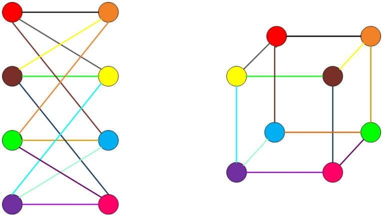


PRIMERJAVA OMREŽIJ

Teja Klemenčič
Jerneja Pikelj
Maja Smrke

10. maj 2012

Uporaba:

- matematična kemija (molekularne strukture, spojine)
- biologija (biološka omrežja)
- načrtovanje elektronskega vezja
- prepoznavanje simetrije

Izomorfizem grafov

Definicija.

Bijektivni preslikavi $\Phi : V_1 \to V_2$*, za katero je* $\forall u, v \in V_1 : \{u, v\} \in E_1 \Leftrightarrow \{\Phi(u), \Phi(v)\} \in E_2$*, pravimo* ***izomorfizem grafov****.*

Definicija.

Grafa G_1 *in* G_2 *sta* ***izomorfna****, oznaka* $G_1 \simeq G_2$*, če med njima obstaja kakšen izomorfizem.*

Definicija.

V primeru, ko je graf G_2 *kar enak grafu* G_1*, izomorfizmu grafov pravimo* ***avtomorfizem grafov****. Če množico avtomorfizmov danega grafa* G *opremimo z operacijo komponiranja preslikav, dobimo grupo, ki ji pravimo* ***grupa avtomorfizmov*** *grafa* G *in jo označimo z* $Aut(G)$*.*

Izomorfizem grafov

Definicija.

Lastnosti grafa, ki jo imajo poleg grafa samega tudi vsi z njim izomorfni grafi, pravimo ***invarianta grafa****. Ta lastnost se pri izomorfizmu ohranja.*

Preproste grafovske invariante so:

- število točk in povezav
- število podgrafov izbrane vrste (npr. trikotniki, štirikotniki, . . .)
- stopnje točk
- število komponent
- dvodelnost
- število mostov

Izomorfizem grafov

Trditev.

Izomorfizem ohranja število vozlišč, število povezav, stopnje vozlišč, število trikotnikov, ...

Trditev.

Izomorfnost je ***ekvivalenčna relacija*** $(\simeq)$.

Trditev.

Grafa G_1 *in* G_2 *sta izomorfna* $\Leftrightarrow$ G_1^c *in* G_2^c *izomorfna.*

Izomorfizem grafov

Definicija.

*Imamo graf G, množico točk $V(G) = v_1, \ldots, v_n$ in množico povezav $E(G) = e_1, \ldots, e_m$. **Matrika sosednosti** $A(G)$ neusmerjenega grafa je kvadratna matrika velikosti $n \times n$, definirana takole:*

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix},$$

kjer je

$$a_{ij} = \begin{cases} 1 & ; \textit{če } v_i \sim v_j, \\ 0 & ; \textit{sicer.} \end{cases}$$

Iskanje izomorfizma

Postopek za iskanje izomorfizma:

- preverimo enakost valenc grafov,
- preverimo invariante grafov,
- poiščemo primerno bijektivno preslikavo točk,
- primerjamo sosednost vseh točk z ostalimi: $u \sim v \Leftrightarrow \Phi(u) \sim \Phi(v)$.

Iskanje izomorfizma

Za iskanje izomorfizma splošnih grafov obstajata v glavnem dve metodi:

- **direktna**, tako da vzamemo dva grafa, ki ju lahko primerjamo in poskusimo najti izomorfizem.
- **preko kanonične oznake** C, ki je funkcija na množici vseh grafov, tako da sta G_1 in G_2 izomorfna če in samo če velja $C(G_1) = C(G_2)$. Na tej metodi temelji McKay's Nauty algoritem.

"A simple backtracking"algoritem

Vhodni podatki:

- naj bo R niz z linearnim redom '$<$'
- naj $inv : V \rightarrow R$ označuje invarianto točk, npr. $inv(v) = d(v)$ in $R = \mathbf{N}$
- naj bo $\Pi(V, inv) = (V_1, \ldots, V_k)$ točkovna delitev V v povezavi z inv, tj.

$$\forall v, w \in V_i : inv(v) = inv(w) \text{ in}$$
$$\forall v \in V_i, w \in V_j, i < j : inv(v) < inv(w).$$

Izhodni podatek:

- preslikava Φ, tako da je $v_i \rightarrow w_{\Phi(i)}, 1 \leq i \leq n$, izomorfizem med grafoma G_1 in G_2. Če taka preslikava ne obstaja, vrne algoritem '$NON - isomorphic$'.

"A simple backtracking"algoritem

Algorithm 26: $\textsc{Isomorph}(G_1, G_2, (V_1, \ldots, V_m), (W_1, \ldots, W_m), \phi')$

Input: Graphs $G_1 = (V = \{v_1, \ldots, v_n\}, E_1)$, $G_2 = (W = \{w_1, \ldots, w_n\}, E_2)$, vertex partitions $(V_1, \ldots, V_m), (W_1, \ldots, W_m)$ with $\bigcup V_i \subset V$, $\bigcup W_i \subset W$ and $|V_i| = |W_i|$, and an isomorphism ϕ' between the subgraphs induced by $V \setminus \bigcup V_i$ and $W \setminus \bigcup W_i$.

Output: ϕ, if ϕ' is extensible to an isomorphism ϕ between G_1 and G_2, 'NON-isomorphic' otherwise.

if $(V_1, \ldots, V_m) = \emptyset$ **then return** ϕ'

compute $V_i \in \{ V_j \mid |V_j| \leq |V_\ell|, \ 1 \leq \ell \leq m\}$

let $V_i = \{v_{i1}, v_{i2}, \ldots\}, W_i = \{w_{i1}, w_{i2}, \ldots\}$

for $j = 1, \ldots, |V_i|$ **do**

  if ϕ' extended by $i1 \to ij$ is an isomorphism between the subgraphs induced by $V \setminus \bigcup V_k \cup \{v_{i1}\}$ and $W \setminus \bigcup W_k \cup \{w_{ij}\}$ **then**

    $branch = \textsc{Isomorph}(G_1, G_2, (V_1, \ldots, V_i \setminus v_{i1}, \ldots V_m), (W_1, \ldots, W_i \setminus w_{ij}, \ldots W_m), \phi' \cup \{i1 \to ij\})$

    if $branch \neq$ *'NON-isomorphic'* **then return** $branch$

return 'NON-isomorphic'

Tabela: **A simple backtracking algoritem**

Definicija. **Kanonična oznaka grafa G**

Naj bo dan $G = (V, E)$ neusmerjen graf, kjer je $V = \{v_1, v_2, \ldots, v_n\}$ množica vozlišč. $Adj(G, \delta)$ pa je matrika sosednosti grafa G glede na razvrstitev vozlišč $v_{\delta(1)}, v_{\delta(2)}, \ldots, v_{\delta(n)}$, kjer je δ permutacija $\{1, \ldots, n\}$. Potem je C_{adj} definirana z

$$C_{adj}(G) = min_{\delta \in S_n} Adj(G, \delta)$$

Razdelitev vozlišč in iskalno drevo $\mathcal{T}$

Definicija.

Razdelitev vozlišč na particije *grafa* G *je urejen seznam* $\Pi = (V_1, V_2, \ldots, V_r)$, *kjer so* $V_i \subseteq V$ *podmnožice množice vozlišč ali celice za katere velja:*

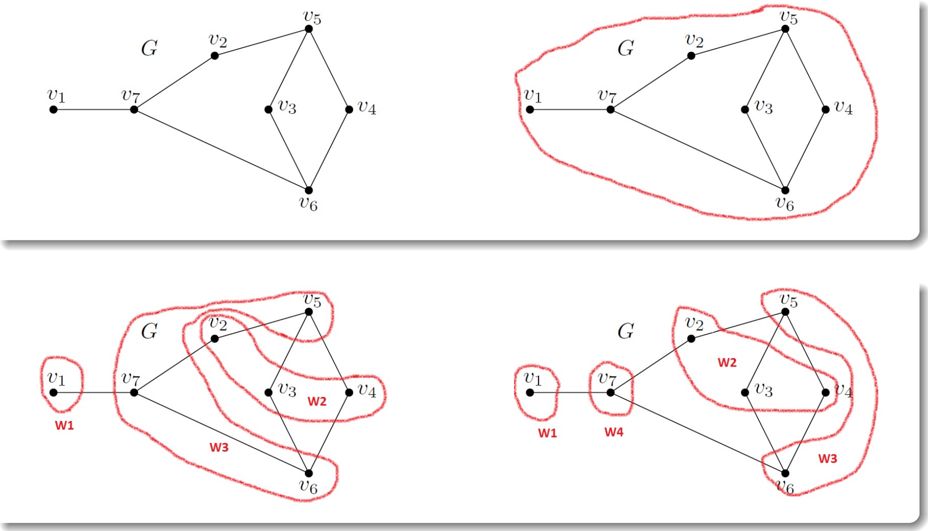
1. $V_i \cap V_j = \emptyset,\ 1 \leq i \neq j \leq r$
2. $\bigcup_{i \in \{1,\ldots,r\}} V_i = V$
3. $|V_i| \geq 1,\ 1 \leq i \leq r.$

Število celic r *dobimo kot* $|\Pi|$. *Particija vozlišč je enotna če je* $r = 1$ *in diskretna če* $r = n$.

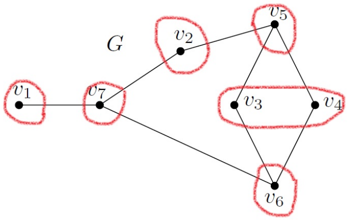
Definicija.

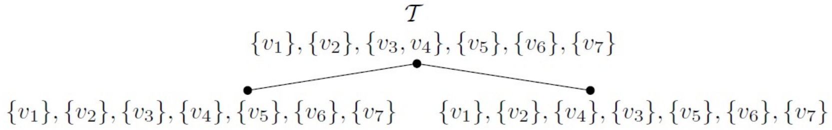
Za vozlišče $v \in V$ *in podmnožico vozlišč* $W \subset V$ *naj bo* $d(v, W)$ *število vozlišč v* W, *ki so sosedna z* v.

Zgled: Postopek refiniranja



Zgled. Postopek refiniranja



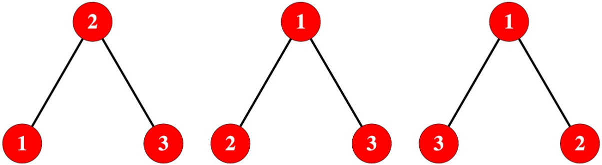


McKay-ev nauty algoritem: Rezanje drevesa $\mathcal{T}$

Izrek.

Naj bosta dana G_1 in G_2 neusmerjena grafa in naj bosta $C(G_1)$ in $C(G_2)$ oznaki grafov, ki smo ju dobili s pomočjo iskalnih dreves. Potem velja

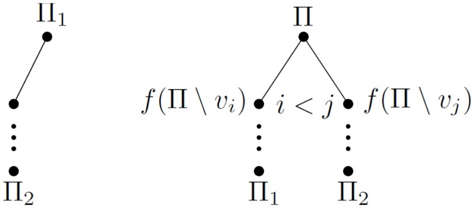
$$C(G_1) = C(G_2) \iff G_1 \simeq G_2.$$



$$\begin{matrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix} \qquad \begin{matrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{matrix}$$

Definicija. **Linearni vrstni red vozlišč iskalnega drevesa $\mathcal{T}$**

Naj bosta Π_1 in Π_2 dve različni vozlišči $\mathcal{T}$ in naj bo Π njun zadnji skupni prednik. Definiramo Π_1"$<$"Π_2, če je $\Pi_1 = \Pi$ ali pa če za vozlišči v_i in v_j, ki pripadata prvi netrivialni celici iz Π in $\Pi_1 \in \mathcal{T}(\Pi \backslash v_i)$ ter $\Pi_2 \in \mathcal{T}(\Pi \backslash v_j)$ kjer $i < j$. Drugače pa je Π_1"$>$"Π_2.



McKay-ev nauty algoritem

Definicija. **Ekvivalenčna relacija na vozliščih $\mathcal{T}$**

Naj bo dana particija $\Pi_1 = (V_1, \ldots, V_m) \in \mathcal{T}$ ter $\Pi_2 = (W_1, \ldots, W_m) \in \mathcal{T}$. Potem je $\Pi_1 \sim \Pi_2$, če in samo če obstaja avtomorfizem $\phi \in Aut(G)$ in permutacija $\delta \in S_m$ tako da velja $\phi(V_i) = W_{\delta(i)}$ za $i = 1, \ldots, m$. Potem pravimo, da ϕ določa $\Pi_1 \sim \Pi_2$.

Izrek.

Naj bo $\Pi_1 \sim \Pi_2 \in \mathcal{T}$ in naj bosta $\mathcal{T}_1$, $\mathcal{T}_2$ poddrevesa $\mathcal{T}$ s korenom v Π_1 in Π_2. Potem za vsako vozlišče $\Pi_1' \in \mathcal{T}_1$ obstaja vozlišče $\Pi_2' \in \mathcal{T}_2$ da je $\Pi_1' \sim \Pi_2'$.

McKay-ev nauty algoritem

```
process(Π = (V_1, ..., V_r))
if r = n then
    identify V_1 = {v'_1}, ..., V_n = {v'_n} with vertex order v'_1, ..., v'_n
    compute adj. matrix A_Π induced by v'_1, ..., v'_n
    if A_ℓ1 = nil then
        A_ℓ1  = A_Π, vertex_order(A_ℓ1)  = v'_1, ..., v'_n
        A_min = A_Π, vertex_order(A_min) = v'_1, ..., v'_n

    else
        if A_min > A_Π then A_min = A_Π, vertex_order(A_min) = v'_1, ..., v'_n
        else
            φ = nil
            if A_ℓ1 = A_Π then
                compute automorphism φ induced
                by vertex_order(A_ℓ1) and v'_1, ..., v'_n
            if A_min ≠ A_ℓ1 and A_min = A_Π then
                compute automorphism φ induced
                by vertex_order(A_min) and v'_1, ..., v'_n
            if φ ≠ nil then
                Φ(G) = ⟨{Φ(G) ∪ φ}⟩
                update Θ
                check jump back

else
    let V_i = {v'_1, ..., v'_m} be the first non-trivial cell of Π
    let v''_1, ..., v''_m' be the minimum cell representatives of Θ ∧ V_i
    for j = 1 to m' do process(f(Π \ v''_j))
```

Podobnost grafov

- teorija podobnosti grafov (graph matching)
- mera podobnosti - razdalja
- na kakšni razdalji je en graf podoben drugemu
- prednost: sposobnost obvladovanja z napakami na vhodnih podatkih

Definicija.

Naj bodo G_1, G_2 in G_3 grafi. Funkcija $d : G_1 \times G_2 \to R_0$ se imenuje mera razdalje med grafi, če veljajo naslednje lastnosti:

$$\textit{refleksivnost: } d(G_1, G_2) = 0 \Leftrightarrow G_1 \cong G_2 \tag{1}$$

$$\textit{simetričnost: } d(G_1, G_2) = d(G_2, G_1) \tag{2}$$

$$\textit{trikotniška neenakost: } d(G_1, G_2) + d(G_2, G_3) \geq d(G_1, G_3) \tag{3}$$

Mera razdalje

- urejevalna razdalja
- razlika v dolžini poti
- največji skupni podgraf

Urejevalna razdalja

- znan primer je urejevalna razdalja niza
- tipične operacije so vstavljanje, brisanje in nadomestitev točk in povezav
- razdalja se določi kot minimalni strošek vseh zaporedij operacij, ki preoblikujejo en graf v drugega.
- Primer: Dovoljenje so naslednje operacije:
  - dodajanje točke - grafu dodamo novo (izolirano) točko
  - izbrisanje točke - grafu izbrišemo (izolirano) točko
  - dodajanje povezave - grafu dodamo novo povezavo med poljubnimi točkami
  - izbrisanje povezave - povezava je izbrisana iz grafa

  Strošek operacij s točkami je enak 1, medtem pri operacijah s povezavami ni stroškov.

$$d_{prim1} = ||V(G_1)| - |V(G_2)||.$$

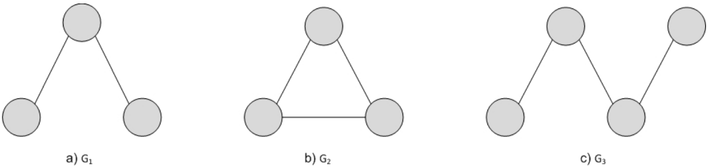
Pot, zvezda in poln graf na enakem številom točk so podobni grafi.

Urejevalna razdalja

Primer: Imamo na voljo uporabo naslednjih operacij, kjer ima vsaka strošek ena:

- dodajanje točke - grafu dodamo novo (izolirano) točko
- izbrisanje točke - grafu izbrišemo (izolirano) točko
- posodobitev povezave - ena končna točka na povezavi se spremeni

Dodajanje povezave pomeni dve operaciji posodobitve povezave.



G_1 bolj podoben G_2 kot G_3

Razlika v dolžini poti

- primer urejevalne razdalje
- povezani grafi z enakim številom točk

Definicija.

Naj bosta G_1, G_2 izomorfna povezana grafa z izomorfizmom $\phi : V(G_1) \to V(G_2)$. Dve točki grafa G_1 sta sosednji, če sta njihovi izomorfni točki v G_2 sosednji. Z drugimi besedami:

$$\forall u, v \in V(G_1) : \{u, v\} \in E(G_1) \Leftrightarrow \{\phi(u), \phi(v)\} \in E(G_2).$$

Razdalje poljudnih parov točk:

$$\forall u, v \in V(G_1) : d_{G_1}(u, v) = d_{G_2}(\phi(u), \phi(v)). \tag{4}$$

Razlika v dolžini poti

Definicija.

Za dva povezana grafa G_1, G_2 *z istim številom točk in bijekcijo* $\sigma : V(G_1) \to V(G_2)$ *definiramo* $\sigma-$***dolžino*** d_σ *z*

$$d_\sigma(G_1, G_2) = \sum_{\{u,v\} \in V(G_1) \times V(G_1)} |d_{G_1}(u, v) - d_{G_2}(\sigma(u), \sigma(v))|,$$

kjer vsota gre po vseh parih točk grafa G_1.

Definicija.

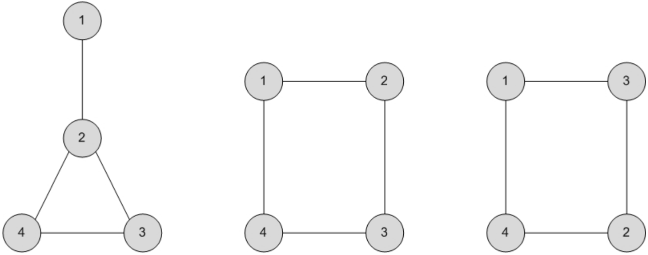
Za dva povezana grafa G_1 *in* G_2 *z istim številom točk definiramo dolžino poti* d_{pot} *kot*

$$d_{pot}(G_1, G_2) = min_{\sigma \in \Lambda} d_\sigma(G_1, G_2),$$

kjer je Λ *množica vseh bijekcij med* $V(G_1)$ *in* $V(G_2)$.

Razlika v dolžini poti

Primer:



a) G_1 b) σ_1 c) σ_2

Obstajata le dve preslikavi. Preslikavi $\sigma : V(G_1) \to V(G_2)$ sta definirani z $\sigma_i j = j$ za $j = 1, \ldots, 4$.

$$d_{\sigma 1}(G_1, G_2) = 2 \text{ in } d_{\sigma 2}(G_1, G_2) = 4.$$

$$\Rightarrow d_{\text{pot}}(G_1, G_2) = 2.$$

Razlika v dolžini poti

- poti vseh parov točk v obeh grafih - problem najkrajše poti v obeh grafih
- izračunamo σ-razdalje za dane bijekcije σ
- minimalno bijekcijo glede na dolžino poti

Največji skupni podgraf

Graf $G' = (V', E')$ je *podgraf* grafa $G = (V, E)$, če $V' \subseteq V$ in $E' \subseteq E$. *Inducirani podgraf* je takrat, ko E' vsebuje vse povezave $e \in E$, ki povezujejo točke v V'. Inducirani podgraf grafa G dobimo z odstranjevanjem točk iz $V(G)$. Ko odstranimo točko, izginejo tudi povezave, ki imajo to točko za krajišče.

Definicija.

Naj bosta G_1 in G_2 neusmerjena grafa. Injektivna funkcija $\phi : V(G_1) \rightarrow V(G_2)$ je ***podgraf izomorfizma*** *od G_1 do G_2, če obstaja inducirani podgraf $G_2' \subseteq G_2$ tak, da je ϕ izomorfizem grafov med G_1 in G_2'.*

Definicija.

Naj bosta G_1 in G_2 neusmerjena grafa. Graf S je ***skupni inducirani podgraf*** *grafa G_1 in G_2, če obstaja podgraf izomorfizma iz S do G_1 in G_2.*

Največji skupni podgraf

Definicija.

Naj bosta G_1 in G_2 neusmerjena grafa. Skupni inducirani podgraf S od G_1 in G_2 je največji, če ne obstaja noben drug skupni podgraf, ki bi imel več točk kot S. ***Največji skupni inducirani podgraf*** *označimo z $mcis(G_1, G_2)$ (maximum common induced subgraph).*

Graf $G' = (V', E')$ je *povezavno-inducirani podgraf* grafa $G = (V, E)$, če $E' \subseteq E$ in V' vsebuje točke povezav v E'. Povezavno-inducirani graf ne vsebuje izoliranih točk.

Definicija.

Naj bosta G_1 in G_2 neusmerjena grafa. Skupni povezavno-inducirani podgraf S od G_1 in G_2 je največji, če ne obstaja noben drug skupni povezavni podgraf, ki bi imel več točk kot S. ***Največji skupni povezavno-inducirani podgraf*** *označimo z $mces(G_1, G_2)$ (maximum common edge subgraph).*

Največji skupni podgraf

Definicija.

Naj bosta G_1 in G_2 neusmerjena grafa. MCIS razdaljo definiramo kot

$$d_{mcis}(G_1, G_2) = 1 - \frac{|V(mcis(G_1, G_2))|}{max(|V(G_1)|, |V(G_2)|)} \tag{5}$$

in razdaljo MCES kot

$$d_{mces}(G_1, G_2) = 1 - \frac{|V(mces(G_1, G_2))|}{max(|V(G_1)|, |V(G_2)|)}. \tag{6}$$