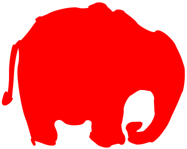
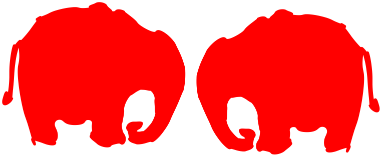
Find out the difference(s)?

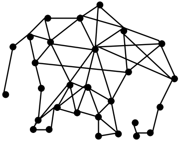


Shape 1

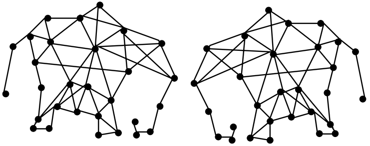


Shape 2

Find out the difference(s)?



Graph G_1



Graph G_2

Graph-based image processing

— Connectivity in graphs —

(Professor version)

(Connectivity in graphs)

Silvio Guimarães

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

Outline of the lecture

Graph-based image processing

— Path —

(Professor version)

(Connectivity in graphs)

Silvio Guimarães

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

Path

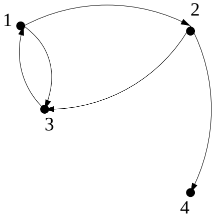
Definition

- *Let* $G = (E, \Gamma)$ *be a graph, and let* x *and* y *be two vertices in* E
- *A* ***path from*** x ***to*** y ***(in*** G***)*** *is a sequence* $\pi = (x_0, \ldots, x_\ell)$ *of vertices in* E *such that*
  - $\forall i \in \{1, \ldots, \ell\}$, $x_i \in \Gamma(x_{i-1})$
  - $x_0 = x$ *and* $x_\ell = y$

Path

Definition

- *Let $G = (E, \Gamma)$ be a graph, and let x and y be two vertices in E*
- *A **path from x to y (in G)** is a sequence $\pi = (x_0, \ldots, x_\ell)$ of vertices in E such that*
  - $\forall i \in \{1, \ldots, \ell\}$, $x_i \in \Gamma(x_{i-1})$
  - $x_0 = x$ *and* $x_\ell = y$



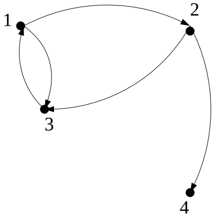
Example

$\pi = (1, 2, 3)$ *is a path*

Path

Definition

- *Let $G = (E, \Gamma)$ be a graph, and let x and y be two vertices in E*
- *A* ***path from x to y (in G)*** *is a sequence $\pi = (x_0, \ldots, x_\ell)$ of vertices in E such that*
  - $\forall i \in \{1, \ldots, \ell\}$, $x_i \in \Gamma(x_{i-1})$
  - $x_0 = x$ *and* $x_\ell = y$
- *If $\pi = (x_0, \ldots, x_\ell)$ is a path, ℓ is called its* ***length***

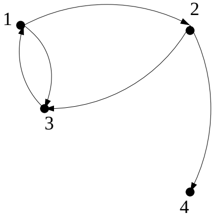


Example

$\pi = (1, 2, 3)$ *is a path of length 2*

Some remarkable paths

- A path of length 0 is called a **trivial path**

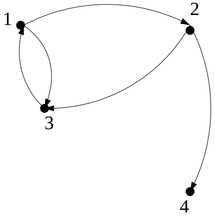


Example

- (3) *is a trivial path*

Some remarkable paths

- A path of length 0 is called a trivial path
- A non-trivial path $\pi = (x_0, \ldots, x_\ell)$ is called a circuit if $x_0 = x_\ell$

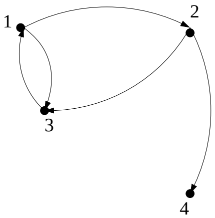


Example

- (3) *is a trivial path*
- $(1, 2, 3, 1)$ *is a circuit*

Some remarkable paths

- A path of length 0 is called a **trivial path**
- A non-trivial path $\pi = (x_0, \dots, x_\ell)$ is called a **circuit** if $x_0 = x_\ell$
- A path $\pi = (x_0, \dots, x_\ell)$ is called **elementary** if any two of its vertices are distinct (except possibly x_0 and x_ℓ): $\forall i, j \in \{0, \dots, \ell\}$, $i \neq j \implies x_i \neq x_j$ (where $\{i, j\} \neq \{0, \ell\}$)

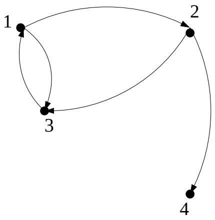


Example

- (3) *is a trivial path*
- $(1, 2, 3, 1)$ *is a circuit that is elementary*

Some remarkable paths

- A path of length 0 is called a **trivial path**
- A non-trivial path $\pi = (x_0, \ldots, x_\ell)$ is called a **circuit** if $x_0 = x_\ell$
- A path $\pi = (x_0, \ldots, x_\ell)$ is called **elementary** if any two of its vertices are distinct (except possibly x_0 and x_ℓ): $\forall i, j \in \{0, \ldots, \ell\}$, $i \neq j \implies x_i \neq x_j$ (where $\{i, j\} \neq \{0, \ell\}$)



Example

- (3) *is a trivial path*
- $(1, 2, 3, 1)$ *is a circuit that is elementary*
- $(1, 3, 1, 2, 3, 1)$ *is a circuit that is not elementary*

Basic properties

Property

- *Any path* π *from* x *to* y *contains an elementary path from* x *to* y

Basic properties

Property

- *Any path π from x to y contains an elementary path from x to y*
- *The length of an elementary circuit is less than n (where $n = |E|$)*

Basic properties

Property

- *Any path π from x to y contains an elementary path from x to y*
- *The length of an elementary circuit is less than n (where $n = |E|$)*
- *The length of an elementary path which is not a circuit is less than $n - 1$*

Undirected paths and cycles

Definition

- *Let* $G_s = (E, \Gamma_s)$ *be the symmetric closure of* G

Undirected paths and cycles

Definition

- *Let $G_s = (E, \Gamma_s)$ be the symmetric closure of G*
- *Any path in G_s is called an* ***undirected path (in G)***

Undirected paths and cycles

Definition

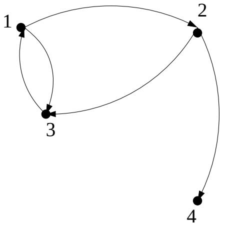
- *Let* $G_s = (E, \Gamma_s)$ *be the symmetric closure of* G
- *Any path in* G_s *is called an* *undirected path (in G)*
- *A* *cycle (in G)* *is a circuit in* G_s *that does not pass twice by the same edge*

Undirected paths and cycles

Definition

- *Let $G_s = (E, \Gamma_s)$ be the symmetric closure of G*
- *Any path in G_s is called an **undirected path (in G)***
- *A **cycle (in G)** is a circuit in G_s that does not pass twice by the same edge*

Remark. If $\pi = (x_0, \ldots, x_\ell)$ is an undirected path, then $\pi' = (x_\ell, \ldots, x_0)$ is an undirected path (since (E, Γ_s) is a symmetric graph, which implies that $x_i \in \Gamma_s(x_{i-1}) \Leftrightarrow x_{i-1} \in \Gamma_s(x_i)$)



Example

- $(1, 2, 1)$ *is not a path*
- $(1, 2, 3, 1, 2, 4)$ *is a path*
- $(4, 2, 1)$ *is an undirected path*
- $(1, 3, 2, 1)$ *is not a circuit*
- $(1, 3, 1)$ *is not a cycle*
- $(1, 2, 3)$ *is a path*
- $(4, 2, 1)$ *is not a path*
- $(1, 2, 3, 1)$ *is a circuit*
- $(1, 3, 2, 1)$ *is a cycle*

Graph-based image processing

— Connectivity —

(Professor version)

(Connectivity in graphs)

Silvio Guimarães

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

Connected component

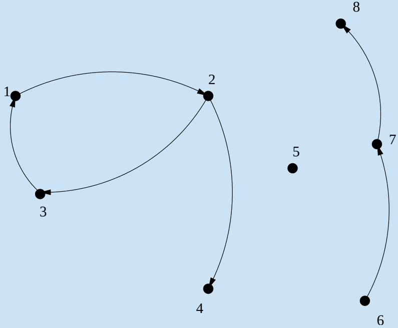
Definition

- *Let* $x \in E$. *The* *connected component of G containing x* *is the subset* C_x *of* E *defined by:*
  - $C_x = \{y \in E \mid \textit{there exists an undirected path from } x \textit{ to } y\}$

Definition

- *Let $x \in E$. The **connected component of G containing x** is the subset C_x of E defined by:*
  - $C_x = \{y \in E \mid \text{there exists an undirected path from } x \text{ to } y\}$

Example



- $C_1 = \{1, 2, 3, 4\} = C_2 = C_3 = C_4$
- $C_5 = \{5\}$
- $C_6 = C_7 = C_8 = \{6, 7, 8\}$

Connected component as equivalence classes

Property

1. $\forall x \in E,\ x \in C_x$ *reflexivity*
2. $\forall x, y \in E,\ y \in C_x \implies x \in C_y$ *symmetry*
3. $\forall x, y, z \in E,\ [y \in C_x \textit{ and } z \in C_y] \implies z \in C_x$ *transitivity*

Proof.

1. $\forall x \in E$, (x) is a (trivial) undirected path, thus $x \in C_x$
2. $y \in C_x \implies \exists$ an undirected path $\pi = (x_0, \ldots, , x_\ell)$ from x to y
   $\implies \pi' = (x_\ell, \ldots, x_0)$ is an undirected path from y to x
   $\implies x \in C_y$
3. $[y \in C_x$ and $z \in C_y] \implies [\exists$ an undirected path $\pi = (x_0, \ldots, x_\ell)$ from x to y and $\exists$ an undirected path $\pi' = (y_0, \ldots, y_m)$ from y to $z] \implies \pi'' = (x_0, \ldots, x_\ell, y_1, \ldots, y_m)$ is an undirected path from x to $z \implies z \in C_x$ $\square$

Strongly connected component

Definition

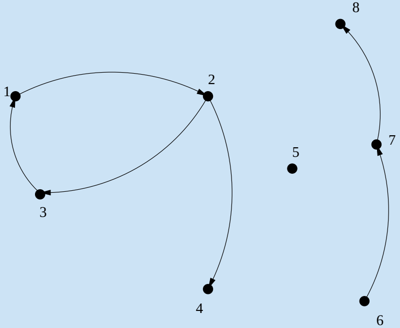
- *Let* $x \in E$. *The* ***strongly connected component*** *(of* G*) containing* x *is the subset* C'_x *of* E *defined by*
  - $C'_x = \{y \in E \mid \exists \text{ a path from } x \text{ to } y \text{ and } \exists \text{ a path from } y \text{ to } x\}$

Strongly connected component

Definition

- Let $x \in E$. The **strongly connected component** (of G) containing x is the subset C'_x of E defined by
  - $C'_x = \{y \in E \mid \exists \text{ a path from } x \text{ to } y \text{ and } \exists \text{ a path from } y \text{ to } x\}$

Example



- $C'_1 = \{1, 2, 3\} = C'_2 = C'_3$
- $C'_4 = \{4\}$; $C'_5 = \{5\}$
- $C'_6 = \{6\}$; $C'_7 = \{7\}$
- $C'_8 = \{8\}$

Strongly connected components as equivalence classes?

Exercise. Are the strongly connected components of a graphs equivalence classes? Proof or counter-example?

Graph-based image processing

— Algorithms —

(Professor version)

(Connectivity in graphs)

Silvio Guimarães

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

- We will first study a characterization of connected components and of strongly connected components based on morphological dilation
- This will allow us to propose efficient algorithm to compute them

The morphological dilation $\delta_\Gamma(X)$ of a subset X of E is th union of the sets of successors of all vertices in X

Iterated operators

- Let γ be an operator and $i \in \mathbb{N}$
- We denote by γ^i the operator defined by
  1. $\gamma^i = \gamma\gamma^{i-1}$
  2. $\gamma^0 = Id$ (*i.e.* $\forall X \subseteq E, \gamma^0(X) = X$)

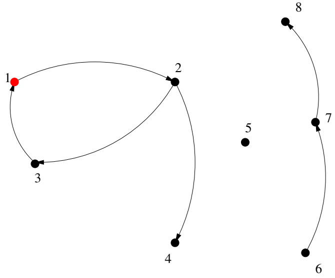
Iterated operators

- Let γ be an operator and $i \in \mathbb{N}$
- We denote by γ^i the operator defined by
  1. $\gamma^i = \gamma\gamma^{i-1}$
  2. $\gamma^0 = Id$ (*i.e.* $\forall X \subseteq E, \gamma^0(X) = X$)

Example (iterated dilation)

1. $\delta_\Gamma^0(X) = X$
2. $\delta_\Gamma^1(X) = \delta_\Gamma(\delta_\Gamma^0(X)) = \delta_\Gamma(X)$
3. $\delta_\Gamma^2(X) = \delta_\Gamma(\delta_\Gamma^1(X)) = \delta_\Gamma(\delta_\Gamma(X))$
4. $\delta_\Gamma^3(X) = \delta_\Gamma(\delta_\Gamma^2(X)) = \delta_\Gamma(\delta_\Gamma(\delta_\Gamma(X)))$
5. $\ldots$
6. $\delta_\Gamma^i(X) = \delta_\Gamma(\delta_\Gamma^{i-1}(X)) = \delta_\Gamma(\ldots\delta_\Gamma(X)\ldots)$

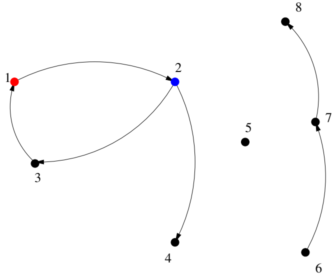
Illustration: iterated dilation



Example

- $X = \delta^0_\Gamma(X) = \{1\}$

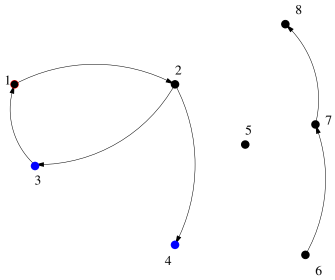
Illustration: iterated dilation



Example

- $X = \delta_{\Gamma}^{0}(X) = \{1\}$
- $\delta_{\Gamma}^{1}(X) = \{2\}$

Illustration: iterated dilation



Example

- $X = \delta_\Gamma^0(X) = \{1\}$
- $\delta_\Gamma^1(X) = \{2\}$
- $\delta_\Gamma^2(X) = \{3, 4\}$

Iterated dilation and paths of given length

Property

- *Let* $x \in E$ *and* $i \in \mathbb{N}$
- *The two following equalities* ***hold true***
  - $\delta^i_\Gamma(\{x\}) = \{y \in E \mid \exists$ *a path from* x *to* y *of length* $i\}$
  - $\delta^i_{\Gamma^{-1}}(\{x\}) = \{y \in E \mid \exists$ *a path from* y *to* x *of length* $i\}$

Iterated dilation and paths of given length

Property

- *Let $x \in E$ and $i \in \mathbb{N}$*
- *The two following equalities hold true*
  - $\delta^i_\Gamma(\{x\}) = \{y \in E \mid \exists$ *a path from x to y of length i*$\}$
  - $\delta^i_{\Gamma^{-1}}(\{x\}) = \{y \in E \mid \exists$ *a path from y to x of length i*$\}$

- We define for any $X \subseteq E$ and any $p \in \mathbb{N}$
  - $\hat{\delta}^p_\Gamma(X) = \cup\{\delta^i_\Gamma(X) \mid i \in \{0, \ldots, p\}\}$

Iterated dilation and paths of given length

Property

- *Let $x \in E$ and $i \in \mathbb{N}$*
- *The two following equalities* ***hold true***
  - $\delta^i_\Gamma(\{x\}) = \{y \in E \mid \exists$ *a path from x to y of length i*$\}$
  - $\delta^i_{\Gamma^{-1}}(\{x\}) = \{y \in E \mid \exists$ *a path from y to x of length i*$\}$

- We define for any $X \subseteq E$ and any $p \in \mathbb{N}$
  - $\hat{\delta}^p_\Gamma(X) = \cup\{\delta^i_\Gamma(X) \mid i \in \{0, \ldots, p\}\}$
- The set $\hat{\delta}^p_\Gamma(X)$ is called the **pre-closure of X of rank p**

Transitive closure and paths

Corolary

- *Let* $x \in E$
- *The two following equalities* ***hold true***
  - $\hat{\delta}^{p}_{\Gamma}(\{x\}) = \{y \in E \mid \exists \text{ a path from } x \text{ to } y \text{ of length } \leq p\}$
  - $\hat{\delta}^{p}_{\Gamma^{-1}}(\{x\}) = \{y \in E \mid \exists \text{ a path from } y \text{ to } x \text{ of length } \leq p\}$

Transitive closure and paths

Corolary

- *Let $x \in E$*
- *The two following equalities* ***hold true***
  - $\delta_{\Gamma}^{\hat{p}}(\{x\}) = \{y \in E \mid \exists$ *a path from x to y of length* $\leq p\}$
  - $\delta_{\Gamma^{-1}}^{\hat{p}}(\{x\}) = \{y \in E \mid \exists$ *a path from y to x of length* $\leq p\}$

- Let $x \in E$, the **(transitive) closure of $\{x\}$** is the set
- $\delta_{\Gamma}^{\hat{\infty}}(\{x\}) = \{y \in E \mid \exists$ a path from x to $y\}$

Transitive closure and paths

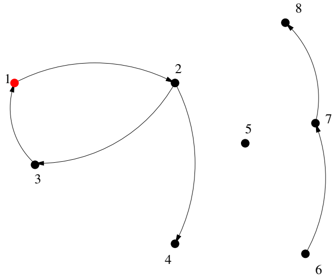
Corolary

- *Let $x \in E$*
- *The two following equalities* **hold true**
  - $\delta_\Gamma^{\hat{p}}(\{x\}) = \{y \in E \mid \exists \textit{ a path from } x \textit{ to } y \textit{ of length } \leq p\}$
  - $\delta_{\Gamma^{-1}}^{\hat{p}}(\{x\}) = \{y \in E \mid \exists \textit{ a path from } y \textit{ to } x \textit{ of length } \leq p\}$

- Let $x \in E$, the **(transitive) closure of $\{x\}$** is the set
- $\delta_\Gamma^{\hat{\infty}}(\{x\}) = \{y \in E \mid \exists \text{ a path from } x \text{ to } y\}$

<u>*Exercise.*</u> Prove that $\delta_\Gamma^{\hat{\infty}}(\{x\}) = \delta_\Gamma^{\widehat{n-1}}(\{x\})$ (where $n = |E|$)

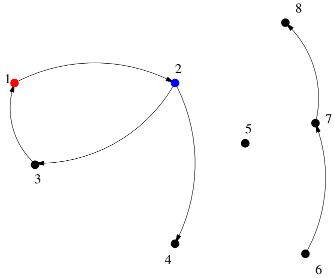
Illustration: transitive closure



Example

- $X = \delta^0_\Gamma(X) = \{1\}$
- $X = \hat{\delta}^0_\Gamma(X) = \{1\}$

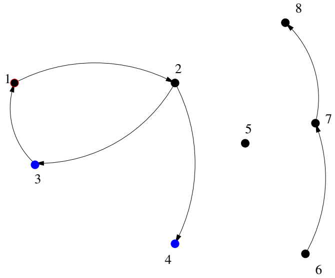
Illustration: transitive closure



Example

- $X = \delta_\Gamma^0(X) = \{1\}$
- $\delta_\Gamma^1(X) = \{2\}$
- $X = \hat{\delta}_\Gamma^0(X) = \{1\}$
- $\hat{\delta}_\Gamma^1(X) = \{1, 2\}$

Illustration: transitive closure



Example

- $X = \delta_\Gamma^0(X) = \{1\}$
- $\delta_\Gamma^1(X) = \{2\}$
- $\delta_\Gamma^2(X) = \{3, 4\}$

- $X = \hat{\delta}_\Gamma^0(X) = \{1\}$
- $\hat{\delta}_\Gamma^1(X) = \{1, 2\}$
- $\hat{\delta}_\Gamma^2(X) = \{1, 2, 3, 4\} = \hat{\delta}_\Gamma^7(X)$

Property

- *Let $x \in E$. Let C_x and C'_x be respectively the connected components and the strongly connected components containing x*

Property

- *Let $x \in E$. Let C_x and C'_x be respectively the connected components and the strongly connected components containing x*
- *The two following equalities hold true*
  - $C'_x = \delta_{\Gamma}^{\widehat{n-1}}(\{x\}) \cap \delta_{\Gamma^{-1}}^{\widehat{n-1}}(\{x\})$
  - $C_x = \delta_{\Gamma_s}^{\widehat{n-1}}(\{x\}) = \delta_{\Gamma_s}^{\widehat{n-1}}(\{x\})$
- *where $n = |E|$ and Γ_s is the symmetric closure of Γ*

Property

- *Let $x \in E$. Let C_x and C'_x be respectively the connected components and the strongly connected components containing x*
- *The two following equalities hold true*
  - $C'_x = \delta_{\Gamma}^{\widehat{n-1}}(\{x\}) \cap \delta_{\Gamma^{-1}}^{\widehat{n-1}}(\{x\})$
  - $C_x = \delta_{\Gamma_s}^{\widehat{n-1}}(\{x\}) = \delta_{\Gamma_s}^{\widehat{n-1}}(\{x\})$
- *where $n = |E|$ and Γ_s is the symmetric closure of Γ*

- To compute the connected and strongly connected components containing x, it is thus sufficient to compute the transitive closure of $\{x\}$ for the graphs (E, Γ), (E, Γ^{-1}) and (E, Γ_s)

Algorithm TRANS_NAIV (Data: (E, Γ), $x \in E$;

Results: $Z = \hat{\delta_\Gamma^{n-1}}(\{x\})$)

- $X := \{x\}$; $Y := \emptyset$; $Z := \{x\}$;
- **For each** i from 1 to $n-1$ **do**
  - $Y := \text{DIL}((E, \Gamma), X)$; /* $Y = \delta_\Gamma^i(\{x\})$ */
  - $Z := Z \cup Y$; /* $Z = \hat{\delta}_\Gamma^i(\{x\})$ */
  - $X := Y$; $Y := \emptyset$;

Naive algorithm for the transitive closure of $\{x\}$

Algorithm TRANS_NAIV (Data: (E, Γ), $x \in E$;

Results: $Z = \hat{\delta_\Gamma^{n-1}}(\{x\}))$

- $X := \{x\}$; $Y := \emptyset$; $Z := \{x\}$;
- **For each** i from 1 to $n-1$ **do**
  - $Y := \mathrm{DIL}((E, \Gamma), X)$; /* $Y = \delta_\Gamma^i(\{x\})$ */
  - $Z := Z \cup Y$; /* $Z = \hat{\delta}_\Gamma^i(\{x\})$ */
  - $X := Y$; $Y := \emptyset$;

Complexity

- *By using the algorithm DIL studied during practical session # 1*
- *The complexity of TRANS_NAIV is*
  - $O(n^2 + nm)$ *(where* $n = |E|$ *and* $m = |\overline{\Gamma}|$*)*

Algorithm TRANS (Data: (E, Γ), $x \in E$;

Result: $Z = \hat{\delta_\Gamma^{n-1}}(\{x\}))$

$X := \{x\}$; $Y := \emptyset$; $Z := \{x\}$;

For each i from 1 to $n-1$ **do**

- **While** $\exists y \in X$ **do**
  - $X := X \setminus \{y\}$;
  - **For each** $z \in \Gamma(y)$ **do**
    - **If** $z \notin Z$ **then** $Y := Y \cup \{z\}$; $Z := Z \cup \{z\}$;
- $X := Y$; $Y := \emptyset$;

Algorithm TRANS (Data: (E, Γ), $x \in E$;

Result: $Z = \delta_{\Gamma}^{\widehat{n-1}}(\{x\}))$

$X := \{x\}$; $Y := \emptyset$; $Z := \{x\}$;

For each i from 1 to $n-1$ **do**

- **While** $\exists y \in X$ **do**
  - $X := X \setminus \{y\}$;
  - **For each** $z \in \Gamma(y)$ **do**
    - **If** $z \notin Z$ **then** $Y := Y \cup \{z\}$; $Z := Z \cup \{z\}$;
- $X := Y$; $Y := \emptyset$;

- If X and Y are represented by LLs and if Z is represented by a BA
- The time-complexity of TRANS is linear
  - $O(n+m)$ (where $n = |E|$ and $m = |\overline{\Gamma}|$)

Algorithm TRANS can be further simplified without changing its complexity:

Algorithm TRANS (Data: (E, Γ), $x \in E$;

Result : $Z = \delta_{\Gamma}^{\hat{n-1}}(\{x\}))$

- $X := \{x\}$; $Z := \{x\}$;
- **While** $\exists y \in X$ **do**
  - $X := X \setminus \{y\}$;
  - **For each** $z \in \Gamma(y)$ **do**
    - **If** $z \notin Z$ **then** $X := X \cup \{z\}$; $Z := Z \cup \{z\}$;

Computing the strongly connected component containing x

Algorithm SCC (Data: (E, Γ), $x \in E$;

Result: $Z = C'_x$)

- $X :=$ TRANS$((E, \Gamma), x)$;
- $\Gamma^{-1} :=$ SYM_1(E, Γ);
- $Y :=$ TRANS$((E, \Gamma^{-1}), x)$
- $Z := X \cap Y$;

Computing the strongly connected component containing x

Algorithm SCC (Data: (E, Γ), $x \in E$;

Result: $Z = C'_x$)

- $X := \text{TRANS}((E, \Gamma), x)$;
- $\Gamma^{-1} := \text{SYM\_1}(E, \Gamma)$;
- $Y := \text{TRANS}((E, \Gamma^{-1}), x)$
- $Z := X \cap Y$;

- Using
  - the linear-time algorithm SYM_1 (first lecture)
  - the linear-time TRANS
- The time-complexity of SCC is linear:
  - $O(n + m)$ (where $n = |E|$ and $m = |\overline{\Gamma}|$)

Computing the connected component containing x

Algorithm CC (Data: (E, Γ), $x \in E$;

Result: $Y = C_x$)

- $\Gamma^{-1} :=$ SYM_1(E, Γ);
- $\Gamma_s := \Gamma \cup \Gamma^{-1}$;
- $Y :=$ TRANS$((E, \Gamma_s), x)$

Computing the connected component containing x

Algorithm CC (Data: (E, Γ), $x \in E$;

Result: $Y = C_x$)

- $\Gamma^{-1} :=$ SYM_1(E, Γ);
- $\Gamma_s := \Gamma \cup \Gamma^{-1}$;
- $Y :=$ TRANS$((E, \Gamma_s), x)$

- Using
  - the linear-time algorithm SYM_1 (first lecture)
  - the linear-time TRANS
- The time-complexity of CC is linear:
  - $O(n + m)$ (where $n = |E|$ and $m = |\overline{\Gamma}|$)

Graph-based image processing

— Degrees —

(Professor version)

(Connectivity in graphs)

Silvio Guimarães

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

Application of graphs properties

Problem

- *In a charity diner, is there always two people having the same number of friends who are present in the diner?*
- *Can we draw in the plan five distinct lines such that any of them has exactly three intersection points with the others?*

Application of graphs properties

Problem

- *In a charity diner, is there always two people having the same number of friends who are present in the diner?*
- *Can we draw in the plan five distinct lines such that any of them has exactly three intersection points with the others?*

- In order to answer these questions read first the two next slides!

Degree

- Let $G = (E, \Gamma)$ be a graph and let $x \in E$
- The outer degree of x (for G) is the value $d^+(x) = |\Gamma(x)|$
- The inner degree of x(for G) is the value $d^-(x) = |\Gamma^{-1}(x)|$
- The degree of x(for G) is the value $d(x) = d^+(x) + d^-(x)$
- Let $G' = (E, \overline{\Gamma})$ be an undirected graph
- The degree of x for G' is the number $d(x)$ of edges that are adjacent to x

Degree: exercise

- Prove that the two following propositions hold true
  - The sum of the degrees of the vertices of a graph is even
  - In any graph, there is an even number of vertices whose degree is odd

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`