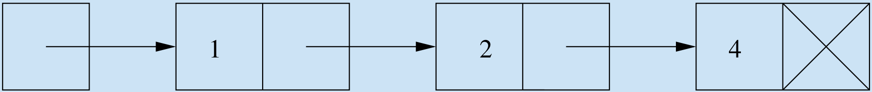
- *Representation of a set in a memory*
- *Representation of a subset S of $E = \{1, \ldots, n\}$*

- The linked list is made of nodes
- Each node represents an element of S
- Each node contains two fields
  1. An element of S (an integer between 1 and n)
  2. A reference (*i.e., a link) to the next node in the LL*

- The linked list is made of nodes
- Each node represents an element of S
- Each node contains two fields
  1. An element of S (an integer between 1 and n)
  2. A reference (*i.e., a link) to the next node in the LL*

Example

- *Let* $S = \{1, 2, 4\}$



Boolean array (BA)

- A Boolean array of size n
- The element at index i is set to True (1) whenever $i \in S$
- The element at index i is set to False (0) whenever $i \notin S$

Boolean array (BA)

- A Boolean array of size n
- The element at index i is set to True (1) whenever $i \in S$
- The element at index i is set to False (0) whenever $i \notin S$

Example

- *Let* $E = \{1, \ldots, 9\}$, *and* $S = \{1, 2, 4\}$

1	2	3	4	5	6	7	8	9
1	1	0	1	0	0	0	0	0

Operations on a subset of E

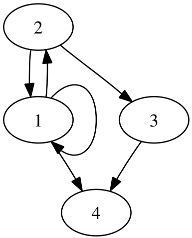
Operation		LL	BA
Initialization	$S = \emptyset$	$O(1)$	$O(n)$
Existence/selection	$\exists\, x \in S?$	$O(1)$	$O(n)$
Search	$x \in S?$	$O(n)$	$O(1)$
Insertion	$S = S \cup \{x\}$	$O(n)/O(1)$	$O(1)$
Suppression	$S = S \setminus \{x\}$	$O(1)$	$O(1)$
Traversal	$\forall x \in S$	$O(n)$	$O(n)$

Representation of a graph $(E, \overrightarrow{\Gamma})$ by a List of Arcs (LA)

- E is made of integers between 1 to n
- The arcs are considered in any order
- $\overrightarrow{\Gamma}$ is represented by two arrays T and H of size $m = |\overrightarrow{\Gamma}|$
- $T[i]$ is the first vertex (the tail) of the i-th arc
- $H[i]$ is the last vertex (the head) of the i-th arc

- E is made of integers between 1 to n
- The arcs are considered in any order
- $\overrightarrow{\Gamma}$ is represented by two arrays T and H of size $m = |\overrightarrow{\Gamma}|$
- $T[i]$ is the first vertex (the tail) of the i-th arc
- $H[i]$ is the last vertex (the head) of the i-th arc

- Example: give two arrays T and H that represent the following graph



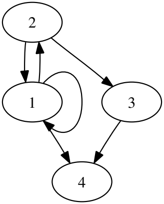
Representation of a graph (E, Γ)

- By an Array A of Linked Lists - ALL
  - $A[i]$ is a reference to a linked list that represents the set $\Gamma(i)$ of the the successors of the vertex i

Representation of a graph (E, Γ)

- By an Array A of Linked Lists - ALL
  - $A[i]$ is a reference to a linked list that represents the set $\Gamma(i)$ of the the successors of the vertex i

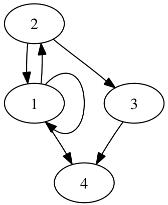
- Give the representation of the following graph by a ALL



Representation of a graph (E, Γ)

- By an Array A of Linked Lists - ALL
  - $A[i]$ is a **reference** to a linked list that represents the set $\Gamma(i)$ of the the successors of the vertex i
- By a Boolean Matrix M - BM
  - **Bi-dimensional** Boolean array of size $n \times n$
  - The row i is the representation of $\Gamma(i)$ as a Boolean array
  - $M[i][j] = 1 \Leftrightarrow j \in \Gamma(i)$

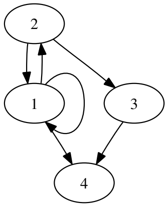
- Give the representation of the following graph by a ALL



Representation of a graph (E, Γ)

- By an Array A of Linked Lists - ALL
  - $A[i]$ is a reference to a linked list that represents the set $\Gamma(i)$ of the the successors of the vertex i
- By a Boolean Matrix M - BM
  - Bi-dimensional Boolean array of size $n \times n$
  - The row i is the representation of $\Gamma(i)$ as a Boolean array
  - $M[i][j] = 1 \Leftrightarrow j \in \Gamma(i)$

- Give the representation of the following graph by a ALL and BM



Graph-based image processing

— First algorithms: computing the symmetric of a graph —

(Professor version)

Silvio Guimarães

Graduate Program in Informatics – PPGINF
Image and Multimedia Data Science Laboratory – IMScience
Pontifical Catholic University of Minas Gerais – PUC Minas

Aug 2022

Outline of the lecture

1 Definitions and first examples of graphs

2 Representation of a graph in a computer memory

3 First algorithms: computing the symmetric of a graph

4 Some remarkable graphs

Symmetric of a graph

Definition

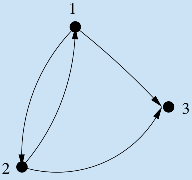
- *Let $G = (E, \Gamma)$ be a graph*
- *The* ***symmetric*** *of G is the graph $G^{-1} = (E, \Gamma^{-1})$ defined by*
  - $\forall x \in E$, $\Gamma^{-1}(x) = \{y \in E \mid x \in \Gamma(y)\}$

Symmetric of a graph

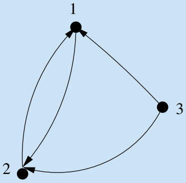
Definition

- *Let $G = (E, \Gamma)$ be a graph*
- *The **symmetric** of G is the graph $G^{-1} = (E, \Gamma^{-1})$ defined by*
  - $\forall x \in E$, $\Gamma^{-1}(x) = \{y \in E \mid x \in \Gamma(y)\}$

Example



Graphe G



Symétrique de G

Map Γ^{-1}

Remark

- Γ^{-1} *maps any vertex of G to the set of its predecessors*
- $y \in \Gamma^{-1}(x) \Leftrightarrow x \in \Gamma(y)$

Computing the symmetric of a graph

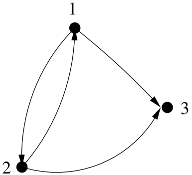
Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;
For each $y \in E$ **do**
    For each $x \in \Gamma(x)$ **do**
        $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

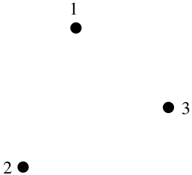
Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; Results : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;
For each $y \in E$ **do**
  For each $x \in \Gamma(x)$ **do**
    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)



(E, Γ^{-1})

Computing the symmetric of a graph

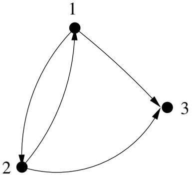
Algorithm SYM_1 (Data: (E, Γ) ; Results : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

For each $y \in E$ **do**

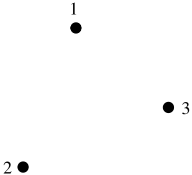
  For each $x \in \Gamma(x)$ **do**

    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)

$y = 1$



(E, Γ^{-1})

Computing the symmetric of a graph

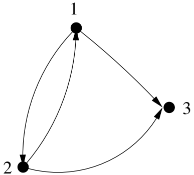
Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

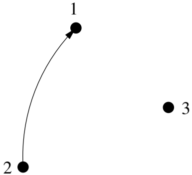
For each $y \in E$ **do**

  For each $x \in \Gamma(x)$ **do**

    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)
$y = 1$

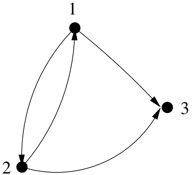


(E, Γ^{-1})
$x = 2$

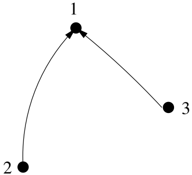
Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;
For each $y \in E$ **do**
  For each $x \in \Gamma(x)$ **do**
    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)
$y = 1$

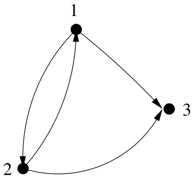


(E, Γ^{-1})
$x = 3$

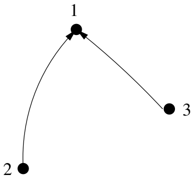
Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;
For each $y \in E$ **do**
  For each $x \in \Gamma(x)$ **do**
    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)
$y = 2$



(E, Γ^{-1})

Computing the symmetric of a graph

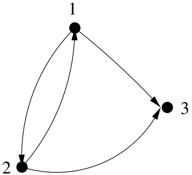
Algorithm SYM_1 (Data: (E, Γ) ; Results : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

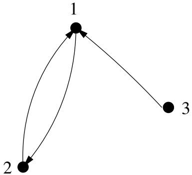
For each $y \in E$ **do**

  For each $x \in \Gamma(x)$ **do**

    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E, Γ)
$y = 2$

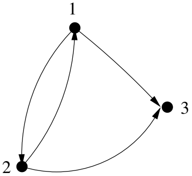


(E, Γ^{-1})
$x = 1$

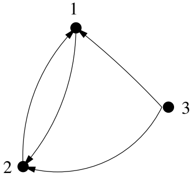
Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E,Γ) ; Results : Γ^{-1})

For each $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;
For each $y \in E$ **do**
  For each $x \in \Gamma(x)$ **do**
    $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;



(E,Γ)
$y = 2$



(E,Γ^{-1})
$x = 3$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

  3 **For each** $x \in \Gamma(x)$ **do**

    4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented by a BM)

- *line 1:* $O(n^2)$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

  3 **For each** $x \in \Gamma(x)$ **do**

    4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented by a BM)

- *line 1:* $O(n^2)$
- *line 2:* $O(n)$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

  3 **For each** $x \in \Gamma(x)$ **do**

    4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented by a BM)

- *line 1:* $O(n^2)$
- *line 2:* $O(n)$
- *line 3:* $O(n^2)$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

  3 **For each** $x \in \Gamma(x)$ **do**

    4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented by a BM)

- *line 1:* $O(n^2)$
- *line 2:* $O(n)$
- *line 3:* $O(n^2)$
- *line 4:* $O(m)$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

    3 **For each** $x \in \Gamma(x)$ **do**

        4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented by a BM)

- *line 1:* $O(n^2)$
- *line 2:* $O(n)$
- *line 3:* $O(n^2)$
- *line 4:* $O(m)$
- *Overall complexity:* $O(n^2 + m) = O(n^2)$

Computing the symmetric of a graph

Algorithm SYM_1 (Data: (E, Γ) ; **Results** : Γ^{-1})

1 **For each** $x \in E$ **do** $\Gamma^{-1}(x) := \emptyset$;

2 **For each** $y \in E$ **do**

  3 **For each** $x \in \Gamma(x)$ **do**

    4 $\Gamma^{-1}(x) := \Gamma^{-1}(x) \cup \{y\}$;

Complexity (If Γ and Γ^{-1} are implemented a ALL)

- *Line 1:* $O(n)$
- *Line 2:* $O(n)$
- *Line 3:* $O(n + m)$
- *Line 4:* $O(m)$
- *Overall complexity:* $O(n + m)$

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; **Results** : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

   3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; Results : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

    3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Complexity (If $\overrightarrow{\Gamma}$ and $\overrightarrow{\Gamma^{-1}}$ are implemented by LA)

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; **Results** : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

  3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Complexity (If $\overrightarrow{\Gamma}$ and $\overrightarrow{\Gamma^{-1}}$ are implemented by LA)

- *Line 1:* $O(1)$

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; Results : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

    3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Complexity (If $\overrightarrow{\Gamma}$ and $\overrightarrow{\Gamma^{-1}}$ are implemented by LA)

- *Line 1:* $O(1)$
- *Line 2 :* $O(m)$

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; **Results** : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

  3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Complexity (If $\overrightarrow{\Gamma}$ and $\overrightarrow{\Gamma^{-1}}$ are implemented by LA)

- *Line 1:* $O(1)$
- *Line 2 :* $O(m)$
- *Line 3 :* $O(m)$

Computing the symmetric of a graph

Algorithm SYM_2 (Data: $(E, \overrightarrow{\Gamma})$; **Results** : $\overrightarrow{\Gamma^{-1}}$)

1 $\overrightarrow{\Gamma^{-1}} := \emptyset$;

2 **For each** $(x, y) \in \overrightarrow{\Gamma}$ **do**

  3 $\Gamma^{-1} := \Gamma^{-1} \cup \{(y, x)\}$;

Complexity (If $\overrightarrow{\Gamma}$ and $\overrightarrow{\Gamma^{-1}}$ are implemented by LA)

- *Line 1:* $O(1)$
- *Line 2 :* $O(m)$
- *Line 3 :* $O(m)$
- *Overall complexity* $O(m)$

Graph-based image processing

— Some remarkable graphs —

(Professor version)

Silvio Guimarães

Graduate Program in Informatics – PPGINF
Image and Multimedia Data Science Laboratory – IMScience
Pontifical Catholic University of Minas Gerais – PUC Minas

Aug 2022

Outline of the lecture

1 Definitions and first examples of graphs

2 Representation of a graph in a computer memory

3 First algorithms: computing the symmetric of a graph

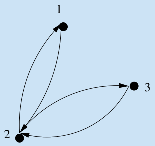
4 Some remarkable graphs

Symmetric and asymmetric graphs

Definition

- *A graph $G = (E, \Gamma)$ is said* **symmetric** *if* $\overrightarrow{\Gamma^{-1}} = \overrightarrow{\Gamma}$
  - *In other words, G is symmetric if:* $\forall x, y \in E \ x \in \Gamma(y) \Leftrightarrow y \in \Gamma(x)$

Example



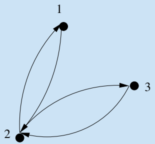
Symmetric graph

Symmetric and asymmetric graphs

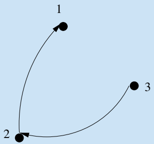
Definition

- *A graph* $G = (E, \Gamma)$ *is said* **symmetric** *if* $\overrightarrow{\Gamma^{-1}} = \overrightarrow{\Gamma}$
  - *In other words,* G *is symmetric if:* $\forall x, y \in E \; x \in \Gamma(y) \Leftrightarrow y \in \Gamma(x)$
- *A graph* $G = (E, \Gamma)$ *is said* **asymmetric** *if* $\overrightarrow{\Gamma} \cap \overrightarrow{\Gamma^{-1}} = \emptyset$
  - G *is asymmetric if:* $\forall x, y \in E$, $(x, y) \in \overrightarrow{\Gamma} \implies (y, x) \notin \overrightarrow{\Gamma}$

Example



Symmetric graph



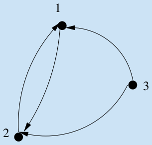
Asymmetric graph

Symmetric closure

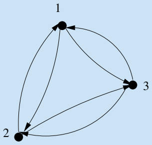
Definition

- *Let* $G = (E, \overrightarrow{\Gamma})$ *be a graph*
- *The* **symmetric closure** *of G is the graph* $G_s = (E, \overrightarrow{\Gamma_s})$ *defined by* $\overrightarrow{\Gamma_s} = \overrightarrow{\Gamma} \cup \overrightarrow{\Gamma^{-1}}$

Example



Graph G



Symmetric closure of G

Undirected graph

Definition

- *A* ***undirected graph*** *is a pair* $(E, \overline{\Gamma})$ *where* E *is a finite set and where* $\overline{\Gamma}$ *is a subset of* $\{ \{x, y\} \mid x \in E,\ y \in E \}$

Undirected graph

Definition

- *A* ***undirected graph*** *is a pair* $(E, \overline{\Gamma})$ *where* E *is a finite set and where* $\overline{\Gamma}$ *is a subset of* $\{\ \{x, y\} \mid x \in E,\ y \in E\ \}$

Example

- $E = \{a, b, c, d\}$ *and* $\overline{\Gamma} = \{\ \{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\ \}$

Undirected graph

Definition

- *A* ***undirected graph*** *is a pair* $(E, \overline{\Gamma})$ *where* E *is a finite set and where* $\overline{\Gamma}$ *is a subset of* $\{\ \{x, y\} \mid x \in E,\ y \in E\ \}$
- *Any element of* $\overline{\Gamma}$ *is called an* ***edge*** *of the graph*

Example

- $E = \{a, b, c, d\}$ *and* $\overline{\Gamma} = \{\ \{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\ \}$

Undirected graph

Definition

- *A **undirected graph** is a pair* $(E, \overline{\Gamma})$ *where* E *is a finite set and where* $\overline{\Gamma}$ *is a subset of* $\{\ \{x, y\} \mid x \in E,\ y \in E\ \}$
- *Any element of* $\overline{\Gamma}$ *is called an **edge** of the graph*
- *The edge* $\{x, y\} \in \overline{\Gamma}$ *is **adjacent** to the vertices* x *and* y

Example

- $E = \{a, b, c, d\}$ *and* $\overline{\Gamma} = \{\ \{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\ \}$

Undirected graphs & symmetric graphs

- Let $(E, \overline{\Gamma})$ be a undirected graph,
- We associate to $\overline{\Gamma}$ the map $\Gamma_{n.o} : E \to \mathcal{P}(E)$ defined by $\Gamma_{n.o}(x) = \{y \in E \mid \{x, y\} \in \overline{\Gamma}\}$

Undirected graphs & symmetric graphs

- Let $(E, \overline{\Gamma})$ be a undirected graph,
- We associate to $\overline{\Gamma}$ the map $\Gamma_{n.o} : E \rightarrow \mathcal{P}(E)$ defined by $\Gamma_{n.o}(x) = \left\{y \in E \mid \{x, y\} \in \overline{\Gamma}\right\}$

Remark

- *The graph $(E, \Gamma_{n.o})$ is symmetric*

Undirected graphs & symmetric graphs

- Let $(E, \overline{\Gamma})$ be a undirected graph,
- We associate to $\overline{\Gamma}$ the map $\Gamma_{n.o} : E \to \mathcal{P}(E)$ defined by $\Gamma_{n.o}(x) = \{y \in E \mid \{x, y\} \in \overline{\Gamma}\}$

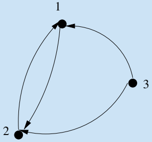
Remark

- *The graph $(E, \Gamma_{n.o})$ is symmetric*
- *The datum of a symmetric graph is equivalent to the datum of a undirected graph*

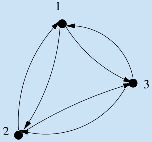
Undirected graph associated to a directed graph

- We associate to any directed graph (E, Γ), the undirected graph $(E, \overline{\Gamma})$ defined by
  - $\{x, y\} \in \overline{\Gamma} \Leftrightarrow (x, y) \in \overrightarrow{\Gamma}$ or $(y, x) \in \overrightarrow{\Gamma}$

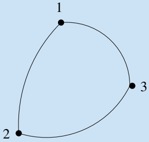
Example



a graph



its symmetric closure



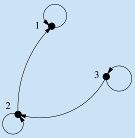
its associated undirected graph

Reflexive graph

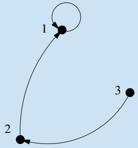
Definition

- *Let $G = (E, \Gamma)$ be a graph*
- *G is a* ***reflexive*** *if $\forall x \in E$, $x \in \Gamma(x)$*
- *G is* ***irreflexive (or without loop)*** *if $\forall x \in E$, $x \notin \Gamma(x)$*
- *Any arc $(x, x) \in \Gamma$ is called a* ***loop of G***

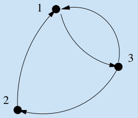
Example



Reflexive graph



Non reflexive graph and non irreflexive



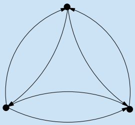
Irreflexive graph (without loop)

Complete graph

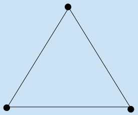
Definition

- *A graph* $(E, \overrightarrow{\Gamma})$ *without loop is a* **complete graph** *if for any pair* (x, y) *of vertices, we have* $(x, y) \in \overrightarrow{\Gamma}$

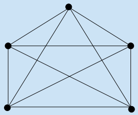
Example



Complete graph (directed) over 3 vertices



Complete graph (undirected) over 3 vertices



Complete graph (undirected) over 5 vertices

Exercise

- Let $(E, \overline{\Gamma})$ be the complete undirected graph whose vertex set is $E = \{1, 2, \ldots, n, n+1\}$.
- Question 1. Describe two different ways to count the edges of G
- Question 2. Deduce from question 1, the equality $1 + 2 + \ldots + n = \frac{n(n+1)}{2}$

Acknowledgement

Thanks to the Prof. Jean Cousty at ESIEE/France that gently sent me the slides used in the Morpho, Graph and Image course. Some slides of the Graph-based Image Processing course at PPGINF/PUC Minas under supervision of Prof. Silvio Guimarães will be adapted versions of that course.

- Course - MorphoGraph and Imagery
  https://perso.esiee.fr/ coustyj/EnglishMorphoGraph/

- Jean Cousty
  - ESIEE Paris, Département Informatique
  - Université Paris-Est, LIGM (UMR CNRS, ESIEE...)
  - E-mail: `j.cousty@esiee.fr`