

# 1 ZÁKLADNÍ TERMINOLOGIE

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

## 1.1 Definice

$P_2(V)$  označuje množinu všech nejvýše dvouprvkových neprázdných podmnožin množiny  $V$ . Nechť  $V$  je neprázdna konečná množina taková, že  $P_2(V) \cap V \times V \neq \emptyset$  a nechť  $E \subseteq P_2(V) \cup V \times V$ . Pak  $(V, E)$  nazveme **prostým grafem**.

Pokud přitom  $E \subseteq P_2(V)$ , budeme mluvit o **prostém neorientovaném grafu**. Pokud  $E \subseteq V \times V$ , budeme mluvit o **prostém orientovaném grafu**.

Prvkům množiny  $V$  říkáme **uzly**, prvkům množiny  $E$  říkáme **hrany**. Přitom prvkům množiny  $E \cap P_2(V)$  říkáme **neorientované hrany** a prvkům množiny  $E \cap V \times V$  říkáme **orientované hrany**. O neorientované hraně  $\{u, v\}$  říkáme, že **spojuje** uzly  $u, v$ . O orientované hraně  $[u, v]$  říkáme, že **vychází** z uzlu  $u$  a **vchází** do uzlu  $v$ .

Neorientovaná hrana se nazývá **smyčka**, jestliže spojuje uzel sám se sebou. Orientovaná hrana se nazývá smyčka, jestliže vchází do téhož uzlu z něhož vychází.

Prostý graf bez smyček se nazývá **obyčejný**. Řekneme, že graf je **diskrétní**, jestliže  $E = \emptyset$ . Označíme  $E(u, v) := E \cap \{\{u, v\}, [u, v]\}$

$$V^+(v) = \{u \in V \mid [u, v] \in E\}$$

$$V^-(v) = \{u \in V \mid [v, u] \in E\}$$

$$V^0(v) = \{u \in V \mid \{v, u\} \in E\}$$

V některých aplikacích je zapotřebí, aby mezi dvěma uzly existovalo více hran. Hrany uvažujeme s jednou násobností. Formálně můžeme za  $E$  považovat zobrazení množiny  $P_2(V) \cup V \times V$  do množiny  $\mathbb{N}$  všech nezáporných celých čísel. Mluvíme pak o **grafu**. Hrany, které nelze rozlišit, jsou **násobné** neboli **paralelní**. Obvykle budeme říkat graf prostému grafu.

## 1.2 Definice

**Polostupněm vstupu** uzlu  $v$  v orientovaném grafu  $(V, E)$  je nezáporné celé číslo  $st^+(v)$  udávající počet hran do tohoto uzlu vcházejících, tedy

$$st^+(v) := |\{e \in E \mid (\exists v' \in V). e = [v', v]\}|$$

**Polostupněm výstupu** uzlu  $v$  v orientovaném grafu  $(V, E)$  je nezáporné celé číslo  $st^-(v)$  udávající počet hran z tohoto uzlu vycházejících, tedy

$$st^-(v) := |\{e \in E \mid (\exists v' \in V). e = [v, v']\}|$$

**Stupněm** uzlu  $v$  v neorientovaném grafu je nezáporné celé číslo  $st(v)$  udávající počet hran spojujících tento uzel s některým uzlem zvětšený o počet smyček. Tedy

$$st(v) = |\{e \in E \mid (\exists v' \in V). e = \{v, v'\}\}| + |\{e \in E \mid e = \{v, v\}\}|$$

### 1.3 Věta

1. Pro každý neorientovaný graf  $(V, E)$  platí

$$2. |E| = \sum_{v \in V} st(v)$$

2. Pro každý orientovaný graf  $(V, E)$  platí

$$|E| = \sum_{v \in V} st^+(v) = \sum_{v \in V} st^-(v)$$

**Důkaz:** Zřejmý.

### 1.4 Definice

Jsou-li hranám grafu přiřazeny určité hodnoty, mluvíme o **hranově ohodnoceném** grafu. Jsou-li uzlům grafu přiřazeny určité hodnoty, mluvíme o **uzlově ohodnoceném** grafu. Číselně ohodnocený graf je **síť**.

### 1.5 Definice

**Částečný graf** grafu  $(V, E)$  je každý graf  $(V, E')$  takový, že  $E' \subseteq E$ .

**Podgraf** grafu  $(V, E)$  je každý graf  $(V', E')$  takový, že  $V' \subseteq V \neq \emptyset$  a  $E' = E \cap (P_2(V') \cup V' \times V')$ .

**Částečný podgraf** grafu je každý částečný graf jeho podgrafu.

Bijekce  $f : V_1 \rightarrow V_2$  je **izomorfismus grafů**  $(V_1, E_1)$  a  $(V_2, E_2)$ , jestliže pro každé uzly  $u, v \in V_1$  platí

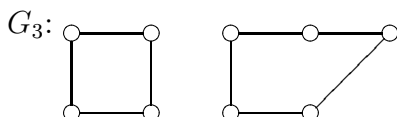
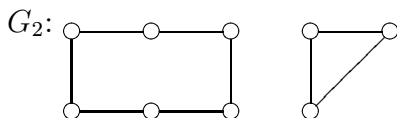
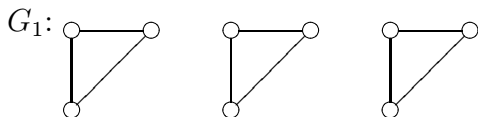
$$\{u, v\} \in E_1 \Leftrightarrow \{f(u), f(v)\} \in E_2$$

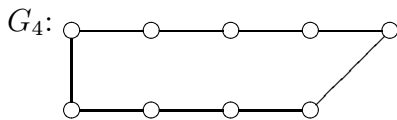
$$[u, v] \in E_1 \Leftrightarrow [f(u), f(v)] \in E_2$$

### 1.6 Definice

**Skóre** obyčejného neorientovaného grafu je neklesající posloupnost stupňů všech jeho uzlů. Skóre obyčejného neorientovaného grafu je určeno jednoznačně, ale navzájem neizomorfní grafy mohou mít totéž skóre.

*Příklad:* Existují čtyři navzájem neizomorfní obyčejné neorientované grafy, které mají skóre  $[2, 2, 2, 2, 2, 2, 2, 2, 2]$ .





## 1.7 Věta

Nechť  $[k_1, \dots, k_n]$  je skóre obyčejného neorientovaného grafu. Pak existuje takový obyčejný neorientovaný graf  $(V, E)$ , že  $V = \{v_1, \dots, v_n\}$ ,  $st(v_i) = k_i$  pro každé  $i$  a  $\{v_i, v_n\} \in E \Leftrightarrow n - k_n \leq i \leq n - 1$  pro všechna  $i$ .

**Důkaz:** Mezi všemi obyčejnými neorientovanými grafy s množinou uzlů  $\{v_1, \dots, v_n\}$  v nichž  $st(v_i) = k_i$  pro každé  $i$  vybereme takový, pro nějž je maximální hodnota výrazu

$$\sum_{v_k \in V^0(v_n)} k$$

Podle předpokladů alespoň jeden takový graf existuje. Pripusťme, že  $v_i \in V^0(v_n)$ ,  $v_j \notin V^0(v_n)$ , kde  $i < j < n$ . Pak také  $v_n \in V^0(v_i)$  a  $v_n \notin V^0(v_j)$ ,  $k_i \leq k_j$ . Odtud  $V^0(v_j) \setminus V^0(v_i) \neq \emptyset$ . Zvolme  $v \in V^0(v_j) \setminus V^0(v_i)$ .

Odstraněním hran  $\{v_i, v_n\}, \{v_j, v\}$  a přidáním hran  $\{v_i, v\}, \{v_j, v_n\}$  dostaneme obyčejný neorientovaný graf s množinou uzlů  $\{v_1, \dots, v_n\}$ , v němž  $st(v_i) = k_i$  pro všechna  $i$  s hodnotou  $\sum_{v_k \in V^0(v_n)} k$  větší o  $j - i$ . To je spor s předpokladem. □

## 1.8 Věta

Nechť  $[k_1, \dots, k_n]$  je neklesající posloupnost přirozených čísel. Pak  $[k_1, \dots, k_n]$  je skóre obyčejného neorientovaného grafu právě tehdy, když neklesající posloupnost, kterou obdržíme přerovnáním posloupnosti  $[k_1, \dots, k_{n-k_n-1}, k_{n-k_n} - 1, \dots, k_{n-1} - 1]$ , je skóre obyčejného neorientovaného grafu.

**Důkaz:**

( $\Rightarrow$ ) Takto to plyne z předchozí věty. Stačí odebrat  $v_n$  spolu s incidentními hranami z grafu  $(V, E)$ .

( $\Leftarrow$ ) Dostaneme přidáním uzlu  $k_n$  a jeho spojením s  $k_n$  posledními uzly, tj. uzly stupňů  $k_{n-k_n} - 1, \dots, k_n - 1$ . □

**Poznámka:** Posloupnost  $[k_1, \dots, k_{n-k_n-1}, k_{n-k_n} - 1, \dots, k_{n-1} - 1]$  z věty 1.8 budeme značit  $S'$ .

Obecně, neklesající posloupnost, kterou obdržíme přerovnáním posloupnosti  $P$ , označíme  $P^<$ .

## 1.9 Algoritmus

**Vstupní data**

Konečná neklesající posloupnost přirozených čísel  $S$ .

**Úkol**

Zjistit, zda  $S$  je skóre obyčejného neorientovaného grafu. Nakreslit diagram tohoto grafu.

*1.krok*

Rekurzivně vytvoříme posloupnosti

$$S_0 := S$$

$$S_{i+1} := (S_i')^<$$

dokud je to možné.

### 2.krok

Pokud skončíme posloupností nul, pak  $S$  je skóre. Jinak nikoliv a algoritmus končí.

### 3.krok

Vezmeme diskrétní graf  $G_{max}$  s tolika uzly, kolik je nul v poslední posloupnosti  $S_{max} = S'_{max-1}$ .

### 4.krok

Rekurzivně vytváříme  $G_{i-1}$  z  $G_i$  přidáním uzlu stupně rovného poslednímu členu v  $S_{i-1}$  spojeného s uzly stupňů uvedených na konci posloupnosti  $S'_{i-1}$ .

Konec algoritmu.

### Příklad:

$$S = [3, 3, 3, 3, 3, 5, 5]$$

$$S_0 = [3, 3, 3, 3, 3, 5, 5]; S'_0 = [3, 2, 2, 2, 2, 4]$$

$$S_1 = [2, 2, 2, 2, 3, 4]; S'_1 = [2, 1, 1, 1, 2]$$

$$S_2 = [1, 1, 1, 2, 2]; S'_2 = [1, 1, 0, 1]$$

$$S_3 = [0, 1, 1, 1]; S'_3 = [0, 1, 0]$$

$$S_4 = [0, 0, 1];$$

Tedy  $S$  není skóre.

### Příklad:

$$S = [2, 3, 3, 3, 3, 3, 5]$$

$$S_0 = [2, 3, 3, 3, 3, 3, 5]; S'_0 = [2, 2, 2, 2, 2, 2]$$

$$S_1 = [2, 2, 2, 2, 2, 2]; S'_1 = [2, 2, 2, 1, 1]$$

$$S_2 = [1, 1, 2, 2, 2]; S'_2 = [1, 1, 1, 1]$$

$$S_3 = [1, 1, 1, 1]; S'_3 = [1, 1, 0]$$

$$S_4 = [0, 1, 1]; S'_4 = [0, 0]$$

$$S_5 = [0, 0];$$

Tedy  $S$  je skóre.

$$G_5 = (\{a, b\}, \emptyset)$$

$$G_4 = (\{a, b, c\}, \{\{a, b\}\})$$

$$G_3 = (\{a, b, c, d\}, \{\{a, b\}, \{c, d\}\})$$

$$G_2 = (\{a, b, c, d, e\}, \{\{a, b\}, \{c, d\}, \{c, e\}, \{d, e\}\})$$

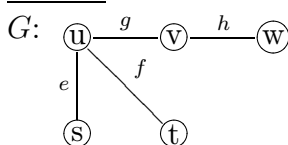
$$G_1 = (\{a, b, c, d, e, f\}, \{\{a, b\}, \{c, d\}, \{c, e\}, \{d, e\}, \{a, f\}, \{b, f\}\})$$

$$G_0 = (\{a, b, c, d, e, f, g\}, \{\{a, b\}, \{c, d\}, \{c, e\}, \{d, e\}, \{a, f\}, \{b, f\}, \{a, g\}, \{f, g\}, \{b, g\}, \{c, g\}, \{d, g\}\})$$

Neorientovaný graf můžeme reprezentovat **incidenční maticí**. Řádky označují uzly a sloupce hrany v pevně zvoleném pořadí. Pokud daná hrana spojuje daný uzel s některým uzlem, je příslušný prvek matice roven 1, jinak 0.

Orientovaný graf můžeme reprezentovat **maticí sousednosti**. Jak řádky, tak sloupce označují uzly v témž pevně zvoleném pořadí. Pokud z uzlu označujícího řádek vychází hrana vcházející do uzlu označujícího sloupec, pak je příslušný prvek matice roven 1, jinak 0. Maticí sousednosti můžeme samozřejmě reprezentovat i neorientované grafy – daná matice je pak symetrická.

### Příklad:



*Incidenční matice grafu  $G$ :*

$G$	$e$	$f$	$g$	$h$
$s$	1	0	0	0
$t$	0	1	0	0
$u$	1	1	1	0
$v$	0	0	1	1
$w$	0	0	0	1

*Matice sousednosti grafu  $G$ :*

$G$	$s$	$t$	$u$	$v$	$w$
$s$	0	0	1	0	0
$t$	0	0	1	0	0
$u$	1	1	0	1	0
$v$	0	0	1	0	1
$w$	0	0	0	1	0

## 2 SLEDY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

Poznámka:

$$\mathbb{N}_0 = \{0, 1, \dots\}$$

Pro každé  $n \in \mathbb{N}_0$  platí  $n = \{0, \dots, n-1\}$ .

### 2.1 Definice

Nechť  $n \in \mathbb{N}_0$ . **Sledem** délky  $n$  v grafu  $(V, E)$  nazveme takovou dvojici posloupností  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$ , že pro každé  $i \in n+1$  je  $v_i \in V$  a pro každé  $i \in n$  je  $e_i \in E(v_i, v_{i+1})$ . Jeho **začátkem** je uzel  $v_0$  a **koncem** uzel  $v_n$ .

Sled  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  je **uzavřený**, jestliže  $n \geq 1$  a  $v_0 = v_n$ .

Sled  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  je **tah**, jestliže  $i \neq j \Rightarrow e_i \neq e_j$ .

Sled  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  je **cesta**, jestliže  $i \neq j \Rightarrow v_i \neq v_j$ .

Uzavřený tah  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  je **kružnice**, jestliže  $(i \neq 0 \ \& \ i \neq n \ \& \ i \neq j) \Rightarrow v_i \neq v_j$ .

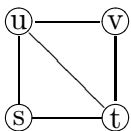
Graf  $(V, E)$  je **souvislý**, jestliže pro každé jeho uzly  $u, v$  existuje pro nějaké  $n \in \mathbb{N}_0$  sled  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  takový, že  $v_0 = u$  a  $v_n = v$ .

**Souvislá komponenta** grafu je každý jeho maximální souvislý podgraf.

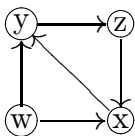
**Graf bez kružnic** je graf, ve kterém se nevyskytuje kružnice.

Příklad:

neorientovaný graf  $G_1$ :



orientovaný graf  $G_2$ :



příklad *uzavřených sledů*, které jsou *tahy*:

$$s - t - u - v - t - s$$

$$x \rightarrow y \rightarrow z \rightarrow x \rightarrow y \rightarrow z \rightarrow x$$

příklady *tahů*, které *nejsou uzavřené*:

$$s - t - u - v - t$$

$$w \rightarrow x \rightarrow y \rightarrow z \rightarrow x$$

příklady *kružnic*:

$$t - u - v - t$$


$$x \rightarrow y \rightarrow z \rightarrow x$$

Je zřejmé, že každý uzel grafu a každá hrana neorientovaného grafu jsou obsaženy právě v jedné souvislé komponentě.

Všimněme si, že kkružnice má začátek a směr.  
Každá smyčka určuje kružnici.

## 2.2 Definice

Pokud definici 2.1 v případě orientovaného grafu pozměníme tak, že budeme požadovat  $e_i \in E(v_i, v_{i+1}) \cup E(v_{i+1}, v_i)$ , budeme mluvit o **polosledu**, **polotahu**, **polocestě**, **polokružnici** a **polosouvislém grafu**.

*Příklad polosledu:* 

## 2.3 Definice

Řekneme, že sled  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  **obsahuje** sled  $[[f_i]_{i \in m}, [u_i]_{i \in m+1}]$ , jestliže existuje prosté izotonní zobrazení  $h : m \rightarrow n$  takové, že

$$(\forall i \in m)(f_i = e_{h(i)} \ \& \ u_i = v_{h(i)} \ \& \ u_{i+1} = v_{h(i)+1})$$

## 2.4 Věta

Každý sled obsahuje cestu s tímž začátkem a tímž koncem, jako má on sám.

**Důkaz:** Necht  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  je nějaký sled a  $[[f_i]_{i \in m}, [u_i]_{i \in m+1}]$  je sled s tímž začátkem  $u_0 = v_0$  a tímž koncem  $u_m = v_m$  v něm obsažený, který má nejmenší možný počet hran. Kdyby pro nějaké indexy  $j < k$  platilo  $v_j = u_k$ , pak sled

$$[[g_i]_{i \in m-k+j}, [w_i]_{i \in m-k+j+1}]$$

definovaný vztahy

$$g_i := \begin{cases} f_i, & i < j \\ f_{i+k-j}, & j \leq i \end{cases}$$

$$w_i := \begin{cases} u_i, & i < j \\ u_{i+k-j}, & j \leq i \end{cases}$$

by měl rovněž týž začátek a týž konec, přitom by však byl kratší. □

## 2.5 Důsledek

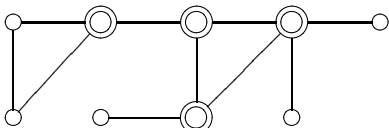
Graf je souvislý právě tehdy, když v něm pro každé uzly  $u, v$  existuje cesta z  $u$  do  $v$ .

## 2.6 Definice

Uzel v neorientovaném grafu je **artikulace**, jestliže jeho vypuštěním včetně s ním incidentních hran vzroste počet souvislých komponent grafu.

Hrana v neorientovaném grafu je **most**, jestliže jejím vypuštěním vzroste počet souvislých komponent grafu.

*Příklad:* v následujícím grafu jsou zvýrazněny artikulace



## 2.7 Věta

- (1) Uzel  $u$  je artikulace právě tehdy, když existují uzly  $v, w$  od něj různé takové, že náleží do téže souvislé komponenty a každá cesta z  $v$  do  $w$  obsahuje  $u$ .
- (2) Hrana  $\{u, v\}$  je most právě tehdy, když ji obsahuje každá cesta z  $u$  do  $v$ . Jestliže je hrana mostem, jejím odstraněním vzroste počet komponent o jednu.
- (3) Hrana je most právě tehdy, když není obsažena v žádné kružnici.



### 3 EULEROVSKÉ A HAMILTONOVSKÉ GRAFY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

#### 3.1 Definice

Graf lze sestrotit **jedním tahem**, jestliže v něm existuje tah obsahující všechny hrany grafu.

Obyčejný neorientovaný graf bez uzlů stupně 0, jehož každý uzel je sudého stupně, se nazývá **eulerovský**.

#### 3.2 Věta

Každý uzel eulerovského grafu je obsažen v alespoň jedné kružnici.

**Důkaz:** Necht  $u$  je libovolný uzel v eulerovském grafu  $(V, E)$ . Podle definice není stupně 0, existuje tedy uzel  $v$  takový, že  $\{u, v\} \in E$ . Kdyby tato hrana byla mostem v grafu  $(V, E)$ , jejím odstraněním ze souvislé komponenty, v níž uzly  $u$  a  $v$  leží, bychom obdrželi graf se dvěma komponentami. Každá z nich by obsahovala právě jeden uzel lichého stupně, totiž  $u$  a  $v$ . To podle věty 1.3 není možné. Podle věty 2.7 leží  $\{u, v\}$  na kružnici, a tedy i uzel  $u$  leží na kružnici. □

#### 3.3 Lemma

Pokud lze obyčejný neorientovaný graf sestrotit jedním uzavřeným tahem  $\tau$  a  $\{u, v\}$  je hrana, pak lze graf sestrotit i uzavřeným tahem začínajícím  $u - v - \dots$

**Důkaz:** Necht  $\tau$  je roven  $[[e_i]_{i \in k}, [u_i]_{i \in k+1}]$ . Pokud  $\{u, v\} = e_j$ , pak buď  $u = u_j$  a  $v = u_{j+1}$  nebo  $v = u_j$  a  $u = u_{j+1}$ .

V prvním případě položíme  $\tau'$  rovno

$$[[e_j, e_{j+1}, \dots, e_{k-1} = e_0, \dots, e_{j-1}], [u, v = u_{j+1}, \dots, u_k = u_0, \dots, u_j = u]],$$

ve druhém případě

$$[[e_j, e_{j-1}, \dots, e_0 = e_{k-1}, \dots, e_{j+1}], [u, v = u_j, \dots, u_0 = u_k, \dots, u_{j+1} = u]].$$

□

#### 3.4 Věta

Souvislý obyčejný neorientovaný graf lze sestrotit jedním uzavřeným tahem právě tehdy, když je eulerovský.

**Důkaz:**

( $\Rightarrow$ ) Jestliže lze neorientovaný graf sestrotit jedním tahem, je souvislý, průběžné sledy spojující dvojice uzlů jsou totiž podtahy nebo tahy k nim obrácené. V uzavřeném tahu z uzlu tolikrát vycházíme, kolikrát jsme do něj vstoupili. Proto je stupeň každého uzlu sudý. Uzavřený tah má alespoň jednu hranu.

( $\Leftarrow$ ) Necht  $(V, E)$  je souvislý eulerovský graf. Podle věty 3.2 obsahuje kružnici a ta je uzavřeným tahem. V grafu  $(V, E)$  tedy existuje uzavřený tah maximální délky, označme jej  $\tau$ . Položme  $E^* = \{\{u, v\} \mid \{u, v\} \text{ nepatří do tahu } \tau\}$  a  $V^* = \{u \in V \mid (\exists v \in V)\{u, v\} \in E^*\}$ .

Pokud  $E^* = \emptyset$ , lze graf sestrojít tahem  $\tau$ . Pripustíme, že  $E^* \neq \emptyset$ . Pak je  $(V^*, E^*)$  zřejmě podgraf grafu  $(V, E)$  a přitom je eulerovský. Protože je  $(V, E)$  souvislý, existuje uzel  $u \in V^*$  obsažený v tahu  $\tau$ . Podle věty 3.2 leží uzel  $u$  na kružnici v  $(V^*, E^*)$  – označme ji  $\gamma$ . Podle lemmatu 3.3 můžeme bez újmy na obecnosti předpokládat, že jak  $\tau$ , tak  $\gamma$  začínají v  $u$ . Spojením  $\tau$  a  $\gamma$  obdržíme uzavřený tah větší délky, než je délka  $\tau$ . To je spor. □

### 3.5 Věta

Souvislý obyčejný neorientovaný graf lze sestrojít jedním otevřeným tahem s alespoň jednou hranou právě tehdy, když obsahuje právě dva uzly lichého stupně.

**Důkaz:**

- ( $\Rightarrow$ ) Jestliže lze graf sestrojít jedním otevřeným tahem s alespoň jednou hranou začínající v uzlu  $u$  a končící v uzlu  $v$ , pak jsou uzly  $u$  a  $v$  lichého stupně a ostatní uzly sudého stupně z podobného důvodu jako v důkazu věty 3.4.
- ( $\Leftarrow$ ) Nechť  $(V, E)$  obsahuje právě dva uzly lichého stupně  $u$  a  $v$ . Doplňme uzel  $z \notin V$  a hrany  $\{u, z\}, \{v, z\}$ . Obdržíme souvislý eulerovský graf  $(V^*, E^*)$ . Podle věty 3.4 jej lze sestrojít jedním uzavřeným tahem, který podle 3.3 začíná  $z - u - \dots$  a končí  $\dots - v - z$ . Vypuštěním první a poslední hrany a prvního a posledního uzlu tahu dostaneme tah, jímž sestrojíme graf  $(V, E)$ . Má alespoň jednu hranu a končí v uzlu  $v$ , který je různý od uzlu, v němž začíná, tedy  $u$ . □

### 3.6 Definice

Obyčejný neorientovaný graf se nazývá **hamiltonovský**, jestliže v něm existuje kružnice obsahující všechny uzly. Taková kružnice se nazývá **hamiltonovská**.

### 3.7 Věta (Diracova-Oreova)

Každý obyčejný neorientovaný graf, v němž platí

$$3 \leq |V| \ \& \ (\forall u, v \in V) (\{u, v\} \notin E \Rightarrow |V| \leq st(u) + st(v)),$$

je hamiltonovský.

**Důkaz:** Pripustíme, že graf  $(V, E)$  splňující výše uvedenou podmínku není hamiltonovský. Poněvadž každý graf splňující  $3 \leq |V|$ , v němž jsou každé dva uzly spojeny hranou, je hamiltonovský. Postupným přidáváním hran ke grafu  $(V, E)$  dostaneme graf  $(V^*, E^*)$ , který není hamiltonovský, ale přidáním další hrany  $\{u, v\}$  už dostaneme hamiltonovský graf. Bez újmy na obecnosti můžeme předpokládat, že  $(V^*, E^*) = (V, E)$  a  $u - v - u_2 - \dots - u_{n-1} - u$  je hamiltonovská kružnice v grafu  $(V, E \cup \{u, v\})$ . Pokud je  $u$  spojeno hranou ve  $(V, E)$  s některým uzlem  $u_i$ , kde  $i \neq n - 1$ , pak  $v$  není spojeno hranou s  $u_{i+1}$ , jinak by  $u - u_i - u_{i+1} - \dots - u_2 - v - u_{i+1} - \dots - u_{n-1} - u$  byla hamiltonovská kružnice ve  $(V, E)$ . Takových uzlů je  $st(u) - 1$ , neboť  $u$  je kromě nich spojen hranou s  $u_{n-1}$ . Odtud však plyne  $|V| + 1 \leq st(u) + st(v) + 1 = (st(u) - 1) + st(v) + 2 = |V \setminus \{u, v\} \setminus V^0(v)| + |V^0(v)| + |\{u, v\}| = |v|$ , což je ovšem spor. □

## 4 STROMY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

### 4.1 Definice

Obyčejný neorientovaný graf bez kružnic je **les**.

Souvislý obyčejný neorientovaný graf bez kružnic je **strom**.

*Poznámky:*

Protože každá smyčka určuje kružnici, je slovo *obyčejný* v definici zbytečné. Souvislé komponenty v lese jsou stromy.

### 4.2 Věta

Pro souvislý obyčejný orientovaný graf  $(V, E)$  jsou tato tvrzení ekvivalentní:

- (i)  $(V, E)$  je strom.
- (ii) Každé dva uzly ve  $(V, E)$  jsou spojeny právě jednou cestou.

**Důkaz:**

( $\Rightarrow$ ) Nechť  $u, v \in V$ . Protože  $(V, E)$  je souvislý, alespoň jedna cesta z  $u$  do  $v$  existuje. Pripusťme, že  $[[e_i]_{i \in n}, [v_i]_{i \in n+1}]$  a  $[[f_i]_{i \in n}, [u_i]_{i \in n+1}]$  jsou různé cesty z  $u$  do  $v$ . Nechť  $l$  je poslední index takový, že  $[[e_i]_{i \in l}, [v_i]_{i \in l+1}] = [[f_i]_{i \in l}, [u_i]_{i \in l+1}]$ . Označme  $j := \min\{i > l \mid (\exists k)v_i = v_k\}$ . Pak

$$[[e_l, e_{l+1}, \dots, e_{j-1}, f_{k-1}, f_{k-2}, \dots, f_l], [v_l, v_{l+1}, \dots, v_j = u_k, u_{k-1}, \dots, u_l]]$$

je kružnice, čímž jsme dospěli ke sporu.

( $\Leftarrow$ ) Pokud by to nebyl strom, obsahoval by kružnici a v ní jsou každé dva uzly spojeny dvěma cestami. □

### 4.3 Věta

Vypuštěním jedné hrany  $\{u, v\}$  ze stromu dostaneme les se dvěma komponentami  $C_u$  a  $C_v$ , pro něž platí  $u \in C_u, v \in C_v$ .

**Důkaz:** Jde o důsledek věty 2.7. Pokud by byl výsledný graf souvislý, existovala by v původním grafu kružnice. □

### 4.4 Věta

Každý strom s alespoň jednou hranou obsahuje alespoň dva uzly stupně 1.

**Důkaz:** Nechť  $u, v$  jsou uzly spojené cestou  $u = v_0 - v_1 - \dots - v_{k-1} - v_k = v$ , která má maximální délku v daném grafu. Kdyby uzel  $v$  nebyl stupně 1, existovala by hrana  $\{v, v'\}$ , kde  $v' \neq v_{k-1}$ . Kdyby  $v' = v_i$ , existovala by kružnice  $v_i - \dots - v_{k-1} - v_k - v_i$ , což je spor. Pokud  $v' \neq v_i$ , pro každé  $i$ , pak by  $u = v_0 - v_1 - \dots - v_{k-1} - v_k = v - v'$  byla delší cesta což je také spor. Tedy  $st(v) = 1$  a podobně  $st(u) = 1$ . □

## 4.5 Věta

Nechť  $(V, S)$  je částěčný graf souvislého neorientovaného grafu  $(V, E)$ . Jestliže  $(V, S)$  je les, pak existuje množina hran  $T$  taková, že  $S \subseteq T \subseteq E$  a  $(V, T)$  je strom.

**Důkaz:** Nechť  $(V, T)$  je maximální graf bez kružnic obsažený ve  $(V, E)$  a obsahující  $(V, S)$ . Pokud by nebyl souvislý, existovaly by uzly  $u$  a  $v$  takové, že ve  $(V, T)$  nejsou spojeny cestou. Leží tedy ve dvou různých souvislých komponentách grafu  $(V, T)$ . Ve  $(V, E)$  jsou spojeny cestou. Ta obsahuje hranu, která má se souvislou komponentou obsahující uzel  $u$  společný právě jeden uzel. Jejím přidáním k  $T$  bychom obdrželi opět graf bez kružnic, což je spor. □

## 4.6 Věta

Pro souvislý neorientovaný obyčejný graf  $(V, E)$  jsou ekvivalentní tato tvrzení:

- (i)  $(V, E)$  je strom
- (ii)  $|V| - 1 = |E|$

**Důkaz:** Indukcí

( $\Rightarrow$ ) *Základní krok:* Pro  $|E| = 0$  a  $|E| = 1$  tvrzení platí.

*Indukční krok:* Předpokládejme, že tvrzení platí pro  $|E| = 1, \dots, p - 1$ . Dokážeme je pro  $|E| = p$ . Nechť  $(V, E)$  je strom s  $p$  hranami. Podle věty 4.4 v něm existuje uzel stupně 1. Vypuštěme tento uzel s přílehlou hranou. Dostaneme strom s  $p - 1$  hranami. Podle indukčního předpokladu platí  $(|V| - 1) - 1 = |E| - 1$ . Odtud plyne  $|V| - 1 = |E|$ .

( $\Leftarrow$ ) Nechť  $(V, E)$  není strom. Podle věty 4.5 obsahuje strom  $(V, T)$ . Pro něj platí podle předcházejícího  $|E| > |T| = |V| - 1$ . □

Je zřejmé, že každý strom s alespoň třemi uzly obsahuje uzel stupně 2. Jinak by platilo

$$2 = |V| - |E| + 1 \geq \sum_{v \in V} st(v) - |E| + 1 = 2 \cdot |E| - |E| + 1 = |E| + 1 = |V|$$

Příklad: Kolik hran má les  $k$  stromů s  $n$  uzly?

$$|E| = |V| - k = n - k$$

## 4.7 Věta

Pro obyčejný neorientovaný graf jsou ekvivalentní tato tvrzení:

- (i) je to strom
- (ii) je souvislý, ale odebráním jakékoliv hrany dostaneme nesouvislý graf

**Důkaz:**

$\Rightarrow$  Sporem. Nechť uzly  $u, v$  ve stromě  $(V, T)$  jsou spojeny hranou. Předpokládejme, že po jejím odebrání dostaneme souvislý graf. Pak však obsahuje cestu z  $u$  do  $v$ . Jejím doplněním o hranu  $\{u, v\}$  dostáváme kružnici. Spor.

$\Leftarrow$  Nechť graf  $(V, E)$  je souvislý a odebráním kterékoliv hrany dostaneme nesouvislý graf. Kdyby to nebyl strom, obsahoval by kružnici a odebráním kterékoliv její hrany dostaneme souvislý graf, což je spor s předpokladem. □

## 4.8 Definice

**Kořenový strom** je struktura  $(V, E, v)$ , kde  $(V, E)$  je strom a  $v \in V$ . Uzel  $v$  se nazývá **kořen**. Uzel  $x$  se nazývá (**přímý**) **předchůdce** uzlu  $y$  a uzel  $y$  (**přímý**) **následník** uzlu  $x$ , jestliže (jsou spojeny hranou a)  $x$  leží na cestě spojující  $v$  a  $y$  a přitom  $x \neq y$ . Uzly, které nemají následníky se nazývají **koncové uzly** neboli **listy**.

**Uspořádaný kořenový strom** je struktura  $(V, E, v, \leq)$ , kde  $(V, E, v)$  je kořenový strom a  $\leq$  je uspořádání množiny  $V \setminus \{v\}$ , které je sjednocením lineárních uspořádání množin jednotlivých přímých následníků uzlů.

**Binární strom** je struktura  $(V, E, v, f)$ , kde  $(V, E, v)$  je kořenový strom a  $f$  je zobrazení  $V \setminus \{v\}$  do množiny  $\{\text{pravý}, \text{levý}\}$ , jehož restrikce na množinu přímých následníků každého uzlu je prostá.

## 4.9 Definice

**Hloubka uzlu** v kořenovém stromu  $(V, E, v)$  je délka cesty spojující tento uzel s kořenem. **Výška kořenového stromu** je délka nejdelší cesty vedoucí z kořene.

## 4.10 Definice

**Centrum**, případně **bicentrum**, stromu  $(V, E)$  je ta neprázdná podmnožina množiny  $V$ , která zůstane po opakovaném současném odebírání všech uzlů stupně 1.

## 4.11 Věta

Izomorfismus převádí centrum, případně bicentrum, stromu na centrum, případně bicentrum.

**Důkaz:** Zřejmě konstrukce vedoucí k centru, případně bicentru, nezávisí na jménech jednotlivých uzlů. □

Každému stromu můžeme přiřadit kořenový strom tak, že pokud má centrum, prohlásíme je za kořen, a pokud má bicentrum, doplníme kořen rozdělením hrany v bicentru.

## 4.12 Věta

Stromy jsou izomorfní právě tehdy, když mají oba centrum nebo oba bicentrum a příslušné kořenové stromy jsou izomorfní.

**Důkaz:**

( $\Rightarrow$ ) Zřejmě izomorfismus je totožný s původním izomorfismem, kořen se převede na kořen, pokud jde o bicentrum

( $\Leftarrow$ ) Protože se převádí kořen na kořen, je izomorfismus kořenových stromů izomorfismem stromů □

Uvedeme si algoritmus, který zjistí, zda dané dva uzlově ohodnocené kořenové stromy jsou izomorfní, tj. zda existuje izomorfismus, který převede kořen na kořen a zachovává ohodnocení.

### 4.13 Algoritmus

*Vstupní data:*

Navzájem disjunktní uzlově ohodnocené kořenové stromy  $T_1 := (V_1, E_1, u_1)$  a  $T_2 := (V_2, E_2, u_2)$ .

*Úkol:*

Zjistit, zda jsou izomorfní.

*Označení:*

$a(v)$ ... ohodnocení uzlů

$c(v)$ ... číslo

$l(v)$ ... seznam čísel

*1.krok:*

Pro oba stromy určíme počet uzlu  $n$ , výšku  $h$  a počet uzlů v každé vrstvě, tj. s touž hloubkou  $j$ . Pokud nejsou hodnoty shodné, stromy nejsou izomorfní, *ZASTAV*.

*2.krok:*

Pro  $j = h, h - 1, \dots, 0$  provádíme

- Pro každý uzel  $v$   $j$ -té vrstvy vytvoříme seznam  $l(v)$  tak, že za ohodnocení  $a(v)$  přidáme posloupnost  $c(v_1), \dots, c(v_k)$ , kde  $v_1, \dots, v_k$  jsou přímí následníci uzlu  $v$  uspořádání tak, aby tato posloupnost byla neklesající.
- Pro každý strom  $T_i$  uspořádáme uzly  $j$ -té vrstvy lexikograficky podle hodnot seznamu  $l(v)$ . Výslednou posloupnost označíme  $v_{i,1}, \dots, v_{i,n_j}$
- Jestliže pro nějaké  $p \in \{1, \dots, n_j\}$  se seznam  $l(j)$  liší od seznamu  $l(v_{2,p})$ , pak stromy nejsou izomorfní, *ZASTAV*.
- Pro uzly  $j$ -té vrstvy určíme  $c$  takto:

$$c(v_{i,1}) := 1$$

$$\text{Pro } l(v_{i,p}) = l(v_{1,p-1}) \text{ položíme } c(v_{i,p}) = c(v_{1,p-1})$$

$$\text{Pro } l(v_{i,p}) \neq l(v_{1,p-1}) \text{ položíme } c(v_{i,p}) = c(v_{1,p-1}) + 1$$

*3.krok:*

Jestliže se algoritmus nezastavil, stromy jsou izomorfní.

### 4.14 Věta

Algoritmus 4.13 funguje správně.

**Důkaz:** Rovnost počtu uzlů ve stromech a uzlů v jednotlivých vrstvách stromů je nutným předpokladem existence izomorfismu. Indukcí dokážeme, že uzly  $v, w$  ležící ve sjednocení vrstev obou stromů získají tutéž hodnotu čísla  $c$  právě tehdy, když jimi určené kořenové podstromy  $T(v)$  a  $T(w)$  jsou izomorfní se zachováním ohodnocení.

*Základní krok:*

Pro poslední vrstvu tvrzení platí. Každý takový podstrom je tvořen jediným uzlem a přitom  $c(v) = c(w) \Leftrightarrow a(v) = a(w)$ .

*Indukční krok:*

Předpokládejme, že tvrzení platí pro všechny vrstvy s indexem vyšším než  $j$ . Rovnost  $c(v) = c(w)$  nastane právě tehdy, když  $l(v) = l(w)$ . To však znamená, že  $a(v) = a(w)$  a  $v$  a  $w$  mají stejný počet přímých následníků s touž hodnotou čísla  $c$ . Podle indukčního předpokladu jsou podstromy určené odpovídajícími si dvojicemi přímých následníků s touž hodnotou čísla  $c$  izomorfní. Sjednocení příslušných izomorfismů podstromu sjednocené se zobrazením  $\{v \rightarrow w\}$  je izomorfismus. □

Chceme zjistit, kolik stromů existuje na dané množině uzlů. Zavedeme pomocné označení. Množinu všech stromů s množinou uzlů  $\{v_1, \dots, v_n\}$  označíme  $T(v_1, \dots, v_n)$ .

Pro  $T \in T(v_1, \dots, v_n)$  a konečnou posloupnost  $s = [k_1, \dots, k_{n-2}] \in \{1, \dots, n\}^{n-2}$  položíme

$$T \approx S \quad :\Leftrightarrow \quad \text{existuje } [i_1, \dots, i_{n-2}] \in \{1, \dots, n\}^{n-2} \text{ taková, že } i_l \text{ je nejmenší prvek} \\ \text{v množině } \{1, \dots, n\} \setminus \{i_1, \dots, i_{l-1}, k_l, \dots, k_{n-2}\}.$$

$\{v_{i_l}, v_{k_l}\}$  je hrana v  $T$  pro každé  $l$  a pro  $\{k, j\} = \{1, \dots, n\} \setminus \{i_1, \dots, i_{n-2}\}$  je  $\{v_k, v_j\}$  hrana v  $T$ .

### 4.15 Lemma

Posloupnost  $\{i_1, \dots, i_{n-2}\}$  je určena jednoznačně a pro každé  $l$  je graf  $T \setminus \{v_{i_1}, \dots, v_{i_{l-1}}\}$  strom, v němž jsou uzly stupně různé od 1 právě  $v_{k_l}, \dots, v_{k_{n-2}}$ .

### 4.16 Věta

Počet stromů s danou  $n$ -prvkovou množinou uzlů je roven  $n^{n-2}$ .

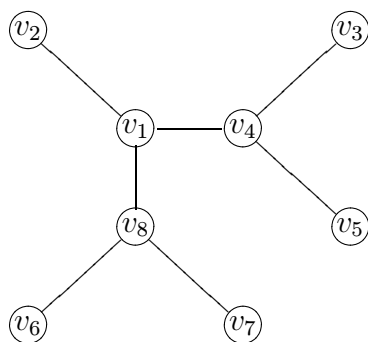
**Důkaz:** Definujeme zobrazení  $s : T(v_1, \dots, v_n) \rightarrow \{1, \dots, n\}^{n-2}$  a  $t : \{1, \dots, n\}^{n-2} \rightarrow T(v_1, \dots, v_n)$  taková, že  $T \approx s(T)$ ,  $t(S) \approx S$ . Je zřejmé, že odebráním uzlu stupně 1 spolu s přílehlou hranou ze stromu s alespoň třemi uzly dostaneme opět strom.

Definujeme zobrazení  $s$ . Nechť  $T \in T(v_1, \dots, v_n)$ . Pro  $l \in \{1, \dots, n-2\}$  položíme  $i_l$  rovno nejmenšímu indexu uzlu stupně 1 ve stromu  $T_e = T \setminus \{v_{i_1}, \dots, v_{i_{l-1}}\}$  a  $k_l$  indexu uzlu s ním sousedního.  $T \approx s(T)$ .

Definujeme zobrazení  $t$ . Nechť  $S = [k_1, \dots, k_{n-2}] \in \{1, \dots, n\}^{n-2}$ . Pro  $l \in \{1, \dots, n-2\}$  položíme  $i_l$  rovno nejmenšímu prvku v  $\{1, \dots, n\} \setminus \{i_1, \dots, i_{l-1}, k_l, \dots, k_{n-2}\}$ . Hranami budou  $\{v_{i_l}, v_{k_l}\}$  a  $\{v_k, v_j\}$ , kde  $\{k, j\} = \{1, \dots, n\} \setminus \{i_1, \dots, i_{n-2}\}$ . Tyto hrany jsou navzájem různé.  $t(S) \approx S$ .

Ještě bychom měli ověřit (ale neučiníme tak), že  $s \circ t = id_{\{1, \dots, n\}^{n-2}}$  a  $t \circ s = id_{T(v_1, \dots, v_n)}$ . □

Příklad: Budeme hledat kód stromu.



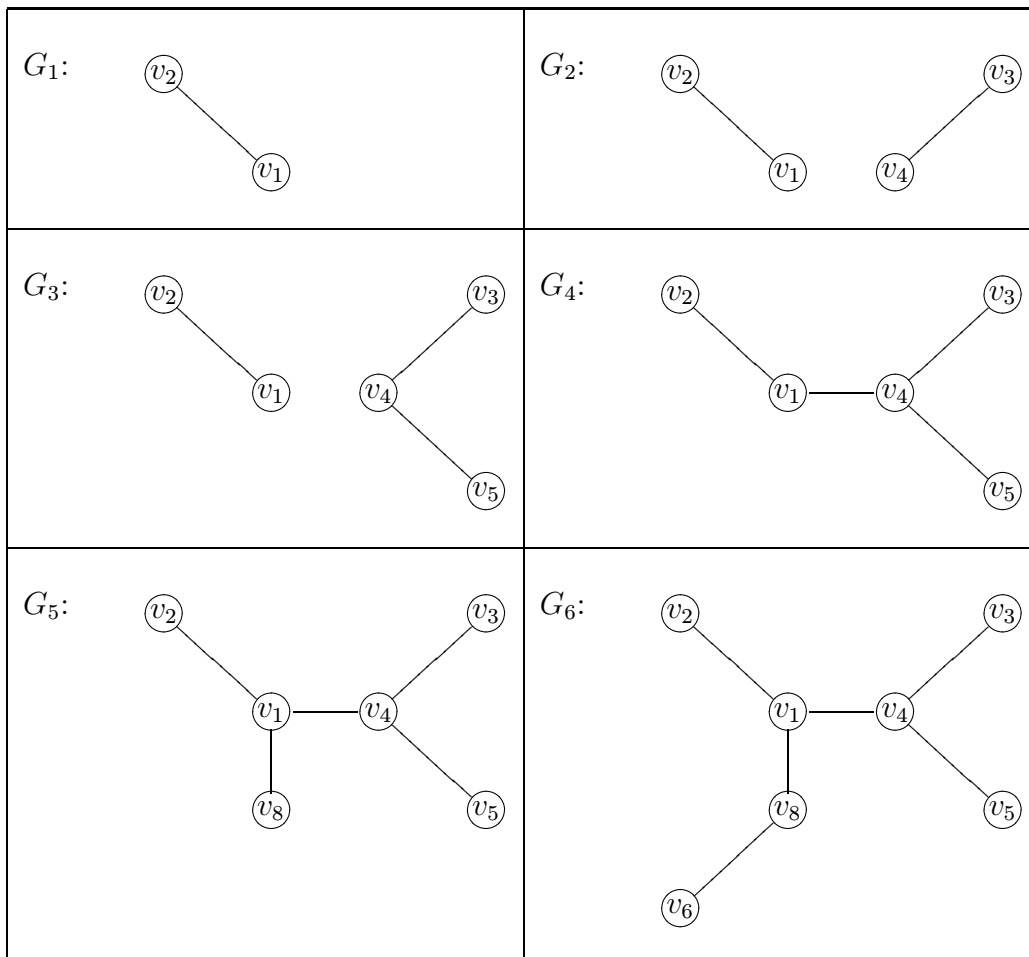
Postupně odebíráme uzly stupně 1 s nejmenším indexem a vytváříme posloupnost indexů sousedů. Nakonec tak dostáváme výsledný kód  $[1, 4, 4, 1, 8, 8]$ .

Nyní výsledný kód dekódujeme. V kódu  $[1, 4, 4, 1, 8, 8]$  je 2 nejmenší index, který zde není. Vytvoříme tedy kód  $[2, 4, 4, 1, 8, 8]$ . V tomto kódu je 3 nejmenší index, který chybí. Vytvoříme tedy kód  $[2, 3, 4, 1, 8, 8]$ . Nyní chybí index 5, vytvoříme proto kód  $[2, 3, 5, 1, 8, 8]$ . Analogicky postupně vytvoříme kódy  $[2, 3, 5, 4, 8, 8]$ ,  $[2, 3, 5, 4, 1, 8]$  a  $[2, 3, 5, 4, 1, 6]$ . V posledním kódu se navyskytují indexy 7, 8. Pro přehlednost si kódy sepíšeme pod sebe:

```

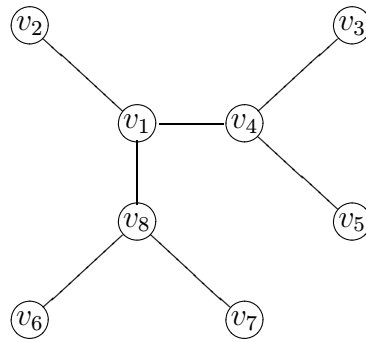
[1, 4, 4, 1, 8, 8]
|
[2, 4, 4, 1, 8, 8]
|
[2, 3, 4, 1, 8, 8]
|
[2, 3, 5, 1, 8, 8]
|
[2, 3, 5, 4, 8, 8]
|
[2, 3, 5, 4, 1, 8]
|
[2, 3, 5, 4, 1, 6] 7
|
                        8
    
```

Původní strom konstruujeme přidáváním hran naznačených pomocí |. "Průběžné" grafy tedy postupně vypadají takto:





A konečně výsledný strom  $G_7$ :



## 5 KOSTRA GRAFU, HLEDÁNÍ MINIMÁLNÍ KOSTRY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

### 5.1 Definice

**Kostra** neorientovaného grafu je každý jeho maximální částečný graf bez kružnic. Kostra v hranově ohodnoceném grafu je **minimální**, jestliže má nejmenší možný součet ohodnocení hran.

### 5.2 Věta

Každý neorientovaný graf obsahuje kostru. Kostra souvislého neorientovaného grafu je strom. Každý hranově ohodnocený neorientovaný graf obsahuje minimální kostru.

**Důkaz:** Plyne z věty 4.5 a konečnosti grafu. □

Existuje několik navzájem si blízkých algoritmů umožňujících nalézt minimální kostru.

### 5.3 Borůvkův-Kruskalův algoritmus

---

*Vstupní data:* Hranově ohodnocený souvislý neorientovaný graf  $G := (V, E)$ .

*Úkol:* Nalézt minimální kostru.

*Označení:*  $H := (V, E_i)$  ... posloupnost částečných podgrafů  
 $R_i$  ... posloupnost rozkladů množiny  $V$

---

*1.krok:* Hrany grafu  $G$  uspořádáme do neklesající posloupnosti podle jejich ohodnocení

*2.krok:*  $E_0 := \emptyset$ ,  
 $R_0 := \{\{v\} \mid v \in V\}$ ,  
 $i := 0$ .

*3.krok:* Pro každou hranu  $\{u, v\}$  grafu  $G$  v pořadí daném ohodnocením provádíme:

Zjistíme, zda uzly  $u$  a  $v$  patří do různých tříd rozkladu  $R_i$ . Pokud ano, provedeme

$$\begin{aligned} i &:= i + 1, \\ E_i &:= E_{i-1} \cup \{\{u, v\}\}, \\ R_i &:= R_{i-1} \cup \{[u]_{R_{i-1}} \cup [v]_{R_{i-1}}\}, \end{aligned}$$

tedy sjednotíme třídy rozkladu  $R_{i-1}$  obsahující uzly  $u$  a  $v$ . Vlastně se ptáme, zda hrana  $\{u, v\}$  uzavře kružnici.

*4.krok:*  $H_i$  je minimální kostra.

## 5.4 Jarníkuv-Primův algoritmus

---

*Vstupní data:* Hranově ohodnocený souvislý graf  $G := (V, E)$  s ohodnocením  $h$

*Úkol:* Nalézt minimální kostru.

*Označení:*  $H := (V, E_i)$  ... posloupnost částečných podgrafů

---

*1.krok:* Zvolíme libovolný uzel  $u \in V$ . Položíme

$$V_0 := \{u\},$$

$$E_0 := \emptyset,$$

$$i := 0,$$

$$M := V \setminus \{u\}.$$

Pro každý uzel  $v \in M$  provedeme:

Je-li  $\{u, v\}$  hrana grafu  $G$ , pak položíme

$$V_v := u,$$

$$d_v := h(\{u, v\}),$$

jinak  $d_v := \infty$ .

*2.krok:* Jestliže  $M = \emptyset$ , pak  $H_i$  je minimální kostra, *ZASTAV*.

*3.krok:* Zvolíme uzel  $v \in M$ , pro který je  $d_v$  minimální. Položíme

$$i := i + 1,$$

$$V_i := V_{i-1} \cup \{v\},$$

$$E_i := E_{i-1} \cup \{\{U_v, v\}\},$$

$$M := M \setminus \{v\}.$$

*4.krok:* Pro každý uzel  $w \in M$  provedeme:

Je-li  $\{v, w\}$  hrana v grafu  $G$  a její ohodnocení je menší než  $d_w$ , položíme

$$U_w := v,$$

$$d_w := h(\{v, w\}).$$

*5.krok:* Jdeme na 2.krok.

## 5.5 Obecný algoritmus

---

<i>Vstupní data:</i>	Hranově ohodnocený neorientovaný graf $G := (V, E)$
<i>Úkol:</i>	Nalézt minimální kostru.
<i>Označení:</i>	$H := (V, E_i) \dots$ posloupnost částečných podgrafů $R_i \dots$ posloupnost rozkladů množiny $V$

---

*1.krok:* Položíme

$$\begin{aligned} i &:= 0, \\ R &:= \emptyset, \\ R_0 &:= \{\{v\} \mid v \in V\}. \end{aligned}$$

*2.krok:* Jestliže  $R = \{V\}$ , pak  $H_i$  je minimální kostra, *ZASTAV*.

*3.krok:* Zvolíme libovolný prvek  $C := R_i$ , tedy souvislou komponentu v  $H_i$ .

*4.krok:* Z hran  $\{u, v\}$ , kde  $u \in C$  a  $v \notin C$ , vybereme hranu s minimálním ohodnocením.

*5.krok:* Položíme

$$\begin{aligned} i &:= i + 1, \\ E_i &:= E_{i-1} \cup \{u, v\}, \\ R_i &:= R_{i-1} \setminus \{[u]_{R_{i-1}}, [v]_{R_{i-1}}\} \cup \{[u]_{R_i} \cup [v]_{R_i}\}, \end{aligned}$$

tedy sjednotíme třídy rozkladu  $R_{i-1}$  obsahující uzly  $u$  a  $v$ .

*6.krok:* Jdeme na 2.krok.

## 5.6 Věta

Algoritmus 5.5 funguje správně.

**Důkaz:** Protože je hran konečně mnoho, algoritmus se zastaví. Indukcí budeme dokazovat, že graf  $H_i$  je vždy obsažen v některé minimální kostře  $K_i$ .

*Základní krok:* Na začátku výpočtu je graf  $H_0$  diskrétní, a proto obsažen v každé minimální kostře.

*Indukční krok:* Předpokládejme, že graf  $H_{i-1}$  je obsažen v minimální kostře  $K_{i-1}$ . Je-li přidávaná hrana  $\{u, v\}$ , kde  $u$  je uzel ležící ve zvolené komponentě  $C$ , hranou v  $K_{i-1}$ , je i  $H_i$  obsažena v  $K_{i-1}$  a stačí položit  $K_i := K_{i-1}$ . Pokud tomu tak není, pak existuje právě jedna cesta z  $u$  do  $v$  v  $K_{i-1}$ . Na ní leží hrana  $\{x, y\}$  taková, že  $x \in C$  a  $y \notin C$ .

Z volby hrany  $\{u, v\}$  plyne, že ohodnocení hrany  $\{x, y\}$  je větší nebo rovno ohodnocení hrany  $\{u, v\}$ . Odebráním hrany  $\{x, y\}$  a přidáním hrany  $\{u, v\}$  dostaneme z  $K_{i-1}$  opět kostru  $K_i$ , jejíž ohodnocení není vyšší, a proto je také minimální. Přitom  $H_i$  je obsažen v  $K_i$ . Algoritmus skončí nalezením kostry a ta je také minimální.

□

## 5.7 Věta

Všechny minimální kostry téhož souvislého neorientovaného grafu mají stejné složení ohodnocení.

**Důkaz:** Jestliže  $(V, T_1), (V, T_2)$  jsou minimální kostry a  $\{u, v\} \in T_1 \setminus T_2$ , pak existuje hrana  $\{\hat{u}, \hat{v}\} \in T_2 \setminus T_1$  taková, že  $\{u, v\}$  leží na cestě v  $T_1$  spojující  $\hat{u}$  a  $\hat{v}$  a  $\{\hat{u}, \hat{v}\}$  leží na cestě v  $T_2$  spojující  $u, v$ . Když totiž vypustíme hranu  $\{u, v\}$  z  $T_1$ , dostaneme les se dvěma souvislými komponentami  $C_u, C_v$ , přičemž  $u \in C_u$  a  $v \in C_v$ .

Na cestě z  $u$  do  $v$  v  $T_2$  leží hrana  $\{\hat{u}, \hat{v}\}$  spojující obě komponenty, přičemž  $\hat{u} \in C_u$  a  $\hat{v} \in C_v$ . Přitom cesta spojující  $u$  a  $\hat{u}$  leží v  $C_u$ , cesta spojující  $v$  a  $\hat{v}$  leží v  $C_v$ . Jejich spojením s hranou  $\{u, v\}$  dostaneme cestu v  $T_1$  spojující  $\hat{u}$  a  $\hat{v}$ . Přitom vypuštěním hrany  $\{u, v\}$  z  $T_1$  a přidáním  $\{\hat{u}, \hat{v}\}$  dostaneme opět kostru, tedy ohodnocení  $\{\hat{u}, \hat{v}\}$  je větší nebo rovno ohodnocení  $\{u, v\}$ . Podobně obráceně, tedy ohodnocení  $\{u, v\}$  je rovno ohodnocení  $\{\hat{u}, \hat{v}\}$ .

Nechť  $K, L$  jsou různé minimální kostry souvislého grafu  $G$ . Pak existuje konečná posloupnost

$$K = K_0, K_1, \dots, K_p = L$$

koster grafu  $G$  taková, že  $K_{i-1}, K_i$  mají stejné složení ohodnocení hran a  $|K_i \setminus L| < |K_{i-1} \setminus L|$ . Nahrazením hrany  $\{u, v\}$  v  $K_{i-1} \setminus L$  hranou  $\{\hat{u}, \hat{v}\}$  v  $L \setminus K_{i-1}$  podle předcházejícího totiž dostaneme kostru  $K_i$  s týmž složením ohodnocení hran, kde  $|K_i \setminus L| = |K_{i-1} \setminus L| - 1$ .

□

## 6 SOUVISLÉ KOMPONENTY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

Obvykle se používají metody založené na prohledávání grafu do hloubky.

### 6.1 Algoritmus hledání souvislých komponent neorientovaného grafu

---

<i>Vstupní data:</i>	Neorientovaný graf $G := (V, E)$
<i>Úkol:</i>	Nalézt souvislé komponenty.
<i>Označení:</i>	$f \dots$ předchůdce uzlu ve stromu prohledávání $v \dots$ referenční uzel $N \dots$ množina nenalezených uzlů $U \dots$ množina neprobádaných hran $D \dots$ množina nalezených uzlů stromu prohledávání $T \dots$ Množina nalezených hran stromu prohledávání

---

*1.krok:* Položíme

$$\begin{aligned} N &:= V, \\ U &:= E, \\ D &:= \emptyset, \\ T &:= \emptyset \end{aligned}$$

*2.krok:* Je-li  $N = \emptyset$ , ZASTAV.

*3.krok:* Zvolíme  $v \in N$ , položíme

$$\begin{aligned} f(v) &:= 0, \\ N &:= N \setminus \{v\}, \\ D &:= \{v\} \end{aligned}$$

*4.krok:* Jestliže neexistuje hrana vycházející z  $v$  a ležící v  $U$ , pak jestliže  $f(v) = 0$ , jdeme na 6.krok, jinak položíme  $v := f(v)$  a jdeme na 4.krok.

*5.krok:* Zvolíme hranu  $h := \{v, z\} \in U$  a položíme  $U := U \setminus \{h\}$ . Pokud je  $z \in N$ , pak provedeme

$$\begin{aligned} N &:= N \setminus \{z\} \\ D &:= D \cup \{z\}, \\ T &:= T \cup \{h\}, \\ f(z) &:= v, \\ v &:= z \end{aligned}$$

a jdeme na 4.krok

*6.krok:* V  $D$  jsou uzly a v  $T$  hrany souvislé komponenty, položíme  $D := \emptyset, T := \emptyset$  a jdeme na 2.krok

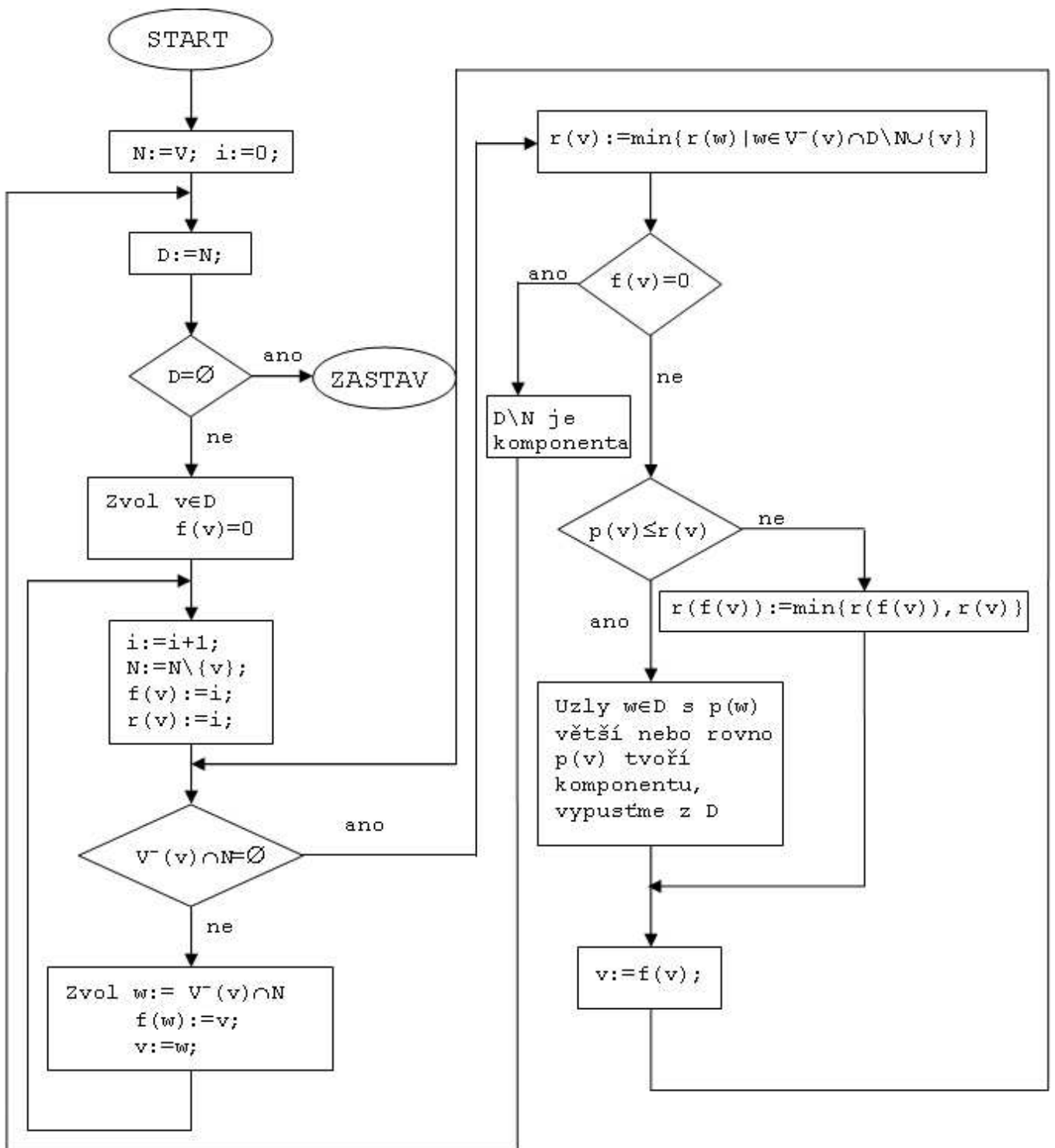
## 6.2 Algoritmus hledání souvislých komponent v orientovaných grafech

---

<i>Vstupní data:</i>	Orientovaný graf $G := (V, E)$
<i>Úkol:</i>	Nalézt souvislé komponenty v $G$
<i>Označení:</i>	$p(v)$ ... pořadí uzlu $v$
	$f(v)$ ... přímý předchůdce uzlu $v$ ve stromu prohledávání
	$r(v)$ ... pomocný index
	$D$ ... množina dovolených uzlů

---

Vývojový diagram je umístěn na následující stránce.





**Kořen komponenty** je ten její uzel, který jsme objevili jako první.

### 6.3 Věta

Algoritmus 6.2 funguje správně.

**Důkaz:** Je zřejmé, že z komponenty  $K$  se vrátíme přes její kořen ve stromu prohledávání teprve po nalezení všech jejích uzlů. Kdyby totiž uzel  $u$  zůstal neobjeven, pak na cestě z  $v$  do  $u$ , která celá patří do  $K$ , leží hrana  $[v_1, u_1]$  taková, že uzel  $v_1$  byl objeven, avšak  $u_1$  ne. Z  $v_1$  jsme se však mohli vrátit až poté, co jsme našli  $u_1$ , což je spor. Každá souvislá komponenta má tedy ve stromu prohledávání tvar

$$T(v) \setminus T(v_1) \setminus \dots \setminus T(v_k),$$

kde  $v_1, \dots, v_k$  jsou uzly, jimiž jsme z komponenty vystoupili. Stačí tedy ukázat, že algoritmus správně určí kořeny komponent.

Indukcí dokážeme, že při návratu z uzlu  $v$  je  $r(v) < p(v)$  právě tehdy, když  $v$  není kořenem komponenty.

Předpokládejme tedy, že dosud jsme při návratu všechny uzly zařadili správně. Pokud  $r(v) < p(v)$ , existuje  $u \in T(v)$ , z něhož vede šipka do  $w$ , kde  $p(w) < p(v)$  a  $w$  je dovozen. Pokud  $w$  je předchůdcem  $v$  v  $T$ , máme

$$w \xrightarrow{*} v \xrightarrow{*} u \rightarrow w$$

a tedy  $v$  a  $w$  patří do téže komponenty. Proto  $w$  není kořenem komponenty. Pokud  $w$  není předchůdcem  $v$  v  $T$ , existuje uzel  $z$ , v němž se větví cesty k  $w$  a  $v$  v  $T$ . Přitom  $z$  a  $w$  patří podle indukčního předpokladu do téže komponenty, jinak by uzel  $w$  nebyl dovozen. Dostáváme

$$w \xrightarrow{*} z \xrightarrow{*} v \xrightarrow{*} u \rightarrow w$$

a tedy  $z$  a  $v$  patří do téže komponenty. Proto  $v$  není kořenem komponenty.

Pokud obráceně  $v$  není kořen komponenty, existují uzly této komponenty objevené před  $v$ . Z  $v$  do kořene komponenty vede cesta

$$v = v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_k = u.$$

Existuje nejmenší index  $l$  takový, že  $v_l$  je uzel komponenty objevený před  $v$ . Pak ovšem  $r(v) \leq r(v_{l-1}) \leq p(v_l) < p(v)$ .

□

## 7 HLEDÁNÍ OPTIMÁLNÍ CESTY

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

Protože algoritmy hledají nejkratší sled, graf nesmí obsahovat kružnici se záporným ohodnocením.

### 7.1 Algoritmus Moorův

---

*Vstupní data:* Orientovaný graf  $G := (V, E)$

uzel  $s \in V$

*Úkol:* Pro každý uzel  $v$  nalézt cestu z  $s$  do  $v$  s nejmenším počtem hran.

---

*0.krok:* Položíme

$$(p_s, d_s) := (0, 0),$$

$$k := 0,$$

$$V_0 := \{s\}.$$

*1.krok:* Položíme  $V_{k+1} = \emptyset$ . Pro každý uzel  $i \in V_k$  provedeme:

každého neoznačeného následníka  $j$  uzlu  $i$  označíme  $(p_j, d_j) := (i, k + 1)$

Položíme  $V_{k+1} := V_{k+1} \cup \{j\}$

*2.krok:* Pokud  $V_{k+1} = \emptyset$ , ZASTAV, jinak jdi na 1.krok.

### 7.2 Věta

Algoritmus 7.1 funguje správně.

**Důkaz:** Máme dokázat, že po skončení platí:

Pokud je uzel  $j$  označován, pak  $d_j := d(s, j)$ , a pokud není označován, pak neexistuje cesta z  $s$  do  $j$ .

Zřejmě uzel dostane značku nejvýše jednou. Pokud má značku  $(i, k + 1)$ , existuje cesta z  $s$  do  $j$  délky  $k + 1$ . Dostaneme ji připojením hrany  $[i, j]$  k cestě z  $s$  do  $i$  délky  $k$ , která existuje z indukčního předpokladu.

Nechť  $d(s, j) = b$ . Indukcí k  $b$  dostáváme, že pak  $j$  má značku a přitom  $d_j = b$ . Nechť  $Q$  je cesta z  $s$  do  $j$  délky  $b$  a nechť  $i$  je její předposlední uzel. Potom  $d(s, i) = b - 1$  a podle indukčního předpokladu má  $i$  značku a  $d_i = b - 1$ . Pro  $k = b - 1$  uzel  $j$  dostane značku  $d_j = k + 1 = b$ .

□

### 7.3 Dijkstrův algoritmus

---

*Vstupní data:* Nezáporně hranově ohodnocený orientovaný graf  $G := (V, E)$

*Úkol:* Pro každý uzel  $v$  nalézt cestu z  $s$  do  $v$  s nejmenším součtem ohodnocení hran.

---

0.krok: Položíme

$$(p_s, d_s) := (0, 0),$$

$$S := \{s\},$$

$$(p_v, d_v) := (0, \infty) \text{ pro } v \neq s, \text{ pokud neexistuje hrana z } s \text{ do } v, (p_v, d_v) := (s, c_{s,v}) \text{ pokud existuje hrana z } s \text{ do } v$$

1.krok: Pokud  $S = V$ , ZASTAV. Najdeme uzel  $k \in V \setminus S$  s nejmenší hodnotou  $d_k$ . Položíme  $S := S \cup \{k\}$ . Pro každý uzel  $j \in V(k) \setminus S$  provedeme:

$$\text{pokud } d_k \neq c_{k,j} < d_j, \text{ pak } d_j := d_k + c_{k,j}$$

Jdeme na 1.krok.

## 7.4 Věta

Algoritmus 7.3 funguje správně.

**Důkaz:** Indukcí dokážeme, že každý uzel  $k$  je zařazen do  $S$  s  $d_k = d(s, k)$ , kde  $d(s, k)$  označuje délku nejkratší cesty z  $s$  do  $k$  a přitom celá tato cesta leží v  $S$ .

*Základní krok:* Uzel  $s$  je zařazen s  $d_s = 0 = d(s, s)$ .

*Indukční krok:* Předpokládejme, že dosud byl každý uzel  $i$  v  $S$  zařazen s  $d_i = d(s, i)$  a  $d_k = \infty$ , pak neexistuje šipka z uzlů v  $S$  do  $V \setminus S$  a tedy  $d(s, k) = \infty$  pro každý uzel  $k \in V \setminus S$ . Předpokládejme  $d_k = \infty$ . Uvažujme cestu z  $s$  do  $k$  délky  $d(s, k)$ , tedy nejkratší cestu, a nechť  $v$  je její první uzel ležící ve  $V \setminus S$  a  $u$  jeho předchůdce. Protože  $u$  byl zařazen do  $S$ ,  $d_v \leq d_u + c_{u,v} = d(s, u) + c_{u,v}$ . Protože  $k$  dostal značku při zařazení jistého uzlu  $i$  do  $S$ ,  $d_k = d_i + c_{i,k} = d(s, i) + c_{i,k}$  je délkou jisté cesty  $p$  z  $s$  do  $k$ . Protože  $d_k$  je nejmenší značka,  $d_k \leq d_v$ . Odtud dostáváme  $d_k = d(s, k)$ , neboť  $d_v \leq d_u + c_{u,v} = d(s, u) + c_{u,v} \leq d(s, k) \leq d_k \leq d_v$ . Proto cesta  $P$  od  $s$  po  $i$  sestává z uzlů v  $S$  podle indukčního předpokladu a po přidání  $k$  do  $S$  je celá v  $S$ .

□

## 7.5 Algoritmus Fordův

---

*Vstupní data:* Hranově ohodnocený orientovaný graf  $G := (V, E)$  bez kružnic záporné délky,  
uzel  $s \in V$ .

*Úkol:* Najít nejkratší cesty z uzlu  $s$  do ostatních uzlů.

---

0.krok: Položíme

$$\begin{aligned}(a_s, p_s, d_s) &:= (0, 0, 0), \\ (a_j, p_j, d_j) &:= (0, 0, \infty) \text{ pro } j \neq s, j \in V, \\ k &:= 0\end{aligned}$$

1.krok: Jestliže  $k = n - 1$  nebo  $a_i \neq k$  pro každý uzel  $i \in V$ , ZASTAV. Jinak pro každý uzel  $i \in V$  takový, že  $a_i = k$ , provedeme

Pro každý uzel  $j \in V^-(i)$  provedeme

Jestliže  $d_i + c_{i,j} < d_j$ , pak provedeme

$$\begin{aligned}d_j &:= d_i + c_{i,j} \\ p_j &:= i \\ a_j &:= a_i + 1\end{aligned}$$

Položíme  $k := k + 1$  a jdeme na 1.krok.

## 7.6 Věta

Algoritmus 7.5 funguje správně.

**Důkaz:** Indukcí dokážeme, že po skončení  $k$ -té iterace je  $d_j$  rovno délce nejkratší cesty s nejvýše  $k + 1$  hranami.

*Základní krok:* Po skončení nulté iterace je  $d_s = 0$  a pro každý uzel  $i \in V^-(s)$  platí  $d_i = c_{s,i}$ , jinak  $d_i = \infty$ . To jsou délky nejkratších cest z uzlu  $s$  s nejvýše jednou hranou.

*Indukční krok:* Předpokládejme, že v předchozích iteracích tvrzení platí. V  $k$ -té iteraci se délka  $d_j$  nejkratší cesty z  $s$  do  $j$  s nejvýše  $k$  hranami porovnává s délkou každé cesty z  $s$  do  $j$  s  $k + 1$  hranami, takže po jejím skončení je  $d_j$  rovno délce nejkratší cesty s nejvýše  $k + 1$  hranami. □

## 7.7 Algoritmus vypouštění zdrojů

---

*Vstupní data:* Hranově ohodnocený orientovaný graf  $G := (V, E)$  bez kružnic, uzel  $s \in V$ .

*Úkol:* Najít nejkratší cesty z uzlu  $s$  do ostatních uzlů.

*Označení:*  $Z \dots$  zásobník

---

0.krok: Položíme

$$(r_s, p_s, d_s) := (0, 0, 0),$$

$$(r_j, p_j, d_j) := (st^+(j), 0, \infty).$$

Do zásobníku vložíme všechny zdroje tak, aby  $s$  byl poslední.

1.krok: Pro poslední prvek  $i$  ze  $Z$  provedeme:

Odebereme  $i$  ze  $Z$ . Pro každý uzel  $j \in V^-(i)$  provedeme:

Jestliže  $d_i + c_{i,j} < d_j$ , pak provedeme:

$$d_j := d_i + c_{i,j}$$

$$p_j := i$$

$$r_j := r_j - 1$$

Jestliže  $r_j = 0$ , přidáme  $j$  do  $Z$ .

2.krok: Jestliže  $Z \neq \emptyset$ , ZASTAV. Jdeme na 1.krok.

## 7.8 Věta

Algoritmus 7.8 funguje správně.

**Důkaz:** Je zřejmé, že skončíme, neboť každý uzel se do zásobníku dostane nejvýše jednou. Z principu fundované indukce vyplývá, že se do něj dostaneme, neboť je zřejmé, že pokud se do zásobníku dostali všichni jeho předchůdci, dostal se tam také. Stačí dokázat, že pak skutečně platí  $d_j = d(s, j)$  pro každý uzel  $j \in V$  indukci podle pořadí, v němž jsme uzly vkládali do zásobníku.

Nechť  $Q$  je cesta z  $s$  do  $j$  délky  $d(s, j)$  a  $i$  její předposlední uzel, podle indukčního předpokladu  $d_i = d(s, i)$ . Tedy  $d_j \leq d_i + c_{i,j} = d(s, i) + c_{i,j} = d(s, j)$ .

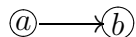
□

Algoritmus 7.8 můžeme použít i k nalezení nejdelších cest.

## Analýza kritické cesty

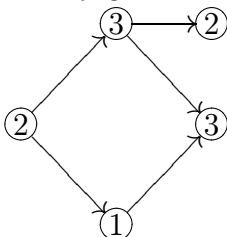
Tato metoda se používá k analýze projektů sestávajících z činností ohodnocených délkou trvání a návazností mezi nimi.

Postupujeme tak, že si nejprve sestrojíme graf projektu, v němž jsou činnosti znázorněny uzly a návaznosti mezi nimi šípkami. Tedy



znamená, že činnost  $b$  může začít teprve po skončení činnosti  $a$ .

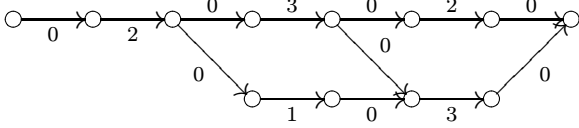
Uzly znázorňují činnosti ohodnocené délkou trvání těchto činností. Obdržíme tak orientovaný graf bez kružnic.



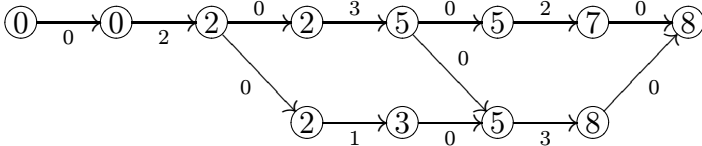
Graf kreslíme tak, aby šípky směřovaly zleva doprava, myšlená časová osa probíhá také zleva doprava.

Abychom mohli použít algoritmus 7.8, nahradíme tento graf grafem s ohodnocenými hranami. Každý uzel nahradíme dvěma uzly a šípkou mezi nimi s tímž ohodnocením, jako měl původní uzel. Šípky znázorňující návaznosti budou mít nulové ohodnocení. Přitom

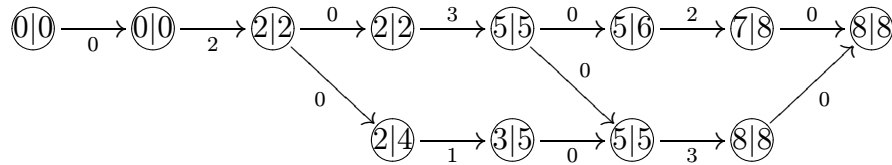
tyto šípky vycházejí z koncových uzlů činností a vcházejí do počátečních uzlů činností. Doplňme počáteční uzel, z něhož vedou šípky do všech zdrojů, a koncový uzel do něhož vedou šípky z koncových uzlů.



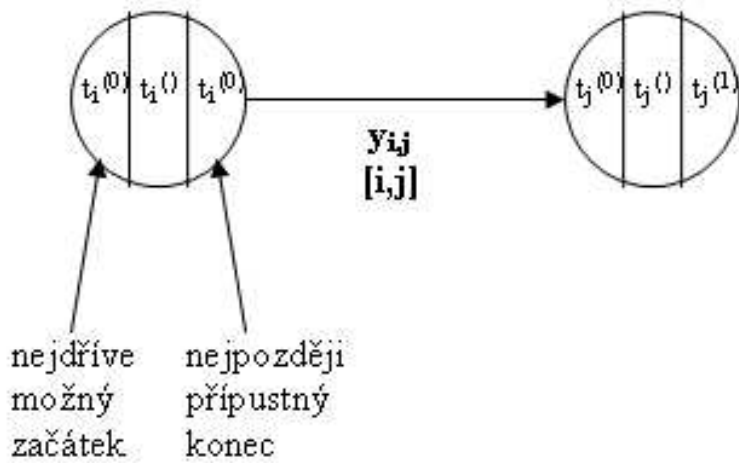
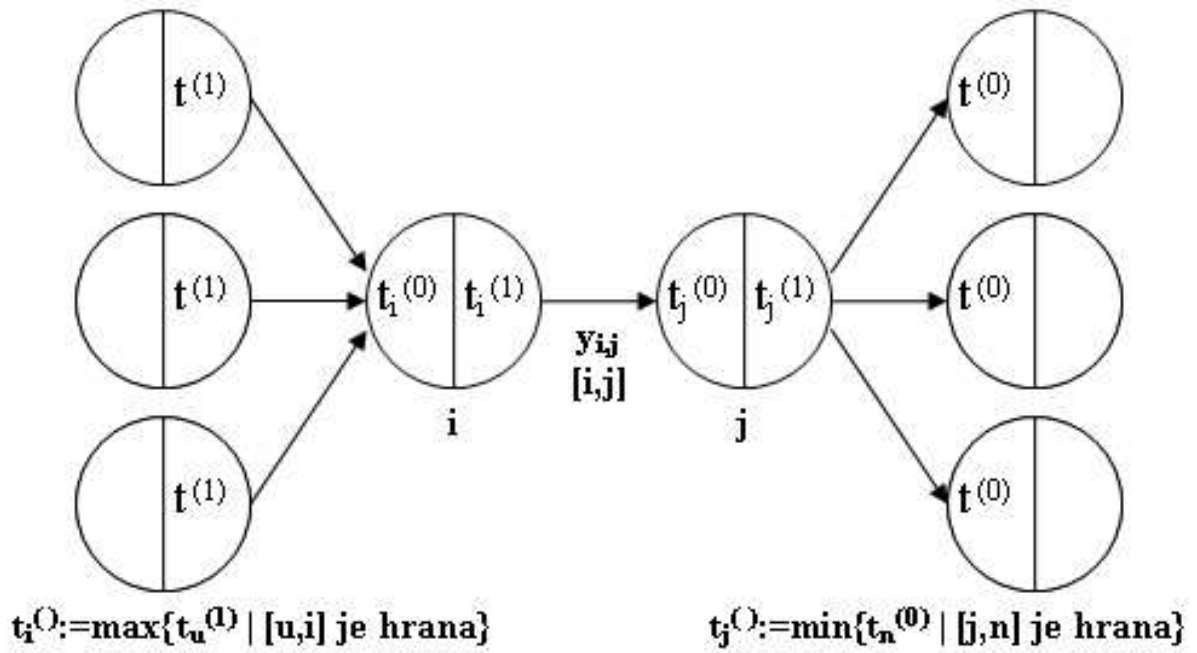
V tomto grafu hledáme nejdříve délku nejdelší cesty z počátečního uzlu do ostatních uzlů, přičemž vzdálenosti udáváme od počátku tak, aby v koncovém uzlu byla hodnota nejdelší cesty nebo předpokládaná délka trvání projektu.



Nyní dohledáme nejdelší cesty z koncového uzlu do ostatních uzlů.



Nejdelší cesty jsou **kritické**. Na nich ve skutečnosti závisí doba trvání projektu. *Kritická cesta je tvořena uzly s ohodnocením  $x|x$ .*



$t_i^{(0)}$  ... nejdříve možný začátek činnosti  $[i, j]$   
 $t_i^{(1)}$  ... nejpozději přípustný konec činnosti  $[i, j]$

$t_j^{(0)}$  ... nejdříve možný konec činnosti  $[i, j]$   
 $t_j^{(1)}$  ... nejpozději přípustný konec činnosti  $[i, j]$   
 $y_{i,j}$  ... doba trvání činnosti  $[i, j]$

### Rezervy činnosti $[i, j]$

*Celková rezerva (CR):*  $CR_{i,j} = t_j^{(1)} - t_i^{(0)} - y_{i,j} = t_i^{(1)} - t_i^{(0)} = t_j^{(1)} - t_j^{(0)}$

*Volná rezerva (RV):*  $RV_{i,j} = t_j^{(1)} - t_i^{(0)} - y_{i,j} = t_j^{(1)} - t_j^{(0)}$

Udává, o kolik lze posunout začátek činnosti, aby se nezměnily nejdříve možné začátky všech činností bezprostředně následujících.

*Závislá rezerva (RZ):*  $RZ_{i,j} = t_j^{(1)} - t_i^{(0)} - y_{i,j} = t_i^{(1)} - t_i^{(0)}$

Udává, o kolik lze prodloužit trvání činnosti, aby se nezměnil nejpozději přípustný konec činností bezprostředně předcházejících.

*Nezávislá rezerva (RN):*  $RN_{i,j} = t_j^{(0)} - t_i^{(0)} - y_{i,j}$

Udává, o kolik lze prodloužit trvání činnosti, aby se nezměnil nejdříve možný začátek činností bezprostředně následujících a nejpozději přípustný konec činností bezprostředně předcházejících.

**Propustnost cesty** v hranově ohodnoceném orientovaném grafu je minimum z ohodnocení jejích hran.

## 7.9 Algoritmus nalezení cesty s největší propustností

---

*Vstupní data:* Hranově ohodnocený orientovaný graf  $G := (V, E)$ ,  
uzel  $s \in V$ .

*Úkol:* Najít cesty s největší propustností z uzlu  $s$  do ostatních uzlů.

*Označení:*  $b_j$  ... dosud nalezená největší propustnost cesty z  $s$  do  $j$   
 $c_{i,j}$  ... ohodnocení hrany  $[i, j]$

---

*1.krok:* Položíme

$$(p_s, b_s) := (0, \infty), \text{ ostatní uzly bez značky,}$$

$$S := \{s\},$$

$$v := s.$$

*2.krok:* Pro všechny uzly  $u \in V^-(v)$  provedeme:

Pokud  $b_u < \min\{b_v, c_{v,u}\}$ , pak provedeme

$$b_u := \min\{b_v, c_{v,u}\}$$

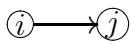
Pokud  $S = V$ , ZASTAV. Najdeme uzel  $v \in S$  s maximálním  $b_v$ , položíme  $S := \{v\}$ , jdeme na 2.krok.



## 8 TOKY V SÍTÍCH

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

Je dán orientovaný graf  $G := (V, E)$  s nezáporným ohodnocením hran, jemuž říkáme **kapacita**, počáteční uzel  $s$  a koncový uzel  $t$ .



$b_{i,j}$ ... kapacita hrany  $[i,j]$

$x_{i,j}$ ... tok hranou  $[i,j]$

Hledáme nezáporné ohodnocení  $x_{i,j}$  takové, že  $0 \leq x_{i,j} \leq b_{i,j}$  a

$$\sum_{j \in V^+(i)} x_{j,i} - \sum_{j \in V^-(i)} x_{i,j} = 0, \quad i \in V \setminus \{s, t\}.$$

Takovému ohodnocení říkáme **tok**. Číslo

$$f(x) := \sum_{j \in V^+(t)} x_{j,t} - \sum_{j \in V^-(t)} x_{t,j}$$

je **velikost toku** z  $s$  do  $t$ .

Budeme hledat tok s maximální velikostí. Takový tok je **maximální**.

Nechť  $P$  je polocesta v  $G$ . Množinu souhlasně orientovaných hran označíme  $P^+$ , množinu nesouhlasně orientovaných hran  $P^-$ .

Souhlasně orientované hraně  $[i, j]$  přiřadíme  $b_{i,j} - x_{i,j}$ , nesouhlasně orientované hraně  $[i, j]$  přiřadíme  $x_{i,j}$ , kterým říkáme **rezervy**. Nejmenší z rezerv hran polocesty je rezerva polocesty.

**Zlepšující polocesta** je polocesta z  $s$  do  $t$  s kladnou rezervou.

Podmnožina  $S \subseteq V$  je **řez** mezi  $s$  a  $t$ , jestliže  $s \in S$ ,  $t \notin S$  (zjednodušené, obvykle se za řez bere množina hran). Součet kapacit hran vycházejících z uzlů v  $S$  a vcházejících do uzlů ve  $V \setminus S$  je **kapacita řezu**. Značíme  $b(S, V \setminus S)$ . Platí

$$f(x) = x(S, V \setminus S) - x(V \setminus S, S) \leq b(S, V \setminus S)$$

**Minimální řez** je řez s minimální kapacitou.

### 8.1 Věta

Velikost maximálního toku z  $s$  do  $t$  je rovna kapacitě minimálního řezu mezi  $s$  a  $t$ .

**Důkaz:** Maximální tok existuje, neboť jde o maximum spojitě funkce na kompaktní množině (=ohraničená a uzavřená). Minimální řez existuje, protože řezů je konečně mnoho. Označme  $x^*$  maximální tok z  $s$  do  $t$  a  $b^*$  hodnotu minimálního řezu mezi  $s$  a  $t$ . Zřejmě  $f(x^*) = b(S, V \setminus S)$ .

Nechť  $S$  je množina všech těch uzlů, do nichž vede z  $s$  polocesta s kladnou rezervou a která navíc obsahuje  $s$ . Pokud by  $t \in S$ , existovala by zlepšující polocesta z  $s$  do  $t$  a tok bychom mohli zvětšit o rezervu této polocesty, což je spor. Tedy  $t \notin S$ , a  $S$  je řez mezi  $s$  a  $t$ . Pro každou hranu  $[i, j]$  vedoucí z  $S$  do  $V \setminus S$  platí, že  $x_{i,j} = b_{i,j}$ , jinak by uzel  $j$  patřil do  $S$  a pro každou hranu  $[i, j]$  vedoucí z  $V \setminus S$  do  $S$  platí  $x_{i,j} = 0$ , jinak by uzel  $i$  patřil do  $S$ . Tedy  $f(x^*) = x^*(S, V \setminus S) - x^*(V \setminus S, S) = x^*(S, V \setminus S) = b(S, V \setminus S)$ .

□

## 8.2 Věta

Tok  $x$  z  $s$  do  $t$  je maximální právě tehdy, když neexistuje zlepšující polocesta z  $s$  do  $t$ .

**Důkaz:**

( $\Rightarrow$ ) Pomocí zlepšující polocesty by se dal tok zvětšit.

( $\Leftarrow$ ) Položíme  $S := \{v \mid \text{existuje polocesta z } s \text{ do } v \text{ s kladnou rezervou}\}$ . Pak  $f(x) = b(S, V \setminus S)$ , takže  $x$  je maximální tok a  $S$  je minimální řez. □

## 8.3 Fordův-Fulkersonův algoritmus

1.krok: Položíme  $x_{ij} = 0$  pro každou hranu  $[i, j]$ .

2.krok: Moorovým algoritmem nalezneme zlepšující polocestu z  $s$  do  $t$  s nejmenším počtem hran. Pokud neexistuje, pak ZASTAV.

2.krok: Zvětšíme tok o rezervu nelené zlepšující polocesty. Jdeme na 2.krok.

## 8.4 Věta

Algoritmus 8.3 funguje správně.

**Důkaz:** Pokud výpočet skončí, neexistuje zlepšující polocesta, a tedy tok je maximální. Dokážeme, že výpočet skončí po konečně mnoha krocích.

Označíme  $\sigma^{(k)}(i)$  délku nejkratší zlepšující polocesty z  $s$  do  $i$  po  $k$  zlepšeních, to jest pro tok  $x^{(k)}$ . Dále označíme  $\tau^{(k)}(i)$  délku nejkratší zlepšující polocesty s (kladnou rezervou) z  $i$  do  $t$  po  $k$  zlepšeních, to jest pro tok  $x^{(k)}$ .

Nejdříve ukážeme, že pro každé  $k$  a každý uzel  $i$  platí

$$\sigma^{(k)}(i) \leq \sigma^{(k+1)}(i) \ \& \ \tau^{(k)}(i) \leq \tau^{(k+1)}(i).$$

Sporem dokážeme první nerovnost. Pripusťme, že pro jisté  $k$  a jistý uzel  $i$  platí  $\sigma^{(k+1)}(i) < \sigma^{(k)}(i)$ . Bez újmy na obecnosti můžeme předpokládat, že pro každý uzel  $j$  platí  $\sigma^{(k+1)}(i) \leq \sigma^{(k+1)}(j)$  nebo  $\tau^{(k)}(j) \leq \tau^{(k+1)}(j)$ . Nechť  $j$  je předposlední uzel na nejkratší zlepšující polocestě  $Q$  z  $s$  do  $i$  pro tok  $x^{(k+1)}$ . Pak  $\sigma^{(k+1)}(i) = \sigma^{(k+1)}(j) + 1$ , a tedy  $\sigma^{(k)}(j) + 1 \leq \sigma^{(k+1)}(i)$ .

Nechť tok  $x^{(k)}$  byl zlepšen po zlepšující polocestě  $P$  z  $s$  do  $t$ . Pokud hrana  $[i, j]$  nepatřila do  $P$  nebo byla v  $P$  nesouhlasná, pak  $0 < x_{ij}^{(k+1)} \leq x_{ij}^{(k)}$ . Proto  $\sigma^{(k)}(i) \leq \sigma^{(k)}(j) + 1 \leq \sigma^{(k+1)}(i)$ , což je spor s předpokladem.

Pokud hrana  $[j, i]$  nepatřila do  $P$  nebo byla souhlasná v  $P$ , pak  $x_{ji}^{(k)} \leq x_{ji}^{(k+1)} < b_{ji}$ . Proto  $\sigma^{(k)}(i) \leq \sigma^{(k)}(j) + 1 \leq \sigma^{(k+1)}(i)$ , což je spor s předpokladem. Pokud hrana  $[j, i]$  byla nesouhlasná v  $P$ , pak  $\sigma^{(k)}(j) = \sigma^{(k)}(i) + 1$ , a tedy  $\sigma^{(k)}(i) + 2 \leq \sigma^{(k+1)}(i)$ , což je spor s předpokladem.

Pro  $\tau$  je důkaz obdobný. □

Řekneme, že souhlasná hrana je **kritická**, jestliže  $x_{ij}^{(k+1)} = b_{ij}$ , a že nesouhlasná hrana je kritická, jestliže  $x_{ij}^{(k+1)} = 0$ .

## 8.5 Věta

Na každé zlepšující polocestě existuje kritická hrana.

**Důkaz:** Nechť  $[i, j]$  je kritická hrana v  $(k+1)$ -ní zlepšující polocestě  $P$ . Délka

$$m(P) = \sigma^{(k)}(i) + \tau^{(k)}(i) = \sigma^{(k)}(j) + \tau^{(k)}(j).$$

Pokud  $[i, j]$  je  $i$  v  $(r + 1)$ -ní zlepšující polocestě a  $r$  je nejmenší takové, že  $k \leq r$ , je v ní opačně.

Pokud tedy  $\tau^{(k)}(i) = \tau^{(k)}(j) + 1$ , pak  $\sigma^{(r)}(i) = \sigma^{(r)}(j) + 1$ . Odtud  $m(P) + 2 = \sigma^{(k)}(j) + \tau^{(k)}(j) + 2 \leq \sigma^{(r)}(j) + \tau^{(k)}(j) + 1 = \sigma^{(r)}(i) + \tau^{(k)}(i) \leq \sigma^{(r)}(i) + \tau^{(r)}(i) = m(Q)$ .

Protože každá polocesta má nejvýše  $n - 1$  hran, může být každá hrana kritická nejvýše  $\frac{n}{2}$ -krát. Můžeme tedy provést nejvýše  $\frac{m \cdot n}{2}$  zlepšení.

□

## 9 Párování

Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto

### 9.1 Definice

Neorientovaný graf je **bipartitní**, jestliže jeho množinu uzlů lze rozdělit na dvě disjunktní podmnožiny tak, že každá hrana má jeden uzel v jedné a druhý v druhé.

### 9.2 Definice

**Párování** v neorientovaném grafu je každý jeho částečný podgraf, v němž má každý uzel stupeň nejvýše 1. Párování je **maximální**, jestliže má největší možný počet hran.

Bipartitnímu grafu můžeme přiřadit orientovaný graf tak, že hrany budou vycházet z uzlu v první množině a vcházet do uzlu ve druhé množině. Můžeme přidat počáteční uzel a hrany vycházející z tohoto uzlu a vcházející do všech uzlů první množiny a koncový uzel a hrany vycházející ze všech uzlů druhé množiny a vcházející do koncového uzlu. Hrany ohodnotíme vhodně číslem 1.

### 9.3 Lemma

Existuje vzájemná jednoznačná korespondence (bijekce) mezi celočíselnými toky z  $s$  do  $t$  v přiřazeném orientovaném grafu a párováním v původním grafu. Je určena tak, že pro tok  $x = \{x_{ab} \mid x_{ab} = 1\}$  je párování a pro párování  $H$  je

$$x_{sv} = 1, \text{ jestliže z } v \text{ vychází hrana párování } H$$

$$x_{vu} = 1, \text{ jestliže } [v, u] \text{ je hrana párování } H$$

$$x_{ut} = 1, \text{ jestliže z } u \text{ vychází hrana párování } H$$

celočíselný tok z  $s$  do  $t$ . Velikost toku je rovna počtu hran párování.

V bipartitním grafu tedy nalezneme maximální párování tak, že k němu sestrojíme příslušný ohodnocený orientovaný graf a najdeme v něm maximální tok. Hrany, v nichž je hodnota toku 1, tvoří maximální párování.

### 9.4 Definice

Uzel je **volný** vzhledem k danému párování, jestliže nesousedí se žádnou hranou párování. Cesta je **střídavá**, jestliže se v ní střídají hrany patřící a nepatřící do párování. Je **volná**, jestliže začíná ve volném uzlu a končí ve volném uzlu.

### 9.5 Bergerova věta

Párování je maximální právě tehdy, když k němu neexistuje volná střídavá cesta.

**Důkaz:**

- ( $\Rightarrow$ ) Kdyby existovala, pak výměnou hran patřících do této cesty dostaneme párování s větším počtem hran
- ( $\Leftarrow$ ) Nechť  $X$  je Párování, ke kterému neexistuje volná střídavá cesta. Nechť  $X^*$  je nějaké maximální párování. Položme  $H = X \cup X^*$ .  $(V, H)$  má uzly stupně 1,2, a proto jeho souvislé komponenty jsou některých z těchto typů: jedna hrana, cesta sudé délky nebo kružnice sudé délky. Počet hran párování  $X$  je tedy týž, jako počet hran maximálního párování  $X^*$ .

□

## 9.6 Algoritmus Edmondsův

---

<i>Vstupní data:</i>	Neorientovaný graf $(V, E)$ a párování $X$
<i>Úkol:</i>	Najít volnou střídavou cestu.
<i>Označení:</i>	$S : L : \dots$ značky (sudá, lichá)

---

*1.krok:* Jestliže neexistuje volný uzel, pak *ZASTAV*.

*2.krok:* Všechny volné uzly dostanou značku  $S : 0$

Pokud  $b_u < \min\{b_v, c_{v,u}\}$ , pak provedeme

$$b_u := \min\{b_v, c_{v,u}\}$$

Pokud  $S = V$ , *ZASTAV*. Najdeme uzel  $v \in S$  s maximálním  $b_v$ , položíme  $S := \{v\}$ , jdeme na 2.krok.

*3.krok:* Označkový ale neprozkoumaný uzel  $i$  prozkoumáme takto:

Pokud  $i$  má značku  $S$ , pak každý neoznačkový uzel  $j$  takový, že  $\{i, j\} \in E \setminus X$ , dostane značku  $L : i$ .

Jestliže  $i$  má značku  $L$  a  $\{i, j\} \in E \setminus X$  a  $j$  je neoznačkový, pak  $j$  dostává značku  $S : i$ .

## 9.7 Věta

Nechť  $Z$  je květ objevený v grafu  $G$  při značkování vzhledem k  $X$ . Nechť po kontrakci květu  $Z$  na uzel  $Z'$  vznikne z  $G$  graf  $G'$  a z  $X$  párování  $X'$ . Potom v  $G'$  existuje volná střídavá cesta vzhledem k  $X'$  právě tehdy, když v  $G$  existuje volná střídavá cesta pro  $X$ .

**Důkaz:**

( $\Rightarrow$ ) Volnou střídavou cestu v  $G'$  vzhledem k  $X'$  doplníme úsekem v květu podle druhu hran.

( $\Leftarrow$ ) Tento směr je složitý.

□

## 10 Míry souvislosti grafu

*Podle přednášky Doc. RNDr. Josefa Niederleho, CSc. zpracoval Ondřej Bitto*

V této kapitole budeme zkoumat neorientované grafy. Necht'  $G := (V, E)$  je souvislý. Míru souvislosti můžeme udat pomocí hranových a uzlových řezů.

### 10.1 Definice

**Uzlový řez** je podmnožina  $A \subseteq V$  taková, že  $(V \setminus A, \{e \in E \mid e \cap A = \emptyset\})$  je nesouvislý.

**Hranový řez** je podmnožina  $A \subseteq E$  taková, že  $(V, E \setminus A)$  je nesouvislý.

Každý netriviální souvislý graf má hranový řez. Každý neúplný souvislý graf má uzlový řez.

### 10.2 Definice

Minimální velikost uzlového řezu se nazývá **uzlová souvislost grafu** a značí  $\kappa(G)$ . Pro úplný graf  $\kappa(K_n) := n - 1$ .

Minimální velikost hranového řezu se nazývá **hranová souvislost grafu** a značí  $\lambda(G)$ .

Triviální graf  $K_1$  má hranovou souvislost  $\lambda(K_1) := 0$ .

Každý nesouvislý graf  $G$  má  $\kappa(G) := 0$  a  $\lambda(G) := 0$ .

Cesty z  $u$  do  $v$  jsou **vnitřně disjunktní**, jestliže nemají společný vnitřní uzel. Cesty z  $u$  do  $v$  jsou **hranově disjunktní**, jestliže nemají společnou hranu.