

Zpravodaj Československého sdružení uživatelů TeXu

Luboš Doležal
O makru PSTricks

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 7 (1997), No. 4, 198–207

Persistent URL: <http://dml.cz/dmlcz/149803>

Terms of use:

© Československé sdružení uživatelů TeXu, 1997

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:
The Czech Digital Mathematics Library <http://dml.cz>

1. Úvod

PSTricks je kolekce maker \TeX u založených na PostScriptu. Jejich autorem je Timothy Van Zandt. Makra jsou vhodná pro vytváření jednoduchých obrázků do technických dokumentů. Všechna makra PSTricks jsou kompatibilní s běžnými formáty \TeX u, a to plain \TeX , \LaTeX , $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\TeX$ – jak je udáno². Sám jsem ovšem PSTricks vyzkoušel pouze v plain \TeX u, takže uvedenou kompatibilitu k jinému formátu nemohu potvrdit, ale ani vyloučit. Tento krátký popis maker balíku PSTricks si nečiní nároky na kompletnost, mnoho možností zde nebude uvedeno – tyto možnosti lze nalézt v [1]. Odtud jsou též se souhlasem autora převzaty ukázky použité v článku.

Hlavní soubor, který na začátku načteme, je `pstricks.tex`. Lze také použít `pstricks.sty`. Tyto dva soubory jsou ekvivalentní, ale druhý uvedený je přizpůsoben pro použití v \LaTeX u.

2. Argumenty, parametry grafiky

Na začátku před vlastním vytvářením čehokoliv pomocí PSTricks musíme zajistit nastavení základních jednotek např. pro délku. To se provede příkazem `\psset` takto:

```
\psset{xunit=1pt, yunit=1pt}
\psset{xunit=1cm, yunit=2pt}
```

Jak je vidět, rozměrové jednotky osy x a y , tedy `xunit` a `yunit`, mohou mít buď stejné nebo i rozdílné velikosti. Implicitně jsou obě uvedené jednotky stejné a mají velikost 1 cm.

Kromě rozměrových jednotek se v PSTricks zadávají též úhly a směry. Tyto údaje zadáváme ve stupních, což je implicitní nastavení.

Uvedenou kontrolní sekvencí `\psset` lze nastavit více parametrů než jen rozměrové jednotky. Například i parametry čar – jejich tloušťku (`linewidth=<dim>`) a barvu (implicitně černá). Pro výplně ploch lze nastavit `fillstyle=<style>`,

¹Tento článek vznikl jako semestrální práce z předmětu „Typografický systém \TeX “, který přednáší Petr Olšák na Elektrotechnické fakultě ČVUT.

²Balík PSTricks předpokládá zpracování DVI-souboru PostScriptovou cestou.

`fillcolor=<color>`. Když je nastaveno `fillstyle=none`, není plocha vyplněna, ale když `fillstyle=solid`, je plocha vyplněna barvou `fillcolor`.

3. Základní grafické objekty

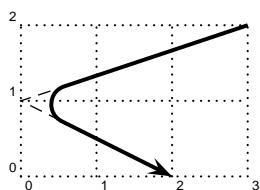
V této části budou uvedeny způsoby vytváření základní grafických objektů, tj. čar, uzavřených objektů vytvořených z čar, kružnic, elips a křivek.

Kontrolní sekvence pro vykreslení čáry má toto základní schema:

```
\psline[<par>]{<arrows>}(x0,y0)(x1,y1)...(xn,yn)
```

Tento příkaz vykreslí čáru přes dané souřadnice. V hranatých závorkách `<par>` jsou parametry dané čáry, např. barva, tloušťka – jak bylo uvedeno výše. Parametr `<arrows>` představuje tvar konců čáry. Například:

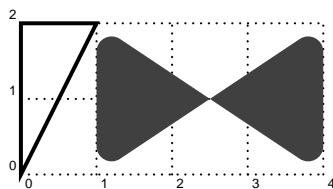
```
\psline[linewidth=1.5pt, linearc=0.25]{->}(3,2)(0,1)(2,0)
```



vykreslí čáru tloušťky 1,5 pt zakončenou šipkou. Počátek čáry je bodě (3,2) a její konec v bodě (2,0). Parametr `linearc` představuje poloměr stočení čáry v „rohu“. Připojená čárkovaná čára (`linestyle=dashed`) na uvedeném obrázku má `linearc=0`, což představuje, jak je vidět, pouhé spojení tří uvedených bodů.

Zvláštním případem `\psline` je příkaz `\pspolygon`, který vytváří uzavřenou cestu. Jeho zápis je obdobný jako pro `\psline`, tedy:

```
\pspolygon[<par>](x0,y0)(x1,y1)...(xn,yn)
```



Příklad užití makra `\pspolygon` byl pořízen následujícími řádky:

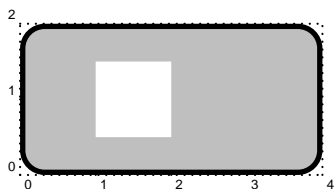
```
\pspolygon[linewidth=1.5pt](0,2)(1,2)
\pspolygon*[linearc=0.2,
linecolor=darkgray](1,0)(1,2)(4,0)(4,2)
```

Trojúhelník na obrázku vytvořil první řádek, tloušťka čáry je zřejmě 1,5 pt. Příkaz `\pspolygon*` je pouze zkráceným vyjádřením pro následující zápis:

```
\pspolygon[linecolor=<color>, fillstyle=solid, fillcolor=<color>]
```

Dalším možným příkazem této části může být `\psframe`. Tento příkaz se zadává ve formě:

`\psframe[par](x0,y0)(x1,y1)`



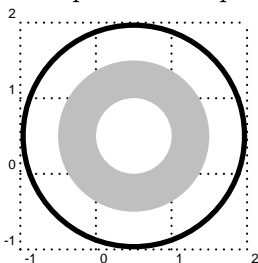
a způsobí vykreslení obdélníku, u něhož (x_0, y_0) je levý dolní a (x_1, y_1) pravý horní roh.

```
\psframe[linewidth=2pt, framearc=0.3,
fillstyle=solid,fillcolor=lightgray](4,2)
\psframe*[linecolor=white](1,0.5)(2,1.5)
```

Poslední řádek v předchozím zápisu uvozený `\psframe*` vyjadřuje, stejně jako v případě `\pssolid`, určitý zkrácený zápis. V tomto případě je možné `\psframe*` chápat jako:

`\psframe[linecolor=color], fillstyle=solid, fillcolor=color]`

což pro `linecolor=white` a také `fillcolor=white` zřejmě dá „vepsaný“ obdélník z předchozího příkladu.



Kružnici lze vykreslit příkazem `\pscircle`. Zadání pozice a rozměrů je provedeno zadáním středu (x_0, y_0) a poloměru $\{radius\}$. Celkové schema je potom následující:

`\pscircle[par](x0,y0){radius}`

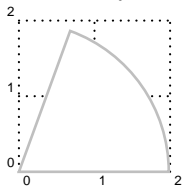
V ukázce je uvedeno několik dalších možností. Kružnice s největším poloměrem má tloušťku čáry 2 pt a její počátek náleží bodu $(.5, .5)$. Jak je patrné, její poloměr je 1,5 cm. Viz první řádek výpisu:

```
\pscircle[linewidth=2pt](0.5,0.5){1.5}
\pscircle*[linecolor=lightgray](0.5,0.5){1}
\pscircle*[linecolor=white](0.5,0.5){0.5}
```

Na druhém uvedeném řádku je zápis pro vykreslení kruhu, který bude vyplněn podle zadání `linecolor`. Rozdíl mezi příkazem `\pscircle` a `\pscircle*` je stejný jako např. u `\psframe`. Poslední řádek je obdobný druhému, pouze je nastavena jiná barva – `white` a menší poloměr – `{0.5}`.

Dalším možným grafickým prvkem definovaným v makru `PSTricks` je kruhová výseč. To znamená, že z kružnice daného poloměru zobrazíme pouze daný úsek

ohraničený mezi dvěma zadanými úhly. Obecné schema zápisu je následující:

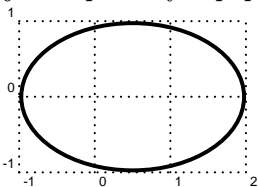


`\pswedge[$\langle par \rangle$](x_0, y_0){ $\langle poloměr \rangle$ }{ $\langle úhel1 \rangle$ }{ $\langle úhel2 \rangle$ }`

kde $\langle par \rangle$ jsou obvyklé parametry (barva, tloušťka čáry). Význam dalších potřebných parametrů – $\langle poloměr \rangle$, $\langle úhel1 \rangle$, $\langle úhel2 \rangle$ – byl již popsán výše. Ukázka tohoto příkazu byla vytvořena následující notací:

`\pswedge[linecolor=lightgray,linewidth=1pt,fillstyle=solid]{2}{0}{70}`

U tohoto příkazu je také možná varianta `\pswedge*`, která má stejnou funkci jako v předešlých popsáných příkladech.



Rozšířením pro vykreslení kružnice je příkaz `\psellipse` resp. `\psellipse*`. Jak název napovídá jedná se o vykreslení elipsy. Ta je zadávána pomocí tří parametrů – středu, horizontálního poloměru a vertikálního poloměru, tedy:

`\psellipse[$\langle par \rangle$](x_0, y_0)(r_1, r_2)`

kde $\langle par \rangle$ jsou znovu již popsané parametry, (x_0, y_0) je bod středu elipsy a r_1 resp. r_2 je velikost horizontálního resp. vertikálního poloměru. Jednoduchý příklad uvedený na obrázku má následující zápis:

`\psellipse[linewidth=1.5pt](.5,0)(1.5,1)`

Je zřejmé, že pomocí tohoto zápisu by šlo vytvořit také kružnici, samozřejmě zadáním $r_1 = r_2$.

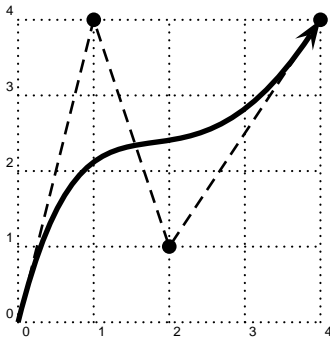
4. Křivky

V makru PSTricks jsou kromě grafických objektů uvedených výše definovány příkazy pro práci s křivkami. Snad nejzákladnějším příkazem této skupiny je `\psbezier`, jehož zápis je následující:

`\psbezier[$\langle par \rangle$]{ $\langle arrows \rangle$ }(x_0, y_0)(x_1, y_1)(x_2, y_2)(x_3, y_3)`

Uvedený příkaz vykreslí mezi (x_0, y_0) a (x_4, y_4) hladkou křivku. Ostatní dva zadané body jsou tzv. kontrolní body křivky. Tím je vytvořena Bézierova

křivka třetího řádu, která je v oblasti počítačové grafiky velice často používána, protože pro její konstrukci existují výkonné výpočetní mechanismy.

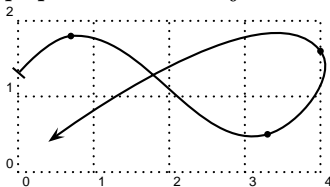


Dalšími parametry, které je možno zadat, jsou již popsané parametry $\langle par \rangle$. Také lze zadat tvar ukončení – parametr $\langle arrows \rangle$.

Příklad byl vytvořen následujícím zápisem:

```
\psbezier[linewidth=2pt,showpoints=true]%
{->}(0,0)(1,4)(2,1)(4,4)
```

Z předchozího popisu je zřejmé použití jednotlivých bodů, proto se blíže věnujeme položce $\langle par \rangle$, ve které je uveden zajímavý a důležitý parametr showpoints=true . Ten nám (pokud je nastaven na true) vykreslí uvedené čtyři body a navzájem je propojí čárkovanou čarou – to je prospěšné, pokud potřebujeme danou křivku dále upravovat. V konečném případě lze samozřejmě nastavit showpoints=false , což nám dané body skryje.

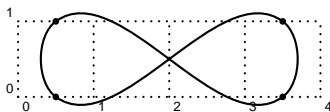


Předchozí příklad pro vykreslení křivky je sice názorný k předvedení Bézierovy kubiky, ale konstrukce pomocí něho je poněkud těžkopádná. Pokud máme několik bodů, které chceme proložit hladkou křivkou, užijeme spíše příkaz \pscurve . Jeho zápis je podobný předchozímu, tedy:

```
\pscurve[ $\langle par \rangle$ ]{ $\langle arrows \rangle$ }(x1,y1) ... (xn,yn)
```

Tento příkaz proloží přes body (x_1, y_1) až (x_n, y_n) otevřenou hladkou křivku. Jak již bylo uvedeno, tvar zakončení se řídí pomocí zadání $\langle arrows \rangle$. Parametry $\langle par \rangle$ mohou znovu obsahovat např. zadání tloušťky čáry, ale také příkaz showpoint , který nastaven na true označí body cesty, ale již je nepropojí čárkovanou čarou jako v případě \psbezier . Příklad byl vytvořen následujícím zápisem:

```
\pscurve[showpoints=true]{|->}(0,1.3)(0.7,1.8)(3.3,0.5)(4,1.6)
(0.4,0.4)
```



Rozšířením příkazu \pscurve je možnost vykreslit i uzavřenou křivku. Pro tyto případy existuje zápis \psccurve , kde „přebytečné c“ znamená *closed*. Zápis celého příkazu je obdobný, tedy:

```
\psccurve[⟨par⟩]{⟨arrows⟩}(x1,y1) \dots (xn,yn)
```

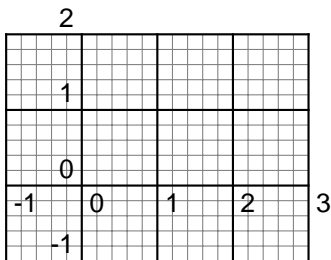
Obrázek k demonstraci tohoto příkazu byl pořízen zápisem:

```
\psccurve[showpoints=true](0,1.3)(0.5,0)%  
(3.5,1)(3.5,0)(0.5,1)
```

5. Mřížky

Jak již bylo možno vidět, pro všechny zařazené ukázky makra PSTricks bylo užito jednoduché mřížky, do které byly příklady příkazů zakresleny. Tato mřížka je pouze speciálním případem pro příkaz `\psgrid`, jehož zápis je následující:

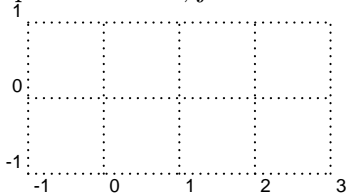
```
\psgrid[⟨par⟩](x0,y0)(x1,y1)(x2,y2)
```



Význam parametrů $\langle par \rangle$ bude uveden dále. Tento příkaz vykreslí mřížku s protilehlými rohy o souřadnicích (x_1, y_1) a (x_2, y_2) . Interval hodnot je číslován. Toto číslování začíná bodem (x_0, y_0) , jak ukazuje uvedený příklad:

```
\psgrid(0,0)(-1,-1)(3,2)
```

Koordináční systém je chápán jako kartézský souřadný systém. Dále je možné poznamenat, že pokud se neuvede pozice bodu (x_0, y_0) , tzn. provedeme zápis pouze dvou bodů pro konstrukci, je brán bod (x_1, y_1) jako počátek.



Na místo $\langle par \rangle$ lze zapsat mnoho parametrů, ze kterých asi nejdůležitější budou uvedeny v následujícím příkladě. Mřížka, která bude dále uvedena, byla v různých rozměrových obměnách použita v dříve uvedených ukázkách. Zápis pro vygenerování příkladu je následující:

```
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt](-1,-1)(3,1)
```

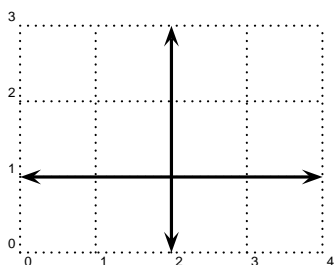
Parametr `subgriddiv` nastavuje jemnější dělení zakreslené mřížky. Implicitní hodnota je `subgriddiv=5`, což také koresponduje s první ukázkou této sekce. Dalším parametrem je `griddots=⟨num⟩`. Pokud je číselná hodnota $\langle num \rangle$ pozitivní, jsou čáry mřížky nakresleny tečkovaně, kde hodnota $\langle num \rangle$ je po-

čet bodů na dělení mřížky. Poslední parametr je `gridlabels`, který nastavuje stupeň písma pro popisek mřížky. Pro popisek je užíván implicitně bezserifový PostScriptový font Helvetica ve velikosti 10 pt, pokud není nastaveno jinak.

6. Osy

Velice podobné vlastnosti jako výše popsané makro `\psgrid` má také makro pro vytváření os (např. pro kreslení matematických funkcí nebo průběhů zjištěných měření). Jedná se o makro `\psaxes`. Před jeho používáním je ale nutné načíst soubor `pst_plot.tex`, ve kterém je popsáno makro této sekce. Zápis je následující:

```
\psaxes[⟨par⟩]{⟨arrows⟩}(x0,y0)(x1,y1)(x2,y2)
```



Stejně jako v případě `\psgrid` jsou jednotlivé body (x_0, y_0) až (x_2, y_2) použity následujícím způsobem: počátek je zadán bodem (x_0, y_0) , levý dolní roh bodem (x_1, y_1) a protilehlý pravý horní (x_2, y_2) , jak je patrné srovnáním mezi uvedeným obrázkem a předpisem:

```
\psaxes[linewidth=1.2pt,labels=none,
ticks=none]{<->}(2,1)(0,0)(4,3)
```

Parametr `⟨arrows⟩` znovu dovoluje nastavit požadované ukončení jednotlivých os, v našem případě se šipkou. To je pravděpodobně nejpoužívanější možnost.

Nyní si blíže všimneme parametrů umístěných v `⟨par⟩`, protože tento příkaz je má, oproti `\psgrid`, poněkud jiné. Některé nejdůležitější je možno vidět již u předchozího zápisu. Parametr `labels=none/x/y/all` dovoluje čtyři možnosti v umístění číslování pozic. Buď se toto číslování nepovolí nebo pouze na jednu nebo druhou osu nebo je možné číslování povolit na obě osy současně. Podobně se chová parametr `ticks`, který má za úkol zakreslit krátké (jejich délka se dá nastavit pomocí `ticksize=⟨num⟩`) kolmé čárky vůči osám na jednotlivých pozicích. Dále lze tyto čárky zakreslit pouze nad nebo pod osou, ale také oboustranně – parametr `tickstyle=full/top/bottom`.

7. Kreslení grafů funkcí

Téma této sekce může být jistým pokračováním sekcí předchozích. Do zakreslené mřížky nebo souřadných os lze dále zakreslit graf matematické funkce. Jediný

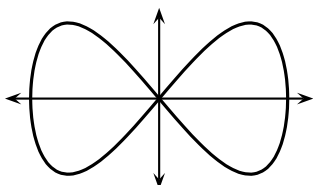
problém může způsobit asi zápis funkce – používá se PostScriptová notace zápisu. Pro vykreslení dané funkce se používá příkaz `\psplot` zápisem:

```
\psplot[⟨par⟩]{⟨xmin⟩}{⟨xmax⟩}{⟨funkce⟩}
```

Parametry `⟨par⟩` jsou již dříve uvedené parametry, např. pro nastavení tloušťky čáry. Numerické hodnoty `⟨xmin⟩` a `⟨xmax⟩` vymezují rozsah nezávisle proměnné x . Funkce je zadána zápisem ve `⟨funkce⟩`. Jinak řečeno, `⟨xmin⟩` a `⟨xmax⟩` udávají interval, ve kterém bude daná funkce vykreslena. Užití jednotlivých parametrů je jistě patrné z následujícího velice jednoduchého příkladu:

```
\psplot[plotpoints=200,plotstyle=curve]{0}{720}{x sin}
```

který vytvoří graf funkce $\sin x$ pro x od 0° do 720° . V tomto intervalu bude vypočteno 200 hodnot (zadáno pomocí `plotpoints`), které budou proloženy hladkou křivkou. To je zadáno pomocí `plotstyle`. Lze také zadat `plotstyle=dots` – namísto hladké čáry bude graf tvořen tečkami.



Pro kreslení funkcí zadaných parametricky existuje příkaz `\parametricplot`, jehož zápis je následující:

```
\parametricplot[⟨par⟩]{tmin}{tmax}{⟨funkce⟩}
```

Snad jedinou změnou oproti příkazu `\psplot` je to, že namísto nezávisle proměnné x je zadáván interval s parametrem t . Způsob zápisu funkce (je to ale spíše vektor $(x(t), y(t))$ proměnné t) je patrný z následujícího příkladu užití:

```
\psset{xunit=1.7cm}
\parametricplot[linewidth=1.2pt,plotstyle=ccurve]{0}{360}{t sin t 2 mul sin}
\psline{<->}(0,-1.2)(0,1.2)
\psline{<->}(-1.2,0)(1.2,0)
```

Na obrázku byla vykreslena parametrická funkce $(\sin t, \sin 2t)$, pro $t_{\min} = 0^\circ$ a $t_{\max} = 360^\circ$. Jednoduché osy byly vytvořeny příkazem `\psline`. Samozřejmě, že by také šlo získat souřadný systém pomocí `\psaxes`.

8. Popisy

Často je třeba zařadit do nakresleného obrázku text, např. názvy os u souřadného systému při kreslení grafů nebo název vykreslené funkce. Pro tyto účely je v makru PSTricks definován příkaz `\uput`, jehož zápis je následující:

```
\uput{⟨labelsep⟩}[⟨refangle⟩]{⟨angle⟩}(x,y){⟨popis⟩}
```

• (1,1) Příkaz provede následující: vzhledem k bodu zadanému v (x, y) vloží `⟨popis⟩` ve vzdálenosti `⟨labelsep⟩` (implicitně je v makru nastaveno `labelsep=5pt`) od tohoto bodu. Dále se pomocí parametru `⟨refangle⟩` udává natočení celého textu v `⟨popis⟩` vzhledem k referenčnímu bodu (x, y) . Parametr `⟨angle⟩` udává potom vlastní natočení textu v `⟨popis⟩`. K popisu se používá standardní T_EXovský font. Implicitně není pro tento účel nastaveno používání PostScriptových písem.

Jednoduchý příklad – popis bodu – byl vytvořen zápisem:

```
\qdisk(1,1){2pt}  
\uput[45](1,1){(1,1)}
```

Uvedený příkaz `\qdisk` je speciálním případem pro `\pscircle`. Zadává se bod umístění, v našem případě je to bod $(1, 1)$, a průměr tohoto bodu pomocí čísla s jednotkou, např. 2 pt.

9. Umísťování grafiky do dokumentu

Grafické objekty, ale např. také příkaz `\uput`, nedokáží samostatně změnit chování T_EXu. To znamená, že vytvářejí bezrozměrný box. Abychom mohli zařadit vytvořený obrázek do dokumentu, musíme příkazy tvořící tento obrázek umístit do prostředí `\pspicture`, jehož schema je následující:

```
\pspicture[⟨baseline⟩](x0,y0)(x1,y1)  
⟨popis objektu...⟩  
\endpspicture
```

Obrázek bude potom vložen do boxu. Levý dolní roh je v bodě (x_0, y_0) (implicitně nastaveno na $(0, 0)$) a pravý horní okraj bude v bodě (x_1, y_1) .

Velikost parametru `⟨baseline⟩` je poměrná část z výšky daného boxu – může nabývat hodnoty od 0 do 1. Pokud je nastavena 0, potom se jedná o spodní okraj. Při hodnotě `baseline 1` je nastaven horní okraj. Hodnoty „mezi“ pochopitelně

nastavují `baseline` mezi spodní a horní okraj. Implicitně je nastaveno `baseline` na spodní část vytvořeného boxu.

Pro uživatele \LaTeX u je zápis pro toto prostředí poněkud jiný – odpovídá konvencím tohoto \TeX ovského makra, takže:

```
\begin{pspicture}[(baseline)](x0,y0)(x1,y1)
<popis objektu...>
\end{pspicture}
```

Dále lze v \LaTeX u používat grafické objekty z `PSTricks` v \LaTeX ovském prostředí `picture` a naopak je možné používat objekty vytvořené v prostředí `picture` v prostředí `pspicture`. Samozřejmě se všemi omezeními, které jsou \LaTeX ovskému prostředí `picture` vlastní.

10. Závěr

Jak již bylo řečeno v úvodu tohoto dokumentu, není jeho cílem podat vyčerpávající popis všech možností a „triků“ s balíkem maker `PSTricks`. Není zde například podán úplný přehled parametrů, které se zapisují na místo `<par>` nebo do příkazu `\psset`. Jsou zde uvedeny pouze nejdůležitější parametry. Čtenář, který by se o toto makro více zajímal, si jistě vyhledá originální dokumentaci, samozřejmě v angličtině. Tam je celé makro velice podrobně popsáno včetně jeho možností. Tato originální dokumentace by měla být součástí každé distribuce `PSTricks`.

Literatura

[1] Timothy Van Zandt: *PSTricks: PostScript macros for Generic \TeX . User's Guide*. Dostupné na CTAN, `graphics/pstricks/`.

Luboš Doležal
dolezal1@feld.cvut.cz