Diskretna matematika 1 / Kombinatorika

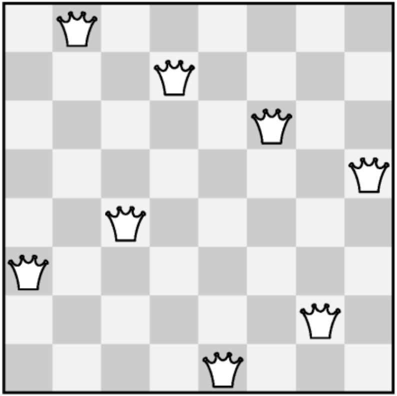
1. Pravila

Vladimir Batagelj

Univerza v Ljubljani

FMF, matematika – Finančna matematika
Ljubljana, februar 2008

Kazalo



brainwagon

Učilnica: http://ucilnica.fmf.uni-lj.si/course/view.php?id=39
Različica: 21. oktober 2013

Kaj je to kombinatorika?

Kombinatorika je del diskretne matematike, ki poskuša dati odgovore predvsem na naslednji vrsti nalog o diskretnih (števnih – končnih ali števno neskončnih) množicah sestavljenih iz "strukturiranih" objektov (konfiguracij):

- naloge o preštetju: Na koliko načinov? Kolikšna je moč množice objektov, ki zadoščajo danim pogojem? Ali sploh obstoja tak objekt?
- naloge o naštetju (generiranju): Kateri so tisti objekti, ki zadoščajo danim pogojem? Sestavi (učinkovit) postopek za generiranje (v izbranem vrstnem redu, naključno) teh objektov.

Kaj je to kombinatorika?

Pomembno podskupino nalog o naštetju sestavljajo:

- optimizacijske naloge: Kateri so tisti objekti, za katere ima dana kriterijska funkcija optimalno vrednost?

V kombinatoriko sodijo še:

- naloge o lastnostih: Katere značilne lastnosti imajo objekti, ki zadoščajo danim pogojem?

ki pa v večini primerov preraščajo v samostojna področja diskretne matematike kot so teorije: grafov, urejenih množic, matroidov, kodiranja, končnih geometrij, ...
Poglejmo si nekaj primerov.

Zgled: Prevozi

Potnik lahko pride iz kraja A v kraj B z avtobusom, vlakom ali taksijem. Iz kraja B pa lahko nadaljuje pot v kraj C z letalom ali ladjo. Kako in na koliko načinov lahko pride potnik iz kraja A v kraj C?
Konfiguracije, ki ustrezajo nalogi so urejeni pari: (vozilo AB, vozilo BC). Obstaja 6 možnosti:

(vlak, letalo)	(avtobus, letalo)	(taksi, letalo)
(vlak, ladja)	(avtobus, ladja)	(taksi, ladja)

ki sestavljajo množico Vozila AB $\times$ Vozila BC.
Če bi imeli na voljo še dodatne podatke, bi lahko odgovorili tudi na vprašanji:

- kako priti najhitreje iz kraja A v kraj C?
- kako priti najceneje iz kraja A v kraj C?

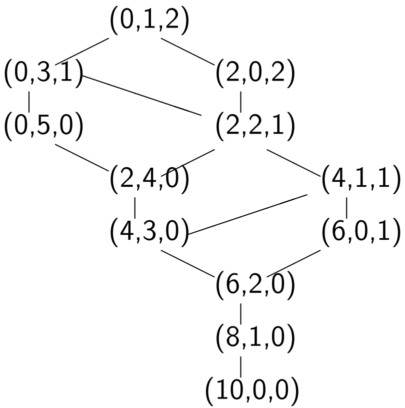
Zgled: Izplačila

Kako in na koliko načinov lahko izplačamo znesek 50 donarjev s kovanci za 5, 10 in 20 donarjev?
Konfiguracije so neurejene (vrstni red ni pomemben) skupine kovancev za 5, 10 in 20 donarjev z vsoto 50. Predstavimo jih lahko kot urejene trojice $(p, q, r) \in \mathbb{N}^3$, za katere velja $5p + 10q + 20r = 50$. Naštejmo jih (glejte sliko na naslednji prosojnici) in še preštejmo. 12 jih je!
Dodatni zahtevi, da pri izplačilu uporabimo čim manj kovancev, zadošča rešitev $10 + 20 + 20$. Vse tri vrste kovancev pa smo uporabili le v rešitvah 6 in 8.
Izplačila lahko uredimo kot je prikazano na prosojnici 6.
Dobljena struktura je mreža.

Zgled: Izplačila – naštetje

1.	$5+5+5+5+5+5+5+5+5+5$	$(10,0,0)$
2.	$5+5+5+5+5+5+5+5+10$	$(8,1,0)$
3.	$5+5+5+5+5+5+10+10$	$(6,2,0)$
4.	$5+5+5+5+5+5+20$	$(6,0,1)$
5.	$5+5+5+5+10+10+10$	$(4,3,0)$
6.	$5+5+5+5+10+20$	$(4,1,1)$
7.	$5+5+10+10+10+10$	$(2,4,0)$
8.	$5+5+10+10+20$	$(2,2,1)$
9.	$10+10+10+10+10$	$(0,5,0)$
10.	$10+10+10+20$	$(0,3,1)$
11.	$5+5+20+20$	$(2,0,2)$
12.	$10+20+20$	$(0,1,2)$

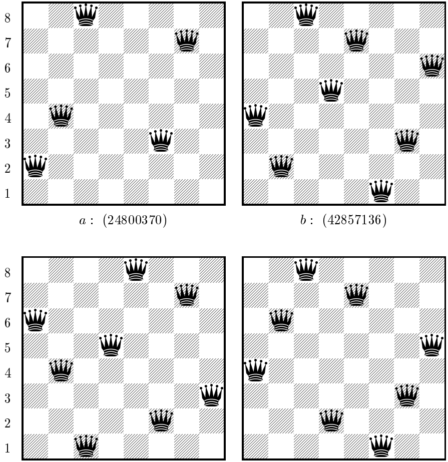
Zgled: Izplačila – mreža



Zgled: Kraljice na šahovnici

Kako in na koliko načinov lahko postavimo na šahovnico kraljice, tako da se ne bodo medsebojno ogrožale in bo vsako polje šahovnice napadeno? Najmanj koliko kraljic je za to potrebnih in največ koliko jih lahko postavimo?
Konfiguracije so sedaj razmestitve kraljic na šahovnici, ki zadoščajo zahtevam naloge.
Naloga o kraljicah sodi v železni repertoar zabavne matematike. Iz knjig izvemo, da potrebujemo vsaj 5 kraljic, na šahovnico pa lahko postavimo največ 8 kraljic. Ker lahko v vsako vrsto postavimo največ eno kraljico, jih več kot 8 očitno ne moremo postaviti.
Poskusite sami!

Zgled: Kraljice na šahovnici – 4 rešitve



a : (24800370) b : (42857136)

c : (64158273) d : (46827135)

Zgled: Kraljice na šahovnici – lastnosti

Iz vsake rešitve lahko z vrtenjem za 90° in z zrcaljenjem dobimo še 7 rešitev. Tako dobimo iz rešitve b, če jo zavrtimo za 180° in prezrcalimo, rešitev c. Vse tako dobljene rešitve niso vedno različne, kar vidimo, če rešitev d zavrtimo za 180°. Vrtenje in zrcaljenje nas pripeljeta do pojma osnovnih rešitev. Izkaže se, da lahko 8 kraljic postavimo na šahovnico na 92 načinov, ki dajo naslednjih 12 osnovnih rešitev.

(72631485)	(61528374)	(58417263)	(35841726)
(46152837)	(57263148)	(16837425)	(57263184)
(48157263)	(51468273)	(42751863)	(35281746)

Rešitev b (oziroma c) je edina osnovna rešitev kjer nobena trojica izmed 8 kraljic ne leži na isti “premici”.

Ozadje preštevanja osnovnih rešitev razkriva Polyajev izrek, ki je lep primer uporabe teorije grup v kombinatoriki. Kako se naloge lotimo z računalnikom, je opisano v knjigi N. Wirth: *Računalniško programiranje*.

Osnovna pravila preštevanja

Pri preštevanju množic kombinatoričnih objektov ponavadi opremo svoje razmišljanje na tako imenovana pravila preštevanja, ki so v bistvu iz teorije množic "izposojeni" izreki o lastnostih moži končnih množic. Poglejmo jih.

Pravilo enakosti

Če lahko iz objektov (končnih) množic A in B sestavimo pare $(a, b), a \in A, b \in B$, tako da vsak objekt nastopa v natanko enem paru, sestavlja množici A in B enako število objektov.

ali v jeziku teorije množic:

Če obstaja bijekcija $\varphi : A \to B$, potem je $|A| = |B|$.

Pravilo enakosti nam omogoča, da na nalogo pogledamo "s prave strani" in jo morda prevedemo na enakovredno nalogo, katere rešitev že poznamo.
Pravilo enakosti lahko uporabimo tudi pri naštevanju. Recimo, da znamo generirati objekte množice A in, da preslikava φ iz objekta $a \in A$ zgradi objekt $b \in B$. če je φ bijekcija, lahko tedaj zgeneriramo tudi vse objekte množice B. Vse objekte iz B dobimo že, če je φ surjekcija, vendar lahko pri tem posamezni objekt dobimo večkrat.

Zgled: Vse podmnožice dane množice

Naj bo X končna množica z n elementi

$$X = \{x_1, x_2, x_3, \ldots, x_n\}$$

Vsaki njeni podmnožici $Y \subseteq X$ lahko povratno enolično priredimo karakteristični vektor $\mathbf{k} = [k_1, k_2, \ldots, k_n]$ določen s predpisom

$$k_i = \begin{cases} 0 & x_i \notin Y \\ 1 & x_i \in Y \end{cases}$$

temu pa, če na vektor pogledamo kot na zapis števila v dvojiškem sestavu, število

$$b(\mathbf{k}) = \sum_{i=1}^{n} k_i 2^{n-i}$$

med 0 in $2^n - 1$ vključno.

Torej je število vseh podmnožic množice X enako 2^n.

Zgled: Vse podmnožice dane množice ...

Poglejmo si stvar še na primeru $X = \{a, b, c\}$

Y	$\mathbf{k}$	$b(\mathbf{k})$
$\emptyset$	$[0, 0, 0]$	0
$\{c\}$	$[0, 0, 1]$	1
$\{b\}$	$[0, 1, 0]$	2
$\{b, c\}$	$[0, 1, 1]$	3
$\{a\}$	$[1, 0, 0]$	4
$\{a, c\}$	$[1, 0, 1]$	5
$\{a, b\}$	$[1, 1, 0]$	6
$\{a, b, c\}$	$[1, 1, 1]$	7

Opisana zveza med Y in $b(\mathbf{k}(Y))$ nam omogoča tudi sestaviti postopek za generiranje vseh podmnožic množice X. Vsako število med 0 in $2^n - 1$ zapišemo v dvojiškem sestavu, iz tega zapisa pa sestavimo pripadajoči karakteristični vektor oziroma množico Y. Druga možnost je, da karakteristični vektor obravnavamo kot dvojiško številko. Začnemo z vektorjem, ki ga sestavljajo same 0 in mu prištevamo 1, dokler ne dobimo vektor iz samih 1.

Zgled: Vse podmnožice dane množice ...

Prva zamisel je uporabljena v postopku v Pythonu:

```
from itertools import compress
def SubSets(k):
   n = 2**k
   for i in range(n):
      yield list(map(int,list(bin(n+i)[3:])))

for s in SubSets(4):
   print(s,set(compress('ABCD',s)))
```

```
[0, 0, 0, 0] set()
[0, 0, 0, 1] {'D'}
[0, 0, 1, 0] {'C'}
[0, 0, 1, 1] {'C', 'D'}
[0, 1, 0, 0] {'B'}
[0, 1, 0, 1] {'B', 'D'}
[0, 1, 1, 0] {'C', 'B'}
[0, 1, 1, 1] {'C', 'B', 'D'}
[1, 0, 0, 0] {'A'}
[1, 0, 0, 1] {'A', 'D'}
[1, 0, 1, 0] {'A', 'C'}
[1, 0, 1, 1] {'A', 'C', 'D'}
[1, 1, 0, 0] {'A', 'B'}
[1, 1, 0, 1] {'A', 'B', 'D'}
[1, 1, 1, 0] {'A', 'C', 'B'}
[1, 1, 1, 1] {'A', 'C', 'B', 'D'}
```

Zgled: Oklepajni izrazi

Izraz, ki ga dobimo, če v pravilno zgrajenem aritmetičnem izrazu zradiramo vse razen oklepajev, imenujemo oklepajni izraz. Pokazati je mogoče, da je izraz sestavljen iz oklepajev oklepajni izraz natanko takrat, ko:

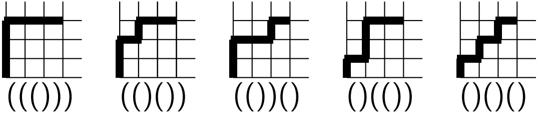
- vsebuje enako število predklepajev in zaklepajev; in
- v nobenem njegovem začetku število zaklepajev ne presega števila predklepajev.

Postavimo se v izhodišče ravninskega koordinatnega sistema in pripišimo oklepajem naslednji pomen:

- predklepaj: pojdi za 1 v smeri osi y;
- zaklepaj: pojdi za 1 v smeri osi x.

Zgled: Oklepajni izrazi in poti po mreži

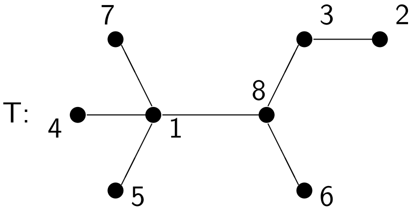
Tedaj očitno vsakemu oklepajnemu izrazu dolžine 2n ustreza naraščajoča pot po enotski mreži iz izhodišča $(0,0)$ v točko (n,n), ki nikoli ne zaide pod diagonalo $y = x$; in obratno.



Seveda si s tem spoznanjem zaenkrat še ne znamo pomagati.

Zgled: Prüferjeva zveza

Spoznali bomo bijekcijo med označenimi drevesi na n točkah in $n-2$ – terkami števil iz $1..n$.
Drevo $\mathcal{T} = (V, E, s)$, $|V| = n$ je označeno, če je $s : V \to 1..n$ bijekcija.



Zgled: Prüferjeva zveza – postopek

Vsakemu označenemu drevesu $\mathcal{T}$ priredimo $n-2$ –terico

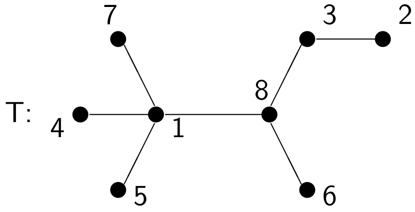
$$\mathbf{a}(\mathcal{T}) = [a_1, a_2, \ldots, a_{n-2}]$$

takole. Označimo z $\mathcal{L}$ množico listov drevesa in naj bo $v = \max\{t : t \in \mathcal{L}\}$. Kot vemo ima vsak graf na vsaj dveh točkah vsaj dva lista. Zato v vedno obstaja. Tedaj je a_1 sosed točke v. Odstranimo povezavo $(a_1 : v)$. Zopet dobimo drevo, na katerem postopek ponovimo. Ponavljamo, dokler ne dobimo drevesa na dveh točkah. Natančneje

$\mathcal{L} = \{t : t \text{ je list drevesa } \mathcal{T}\}$;
for $i := 1$ **to** $n-2$ **do begin**
  $v := \max\{t \in \mathcal{L}\}$; $\mathcal{L} := \mathcal{L} \setminus \{v\}$;
  $a_i : (a_i, v) \in \mathcal{T}$; $\mathcal{T} := \mathcal{T} \setminus (a_i, v)$;
  if a_i je list v $\mathcal{T}$ **then** $\mathcal{L} := \mathcal{L} \cup \{a_i\}$;
end;

Zgled: Prüferjeva zveza – kleščenje drevesa

Za drevo na prosojnici 18 dobimo šesterico:



$$\mathbf{a}(\mathcal{T}) = [1, 8, 1, 1, 3, 8]$$

Pokažimo še, da tudi vsaki $n-2$ –terki $\mathbf{a} = (a_k)_{k=1}^{n-2}$ ustreza natanko določeno drevo.

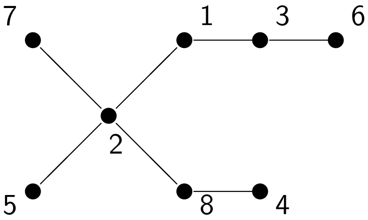
Zgled: Prüferjeva zveza – obratni postopek

Stopnja posamezne točke je za 1 povečano število pojavitev točke v $n-2$ –terici. Listi so natanko tiste točke, ki ne nastopajo v $n-2$ –terici.
Naj bo $v = \max\{t : t \in \mathcal{L}\}$. Dodajmo povezavo $(v : a_1)$. Izločimo točko v iz $\mathcal{L}$ in a_1 iz $\mathbf{a}$. Če a_1 ni v preostanku $\mathbf{a}$-ja, dodamo a_1 v $\mathcal{L}$ ... Na koncu, če je $a_{n-2} \neq 1$ povežemo še a_{n-2} z 1, sicer pa 1 z 2.

$\mathcal{L} = 1..n \setminus \mathrm{Set}(\mathbf{a})$; $\mathcal{T} := \emptyset$;
for $i := 1$ **to** $n-2$ **do begin**
  $v := \max\{t \in \mathcal{L}\}$; $\mathcal{L} := \mathcal{L} \setminus \{v\}$;
  $\mathcal{T} := \mathcal{T} \cup (a_i, v)$;
  if $a_i \notin (a_k)_{k=i+1}^{n-2}$ **then** $\mathcal{L} := \mathcal{L} \cup \{a_i\}$;
end;
$\mathcal{T} := \mathcal{T} \cup (u, v),\ u, v \in \mathcal{L}$;

Zgled: Prüferjeva zveza – obnova

Na primer, šesterici $\mathbf{a} = [2, 3, 2, 8, 2, 1]$ ustreza drevo na sliki:



Pravilo vsote

Če lahko izberemo objekt množice A na m načinov in objekt množice B na n načinov, potem, če se izbori izkjučujejo, lahko izberemo ali objekt iz A ali objekt iz B na $m+n$ načinov.

Pravilo vsote izhaja iz enakosti $|A \cup B| = |A| + |B| - |A \cap B|$, ki v primeru $A \cap B = \emptyset$, dobi obliko

$$|A \cup B| = |A| + |B|$$

Posplošeno pravilo vsote

Pravilo lahko posplošimo na razbitja. Naj bo $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$ razbitje množice A. Torej velja

$$\bigcup_{X \in \mathcal{A}} X = A \qquad \text{in} \qquad i \neq j \iff A_i \cap A_j = \emptyset$$

Potem je

$$|\bigcup_{X \in \mathcal{A}} X| = \sum_{X \in \mathcal{A}} |X|$$

Pravilo vsote nam omogoča priti do rekurzivnih zvez med števili objektov. Pri naštevanju pa nam pove: če znamo zgenerirati vse množice, ki sestavljajo razbitje, znamo zgenerirati tudi osnovno množico.

Zgled: Izplačila

Poglejmo si ponovno nalogo o izplačilih. Na koliko načinov lahko izplačamo znesek n donarjev s kovanci za 5, 10 in 20 donarjev?
Ker kovancev iste vrednosti ne ločimo in tudi vrstni red kovancev ni pomemben, lahko izplačujemo vedno s kovancem največje vrednosti. Tako dobimo razbitje množice izplačil glede na največji uporabljeni kovanec.
Naj bo a kovanec za 20, b za 10 in c za 5 donarjev. Označimo s $\mathcal{S}(n,k)$ množico vseh različnih izplačil zneska n, pri čemer lahko uporabljamo le kovance, ki ne presegajo k. Naj $+$ pomeni izbiro med dvema možnostima in $\cdot$ izbiro obeh možnosti.

Zgled: Izplačila

Poglejmo najprej izplačila zneska 50 donarjev:

$$\mathcal{S}(50,a) = \left\{\begin{array}{l} a^2b \\ a^2c^2 \\ ab^3 \\ ab^2c^2 \\ abc^4 \\ ac^6 \\ b^5 \\ b^4c^2 \\ b^3c^4 \\ b^2c^6 \\ bc^8 \\ c^{10} \end{array}\right. = \begin{array}{ll} a\cdot\left\{\begin{array}{l} ab \\ ac^2 \\ b^3 \\ b^2c^2 \\ bc^4 \\ c^6 \end{array}\right. & = a\cdot\mathcal{S}(30,a) \\ b\cdot\left\{\begin{array}{l} b^4 \\ b^3c^2 \\ b^2c^4 \\ bc^6 \\ c^8 \end{array}\right. & = b\cdot\mathcal{S}(40,b) \\ c\cdot c^9 & = \mathcal{S}(50,c) \end{array}$$

Zgled: Izplačila

Potemtakem velja na splošno:

$$\mathcal{S}(n,a) = a \cdot \mathcal{S}(n-20,a) + b \cdot \mathcal{S}(n-10,b) + \mathcal{S}(n,c)$$

$$\mathcal{S}(n,b) = b \cdot \mathcal{S}(n-10,b) + \mathcal{S}(n,c)$$

in

$$\mathcal{S}(n,c) = \begin{cases} c^{n/5} & n \bmod 5 = 0 \\ 0 & \text{sicer} \end{cases}$$

Zgled: Izplačila

Te zveze nam omogočajo določiti naštetje vseh možnih izplačil:

$$
\begin{aligned}
\mathcal{S}(50, a) &= a \cdot \mathcal{S}(30, a) + b \cdot \mathcal{S}(40, b) + c^{10} \\
\mathcal{S}(30, a) &= a \cdot \mathcal{S}(10, a) + b \cdot \mathcal{S}(20, b) + c^{6} \\
\mathcal{S}(10, a) &= \mathcal{S}(10, b) \\
\mathcal{S}(40, b) &= b \cdot \mathcal{S}(30, b) + c^{8} \\
\mathcal{S}(30, b) &= b \cdot \mathcal{S}(20, b) + c^{6} \\
\mathcal{S}(20, b) &= b \cdot \mathcal{S}(10, b) + c^{4} \\
\mathcal{S}(10, b) &= b + c^{2}
\end{aligned}
$$

in naprej, če privzamemo distributivnost,

Zgled: Izplačila

$$
\begin{aligned}
\mathcal{S}(20, b) &= b^2 + bc^2 + c^4 \\
\mathcal{S}(30, b) &= b^3 + b^2c^2 + bc^4 + c^6 \\
\mathcal{S}(40, b) &= b^4 + b^3c^2 + b^2c^4 + bc^6 + c^8 \\
\mathcal{S}(30, a) &= ab + ac^2 + b^3 + b^2c^2 + bc^4 + c^6 \\
\mathcal{S}(50, a) &= a^2b + a^2c^2 + ab^3 + ab^2c^2 + abc^4 + ac^6 + \\
&\quad b^5 + b^4c^2 + b^3c^4 + b^2c^6 + bc^8 + c^{10}
\end{aligned}
$$

Zgled: Izplačila

Označimo $N(n,k) = |\mathcal{S}(n,k)|$. Tedaj dobimo iz zvez za $\mathcal{S}(n,k)$ naslednje zveze za $N(n,k)$:

$$N(n,c) = \begin{cases} 1 & n \bmod 5 = 0 \\ 0 & \text{sicer} \end{cases}$$

$$N(n,a) = \begin{cases} N(n-20,a) + N(n-10,b) + N(n-5,c) & n \geq 20 \\ N(n,b) & 0 < n < 20 \\ 1 & n = 0 \end{cases}$$

ter

$$N(n,b) = \begin{cases} N(n-10,b) + N(n-5,c) & n \geq 10 \\ N(n,c) & 0 < n < 10 \\ 1 & n = 0 \end{cases}$$

Zgled: Izplačila

Določimo z uporabo teh zvez $N(50, a)$:

$$
\begin{aligned}
N(50, a) &= N(30, a) + N(40, b) + 1 \\
N(30, a) &= N(10, a) + N(20, b) + 1 \\
N(10, a) &= N(10, b) \\
N(40, b) &= N(30, b) + 1 \\
N(30, b) &= N(20, b) + 1 \\
N(20, b) &= N(10, b) + 1 \\
N(10, b) &= N(0, b) + 1 = 2
\end{aligned}
$$

in naprej

$$
\begin{array}{lll}
N(20, b) = 3, & N(30, b) = 4, & N(40, b) = 5, \\
N(10, a) = 2, & N(30, a) = 6, & N(50, a) = 12
\end{array}
$$

Do istega rezultata bi prišli, če bi v izraz za $\mathcal{S}(50, a)$ postavili $a = b = c = 1$.

Pravilo produkta

Če lahko izberemo objekt $a \in A$ na m načinov in neodvisno od tega objekt $b \in B$ na n načinov, potem lahko urejeni par objektov (a, b) izberemo na $n.m$ načinov.

Pravilo produkta zapišemo v teoriji množic takole:

$$|A \times B| = |A|.|B|$$

Zgled: Štetje števil in štetje dreves

Koliko je dvomestnih števil v desetiškem sestavu? Vsaki dvomestni številki $10a + b$ ustreza povratno enolično urejen par (a, b), $a, b \in 0..9$. Torej je po pravilu enakosti in pravilu produkta število vseh dvomestnih števil enako $10 \cdot 10 = 100$.
Pri Prüferjevi zvezi smo pokazali, da je število označenih dreves na n točkah enako številu vseh $n-2$–teric sestavljenih iz števil iz $1..n$. Po pravilu produkta je teh n^{n-2}.
V primeru 1.8 sta bili izbiri obeh števk neodvisni – izbira prve števke v ničemer ne omejuje izbiro druge števke. Kaj pa, če so izbire povezane? Tedaj lahko včasih uporabimo posplošeno pravilo produkta.

Posplošeno pravilo produkta

Če lahko izberemo objekt $a \in A$ na m načinov, in če lahko po vsaki taki izbiri izberemo objekt $b \in B$ na n načinov, potem lahko urejeni par objektov (a, b) izberemo na $m \cdot n$ načinov.

V veljavnost tega pravila se prepričamo takole:

$$|\cup_{a \in A}(\{a\} \times B_a)| = \sum_{a \in A} |\{a\}| \cdot |B_a| = n \sum_{a \in A} 1 = m \cdot n$$

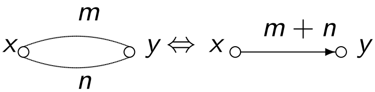
DiMa 1
Pravila

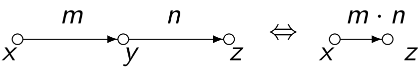
V. Batagelj

Zgled: Štetje števil ponovno

Koliko je dvomestnih števil v desetiškem sestavu, ki so sestavljene iz dveh različnih števk?
Tokrat lahko za prvo števko izberemo katerokoli izmed desetih števk; za drugo števko pa katerokoli, razen izbrane – torej eno izmed devetih. Zato je število vseh izbir enako $10 \cdot 9 = 90$.

Pravilo vsote in pravilo produkta ter grafi

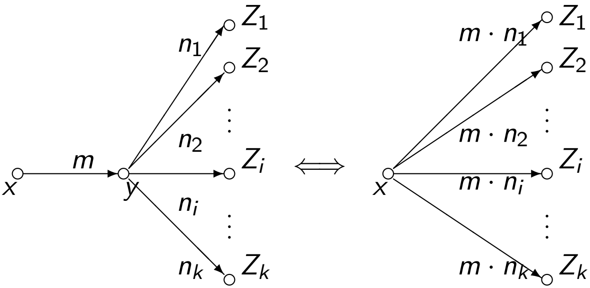




Pravili vsote in produkta si lahko ponazorimo tudi kot transformaciji grafa števila sprehodov med točkami (glej sliko).

Posplošeno pravilo produkta in grafi

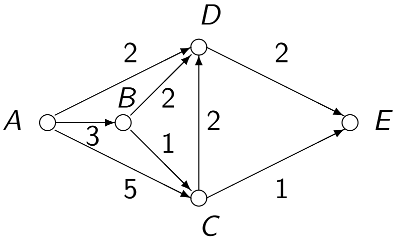
Prav nam bo prišla še posplošenega oblika pravila produkta prikazana na sliki; oziroma njej zrcalna oblika.



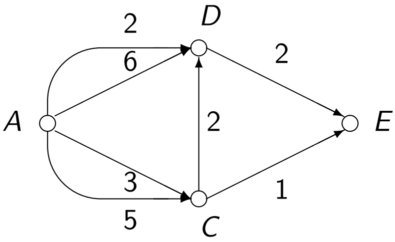
Poglejmo si to še na preprostem primeru.

Štetje poti

Števila na povezavah grafa na sliki povedo na koliko načinov lahko pridemo iz začetka povezave v njen konec. Na koliko načinov lahko pridemo iz točke A v točko E?

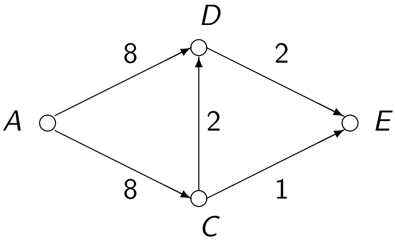


Štetje poti



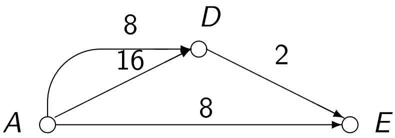
Do odgovora pridemo z uporabo pravil. Najprej uporabimo posplošeno pravilo na točki B. Vzporedne povezave združimo po pravilu vsote.

Štetje poti

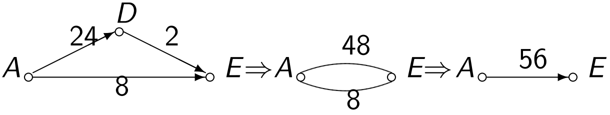


Zopet uporabimo posplošeno pravilo produkta, tokrat na točki C.

Štetje poti



Združimo še povezavi po pravilu vsote, uporabimo pravilo produkta in še enkrat pravilo vsote pa imamo iskani odgovor.



Računovodsko pravilo

Naj bo dana relacija $R \subseteq A \times B$. Označimo:

$$R(x) = \{y : xRy\} \qquad \text{in} \qquad R^{-1}(y) = \{x : xRy\}$$

Družini množic $\{\{x\} \times R(x) : x \in A\} \setminus \{\emptyset\}$ in $\{R^{-1}(y) \times \{y\} : y \in B\} \setminus \{\emptyset\}$ sta razbitji relacije R. Zato velja po pravilu vsote

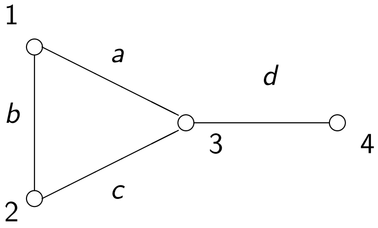
$$\sum_{x \in A} |R(x)| = \sum_{y \in B} |R^{-1}(y)| = |R|$$

Računovodsko pravilo: grafi

Naj bo $G = (V, E)$ enostaven, neusmerjen graf brez zank.
Relacija $R \subseteq V \times E$ naj povezuje povezave z njihovimi krajišči:

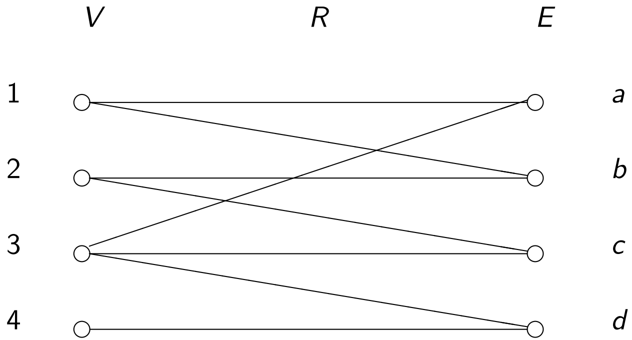
$$(v, e) \in R \Leftrightarrow \exists u \in V : e(v : u)$$

Tako na primer grafu na sliki



pripada relacija na naslednji sliki.

Računovodsko pravilo: grafi



Ustrezne množice $R(v)$ in $R^{-1}(e)$ so tedaj

$$R(v) = \{e : \exists u \in V : e(v : u)\}$$

in

$$R^{-1}(e) = \{v : \exists u \in V : e(v : u)\}$$

Računovodsko pravilo: grafi

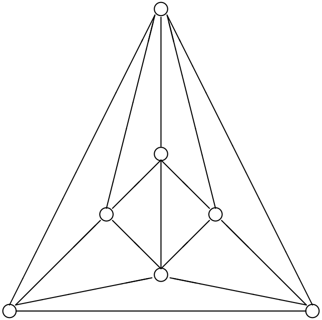
za naš graf sta na primer:

$$R(3) = \{a, b, c\} \qquad \text{in} \qquad R^{-1}(a) = \{1, 3\}$$

Poglejmo še moči teh množic: $|R(v)| = d(v)$ je stopnja točke v; in $|R^{-1}(e)| = 2$. Pravilo dveh vsot nam da znano enakost

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

Računovodsko pravilo: Triangulacije ravnine



Relacija $R \subseteq E \times F$ naj povezuje trikotnike z njihovimi stranicami. Vsakemu trikotniku $T \in F$ pripadajo tri stranice $R^{-1}(T) = \{e_1, e_2, e_3\}$; in vsaka stranica ločuje dva trikotnika $R(e) = \{T_1, T_2\}$.

Torej je $|R(e)| = 2$ in $|R^{-1}(T)| = 3$. Po pravilu dveh vsot dobimo znano enakost $2|E| = 3|F|$.

Dirichletovo načelo

Kadar je potrebno dokazati obstoj konfiguracije z želenimi lastnostmi, se pogosto obnese naslednje načelo:

> *Če kakorkoli porazdelimo $n+1$ elementov v n celic, sta v vsaj eni celici vsaj dva elementa.*

To načelo je dobilo ime po nemškem matematiku Dirichletu (1805-1859). V teoriji množic mu je enakovredna trditev:

> *Naj bosta A in B končni množici in $f : A \to B$ injektivna preslikava, tedaj je $|A| \leq |B|$.*

Dirichletovo načelo

Pokažimo, da v zaporedju n naravnih števil

$$a_1,\ a_2,\ a_3,\ \ldots,\ a_{n-1},\ a_n$$

ki niso deljiva z n, vselej obstaja strnjeno podzaporedje, katerega vsota je deljiva z n.
Sestavimo delne vsote:

$$s_i = a_1 + a_2 + \ldots + a_i, \quad i = 1, 2, \ldots, n$$

in si oglejmo njihove ostanke pri deljenju z n. Če je kakšen enak 0, smo iskano podzaporedje našli. Sicer morata po Dirichletovem načelu (n ostankov, $n-1$ različnih) vsaj dve delni vsoti s_p in s_q, $p < q$ dati enak ostanek. Tedaj pa je vsota

$$s_q - s_p = a_{p+1} + a_{p+2} + \ldots + a_q$$

deljiva z n.

Lastnosti konfiguracij

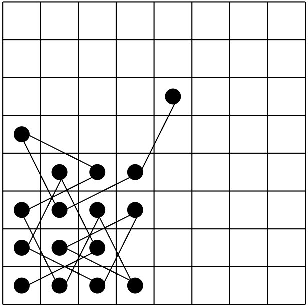
Večkrat je množica konfiguracij določena s pogoji, ki naj jih konfiguracije izpolnjujejo. Včasih so pogoji precej zapleteni in ni lahko ugotoviti ali sploh obstaja kaka konfiguracija, ki jim zadošča. Pri odgovoru na to vprašanje nam večkrat pride prav naslednje pravilo:

> *Naj imajo vse konfiguracije iz dane množice A lastnost P in naj konfiguracija X nima lastnosti P, tedaj X ne pripada množici A.*

Pri tem je treba opozoriti, da lastnost P ne določa nujno množice A, temveč samo izhaja iz njenega opisa. Poglejmo si stvar na dveh primerih.

Lastnosti konfiguracij: Konjiček na šahovnici

Ali lahko šahovski konjiček obišče vsa polja šahovnice, vsako natanko enkrat, tako da začne na polju v levem spodnjem vogalu in konča na polju v desnem zgornjem vogalu?



Lastnosti konfiguracij: Konjiček na šahovnici

Konfiguracije, ki ustrezajo sprehodom konjička po šahovnici, lahko predstavimo kot zaporedja oznak polj, na katera je konjiček skočil:

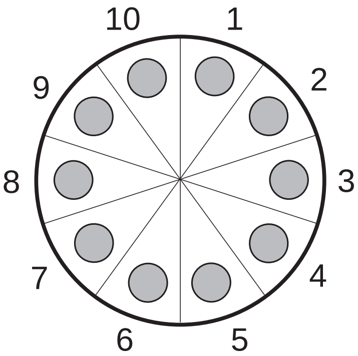
$$S = (p_1, p_2, p_3, \dots)$$

Upoštevajmo, da šahovski konjiček skače z belega na črno in s črnega na belo polje, pa vidimo, da imajo vsa zaporedja, ki opisujejo sprehajanje šahovskega konjička po šahovnici lastnost:

> $P(S) \equiv$ *polja, ki ustrezajo oznakam iz zaporedja S so izmenoma črna-bela.*

Recimo, da obstaja rešitev R naše naloge. Tedaj ima zaporedje R 64 členov in je potemtakem po lastnosti P polje, na katerem konjiček sprehod konča, druge barve kot je polje, na katerem je sprehod začel. To pa je v protislovju z zahtevo naloge – polji v levem spodnjem in desnem zgornjem vogalu šahovnice sta iste barve. Torej naloga ni rešljiva.

Lastnosti konfiguracij: Kovanci



Krog je razdeljen na 10 delov (izsekov). Na vsak del je postavljen po en kovanec. V posamezni potezi lahko izberemo dva kovanca in ju prestavimo vsakega na enega izmed njemu sosednjih delov. Ali lahko z več zaporednimi potezami dosežemo, da bodo vsi kovanci zbrani na istem izseku?

Lastnosti konfiguracij: Kovanci

Oštevilčimo izseke z zaporednimi številkami od 1 do 10. Naj bo vrednost razporeditve kovancev po krogu enaka vsoti številk izsekov, ki pripadajo posameznim kovancem. Torej je vrednost začetne razporeditve enaka $1 + 2 + 3 + \ldots + 9 + 10 = 55$; vrednost končne pa bi bila $10.n$, kjer je n številka izseka, na katerem so zbrani vsi kovanci.
Poglejmo sedaj, kako se vrednost razporeditve spremeni pri posamezni potezi. Pri prestavitvi kovanca na sosednji izsek se vrednost zmanjša / poveča za 1 ali 9. Torej se pri potezi vrednost spremeni za sodo število ($=$ vsota dveh lihih števil). To pomeni, da lahko iz začetne razporeditve dobimo z dovoljenimi potezami le razporeditve lihe vrednosti. Ker je razporeditev, ki jo zahteva naloga, soda, naloga nima rešitve.