Technische Universität München

Fakultät für Mathematik
Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

Visualization of advanced graph algorithms

`push-relabel` algorithm
to solve the maximum flow problem
`label-setting` algorithm
to solve the shortest path problem with resource constraints

Interdisciplinary Research Project (IDP) by Adrian Haarbach

Examiner: Prof. Dr. Peter Gritzmann

Advisor: Wolfgang F. Riedl

Submission Date: September 15, 2015

I hereby confirm that this is my own work, and that I used only the cited sources and materials.

München, September 15, 2015

<div style="text-align:center">———————————————————</div>

<div style="text-align:center">Adrian Haarbach</div>

Abstract

This interdisciplinary project deals with the vivid visualization of advanced graph algorithms. In particular, algorithms to solve two distinctive problems in discrete math are considered. Namely the `max-flow problem` as well as the `shortest path problem with resource constraints`. For efficiency reasons, one often employs advanced graph algorithms to solve above problems. The `max-flow problem` is solved using the efficient push-relabel algorithm of Goldberg-Tarjan. To solve the `shortest path problem with resource constraints`, we employ a generic label-setting algorithm which follows the dynamic programming principle.

Zusammenfassung

Das vorliegende interdisziplinare Projekt beschaftigt sich mit der anschaulichen Darstellung von fortgeschrittenen Graphalgorithmen. Betrachtet werden zwei Verfahren zur Losung von Problemstellungen der diskreten Mathematik. Die zu visualisierenden Problemstellungen sind hierbei das `Max-Flow Problem` sowie das `Kurzeste-Wege Problem mit Ressourcenbeschrankungen`. Als Losungsverfahren fur die aufgefuhrten Problemstellungen werden aus Effizienzgrunden haufig fortgeschrittene Graphalgorithmen herangezogen. Fur die Losung des Max-Flow Problems findet der bekannte Push-Relabel Algorithmus von Goldberg-Tarjan in der Praxis haufig Anwendung. Zur Losung des Kurzeste-Wege Problems mit Ressourcenbeschrankungen wird mit einem Label-Setting Algorithmus ein bekanntes Verfahren der dynamischen Programmierung vorgestellt.

Contents

Chapter 1

Introduction

This chapter provides a short example of the use of the `tumthesis` class.

1.1 Files

Table 1.1 shows all the files associated with this example, each with a short description.

File name	Description
`tumthesis.cls`	class file, defines basic commands and incorporates important packages
`tumcolors.sty`	LaTeX package in which the official TUM colours are defined; used by tumthesis.cls.
`thesis.tex`	Main file of this example and starting point for own project. All other `.tex` files are included using this file.
`thesis.pdf`	PDF version of thesis.tex
`preamble.tex`	Preamble loading custom packages of user
`abstract.tex`	Abstract text in English and German
`introduction.tex`	Text for this chapter
`conclusion.tex`	Text for the following chapter
`appendix.tex`	Text for the appendix
`thesis.bib`	BibTeX file for the bibliography
`TUMlMblau`	Logo of the Department of Mathematics for the title page
`TUMloblauSchriftzugL`	TUM Logo for the title page
`TopMath-Bildmotiv.jpg`	TopMath Logo for the title page

Table 1.1: Files for this example

The following commands should be used to to compile the final pdf from these source files:

```
pdflatex thesis
biber thesis
```

```
makeindex -s myindex.ist
pdflatex thesis
pdflatex thesis
```

The first run-through of `pdflatex` creates various auxiliary files and a (mostly complete) pdf output – some graphics may appear in the wrong place and the references and citations do not yet work correctly. With the `biber` command, the system works through the `thesis.bib` file and creates the bibliography (c.f. Section 1.7.2), while the `makeindex` command creates the index (c.f. Section 1.8.1). The subsequent two `pdflatex` commands set references and place graphics correctly.

Advanced users can also automise the process by using the command

```
latexmk --pdf thesis
```

The `latexmk` tool then ensures each command in the BibTeX runthrough is called the correct number of times.

1.2 Configuration and Options

Custom packages of the user or overwritten settings can be loaded in file `preamble.tex`. The code contained in this file will be included and executed right at the end of the class (just before loading the last packages `hyperref` and `cleveref`).

The `tumthesis` class accepts some options, that help to adjust the titlepage and some behaviour:

- `topmath`: This option places the "TopMath-Bildmotiv" on the titlepage:

  ```
  \documentclass[topmath]{tumthesis}
  ```

Alternatively, you can place your own logo on the titlepage:

- `titlepicture`: Filename of your logo

- `titlepictureX`: Horizontal distance (including unit) between lower right corner of the titlepage and lower right corner of the logo

- `titlepictureY`: Vertical distance (including unit) between lower right corner of the titlepage and lower right corner of the logo

  ```
  \documentclass[titlepicture=MA_CMYK.pdf,titlepictureX=25mm,
  titlepictureY=40mm]{tumthesis}
  ```

  would include the math logo once again in the lower right corner of the titlepage.

Furthermore, you can adjust the behaviour of theorems' titles:

- `theoremtitle`: Whether the content of a theorem is typeset next to its name (nobreak) or not (break, standard option)

  `\documentclass[theoremtitle=nobreak]{tumthesis}`

A further option allows the user to specify the BibLaTeX backend to be used:

- `biblatexBackend`: The default is set to biber, every valid option for parameter "backend" of the biblatex package is an alternative:

  `\documentclass[biblatexBackend=bibtex]{tumthesis}`

1.3 Basic Settings

At the very beginning, the `thesis.tex` file fixes some basic settings. The code is as follows:

```
% -------------------------------
% PDF-Information
\hypersetup{
 pdfauthor={Wolfgang Ferdinand Riedl, Michael Ritter},
 pdftitle={The tumthesis Class},
 pdfsubject={A Tutorial for Theses},
 pdfkeywords={Master's Thesis, Bachelor's Thesis},
 colorlinks=true, %coloured links (for the PDF version)
% colorlinks=false, % no coloured links (for the print version)
}

 % -----------------------------

 % Basisdaten

 \author{Wolfgang F. Riedl, Michael Ritter}
 \title{The \texttt{tumthesis} Class}
 \subtitle{A Tutorial for Theses}
 \faculty{Fakultät für Mathematik}
\institute{Lehrstuhl für Angewandte Geometrie und Diskrete ↷
     ↪ Mathematik}
%\subject{master}
%\subject{bachelor}
%\subject{diploma}
%\subject{project}
%\subject{seminar}
```

```
   %\subject{idp}
   \subject{Short Overview}
   \professor{Prof. Dr. Peter Gritzmann} %Themensteller
   \advisor{Dr. René Brandenberg} %Betreuer
30 \date{26.12.2012} %Submission Date
   \place{München} %Place where document is signed
```

The inputs in the `hypersetup` command do not appear in the document itself, but are embedded into the pdf file as metadata and can be viewed in Acrobat Reader (and many other pdf-viewers). The remaining commands should be largely self-explanatory. The `subject{}` command accepts any desired text as input (such as "Short Overview" in this example), or you may use one of the pre-defined key words, which automatically create a title of "Master's Thesis", "Bachelor's Thesis" or other suitable output. To do this, simply remove the commentary marks from the appropriate line and comment out the currently active `subject` command. Be careful, of course, to match the selected language (see Section 1.4). (The `\subject{}` can also be omitted entirely, in which case there will be no designation on the title page and only the author name will appear.)

1.4 Language selection und Character set

The class supports English and German as language options. The language is set using the command

```
   \selectlanguage{english}
```

at the start of `thesis.tex`. The language can be changed at any point in the document using this command or respectively `\selectlanguage{ngerman}`. This also means that some settings are automatically changed to match, e.g. the commands `\eg` and `\ie` result in the appropriate text (these commands should be used in any case, since they ensure that the spacing is typographically correct) and the headings change, but there are also some more subtle changes such as the rules for automatic hyphenation. An example of such a language switch can be found in the abstract.

For your own files it is important to select the correct "encoding". Here the default is Unicode (UTF-8). This allows you to type in umlauts and other special characters directly, but may require the correct settings in the editing programme. Some editors, especially in Windows systems, are set to Latin-1 rather than Unicode – this can lead to interesting errors!

1.5 Printing

When printing, be sure to print the thesis double-sided. The side margins, headers and footers are set out for double-sided printing and binding. If you want to change the

space left for binding simply adjust the line

```
BCOR =5 mm % Binding correction , ensures sufficient space ↷
    ↪ for binding
```

in file `tumthesis.cls`.

1.6 Titlepage

Per default, the titlepage is set to a standard titlepage following the TUM-Styleguide as close as possible. By replacing line

`\maketitlepage%`

in the file `thesis.tex` with

`\maketitlepageDissertation%`

a titlepage suitable for a dissertation is created.

1.7 Important notes

1.7.1 Math Environments

As this class loads the package `ntheorem`, math environments of the following form lead to errors: `\[ ... \]`. Replace them by `\begin{equation*} ... \end{equation*}` instead.

1.7.2 Biblatex

he package currently uses the biber backend which can handle UTF-8 encoded bibliography files. This default option can be changed by the parameter `biblatexBackend` as described above.

You can find manuals for the setup of your LATEXeditor with biber for most editors by searching the web.

1.8 Some Packages

The `tumthesis.cls` class includes a range of useful packages, which are listed below.

1.8.1 Index

The tumthesis style file loads the package imakeidx, which allows quick and easy generation of an *index*. To include a word or definition the index, simply append `\index{keyword}`. For example, the keyword "index" is added to the index of this document in the following way:

```
... easy generation of an \emph{index}\index{index}. To ...
```

Adding symbols is just as easy: The symbol ζ is included in the index by the line

```
... the symbol $\zeta$\index{$\zeta$} is included ...
```

To change the position of the symbols (or any other elements) in your index, you can specify an additional keyword (which may also be a formula):

To include the symbol π in the index such that it appears in the place where the word "pi" would appear and also in the place where the symbol "p_i" would appear, you can use the code

```
... symbol $\pi$\index{pi@$\pi$}\index{$p_i$@$\pi$} is ↷
    ↪ included ...
```

Subcategories can also easily be realized, by using `\index{keyword!subkeyword}`: For example, the definition of a metric space

Definition 1.1 (Metric)
Let X be a set and $d : X \times X \longrightarrow \mathbb{R}$. The function d is a metric on X if the following three properties hold for all $x, y, z \in X$

1. $d(x,x) \geq 0$ and $d(x,y) = 0 \iff x = y$ (non-negativity)

2. $d(x,y) = d(y,x)$ (symmetry)

3. $d(x,z) \leq d(x,y) + d(y,z)$ (triangle inequality).

can be properly referenced in the index by the following code:

```
\begin{definition}[Metric]
  \index{metric}
  Let $X$ be a set and $d: X \times X \longrightarrow ↷
      ↪ \mathbb{R}$. The function $d$ is a metric on $X$ if ↷
      ↪ the following three properties hold for all $x,y,z \in ↷
      ↪ X$
  \begin{enumerate}
5     \item $d(x,x) \geq 0$ and $d(x,y) = 0 \iff x = y$ ↷
        ↪ (non-negativity)\index{metric!non-negativity}
    \item $d(x,y) = d(y,x)$ (symmetry)\index{metric!symmetry}
    \item $d(x,z) \leq d(x,y) + d(y,z)$ (triangle ↷
        ↪ inequality)\index{metric!triangle inequality}.
  \end{enumerate}
\end{definition}
```

To create the index, the line

```
\makeindex[title=Index,options=-s myindex]
```

has to be added to the file `thesis.tex` *before* the `\begin{document}` command! The option `title=Index` specifies the heading of the index, in this case "Index"; the second option sets a custom style file (does not work in any environment, see compiler options below if it does not work).

The index can then be added to the document by the command

```
%Add the index to the table of contents
\addcontentsline{toc}{chapter}{Index}
%print the index
\printindex
```

The index has to be compiled by the command `makeindex` (which most editors will apply automatically). The layout of the index can be modified by specifing a style file via the option `-s stylefile.ist` (or sometimes the option given to the `\makeindex` command above). This document was compiled using the file `myindex.ist`, which is included in the package.

1.8.2 scrbook

The `tumthesis.cls` class builds entirely on `scrbook.cls`. In particular, this means that all of the options and commands in `scrbook` are available here. More information can be found in the documentation [KM12a] or in the printed version Kohm and Morawski [KM12b].

1.8.3 csquotes

Among other things, this package makes the `\enquote{}` command available, which automatically ensures that quotation marks are correct. The package takes account of the currently active language: in English text "English quotation marks" appear, wobei deutsche Texte „entsprechende Anführungszeichen" bekommen.

1.8.4 cleveref

In LaTeX we normally make references using `\ref{}`. This package defines the new commands `\cref` and `\Cref`, which ensure that as well as the correct number, the correct descriptive text also appears (in the currently selected language). The latter command also ensures capitalisation and should therefore be used at the beginning of a sentence (although in German this often makes no difference, since labels are usually nouns, which are capitalised in any case). An example can be found above and also here: The references to Table 1.1 are made using `cleveref`, the word "Table" is included automatically.

1.8.5 ntheorem

This package prepares a range of standard environments for definitions, theorems, proofs etc. The labels are determined by the language. For example:

Definition 1.2
Every element of a vector space is called a *vector*.

Theorem 1.3 (Fundamental theorem of vector space terminology)
For every vector v there is a vector space V with $v \in V$.

Proof. The proof is trivial and is left as an exercise for the reader. It is really not hard, just try it. □

Wir zeigen jetzt eine deutsche Version dieses Beweises:

Beweis. Der Beweis von Theorem 1.3 ist höchst trivial und nur ein Vollidiot würde es nicht selber können. Wenn Sie sich überhaupt die Mühe gemacht haben, diesen Beweis zu lesen, überlegen sie sich vielleicht, ob Sie nicht lieber ein anderes Fach studieren sollten. □

We formulate another theorem in order to demonstrate another feature of `cref` with which one may group several references together. To this end, we reference Theorems 1.3 and 1.4

Theorem 1.4
*L*A*T*E*X is great!*

The environment can of course be extended and customised. For more information, see the examples in `tumthesis.cls` or the documentation for the `ntheorem` package [MS11]. All of the pre-defined environments are listed in Table 1.2.

1.8.6 booktabs

This package allows for nicer tables, such as Table 1.1. There are many tips on setting out tables in the extensive documentation for this package.

1.8.7 tabularx

A tabular modifying the width of certain columns in order to achieve a custom width can be constructed using package `tabularx`.

Environment	Text	
	English	**German**
`definition`	Definition	Definition
`theorem`	Theorem	Satz
`satz`	Theorem	Satz
`lemma`	Lemma	Lemma
`proposition`	Proposition	Proposition
`corollary`	Corollary	Korollar
`korollar`	Corollary	Korollar
`remark`	Remark	Bemerkung
`bemerkung`	Remark	Bemerkung
`example`	Example	Beispiel
`beispiel`	Example	Beispiel
`proof`	Proof	Beweis
`beweis`	Proof	Beweis
`conjecture`	Vermutung	Conjecture
`vermutung`	Vermutung	Conjecture
`problem`	Problem	problem

Table 1.2: predefined `ntheorem` environments

1.8.8 TikZ

TikZ may not be a graphics program, but you can still create some excellent graphics with it. The documentation [Tan08] is extensive. Online under `http://www.texample.net` are a number of examples which demonstrate what you can do with this package. Lest the list of figures remain empty, we include a TikZ graphic in Fig. 1.1a. Fear not, you do not have to understand the code straight away. There are some simpler and well-described examples in the TikZ manual.

1.8.9 subcaption

To create multiple subfigures within a figure, you can use the package `subcaption` and its environment `\begin{subfigure} ... \end{subfigure}`. It gives the opportunity to create subfigures and subtables using the same syntax as used for figures and tables.

1.8.10 fixme

This package allows you to make notes in your document which mark points where more work is needed. A "List of Corrections" then appears at the very end of the document, in which all the notes are listed. To demonstrate, this paragraph contains two such

FiXme Note!
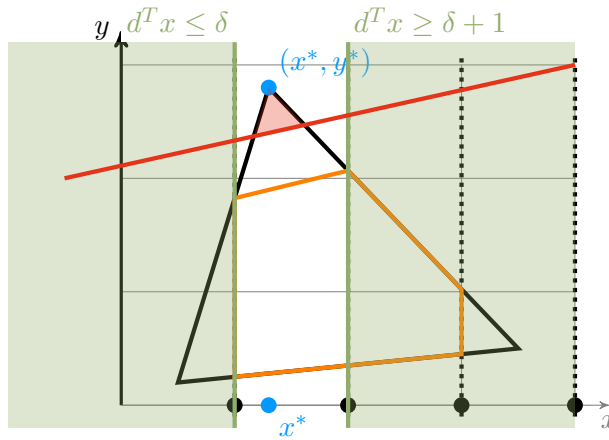
(a) Example of a split disjunction   **(b)** The TUM Logo

Figure 1.1: Two graphics

FixMe notes—these appear as notes at the side, but also in the aforementioned "List of Corrections" at the very end of the document. Here also there are many possible settings and the *documentation* is recommended reading. We mention here one setting which should appear at the beginning of the `thesis.tex` file:

```
%FixMe-Status: final (no FixMe notes) or draft (notes visible)
\fxsetup{draft}
%\fxsetup{final}
```

Replacing the "draft" line with the "final" line will result in two things: All `\fxfatal{}` commands will become LATEXerrors which break off the TEXcompilation of the document (useful for highlighting really bad mistakes which should definitely not be overlooked). All other fixme commands (i. e. `\fxnote{}`, `\fxwarning{}`, `\fxerror{}`) will simply become invisible, the notes in the text and the "List of Corrections" at the end of the document disappear. More information on the available commands and the many possible settings can be found in the documentation [Ver09].

1.8.11 hyperref

The `hyperref` package fixes a number of pdf settings (see Section 1.3). Furthermore, the package ensures that all references, citations and the table of contents become clickable links which allow the reader to jump back and forth in the document. By default these links appear in black and are therefore not immediately visible. Alternatively with the `colorlinks=true` setting in the `\hypersetup{}` command at the beginning of the document, these links become dark blue. This is convenient for the on-screen version; however for the print version you should revert to black, i. e. use the option

`colorlinks=false` (after all, you can't click on links in the printed document).

```
\hypersetup{
 pdfauthor ={Wolfgang Ferdinand Riedl, Michael Ritter },
 pdftitle ={The tumthesis Class },
 pdfsubject ={A Tutorial for Theses },
5 pdfkeywords ={Master 's Thesis, Bachelor 's Thesis },
 colorlinks =true , %coloured links (for the PDF version)
% colorlinks=false , % no coloured links (for the print version)
}
```

1.8.12 listings

The `listings` package allows you to produce nicely formatted source code listings. In this example it is used to produce LaTeX source code. The standard settings take care of line numbering, line breaks and various other details. Of course many settings can be customised; for details see the documentation [HM07]. One small warning: The package is set up to deal with umlauts and "ß" in the source texts, but other special characters may cause problems (even in comments). It is best to avoid special characters altogether in source texts—if this is not possible, take a look at `tumthesis.cls` to see how to modify the settings to deal with other special characters.

1.8.13 algorithm2e

`algorithm2e` provides the possibility to create floating algorithm environments. In contrast to `listings` it does not produce source code formatted according to a certain programming language, instead you write formatted pseudo-code with a predefined syntax.

Chapter 2

Further Reading

In this chapter we collect some suggested literature for LaTeX matters, which may be of use for beginners and for more advanced users and may provide some useful tips.

lshort: "The Not So Short Introduction to LaTeX" (see [Oet+11]) is an up-to-date introduction which can be worked through in a moderate amount of time (the authors give an estimated time of 157 minutes for version 5.01, the most recent at the time of writing.) An up-to-date version can be found at `http://tobi.oetiker.ch/lshort/lshort.pdf`.

LaTeX and Friends: The book [Don12] is a recommended and up-to-date introduction to LaTeX which addresses many current packages. Worth a look for both beginners and advanced users.

l2tabu: There are many tips for older packages and LaTeX commands in [ET11], which are particularly recommended for advanced LaTeX users. Here you can learn why certain commands are best avoided and what the alternatives are. Note: The `tumthesis.cls` class automatically calls upon the `nag` package, which immediately rings alarm bells with many of the mistakes listed in l2tabu.

Appendix A

Remarks on Implementation

In the appendix you can include e. g. computer codes or further remarks which would disturb the flow of the main text. If you do not need an appendix, you can simply leave out this file (in which case you should also delete the `\include` command in `thesis.tex`).

List of Figures

All figures in this document were created by the author using TikZ, the excellent TEX-package by Till Tantau, see [Tan08].

List of Tables

Index

Bibliography

[Don12]   M. R. C. van Dongeren. *LaTeX and Friends*. Springer, 2012.

[ET11]   M. Ensenbach and M. Trettin. *Das LATEX 2e-Sündenregister*. 2011.

[HM07]   C. Heinz and B. Moses. *The Listings Package*. 2007.

[KM12a]   M. Kohm and J.-U. Morawski. *KOMA-Script*. 2012.

[KM12b]   M. Kohm and J.-U. Morawski. *KOMA-Script*. 4. Auflage. Berlin: Verlag:Lehmanns, 2012.

[MS11]   W. May and A. Schedler. *An Extension of the LATEX-Theorem Evironment*. 2011.

[Oet+11] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *The Not So Short Introduction to LATEX2e*. 2011.

[Tan08]   T. Tantau. *The TikZ and PGF Packages. Manual for version 2.00*. Lübeck, Feb. 2008.

[Ver09]   D. Verna. *FiXme – Collaborative annotation tool for LATEX*. 2009.

List of Corrections