Krótkie wprowadzenie do pakietu TikZ

Łukasz Strąk

Uniwersytet Śląski
Instytut Informatyki
lukasz.strak@gmail.com

16-04-2015

Spis treści

Rozdział 1

Wprowadzanie

Pakiet TikZ udostępnia narzędzia do tworzenia obrazów osadzonych w kodzie LaTeX. Zawiera zestaw wysokopoziomowych komend do łatwego i szybkiego tworzenia rysunków wektorowych. Napisany kod jest transformowany do języka PGF (Portable Graphics Format), który odpowiada za rysowanie niskopoziomowe. Następnie w zależności od wyboru, możliwymi postaciami wyjściowymi jest: PDF, PostScript lub SVG. Wiele pakietów graficznych posiada możliwość zapisu obrazów do postaci TikZ (m. in. Inkscape w funkcji export). Jego możliwości są bardzo duże. Dokumentacja do PGF/TikZ w wersji 3.0.0.0 zawiera 1165 stron. Na stronie `http://www.texample.net/tikz/examples` można znaleźć wiele przykładów zastosowania pakietu na potrzeby różnych dziedzin nauki m.in. matematyki, fizyki, chemii czy informatyki. Pakiet (poprzez PGF) wspiera również częściowo grafikę 3d oraz animacje.

1.1 Uruchamianie środowiska

W pierwszej kolejności należy dodać pakiet do listy aktywnych modułów komendą: \usepackage{tikz}. Tworzenie środowiska TikZ w LaTeXwykonywane jest komendą "{tikzpicture}", którą osadzić można w środowisku "figure". Następnie wewnątrz należy umieścić instrukcje dla procesora graficznego.

1.2 Współrzędne

Pakiet wykorzystuje współrzędne z I ćwiartki (dodatnie, rosnące wartości). Lewy dolny róg rysunku ma współrzędne (0,0). Wartości mogą być również ujemne. Środek rysunku obliczany jest na podstawie skrajnych punktów. Przykładowo dla dwóch wierzchołków o współrzędnych kolejno (-2,-2) oraz (2,2), środek zostanie wyznaczony o współrzędnych (0,0).

1.3 Linie i punkty

Ogólna składania instrukcji rysującej jest następująca:

Kod TikZ 1: Ogólna składania instrukcji rysującej

```
\begin{tikzpicture}
    \draw[opcje] (współrzędne) obiekt (współrzędne/specyficzne dane);
\end{tikzpicture}
```

Jako pole "obiekt" można użyć następujących wartości (podstawowe):

a) "--" linia —,

b) "circle" koło ◯,

c) "rectangle" prostokąt ▭,

d) "grid" rysuje linie siatki ▦,

e) "arc" krawędź ◠.

Jeśli obiekt ma mieć dodatkowy styl np. ma to być linia i ma być zakończona jakimś grotem należy w "opcjach" podać jej styl. Dostępne są następujące wartości:

a) "<-" strzałka w lewo ⟵,

b) "->" strzałka w prawo ⟶,

c) "<->" strzałka w oba kierunki ⟷,

d) "|->" strzałka zakończona grotem i linią poprzeczną ⟼,

e) "help lines" pozwala narysować pomocnicze linie,

f) "dotted" kształt (linia, strzałka, obiekt) zamiast linii ciągłej będzie miał kropki ułożone w linii ⋯→,

f) "'dashed" kształt (linia, strzałka, obiekt) zamiast linii ciągłej będzie miał linie przerywane - - →.

Separatorami opcji są przecinki. Niektóre style np. groty są dedykowane dla konkretnych obiektów z kolei inne są ogólnodostępne.

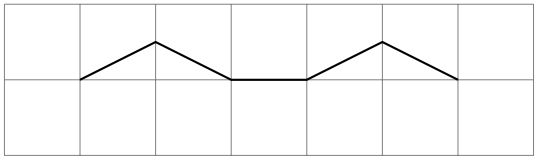
Linie można układać w nieskończone ciągi. Dodatkowo jeśli sekwencja będzie zakończona "–cycle" figura zostanie zamknięta (dodana zostanie linia łącząca ostatnie współrzędne z pierwszymi).

Przykładowo, aby narysować sekwencję linii należy wywołać następujące instrukcje:

Kod TikZ 2: Kod rysujący prostokąta

```
\begin{tikzpicture}
    \draw[help lines] (-1,-1) grid (6,1);
    \draw (0,0) – (1,0.5) – (2,0) – (3,0) – (4,0.5) – (5,0);
\end{tikzpicture}
```

co pozwoli otrzymać wynik:



Oprócz komendy "draw" dostępna jest również komenda "node" oznaczająca wierzchołek. Można jej używać zarówno jako etykiety do podpisu oraz do zdefiniowania punktów w przestrzeni, które następnie można połączyć. Poniższy kod zawiera uogólnioną komendę "node":

Kod TikZ 3: Ogólna składania instrukcji "node"

```
\begin{tikzpicture}
    \node[typ wierzchołka] (nazwa) at (x,y) {tekst};
\end{tikzpicture}
```

Rysowanie wierzchołka wygląda następująco:

Kod TikZ 4: Dwa połączone wierzchołki

```
\begin{tikzpicture}
    \node (p) at (0,0) {$p$};
    \node (k) at (2,0) {$k$};
    \draw[|->] (p)–(k);
\end{tikzpicture}
```

Wynikiem jest następujący obrazek:

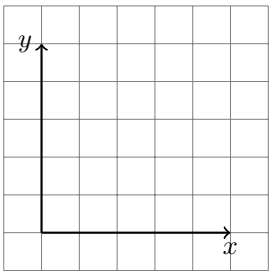
$$p \longmapsto k$$

Kolejny kod rysuje współrzędne kartezjańskie. Zastosowano tu również wierzchołek jako etykietę.

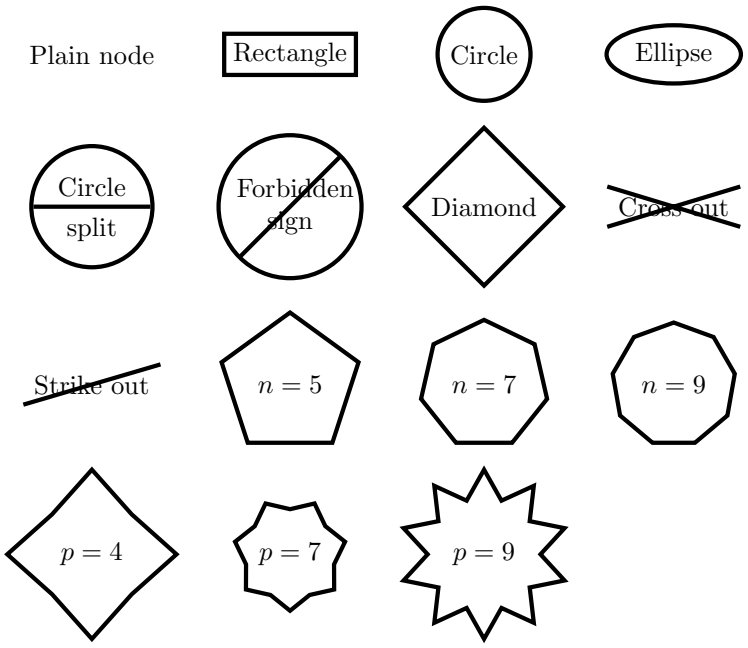
Kod TikZ 5: Układ kartezjański

```
\begin{tikzpicture}
    \draw[help lines]
      (-1,-1) grid (6,6);
    \draw[->] (0,0) – (5,0);
    \draw[->] (0,0) – (0,5);
    \node[left] at (0,5) {$y$};
    \node[below] at (5,0) {$x$};
\end{tikzpicture}
```

Wynikiem kodu jest obraz:



Dostępne są 4 rodzaje modyfikatorów pozycji: “below”, “above”, “left”, “right” oraz kombinacje dwuelementowe np. “below left”. Istnieje wiele typów wierzchołków, które zmieniają jego wygląd. Poniższy rysunek przedstawia najważniejsze jego możliwości:



Rysunek 1.1: Przykład różnego rodzaju wierzchołków

Niektóre z tych typów wymagają importu odpowiednich stylów poprzez umieszczenie \usetikzlibrary{shapes} (przed “\begin{document}”). Można również wykorzystać niektóre style z komendy “draw” jak np. “dotted” co spowoduje, że ramka będzie się składać z kropek.

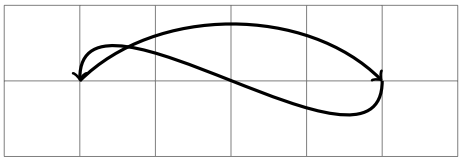
1.4 Krzywe

Krzywe można rysować na wiele sposobów. Podstawowe dwa sposoby są następujące:

Kod TikZ 6: Przykład tworzenia krzywych

```
\begin{tikzpicture}
    \draw [help lines] (-1,-1) grid (5,1);
    \draw (0,0) to [out=90,in=270] (4,0);
    \draw (0,0) .. controls (1,1) and (3,1) .. (4,0);
\end{tikzpicture}
```

Wynikiem poprzedniego kodu jest następujący obrazek:



Tekst bardzo łatwo dopasować do krzywych. Wystarczy do stylu dodać odpowiednie opcje:

Kod TikZ 7: Przykład rysowania tekstu dopasowanego do krzywej

```
\begin{tikzpicture}
    \draw [help lines] (-1,-1) grid (5,1);
    \draw [postaction={decorate,decoration=
      {text along path,text align=center, text={tekst tekst... }}}
       (0,0) .. controls (1,1) and (3,1) .. (4,0);
\end{tikzpicture}
```

Wynikiem wykonania powyższych instrukcji jest rysunek:

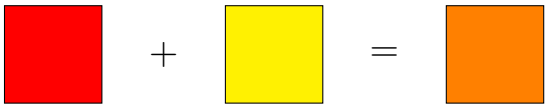
1.5 Kolory i szerokości

Oprócz standardowych nazw kolorów LaTeX, TikZ umożliwia również łatwe manipulowanie nimi. Przykładowo deklaracja "color=red!60" oznacza, że użyty kolor będzie jaśniejszy, niż czerwony (bardziej blady). Liczba 60 oznacza, że wynikowy kolor będzie się składał z 60% czerwonego i 40 białego (brak koloru oznacza przyjęcie wartości domyślnej jaką jest kolor biały). Kolejny kod łączy kolor czerwony i żółty:

Kod TikZ 8: Tworzenie koloru na podstawie mieszania

```
\begin{tikzpicture}
  \matrix[nodes={draw}, row sep=0.3cm,column sep=0.5cm]
  {
    \node[rect, fill=red] {}; &
    \node[texts] {\Large{+}}; &
    \node[rect, fill=yellow] {}; &
    \node[texts] {\Large{=}}; &
    \node[rect, fill=red!50!yellow] {}; &
  };
\end{tikzpicture}
```

Wynikiem jest następujący obraz:



Rysunek 1.2: Łączenie kolorów w pakiecie TikZ

Kolor definiuje się w stylu (zaczynającym się od nawiasu kwadratowego). Wielkości linii definiuje się na podstawie jej nazwy. Do wyboru są następujące wielkości:

a) ultra thick ——,

b) very thick ——,

c) thick ——,

d) thin ——,

e) very thin ——,

f) ultra thin ——.

Można również podać konkretną wartość np. "line width=10pt".

1.6 Skalowanie i obracanie

Zarówno skalowanie i jak i obracanie wykonuje się w parametrach opcjonalnych środowiska \tikzpicture. Służą do tego nazwy "scale" (wartość 1 to wartość domyślna) i "rotate" (w stopniach, domyślnie 0). Powyższe opcje można również użyć bezpośrednio na rysowanym obiekcie. Wykonuje się to na stylu (w nawiasach kwadratowych). Środowisko można również wykorzystywać jako "box" dla tekstu, który następnie można obracać.

1.7 Rysunek w tekście

Poprzednio rysunki były zagnieżdżone w środowisku "figure". Jednak nic nie stoi na przeszkodzie, aby osadzić je w tekście np.:

> Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus ⟶. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt.

Pakiet zawiera specjalną komendę do tworzeni krótkich instrukcji graficznych, szczególnie przydatnych w zagnieżdżaniu w tekście: \tikz "lista komend", zakończona średnikiem.

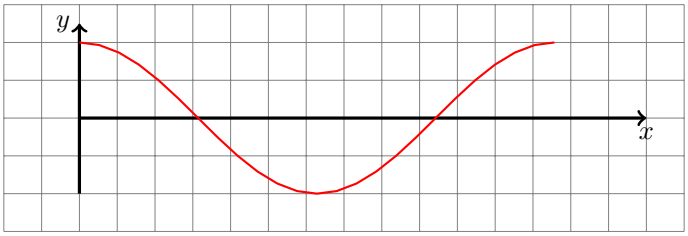
1.8 Współpraca z Gnuplot

W przypadku, gdy na hoście na którym kompilowany jest kod LaTeXi zainstalowany jest Gnuplot, wykresy można rysować bezpośrednio w TikZ. Poniższy kod przedstawia przykład takiego użycia:

Kod TikZ 9: Rysowanie wyniku programu Gnuplot w TikZ

```
\begin{tikzpicture}
    \draw[help lines,step=0.5cm] (-1,-1.5) grid (8,1.5);
    \draw[very thick,->] (0,0) – (7.5,0);
    \draw[very thick,->] (0,-1) – (0,1.25);
    \node[left] at (0,1.25) {$y$};
    \node[below] at (7.5,0) {$x$};
    \draw[red, thick, domain=0:2*pi] plot (\x, {cos(\x r)});
\end{tikzpicture}
```

Poprzedni kod powoduje powstanie poniższego rysunku:



1.9 Style

Style w pakiecie można grupować i nadawać im nazwy. Poniższy kod tworzy dwa takie style i używa ich do rysowania:

Kod TikZ 10: Ogólny algorytm wykorzystania zdefiniowanego stylu

```
\begin{tikzpicture}[nazwa1/.style={opcje}]
   \tikzstyle{nazwa2}=[opcje];
     ...
   \draw[nazwa1] ...;
   \draw[nazwa2] ...;
\end{tikzpicture}
```

Style można również łączyć i nadawać im nową nazwę. Można również przedefiniować standardowe style np. “help lines”. Poniższy kod tworzy styl, a następnie go zmienia:

Kod TikZ 11: Przykład przedefiniowania stylu

```
\begin{tikzpicture}
  \tikzset{style1/.style={draw,circle}}
  \tikzset{style1/.append style={fill=red,rectangle}}
    \node[style1] (nazwa) {tekst};
\end{tikzpicture}
```

Rozdział 2

Biblioteki stylów

Pakiet jest szczególnie przydatny w przypadku, gdy rysowany obraz może wykorzystać bibliotekę stylu. Zawiera zdefiniowane różne typy obiektów, które można w łatwy sposób wykorzystać. Aby to zrobić należy je importować. Wśród najważniejszych należy wymienić: “arrows”, “automata”, “backgrounds”, “calendar”, “chains”, “matrix”, “mindmap”, “patterns”, “petri”, “shadows”, “shapes.misc”, “shapes.geometric”, “spy”, “trees”. Kolejne sekcje przedstawiają kody źródłowe, które wykorzystują bibliotekę stylów.

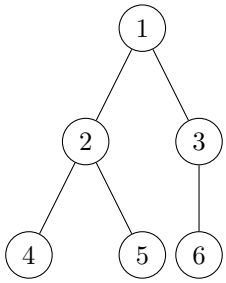
2.1 Drzewa

W przypadku, gdy zadaniem jest narysowanie struktury drzewa można skorzystać z uproszczonej składni. Poniższy kod przedstawia przykład takiego drzewa:

Kod TikZ 12: Uproszczony kod do tworzenia drzewa

```
\begin{tikzpicture}[every node/.style={circle, draw=black}]
  \tikzstyle[sibling distance=20mm]
  \tikzstyle[sibling distance=15mm]
 \node {1}
   child { node {2}
     child { node {4} }
     child { node {5} } }
   child { node {3}
     child { node {6} } };
\end{tikzpicture}
```

Wynikiem działa jest poniższy rysunek:



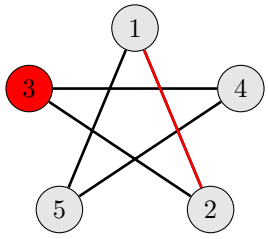
2.2 Grafy

Poprzez łączenie wierzchołków łatwo tworzyć różnego rodzaju grafy. Poniższy kod tworzy spójny graf:

Kod TikZ 13: Przykład kodu tworzącego graf

```
\begin{tikzpicture}[scale=4]
  \tikzstyle{vertex} = [circle,draw,fill=black!10];
  \tikzstyle{selected vertex} = [vertex, fill=red];
  \tikzstyle{selected edge} = [draw,line width=1pt,-,red];
  \tikzstyle{edge} = [-,black,line width=1pt];
    \node[vertex] (v1) at (1.25,1.7) {1};
    \node[vertex] (v2) at (1.5,1.1) {2};
    \node[selected vertex] (v3) at (0.9,1.5) {3};
    \node[vertex] (v4) at (1.6,1.5) {4};
    \node[vertex] (v5) at (1,1.1) {5};
    \draw[edge] (v1)–(v2)–(v3)–(v4)–(v5)–(v1);
    \draw[selected edge] (v1)–(v2);
\end{tikzpicture}
```

Wynikiem poprzedniego kodu jest rysunek:

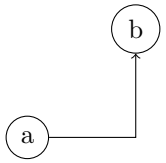


Dostępne są również złamania linii tak jak na poniższym przykładzie:

Kod TikZ 14: Przykład złamania linii dla diagramu

```
\begin{tikzpicture}
  \node [circle, draw] (a) {a};
  \node [circle, draw] (b) [above right=of a] {b};
  \draw [->] (a) -| (b);
\end{tikzpicture}
```

Wynikiem działania kodu jest następujący rysunek:



Przykład wykorzystuje bibliotekę "positioning", którą trzeba importować przed wywołaniem kodu.

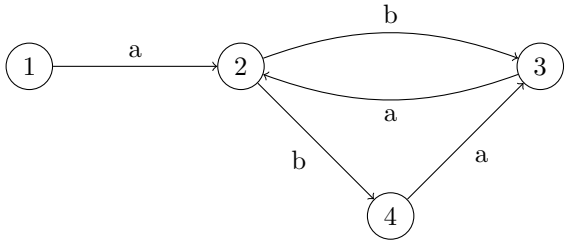
W teorii grafów niektóre krawędzie zaznacza się łukami. W pakiecie TikZ łuki tak oznacza się odpowiednim stylem - "bend left" lub "bend right". Poniższy kod tworzy prosty graf:

Kod TikZ 15: Tworzenie łuków jako połączeń miedzy wierzchołkami

```
\begin{tikzpicture}[auto,node distance=2.8cm]
  \tikzstyle{node}=[circle,draw]
  \node[node] (q1) {1};
  \node[node] (q2) [right of=q1] {2};
  \node[node] (q4) [below right of=q2] {4};
  \node[node] (q3) [above right of=q4] {3};
  \draw[->] (q1) edge node {a} (q2);
  \draw[->] (q2) edge [bend left=20] node {b} (q3);
  \draw[->] (q3) edge [bend left=20] node {a} (q2);
  \draw[->] (q2) edge node [swap] {b} (q4);
  \draw[->] (q4) edge node [swap] {a} (q3);
\end{tikzpicture}
```

Użycie "auto" oznacza, że etykiety linii zostaną automatycznie dopasowane nad łukiem lub pod nim.

Wynikiem poprzedniego kodu jest następujący obraz:



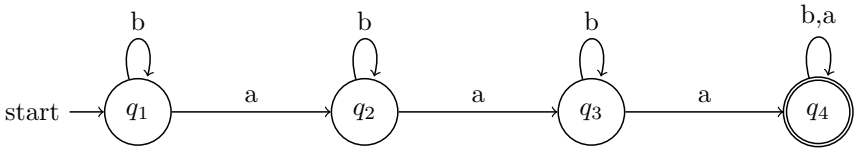
2.3 Automaty

Korzystanie z komend "automata" dostępne jest po importowaniu stylu "automata". Poniższy kod tworzy prostą maszynę stanów:

Kod TikZ 16: Kod tworzący maszynę stanów

```
\begin{tikzpicture}
  \node[initial,state] (A) {$q_1$};
    \node[state] (B) [right of=A] {$q_2$};
    \node[state] (C) [right of=B] {$q_3$};
    \node[state,accepting](D) [right of=C] {$q_4$};
    \draw
      (A) edge [loop above] node {b} (A)
        edge node {a} (B)
      (B) edge [loop above] node {b} (B)
        edge node {a} (C)
      (C) edge [loop above] node {b} (C)
        edge node {a} (D)
      (D) edge [loop above] node {b,a} (D);
\end{tikzpicture}
```

Wynikiem działania jest poniższy rysunek:



2.4 Mapa myśli

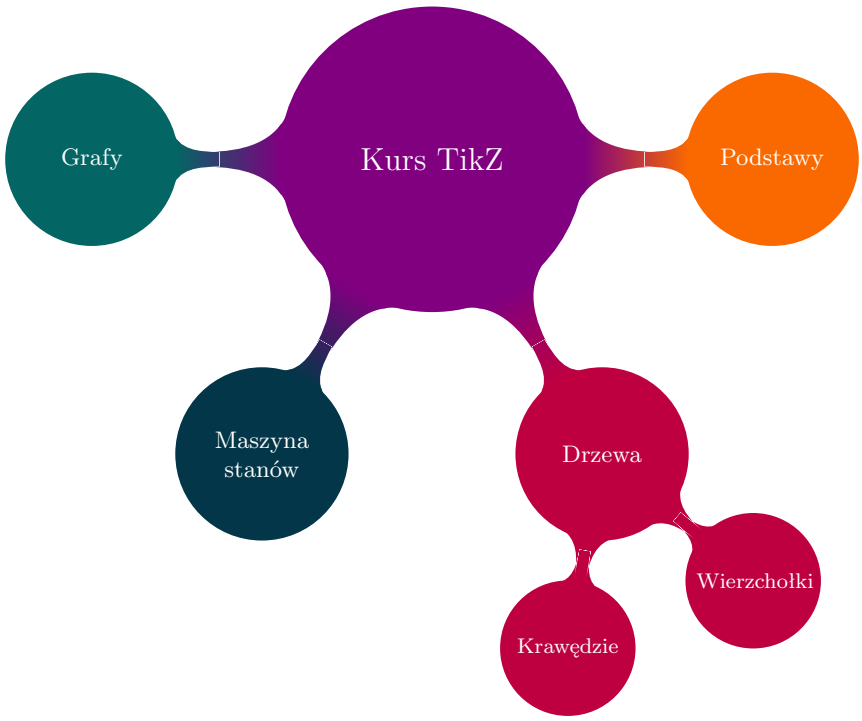
Aby skorzystać z poniższych kodów należy zaimportować bibliotekę stylu "mindmap".

Poniższy listing przedstawia mapę myśli zawierającą projekt tego kursu:

Kod TikZ 17: Kod tworzący mapę myśli

```
\begin{tikzpicture}
    \draw[mindmap,concept color=violet,text=white]
      node[concept] {Kurs TikZ} [clockwise from=0]
      child[concept color=ColorA]
      {
        node[concept] {Podstawy} [clockwise from=90]
      }
      child[concept color=purple]
      {
        node[concept] {Drzewa} [clockwise from=-40]
        child { node[concept] {Wierzchołki}   }
        child { node[concept] {Krawędzie} }
      }
      child[concept color=ColorB] { node[concept] {Maszyna stanów} }
      child[concept color=ColorC] { node[concept] {Grafy} };
\end{tikzpicture}
```

Wynikiem działania jest następujący rysunek:



2.5 Strzałki

Biblioteka stylów “arrows” zawiera zestaw dużej liczby różnorodnych grotów, które można użyć w swoim dokumencie. Tak jak wszystkie dodatkowe biblioteki należy ją najpierw zadeklarować. Następnie nazwę odpowiedniego typu należy podać w nawiasach kwadratowych linii, tak jak używa się standardowych grotów. Poniżej znajduje się lista przykładowych nazw:

a) latex-latex
b) latex'-latex'
c) stealth-stealth
d) stealth'-stealth'
e) triangle 90-triangle 90
f) triangle 60-triangle 60
g) triangle 45-triangle 45
h) angle 90-angle 90
i) [-[
j)]-]
k) (-(
l))-)
m) —-—
n) o-o
o) *-*
p) diamond-diamond
r) square-square
s) serif cm-serif cm

Jeśli grot ma być odwrócony należy po definicji jego rodzaju dodać frazę “reversed”. W przypadku, gdy strzałka ma mieć niezamalowany grot należy dodać frazę “open”.

Rozdział 3

Integracja z Beamer

Pakiet TikZ jest kompatybilny z klasą “Beamer”. Zawiera dodatkowo elementy dostępne tylko dla tej klasy np.:

Kod TikZ 18: Przykład zastosowania pakietu TikZ w klasie Beamer

```
\begin{frame}
  parę linijek
  \begin{center}
    \begin{tikzpicture}
      \draw [blue, ultra thick] (-1,2) – (6,3);
      \uncover<1>{\draw [green,thick] (-4,3) – (2,2.5);}
      \uncover<2>{\draw [red,thick] (0,0) – (0,5);}
    \end{tikzpicture}
  \end{center}
  coś pod spodem.
\end{frame}
```

Rozdział 4

Elementy języka programowania

Poniższy kod przedstawia ogólny szablon pętli:

Kod TikZ 19: Ogólna składania instrukcji "foreach"

```
\begin{tikzpicture}
   \foreach zmienna in {lista wartości}
     komenda;
\end{tikzpicture}
```

Można w liście wartości podać pojedyncze wartości po przecinku lub zbiory oddzielone trzema kropkami. Poniżej znajduje się przykład użycia pętli:

Kod TikZ 20: Przykładowy kod użycia pętli w pakiecie

```
\begin{tikzpicture}
   \foreach \r in {0.5, 1.5,...,7}
      \draw (0,0) circle (\r);
   \foreach \x in {0,1,2,3,4}
      \draw (\x,0) circle (1);
\end{tikzpicture}
```

Kolejną instrukcją sterującą jest "ifthenelse". Zastosowanie wygląda następująco:

Kod TikZ 21: Przykładowy kod użycia pętli w pakiecie

```
\begin{tikzpicture}
   \ifthenelse {\x=3 \OR \y=3 \OR \x=\y}{}{\x \y}
\end{tikzpicture}
```

Bibliografia

[1] ShareLatex. Tikz package. https://www.sharelatex.com/learn/TikZ_package. Dostęp: 2015-04-16.

[2] Jacques Crémer. A very minimal introduction to TikZ. http://cremermisc.com/LaTeX/minimaltikz.pdf, 2011. Dostęp: 2015-04-16.

[3] Veronika Heimsbakk. Tegnekurs i TikZ. http://folk.uio.no/veronahe/TikZ/tikzkurs.pdf, 2014. Dostęp: 2015-04-16.

[4] Norbert Manthey. A brief introduction into TikZ. http://www.computational-logic.org/content/study/master/documents/softskills_tikz.pdf, 2013. Dostęp: 2015-04-16.

[5] Jan-Philipp Kappmeier. How to tikz? an overview. https://www.coga.tu-berlin.de/fileadmin/i26/download/AG_DiskAlg/FG_KombOptGraphAlg/kappmeier/talks/How_to_TikZ.pdf, 2010. Dostęp: 2015-04-16.

[6] Texample. Tikz example. http://www.texample.net/tikz/, 2015. Dostęp: 2015-04-16.