

Izrada grafike koristeći TikZ paket

Računalne vještine

V. Salamon, K.Pavlović, D.Silić

- Vektorska grafika
- PGF/TikZ
- TikZ
- Korištenje TikZ-a
- Dijagrami toka
- Definicija okruženja
- Korištene naredbe
- Struktura stabla
- Grafovi

- Koristi dvodimenzionalne poligone za reprodukciju slika u računalnoj grafici
- Svaka točka ima određenu poziciju na x i y osi ravnine te određuje put staze
- Svakoj stazi je moguće dodijeliti različite attribute poput: boje, oblika, krivulje, debljine i punjenja
- Primjer: razmatramo krug radijusa r ; podatci koji su potrebni računarskom programu za iscrtavanje tog kruga su:
 - 1 radijus r
 - 2 koordinatna pozicija središnje točke kruga
 - 3 stil i boja linije
 - 4 stil i boja punjenja objekta

Prednosti vektorske grafike

- Prednost vektorske grafike naprema rasterskoj su:
 - 1 manja količina informacija omogućava manju veličinu datoteke
 - 2 sve su informacije zapamćene i mogu se kasnije mijenjati; to znači da kasnije možemo izmjenjivati svojstva slike bez smanjenja kvalitete crteža kao kod rasterkse slike

- TikZ - najsloženiji i najmoćniji alat za stvaranje grafičkih elemenata u LaTeX-u
- PGF/TikZ - dva jezika koji rade u paru, služe nam za izradu vektorske grafike pomoću geometrijskog odnosno algebarskog opisa
- PGF (Prijenosni grafički format”) - osnovni sloj koji pruža korisniku osnovne naredbe za izradu grafike
- TikZ - sučelje sa posebnom sintaksom koja olakšava uporabu PGF-a
- PGF je jezik niže razine, dok je TikZ skup makronaredbi na višoj razini koje koriste PGF

- TikZ ist kein Zeichenprogramm” (TikZ nije program za crtanje)
- Dobivamo sve prednosti korištenja TeX-a za grafiku poput:
 - ➊ brzog stvaranja jednostavne grafike
 - ➋ korištenje makronaredbi
 - ➌ precizno pozicioniranje
- Takoer i nedostatke:
 - ➊ strma krivulja učenja
 - ➋ nema WYSIWYG pristupa rada
 - ➌ male promjene zahtijevaju dugo vrijeme ponovnog prevoenja

Usporedba sa ostalim grafičkim paketima

- Usporedba TikZ-a u odnosu na druge pakete:
 - ▶ standardna LaTeX-ova picture okolina nam omogućava izradu jednostavne grafike, ali ništa više od toga - rezultat portabilnosti picture okoline. Ne radi s pdftexom niti s bilo kojim drugim upravljačkim programom koji ne proizvodi ništa drugo osim PostScript koda.
 - ▶ Xypic je nešto stariji paket za izradu grafike. Nešto teži za uporabu i općenito učenje jer je njegova sintaksa, moglo bi se reći, kriptirana.
 - ▶ Dratex paket je vrlo mali paket za izradu grafike. U usporedbi sa TikZ-om i ostalim paketima je vrlo malen što može, a i ne mora biti njegova prednost.
 - ▶ Metapost je program koji je vrlo moćna alternativa TikZ-u. Prije je bio zaseban program što je prouzročilo mnogo problema, no sada je ugrađen unutar LaTeX-a.
 - ▶ Xfig program je važna alternativa TikZ-u za korisnike koji ne žele "programirati" svoju grafiku kao što je to potrebno uz TikZ i ostale navedene pakete. Postoji program koji pretvara xfig grafiku u TikZ.

Dijagrami toka u Tikz-u

- Jedna od mnogih stvari za koje možemo upotrebljavati Tikz okruženje u latex ureivaču je svakako i izrada dijagrama toka.
- Dijagrame toka korištenjem Tikz-a izraujemo pomoću raznih elemenata predefiniranih unutar samoga okruženja te njegovih raznih biblioteka.

Dijagrami toka u Tikz-u

```
1 \documentclass{article}
2
3 \usepackage{tikz}
4 \pagenumbering{gobble}
5
6 \usetikzlibrary{shapes, arrows, positioning}
7
8 \begin{document}
9 \tikzstyle{pravokutnik} = [rectangle, text width=2cm, align=center, fill=green!10, draw=blue!75, minimum size= 2.5cm, thick]
10 \tikzstyle{pravokutnik2} = [rectangle, text width=2cm, align=center, fill=red!20, draw=black!75, minimum size= 2.5cm, ultra thick]
11 \tikzstyle{krug} = [circle, text width=1.5cm, align=center, fill=red!10, draw=black!75, minimum size= 2.5cm, very thick]
12 \tikzstyle{krug2} = [circle, text width=1.5cm, align=center, fill=green!10, draw=blue!75, minimum size= 2.5cm, thick]
13 \tikzstyle{romb} = [diamond, text width=1.6cm, align=center, fill=violet!10, draw=blue!75, minimum size= 2.5cm, semithick]
14 \begin{center}
15 \begin{tikzpicture}
16
17 \node[krug2](projekt){clan tima predlaže projekt};
18
19 \node[romb][below-of=projekt](diskutiranje){Tim odlucuje jeli dobra kvaliteta projekta};
20
21 \node[pravokutnik][below-of= diskutiranje](prihvacanje){Predlagatelj obrazlaže ideju te pise prijedlog};
22
23 \node[pravokutnik2][left-of= prihvacanje](odbacivanje){Tim odbacuje ideju};
24
25 \node[pravokutnik2][right-of= prihvacanje](dodatne informacije){predlagatelj vrsi dodatna istrazivanja};
26
27 \node[pravokutnik][below-of= prihvacanje](prijedlog){Tim pregledava prijedlog};
28
29 \node[romb][below-of= prijedlog](izbor prijedloga){Je li prijedlog projekta zadovoljio ocekivanja};
30
31 \node[pravokutnik2][right-of= izbor prijedloga](korekcija prijedloga){predlagatelj korigira projekt};
32
33 \node[krug][below-of= izbor prijedloga](kraj){kraj procesa predlaganja};
34
35 \draw[->] (projekt.south) -- (diskutiranje.north);
36
37 \draw[->] (diskutiranje.south) -- node[right] {$DAS$} (prihvacanje.north);
38
39 \draw[->] (diskutiranje.south) -- node[above] {$NES$} (odbacivanje.north);
40
41 \draw[->] (diskutiranje.south) -- node[right] {$INFORMACIJE$} (dodatne informacije.north);
42
43 \draw[->] (prihvacanje.south) -- (prijedlog.north);
44
45 \draw[->] (prijedlog.south) -- (izbor prijedloga.north);
46
47 \draw[->] (izbor prijedloga.south) -- node[right] {$DAS$} (kraj.north);
48
49 \draw[->] (izbor prijedloga.east) -- node[above] {$NES$} (korekcija prijedloga.west);
50
51 \draw[->] (korekcija prijedloga.north) .. controls +(up:10mm) and +(right:20mm) .. (prijedlog.east);
52
53 \draw[->] (odbacivanje.south) .. controls +(down:20mm) and +(left:30mm) .. (kraj.west);
54
55 \draw[->] (dodatne informacije.east) .. controls +(right:20mm) and +(right:40mm) .. (diskutiranje.east);
56 \end{tikzpicture}
57 \end{center}
58
59 \end{document}
```

Figure: Kod dijagrama toka

Dijagrami toka u Tikz-u

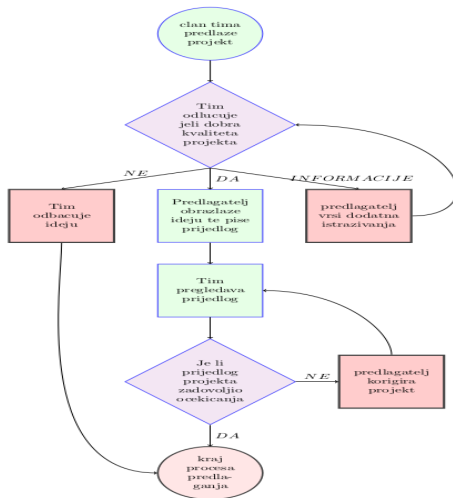


Figure: Izgled dijagrama toka

Preambula Tikz dokumenta

- Na početku izrade dijagrama u preambuli dokumenta potrebno je deklarirati kako se radi o paketu naredbi Tikz pomoću naredbe `\usepackage{tikz}`
- `\usetikzlibrary{shapes, arrows, positioning}`
- Neke od najčešće korištenih biblioteka su:
 - ➊ arrows – koristi se za crtanje različitih vrsta strelica
 - ➋ backgrounds – koristi se za definiranje različitih pozadina slika
 - ➌ chains – služi za lakše poravnavanje grafičkih elemenata u dokumentu
 - ➍ shapes – Jedna od najbitnijih biblioteka Tikz-a koja služi za definiranje različitih geometrijskih elemenata kao i raznih drugih oblika
- Biblioteke koje smo koristili pri izradi dijagrama toka u primjeru su:
 - ➊ shapes – upravo zbog mogućnosti crtanja različitih geometrijskih oblika koji su nam biti pogodni u izradi dijagrama.
 - ➋ arrows – kako bismo mogli povezati nacrtane objekte
 - ➌ positioning – kako bismo mogli slagati elemente u određene odnose jedan prema drugom

Preambula Tikz dokumenta

- `\tikzstyle{romb} = [diamond, text width=1.6cm, align=center, fill=violet!10, draw=blue!75, minimum size= 2.5cm, semithick]`
- Pomoću ove naredbe predefiniramo procesoru kako će izgledati pojedini objekt kojega ćemo kasnije moći jednostavno pozvati pomoću dodjeljene oznake
- Opcije koje smo u konkretnom primjeru koristili:
 - 1 diamond – označava kako se radi o obliku romba odnosno dijamanta
 - 2 text width=1.6cm – označava kako će širina teksta biti 1.6 cm
 - 3 align=center – tekst unutar objekta će biti centralno poravnat
 - 4 fill=violet!10 – označava kako će objekt biti ispunjen ljubičastom bojom kojoj će intenzitet biti 10
 - 5 draw=blue!75 – obrubljuje objekt sa crtom plave boje intenziteta 75
 - 6 minimum size=2.5cm – označava kako minimalna veličina tog elementa mora biti 2.5cm
 - 7 semithick – označava debljinu crte obruba elementa

Definicija okruženja

- Nakon što smo definirali objekte koje ćemo koristiti potrebno je definirati okruženje u kojem ćemo raditi.
- Dekaraciju okruženja vršimo naredbom
- Glavna naredba koju ćemo koristiti, a koja će nam poslužiti upravo zbog svoje svestranosti je naredba

```
\begin{tikzpicture}
```

```
\node
```

- Druga naredba koju ćemo koristiti kako bismo povezali napravljene objekte je naredba

```
\draw
```

- `\node[pravokutnik2][left=of prihvacanje](odbacivanje){Tim odbacuje ideju};`
- Node se tipično koristi kada se radi o nekom jednostavnom obliku koji u sebi može i ne mora sadržavati neki tekst, a u najjednostavnijem slučaju označava neki tekst koji je postavljen na određenu koordinatu.
- U slučaju iz primjera koristit ćemo node element kako bismo pozvali oblike koje smo prethodno deklarirali
- U prvoj uglatoj zagradi pišemo ime deklariranog elementa kojega želimo pozvati
- Drugu uglatu zagradu koristimo kako bismo definirali odnos između trenutnog i njemu susjednog elementa
- Oblik zagrada definiramo ime objekta
- U uglate zagrade stavljamo tekst koji želimo da se pojavi u objektu

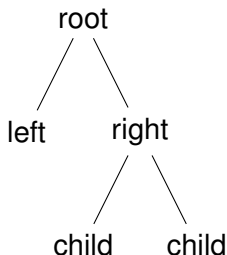
- `\draw[->] (projekt.south) -- (diskutiranje.north);`
- draw naredbu isto je tako moguće koristiti u mnogim primjenama
- U ovom primjeru koristit ćemo je kako bismo strelicama spojili stvorene objekte
- Uglata zagrada nam pokazuje o kakvoj se strelici radi
- U obliku zagradama specificiramo koje objekte povezujemo te kako ih povezujemo
- Dodatkom node elementa moguće je dodati tekst u strelicu
- naredbom controls možemo mijenjati izgled strelice

Zatvaranje okruženja

- Kada smo završili sa deklaracijom svih elemenata preostaje nam još samo zatvoriti Tikz okruženje
- Zatvaranje vršimo naredbom
`\end{tikzpicture}`

- Struktura stabla jedan je od najčešćih načina vizualizacije hijerarhijske strukture
- Struktura stabla omogućuje nam postavljanje elemenata u takozvani roditeljski odnos gdje svaki element roditelj(parent) može imati mnogo pod elemenata koji se nazivaju djeca(child)

Jednostavna stabla



- ```
\begin{tikzpicture}
 \node {root}
 child {node {left}}
 child {node {right}}
 child {node {child}}
 child {node {child}}
};
\end{tikzpicture}
```

- kao i u primjerima od prije osnovna naredba za pokretanje strukture bit će naredba

```
\node {root}
```

- kako bismo kreirali podelemente koristimo naredbu

```
child {node {left}}
```

svaki dodani element moguće je dalje dijeliti na manje dijelove tipkom `tab`

- I ovdje je moguće unaprijed definirati postavke prikazivanja stabla naredbom

```
\tikzstyle{level 1}=[sibling distance=8mm]
```

- Još jedan neizostavan dio TikZ paketa jest stvaranje grafova, posao koji inače obavljaju aplikacije kao što su gnuplot ili mathematica
- Prednost stvaranja grafova direktno uz pomoć TikZ-a jest ta što stvoreni graf poprima automatski isti izgled i font kao i ostatak dokumenta – stvar koju je teško ostvariti kada pokušamo napraviti graf uz pomoć drugih programa
- Takoer među ostale probleme kod korištenja drugih programa za stvaranje grafova jesu:
  - ▶ jednačbe dolaze drugačije ili se uopće ne prikazuju
  - ▶ linije su ili pre debele ili pre tanke
  - ▶ koordinatna mreža koja se automatski stvori ne izgleda onako kako mi želimo
  - ▶ strelice koje se nalaze na grafu nisu iste onima koje se koriste u ostatku dokumenta

- Postoje tri načina za stvaranje grafova uz pomoć Tikz – a:
  - 1 koristeći plot naredbu – ovo je najjosnovniji način i tjera nas da dosta stvari radimo ručno kao što su dodavanje osi ili strelica
  - 2 koristeći datavisualization naredbu – ona je mnogo moćnija od plot naredbe
  - 3 koristeći pgfplots paket – on je alternativa datavisualization naredbi

- Kada se stvara funkcija, koordinate te funkcije se automatski generiraju uz pomoć definirane matematičke jednadžbe
- Sintaksa je jednostavna: nakon što upišemo plot naredbu u zagradama definiramo varijablu

`\x`

te zatim slijedi matematička jednadžba koja sadrži tu varijablu

# Primjer stvaranja grafa uz pomoć plot naredbe

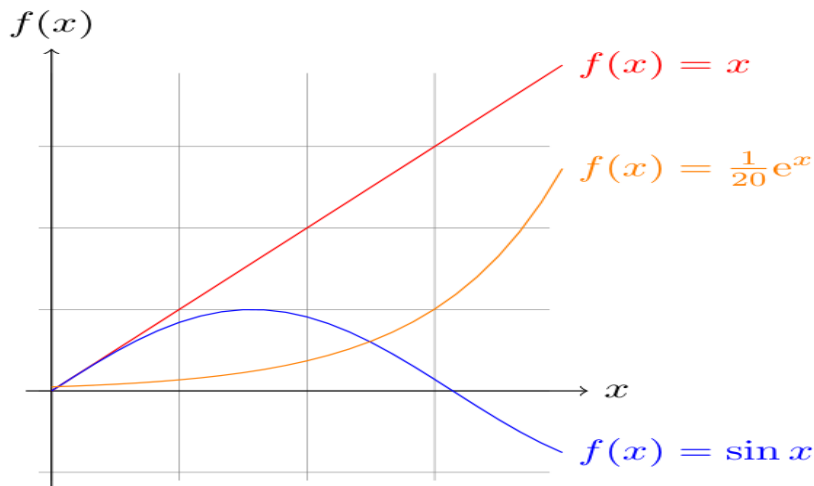


Figure: Izgled grafa 1

# Primjer stvaranja grafa uz pomoć plot naredbe

- Prije nego što počnemo moramo definirati tikzpicture te domenu
  - ▶ domena označava od kuda do kuda da se prikazuju funkcije po x – osi

```
\begin{tikzpicture}[domain=0:4]
\end{tikzpicture}
```

- Prvo stvaramo koordinatnu mrežu
  - ▶ koristimo sivu tanku liniju i to definiramo u zagradama odmah nakon što upišemo  
`\draw`  
naredbu
  - ▶ postavljamo da je početak u točki  $x = -0.1$  i  $y = -1.1$ , kažemo da stvaramo koordinatnu mrežu pa pišemo  
`grid`  
i na kraju definiramo krajnju točku  $x = 3.9$  i  $y = 3.9$

```
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
```



# Primjer stvaranja grafa uz pomoć plot naredbe

- Zatim stvaramo koordinatne ose

- ▶ kad stvaramo koordinatnu osu želimo da na kraju ima strelicu tako da to definiramo odmah nakon

```
\draw
```

naredbe u zagradama –

```
[->]
```

- ▶ definiramo točku u kojoj će biti početak crte  $(-0.2, 0)$  i onda stavljamo dvije crtice – i s tim kažemo da želimo spojiti dvije točke (stvoriti liniju) te slijede koordinate druge točke  $(4.2, 0)$
- ▶ svaka koordinatna osa ima i tekst na kojoj piše što ona predstavlja tako da ako želimo to ubaciti prvo moramo reći gdje da tekst bude smješten s obzirom na kraj funkcije / linije.. i to označavamo sa `node[pozicija]` i tek onda slijedi tekst, a zapisujemo ga u formatu `{ $\$$ tekst $\$$ }`

```
\draw[->] (-0.2,0) -- (4.2,0) node[right] { x };
```

```
\draw[->] (0,-1.2) -- (0,4.2) node[above] { $f(x)$ };
```

# Primjer stvaranja grafa uz pomoć plot naredbe

- I na kraju stvaramo funkcije
  - ▶ svaka funkcija će imati zasebnu boju i to definiramo odmah na početku u zagradama
  - ▶ za stvoriti funkciju pišemo plot, definiramo varijablu  $x$  u zagradama te stavljamo zarez i pravimo funkciju koju želimo da se prikaže
- npr. za funkciju  $\sin x$  upisujemo:

```
plot (\x,{sin(\x r)})
```

gdje  $r$  predstavlja da želimo pretvoriti kutove iz stupnjeva u radijane

```
\draw[color=red] plot (\x,\x) node[right] {$f(x) = x$};
```

```
\draw[color=blue] plot (\x,{sin(\x r)}) node[right] {$f(x) = \sin x$};
```

```
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right] {$f(x) = 0.05e^x$};
```

# Primjer stvaranja grafa uz pomoć plot naredbe

```
\begin{tikzpicture}[domain=0:4]
\draw[very thin,color=gray] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-0.2,0) -- (4.2,0) node[right] {x};
\draw[->] (0,-1.2) -- (0,4.2) node[above] {$f(x)$};
\draw[color=red] plot (\x,\x) node[right] {$f(x) = x$};
\draw[color=blue] plot (\x,{sin(\x r)}) node[right] {$f(x) = \sin x$};
\draw[color=orange] plot (\x,{0.05*exp(\x)}) node[right] {$f(x) = \frac{1}{20} \mathrm{e}^x$};
\end{tikzpicture}
```

Figure: Cijeli kod

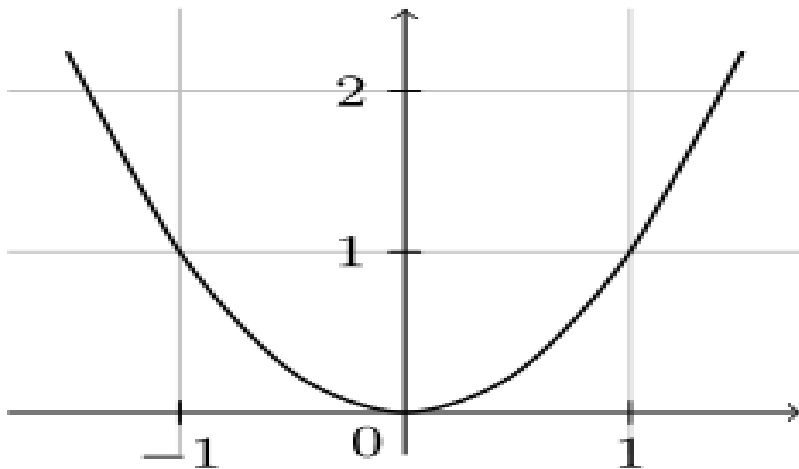
# Data Visualization naredba

- Kako bi mogli stvarati grafove uz pomoć dataisualization naredbe prvo moramo uključiti pripadajuću biblioteku:

```
\usetikzlibrary{datavisualization}
```

- Postoji par koraka koje moramo proći da bi dobili željeni graf:
  - ▶ na početku moramo definirati kakav stil grafa želimo (“school book plot” ili “scientific 2d plot” ili “scientific spherical plot”)
  - ▶ zatim upisujemo koordinate točaka uz pomoć data naredbe
  - ▶ dodatno možemo dodati još opcija koje nam daju više kontrole oko izgleda grafa. Možemo konfigurirati broj linija koordinatne mreže, poziciju tekstova, boje, fontovi. . .
- Postoje dva načina zapisa koordinatnih točaka:
  - 1 mi direktno upisujemo koje su točke i onda on ovisno o tim točkama napravi funkciju
  - 2 definiramo u kojem intervalu se nalazi te zatim definiramo funkciju

# Primjer stvaranja grafa uz pomoć datavizualization naredbe



# Korištenje datavisualization naredbe - 1. način

- kao i prije prvo definiramo tikzpicture:

```
\begin{tikzpicture}
\end{tikzpicture}
```

- na početku definiramo izgled grafa, linije i koordinatnu mrežu:

```
\datavisualization [
 school book axes,
 visualize as smooth line,
 all axes={grid}
]
```

- zatim unutar zagrada definiramo koordinate točaka

# Korištenje datavisualization naredbe - 1. način

```
\begin{tikzpicture}
\datavisualization [
 school book axes,
 visualize as smooth line,
 all axes={grid}
]
data {
x, y
-1.5, 2.25
-1, 1
-.5, .25
0, 0
.5, .25
1, 1
1.5, 2.25
};
\end{tikzpicture}
```

Figure: Cijeli kod

## Korištenje datavisualization naredbe - 2. način

- prvo na isti način definiramo tikzpicture i datavisualization
- nakon toga upišemo data i definiramo da je to funkcija  
[format=function]  
, a unutar zagrada je i pravimo
- pravimo varijablu x koja je iz intervala -1.5 do 1.5 i uzima iz tog intervala 7 točaka, što više točaka to će funkcija biti preciznija
- a nakon definiranja varijable pravimo funkciju func y i definiramo njenu vrijednost

```
data [format=function] {
var x : interval [-1.5:1.5] samples 7;
func y = \value x*\value x;
};
```



## Korištenje datavisualization naredbe - 2. način

```
\begin{tikzpicture}
\datavisualization [
 school book axes,
 visualize as smooth line,
 all axes={grid}
]
data [format=function] {
 var x : interval [-1.5:1.5] samples 7;
 func y = \value x*\value x;
};
\end{tikzpicture}
```

Figure: Cijeli kod

- [1] Sharelatex tutorial. [Online]. Dostupno na:  
[https://www.sharelatex.com/learn/TikZ\\_package](https://www.sharelatex.com/learn/TikZ_package)
- [2] Latex wiki. [Online]. Dostupno na:  
<https://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>
- [3] Latex tutorial. [Online]. Dostupno na:  
<https://www.latex-tutorial.com/tutorials/tikz/>
- [4] TikZ/PGF manual. [Online]. Dostupno na: <http://mirror.utexas.edu/ctan/graphics/pgf/base/doc/pgfmanual.pdf>