

Die T_EXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung T_EX e.V.

16. Jahrgang Heft 2/2004 Juni 2004

2/2004

Impressum

„Die T_EXnische Komödie“ ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit Hilfe folgender Programme erstellt: `pdfTeX 3.14159-1.00b-pretest-20020211 (Web2C 7.3.7x)`, `LaTeX2e (2001/06/01)`, `Acrobat Reader 5.0.5` und `xdvi(k) 22.40k` für die Bildschirmdarstellung. Als Standard-Schriften kamen die Type-1-Fonts Latin-Modern zum Einsatz.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de

dtk-redaktion@dante.de (Redaktion)

Druck: Konrad Tritsch Print und digitale Medien GmbH
Johannes-Gutenberg-Str. 1–3, 97199 Ochsenfurt-Hohe Stadt

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)

Luzia Dietsche	Gert Ingold	Volker RW Schaa
Rudolf Herrmann	Rolf Niepraschk	Herbert Voß
Moriz Hoffmann-	Günter Partosch	
Axthelm	Bernd Raichle	

Redaktionsschluss für Heft 3/2004: 1. August 2004

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

hier haben wir nun die erste Farb-Ausgabe der Mitgliedszeitschrift „Die T_EXnische Komödie“. Lange Zeit war die Idee schon vorhanden, bis sie jetzt zum 15-jährigen Jubiläum von DANTE, Deutschsprachige Anwendervereinigung T_EX e.V. verwirklicht werden konnte. Dafür ist diese Ausgabe auch besonders umfangreich geworden.

Dass dies eine Farbausgabe ist, bedeutet nicht, dass jetzt jede Seite in bunt schillernden Farbtönen daherkommt. Auch bei Farbe ist weniger manchmal mehr. Deshalb beschränkt sich der Einsatz von Farbe in diesem Heft im Wesentlichen auf die Stellen, an denen mit der Farbe etwas gezeigt werden soll, während der Rest in klassischem Schwarz gehalten ist.

Es ist heute insbesondere den T_EX-Anwendern auf dem Gebiet der Schriften klar, dass es eine typografische Todsünde ist, beliebige Schriften wild zu mischen. Hier bekommen wir – beispielsweise durch das NFSS in L^AT_EX – einige Hilfe, immer nur zusammenpassende Schriften einzusetzen.

Eine entsprechende Unterstützung für Farben fehlt aber noch weitgehend. Bei diesem Thema sind wir noch in dem Zustand, dass wir beliebige Farben einfach auswählen und Text oder Linien damit auf das Papier bringen können.

Die krassesten Sünden fallen den meisten sofort auf. Der feine Punkt liegt aber auch hier im Detail. Schriftfamilien erlauben es, zusammenpassende Schriften und Schnitte zu kennzeichnen und auszuwählen. In ähnlicher Weise sollte man nur zusammenpassende Farben in einem einzelnen Dokument benutzen. Solche Farben könnte man zu „Paletten“ zusammenfassen.

Paletten sind noch Zukunftsmusik. Trotzdem hoffe ich, dass wir mit dieser Ausgabe dem Ziel einer ansehnlichen Farb-Ausgabe nahegekommen sind, auch, oder gerade weil nicht jede Seite bunt schillert.

Mit T_EXnischen Grüßen

Ihr Gerd Neugebauer

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

dies ist die erste farbige Ausgabe der Zeitschrift „Die T_EXnische Komödie“. Es hatte zwar schon seit Jahren wiederholt Überlegungen gegeben, einzelne Farbseiten aufzunehmen, es blieb aber der Jubiläumsausgabe zum 15. Geburtstag von DANTE e.V. vorbehalten, dies umzusetzen. Wir hoffen, dass durch diese zusätzliche Dimension die Artikel über Farbspielereien und Präsentationen im richtigen Licht erscheinen.

In Ergänzung zu dem Bericht über die Tagung DANTE 2004 in Darmstadt finden Sie in dieser Ausgabe eine kleine Fotogalerie über dieses Ereignis mit Bildern von Teilnehmern und Vortragenden der Tagung und einen Eindruck von der Jubiläumsfeier im „Haus für Industriekultur“.

Sicherlich haben Sie bemerkt, dass die Beilage „typography“ in der Ausgabe 1/2004 von „Die T_EXnische Komödie“ gänzlich unerwähnt blieb. Der Grund hierfür war, dass sich der dieser Ausgabe beiliegende Fontsampller in der Herstellung sehr verzögert hatte und nicht rechtzeitig für den Versand zur Verfügung stand. Der Sonderdruck lag jedoch schon vorzeitig bei der Druckerei bereit und wurde dann so kurzfristig mitversandt, dass keine Änderung von „Die T_EXnische Komödie“ und Anschreiben möglich war. Übrigens befindet sich auf der CD, die dem Sonderdruck von „typography“ für DANTE e.V. beilag, auch eine T_EX-Anpassung für die darin verwendeten Zierschriften – mit Ausnahmen von zwei Schriften, da bei einer Schrift die Lizenz dies nicht zuließ und eine andere die Programme fontinst und afm2tmx „sprengte“

Wir freuen uns, die Herbsttagung von DANTE e.V. in Hannover ankündigen zu können. Mark Heisterkamp und das Regionale Rechenzentrum für Niedersachsen (RRZN) an der Universität Hannover haben sich freundlicherweise bereit erklärt, dieses Treffen auszurichten, welches von Freitag, den 29. Oktober bis Samstag, den 30. Oktober 2004, stattfindet. Für die Frühjahrstagung,

die als gemeinsame Tagung mit der französischen Gruppe GUTenberg als EuroT_EX 2005 stattfinden wird, stehen Zeitpunkt und Ort fest: 7. bis 11. März mit Exkursion am 12. März im Kloster Prémontrés in Pont-à-Mousson, 25 km von Metz/Frankreich entfernt.

Gerne möchten wir auch noch einmal die Gelegenheit nutzen, auf die Tagung TUG 2004 der internationalen T_EX Users Group hinzuweisen. Diese findet vom 30. August bis zum 3. September (direkt nach dem Ende der olympischen Spiele in Athen) in Xanthi im Nordosten Griechenlands statt. Wir hoffen, dass möglichst viele Mitglieder von DANTE e.V. die Gelegenheit, dass die jährliche TUG-Konferenz wieder einmal in Europa stattfindet, nutzen werden.

Diejenigen, die auf der vereinsinternen Mailing-Liste **dante-ev** eingetragen sind, werden mitbekommen haben, dass dort in den letzten Wochen eine rege Diskussion über die Möglichkeiten der Förderung von Projekten durch DANTE e.V. ablief. Wir sind für die Anregungen, die dort kamen, sehr dankbar. Aus unserer Sicht ist der Projektfonds eine Möglichkeit, bestehende Projekte unbürokratisch zu fördern. Dies betraf in der Vergangenheit im Wesentlichen Reisekosten für Entwicklertreffen. Wir sind gerne bereit, bei der Projektförderung auch Kooperationen mit Instituten (beispielsweise in Form der Finanzierung von Studienarbeiten) einzugehen. Wir sehen uns jedoch nicht in der Lage, Projektpläne von uns aus zu definieren, interessierte Projektteilnehmer zu suchen und in erheblichem Maß an der Betreuung der Projekte mitzuwirken. Dies würde einen erheblichen Arbeitsaufwand nötig machen, der wohl nur mit Hilfe eines Planungskomitees zu leisten wäre.

Ein weiterer Diskussionspunkt betraf die WWW-Seiten von DANTE e.V. Leider werden diese Seiten derzeit praktisch nur technisch betreut, während für die inhaltliche Pflege Mitarbeiter fehlen. Wir werden im Vorstand auch über dieses Thema beraten.

Mit freundlichem Gruß,

Volker RW Schaa	Klaus Höppner
Vorsitzender	Stellvertretender Vorsitzender

Umstellung der Kreditkartenakzeptanz bei DANTE e.V.

Klaus Höppner

Vorgeschichte

Seit 1997 bietet DANTE e.V. die Möglichkeit, Mitgliedsbeiträge oder Bestellungen per VISA-Karte zu bezahlen. Die Zahlungen werden über ein Kreditkartenterminal abgewickelt, wie Sie es aus vielen Geschäften und Tankstellen kennen dürften – mit dem Unterschied, dass mangels vorliegender Karte die Kartendaten von Hand eingetippt werden.

Da Zahlungen per Kreditkarte den Empfänger Provision kosten (für einen normalen Mitgliedsbeitrag derzeit 2 €), ist diese Zahlungsmethode als Ausnahme gedacht und wird von uns daher mit einer Extragebühr von 2,50 € belegt. Das Angebot ist vorrangig für Mitglieder gedacht, die außerhalb Deutschlands leben und daher ansonsten hohe Bankgebühren für Überweisungen zahlen müssten.

Seit Mitte 2003 sind Überweisungen innerhalb der Euro-Zone zu Inlandspreisen möglich. Daher ist die Zahl von Kreditkartenzahlungen stark zurückgegangen. Leider bezahlen wir neben der oben erwähnten Provision auch eine Grundgebühr für das Kartenterminal. Da es sich abzeichnet, dass wir nur noch etwa 10 Kartenzahlungen pro Jahr haben werden, würde jede Zahlung effektiv über 10 € kosten.

Aus diesem Grund haben wir den bestehenden Vertrag nicht verlängert, sodass Zahlungen nach dem bisherigen Verfahren nur noch bis Ende Mai 2004 möglich waren.

Alternativen

Wir haben uns daher nach alternativen Anbietern umgesehen. Da für alle konventionellen Anbieter von Kartenterminals das Problem mit der Grundgebühr besteht, haben wir uns auf Bezahldienste im Internet konzentriert.

Klassiker in diesem Bereich ist Paypal, mittlerweile Teil des Ebay-Konzerns. Paypal ermöglicht die Akzeptanz von Kreditkartenzahlungen zu günstigen

Konditionen. Allerdings unterliegt es als amerikanisches Unternehmen keinerlei staatlicher Regulierung, die Geschäftsbedingungen sind äußerst undurchsichtig, und es häufen sich die Klagen in Internet-Foren über willkürlich eingefrorene Konten. Daher haben wir uns gegen Paypal entschieden.

Wir werden zukünftig die Kreditkartenzahlungen über den englischen Anbieter Moneybookers (<http://www.moneybookers.com>) durchführen, der von der englischen Finanz- und Kapitalmarktaufsicht FSA nach der EU-Richtlinie zur Ausgabe digitalen Geldes lizenziert ist. Für das Konto bei Moneybookers fällt keine Grundgebühr an, allerdings sind die Gebühren pro Zahlung mit 8 % hoch, was bei der derzeitigen Zahl von Kreditkartenzahlungen aber immer noch deutlich günstiger ist als das bisherige Verfahren. Nebenbei sind zukünftig neben VISA auch Zahlungen mit Euro-/Mastercard möglich.

Praktisches Verfahren

Für die Zahlung über Moneybookers benötigen Sie, neben einer Kreditkarte, noch einen Internetzugang mit WWW-Browser und ein SMS-fähiges Handy.

Erfüllen Sie diese Voraussetzungen, können Sie loslegen:

1. Gehen Sie auf die Seite <http://www.dante.de/dante/moneybookers/>
2. Hier finden Sie das Verfahren noch einmal genau beschrieben.
3. Klicken Sie auf den Link zum Zahlungsformular.
4. Geben Sie Ihren Namen, Mitgliedsnummer, Betrag und Verwendungszweck ein.
5. Nachdem Sie die Richtigkeit Ihrer Angaben noch einmal bestätigt haben, werden Sie über eine verschlüsselte Verbindung auf die Seite von Moneybookers umgeleitet.
6. Falls Sie noch kein Konto bei Moneybookers haben, müssen Sie Ihre Anschrift, E-Mail-Adresse, Kreditkartennummer und Handy-Nummer eingeben und sich ein Passwort für Ihr Benutzerkonto aussuchen. (Merken Sie sich die E-Mail-Adresse und das Passwort, mit dem Sie sich angemeldet haben, da Sie sich hierüber in Ihr Konto einloggen können – wir empfehlen, dies mindestens einmal zu tun, um in Ihrem Nutzerprofil den Newsletter abzubestellen).

Falls Sie bereits ein Benutzerkonto bei Moneybookers besitzen, loggen Sie sich einfach mit E-Mail-Adresse und Passwort ein.

7. Sie müssen nun die so genannte CVV-Nummer Ihrer Kreditkarte eingeben. Dies ist eine drei- bis vierstellige Zahl auf der Rückseite der Karte, die nicht auf dem Magnetstreifen der Karte gespeichert ist und auch bei Moneybookers nicht gespeichert wird.
8. Weiterhin erhalten Sie auf Ihr Handy eine SMS mit einem Freischaltcode für die Zahlung.
9. Nach Eingabe dieses Codes können Sie Ihre Zahlung dann abschließen.
10. Auf Ihrer Kreditkartenabrechnung erscheint als Empfänger Moneybookers, nicht DANTE e.V.

Für diese Kreditkartenzahlungen wird auf den Rechnungsbetrag automatisch eine Extragebühr von 2 € aufgeschlagen.

Billiger (sowohl für Sie als auch für uns) wird es, wenn Sie bereits ein Konto bei Moneybookers besitzen und in einem der etwa 30 Länder wohnen, in denen Moneybookers eine lokale Bankverbindung hat. In diesem Fall können wir das Geld von Ihnen per E-Mail anfordern, und Sie überweisen dann den Betrag per Inlandsüberweisung an Moneybookers. Diese Methode kostet Sie (neben den Bankgebühren für eine Inlandsüberweisung) eine Gebühr von 1 % der Zahlungssumme, maximal 0,50 €. Details zu dieser Methode erhalten Sie auf der oben genannten WWW-Seite oder per E-Mail von dante@dante.de.

Feedback

Obiges Verfahren ist ein Test. Wir haben es nur in Deutschland getestet und wissen daher nicht, wie zuverlässig das Verfahren mit SMS-Empfang etc. außerhalb Deutschlands funktioniert. Daher sind wir für Rückmeldungen dankbar und werden uns notfalls auf die Suche nach einem anderen Anbieter machen.

Tagungsbericht Darmstadt 2004: Im Zeichen der 15

Falk Gerwig, Martin Müller, Jan Theofel

15 Jahre – oder das Jahr „davor“

Als binärer Verein steht man immer vor dem Dilemma, entweder die runden Geburtstage im Dezimal- oder die runden im Binärsystem zu feiern. Die pragmatische Lösung ist es, einfach beide Jahrestage entsprechend zu würdigen.

Dieses Jahr wird DANTE e. V. 15¹ Jahre alt, was auf der Frühjahrstagung 2004 in Darmstadt gebührend gefeiert wurde. Dieser Tagungsbericht soll eine traditionell persönliche Darstellung der Autoren wiedergeben.

Die ersten 15 Stunden – oder etwas weniger

Für die Anreise aus Stuttgart haben wir uns für das Auto entschieden, um auch vor Ort etwas mobiler zu sein. Da wir in Mörfelden-Walldorf untergebracht waren, war dies sicher eine weise Entscheidung.

Im Unterschied zum Zug ist das Essen im Auto allerdings etwas ungeschickt, sodass wir vorher stehend unser erstes Tagungessen zu uns nahmen. Zum Nachtisch gab es dann dicke Suppe auf der Autobahn.

Diese wurde passend zur Autobahnausfahrt hin klarer, aber durch die zwei Ortsteile Mörfelden und Walldorf war der Weg dennoch nicht zu finden. Nachdem wir mehr durch Glück als durch zielgerichtete Navigation schließlich unsere „bed&breakfast“-Unterkunft gefunden hatten, haben wir uns noch im Supermarkt mit Lebensmitteln eingedeckt. Schließlich war sie auf unseren Wunsch hin eher „bed-without-breakfast“.

Frisch bevorratet haben wir uns dann nach Darmstadt aufgemacht, um dort noch ein paar Kleinigkeiten einzukaufen und schließlich beim Vorabendtreffen im Restaurant „Poseidon“ auf die ersten anderen Tagungsteilnehmer zu treffen.

¹ Oder um es anders auszudrücken: $2^{2^2} - 1$. Nächstes Jahr findet dann der besonders runde 16. Geburtstag statt: 2^{2^2}

15 „Torturen“ während der Tutorien

Der zweite Tag bestand aus Torturen und Tutorien, wobei letztere geplant waren und erstere selbst durch die perfekte Organisation nicht verhindert werden konnte. Widrige Umstände gibt es eben immer – vor allem wenn Technik im Spiel ist.

Für uns ging es damit los, dass wir viel zu früh aufstehen mussten – *die erste Tortur*. Ein Bad für vier Personen, selbst Frühstück richten und die Anfahrt konnten als Hauptschuldige ausgemacht werden. Womit wir auch schon bei der Suche nach einem Parkplatz und dem Gebäude wären – *die zweite Tortur*.

Dadurch kamen wir auch etwas zu spät und haben uns erst mal im Tagungsbüro unsere „Hundemarken“ und Jubiläums-Tassen abgeholt. Eilig bewegten wir uns in den Vortragsraum, wo David Kastrup statt mit seinem Thema mit einer kaputten Tafel kämpfte – *die dritte Tortur*. Nachdem der Kampf mehr oder weniger erfolgreich von ihm beendet wurde, stellte er seine Fußnoten-Torturen (*Nummer vier*) vor und führte die anwesende T_EX-Gemeinde in die Interna von T_EX ein. Im Laufe des Vortrags konnten wir unsere Tassen auch mal eingehender anschauen und mussten dabei feststellen, dass sich eine Tasse der Tafel anschloss und auch kaputt war – *die fünfte Tortur*.

Es folgte die Kaffeepause, die aufgrund lächerlicher Brandschutzbestimmungen nicht im kilometerbreiten Vortragsvorraum durchgeführt werden durfte. Statt dessen wurden alle Teilnehmer in das Tagungsbüro gepfercht – *die sechste Tortur*. Dort hatte man gestattet, Kaffee und Kekse aufzubauen.

In der Pause versuchten auch die ersten Teilnehmer, sich an das Tagungsnetz anzukoppeln, was aufgrund sehr restriktiver Firewalls des Rechenzentrums scheiterte. Ein Vormittag ohne E-Mail und ohne Heise-Newsticker ... – *die siebte Tortur*. Dank der Organisatoren konnte dieses Manko aber schnell behoben werden. Auch die beiden fest installierten PCs waren zunächst nicht nutzbar – *die achte Tortur*. Erst als die Jalousie herabgelassen wurde, konnte man auf den Monitoren erkennen, was man gerade tat. Diese Lösung wurde natürlich auch schnell gefunden.

Nachdem die Rechner-Probleme gelöst waren, hielten Luzia Dietsche und Bernd Raichle ihren Vortrag über das Schreiben von L^AT_EX-Klassen. Ab und zu wurden sie von ihrem kleinen David unterbrochen, der sich nicht davon abhalten ließ, zu seiner Mutter zu wollen. Wir haben ihm zwar erklärt, dass sie arbeitet, aber er wollte unbedingt mitarbeiten. Aber David gehört natürlich

zu den ganz besonderen persönlichen Kontakten der Tagung und geht daher nicht in die Wertung ein.

In der Zwischenzeit hatten die meisten Teilnehmer auch ihre persönlichen Erfahrungen mit den Toiletten gesammelt. Die ausgeschilderten Toiletten waren leider nur zwischen 12:00 Uhr und Mittag geöffnet, sodass man auf die unverschlossenen Alternativen ausweichen musste. Diese zu finden war in der Tat *die neunte Tortur*.

Zum Mittagessen wurden uns mehrere Möglichkeiten nahe der Uni angeboten. Unsere Wahl der Qual – *die zehnte Tortur* – fiel auf einen Taiwanesen mit eindeutig auffindbaren Toiletten, bei dem man nebenbei auch vom All-You-Can-Eat-Büffet essen konnte.

Frisch ’ ’gestärkt ’ ’ wurden wir von Peter Reichard darüber aufgeklärt, dass Typografie BenutzerInnen-freundlichkeit ist. Unter anderem wurden uns einige Sünden aufgezeigt und einige Regeln genannt...

Dabei wurden uns auch die <<extrem wichtigen>> AUSZEICHNUNGEN und sehr elegante S p e r r u n g e n vorgestellt. Die korrekte Darstellung von Kapitälchen in GROßBUCHSTABEN ist natürlich auch wichtig. Des Weiteren ist die Wahl der **richtigen** Schriftart von enormer Bedeutung. Die Typo’s-ünden wurden uns in realen Beispielen² am Beamer gezeigt - **die elfte Tortur**.

Es folgte die obligatorische Kaffeepause, nach der dann eine weitere Wahl zu treffen war, weil das Vortragsprogramm zweigeteilt wurde – *die zwölfte Tortur*. Wir entschieden uns mutig als streng gläubige vi-Jünger in David Kas-trups Vortrag „Emacs für L^AT_EX-Autoren“ zu gehen – *die dreizehnte Tortur*. Überzeugen ließen wir uns allerdings nicht.

Vor dem Abendtreff wollten wir noch einige Besorgungen erledigen. Aber leider scheinen sich die verlängerten Ladenöffnungszeiten noch nicht bis Darmstadt herumgesprochen zu haben. Um Punkt 19:00 Uhr wurde man aus den Läden hinauskomplimentiert – *die vierzehnte Tortur*.

Schließlich ließen wir den Abend gemütlich im Ratskeller ausklingen. In gewohnt lockerer Runde konnten wir uns mit unseren T_EX-Freunden unterhalten und neue Leute kennen lernen. Ach ja, es fehlt ja noch eine Tortur: Die Pasta-Karte war den Teilnehmern vorbehalten, die früh kamen. Und so

²Einige wurden hier wiedergegeben und der geneigte Leser wird sicher keine Schwierigkeiten haben, die 15 Sünden in diesem Absatz zu finden.

konnte nicht jeder die vom Hörensagen gute Pasta genießen. Eine ungerechte *letzte, fünfzehnte Tortur*.

15 Reimplementierungen von $\text{T}_{\text{E}}\text{X}$?

Nach einer wieder viel zu kurzen Nachtruhe begann der nächste Morgen mit der 30. Mitgliederversammlung. Nach einer kurzen Eröffnungsbegrüßung von Klaus Höppner spannte Prof. Jürgen Lehn, Dekan des Fachbereichs Mathematik der TU Darmstadt, einen besonders gelungenen Bogen von der Universität Darmstadt bis zu DANTE. Er zeigte uns zunächst eine sehr direkte Verknüpfung auf, um dann doch den längeren Weg über Fibonacci, den Goldenen Schnitt, ein griechisches Theater und Donald E. Knuth aufzuzeigen.

Im Anschluss wurden die verschiedenen Tagesordnungspunkte behandelt. Zum Inhalt der Versammlung möchten wir auf das Protokoll verweisen. Als besonders bemerkenswert möchten wir noch erwähnen, dass die Mitgliederversammlung trotz angeregter Diskussionen pünktlich beendet werden konnte.

Durch kurzfristige Änderungen am Vortragsprogramm kam es bei einigen Teilnehmern zu Irritationen. Ein Vortrag entfiel und die restlichen wurden so umgestellt, dass wir am nächsten Morgen zu unserer Erleichterung nicht eine halbe Stunde früher anfangen mussten.

Trotz dieser Umstellungen erschienen nach der Mittagspause alle Teilnehmer pünktlich zum Vortrag. Frank Schäfer referierte über das von ihm entwickelte $\text{ŠäferT}_{\text{E}}\text{X}$. Dieses stellt eine Dokumentbeschreibungssprache dar, die zum eigentlichen Satz dann in $\text{T}_{\text{E}}\text{X}$ umgewandelt wird. Ziel seiner Entwicklung ist es, die Arbeit von Autoren durch eine einfachere Beschreibungssprache zu erleichtern.

Es folgte die Vorstellung von $\epsilon\chi\text{T}_{\text{E}}\text{X}$ durch Michael Niedermaier und Markus Kohm. Dieses Projekt, durch den Projektfonds von DANTE e.V. unterstützt, war ursprünglich als Weiterentwicklung und Optimierung von $\mathcal{N}\mathcal{T}\mathcal{S}$ gedacht, entwickelte sich aber schnell zu einer völlig eigenständigen Implementierung.

Bevor wir uns in der nachmittäglichen Kaffeepause stärken durften, stellte uns Volker RW Schaa seine Lösung zum Erstellen von Konferenz-Proceedings vor. Er zeigte uns, wie man mit einfachen Mitteln Open-Source-Software einsetzen kann, um diese recht komplexe Aufgabe zu bewältigen.

Nach der Stärkung durch E-Mails und Kaffee folgte Hartmut Henkels Vortrag über das Font-Mapping von Pdf \TeX . Er zeigte, welche Besonderheiten zu beachten sind, um auch bei Pdf \TeX qualitativ gute Ergebnisse zu erzielen.

An diesen Vortrag schloss sich die Präsentation einer weiteren Reimplementierung von \TeX an. \aleph (Aleph) wird ebenfalls durch den Projektfonds von DANTE e.V. unterstützt und ist eine Implementierung, die große Teile von Omega und ε - \TeX abzubilden versucht.

Hans Hagen schloss den Nachmittag mit einem Einblick in die Multimedia-Möglichkeiten der neuen PDF-Version. Sehr zur Freude unserer jüngeren Teilnehmer wählte er hierzu auch Ausschnitte aus „Findet Nemo“, um die Einbettung von Videos in PDF zu demonstrieren.

Bei der Fahrt zum Tagungssessen im „Weißen Schwan“ fragten wir uns nicht, ob, sondern wann wir die Reimplementierung Nummer 15 nach $\mathcal{N}\mathcal{T}\mathcal{S}$ erleben dürfen. Das Abendessen war gewohnt gut und das Büffet reichlich – auch wenn es zögerlich aufgetragen wurde. Etwas ungewohnt war die prompte „Abkassierung“ nach jedem Getränk. In sieben Jahren Berufserfahrung hatte unsere Bedienung gelernt, dass es so besser sei (und vermutlich mehr Trinkgeld gibt). Harald König hielt dem nur entgegen, dass wir bereits 15 Jahre Erfahrung haben und damit sicher noch nie Probleme hatten.

Fast 15

Am nächsten Morgen stellte Gerd Neugebauer die Innereien von Bib \TeX vor, die er im Laufe seiner Reimplementierung von Bib \TeX kennengelernt hatte. Die darin enthaltenen Schwachstellen sollen durch seine oder potentielle weitere Reimplementierungen verbessert werden.

Es folgte ein Trio von Vorträgen über Grafiken in und um \TeX . Den Anfang machte Georg Lachenmayr mit seiner Frage, ob Pic \TeX noch zeitgemäß sei. Das Resümee seines Vortrags war ein klares „Ja“, was er auch mit eindrucksvollen Pic \TeX -Grafiken belegten konnte.

Als Alternative folgte die Vorstellung neuer Tricks mit PStricks durch Rolf Niepraschk und Herbert Voß. Auch hier gab es eine Reihe von sehr ansprechenden und komplexen Grafiken zu bestaunen.

Das Trio schloß mit André Wobsts Vortrag über sein Projekt $\mathbb{R}\mathbb{X}$. Dieses nutzt \TeX zur Erzeugung von Grafiken und setzt als Programmiersprache

Python voraus, da es in dieser Sprache implementiert ist und die Eingabe selbst auch als Python-Code erfolgt.

Nach der Mittagspause folgte ein weiterer Vortrag über die allgemein bekannte Wollmilchsau AucTeX und preview-latex wiederum durch David Kastrop. Dabei wurden vor allem neue Features dieser speziellen Emacs-Umgebung für TeX vorgestellt. Aber auch damit schaffte er es nicht, uns von Emacs als Editor zu überzeugen.

Hiermit endete das Vortragsprogramm, da noch der Festakt „15 Jahre DAN-TE e.V.“ anstand. Dieser fand im Haus für Industriekultur in Darmstadt statt. Obwohl der eigentliche Geburtstag noch nicht ganz anstand, er ist eigentlich erst am 14. April, wurde dieser gebührend gefeiert. Zum fast fünfzehnten Geburtstag waren auch fast 15 Gründungsmitglieder und die Ehrenmitglieder Luzia Dietsche und Hermann Zapf anwesend.

Hans Hagen hielt seine Geburtstagsrede mit Grüßen vom TUG-Präsidenten Karl Berry in Form einer PDF-Multimedia-Show mit vielen (lustigen) Fakten über die letzten 15 Jahre. Danach ehrte Volker RW Schaa die anwesenden Gründungsmitglieder, die alle ihren persönlichen Weg zum Verein preisgaben. Sie hoben dabei die TeX-Community hervor, deren besonderer Geist auch bei dieser Tagung wieder zu spüren war. Auch die lokalen Organisatoren der diesjährigen Tagung wurden mit Lob und Geschenken bedacht. Anschließend wurden die beiden mit einem TeX-Kreuzworträtsel verzierten Geburtstags-torten angeschnitten und verteilt.

Sehr gut in den festlichen Rahmen passte die folgende Besichtigung der Druckerausstellung und Schriftgießerei im Haus. Die Ausstellung beeindruckte vor allem durch die praktisch vorgeführten Darstellungen des Satzsetzes mit Bleiletern.

Danach klang die Tagung beim gemütlichem Beisammensein im „Braustüb“ aus und die Teilnehmer machten sich nach und nach auf die Heimreise.

15 Stunden in 15 Zeilen

Wegen unserer Unterkunft nahe Frankfurt entschieden wir uns, nicht an der Fahrt zur Grube Messel am Samstag teilzunehmen, sondern statt dessen nach Frankfurt zu fahren und die Stadt zu besichtigen. Diese Besichtigung bestand zunächst aus der Suche nach einem Sushi-Restaurant. Als wir dieses gefunden hatten, suchten wir ein nahe gelegenes Café auf, wo wir begannen, den ersten

Teil dieses Tagungsberichts niederzuschreiben. Nach ausführlichem Genuß der japanischen Esskultur besichtigten wir zahlreiche Konsumtempel entlang der Fußgängerzone. Schwer beladen schafften wir den Rückweg zu unserem Auto nur durch eine Starbucks-Stärkung.

Die Unterbringung all unserer Einkäufe (eine komplette, neue Küchenausstattung) im Auto gestaltete sich schwieriger als erwartet, aber schließlich konnten wir auch dieses 3D-Puzzle lösen. Durch die nicht ganz so dicke Suppe wie auf der Hinfahrt traten wir schließlich unsere Heimreise an. Zuhause angekommen vollenden wir nun diese fast letzten 15 Zeilen, um noch einmal die entscheidende Zahl 15 angemessen hervorheben zu können.

Resümee

Die Tagung hat wie immer viel Spaß gemacht und für uns viele neue Dinge bereitgehalten. Neben dem Austausch von T_EXnischem kam natürlich auch der Geist der T_EX-Community bei den vielen gemeinsamen Veranstaltungen zur Geltung. Die kulinarische Reise der letzten Jahre haben wir dabei nahtlos fortgesetzt und dieses Jahr gebührend den fünfzehnten Geburtstag von DANTE e.V. gefeiert.

Wir möchten im Namen aller Teilnehmer den Organisatoren und zahlreichen Helfern für die perfekte Organisation danken. Ein besonderer Dank gilt natürlich auch den Gründungsmitgliedern, ohne die wir diese Tagung nicht gehabt hätten.

Abschließend bleibt uns nur zu sagen: Auf ein Wiedersehen zur Herbsttagung und eine Steigerung zum richtigen 16. Geburtstag zusammen mit GUTenberg in Pont-à-Mousson 2005.

Eine Bildergalerie zum $(2^{2^{2^0}} - 1)$. Geburtstag

Liebe Mitglieder,

die folgende Bildergalerie soll Ihnen einen kleinen Überblick über die Tagung, die Abendveranstaltungen und die Jubiläumsfeier zum 15. Geburtstag von DANTE e.V. im „Haus für Industriekultur“ mit dem sich anschließenden Besuch in den Ausstellungsräumen des Museums verschaffen. Wer sich selbst ein Bild von dem Museum machen möchte, sei auf die Web-Seiten des Hessischen Landesmuseums Darmstadt mit der Außenstelle „Spezialmuseum für Drucktechnik“ (www.hlmd.de/museum/muaushik.html) verwiesen.

Die Bilder wurden dankenswerterweise von Markus Ackermann, Karlheinz Geyer, Walter Obermiller, Bernd Raichle, Volker RW Schaa, Martin Schröder und Ulrik Vieth zur Verfügung gestellt.

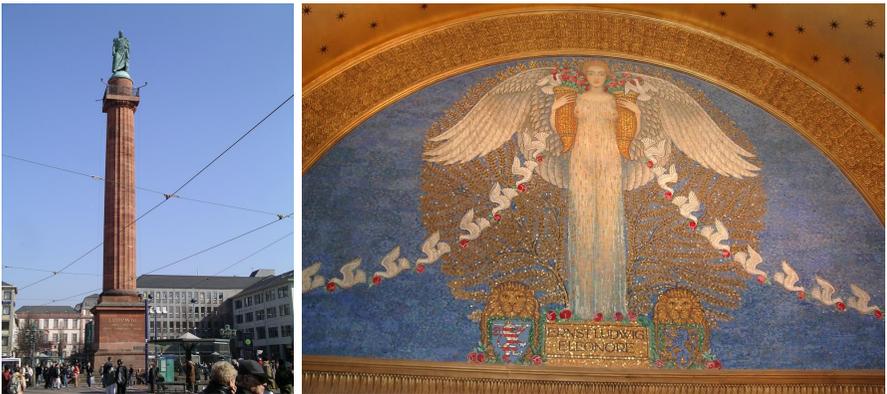


Abbildung 1: Tagung von DANTE e.V. in Darmstadt, Stadt des Jugendstils.



Abbildung 2: Der 15. Geburtstag von DANTE e.V. war ein Anlass, zum ersten Mal einen DANTE-Becher zu produzieren (es gibt in der Geschäftsstelle noch einige zu erwerben). Grafisches Design: Bernd Raichle.



Abbildung 3: Die ersten weit gereisten Gäste kommen früh (von links): Jim Hefferon (CTAN-Team), USA (Ankunft 07:15); Kim Roberts (UK-TUG Board), Großbritannien; Staszek Wawrykiewicz (TeXLive-Team) und Jerzy Ludwiczowski (GUST Board), Polen.



Abbildung 4: Die Registrierung beginnt: der $\text{T}_{\text{E}}\text{X}$ -Nachwuchs hilft mit – David mit Karin Dornacher.



Abbildung 5: Robin Fairbairns (CTAN-Team) im Gespräch mit Jim Hefferon, Kim Roberts und Manfred Lotz ($\text{T}_{\text{E}}\text{X}$ -Collection) – bevor es wieder an das Einchecken von neuen Paketen auf CTAN geht ...



Abbildung 6: Fabrice Popineau (fp \TeX , \TeX Live-Team) nutzt die Zeit vor dem Treffen der CTAN-Maintainer, der \TeX Live-, \TeX -Distributions- und \LaTeX -Teams zum E-Mail-Check, Joachim Schrod und Chris Rowley (\LaTeX -Team) harren der Dinge ...



Abbildung 7: Vor einem Brain-Storming mit Entwicklern zeigt Hans Hagen (Con \TeX t) Karel Skoupý ($\mathcal{N}\mathcal{T}\mathcal{S}$) die Höhe eine `baselineskips`; im rechten Bild links: Johannes Küster (Typoma, Latin-Modern-Math-Fonts), rechts: Guiseppa Biotta (\mathfrak{R} , $e\Omega$).

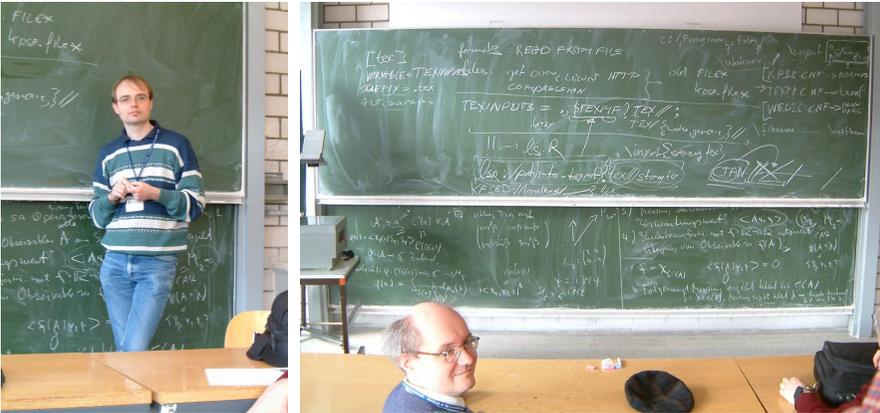


Abbildung 10: Was ändert sich mit dem neuen web2c und welche Auswirkungen hat es auf $\text{T}_{\text{E}}\text{XLive}$ und $\text{T}_{\text{E}}\text{XCollection 2004}$? Olaf Weber skizziert die Antwort an der Tafel.



Abbildung 11: Die Abende werden für weiterführende Diskussionen genutzt, Konzepte niedergeschrieben: Jerzy Ludwichowski mit Manfred Lotz und Staszek Warykiewicz. Oder wird der Zeitplan für den $\text{T}_{\text{E}}\text{XLive 2004-Release-Candidate 1}$ durchgesprochen? Thomas Esser mit Fabrice Popineau, Olaf Weber und Marcel Herbst.



Abbildung 12: Abend stilvoll (von links): Robin Fairbairns, Jim Hefferon, Frank Mittelbach und Kim Roberts.



Abbildung 13: Drei sprachlose Präsidenten (von links): Jerzy Ludwichowski (GUST), Hans Hagen (NTG) und Volker RW Schaa (DANTE e.V.). „Mister X“ immer gut gelaunt (rechts): Guiseppe Bilotta.



Abbildung 14: Hans Hagen, Thomas Esser und Sebastian Rahtz.



Abbildung 15: Der Schatzmeister erstattet in der Mitgliederversammlung seinen Finanzbericht: Tobias Sterzl.

Die Kassenprüfer bleiben skeptisch: Knut Lickert und Harald König mit Volker RW Schaa.



Abbildung 16: Die Neuwahl des Präsidiums, durchgeführt und beaufsichtigt durch Wahlleiter und AssistentIn (von links): Karlheinz Geyer, Jörg Knappen und Kerstin Schiebel.

Sichtliche Freude beim wiedergewählten Vizepräsidenten Klaus Höppner.



Abbildung 17: Am Abend folgt das Bankett zum 15. Geburtstag im „Weißen Schwan“ mit Darmstädter „Weißbier“, Fabrice (rechts) greift lieber zum „Schwarzen“.



Abbildung 18: Bei gelöster Stimmung wartet man auf das Startsignal zum Gang an das Büffet (von links): Harald König, Klaus Höppner, Thomas Feuerstack und Karin Dornacher.



Abbildung 19: Jubiläumsfeier im „Haus für Industriekultur“ mit zehn Gründungsmitgliedern: (von rechts) Anreas Daffener (Mitgliedsnummer 8), Friedhelm Sowa (3), Joachim Lammarsch (1), Luzia Dietsche (4), Klaus Guntermann (10), Joachim Schrod (11), Christine Detig (12), Klaus Braune (als Vertreter des Rechenzentrums der Universität Karlsruhe, 13), Harald König (14), Frank Holzwarth (als Vertreter des Springer-Verlags, 17).



Abbildung 20: Für viele Jahre das starke Team von DANTE e.V. (von links): Luzia Dietsche als Schriftführerin, Joachim Lammarsch als Präsident und Friedhelm Sowa als Schatzmeister.



Abbildung 21: Gründungsmitglieder, Vorstand und Ehrenmitglieder; (von links) K. Braune, J. Lammarsch, K. Guntermann, H. König, V. Schaa, Chr. Detig, K. Höppner, T. Sterzl, Prof. H. Zapf, B. Raichle, J. Schrod, L. Dietsche, F. Sowa, G. Partosch, A. Dafferner und F. Holzwarth.



Abbildung 22: Zum Abschluss die Ehrung des lokalen Organisationsteams: (von rechts) Holger Grothe, Karlheinz Geyer, Klaus Höppner, Volker RW Schaa.



Abbildung 23: Beginn der Feierlichkeiten mit Löwen-Torten, grafisches Design: Klaus Höppner.



Abbildung 24: Kritische Augen begleiten das „Zerlegen“ der Torte: Prof. Hermann Zapf, Volker RW Schaa (am Messer), Luzia Dietsche, Jerzy Ludwichowski und Ulrik Vieth; dem sehen Fabrice Popineau, Robin Fairbairns, Guiseppe Bilotta und Jim Hefferon gelassen entgegen ...



Abbildung 25: Auch Jörg Knappen, Walter Schmidt und Frank Mittelbach schauen interessiert herüber, während Manfred Lotz Sekt nachschenkt.



Abbildung 26: Luzia scheint das richtige Stück erwischt zu haben, wie sie gerade Karlheinz Geyer und Prof. Zapf zeigt.

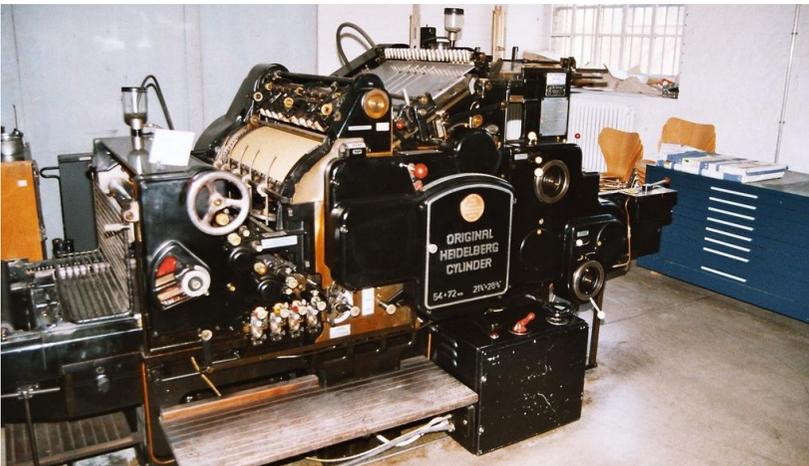


Abbildung 27: Nach der Stärkung beginnt die Führung durch das Museum – ein „Original Heidelberg Cylinder“ direkt im Eingangsbereich.



Abbildung 28: Werbung für die Linotype-Setzmaschinen – in dieser Druckerei aber „Fünf gegen Eine“!



Abbildung 29: Die Utensilien für den Schriftguss: Handgießinstrument zum Probe-gießen, gegossene Buchstaben mit Angusszapfen, justierte Matern, ...



Abbildung 30: Ein Blick auf die Schriftregale mit Setzkästen, die aneinandergereiht ca. 150 laufende Meter ergeben.



Abbildung 31: Rechts eine reichdekorierete Handpresse „Columbia Press“ von 1825.

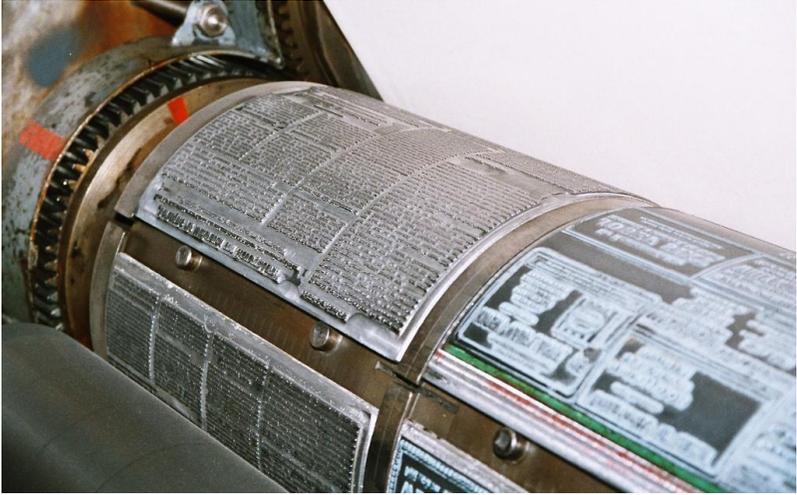


Abbildung 32: . . .

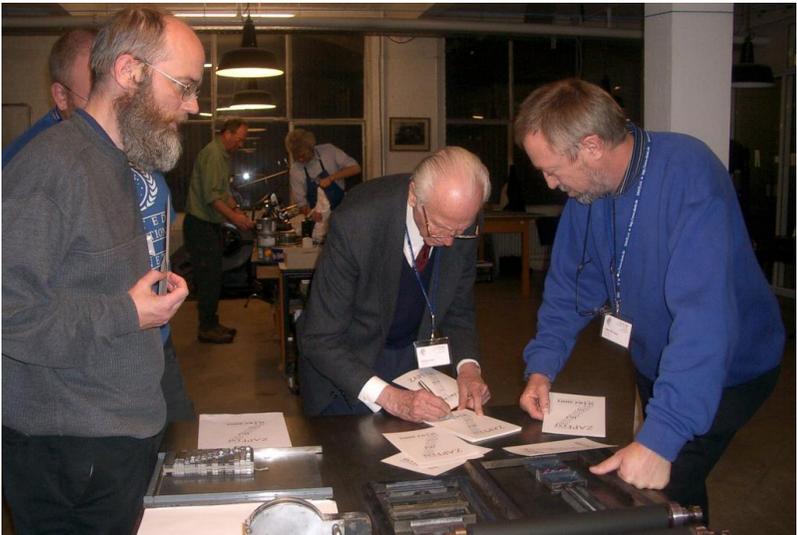


Abbildung 33: Zum Abschluss signiert Prof. Hermann Zapf Druckmuster seiner Schriften, Frank Mittelbach und Volker RW Schaa gedulden sich . . .

Einladung zur Herbsttagung und 31. Mitgliederversammlung von DANTE e.V.

Volker RW Schaa, Mark Heisterkamp

Hiermit laden wir Sie herzlich zur Herbsttagung und 31. Mitgliederversammlung von DANTE e.V. ein. Die Tagung wird vom Regionalen Rechenzentrum für Niedersachsen gemeinsam mit DANTE e.V. ausgerichtet und findet am Freitag, den 29. Oktober, und Samstag, den 30. Oktober 2004 am

Regionales Rechenzentrum für Niedersachsen (RRZN)
Universität Hannover
Schloßwender Str. 5
30159 Hannover

statt.

Für Tagung und Mitgliederversammlung ist folgender Zeitplan vorgesehen:

Donnerstag, 28. Oktober:	Vorabendtreff
Freitag, 29. Oktober:	Tutorien; Abendtreff
Samstag, 30. Oktober, 10.00 Uhr:	Mitgliederversammlung von DANTE e.V.

Die Tagesordnung der Mitgliederversammlung am Samstag, den 30. Oktober 2004, um 10.00 Uhr lautet:

1. Begrüßung; Vorstellung des Vorstands; Tagesordnung
2. Bericht des Vorstands
3. Anpassung der Mitgliedsbeiträge
4. Erhöhung der Projektmittel des T_EX-Projektfonds
5. Verschiedenes

Wie üblich sind auch Nichtmitglieder als Gäste der Mitgliederversammlung willkommen.

Weitere Informationen finden Sie unter

<http://www.dante.de/dante/events/mv31/>

Bitte melden Sie sich mit dem Formular zu den Tutorien und der Mitgliederversammlung an, Sie erleichtern uns dadurch die Vorbereitung der Unterlagen. Mitglieder, die über keinen Internetzugang verfügen, mögen sich bitte an das Büro wenden. Wir werden Ihnen die weiteren Informationen dann per Post zusenden.

Die Adresse für schriftliche Anmeldungen und weitere Fragen lautet:

Mark Heisterkamp
Regionales Rechenzentrum für Niedersachsen (RRZN)
Universität Hannover
Stichwort: Mitgliederversammlung DANTE e.V.
Schloßwender Str. 5
30159 Hannover

oder per E-Mail an

`mv31@dante.de` (bevorzugt)

Wir würden uns freuen, Sie zahlreich auf der Mitgliederversammlung begrüßen zu dürfen.

Mit freundlichem Gruß,

Volker RW Schaa (DANTE e.V.)

Mark Heisterkamp (Universität Hannover, RRZN)

Bretter, die die Welt bedeuten

Farbspielereien in \LaTeX mit dem `xcolor`-Paket

Uwe Kern

Das `xcolor`-Paket ermöglicht die einfache Nutzung verschiedener Arten von Farbschattierungen und -mischungen sowie die Erzeugung von Farbenfolgen und alternierend gefärbten Tabellenzeilen. Es stellt weiterhin Routinen für die (manuelle oder automatische) Konvertierung zwischen verschiedenen Farbmodellen zur Verfügung. Dieser Artikel erläutert anhand von Beispielen Anwendungen des Paketes.

Einleitung

Das `color`-Paket [2] stellt eine Reihe nützlicher Funktionen zur konsistenten und treiberunabhängigen Handhabung von Farben in (pdf) \LaTeX zur Verfügung. Es unterstützt dabei, allerdings nicht ganz treiberunabhängig, mehrere Farbmodelle.

Gleichwohl ist der Umgang mit Farben in (pdf) \LaTeX manchmal etwas umständlich, insbesondere wenn leichte Farbveränderungen, Mischfarben oder gar Konvertierungen ins Spiel kommen: Hier ist man häufig auf andere Programme angewiesen, welche die erforderlichen Parameter berechnen, die anschließend in einen `\definecolor`-Befehl hineinkopiert werden. Hin und wieder kommt auch der gute alte Taschenrechner zum Einsatz, wenn es um Aufgaben der folgenden Art geht:

- Beim Umgang mit Druckereien lernt man, dass es wesentlich preisgünstiger sein kann, eine speziell definierte Farbe (*spot color*) – etwa aus einem Farbfächer – in verschiedenen Helligkeitsstufen zu verwenden, als auf die Standarddruckfarben *cyan*, *magenta*, *yellow* und deren Mischungen zurückzugreifen. Wie sieht nun aber die 75%-Version der mühsam ausgesuchten Spezialfarbe aus?

- Man sieht eine gefällige Farbe, die man gerne selbst verwenden möchte. Leider ist sie im hsb-Modell definiert, man selbst arbeitet jedoch mit pdfL^AT_EX ohne hsb-Unterstützung. Was tun?
- Wie sieht eigentlich eine Mischung aus 40% *green* und 60% *yellow* aus? (Antwort: 40%  + 60%  = )
- Und wie sieht die Komplementärfarbe davon aus? (Antwort: )
- Die Druckerei verlangt, dass alle Farbdefinitionen im Dokument in das cmyk-Format transformiert werden sollen. Wie kann man die Berechnungen effizient durchführen?
- Die 50 Zeilen einer Tabelle sollen alternierend gefärbt werden. Geht das auch einfacher, als 50 Kopien des `\rowcolor`-Befehls in den Quelltext einzufügen?

Das xcolor-Paket [8] bietet Lösungen für diese und andere Fragestellungen. Neben Kompatibilität zu color und den zugehörigen Treiberdefinitionen wurde vor allem auf eine möglichst einfache Bedienbarkeit Wert gelegt. Dazu gehört der weitgehende Verzicht auf explizite Farbdefinitionen – die direkte Anwendung steht im Vordergrund. Wie das geht, wird weiter unten beschrieben.

Technisches

Installation

Die Installation besteht darin, die Datei `xcolor.sty` an einen Ort zu kopieren, wo sie von (pdf)L^AT_EX gefunden wird. Dann kann xcolor einfach anstelle von color benutzt werden. Der allgemeine Befehl zur Aktivierung lautet demnach `\usepackage[options]{xcolor}` in der Dokumentenpräambel. Hierbei werden mit *options* die im folgenden Abschnitt aufgeführten Paketoptionen bezeichnet.

Paketoptionen

Die Paketoptionen lassen sich in vier Klassen einteilen:

- Optionen, die an das color-Paket weitergereicht werden (beispielsweise `dvips`, `pdftex`),

Tabelle 1: Die Paketoptionen von xcolor

<i>Option</i>	<i>Beschreibung</i>
natural	(Default) Belasse alle Farben in ihrem Farbmodell, außer RGB (konvertiert in rgb), HSB (konvertiert in hsb) und Gray (konvertiert in gray).
rgb	Konvertiere alle Farben in das rgb -Modell.
cmv	Konvertiere alle Farben in das cmv -Modell.
cmv	Konvertiere alle Farben in das cmv -Modell.
hsb	Konvertiere alle Farben in das hsb -Modell.
gray	Konvertiere alle Farben in das gray -Modell. Insbesondere nützlich, um die Ausgabe eines Schwarz-Weiß-Druckers zu simulieren.
RGB	Konvertiere alle Farben in das RGB-Modell (und hinterher nach rgb).
HSB	Konvertiere alle Farben in das HSB-Modell (und hinterher nach hsb).
Gray	Konvertiere alle Farben in das Gray-Modell (und hinterher nach gray).
pst	Lade das pstcol -Paket [5], um die „normalen“ Farbdefinitionen innerhalb von pstricks -Makros verwenden zu können.
table	Lade das colortbl -Paket [4], um Zeilen, Spalten und Zellen innerhalb von Tabellen einzufärben.
override	Ersetze den ursprünglichen <code>\definecolor</code> -Befehl durch die Definition von <code>\xdefinecolor</code> .
showerrors	(Default) Zeige eine Fehlermeldung an, falls eine undefinierte Farbe verwendet wird (entspricht dem Verhalten des color -Pakets).
hideerrors	Zeige eine Warnmeldung an, falls eine undefinierte Farbe verwendet wird und ersetze diese Farbe durch <i>black</i> .

- Optionen, die das dokumentenweite Farbmodell festlegen (`natural`, `rgb`, `cm`, `myk`, `hsb`, `gray`, `RGB`, `HSB`, `Gray`),
- Optionen, die andere Pakete laden (`pst`, `table`),
- Optionen, die das Verhalten bestimmter Befehle beeinflussen (`override`, `showerrors`, `hideerrors`).

Die xcolor-spezifischen Paketoptionen sind in Tabelle 1 beschrieben.

Farbmodelle

Eine umfassende Erklärung oder gar Herleitung von Farbmodellen berührt Gebiete wie Physik, Biologie und Drucktechnik und geht weit über das Ziel dieses Artikels hinaus. Daher seien aus der Fülle der vorhandenen Literatur als Einführung die Bücher [1] und [7] empfohlen.

Wir vertreten hier einen schlichten, algorithmisch-technokratischen Standpunkt: Farbmodelle sind ein- oder mehrdimensionale numerische Repräsentationen bestimmter physikalischer Effekte (nämlich Farben), die – mit gewissen Einschränkungen – rechnerisch ineinander überführt werden können.

Die unterstützten Farbmodelle sind in Tabelle 2 aufgeführt. Es sei betont, dass diese Farbmodelle unabhängig vom ausgewählten Treiber zur Verfügung gestellt werden. Jedoch geben manche Treiber (beispielsweise `dvipdfm`) nur vor, das `hsb`-Modell zu unterstützen, sodass eine Strategie implementiert wurde, die dieses Verhalten korrigiert. Die tatsächlich von xcolor hinzugefügten Farbmodelle kann man nach einem L^AT_EX-Lauf der `log`-Datei entnehmen.

Für die „ganzzahligen“ Modelle `RGB`, `HSB` und `Gray` können die in Tabelle 2 angegebenen Konstanten L , M , N in einfacher Weise geändert werden. Dazu dienen die Befehle `\def\rangeRGB{<L>}`, `\def\rangeHSB{<M>}` und `\def\rangeGray{<N>}`, die allerdings vor dem Laden von xcolor auszuführen sind.

Definition von Farben

`\xdefinecolor{<name>}{<model>}{<color specification>}` ist der neue und zentrale Befehl zur Definition von Farben, für welche die erweiterten Möglichkeiten von xcolor genutzt werden sollen. Er wird verwendet wie bisher `\definecolor` und ersetzt dieses Kommando, das gleichwohl weiterhin mit seiner ursprünglichen Bedeutung verfügbar ist. Jedoch kann man es

Tabelle 2: Unterstützte Farbmodelle

<i>Name</i>	<i>Basisfarben/Begriffe</i>	<i>Wertebereich</i>	<i>Default</i>
rgb	<i>red, green, blue</i>	$[0, 1]^3$	
cmY	<i>cyan, magenta, yellow</i>	$[0, 1]^3$	
cmYk	<i>cyan, magenta, yellow, black</i>	$[0, 1]^4$	
hsb	<i>hue, saturation, brightness</i>	$[0, 1]^3$	
gray	<i>gray</i>	$[0, 1]$	
RGB	<i>Red, Green, Blue</i>	$\{0, 1, \dots, L\}^3$	$L = 255$
HSB	<i>Hue, Saturation, Brightness</i>	$\{0, 1, \dots, M\}^3$	$M = 240$
Gray	<i>Gray</i>	$\{0, 1, \dots, N\}$	$N = 15$

L, M, N sind natürliche Zahlen

per `\let\definecolor=\xdefinecolor` oder mit der Paketoption `override` „überbügeln“, sofern nicht das Farbmodell *named* genutzt werden soll, das von `\xdefinecolor` nicht unterstützt wird.

In `xcolor` werden die folgenden Farben mittels `\xdefinecolor` standardmäßig (re)definiert: *red* , *green* , *blue* , *cyan* , *magenta* , *yellow* , *black* , *white* , *darkgray* , *gray* , *lightgray* .

`\colorlet{<name>}[<model>]{<color expression>}` kopiert die zum Ausführungszeitpunkt aktuelle Bedeutung von `<color expression>` nach `<name>` und definiert damit eine neue Farbe bzw. überschreibt eine bereits vorhandene dieses Namens. Bei nicht leerem `<model>`-Parameter wird `<color expression>` zuvor noch in das spezifizierte Farbmodell transformiert. Die neue Farbe `<name>` kann sodann selbstverständlich wieder in anderen Farbausdrücken benutzt werden. Beispielsweise wurde am Anfang dieses Artikels durch das Kommando `\colorlet{tableheadcolor}{gray!25}` eine Farbe definiert, die in den Tabellen per `\rowcolor{tableheadcolor}` zur Färbung der ersten Tabellenzeile Verwendung findet.

Expressionen

Farbausdrücke

Aus Kompatibilitätsgründen gestattet das xcolor-Paket die Verwendung der in color bereitgestellten Methoden zur Definition benannter Farben. Um daraus resultierende Begriffsverwirrungen zu vermeiden, wird im Folgenden immer unterschieden zwischen

- *Standardfarben*, die direkt oder indirekt über den `\definecolor`-Befehl definiert werden (dabei wäre eine indirekte Definition von „bar“ etwa `\colorlet{bar}{foo}` nach `\definecolor{foo}...`), und
- *erweiterten Farben*, die direkt oder indirekt über die neuen Befehle `\xdefinecolor` oder `\definecolorseries` definiert werden.

Die *aktuelle Farbe*, repräsentiert durch den reservierten Farbnamen „“ (ohne Anführungszeichen), wird ebenfalls als *erweiterte Farbe* betrachtet. Falls die Paketoption `override` verwendet wird, sind natürlich sämtliche Farben automatisch erweiterte Farben.

Triviale Farbausdrücke

Ein trivialer Farbausdruck ist

$$\langle color\ expression \rangle = \langle name \rangle,$$

wobei $\langle name \rangle$ den Namen einer *Standardfarbe* oder einer *erweiterten Farbe* bezeichnet.

Nichttriviale Farbausdrücke

Die allgemeine Form eines nichttrivialen Farbausdrucks ist

$$\langle color\ expression \rangle = \langle prefix \rangle \langle name \rangle \langle mix\ expression \rangle \langle postfix \rangle$$

mit folgenden Bestandteilen:

- $\langle prefix \rangle$ ist entweder ein leerer String oder ein Minuszeichen „-“ (ohne Anführungszeichen); das Minuszeichen bewirkt, dass die aus dem restlichen Ausdruck resultierende Farbe in ihre Komplementärfarbe konvertiert wird;
- $\langle name \rangle$ ist der Name einer *erweiterten Farbe*;

- $\langle \text{mix expression} \rangle$ ist entweder ein *vollständiger* oder ein *unvollständiger* Mix-Ausdruck, wie unten beschrieben;
- $\langle \text{postfix} \rangle$ ist entweder ein leerer String oder der String „!!+“ (ohne Anführungszeichen); letzterer Fall bedarf noch folgender Voraussetzungen:
 - ▷ $\langle \text{name} \rangle$ ist der Name einer *Farbenfolge*,
 - ▷ $\langle \text{mix expression} \rangle$ ist ein *vollständiger* Mix-Ausdruck (siehe unten)
 und bewirkt, dass im Anschluss an die Auswertung und Anzeige des Farbausdrucks für die Farbenfolge $\langle \text{name} \rangle$ ein Induktionsschritt durchgeführt wird.

Vollständige Mix-Ausdrücke

Die allgemeine Form eines vollständigen Mix-Ausdrucks ist entweder ein leerer String oder

$$\langle \text{mix expression} \rangle = !\langle \text{num}_1 \rangle !\langle \text{name}_1 \rangle !\langle \text{num}_2 \rangle !\langle \text{name}_2 \rangle ! \dots !\langle \text{num}_n \rangle !\langle \text{name}_n \rangle$$

wobei gilt

- $n \geq 1$ ist eine ganze Zahl;
- jeder $\langle \text{num}_i \rangle$ -Parameter ist eine reelle Zahl aus dem Intervall $[0, 100]$, also $0 \leq \langle \text{num}_i \rangle \leq 100$;
- $\langle \text{name}_i \rangle$ ist jeweils der Name einer *erweiterten Farbe*.

Unvollständige Mix-Ausdrücke

Ein unvollständiger Mix-Ausdruck ist einfach eine Abkürzung:

$$\begin{aligned} \langle \text{mix expression} \rangle &= !\langle \text{num}_1 \rangle !\langle \text{name}_1 \rangle !\langle \text{num}_2 \rangle !\langle \text{name}_2 \rangle ! \dots !\langle \text{num}_n \rangle \\ &= !\langle \text{num}_1 \rangle !\langle \text{name}_1 \rangle !\langle \text{num}_2 \rangle !\langle \text{name}_2 \rangle ! \dots !\langle \text{num}_n \rangle !\text{white} \end{aligned}$$

Bedeutung von Farbausdrücken

Hier folgt nun die Erklärung, wie ein Ausdruck der Form

$$\langle \text{prefix} \rangle \langle \text{name} \rangle !\langle \text{num}_1 \rangle !\langle \text{name}_1 \rangle !\langle \text{num}_2 \rangle ! \dots !\langle \text{num}_n \rangle !\langle \text{name}_n \rangle \langle \text{postfix} \rangle$$

interpretiert und verarbeitet wird:

1. Zunächst werden die Modell- und Farbparameter von $\langle name \rangle$ ermittelt; damit wird eine temporäre Farbe $\langle temp \rangle$ definiert.
2. Dann wird eine Farbmischung, bestehend aus $\langle num_1 \rangle\%$ von Farbe $\langle temp \rangle$ und $(100 - \langle num_1 \rangle)\%$ von Farbe $\langle name_1 \rangle$ ausgerechnet; dies ergibt die neue temporäre Farbe $\langle temp \rangle$.
3. Der vorige Schritt wird wiederholt für alle verbleibenden Parameterpaare $(\langle num_2 \rangle, \langle name_2 \rangle), \dots, (\langle num_n \rangle, \langle name_n \rangle)$.
4. Falls $\langle prefix \rangle$ nicht leer ist, wird $\langle temp \rangle$ in seine Komplementärfarbe transformiert.
5. Falls $\langle postfix \rangle$ nicht leer ist, wird der entsprechende Induktionsschritt für die Farbenfolge $\langle name \rangle$ ausgeführt.
6. Nun wird die Farbe $\langle temp \rangle$ angezeigt oder dient als Input für andere Operationen, je nachdem, welches Kommando den ursprünglichen Farbausdruck aufgerufen hat.

In einem typischen Ausdruck in Schritt 2, also $\langle temp \rangle! \langle num_i \rangle! \langle name_i \rangle$, wird in den Spezialfällen mit $\langle num_i \rangle=100$ bzw. $\langle num_i \rangle=0$ die Farbe $\langle temp \rangle$ bzw. $\langle name_i \rangle$ ohne weitere Transformationen verwendet. Im echten Mischfall $0 < \langle num_i \rangle < 100$ können natürlich die beiden beteiligten Farben in unterschiedlichen Modellen definiert sein, etwa `\xdefinecolor{foo}{rgb}{...}` und `\xdefinecolor{bar}{cmyk}{...}`. Dann wird in der Regel die zweite Farbe $\langle name_i \rangle$ in das Modell der ersten Farbe $\langle temp \rangle$ transformiert; danach wird die Mischung in diesem Modell berechnet.¹ Infolgedessen werden die Ausdrücke $\langle temp \rangle! \langle num_i \rangle! \langle name_i \rangle$ und $\langle name_i \rangle! (100 - \langle num_i \rangle)! \langle temp \rangle$, obwohl theoretisch gleichwertig, in der Praxis nicht notwendigerweise zum selben optischen Ergebnis führen.

Verwendung der Erweiterungen

Die im `color`-Paket definierten Farbbefehle (`\color`, `\textcolor`, `\colorbox`, `\fcolorbox`, `\pagecolor`) können wie üblich benutzt werden. Allerdings wird durch `xcolor` die Möglichkeit geschaffen, überall dort, wo *Namen* von Farben als Argumente erwartet werden, *Farbausdrücke* zu verwenden, wie das folgende Beispiel zeigt (mit `\fboxrule=3pt`):

¹ Ausnahme: Um seltsame Ergebnisse zu vermeiden, wird diese Regel umgekehrt, sofern $\langle temp \rangle$ im `gray`-Modell definiert ist; in diesem Fall erfolgt eine Konvertierung in das zu $\langle name_i \rangle$ gehörende Farbmodell.

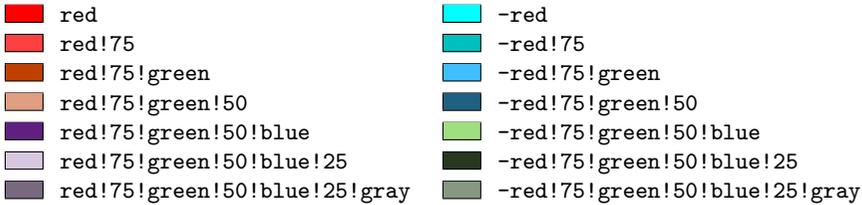


Abbildung 1: Anwendungsbeispiele für Farbausdrücke

 `\fcolorbox{red!70!green}{-red!70!green}{Test}`

Es sei hier noch einmal darauf hingewiesen, dass die erweiterten Möglichkeiten nur für *erweiterte Farben* verwendet werden können, die also direkt oder indirekt per `\xdefinecolor` definiert worden sind. Daneben können, wie bei `\color[⟨model⟩]{⟨specification⟩}`, Farbspezifikationen wie gewohnt direkt angegeben werden, siehe die Dokumentation [3].

Anwendungsbeispiele für Farbausdrücke sind in den Abbildungen 1 und 2 bis 4 zu sehen. Dabei wurden die folgenden Definitionen verwendet:

```

1 \xdefinecolor{MyGreen}{cmyk}{0.92,0,0.87,0.09}
2 \colorlet{MyGreen-hsb}[hsb]{MyGreen}
3 \colorlet{MyGreen-gray}[gray]{MyGreen}

```

Zugriff auf die aktuelle Farbe

Innerhalb eines Farbausdrucks dient „.“ als Platzhalter für die aktuelle Farbe. Beispielsweise erzeugt

```

1 {Test \color{.!75}Test \color{.!75}Test \color{.!75}Test
2 \color{.!75}Test \color{.!75}Test}.

```

die Ausgabe Test Test Test Test Test Test. Es ist auch möglich, die aktuelle Farbe für die spätere Verwendung zwischenzuspeichern, etwa mit dem Kommando `\colorlet{foo}{.}`.

In manchen Fällen ist die aktuelle Farbe von eher begrenztem Nutzen: So impliziert etwa die interne Konstruktion einer `\fcolorbox`, dass zum Zeitpunkt der Auswertung des Ausdrucks für die Hintergrundfarbe inzwischen

color-Definition: 0.92 0 0.87 0.09 k 0.92 0 0.87 0.09 K
 xcolor-Definition: {cmyk}{0.92,0,0.87,0.09}

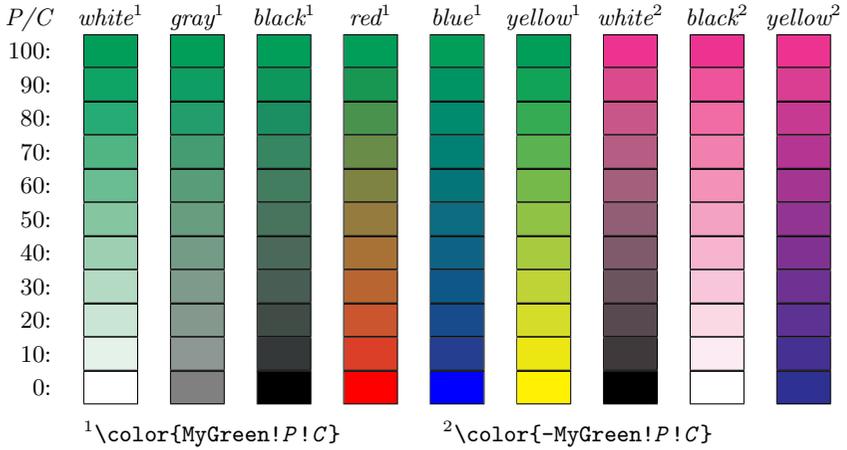


Abbildung 2: Farbtöne – „MyGreen“

color-Definition: 0 0.91 0.03995 rg 0 0.91 0.03995 RG
 xcolor-Definition: {hsb}{0.34065,1,0.91}

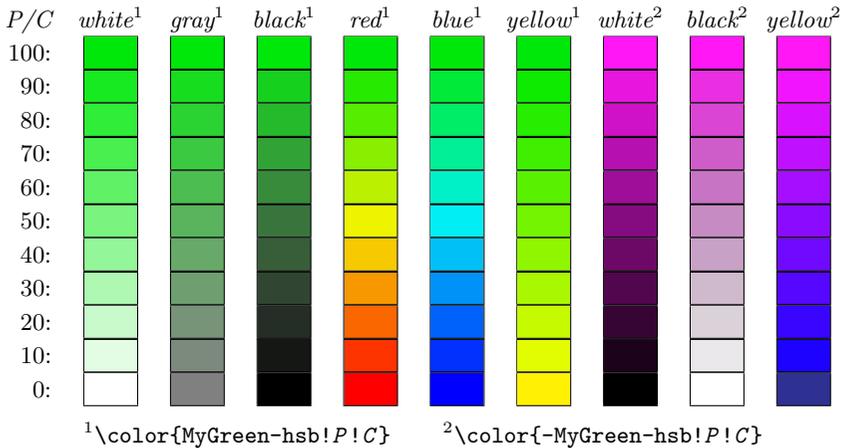


Abbildung 3: Farbtöne – „MyGreen-hsb“

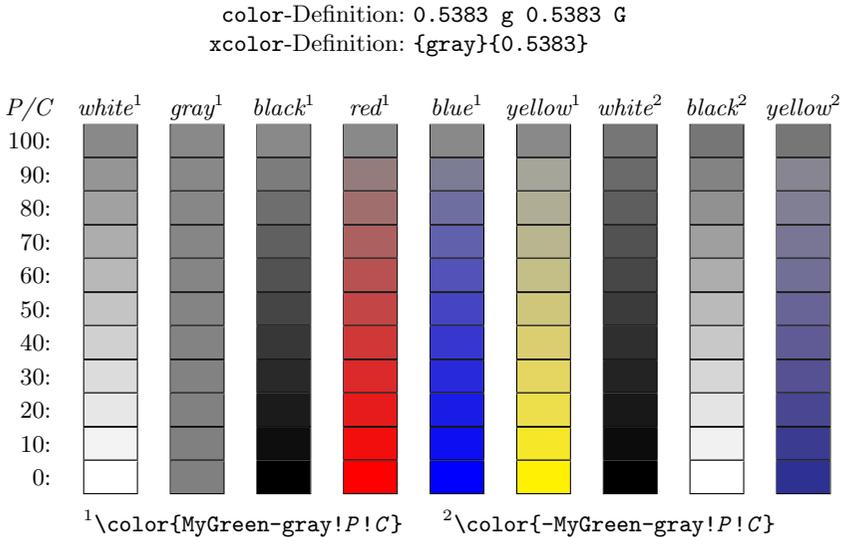


Abbildung 4: Farbtöne – „MyGreen-gray“

die Rahmenfarbe lokal zur aktuellen Farbe geworden ist – und diese muss rein gar nichts mit der aktuellen Farbe *außerhalb* der Box zu tun haben.

Informationsgesellschaft

Beschaffung von Informationen . . .

Das `color`-Paket verlässt den Pfad der Treiberunabhängigkeit unmittelbar bei der Definition einer Farbe: `\definecolor{foo}{rgb}{.8,.6,.4}` erzeugt intern ein Makro `\color@foo`, das im Falle des Treibers `dvips` den Inhalt „`rgb .8 .6 .4`“, bei `pdftex` jedoch „`.8 .6 .4 rg .8 .6 .4 RG`“, hat. Es bedarf externer Hilfe in Gestalt des Pakets `colorinfo` [9], um hieraus wieder allgemein verwendbare Informationen zu erzeugen.

Dahingegen erzeugt `\xdefinecolor{foo}{rgb}{.8,.6,.4}` intern ein Makro `\xcolor@foo` mit dem Inhalt „`\@xcolor@{rgb}{0.8,0.6,0.4}`“, es sei denn, per Paketoption wurde ein anderes Zielmodell festgelegt.

`\extractcolorspec{⟨color expression⟩}{⟨cmd⟩}` dient dazu, die genauen Parameter eines Farbausdrucks in ein Hilfsmakro zu schreiben, siehe dazu das Beispiel im nächsten Abschnitt.

`\tracingcolors=⟨num⟩` steuert, wieviel Information über die verwendeten Farben in die Protokolldatei geschrieben wird.

... und deren Verarbeitung

`\convertcolorspec`

`{⟨source model⟩}{⟨color specification⟩}{⟨target model⟩}{⟨cmd⟩}`

konvertiert eine Farbe, die durch die numerische *⟨color specification⟩* im Modell *⟨source model⟩* gegeben ist, in das Modell *⟨target model⟩* und speichert die neue Farbspezifikation im Makro *⟨cmd⟩*. Dabei können *⟨source model⟩* und *⟨target model⟩* beliebige Modelle aus Tabelle 2 sein. Diesen Befehl kann man auch verwenden, wenn man eine Konvertierung vornehmen will, ohne das Ausgangsmodell explizit zu kennen:

```

1 \extractcolorspec{red!75!green!50}\tmpa
2 \expandafter\convertcolorspec\tmpa{hsb}\tmpb
3 \expandafter\convertcolorspec\tmpa{RGB}\tmpc

```

liefert die Parameter des Ausdrucks „red!75!green!50“:

```

\tmpa: {rgb}{0.875,0.625,0.5}
\tmpb: 0.05557,0.42857,0.875
\tmpc: 223,159,128

```

Folgen von Farben

Automatische Farbgenerierung kann in grafischen Anwendungen nützlich sein, bei denen eine möglicherweise große und unspezifizierte Anzahl an Farben benötigt wird und der Anwender nicht jede einzelne Farbe festlegen will oder kann. Dafür wird hier der Begriff der *Farbenfolge* eingeführt, bestehend aus einer Basisfarbe und einem Schema, wie aus der aktiven die nächste Farbe konstruiert wird. Es liegt also ein induktives Verfahren vor.

Die praktische Anwendung besteht aus drei Teilen: Definition einer Farbenfolge (üblicherweise einmal pro Dokument), Initialisierung der Folge (potentiell mehrfach) und Anwendung – mit oder ohne Voranschreiten – der aktiven Farbe (potentiell vielfach).

Definition einer Farbenfolge

`\definecolorseries`

`{⟨name⟩}{⟨model⟩}{⟨method⟩}[⟨b-model⟩]{⟨b-spec⟩}[⟨s-model⟩]{⟨s-spec⟩}`

definiert eine Farbenfolge $\langle name \rangle$, deren Schrittberechnungen innerhalb des Farbmodells $\langle model \rangle$ (rgb, cmy, cmyk, hsb oder gray) durchgeführt werden, wobei $\langle method \rangle$ den gewünschten Algorithmus auswählt (step, grad oder last, siehe unten). Weitere Details werden durch die restlichen Argumente festgelegt:

- `[⟨b-model⟩]{⟨b-spec⟩}` spezifiziert die Basisfarbe *base* im Algorithmus entweder direkt, etwa `[rgb]{1,0.5,0.5}`, oder als Farbausdruck, etwa `{-yellow!50}`, wenn das optionale Argument fehlt.
- `[⟨s-model⟩]{⟨s-spec⟩}` spezifiziert, wie der Schrittvektor *step* im ausgewählten Algorithmus `method` berechnet wird:
 - ▷ **step, grad:** Das optionale Argument ist bedeutungslos und $\langle s-spec \rangle$ ist ein Parametervektor, dessen Dimension durch $\langle model \rangle$ determiniert wird, beispielsweise `{0.1,-0.2,0.3}` im Falle von `rgb`, `cmy` oder `hsb`.
 - ▷ **last:** Die letzte Farbe der Folge wird entweder direkt spezifiziert, also `[rgb]{1,0.5,0.5}`, oder als Farbausdruck, also `{-yellow!50}`, falls das optionale Argument fehlt.

Das allgemeine Schema lautet

$$color_1 := base, \quad color_{n+1} := U(color_n + step)$$

für $n = 1, 2, \dots$, wobei U beliebige reelle m -Vektoren in den m -Einheitswürfel abbildet:

$$U(x_1, \dots, x_m) = (u(x_1), \dots, u(x_m)), \quad u(x) = \begin{cases} 1 & \text{für } x = 1 \\ x - [x] & \text{für } x \neq 1 \end{cases}$$

Somit liefert jeder Schritt des Algorithmus eine zulässige Farbe mit Parametern aus dem Intervall $[0, 1]$.

Die unterschiedlichen Methoden benutzen unterschiedliche Varianten zur Berechnung des Schrittvektors *step*:

- **step, grad:** Das letzte Argument `{⟨s-spec⟩}` definiert den Richtungsvektor *grad*.
- **last:** `{⟨s-spec⟩}` bzw. `[⟨s-model⟩]{⟨s-spec⟩}` definiert den Parametervektor der Farbe *last*.

Durch den Befehl `\resetcolorseries` wird schließlich der tatsächliche Schrittvektor *step* ausgerechnet:

$$step := \begin{cases} grad & \text{falls } \langle method \rangle = \mathbf{step} \\ \frac{1}{\langle cycle \rangle} \cdot grad & \text{falls } \langle method \rangle = \mathbf{grad} \\ \frac{1}{\langle cycle \rangle} \cdot (last - base) & \text{falls } \langle method \rangle = \mathbf{last} \end{cases} \quad (1)$$

Bei der Definition einer Farbenfolge darf auch der Platzhalter „“, der für die aktuelle Dokumentenfarbe steht, verwendet werden. Beispielsweise wird durch `\definecolorseries{foo}{rgb}{last}{.}{-}` eine Folge definiert, die mit der aktuellen Farbe beginnt und mit deren Komplementärfarbe endet. Allerdings wird hier die konkrete Farbdefinition zur Ausführungszeit von `\definecolorseries` verwendet (entsprechend dem `\let`-Befehl in T_EX); es besteht kein Zusammenhang mit irgendeiner Farbe, die an einer anderen Stelle des Dokuments durch „“ repräsentiert wird.

Initialisierung einer Farbenfolge

`\resetcolorseries[<cycle>]{<name>}` ist ein Kommando, das mindestens einmal ausgeführt werden muss, um die Farbenfolge *<name>* benutzen zu können. Es setzt die aktive Farbe der Folge auf den Basiswert zurück und berechnet – im Falle der Methoden `grad` und `last` – den tatsächlichen Schrittvektor gemäß der Größe *<cycle>*, einer von Null verschiedenen reellen Zahl, und Gleichung (1). Falls das optionale Argument leer ist, wird der im Makro `\colorseriescycle` gespeicherte Wert herangezogen (Default: 16). Das optionale Argument wird im Falle der `step`-Methode ignoriert.

Anwendung einer Farbenfolge

Um die aktive Farbe einer Farbenfolge anzuzeigen, kann der Name der Folge wie ein gewöhnlicher Farbname in Ausdrücken verwendet und damit per `\color`, `\textcolor`, ... aufgerufen werden. Benutzt man darüber hinaus die weiter vorne beschriebene spezielle Syntax mit dem *<postfix>* „!+“, dann wird durch `\color{<name>!+}` etc. nicht nur die Farbe angezeigt, sondern auch ein Induktionsschritt ausgeführt. In diesem Fall wird also die aktive Farbe der Folge geändert. In Abbildung 5 werden verschiedene Ausprägungen von Farbenfolgen demonstriert.

S_1	S_2	G_1	G_2	L_1	L_2	L_3	L_4	L_5
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12	12
13	13	13	13	13	13		13	13
14	14	14	14	14	14		14	14
15	15	15	15	15	15		15	15
16	16	16	16	16	16		16	16

Individuelle Definitionen

```

 $S_1$  \definecolorseries{test}{rgb}{step}[rgb]{.95,.85,.55}{.17,.47,.37}
 $S_2$  \definecolorseries{test}{hsb}{step}[hsb]{.575,1,1}{.11,-.05,0}
 $G_1$  \definecolorseries{test}{rgb}{grad}[rgb]{.95,.85,.55}{3,11,17}
 $G_2$  \definecolorseries{test}{hsb}{grad}[hsb]{.575,1,1}{.987,-.234,0}
 $L_1$  \definecolorseries{test}{rgb}{last}[rgb]{.95,.85,.55}[rgb]{.05,.15,.55}
 $L_2$  \definecolorseries{test}{hsb}{last}[hsb]{.575,1,1}[hsb]{-.425,.15,1}
 $L_3$  \definecolorseries{test}{rgb}{last}{yellow!50}{blue}
 $L_4$  \definecolorseries{test}{hsb}{last}{yellow!50}{blue}
 $L_5$  \definecolorseries{test}{cmy}{last}{yellow!50}{blue}

```

Gemeinsame Definitionen

```

\resetcolorseries[12]{test}
\rowcolors[\hline]{1}{test!!+}{test!!+}
\begin{tabular}{c}
\number\rownum\ \ \number\rownum\ \ \number\rownum\ \ \number\rownum\ \
\end{tabular}

```

Abbildung 5: Beispiele für Farbenfolgen

Um dem Titel dieses Artikels gerecht zu werden, sei hier noch ein weiteres Beispiel (Abbildung 6) mit Quelltext gezeigt. Es ist im Wesentlichen der Dokumentation [6] entnommen.

```

1 \documentclass{article}
2 \usepackage[pst]{xcolor}
3 \pagestyle{empty}
4 \newcommand*\SheepHead
5   {\begin{pspicture}(3,1.5)
6     \pscustom[liftpen=2,fillstyle=solid,fillcolor=sheep!!+]{%
7       \pscurve(0.5,-0.2)(0.6,0.5)(0.2,1.3)(0,1.5)(0,1.5)
8         (0.4,1.3)(0.8,1.5)(2.2,1.9)(3,1.5)(3,1.5)(3.2,1.3)
9         (3.6,0.5)(3.4,-0.3)(3,0)(2.2,0.4)(0.5,-0.2)}
10    \pscircle*(2.65,1.25){0.12\psunit}% Eye
11    \psccurve*(3.5,0.3)(3.35,0.45)(3.5,0.6)(3.6,0.4)% Muzzle
12    % Mouth
13    \pscurve(3,0.35)(3.3,0.1)(3.6,0.05)
14    % Ear
15    \pscurve(2.3,1.3)(2.1,1.5)(2.15,1.7)
16    \pscurve(2.1,1.7)(2.35,1.6)(2.45,1.4)
17  \end{pspicture}}
18 \newcommand*\Sheep
19   {\SheepHead\SheepHead\SheepHead\SheepHead\SheepHead}
20 \definecolorseries{sheep}{rgb}{step}[rgb
21   ]{.95,.85,.55}{.17,.47,.37}
22 \resetcolorseries{sheep}
23 \begin{document}
24 \psset{unit=0.5}
25 \begin{pspicture}(-8,-1.5)(8.5,7.5)
26   \rput(0,6){\Sheep}
27   \rput(0,4.5){\Sheep}
28   \rput(0,3){\Sheep}
29   \rput(0,1.5){\Sheep}
30   \rput(0,0){\Sheep}
31 \end{pspicture}
\end{document}

```

Unterschiede zwischen Farben und Farbenfolgen

Obwohl sie sich bei der Anwendung innerhalb von Farbausdrücken gleichartig verhalten, sind die durch `\xdefinecolor` und `\definecolorseries` definierten Objekte grundsätzlich verschieden hinsichtlich ihres Gültigkeitsbereiches:



Abbildung 6: Mit Farbenfolgen eingefärbte Schafherde

`\definecolor` und `\xdefinecolor` erzeugen stets *lokale* Farben, wohingegen `\definecolorseries` *globale* Objekte generiert (sonst wäre es ziemlich unbequem, den Induktionsmechanismus innerhalb von Tabellen oder Grafiken anzuwenden). Nimmt man beispielsweise an, dass „bar“ eine undefinierte Farbe ist, so kann man nach

```

1 \begingroup
2 \definecolorseries{foo}{rgb}{last}{red}{blue}
3 \resetcolorseries[10]{foo}
4 \xdefinecolor{bar}{rgb}{.6,.5,.4}
5 \endgroup

```

Befehle wie `\color{foo}` oder `\color{foo!!+}` uneingeschränkt verwenden, während `\color{bar}` unweigerlich zu einer Fehlermeldung führt. Jedoch kann man vermöge `\colorlet{bar}{foo}` oder `\colorlet{bar}{foo!!+}` die aktive Farbe einer Farbenfolge lokal speichern – ohne oder mit Induktionsschritt.

Zeilenweise

Zur Erzeugung farbiger Zeilen, Spalten oder Zellen in Tabellen hat sich das Paket `colortbl` [4] bewährt. In `xcolor` wird darauf aufbauend die Möglichkeit zur Verfügung gestellt, Tabellenzeilen automatisch alternierend (also ungerade und gerade Zeilen jeweils einheitlich) zu färben.

```

\rowcolors[\hline]{3}{green!25}{yellow!50}
\arrayrulecolor{red!75!gray}
\begin{tabular}{ll}
test & row \number\rownum\\
test & row \number\rownum\\
test & row \number\rownum\\
test & row \number\rownum\\
\arrayrulecolor{black}
test & row \number\rownum\\
test & row \number\rownum\\
\rowcolor{blue!25}
test & row \number\rownum\\
test & row \number\rownum\\
\hiderowcolors
test & row \number\rownum\\
test & row \number\rownum\\
\showrowcolors
test & row \number\rownum\\
test & row \number\rownum\\
\multicolumn{1}%
{>{\columncolor{red!12}}1}{test} & row \number\rownum\\
\end{tabular}

```

test	row 1	test	row 1
test	row 2	test	row 2
test	row 3	test	row 3
test	row 4	test	row 4
test	row 5	test	row 5
test	row 6	test	row 6
test	row 7	test	row 7
test	row 8	test	row 8
test	row 9	test	row 9
test	row 10	test	row 10
test	row 11	test	row 11
test	row 12	test	row 12
test	row 13	test	row 13

Abbildung 7: Alternierende Zeilenfarben in Tabellen: `\rowcolors` und `\rowcolors*`

`\rowcolors[commands]{num}{odd-row color expression}{even-row color expression}`

muss ausgeführt werden, *bevor* die Tabelle beginnt. *num* gibt an, welches die erste gemäß dem durch *odd-row color expression* (ungerade Zeilennummern) und *even-row color expression* (gerade Zeilennummern) vorgegebenen Farbschema zu färbende Tabellenzeile ist. Die Farbgargumente können auch leer bleiben (= keine Farbe). Neben `\rowcolors` existiert auch eine Variante `\rowcolors*`, bei der die in *commands* angegebenen Befehle in Zeilen mit deaktiviertem Farbschema (siehe unten) ignoriert werden, während in der „normalen“ Version *commands* in jeder Tabellenzeile angewandt werden. Solche optionalen Befehle können beispielsweise `\hline` oder `\noalign{stuff}` sein.

Mit den Befehlen `\showrowcolors` und `\hiderowcolors` kann man temporär das Farbschema aktivieren und deaktivieren. Daneben kann jederzeit mittels

`\rowcolor` eine individuell abweichende Färbung einzelner Zeilen erreicht werden. Der Zähler `\rownum` gewährt innerhalb einer Tabelle Zugriff auf die aktuelle Tabellenzeile.

Ein ausführliches Beispiel ist in Abbildung 7 aufgeführt.

Zusammenfassung und Ausblick

In diesem Artikel wurde geschildert, wie das Paket `xcolor` für diverse Arten von Farbschattierungen, Farbmischungen und Farbenfolgen eingesetzt werden kann. Eine eingehende Diskussion der zugrunde liegenden Umrechnungsformeln zwischen den verschiedenen Farbmodellen wird Gegenstand eines späteren Artikels sein.

Literatur

- [1] Adobe Systems Incorporated: *PostScript Language Reference Manual*; Addison-Wesley; 3. Aufl.; 1999; <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>.
- [2] David P. Carlisle: *color – Standard \LaTeX Color*; 1999; CTAN: `macros/latex/required/graphics/color.*`.
- [3] David P. Carlisle: *Packages in the ‘graphics’ bundle*; 1999; CTAN: `macros/latex/required/graphics/grfguide.tex`.
- [4] David P. Carlisle: *colortbl – Color table columns*; 2001; CTAN: `macros/latex/contrib/carlisle/colortbl.*`.
- [5] David P. Carlisle: *pstcol – PSTricks color compatibility*; 2001; CTAN: `macros/latex/required/graphics/pstcol.*`.
- [6] Denis Girou: *pst-fill – A PSTricks package for filling and tiling areas*; 2000; CTAN: `graphics/pstricks/doc/pst-fill.doc`.
- [7] Michel Goossens, Sebastian Rahtz und Frank Mittelbach: *The \LaTeX Graphics Companion*; Addison-Wesley; 1997.
- [8] Uwe Kern: *xcolor – \LaTeX color extensions*; 2003; CTAN: `macros/latex/contrib/xcolor/`.
- [9] Rolf Niepraschk: *colorinfo – Info from defined colors*; 2003; CTAN: `macros/latex/contrib/colorinfo/`.

Strahlende Präsentationen mit \LaTeX

Till Tantau

Die \LaTeX -Klasse `beamer` dient dem Erstellen von Präsentationen mittels \LaTeX , `pdf \LaTeX` oder `L \TeX` . Wie der Name der Klasse schon verrät, will `beamer` es insbesondere leicht machen, Beamer-Vorträge zu erzeugen; klassische Folienvorträge werden aber genauso unterstützt. Soweit möglich baut `beamer` auf den dem Benutzer vertrauten \LaTeX -Befehlen auf, wie `\section` zur Erzeugung von Abschnitten oder `enumerate` zur Erzeugung von Aufzählungen. Beamer-Vorträge profitieren oft von der Verwendung von Overlays, und deren Erstellung wird durch eine einfache, aber mächtige Syntaxerweiterung erleichtert. Die Klasse folgt der \LaTeX -Philosophie der Trennung von Form und Inhalt: Aus einer einzigen Quelle lassen sich durch Variation von Layout-Parametern verschieden gestaltete Vorträge erzeugen. Vorgefertigte Layouts reichen von klassisch-sachlich bis modisch-effektiv, folgen aber immer dem Bauhausgrundsatz „form follows function“.

Einleitung

Wenn es darum geht, wissenschaftliche Aufsätze zu erstellen, geht oft kein Weg an \LaTeX vorbei. Für geübte Nutzer ist es ein mächtiges Werkzeug, mit dem sie auch umfangreiches, kompliziertes Material schnell und professionell setzen können. Wissenschaftliche Zeitschriften bitten Autoren oft, Manuskripte im \LaTeX -Format einzureichen, da dessen klare Trennung von Inhalt und Form es den Setzern leicht macht, Manuskripte mit minimalem Aufwand den typografischen Normen der Zeitschrift anzupassen.

Wissenschaftliche Resultate wollen aber nicht nur in Papierform verbreitet werden. Sie wollen auch vorgetragen und diskutiert werden, wozu am Anfang meist ein ruhiger Raum und vielleicht eine Tafel reichen. Vorträge vor großem Publikum verlangen aber – zumindest im mathematisch-naturwissenschaftlichen Bereich –, dass Folien oder eine Beamer-Präsentation vorbereitet werden.

Wer \LaTeX zum Erstellen seiner Manuskripte benutzt, dem wird es nahe liegend erscheinen, es auch zur Erstellung von Vorträgen einzusetzen. Schließlich liegt der Text oft schon im \LaTeX -Format vor, das Programm ist vertraut. Das Layout von Folien und deren Typografie folgen aber anderen Gesetzmäßigkeiten als das Layout von Texten, die gedruckt werden sollen. Böse Zungen

behaupten, dass „nur ein deutscher Professor seine Zuhörer mit A4-Folien in Zehn-Punkt-Schrift quält“. Da deshalb Standardklassen wie `article` oder `report` ausscheiden, sind von T_EX-Nutzern Alternativen entwickelt worden. Die Klasse `beamer`, die im Folgenden auszugsweise vorgestellt wird, ist eine davon.

Eine Stärke von L^AT_EX ist der standardisierte Satz von Grundbefehlen zur Strukturierung von Texten. Wer schon einmal Artikel bei unterschiedlichen Zeitschriften eingereicht hat, wird es zu schätzen wissen, dass immer die gleichen `\section-`, `\subsection-` und `itemize`-Befehle benutzt werden können. Die `beamer`-Klasse trägt dem Rechnung, indem möglichst alle bekannten L^AT_EX-Befehle zur Verfügung stehen und den „erwarteten Effekt“ – oder zumindest einen „sinnvollen Effekt“ – haben. So erzeugt beispielsweise `\tableofcontents` wie erwartet ein Inhaltsverzeichnis und `\maketitle` fügt einen Titelblock ein.

Eine Stärke von Beamer-Präsentationen ist die Möglichkeit, Overlays zu nutzen. Overlaying ist eine althergebrachte Technik bei Folienvorträgen, bei der man auf eine Basisfolie nach und nach immer mehr Folien legt, um schrittweise Änderungen zu veranschaulichen. Die `beamer`-Klasse bietet verschiedene Möglichkeiten, Overlays zu erzeugen. Hauptidee ist eine einfache, aber mächtige Syntax-Erweiterung: Folgt bestimmten Befehlen eine Zahlenfolge in spitzen Klammern, so „wirkt“ der Befehl nur auf die Overlays mit den angegebenen Nummern. Schreibt man beispielsweise innerhalb einer `itemize`-Umgebung `\item<2>` statt `\item`, so wird der gesamte Punkt lediglich auf der zweiten Seite der aktuellen Overlay-Folge angezeigt.

Eine L^AT_EX-Grundphilosophie ist die Trennung von Form und Inhalt. Diesem Prinzip folgt auch `beamer`: Im Quelltext steht, *was* alles auf einer Seite zu sehen sein soll, nicht aber, *wie* es nachher genau aussehen wird. Dies ist die Aufgabe eines so genannten „Themas“ (englisch *theme*). Zusätzlich zu vorgefertigten Themen lassen sich auch neue Themen nach einem Baukastenprinzip zusammensetzen.

Im Folgenden werden die grundlegenden Fähigkeiten der `beamer`-Klasse zunächst anhand eines einfachen Beispiels illustriert. Danach wird die Syntax-Erweiterung zur Erzeugung von Overlays vorgestellt. Dem folgen einige Hinweise, wie man das Aussehen von `beamer`-Vorträgen den eigenen Bedürfnissen anpassen kann. Den Abschluss bilden zwei Abschnitte über das Umfeld und die Entwicklung der Klasse. In der Zusammenfassung finden sich Hinweise, wo die Klasse zu beziehen ist.

```

1 \documentclass{beamer}
2
3 \usepackage[german]{babel}
4 \usepackage[latin1]{inputenc}
5
6 \title{Es gibt keine größte Primzahl}
7 \author{Euklid von Alexandria}
8 %\date{1.\ Mai $-280$}
9
10 \begin{document}
11   \frame{\maketitle}
12
13   \section{Hauptresultat}
14   \frame{
15     \frametitle{Ich zeige, dass es keine größte Primzahl gibt.}
16     \begin{Satz}
17       Es gibt keine größte Primzahl.
18     \end{Satz}
19     \begin{Beweis}
20       \begin{enumerate}
21         \item Nehmen wir an,  $(p)$  wäre die größte Primzahl.
22         \item Sei  $(q := 1 + \prod_{i=1}^p i = 1+p!)$ .
23         \item Dann ist  $(q)$  teilerfremd zu allen  $(p' \in \{1, \dots, p\})$ .
24         \item Also ist  $(q > p)$  prim.
25       \end{enumerate}
26     \end{Beweis}
27   }
28 \end{document}

```

Abbildung 1: Beispiel-Quelltext einer beamer-Präsentation.

Herr Euklid hält einen Beamer-Vortrag

Herr Euklid möchte in einem kurzen Vortrag seinen verehrten Kollegen von seiner kürzlich gemachten Entdeckung berichten, dass es keine größte Primzahl gibt. Der Quelltext seines Vortrags ist in Abbildung 1 dargestellt. Übersetzt er diesen beispielsweise mit pdfL^AT_EX, so erhält Herr Euklid einen Vortrag im PDF-Format, der in Abbildung 2 dargestellt ist.

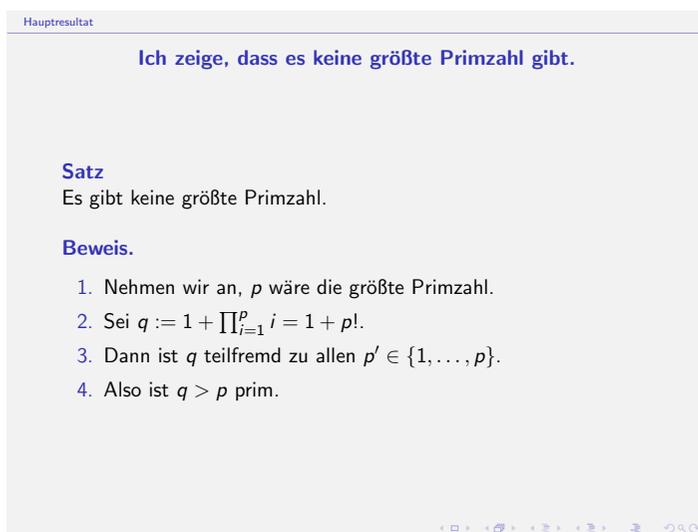
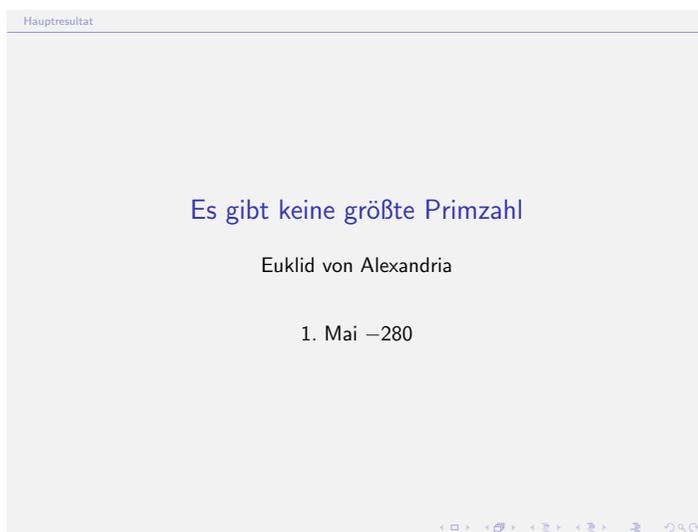


Abbildung 2: Übersetzt man den Quelltext aus Abbildung 1 beispielsweise mittels pdfL^AT_EX, so enthält die resultierende PDF-Datei die beiden oben abgebildeten Seiten.

Der Quelltext im Einzelnen

In der Präambel des Quelltextes aus Abbildung 1 werden der Titel und der Vortragende festgelegt. Neben den Standardbefehlen `\title`, `\author` und `\date` stellt `beamer` auch noch den Befehl `\institute` zur Verfügung, der dem Leser vielleicht aus den L^AT_EX-Klassen des Springer-Verlages bekannt ist.

Die eigentliche Präsentation besteht aus zwei so genannten *Rahmen*, die jeweils durch den Befehl `\frame` erzeugt werden. Ein Rahmen enthält den Text einer „Folie“ oder „Seite“, wobei beide Bezeichnungen bei Beamer-Vorträgen eigentlich anachronistisch sind. Falls Overlays eingesetzt werden, enthält ein Rahmen mehrere Seiten, dazu später mehr. Der erste Rahmen wird mit dem Befehl `\maketitle` mit einer Titelseite gefüllt. Ihm folgt ein `\section`-Befehl, der einen neuen Abschnitt eröffnet. Im Gegensatz zum normalen L^AT_EX erzeugt dieser Befehl aber lediglich einen Eintrag im Inhaltsverzeichnis und in den Navigationsleisten. Er fügt keinen Text direkt in einen Rahmen ein, da dies in Vorträgen unpassend wäre; mit dem Befehl `\AtBeginSection` lässt sich dies aber ändern.

Der zweite Rahmen beginnt mit dem beamer-eigenen Befehl `\frametitle`, dessen Bedeutung wohl selbsterklärend ist. Innerhalb des Rahmens werden die zwei vordefinierten theoremartigen Umgebungen `Satz` und `Beweis` benutzt. Neben diesen Umgebungen sind noch einige weitere typische Umgebungen wie `Korollar` oder `Definition` vordefiniert. Natürlich sind auch die englischen Pendant definiert, aber hier merkt man der Klasse die Herkunft und die Nöte ihres Autors an.

Änderung des Erscheinungsbilds durch Variation des Themas

Das Erscheinungsbild des Vortrags lässt sich ändern, indem man das so genannte *Thema* wechselt. Ein Thema legt fest, wie die einzelnen Elemente einer Präsentation dargestellt werden sollen. Um ein anderes Thema festzulegen, muss lediglich eine der mitgelieferten Stildateien aus dem Verzeichnis `beamer/themes` mit dem üblichen `\usepackage`-Befehl geladen werden. Wie man den Abbildungen 2 bis 5 entnehmen kann, unterscheidet sich das Aussehen der Themen recht stark.

Die vordefinierten Themen legen einige Aspekte des Erscheinungsbildes nicht fest. So wird zwar der Schriftschnitt, nicht aber die Schriftart festgelegt. Die Schriftart lässt sich wie üblich wechseln, indem ein geeignetes Paket wie `times`

oder `utopia` geladen wird. Ebenfalls legen die Themen die „Strukturfarbe“ (in den Beispielen blau) nicht fest. Die Strukturfarbe lässt sich durch Klassenoptionen wie `red` oder auch durch Änderung der Farbe `beamerstructure` anpassen.

Die Artikelversion

Herr Euklid möchte seinen Vortrag durch einen kleinen Artikel in Papierform begleiten (später soll dieser Teil eines größeren Werkes werden, für das er bereits den Arbeitstitel „Elemente“ vorgesehen hat). Die `beamer`-Klasse bietet Instrumente, einen Artikel und einen oder sogar mehrere Vorträge in einer Datei zu halten und Teile des L^AT_EX-Quelltextes beiden zugänglich zu machen. Im einfachsten Fall ersetzt man die erste Zeile des Dokuments durch folgende zwei Zeilen, wodurch man die in Abbildung 6 dargestellte Seite erhält:

```
1 \documentclass[class=article,a5paper]{beamer}
2 \usepackage{beamerbasearticle}
```

Es ist möglich, bestimmte Rahmen oder auch nur Teile davon in der Artikelversion zu unterdrücken, Texte außerhalb von Rahmen automatisch nur in die Artikelversion aufzunehmen oder einzelne `\section-` oder `\subsection-` Befehle nur in der Artikelversion oder nur in der Vortragsversion wirken zu lassen.

Dynamische Effekte

In einem Vortrag entsteht schnell eine Diskrepanz zwischen der inhärenten Dynamik der vorgestellten Konzepte und der Statik der gezeigten Texte, Formeln und Grafiken. Eine Folie mit dem Pseudo-Code des Merge-Sort-Algorithmus ist einer Animation, die den Algorithmus „in Aktion“ zeigt, klar unterlegen. In anderen Fällen ist die innere Dynamik des Vortrags nicht offensichtlich. Beispielsweise enthält der Beweis aus dem Beispiel in Abbildung 1 einen Spannungsbogen, der Mathematikern oft gar nicht auffällt, da sie ihn nicht mehr wahrnehmen: Der Beweis beginnt mit einer Annahme, die am Ende zum Widerspruch geführt wird. Dazwischen finden sich Argumente, die den Widerspruch herleiten. Herr Euklid könnte sich überlegen, den ersten und den letzten Punkt seines Beweises *zuerst* zu zeigen und erst danach die Zwischenschritte zu erklären.

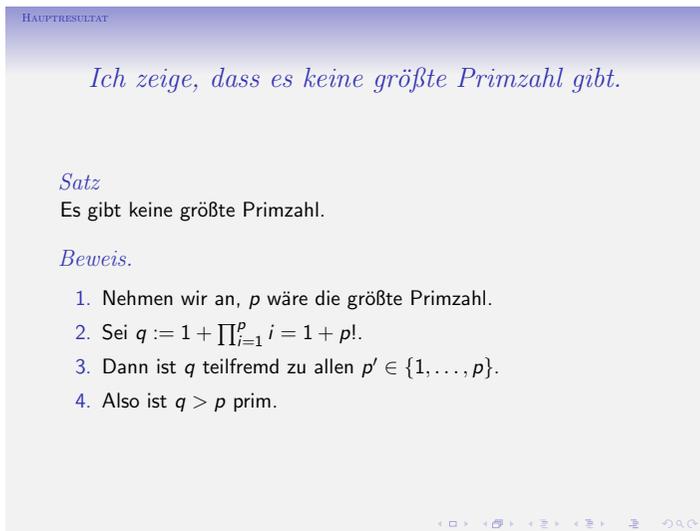


Abbildung 3: Dieser Vortrag ist aus dem selben Quelltext erzeugt wie der Vortrag aus Abbildung 2. Der einzige Unterschied besteht darin, dass nun das Thema `classic` genutzt wurde, indem das Paket `beamerthemeclassic` eingebunden wurde.

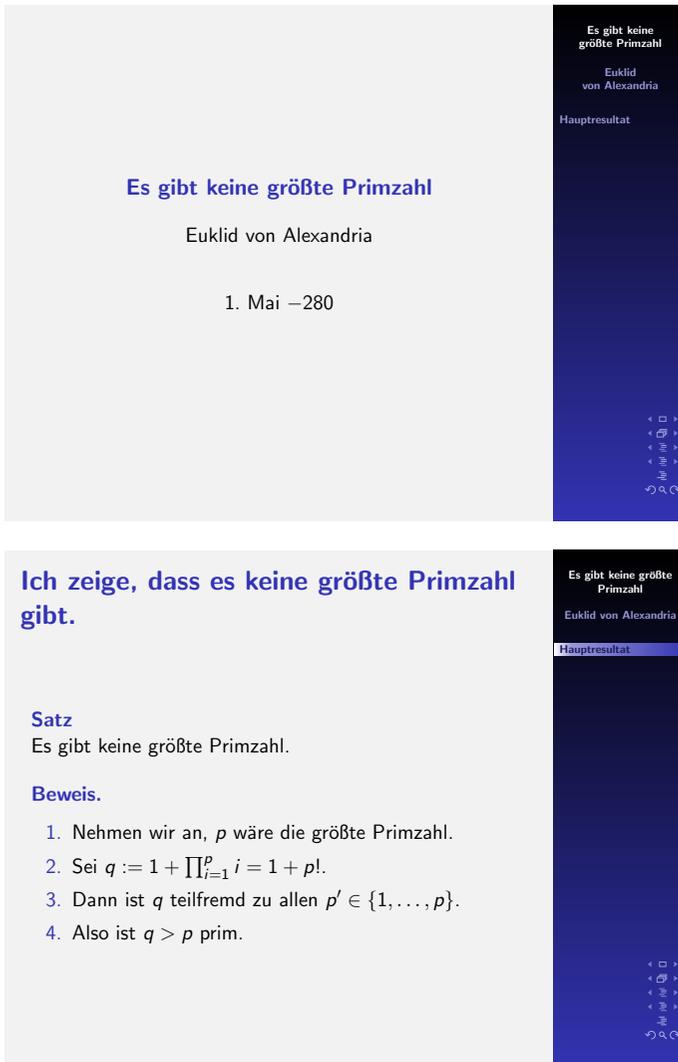
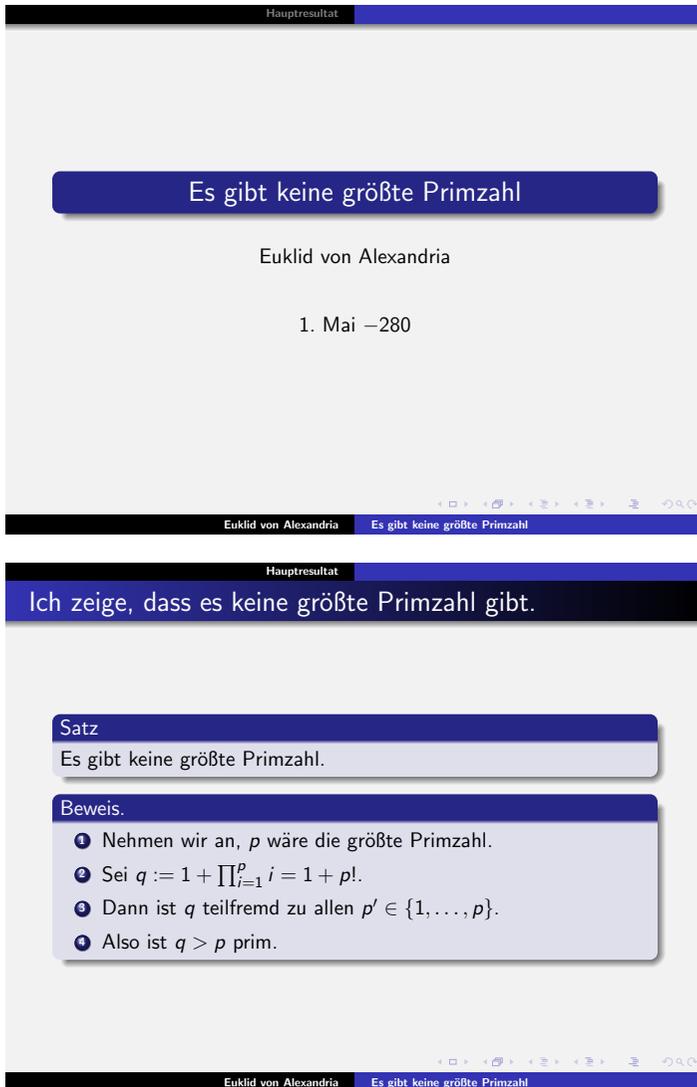


Abbildung 4: Der Vortrag mit dem Thema aus dem Paket `beamerthemesidebar` mit den Optionen `dark` und `tab`. Die Navigationsleiste dient nicht nur der Orientierung der Zuhörerschaft, sondern ist auch „anklickbar“, ein geeignetes Betrachtungsprogramm vorausgesetzt.

Abbildung 5: Der Vortrag mit dem Thema `beamerthemeshadow`.

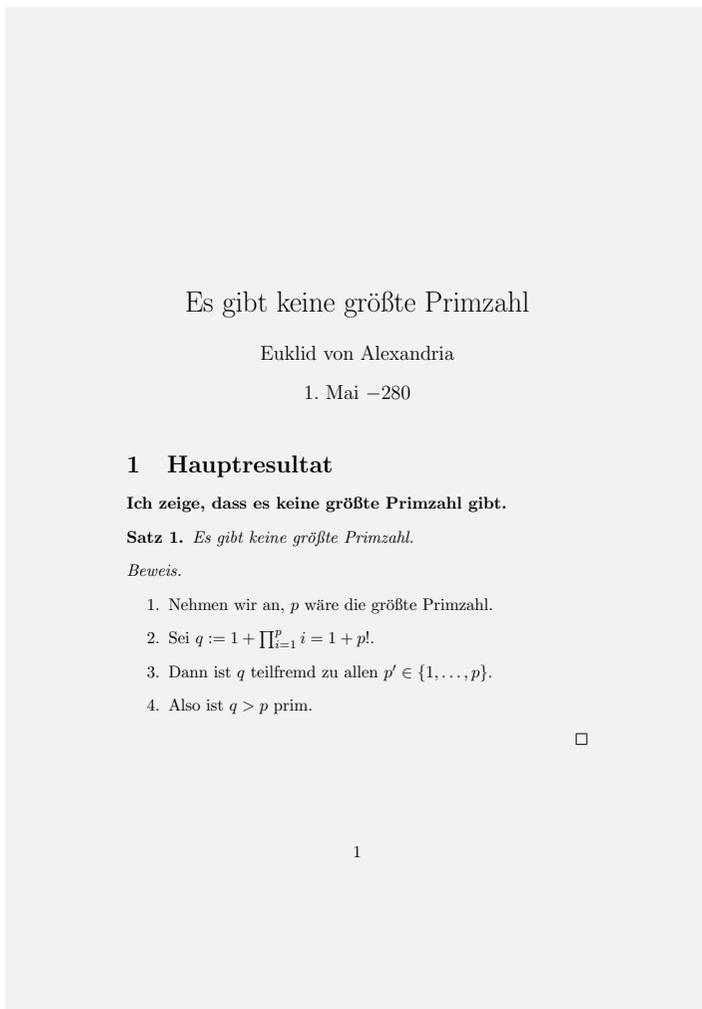


Abbildung 6: Die „Artikelversion“ des Beispiels aus Abbildung 1.

Das Konzept der Overlay-Spezifikationen

Die `beamer`-Klasse unterstützt die Visualisierung von Dynamik durch ein einfaches Konzept: Bestimmten Befehlen kann in spitzen Klammern eine *Overlay-Spezifikation* nachgestellt werden. Eine solche Spezifikation ist im einfachsten Fall eine durch Kommata getrennte Liste von Nummern oder Nummernbereichen, wobei bei Nummernbereichen auch der Anfang oder das Ende offen gelassen werden können. So verweist die Spezifikation `<-3,9,5-7,11->` auf die Overlays mit den Nummern 1, 2, 3, 5, 6, 7, 9, 11 und alle folgenden; die erste Seite eines Rahmens ist Overlay Nummer 1. Ist einem Befehl eine solche Spezifikation nachgestellt, so „wirkt“ der Befehl nur auf diese Overlays. Was das genau bedeutet, hängt vom Befehl ab. Betrachten wir dazu folgende Änderung in Herrn Euklids Vortrag:

```

1 \item Nehmen wir an,  $\!(p\)$  wäre die größte Primzahl.
2 \item<2-> Sei  $\!(q := 1 + \prod_{i=1}^p i = 1+p!\)$ .
3 \item<3-> Dann ist  $\!(q\)$  teulfremd zu allen  $\!(p' \text{ in } \{1, \dots, p\})\)$ .
4 \item Also ist  $\!(q>p\)$  prim.

```

Die zwei Spezifikationen haben folgende Effekte: Der zweite Punkt des Beweises wird lediglich auf dem zweiten Overlay des Rahmens und auf allen folgenden (also ab dem zweiten Overlay) angezeigt; der dritte Punkt wird ab dem dritten Overlay angezeigt. Da die anderen beiden Punkte keine Overlay-Spezifikation aufweisen, werden sie ganz normal auf allen Seiten des Rahmens angezeigt. Somit würden also auf dem ersten Overlay nur der erste und der letzte Punkt, auf dem zweiten Overlay zusätzlich der zweite Punkt und auf dem dritten und letzten Overlay alle Punkte angezeigt.

Woher weiß die Klasse, wie viele Overlays zu einem Rahmen gehören? Diese Zahl wird automatisch nach folgender Regel berechnet: Die größte Zahl, die in irgendeiner Overlay-Spezifikation innerhalb des Rahmens vorkommt, bestimmt die Anzahl der Overlays des Rahmens. Man kann aber auch dem `\frame`-Befehl selbst eine Overlay-Spezifikation nachstellen, wodurch genau die in dieser Spezifikation angegebene Anzahl an Overlays erzeugt werden.

Nehmen wir als Nächstes an, dass Herr Euklid den „neuen“ Anteil jedes Overlays farblich hervorheben möchte. Auf dem ersten Overlay sollen der erste und der letzte Punkt rot erscheinen, auf dem zweiten hingegen der (neu hinzugekommene) zweite Punkt und auf dem dritten der dritte Punkt. Da der Befehl `\color` eine Overlay-Spezifikation akzeptiert und keine Wirkung

zeigt, wenn die Spezifikation auf das aktuelle Overlay nicht zutrifft, ließe sich dies wie folgt bewerkstelligen:

```

1 \item {\color<1>{red} Nehmen wir an, ...}
2 \item<2-> {\color<2>{red} Sei ...}
3 \item<3-> {\color<3>{red} Dann ist \langle q \rangle ...}
4 \item {\color<1>{red} Also ist \langle q \rangle prim.}

```

Im obigen Beispiel wäre es allerdings günstiger, statt des `\color`-Befehls den beamer-eigenen Befehl `\alert` zu benutzen, der sein Argument rot darstellt, sich aber „hinter der Bühne“ auch noch um einige andere Dinge kümmert.

Overlay-Spezifikationen lassen sich nicht nur an bestimmte Befehle, sondern auch an bestimmte Umgebungen anfügen. So würde `\begin{Beweis}<2->` dafür sorgen, dass der ganze Beweis erst ab dem zweiten Overlay angezeigt würde.

Will man beliebigen Text verschwinden lassen, so stehen die beiden Befehle `\only` und `\uncover` zur Verfügung. Normalerweise fügen diese ihr Argument unverändert in den Text ein. Schreibt eine Overlay-Spezifikation aber vor, dass `\only` auf einem bestimmten Overlay nicht ausgeführt werden soll, so wird das Argument komplett unterdrückt. Der Befehl `\uncover` verhält sich ähnlich, lässt aber noch einen passend großen Leerraum für den unterdrückten Text. So ließe sich `\item<2-> Sei ...` auch etwas umständlicher schreiben als `\uncover<2->{\item Sei ...}`.

Aktions-Spezifikationen

Für viele Befehle, wie zum Beispiel `\only`, ist nur ein Verhalten sinnvoll für Overlays, auf denen der Befehl nicht „wirken“ soll. Für manche Befehle, wie den `\item`-Befehl, ist die Festlegung lediglich dem Umstand geschuldet, dass das Verdecken das am häufigsten geforderte Verhalten ist. Man könnte sich aber auch wünschen, dass der Text auf den spezifizierten Seiten rot und auf allen anderen schwarz erscheint oder dass in einer `itemize`-Umgebung das Symbol vor dem Text auf bestimmten Overlays ein Haken und auf anderen ein Kreuz ist.

Aktions-Spezifikationen erweitern Overlay-Spezifikationen, indem sie es erlauben, auf bestimmten Overlays bestimmte „Aktionen“ auszuführen. Dazu wird die gewünschte Aktion vor die Nummern der Overlay-Spezifikation,

durch ein @-Zeichen getrennt geschrieben. Mehrere Aktionen lassen sich angeben, wenn sie durch einen senkrechten Strich und ein Leerzeichen getrennt werden. Als Beispiel soll wieder der Vortrag von Herrn Euklid dienen:

```

1 \item<alert@1> Nehmen wir an, ...
2 \item<2-| alert@2> Sei ...
3 \item<3-| alert@3> Dann ist  $(q \setminus)$  ...
4 \item<alert@1> Also ist  $(q \setminus p \setminus)$  prim.

```

In Abbildung 7 ist die zweite Seite dieses Rahmens dargestellt. Zusätzlich zu den vordefinierten lassen sich eigene Aktionen definieren.

Nichtlineare Vorträge

Vorträge leben, wie der Name schon sagt, vom „Vortragen“. Gleichgültig wie perfekt die Bilder, Texte oder Animationen eines Vortrags vorbereitet sind, letztendlich ist es immer das gesprochene Wort, das darüber entscheidet, ob die Zuhörer gebannt lauschen oder sanft entschlummern.

Anders als beispielsweise in einem Theaterstück oder in einer Diskussion nehmen wir Sprache in einem Vortrag als einen langen, linearen Strom wahr. Da Bilder und Texte das gesprochene Wort lediglich unterstützen, müssen Vorträge im Kern ebenfalls linear aufgebaut sein: Während gesprochen wird, ändern sich die Bilder und Texte in einer vorbestimmten Reihenfolge.

In bestimmten Situationen wird man aber nichtlineare Komponenten in einen Vortrag einbauen wollen. Das wichtigste Beispiel ist das antizipierte Überspringen oder Vertiefen eines Themas. Häufig weiß man während der Vorbereitung eines Vortrags nicht, ob ein bestimmtes Thema eher ausführlich oder nur kurz behandelt werden sollte. Dies lässt sich oft erst aufgrund der Reaktion der Zuhörerschaft einschätzen.

Die `beamer`-Klasse bedient sich der Befehle des `hyperref`-Pakets, um antizipierte Sprünge zu erzeugen. Klickt man während des Vortrags auf Text, der als Hypersprung deklariert wurde, so „springt“ das Betrachtungsprogramm an die Stelle, auf die der Hypersprung verweist.

Viele Hypersprünge werden automatisch erzeugt. So ist jeder Punkt des Inhaltsverzeichnisses „anklickbar“, ebenso wie die Einträge in den Navigationsleisten. Will man aber manuelle Sprünge erzeugen, so ist dies mit `hyperref` etwas umständlich, und den erzeugten Sprüngen fehlt eine visuelle Rückmeldung, dass es sich um Sprungstellen handelt. Aus diesem Grund erweitert

Ich zeige, dass es keine größte Primzahl gibt.

Satz
Es gibt keine größte Primzahl.

Beweis.

- ① Nehmen wir an, p wäre die größte Primzahl.
- ② Sei $q := 1 + \prod_{i=1}^p i = 1 + p!$.
- ③ Dann ist q teilerfremd zu allen $p' \in \{1, \dots, p\}$.
- ④ Also ist $q > p$ prim.

Abbildung 7: Zweite Seite des zweiten Rahmens von Herrn Euklids Vortrag mit folgenden Optionen: Als Thema ist `beamerthemesidebar` gewählt mit den Paketoptionen `dark` und `tab`; als Strukturfarbe ein Grünton; als Grundschrift Bookman; verdeckter Text wird fast, aber nicht vollständig transparent dargestellt; Blöcke erhalten einen farbigen Hintergrund; Nummerierungen sind auf kleinen Kugeln projiziert; der zweite Punkt wird mittels `alert@2` hervorgehoben und der dritte Punkt verdeckt.

`beamer` das `hyperref`-Paket um eine Reihe nützlicher Befehle, die Abhilfe schaffen.

Zunächst gibt es mehrere Befehle, die einen Text als offensichtlich anklickbar darstellen. Der grundlegende Befehl lautet schlicht `\beamerbutton`. So liefert

```
1 \beamerbutton{Zum Beweis}
```

einen der folgenden „Knöpfe“, wobei das genaue Aussehen vom Thema abhängt:

Zum Beweis

Zum Beweis

Die visuelle Rückkopplung lässt sich noch verstärken, indem kleine Symbole an den Anfang der Knöpfe gestellt werden, deren Aussehen vom Zweck des Knopfes abhängen. Dazu stellt die Klasse drei weitere Befehle zur Verfügung:

```

1 \beamergetobutton{Zum Beweis}
2 \beamerreturnbutton{Zurück zur Behauptung}
3 \beamerskipbutton{Beweis überspringen}

```

liefern

▶ Zum Beweis

◀ Zurück zur Behauptung

▶ Beweis überspringen

Woher weiß der Knopf aber nun, wohin man springen möchte, wenn man ihn anklickt? Um einen Knopf „anklickbar“ zu machen, wird er mit dem aus `hyperref` bekannten Befehl `\hyperlink` als Hypersprung deklariert. Neben diesem Befehl stehen auch neue Befehle zur Verfügung, die lediglich in einer `beamer`-Präsentation Sinn machen. So erzeugt der folgende Befehl einen Hypersprung an den Anfang des nächsten Rahmens:

```

1 \hyperlinkframestartnext{\beamerskipbutton{Beweis überspringen}}

```

Ein Beispielszenario, in dem man diesen Befehl verwenden könnte, sieht wie folgt aus: Auf dem ersten Overlay eines Rahmens werden nur eine Behauptung und dieser Knopf angezeigt. Auf den folgenden Overlays wird dann ein Beweis angezeigt. Stellt man während des Vortrags fest, dass der Beweis übersprungen werden kann, drückt man den Knopf. Die Zuhörer sehen dies und wissen, dass ihnen der Beweis nicht gezeigt wurde. Dies kann ein Vorteil gegenüber der Schnell-und-offt-die-Seite-runter-Taste-drücken-Technik sein. Bei dieser Technik entsteht beim Zuhörer oft das Gefühl, viel mehr „verpasst“ zu haben, als tatsächlich übersprungen wurde.

Eine wichtige Frage bei nichtlinearen Vorträgen lautet: Wohin mit optionalem Material? Häufig ist das Überspringen die Regel und nicht die Ausnahme. Vertiefende Erklärungen sollen nur bei Bedarf, beispielsweise auf Nachfrage,

gezeigt werden – im Normalfall soll der Vortrag die Erklärungen überspringen. In einem solchen Fall möchte man das vertiefende Material in einen Anhang schieben und es im Normalfall nicht anzeigen. Lediglich durch Hyper-Sprünge sollte dieser Anhang erreichbar sein.

Die **beamer**-Klasse unterstützt dieses Konzept durch zwei Befehle. Zunächst lässt sich durch den Befehl `\appendix` ein Anhang beginnen, der vom eigentlichen Vortrag getrennt gehalten wird und im Inhaltsverzeichnis nicht auftaucht. Der zweite nützliche Befehl lautet `\againframe`. Dieser Befehl wiederholt einen Rahmen, der zuvor schon einmal gezeigt wurde, was auf den ersten Blick nicht sonderlich nützlich erscheint. Der Trick ist, dass sich beim ersten Anzeigen und bei der Wiederholung unterschiedliche Overlay-Spezifikationen für den Rahmen angeben lassen. Dadurch lässt sich beim ersten Mal beispielsweise nur das erste Overlay mit der Behauptung und dem Sprungknopf, bei der Wiederholung dann das zweite Overlay anzeigen. Das zweite Overlay ersetzt den Sprungknopf durch einen Rücksprungknopf und enthält zusätzliches Material. Ein Beispiel hierzu findet sich in den Abbildungen 8 und 9.

Das Aussehen von Vorträgen

Über Geschmack lässt sich bekanntlich vortrefflich streiten. Das Design der vordefinierten Themen der **beamer**-Klasse erhebt keinen Anspruch auf Perfektion. Manchen Benutzern werden die Themen langweilig erscheinen, anderen effekthascherisch.

Glücklicherweise lassen sich Themen aber leicht ändern und Teile neu zusammenstellen, wie in Abbildung 7 oder 9 zu sehen ist. Diese Flexibilität ist wichtig, da bestimmte Themen für einen Vortrag geeigneter sein können als andere. Will man einen fünfminütigen Kurzvortrag halten, so ist ein Thema mit aufwendigen Navigationsleisten, die Abschnitte und Unterabschnitte anzeigen, völlig ungeeignet. Umgekehrt sollte während eines langen Vortrags, wie zum Beispiel während einer Vorlesung, allen Zuhörern auch dann noch klar sein, „wo wir uns befinden“, wenn sie kurzzeitig nicht zuhören. Kein Zuhörer kann eineinhalb Stunden voll aufmerksam sein, wie ein Sprichwort weiß: „Man kann über alles reden, nur nicht über 20 Minuten.“

Als Nutzer der **beamer**-Klasse sollte man sich kein Lieblingsthema zulegen und dieses immer und überall verwenden, sondern sich bei jedem Vortrag

```

1  ...
2  \frame<1>[label=pspacecomplete]
3  {
4    \frametitle{Overhead-Free Languages can be PSPACE-Complete}
5
6    \begin{Theorem}
7      \(\Class{DOF}\) contains languages that are complete for
8      \(\Class{PSPACE}\).
9    \end{Theorem}
10
11   \only<1>{
12     \hyperlink{pspacecomplete<2>}{\beamergotobutton{Proof details
13       }}}
14
15   \only<2>{% Dies wird nur im Anhang angezeigt
16     \begin{proof}
17       \begin{enumerate}
18         \item
19           Let \(\A \in \Class{DLINSPACE}\) be \(\Class{PSPACE}\)-
20             complete.\\
21             Such languages are known to exist.
22             ...
23             \item
24               Then \(\h(L)\) is in \(\Class{DOF}\) and it is
25               \(\Class{PSPACE}\)-complete.
26             \end{enumerate}
27           \end{proof}
28           \hfill\hyperlink{pspacecomplete<1>}{\beamerreturnbutton{Return
29             }}
30         }
31       }
32     }
33   }
34   ...
35   \appendix
36   \againframe<2>{pspacecomplete}

```

Abbildung 8: Auszug aus dem Quelltext zu den Overlays aus Abbildung 9. Die Option `label=pspacecomplete` hat zwei Effekte: Erstens vergibt sie einen Namen für den Rahmen, der dann an das Kommando `againframe` übergeben werden kann. Zweitens fügt es auf jedem Overlay des Rahmens folgendes Label ein: `pspacecomplete` gefolgt von der Overlaynummer in spitzen Klammern. Da Labels als Ziele von Hybersprüngen dienen dürfen, lassen sich die zwei Overlays des Rahmens bequem anspringen.

The image shows two overlapping Beamer presentation slides. The top slide is titled 'Overhead-Free Languages can be PSPACE-Complete' and contains a theorem stating 'DOF contains languages that are complete for PSPACE.' with a 'Proof details' link. The bottom slide is titled 'Overhead-Free Languages can be PSPACE-Complete' and contains the same theorem, followed by a 'Proof' section with four numbered steps. Both slides have a yellow background and a navigation bar at the bottom with the text 'Computation with Absolutely No Space Overhead' and a 'Return' button. The slides are part of a presentation by Hemaspaandra, Mukherji, and Tantau.

Slide 1 (Top):

- Models: ○○○○○○○○
- Power of the Model: ○○○○○○○○○○●●●
- Limitations of the Model: ○○○○
- Summary: ○○
- Languages Complete for Polynomial Space
- Overhead-Free Languages can be PSPACE-Complete**
- Theorem**
- DOF contains languages that are complete for PSPACE.
- ▶ Proof details
- Hemaspaandra, Mukherji, Tantau
- Computation with Absolutely No Space Overhead

Slide 2 (Bottom):

- Appendix: ●
- Complete Languages
- Overhead-Free Languages can be PSPACE-Complete**
- Theorem**
- DOF contains languages that are complete for PSPACE.
- Proof**
- ① Let $A \in \text{DLINSPACE}$ be PSPACE-complete. Such languages are known to exist.
- ② Let M be a linear space machine that accepts $A \subseteq \{0, 1\}^*$ with tape alphabet Γ .
- ③ Let $h: \Gamma \rightarrow \{0, 1\}^*$ be an isometric, injective homomorphism.
- ④ Then $h(L)$ is in DOF and it is PSPACE-complete.
- Return
- Hemaspaandra, Mukherji, Tantau
- Computation with Absolutely No Space Overhead

Abbildung 9: Die zwei Overlays des Rahmens aus Abbildung 8. Das obere Overlay ist die 65ste Seite des Vortrags, das untere die 78ste. Das Thema ist `beamerthemebars` zusammen mit Standard- und kleineren Adhoc-Vorlagen. Die Farben sind den Firmenfarben der Lufthansa entlehnt.

neu fragen: Welches Thema ist das geeignetste? Welche visuellen Mittel wie Farbe oder Schrift kommunizieren meine „Botschaft“ am besten?

Form Follows Function

Alle Themen folgen einem übergeordneten Designprinzip: „form follows function“. Dieses von der Bauhausschule propagierte Konzept sieht vor, dass die Form (das Aussehen) eines Objektes vorrangig von dessen Funktion (seiner Benutzung) abgeleitet wird.

Betrachten wir als konkretes Beispiel die rechte Seitenleiste in Abbildung 7. Sie ist nicht einfach Verzierung. Vielmehr gibt sie den Zuhörern eine visuelle Rückmeldung über den Fortschritt des Vortrags: Sie zeigt an, welche Abschnitte schon behandelt wurden und welche noch behandelt werden, sie erinnert daran, wie der oder die Vortragende heißt und worum der Vortrag überhaupt geht, und sie ist „anklickbar“, wodurch man Teile des Vortrags überspringen kann oder auch zu bestimmten Abschnitten zurückkehren kann.

Ganz ähnlich verhält es sich mit den verschieden farbigen Balken in Abbildung 10. Im oberen Bereich sind links die Abschnitte des Vortrags, rechts die Unterabschnitte des aktuellen Abschnitts angezeigt. Der jeweils aktuelle (Unter-)Abschnitt ist farblich hervorgehoben. Wiederum sind diese Balken „anklickbar“.

In jedem Vortrag werden im unteren Bereich eine ganze Reihe Symbole eingeblenet. Diese erlauben es, zum nächsten oder vorherigen Rahmen, Unterabschnitt oder Abschnitt oder auch an dessen Ende oder Anfang zu springen. Dazu klickt man auf die Pfeile links und rechts eines Symbols beziehungsweise auf dessen linke oder rechte Seite. Mit einem Klick auf die rechte Seite des Rahmensymbols lassen sich so beispielsweise alle weiteren Overlays eines Rahmens überspringen und das Endresultat anzeigen.

Ein weiteres Beispiel des Designprinzips ist in Abbildung 12 dargestellt. Die Symbole dienen nicht (nur) der optischen Auflockerung, sondern klären den Betrachter auch darüber auf, dass die erste Literaturreferenz ein Buch ist, vermutlich ein Überblickswerk, und dass die folgenden zwei Referenzen Papiere sind.

Schon in der Grundeinstellung sind `beamer`-Vorträge recht farbenfroh. Die Farbgestaltung folgt aber einem festen – dadurch für die Zuhörer leicht zu verinnerlichenden – Schema: Die so genannte Strukturfarbe (blau in der Voreinstellung) signalisiert „strukturelle“ Elemente wie Rahmen, Überschriften

Introduction
 Review
Finding Paths in Tournaments
 Summary

Complexity of: Does a Path Exist?
Complexity of: Construct a Shortest Path
 Complexity of: Approximating the Shortest Path

Proof That $\text{DISTANCE}_{\text{tourn}}$ is NL-complete

Reduce REACH to $\text{DISTANCE}_{\text{tourn}}$

- 1 Is input (G, s, t) in REACH?
- 2 Map G to G' .
- 3 Query:
 $(G', s', t', 3) \in \text{DISTANCE}_{\text{tourn}}?$

Correctness

- 1 A path in G induces a length-3 path in G' .
- 2 **A length-3 path in G' induces a path in G .**

Example

Till Tantau
The Complexity of Finding Paths in Tournaments

Abbildung 10: Beispielseite einer mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ erstellten **beamer**-Präsentation. In der Präsentation wird das Standardfarbschema benutzt: normaler Text ist schwarz, hervorgehobener rot, Beispiele sind grün und alle „strukturellen“ Komponenten sind blau.

und ähnliches. Normaler Text ist schwarz, was den Kontrast maximiert. Rot wird für Hervorhebungen benutzt – das in normalen Texten übliche Schrägstellen ist bei Vorträgen in der Regel als Hervorhebung nicht stark genug. Ein dunkles Grün wird für Beispiele verwendet. In den Abbildungen 10 und 11 sind Beispielseiten abgebildet, die dem Farbschema folgen.

Änderung des Aussehens mittels Vorlagen

Als Nutzer einer Klasse wie **beamer** wird man früher oder später Teile des Aussehens von Vorträgen ändern wollen, ohne gleich ein komplett neues Thema programmieren zu wollen. Man stellt schnell kleine Wünsche: Wäre es

Introduction
Review
Finding Paths in Tournaments
Summary

Complexity of: Does a Path Exist?
Complexity of: Construct a Shortest Path
Complexity of: Approximating the Shortest Path

The Complexity Class Hierarchy

Till Tantau

The Complexity of Finding Paths in Tournaments

Abbildung 11: Eine weitere Beispielseite aus einer mit \LaTeX erstellten **beamer**-Präsentation. Auch hier wird das Standardfarbschema eingehalten. Obwohl die Seite völlig anders aufgebaut ist als die Seite aus Abbildung 10, erkennt man, dass die Texte an den Seiten offenbar Beispiele für Dinge in den Klassen in der Mitte sein sollen. Ebenso ist sofort klar, dass die Klasse AC^0 in einem besonderen Verhältnis zu den anderen steht.

nicht netter, wenn die Rahmenüberschrift ein wenig größer wäre? Oder eine Spur fetter? Und am liebsten auch ein ganz klein wenig weiter unten.

Das Aussehen der von der **beamer**-Klasse erstellten Vorträge wird durch so genannte „Vorlagen“ (englisch *templates*) gesteuert. Vorlagen werden auch von \LaTeX selbst zur Steuerung des Aussehens bestimmter Textelemente genutzt, wenn auch etwas halbherzig. In **beamer** ist das Vorlagenkonzept „konsequent zu Ende gedacht“, wodurch allerdings mit einigen \LaTeX -Traditionen gebrochen wird.

Introduction Review Finding Paths in Tournaments Summary	Summary For Further Reading
---	--------------------------------

For Further Reading

-  **John Moon.**
Topics on Tournaments.
 Holt, Rinehart, and Winston, 1968.
-  **Arfst Nickelsen and Till Tantau.**
 On reachability in graphs with bounded independence number.
 In *Proc. of COCOON 2002*, Springer-Verlag, 2002.
-  **Till Tantau**
 A logspace approximation scheme for the shortest path problem for graphs with bounded independence number.
 In *Proc. of STACS 2004*, Springer-Verlag, 2004.
 In press.



Till Tantau
The Complexity of Finding Paths in Tournaments

Abbildung 12: In dieser Bibliographie wird durch die unterschiedlichen Symbole verdeutlicht, um welche Art Referenzen es sich handelt.

Nehmen wir an, unsere Designabteilung beschließt, dass nur fette, rote, zentrierte Rahmentitel der Wichtigkeit von Rahmentiteln gerecht werden. Würden wir einen solchen Rahmentitel „per Hand“ setzen, so könnte dies beispielsweise wie folgt geschehen:

```

1 \frame{
2   \begin{centering}
3     \color{red}
4     \textbf{Der Titel dieses Rahmens}\par
5   \end{centering}
6   Text, der zum Rahmen gehört.
7 }

```

Das Ziel ist nun, dafür zu sorgen, dass der Befehl `\frametitle` genau die `centering`-Umgebung einfügt – nur sollte natürlich der Text „Der Titel dieses Rahmens“ durch den aktuellen Rahmentitel ersetzt werden. Dazu ändern wir die Vorlage für den Rahmentitel so, dass sie genau den gewünschten Text enthält. An der Stelle, an der der aktuelle Rahmentitel eingefügt werden soll, steht statt „Der Titel dieses Rahmens“ das Kommando `\insertframetitle`.

```

1 \useframetitletemplate{
2   \begin{centering}
3     \color{red}
4     \textbf{\insertframetitle}\par
5   \end{centering}}

```

Praktisch alle Aspekte des Aussehens einer `beamer`-Präsentation werden durch Vorlagen festgelegt. Es gibt Vorlagen für die Titelseite, für den Hintergrund, für die Symbole in Aufzählungen, für die Einträge im Inhaltsverzeichnis, für die kleinen Minirahmensymbole in den Navigationsleisten und so weiter. Es gibt sogar eine Vorlage für das Aussehen von Fußnotensymbolen. In all diesen Vorlagen steht eine große Menge unterschiedlicher `\insertxxxx`-Befehle zur Verfügung, die solch unterschiedliche Dinge wie den Rahmentitel, den Namen des Vortragenden, den aktuellen Abschnittsnamen oder auch die Nummer der aktuellen Fußnote einfügen.

Für viele Standardsituationen sind bereits geeignete Vorlagen vordefiniert. Will man beispielsweise in `itemize`-Umgebungen kleine Kugeln als aufzählende Symbole benutzen, so genügt es, den Befehl `\beamertemplateballitem` aufzurufen. Dieser ersetzt die betroffenen Vorlagen durch geeignete Definitionen.

Die Themen der `beamer`-Klasse bauen intern auf Vorlagen auf. Wie ein Blick beispielsweise in die Datei `beamerthemebars.sty` zeigt, installiert das Thema lediglich einen Satz von Standardvorlagen und zwei speziellere Vorlagen für die Kopf- und Fußzeile.

Umfeld der Klasse

Die `beamer`-Klasse versteht sich gut mit Standard-L^AT_EX und `dvips`, mit `pdfLATEX` und auch mit `LAX`. Ursächlich für die nahtlose Zusammenarbeit ist die vollständige Implementation in L^AT_EX. Alle Funktionen, auch das Overlaying, werden durch T_EX- und `\special`-Befehle bewerkstelligt und nicht

durch externe Programme. Neben `dvips` werden auch alle darauf aufbauende Treiber unterstützt, einzig `dvipdfm` bleibt im Moment leider außen vor.

Die grundlegende Entscheidung, auf externe Programme zu verzichten, bringt einige kleinere Einschränkungen mit sich. So verträgt sich `verbatim`-Text mit Overlays eher schlecht. Soll ein Rahmen `verbatim`-Text enthalten, so muss dies durch die Rahmen-Option `containsverbatim` angezeigt werden und der Rahmen wird (höchstens) eine Seite enthalten. Benötigt man `verbatim`-Text auf Rahmen mit mehreren Overlays, so müssen diese etwas umständlich vorher deklariert werden. Ein zweites, weniger schwer wiegendes Problem ist der Umstand, dass `\frame` prinzipbedingt nicht als Umgebung, sondern nur als Befehl zur Verfügung steht.

Die LyX-Schnittstelle macht es leicht, einen Vortrag ohne genaue L^AT_EX-Kenntnisse zu erstellen – aber auch alte Hasen können von LyX profitieren, wenn sie „schnell mal“ einen Vortrag erstellen wollen. Ein Screenshot von LyX mit einer geladenen `beamer`-Präsentation ist in Abbildung 13 dargestellt. Abbildungen 10 bis 12 zeigen, dass sich auch komplexe Overlays mit LyX meistern lassen, obwohl diese Rahmen in LyX zugegebenermaßen noch einigen T_EX-Text enthalten.

Interessante, verbreitete Alternativen zur Klasse `beamer` sind `prospcr` und deren Erweiterung `HA-prospcr`. Die `beamer`-Klasse verträgt sich gut mit diesen Schwesterklassen: Mit der in der Stildatei `beamerprospcr` definierten Emulationsschicht lässt sich `prospcr`-Quelltext direkt von `beamer` verarbeiten. Prosper-eigene Befehle wie `\fromSlide` werden von der Emulationsschicht – soweit prinzipiell überhaupt möglich – auf passende `beamer`-Befehle abgebildet. Neben dem Portieren und Recyceln von `prospcr`-Vorträgen eröffnet die Emulationsschicht auch die Möglichkeit, Artikelversionen von `prospcr`-Vorträgen zu erzeugen – eine Fähigkeit, die der `prospcr`-Klasse selbst leider fehlt.

Entwicklung der Klasse

Die frühesten Codefragmente der `beamer`-Klasse reichen in mein Studium zurück. Für Seminarvorträge brauchte ich ein Programm, mit dem sich leicht Folien für Overhead-Projektoren erstellen ließen. Da die Vorträge in aller Regel viel mathematischen Text enthielten und ich L^AT_EX für meine sonstigen Texte benutzte, hielt ich Ausschau nach L^AT_EX-Paketen zur Erstellung von Folien und wurde mit der `seminar`-Klasse schnell fündig. Wie viele Leser si-

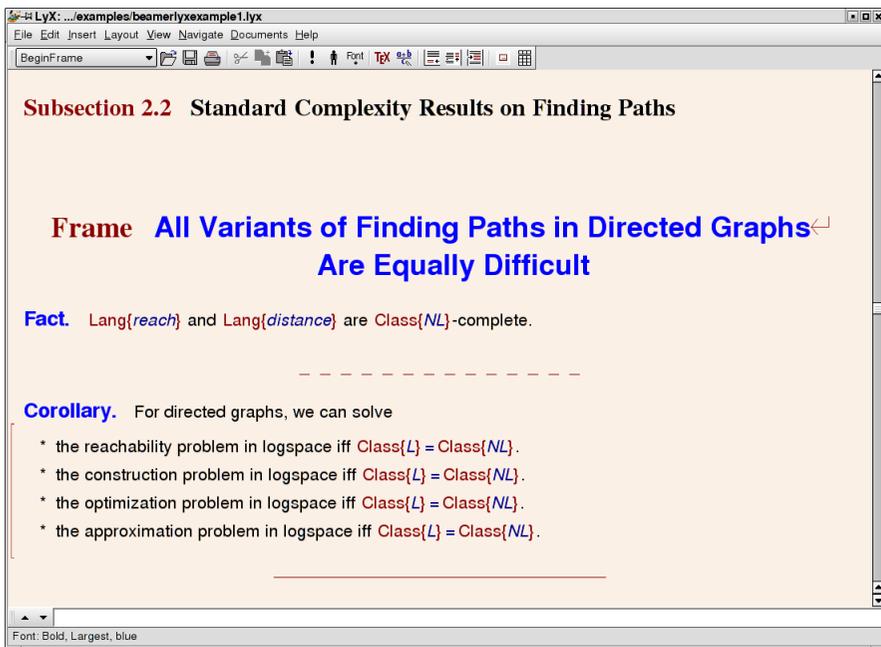


Abbildung 13: Screenshot von LyX mit dem „Quelltext“ einer `beamer`-Präsentation. Die gestrichelte Linie leitet eine Pause ein, wodurch das Korollar erst auf dem zweiten Overlay des Rahmens aufgedeckt wird. Statt der Pause hätte man auch die Overlay-Spezifikation `<2->` am Anfang des Korollars einfügen können. Die langgezogene Linie markiert das Ende des Rahmes (sie kann weggelassen werden, wenn wieder ein Rahmen oder ein Abschnitt folgt).

cherlich auch, begann ich bald, einen persönlichen Fundus nützlicher Makros aus sich wiederholenden Lösungen für immer wieder vorkommende Probleme, wie das Setzen von Spalten, zu destillieren. Diese Makros bilden den ursprünglichen Kern der Klasse.

Zu einer eigenen L^AT_EX-Klasse wurde `beamer` erst vor rund einem Jahr. Anlass war ein für mich besonders wichtiger Vortrag: meine Dissertationsdisputation. Einige meiner Kollegen hatten mich damit verrückt gemacht, dass dieser Vortrag unbedingt „perfekt“ sein müsse. Da ich deshalb viel Zeit auf die Vorbereitung dieses Vortrags verwenden wollte, schien mir die Gelegenheit

günstig, endlich Nägel mit Köpfen zu machen. Da dieser Vortrag der erste (!) Vortrag sein würde, den ich mit einem Beamer halten würde, war der Name für die Klasse auch leicht gefunden. Anders als man vielleicht denken könnte, ist „Beamer“ ein „deutsches“ Wort, das von Englisch sprechenden Menschen bestenfalls mit Star Trek in Verbindung gebracht wird. Beamer heißen im Englischen etwas technikverliebt „LCD projectors“ oder „video projectors“.

Das Design dieses ersten mit `beamer` gesetzten Vortrags überlebt heute weitgehend in Form des doch recht schlichten Standardthemas. Die Fähigkeiten der Klasse waren im Vergleich zu heute recht beschränkt: Weder wurden Overlays unterstützt, noch gab es Themen. Lediglich das Inhaltsverzeichnis und die obere Navigationsleiste wurden automatisch erzeugt. Trotzdem wurde ich von Zuhörern des Vortrags gebeten, ihnen doch die Klasse zu schicken. Dieser Bitte kam ich erst Monate später nach – nach wiederholter Aufforderung auch von anderer Stelle –, da mir die Klasse zu sehr auf meine eigenen Bedürfnisse zugeschnitten schien. Ich wollte sie schlichtweg niemand anderem „zumuten“.

Anstatt die Klasse per E-Mail an interessierte Kollegen zu schicken, schickte ich sie an das Comprehensive T_EX Archive Network (CTAN) – was sich als Segen und auch als Fluch herausstellte. Meine Vorstellung war gewesen, dass die Klasse im Dickicht der Pakete untergehen würde, ich aber die Adresse leicht an interessierte Kollegen schicken könnte. Statt dessen trudelten bereits in den ersten Tagen über ein Dutzend Fehlerberichte und Erweiterungswünsche von Leuten ein, die kreuz und quer über den Globus verteilt waren. Da mir sowohl die Fehlerberichte wie die Änderungswünsche berechtigt erschienen (und sich oft mit meinen eigenen Wünschen deckten), fing ich an, diese Berichte und Wünsche einzuarbeiten und das Ergebnis wieder an CTAN zu schicken – um dann Ruhe zu haben.

Leider hat sich seitdem weder an der Frequenz der Erweiterungswünsche, noch an deren Berechtigung viel geändert. Bis die ersehnte Ruhe in der Entwicklung der `beamer`-Klasse einkehrt, wird es wohl noch eine Weile dauern.

Zusammenfassung

In diesem Artikel wurde ein Überblick über die wesentlichen Eigenschaften, das Design und die Entstehung der `beamer`-Klasse gegeben. Die Klasse kann das Erstellen komplexer Vorträge mittels L^AT_EX, pdfL^AT_EX oder L_YX erheblich erleichtern. Das automatische Inhaltsverzeichnis und die Navigationsleiste

machen es auch Anfängern leicht, strukturierte Vorträge zu erstellen, bei denen weder der Vortragende noch die Zuhörer den roten Faden verlieren.

Trotz all seiner Fähigkeiten ersetzt die Klasse nicht Erfahrung und Augenmaß: Nicht jedes Thema ist für jeden Vortrag geeignet, nicht jedes Overlay macht den Vortrag verständlicher. Als Leitschnur sollte man sich an zwei alte typografische Regeln halten:

- Weniger ist mehr.
- Die beste Typografie ist die, die man nicht bemerkt.

Wie in einem gut gesetzten Buch sollte die Botschaft von der Schrift an der Wand zu den Zuhörern fließen, ohne dass sie dabei typografische oder optische Stolpersteine überwinden muss.

Zu finden ist die Klasse beispielsweise im CTAN-Verzeichnis `macros/latex/contrib/beamer`. Die Weiterentwicklung der Klasse wird auf SourceForge, einem bekannten Server für die Entwicklung von freier Software, unter `latex-beamer.sourceforge.net` betrieben. Teil der Klasse ist eine Beispielsammlung und eine ausführliche Dokumentation, in der sowohl die in diesem Artikel angeschnittenen Konzepte erklärt werden, als auch eine Reihe weitere Fähigkeiten dokumentiert sind, die nicht zur Sprache kamen.

Farbige Mathematik

Herbert Voß

Die Zahl der Veröffentlichungen, die als PDF allgemein zur Verfügung gestellt werden, nimmt stetig zu. Da diese sehr häufig mit dem Acrobat Reader online betrachtet und gelesen werden, spielt die farbliche Darstellung eine nicht unbedeutende Rolle. Im Folgenden soll daher gezeigt werden, wie man insbesondere für den Mathematikmodus farbige Darstellungen erreichen kann.

Das Prinzip

Grundsätzlich wird nichts weiter als das allgemein bekannte Paket `color` [2] oder das hier behandelte Paket `xcolor` benötigt. Dieses stellt die vordefinierten Farben `black`, `red`, `green`, `blue`, `cyan`, `magenta`, `yellow` und natürlich `white` bereit. Weitere lassen sich über den Befehl `\definecolor` neu definieren.

Partielles Einfärben

Für das Färben einzelner Wörter existiert das Makro `\textcolor`, welches ebenfalls für das Einfärben einzelner Teile einer mathematischen Formel benutzt werden kann.

$$\int_1^2 \frac{1}{x^2} dx = 0.5 \quad (1)$$

```

1 \begin{align}
2 \int_1^2 \textcolor{red}{\frac{1}{x^2}} dx &= 0.5
3 \end{align}

```

Das entsprechende Makro mit der Syntax `\textcolor{<Farbe>}{<Text>}` ist zwar vom Namen her missverständlich, lässt sich jedoch im Mathematikmodus in der gleichen Weise anwenden, ohne dass dieser Modus verlassen werden muss. Gleichung 1 zeigt, dass die Variable x kursiv gedruckt wird, was für den mathematischen Modus Standard ist.

sich die Formel $y = x^2 + px + q$ auf die Gleichung 2. Hierbei spielt es keine Rolle, ob der mathematische Teil komplett als Argument erscheint `\textcolor{blue}{(y=x^2+px+q)}` oder umgekehrt das Makro `\textcolor` innerhalb der Formel, wie bei diesem Ausdruck $c^2 = a^2 + b^2$ (`(\textcolor{magenta}{c^2=a^2+b^2})`).

Im Gegensatz zum Makro `\textcolor` erwarten `\colorbox` und `\fcolorbox` den Parameter im Textmodus, sodass in diesem Fall der mathematische Modus erneut aktiviert werden muss, wenn diese Makros innerhalb einer mathematischen Umgebung angewendet werden (Gleichung 6).

$$E = a_v A - a_f A^{2/3} - a_c \frac{Z(Z-1)}{A^{1/3}} - a_s \frac{(A-2Z)^2}{A} + E_p \quad (6)$$

```

1 \def\xstrut{\vphantom{\frac{(A)^1}{(B)^1}}}
2 \begin{equation}\label{eq:6}
3 E = %
4 \colorbox{darkyellow}{\(\xstrut a_v A\)} +
5 \colorbox{hellmagenta}{\(\xstrut -a_f A^{2/3}\)} +
6 \colorbox{green}{\(\xstrut -a_c \frac{Z(Z-1)}{A^{1/3}}\)} +
7 \colorbox{cyan}{\(\xstrut -a_s \frac{(A-2Z)^2}{A}\)} +
8 \colorbox{yellow}{\(\xstrut E_p\)}
9 \end{equation}

```

Abgesetzte Formeln können ebenso komplett eingefärbt werden, wobei gleichzeitig für Gleichung 7 noch eine partielle Rotfärbung vorgenommen wurde. Die globale Umschaltung auf eine andere Farbe geschieht mit dem Befehl `\color{<Farbe>}`, der einfach an den Beginn der abgesetzten Gleichung gesetzt wird.

$$\left(\prod_{j=1}^n \hat{x}_j \right) H_c = \frac{1}{2} \hat{k}_{ij} \det \hat{\mathbf{K}}(i|i)$$

$$\begin{array}{ccc}
 R \times S \times T & \xrightarrow{\text{restriction}} & S \times T \\
 \text{proj} \downarrow & & \downarrow \text{proj} \\
 R \times S & \xleftarrow{\text{inclusion}} & S
 \end{array} \quad (7)$$

```

1  \[\color{magenta}
2  \left(\prod_{j=1}^n \hat{x}_j\right) H_c = %
3    \frac{1}{2} \hat{k}_{ij} \det \widehat{\mathbf{K}}(i|i)
4  \]
5
6  \begin{align}\label{eq:cd}\color{blue}
7  \begin{CD}
8    R \times S \times T @>\text{restriction}>> S \times T \\
9    @VV\text{proj}VV @VV\text{proj}V \\
10   R \times S @<<\text{\textcolor{red}{inclusion}}<< S
11 \end{CD}
12 \end{align}

```

Farbboxen

Hier sind aus dem Textmodus die Makros `\colorbox` und `\fcolorbox` bekannt, die sich ebenfalls auf den mathematischen Modus übertragen lassen:

$x_{n+1} \leftarrow x_n^2 + c$ in der komplexen Zahlenmenge betrachtet, hat etwas mit Benoit Mandelbrot zu tun.

```

1  \colorbox{yellow}{\leftarrow x_n^2+c)} in der
   komplexen
2  Zahlenmenge betrachtet, hat etwas mit Benoit Mandelbrot zu tun.

```

Ebenso hätte man auch bei dieser im Inline-Modus geschriebenen Formel `\fcolorbox` verwenden können, welches das Gegenstück zu `\fbox` darstellt:

$x_{n+1} \leftarrow x_n^2 + c$. Für den Rahmen ist dann eine zusätzliche Farbe als Parameter zu übergeben `\fcolorbox{red}{yellow}{\leftarrow x_n^2+c)}`.

Etwas schwieriger ist der Fall, wenn man abgesetzte Formeln einfärben will, die gleichzeitig eine Gleichungsnummer aufweisen, denn diese könnte sowohl innerhalb als auch außerhalb der farbigen Box erscheinen. Weiterhin muss die abgesetzte Formel zusätzlich in eine `\parbox`-Umgebung gepackt werden, um dann beispielsweise so etwas wie Gleichung 8 zu erhalten.

$$\frac{I(\alpha)}{I_0} = \begin{cases} \sqrt{1 - \frac{\alpha}{\pi} + \frac{1}{2\pi} \sin 2\alpha} & \varphi = 0 \\ \sqrt{2 \left(1 - \frac{\alpha}{\pi}\right) (2 + \cos 2\alpha) + \frac{3}{\pi} \sin 2\alpha} & \varphi = \frac{\pi}{2} \end{cases} \quad (8)$$

```

1 \colorbox{hellmagenta}{%
2 \parbox{\linewidth-2\fbboxsep}{%
3 \begin{align}\label{eq:3}
4 \frac{I(\alpha)}{I_0}=%
5 \begin{cases}
6 \sqrt{1-\frac{\alpha}{\pi}+\frac{1}{2\pi}\sin 2\alpha}
7 & \varphi =0\ [0.4\text{cm}]
8 \sqrt{2\left(1-\frac{\alpha}{\pi}\right)\left(2+\frac{\cos 2\alpha}{\pi}\right)+\frac{3}{\pi}\sin 2\alpha}
9 & \varphi =\frac{\pi}{2}
10 \end{cases}
11 \end{align}%
12 }
13 }

```

Wenn man dagegen keine Gleichungsnummer hat und die Formel nur für den Bereich einfärben will, den sie auch tatsächlich benötigt, hat man das Problem, dass man schon bei der Definition der `\parbox` ihre Breite kennen muss, die zu diesem Zeitpunkt jedoch noch nicht bekannt ist. Hierbei kann man sich mit dem Paket `empheq` [3] helfen, welches die Aufgabe der Boxbreitenbestimmung vornimmt, wobei dieses Paket dies auch unabhängig davon kann, ob eine Gleichungsnummer erscheint oder nicht.

$$x(t) = \frac{a(t^2 - 1)}{t^2 + 1}$$

$$y(t) = \frac{at(t^2 - 1)}{t^2 + 1}$$

```

1 \begin{empheq}[box=\fcolorbox{blue}{hellcyan}]{align*}
2 x(t) & =\frac{a\left(t^2-1\right)}{t^2+1}\backslash
3 y(t) & =\frac{at\left(t^2-1\right)}{t^2+1}
4 \end{empheq}

```

Für komplett farbig hinterlegte Gleichungen kann noch das Paket `framed` [1] benutzt werden. Es definiert die Umgebung `shaded`, welche die mathematischen Umgebungen einfach umschließen kann. `framed` hat den Vorteil, dass es auch ohne Weiteres über Seitenumbrüche hinweg funktioniert und zudem mehrere, auch verschiedene, mathematische Umgebungen umfassen kann.

$$\iiint_G [u\nabla^2 v + (\nabla u, \nabla v)] d^3V = \oiint_S u \frac{\partial v}{\partial n} d^2A \quad (9)$$

$$\iiint_G [u\nabla^2 v - v\nabla^2 u] d^3V = \oiint_S \left(u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right) d^2A \quad (10)$$

```

1 \begin{shaded}
2 \begin{align}
3 \underset{\{\mathcal{G}\}\quad}{\iiint}\!%
4 \left[u\nabla^2 v + \left(\nabla u, \nabla v\right)\right]d^3V
5 =\underset{\{\mathcal{S}\}\quad}{\oiint} uQ\{v\}\{n\}d^2A
6 \end{align}
7 \begin{align}
8 \underset{\{\mathcal{G}\}\quad}{\iiint}\!%
9 \left[u\nabla^2 v - v\nabla^2 u\right]d^3V%
10 =\underset{\{\mathcal{S}\}\quad}{\oiint}%
11 \left(uQ\{v\}\{n\} - vQ\{u\}\{n}\right)d^2A
12 \end{align}
13 \end{shaded}

```

Zusammenfassung

Es wurde gezeigt, dass auch für den mathematischen Modus das Einfärben von Formeln oder Teilen davon mit den bestehenden Makros ohne weiteres möglich ist. Andere oder weitergehende Möglichkeiten kann man [4] entnehmen.

Literatur

- [1] Donald Arseneau: *framed.sty*; CTAN: macros/latex/contrib/misc/framed.sty; 2003.
- [2] David Carlisle: *color.dtx*; CTAN: macros/latex/required/graphics/color.dtx; 1999.
- [3] Morten Høgholm: *empheq.dtx*; CTAN: macros/latex/contrib/empheq/; 2003.

- [4] Herbert Voß: *Mathmode*; <http://www.perce.de/LaTeX/math/Mathmode-TeX.pdf>; 2004.

Zeichnen von Sternkarten mit PP3

Torsten Bronger

PP3 ist ein Programm, das \LaTeX und `pstricks` benutzt, um Sternkarten zu zeichnen. Dabei wird auf grafische und typografische Qualität und auf Konfigurierbarkeit besonderer Wert gelegt.

Einleitung

In der ersten Blütezeit der Shareware und Public-Domain-Software waren Astronomieprogramme äußerst beliebt, insbesondere Programme, die auf die eine oder andere Art und Weise den Sternhimmel darstellten. Dieser Boom verflachte aber ab Mitte der 90er immer mehr. (Halb-)kommerzielle Programme haben inzwischen den größten Teil dieses Software-Bereichs übernommen [2]. Der wichtigste freie Vertreter ist heute *Cartes du Ciel* [4], das leider nur auf der Windowsplattform läuft. Es bietet jedoch Zugang zu sehr umfangreichen Sternen- und Nebelkatalogen.

Diese Programme sind mit ihrer Detailfülle und ihrer Funktionsvielfalt perfekt geeignet, um eine Beobachtungsnacht vorzubereiten und zu begleiten oder um in der Lehre eingesetzt zu werden.

Manchmal ist jedoch weniger mehr. Wenn man ein Aufsuchkärtchen oder eine Sternbild-Darstellung für einen Himmelsführer erstellen möchte, reichen schon ein paar Dutzend Sterne pro Sternbild aus. Viel wichtiger ist jedoch, dass die Karten übersichtlich gestaltet sind. Dabei ist zu überlegen, welche Himmelsobjekte eine Bezeichnung bekommen sollen, wie diese aussehen und wohin sie platziert werden soll. Die vollautomatischen Programme bieten da zu wenige Möglichkeiten.

Schlimmer noch: Das Ergebnis dieser Programme ist bestenfalls eine hochauflösende Bitmap mit leidlich guter Typografie. \LaTeX kann das besser. Wenn

man dann noch irgendwie dem DVI-Treiber sagt, wo die Linien, Sterne und Nebel hingehören, wird eine grafisch ansprechende Sternkarte möglich.

Die Grundidee von PP3

Das Programm PP3 [3] liest eine Skriptdatei, in der der Benutzer seine gewünschten Parameter vorgibt und daraus eine \LaTeX -Datei erzeugt, deren Bearbeitung die Sternkarte ergibt.

Dabei macht PP3 massiv vom `pstricks`-Paket [10] Gebrauch, das den besten Zugang zu PostScript bietet und daher die beste Möglichkeit ist, anspruchsvolle Bilder innerhalb eines \LaTeX -Dokumentes zu zeichnen. Es ist außerdem sehr leicht zu handhaben.

Es handelt sich hier also um *generiertes* \LaTeX . Theoretisch könnte man hingehen und aus der \LaTeX -Datei die Sternkarte extrahieren – sie ist eine einzige `\vbox` – und in das eigene Dokument einfügen. Man würde sich so aber bloß Ärger einhandeln, da die Bearbeitung durch \TeX dann jedesmal (unter Umständen sehr viel) länger dauert, eventuell sogar mangels Speicher schiefgehen kann, und Einstellungen im umgebenden Dokument die Sternkarte beeinflussen können.

Ein solcher Weg ist gar nicht nötig. PP3 ruft `dvips` und eventuell `ps2pdf` auf, um letztendlich eine EPS- bzw. PDF-Datei zu liefern. Diese kann man dann ganz einfach in ein Dokument einbinden oder sonstwie verarbeiten.

Das Eingabeskript

Im Eingabeskript gibt der Benutzer seine Wunschliste an, also was für eine Sternkarte er gerne hätte. Theoretisch kann dafür auch eine leere Datei dienen, dann bewirken die Vorgabewerte, dass eine Karte des Orion ausgegeben wird. Man kann nun Schritt für Schritt diese Vorgabewerte überschreiben.

Das folgende Skript ist beispielsweise geeignet, eine Karte des Skorpions zu zeichnen:

```

1 # Scorpius, der Skorpion
2 #
3 filename output sco.tex
4
5 set constellation SCO
6 set center_rectascension 16.8

```

```

7  set center_declination -30
8  set box_height 9
9  set box_width 9
10 set grad_per_cm 4

```

Die Zeilen 1 und 2 sind einfach Kommentare, die mit einem freistehenden ‘#’ eingeleitet werden. Zeile 3 sagt aus, dass PP3 die Sternkarte in `sco.tex` schreiben soll.

Zeile 5 gibt PP3 einen Hinweis, dass wir das Sternbild Skorpion zeichnen wollen.¹ Zur Zeit bedeutet das lediglich, dass die Sternbildgrenzen des Skorpions in einer anderen Farbe gedruckt werden.

Die Zeilen 6 und 7 geben die Himmelskoordinaten des Zentrums der Abbildung an, und zwar in Rektaszension (in Stunden) und Deklination (in Grad). Selbstredend fallen diese Koordinaten mit dem Zentrum des Sternbildes Skorpion zusammen.

Die nächsten beiden Zeilen 8 und 9 bestimmen die Größe der Abbildung. Alle Längenangaben in PP3 werden in Zentimetern gemessen.

Die letzte Zeile schließlich ist die Auflösung bzw. der Maßstab. Der Vorgabewert dafür ist 3 grad/cm. Man darf das nicht mit einer globalen Skalierung verwechseln, die man später vielleicht im `\includegraphics`-Befehl vornimmt, denn alle Schriftgrößen, Linienstärken etc. machen eine *Maßstabs*-Änderung *nicht* mit.

Man kann nun dieses kleine Skript in der Datei `sco.pp3` speichern und

```

1  pp3 sco.pp3
2  latex sco
3  dvips sco

```

aufzurufen. Die sich ergebende Karte ist in Abbildung 1 zu sehen. Allerdings sind insbesondere die Farben für einen tonersparenden Ausdruck auf einem gewöhnlichen Schwarz/Weißdrucker ungünstig. Also setzen wir sie auf Werte, die dafür besser geeignet sind:

```

1  switch colored_stars off
2  color stars 0 0 0
3  color nebulae 0 0 0   color background 1 1 1

```

¹ „SCO“ ist die astronomische Abkürzung dafür.

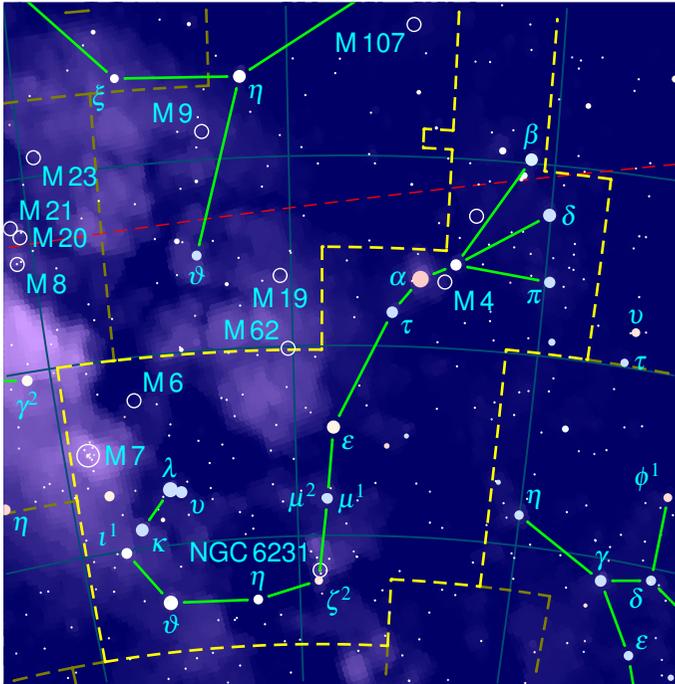


Abbildung 1: Das Sternbild Skorpion.

```

4 color grid 0.5 0.5 0.5 color ecliptic 0.3 0.3 0.3
5 color constellation_lines 0.7 0.7 0.7 # Linien zwischen Sternen
6 color labels 0 0 0
7 color boundaries 0.8 0.8 0.8 # Sternbildgrenzen
8 color highlighted_boundaries 0 0 0 # Selektiertes Sternbild, hier
  Skorpion
9 color milky_way 0.5 0.5 0.5

```

Farben werden bei PP3 in der Form „Rot Grün Blau“ (RGB-Farbmodell) eingegeben, wobei die einzelnen Werte von 0.0 bis 1.0 reichen. „0 0 0“ ist also Schwarz, „1 1 1“ Weiß, und beispielsweise „1 1 0“ ist Gelb.

Die erste Zeile sorgt dafür, dass alle Sterne dieselbe Farbe erhalten. Normalerweise versucht PP3 nämlich, die Originalfärbung des Sternes anhand seiner

Spektralklasse nachzuahmen. Nur durch diese Zeile bekommt Zeile 2 einen Sinn: Alle Sterne werden nun Schwarz gedruckt.

Die restlichen Angaben sollten weitgehend selbsterklärend sein. Das Ergebnis ist also eine Karte des Skorpions in Schwarz auf Weiß.

Bezeichnungen (Labels)

Bis hierhin ist die ganze Sache noch einigermaßen konventionell, sieht man mal von der Tatsache ab, dass PP3+L^AT_EX+dvips Vektorgrafiken statt Bitmaps erzeugt, was für viele Anwendungen bereits ein großer Vorteil ist.

Es wäre jedoch schade, wenn L^AT_EX als bloßer pstricks-Bereitsteller herhalten müsste. Stattdessen kommt es ebenso für alle Bezeichnungen (Labels) zum Einsatz. Dafür liegen die Labels in PP3-Datendateien bereits in L^AT_EX-Form vor (beispielsweise `\(\phi^{\{1\}}\)`).

Zusätzlich kann man Labels aber auch ausdrücklich im Eingabeskript setzen. Als Erstes muss man dazu das Eingabeskript durch den Befehl `objects_and_labels` in zwei Teile teilen, in denen jeweils nur bestimmte Befehle erlaubt sind. Der erste Teil ist der schon bekannte, während der zweite Teil Himmelsobjekte und ihre Labels löscht, ergänzt, verändert oder repositioniert. In diesen zweiten Teil kann man beispielsweise schreiben

```
set_label_text SCO 21 "\fontnotesize Antares"
```

Das gibt dem Stern 21 Sco das Label „Antares“, außerdem stellt es dafür auf eine kleinere Schrift um. „21 Sco“ ist die astronomische Bezeichnung für Antares, „21“ ist dabei die so genannte Flamsteed-Nummer. Für sehr südliche Sternbilder, die Herr Flamsteed von London aus nicht sehen konnte, und für sehr kleine Sterne muss man die Henry-Draper-Katalognummern benutzen:

```
set_label_text HD 128620 "\fontnotesize Toliman" # Alpha
Centauri
```

(Das Programm *Celestia* [8] ermöglicht, diese Nummern für einen gegebenen Stern bequem herauszufinden.)

Labelkollision vermeiden

Eine wichtige Funktion von PP3 ist die Vermeidung von Überlappungen von Labels mit Himmelsobjekten, Linien oder anderen Labels. Die Herangehens-

weise dafür ist ein wenig von dem Zeilenumbruchalgorithmus von T_EX inspiriert: PP3 probiert die acht Positionen der Windrose um das jeweilige Himmelsobjekt, das bezeichnet werden soll, aus und berechnet für jede Position eine Strafpunkte-Zahl, die sich durch die verschiedenen möglichen Arten von Überlappungen ergibt. Die Position, die die geringste Strafpunkte-Zahl aufweisen kann, wird genommen. Übersteigen diese Strafpunkte einen bestimmten Wert, wird das Label unterdrückt.

Man kann diesen automatischen Mechanismus durch explizite Positionierungsangabe überstimmen:

```
reposition ORI 34 E ; # Mintaka
```

Diese Anweisung setzt das Label für den Stern Mintaka im Orion auf östliche Position (also rechts). Die acht möglichen Positionen werden mit den typischen englischen Abkürzungen der Windrose bezeichnet.

Es zeigt sich, dass in der Praxis nur selten in diesen automatischen Algorithmus zur Label-Erzeugung eingegriffen werden muss.

Manuelle Labels

Labels werden als einfache L^AT_EX-Boxen in die Karte gedruckt. Innerhalb dieser Boxen kann man allerlei L^AT_EX- und pstricks-Tricksereien anwenden wie in

```
text "\\small Wolf 359\\hskip0.3em
      \\psdots[dotstyle=+,dotangle=45](0,0)"
      at 10.902 7.32 color 0.3 0.3 0.9333 towards NW ;
```

Dieses Kommando druckt ein kleines ‚×‘ an die Position des Sterns Wolf 359 im Löwen und schreibt „Wolf 359“ in einer grau-blauen Farbe oben links daneben.

Das Kommando `text` setzt grundsätzlich alle benutzerdefinierten Labels. Es folgt der Label-Text im L^AT_EX-Format. Das Schlüsselwort `at` leitet die Himmelskoordinaten ein, in diesem Fall 10^h 54' 7,2" Rektaszension und +7,32° Deklination. Es folgen optional die Farbe nach `color` und die Lage der Box nach `towards`.

Im obigen Beispiel ist die Lage `NW`, also oben links; das bedeutet, dass die rechte untere Ecke der Box genau auf den gegebenen Himmelskoordinaten

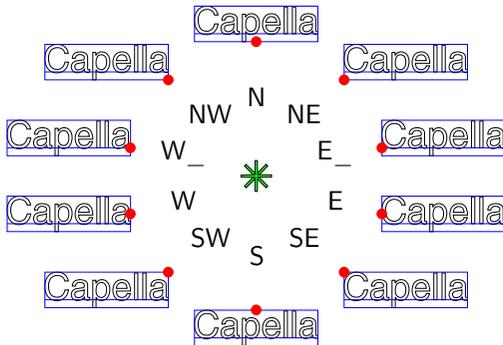


Abbildung 2: PP3s Windrose und die sich ergebene Ausrichtung. Der rote Punkt liegt jeweils exakt auf den Himmelskoordinaten. Dieser Parameter kommt bei `towards` und `reposition` zum Einsatz.

liegt. Daher kommt das `\psdots`-Makro, das das ‚ \times ‘ erzeugt, als *letztes* in der Reihe der \LaTeX -Befehle.

Wäre die Lage beispielsweise „`towards N`“ gewesen, wäre die *Mitte* der unteren Kante der Box auf den Himmelskoordinaten zu liegen gekommen. Man beachte ferner, dass die äußersten Boxgrenzen und nicht die Grundlinie als Bezug benutzt werden. Die Abbildung 2 zeigt alle möglichen Parameter für `towards` und welche Ausrichtung sie bewirken.

Offensichtlich ist es für den Trick mit dem ‚ \times ‘ besser, „`towards W_`“ zu benutzen, denn beispielsweise der Sternname „Luyten 726-8“ hat eine Unterlänge und würde daher bei Angabe von „`NW`“ mitsamt dem ‚ \times ‘ etwas nach oben verschoben.

Wenn immer dieselben Konstrukte in den Labels vorkommen, ist es selbstverständlich möglich und sinnvoll, daraus \LaTeX -Makros zu machen und diese in eine benutzerdefinierte Präambel (s. u.) einzubauen. Dann werden die Labels etwas übersichtlicher.

Gebogene Labels: Flexes

Gerade in der Nähe der Himmelspole kann es grafisch ansprechend sein, eine Bezeichnung nicht als gewöhnliche Box zu setzen, sondern als Schriftzug, der

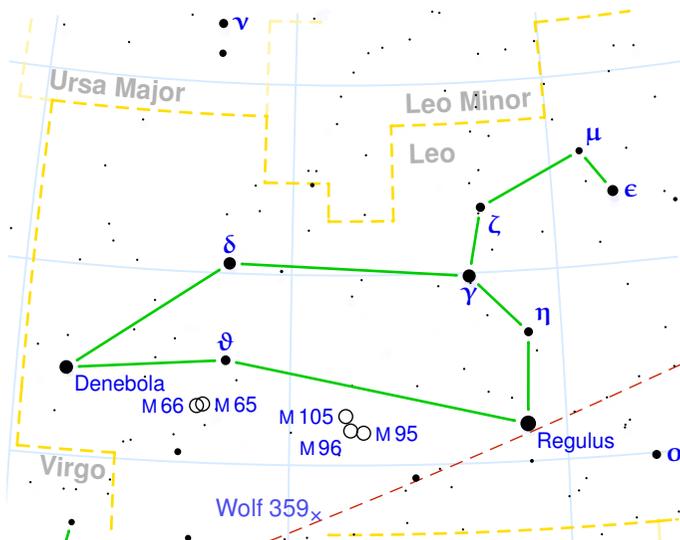


Abbildung 3: Das Sternbild Löwe mit verschiedenen Arten von Labels.

auf einem Deklinationskreis ($\hat{=}$ Breitenkreis) entlangläuft. Das `pst-text`-Paket von `pstricks` stellt einen solchen Spezialeffekt zur Verfügung. Das Problem besteht also nur darin, das entsprechende Fragment des Deklinationskreises zu finden, insbesondere seine Länge zu bestimmen. PP3 kann sie wegen der nicht längentreuen Abbildung nur abschätzen, macht es aber ganz ordentlich. Das Ergebnis sind so genannte Flex-Labels oder kurz Flexes. Man kann sie sehr schön für etwas längere Labels, beispielsweise die Sternbildnamen, benutzen:

```
text "Ursa Major" at 12 30 along declination towards SE ;
```

Der einzige Unterschied ist also der Schlüsselausdruck „along declination“.

Als Beispiel für die Labels (und etwas mehr) gebe ich hier das Skript für das Sternbild Löwe an, dessen Ergebnis man in Abbildung 3 sieht:

```
1 # Leo, der Löwe
2
3 filename output leo.tex
4 filename include wiki.pp3
```

```

5  switch eps_output on
6
7  set constellation LEO
8  set center_rectascension 10.8    set center_declination 20
9  set box_height 7    set box_width 9
10
11 objects_and_labels
12
13 reposition LEO 32 SE ; # Regulus
14 add_labels LEO 24 ;
15
16 text Leo at 10.55 27 along declination towards SE ;
17 text "Leo Minor" at 10.05 28.5 along declination towards NW ;
18 text "Ursa Major" at 12 30 along declination towards SE ;
19 text Virgo at 11.65 10 along declination towards SW ;
20
21 text "\\small Wolf 359\\hskip0.3em
22     \\psdots[dotstyle=+,dotangle=45](0,0)"
23     at 10.902 7.32 color 0.3 0.3 0.9333 towards W_ ;
24 delete LEO 63 HD 97605 ;

```

Zeile 4 zeigt, wie man eine globale Stildatei einbinden kann: Die Datei `wiki.pp3` ist vom Aufbau her ein gewöhnliches Eingabeskript, in dem Präambel, Farben, einige Sternbildnamen wie „Regulus“ etc. gesetzt werden. So etwas ist sehr hilfreich, wenn man eine ganze Reihe von Karten für ein Projekt erzeugt.

Zeile 5 weist PP3 an, direkt `dvips` aufzurufen, um eine EPS-Datei zu erzeugen. Es existiert ein analoger Befehl für PDF.

In Zeile 11 wird der zweite Teil eingeläutet: Objekte und Labels werden eingestellt: Zeile 13 zieht den Namen „Regulus“ von der gestrichelten Ekliptiklinie weg, auf die PP3 ihn sonst gesetzt hätte. Die Ekliptik nimmt bislang leider noch nicht an dem oben erwähnten Strafpunkte-Schema teil.

Zeile 14 fügt das Label von 24 Leo ($\equiv \mu$ Leo) ein, das PP3 standardmäßig unterdrückt hätte, weil der Stern zu schwach ist. Zeile 24 schließlich macht das Umgekehrte: Da 63 Leo und HD 97605 dem Schriftzug „Wolf 359“ gefährlich nahe kämen, werden beide einfach aus der Karte gelöscht. Alternativ könnte man mit dem `\colorbox`-Befehl den Schriftzug auf einen weißen Hintergrund setzen.

Koordinaten-Labels

Manchmal besteht der Wunsch, die Gitterlinien des Koordinatensystems mit automatisch generierten Labels zu beschriften. Das können Befehle wie

```
1 text "\(#3\)" at 0 20 along declination tics rectascension 1
   towards N ;
2 text "\(#5\)" at 11 0 along declination tics declination 10
   towards S ;
```

leisten. Die `tics`-Option sorgt für eine Vervielfältigung des Labels mit der angegebenen Schrittweite. In der ersten Zeile wird in Richtung aufsteigender Rektaszension mit der Schrittweite 1^h vervielfältigt. Da es mit dem Punkt ($0^h, 20^\circ$) beginnt, wird also der 20° -Deklinationkreis beschriftet.

Die zweite Zeile macht das Gleiche analog für den 11^h -Rektaszensions-Kreis.

Im Label selber kann man folgende Platzhalter benutzen: `#1` ist die Rektaszension, `#2` die Deklination. `#3` sind die vollen Stunden und `#4` die vollen Minuten der Rektaszension. Damit lassen sich Dinge wie $11^h 20'$ realisieren. `#5` ist die auf eine ganze Zahl gerundete Deklination mit Vorzeichen, beispielsweise „+20“.

L^AT_EX-Präambel

Es ist möglich, PP3 eine L^AT_EX-Präambel zu übergeben, die dann für die Erstellung der Karte benutzt wird. So kann man eigene Stile realisieren und insbesondere für typografische Konsistenz zwischen den Karten und dem Dokument, in das die Karten eingefügt werden sollen, sorgen. Die Datei `wiki.pp3` des Löwe-Beispiels enthält beispielsweise

```
filename latex_preamble wiki.tex
```

PP3-L^AT_EX-Dateien benutzen einige Hooks, in die man sich einhängen kann, um globale Veränderungen vorzunehmen. Die Datei `wiki.tex` (die somit auch für Abbildung 3 zum Einsatz kam) sieht beispielsweise so aus:

```
1 \usepackage[T1]{fontenc}
2 \usepackage{eulervm}
3 \renewcommand*{\sfdefault}{phv}
4 \usepackage{relsize}
5 \renewcommand{\Messier}[1]{\footnotesize{\smaller M}\, #1}
```

```

6 \renewcommand{\NGC}[1]{\footnotesize{\smaller NGC}\,#1}
7 \renewcommand{\IC}[1]{\footnotesize{\smaller IC}\,#1}
8 \renewcommand{\FlexLabel}[1]{\bfseries #1}
9 \AtBeginDocument{\sffamily\boldmath}

```

Dabei sind `\Messier`, `\NGC`, `\IC` und `\FlexLabel` in den Zeilen 5 bis 8 die besagten Hooks. Ihnen wird im Parameter `#1` die Katalognummer übergeben. In diesem Beispiel sorgen sie dafür, dass Nebel-Label klein und Katalogkürzel sogar noch etwas kleiner gedruckt werden. Außerdem werden alle Flexes halbfett gedruckt.

Intern

PP3 ist als Programm ein bunter Haufen aus vielen Einzelmodulen, allein schon, weil eine Sternkarte aus vielen Arten von Objekten besteht. Ich möchte deshalb nur ein paar Dinge herausgreifen und etwas näher erläutern.

Datenbanken

PP3 baut auf eigenen, recht kleinen Datenbanken auf. Sie liegen allesamt als Textdateien vor. Es werden vier externe Datenkompilationen benutzt: der BSC (Sterne), der NGC/IC (Nebel), der „Catalogue of Constellation Boundary Data“ (Sternbild-Grenzen) und das Himmels-Panorama (Milchstraße) [9]. Bei den Sternbild-Linien habe ich die traditionellen Figuren, die mir am besten gefielen, benutzt, allerdings nur für die Sternbilder, die schon beim Wikipedia-Projekt (siehe unten) abgearbeitet sind.

Die Originaldaten sind nicht Teil der Distribution. Für alle vier Fälle habe ich jeweils ein kleines Programm geschrieben, das die unter Umständen binären Daten in Textdaten umwandelt, auf die relevanten Felder reduziert und eventuell noch in anderer Weise aufbereitet. Diese „Digesters“ sind im CVS-Baum verfügbar, aber nicht Teil der Distribution.

Kartenwurf

PP3 ist so aufgebaut, dass verschiedene Projektionen unterstützt werden können, aber es ist de facto nur eine einzige realisiert, nämlich die azimutale äquidistante Projektion. Sie ist in der Astronomie sehr beliebt. PP3 dreht das Zentrum jeder Karte in den Zenit dieser Projektion und kann daher immer die bestmögliche Abbildung bieten.

Milchstraße

Die Milchstraße wird gezeichnet, indem eine Panorama-Bitmap [9] in Himmelskoordinaten-Grauwert-Paare umgewandelt wurde und dann PostScript-Kreise gezeichnet werden, die gerade so klein sind, dass garantiert keine Lücken entstehen. Das führt bei zu starker Vergrößerung natürlich zu Pixel-Effekten, aber für typische Ansichten ist das Ergebnis bei richtiger Farbwahl ausgesprochen hübsch, zumal wenn man bedenkt, dass es sich immer noch um Vektordaten handelt. Als Seiteneffekt entstehen kleine Höfe um die sehr hellen Sterne, aber das ist eher ein nettes Feature als ein Bug.

Für sehr starke Vergrößerungen kann man natürlich noch hochauflösendere Milchstraßenbitmaps für PP3 aufbereiten. Allerdings lassen sie sich für größere Himmelsausschnitte nicht mehr verwenden, weil dann der Speicher von T_EX überläuft. Schon jetzt ist es ratsam, für die Milchstraße den Hauptspeicher von T_EX auf den implementationsspezifischen Maximalwert zu setzen.

Label-Abmessungen

Damit PP3 Überlappungen von Labels erkennen kann, muss es wissen, wie groß sie sind. Dafür wird eine Datei namens `labeldimens.dat` gelesen, die alle Label-Abmessungen enthält. Da diese Datei zuerst nicht existiert und der Benutzer ja beliebige Labels hinzufügen kann, erzeugt PP3 für alle unbekanntes Labels eine temporäre L^AT_EX-Datei, die durch L^AT_EX geschickt wird und als Ergebnis statt einer DVI-Datei alle Label-Abmessungen an PP3 zurückgibt. Diese werden dann für zukünftige Durchläufe in `labeldimens.dat` gespeichert.

Die aktuell eingestellte L^AT_EX-Präambel kommt natürlich auch bei der Berechnung der Label-Abmessungen zum Einsatz.

Wenn man die globale Font-Konfiguration in irgendeiner Weise verändert, muss man die Datei `labeldimens.dat` im aktuellen Verzeichnis löschen, um eine Neuberechnung zu erzwingen.

Originaldokumentation von PP3

Es würde den Rahmen dieses Artikels sprengen, alle möglichen Kommandos, die in einer Skriptdatei benutzt werden können, aufzulisten. Es existieren

zahlreiche Einstellmöglichkeiten für die Karte selber, die darzustellenden Objekte, die Labels, die Strafpunkte-Berechnung, die Farben, die verschiedenen Hilfslinien und noch mehr.²

Der C++-Quellcode von PP3 ist in CWEB [7] geschrieben und daher sehr gut dokumentiert. Eine umfassende Beschreibung des Formates der Eingabeskripte findet sich auf den ersten Seiten des geT_EXten Quellcodes, der als PDF auch von PP3s Homepage heruntergeladen werden kann. Leider existiert keine eigenständige Bedienungsanleitung oder ein Tutorial, sondern nur eine fast vollständige Referenzkarte.

Wikipedia-Beispielskripte

Quasi als Entschädigung dafür sind viele Beispiele für Eingabeskripte Teil der Distribution von PP3. Es handelt sich dabei um die Originalskripte, die bei der Erstellung der Sternbild-Karten des Wikipedia-Projektes [1] zum Einsatz kamen. PP3 zeichnet für alle Sternkarten dieser Internet-Enzyklopädie verantwortlich und hat sich dabei bewährt, auch was den erforderlichen Arbeitseinsatz angeht.

Ausblick

Folgende Erweiterungen von PP3 könnten sich als nützlich erweisen:

- Ein Ephemeridenmodul, das die Darstellung von Sonne, Mond, Planeten etc. in Abhängigkeit vom Datum zulässt. (Es wäre interessant, die Mondphasen mit `pstricks` nachzuempfinden.)
- Mehr, insbesondere echte winkeltreue Kartenprojektionen.
- Die Ekliptik und die Gitterlinien in das Strafpunkte-Schema aufnehmen; sehr kleine Sterne, die sich mit benutzerdefinierten Labels überlappen, automatisch entfernen.
- Intern PP3 auch für sehr große Datenbanken fit machen; die Auflösung der Milchstraße dynamisch so wählen, dass trotz extrem genauer Milchstraßendaten T_EXs Speicher nicht überläuft.
- Klären, ob `pdftricks` [5]+pdfT_EX besser geeignet sind, um PDFs zu erzeugen.

²Insgesamt 14 Top-Level-Schlüsselworte kombinierbar mit 48 weiteren Schlüsselworten.

- Mit `preview.sty` [6] kleine PBMs der Labels machen, um die Unterschneidungs-Parameter auszumessen. Ein „ μ “ beispielsweise sollte, wenn es oben links platziert wird, ein wenig näher an den Stern gerückt werden.

Zusammenfassung

Das Programm PP3 ist gut geeignet, um Sternkarten für populärwissenschaftliche Zwecke zu erzeugen. Auch das Erstellen einer ganzen Serie von Sternkarten ist relativ bequem möglich. Die meisten Labels werden automatisch erzeugt und positioniert. In manuellen Labels stehen alle in einer horizontalen Box erlaubten L^AT_EX-Befehle zur Verfügung. Das Layout lässt sich in vielfältiger Weise beeinflussen, und insbesondere sind nachträgliche globale Stiländerungen kein Problem.

Schwächen sind die spartanische Dokumentation, das wenig intuitive und etwas zerbrechliche Eingabeformat und der unter Umständen recht große Ressourcen-Hunger.

Die komplette PP3-Distribution steht unter einer Freie-Software-Lizenz.

Literatur

- [1] *Wikipedia – the free encyclopedia*; <http://www.wikipedia.org>.
- [2] *FAQ von de.sci.astronomie: Astro-Software*; Jan. 2003; http://dsa-faq.berlios.de/dsa-faq-html/dsa-faq_6.html.
- [3] Torsten Bronger: *Celestial charts with PP3*; Febr. 2003; <http://pp3.sourceforge.net>.
- [4] Patrick Chevalley: *Cartes du Ciel (Sky Charts) – free astronomy software*; Jan. 2003; <http://www.stargazing.net/astropc/>.
- [5] C.V. Radhakrishnan, C.V. Rajagopal und Antoine Chambert-Loir: *Trivial Experiments with pstricks manipulation*; Sept. 2001; CTAN: macros/latex/contrib/supported/pdftricks/.
- [6] David Kastrup, Alan Shutko et al.: *The preview-latex package*; Apr. 2003; <http://preview-latex.sourceforge.net>.
- [7] Donald E. Knuth und Silvio Levy: *The CWEB System of Structured Documentation*; Jan. 2001.

- [8] Chris Laurel: *Celestia: A real-time visual space simulation*; 2002; <http://www.shatters.net/celestia/>.
- [9] Axel Mellinger: *All-Sky Milky Way Panorama*; Apr. 2003; <http://home.arcor-online.de/axel.mellinger/>.
- [10] Timothy Van Zandt: *PSTricks – PostScript macros for generic T_EX*; 1993; <http://www.tug.org/application/PSTricks>.

3D-Ansichten mit `pst-vue3d`

Herbert Voß

Die 3D-Ansicht eines grafischen Objekts oder einer Landschaft gehört zu den interessantesten Dingen der Computergrafik und hat daher auch vielfältige Anwendungsprogramme zutage gefördert, die nicht zuletzt in den 3D-Spielen eingesetzt werden. Für Veröffentlichungen im Bereich der Printmedien ist dagegen keine Animation notwendig, sodass häufig einfache 3D-Ansichten ausreichen. Diese sind dennoch grafisch relativ anspruchsvoll und nicht immer einfach zu erstellen.

Einleitung

Es existieren mehrere `PSTricks`-basierte Pakete (siehe Tabelle 1), die 3D-Darstellungen von mehr oder weniger komplizierten Körpern und mathematischen Funktionen ermöglichen und auf der letzten T_EXLive oder eben auf den CTAN-Servern vorhanden sind. Das erste 3D-Paket wurde vom `PSTricks`-Autor entwickelt und enthält die wesentlichen Algorithmen für die Umsetzung der Makros in PostScript-relevante Befehlsfolgen.[4] Die Festlegung des Betrachtungspunktes erfolgt durch die Option `viewpoint=x y z`, in welcher die drei Koordinaten (x, y, z) den Blickrichtungsvektor entsprechend Abbildung 1 festlegen. Diese Abbildung veranschaulicht den Unterschied zwischen einem internen (relativen) und einen externen (absoluten) Beobachter. Der abgebildete Kopf als interner Beobachter ist an das Bezugssystem gebunden, während wir als außenstehende Leser dieses Artikels externe Beobachter

Tabelle 1: Zusammenstellung verfügbarer 3D-Pakete

Paketname	Funktion
<code>pst-3d</code>	Grundlegende 3D-Makros
<code>pst-gr3d</code>	Einfache Makros zum Zeichnen von 3D-Gittern [1]
<code>pst-3dplot</code>	Parallele Projektion von 3D-Objekten als Gitternetz und Plotten von mathematischen Funktionen [3] sowie Datensätzen
<code>pst-vue3d</code>	Perspektivische Darstellung von 3D-Objekten

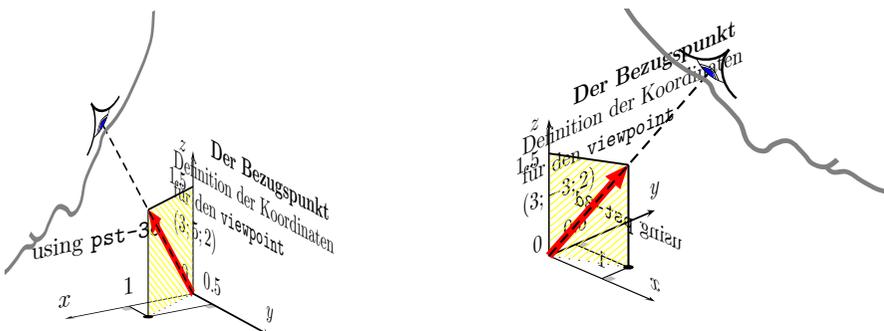


Abbildung 1: Festlegung des so genannten **viewpoint** des externen Beobachters, der eine absolute Position im Raum einnimmt.

sind, die in Form einer zentrischen Streckung, vom **viewpoint** ausgehend, das „Geschehen“ beobachten.

Das Paket `pst-vue3d` [2] erweitert das Paket `pst-3d` durch unterschiedlichste Betrachtungsmöglichkeiten von beliebigen 3D-Objekten, was im Folgenden an verschiedenen Beispielen gezeigt wird.

Die ersten Versuche zur Darstellung von dreidimensionalen Abbildungen wurden durch [4] vorgenommen. Dieses Paket ist zwar ebenso wie [3] auf die Parallelprojektion beschränkt, eröffnet dennoch die Möglichkeit, einfache grafische Objekte beliebig zu platzieren. `pst-vue3d` basiert auf der französischen Sprache, sodass die meisten Namen der Makros oder Optionen ungewöhnlich erscheinen, was aber der Anwendung nicht entgegensteht.

Tabelle 2: Zusammenstellung der verfügbaren Optionen

<i>Name</i>	<i>Wert</i>	<i>Bedeutung</i>
THETA	30	Drehwinkel des Koordinatensystems
PHI	10	Kippwinkel des Koordinatensystems
Dobs	100	Zentrischer Bildabstand
Decran	10	Paralleler Bildabstand (Vergrößerungsfaktor)
normaleLongitude	45	Winkel des vertikalen Normalenvektors einer Ebene
normaleLatitude	45	Winkel des horizontalen Normalenvektors einer Ebene
PhiCercle	0	Breitengrad
ThetaMeridien	0	Längengrad
grille	10	Gitterabstand der Schraffur
scale	1	Skalierungsfaktor
RotX	0	Rotationswinkel um die x-Achse
RotY	0	Rotationswinkel um die y-Achse
RotZ	0	Rotationswinkel um die z-Achse
A	10	Halbe Seitenlänge eines Quaders
B	10	Halbe Seitenlänge eines Quaders
C	10	Halbe Seitenlänge eines Quaders
Rtetraedre	5	Radius des Innenkreises einer Tetraederseitenfläche
ColorFaceA	magenta	Seitenfarbe des Quaders
ColorFaceB	red	Seitenfarbe des Quaders
ColorFaceC	blue	Seitenfarbe des Quaders
ColorFaceD	cyan	Seitenfarbe des Quaders
ColorFaceE	yellow	Seitenfarbe des Quaders
fracHeight	1	Höhenverhältnis eines Kegel- oder Pyramidenstumpfes
DeltaTHETA	10	Relativer Drehwinkel für den Ausschnitt einer Kugeloberfläche
DeltaPHI	10	Relativer Kippwinkel für den Ausschnitt einer Kugeloberfläche
PortionSphereTHETA	0	Absoluter Drehwinkel für den Ausschnitt einer Kugeloberfläche

<i>Name</i>	<i>Wert</i>	<i>Bedeutung</i>
PortionSpherePHI	0	Absoluter Drehwinkel für den Ausschnitt einer Kugeloberfläche
CubeColorFaceOne	1 1 0	Seitenflächenfarbe des Würfels (Angabe der Farbe im RGB-Farbmodell)
CubeColorFaceTwo	0.9 0.9 0	
CubeColorFaceThree	0.8 0.8 0	
CubeColorFaceFour	0.7 0.7 0	
CubeColorFaceFive	0.6 0.6 0	
CubeColorFaceSix	0.5 0.5 0	
CubeInside	false	Offene oder geschlossene Box
SphericalCoor	false	Koordinatentripel (x, y, z) wird sphärisch interpretiert (R, θ, ϕ)

Grafische Objekte

Die Position eines externen Beobachters kann sowohl durch kartesische Koordinaten als auch Kugelkoordinaten vorgegeben werden. Letztere sind die Winkel für den Längengrad θ und den Breitengrad ϕ , sowie den Radius, der den Abstand vom Koordinatenursprung markiert.

Im Folgenden sollen die verwendeten grafischen Objekte kurz zusammengestellt werden. Für jedes der elementaren Objekte ist der PSTricks-Befehl mit angegeben. Dieser ist Teil der folgenden Umgebung. [...] ist durch das in Tabelle 2 angegebene Makro mit entsprechenden Optionen und Parametern zu ersetzen.

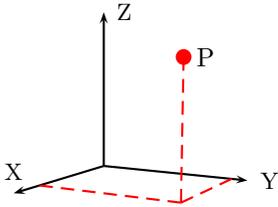
```

1 \begin{pspicture}(-1.5,-0.75)(2.5,2.25)
2   \AxesThreeD(20)
3   [ ... ]
4 \end{pspicture}

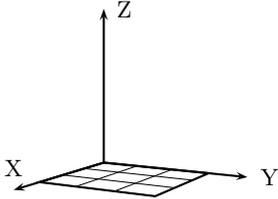
```

Das jeweils zusätzlich mit eingezeichnete Koordinatensystem dient lediglich der besseren räumlichen Zuordnung.

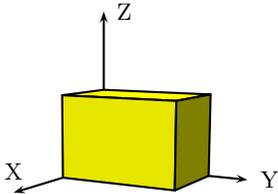
Tabelle 3: Zusammenstellung der 3D-Grafikobjekte



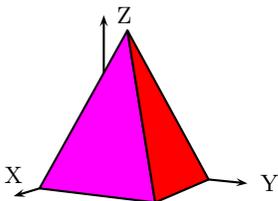
`\pNodeThreeD`



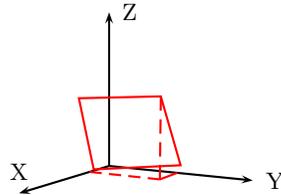
`\QuadrillageThreeD`



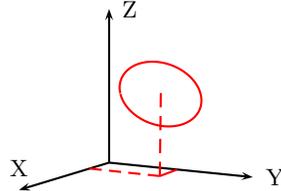
`\CubeThreeD`



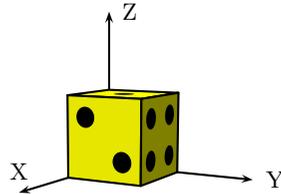
`\PyramideThreeD`



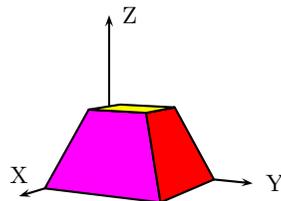
`\FrameThreeD`



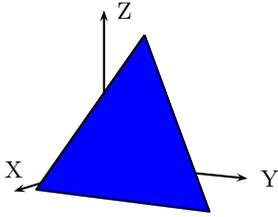
`\CircleThreeD`



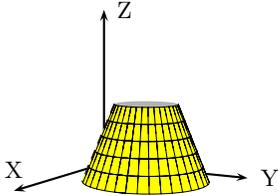
`\DieThreeD`



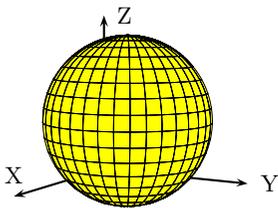
`\PyramideThreeD[fracHeight=0.5]`



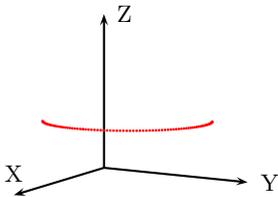
`\TetraedreThreeD`



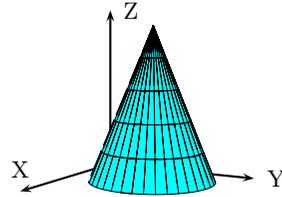
`\ConeThreeD[fracHeight=0.5]`



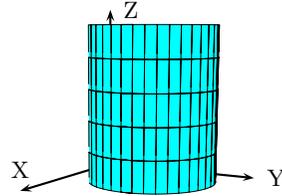
`\SphereThreeD`



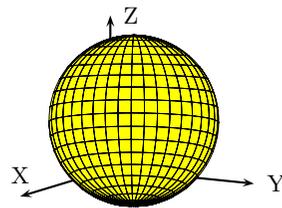
`\SphereCercleThreeD`



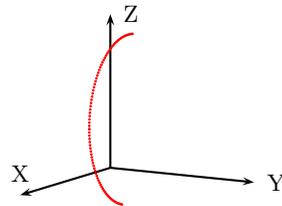
`\ConeThreeD`



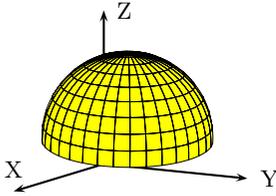
`\CylindreThreeD`



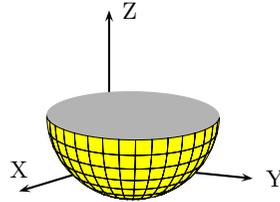
`\SphereInverseThreeD`



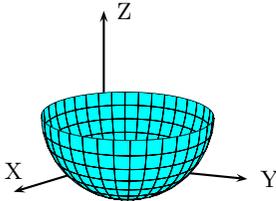
`\SphereMeridienThreeD`



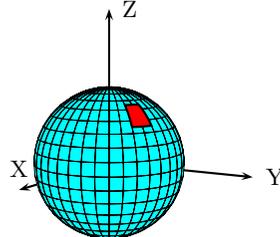
`\DemiSphereThreeD`



`\DemiSphereThreeD[RotX=180]`



`\SphereCreuseThreeD`



`\PortionSphereThreeD`

Knoten-Punkt

Ein Knoten (Punkt) ist gegeben durch

- seine kartesischen Koordinaten (x, y, z) ,
- oder seine Kugelkoordinaten $(r; \theta; \phi)$, wobei θ für den Längengrad und ϕ für den Breitengrad steht.

<ol style="list-style-type: none"> 1 <code>\pNodeThreeD[<Optionen>](<x,y,z>){<Knotenname>}</code> 2 <code>\pNodeThreeD[<Optionen>](Radius;Theta;Phi){<Knotenname>}</code>

Die Angaben werden also durch die Art des Trennzeichens unterschieden. Diese sind analog zu PSTricks, ein Komma markiert kartesische und ein Semikolon sphärische Koordinaten.

Sämtliche Punkte werden als Knoten (node) definiert, sodass der Knotenname hier zwingend ist. Auf diesen kann dann auch mit zweidimensionalen Objekten Bezug genommen werden. Mit dem Makro `showCoor[<Optionen>](<x,y,z>)` beziehungsweise `showCoor[<Optionen>](<x;y;z>)` können die kartesischen Koordinaten durch gestrichelte Linien angezeigt werden, dies allerdings nur im kartesischen Koordinatensystem.

Linie

Eine Linie verbindet zwei dreidimensionale Punkte:

```
1 \LineThreeD[<Optionen>] (<x1,y1,z1>) (<x2,y2,z2>)
```

Rechteck

Ein Rechteck mit dem Spezialfall Quadrat ist durch zwei diagonal gegenüberliegende Punkte festgelegt, die beide in der xy -Ebene liegen.

```
1 \FrameThreeD[<Optionen>] (x,y,z) (<x1,y1>) (<x2,y2>)
```

Die räumliche Lage des Rechtecks kann durch die Optionen `normaleLongitude` und `normaleLatitude` beeinflusst werden (siehe Abbildung 2), wobei sich räumliche Situationen konstruieren lassen. Diese beiden Parameter sind auf 45° voreingestellt, sodass sich ohne weitere Änderung ein schräg im Raum liegendes Rechteck ergibt. Man kann sich die Bedeutung der Parameter so vorstellen, dass man ein Rechteck beliebig auf einer Kugel derart verschieben kann, dass der Bezugspunkt des Rechtecks die Kugel im Punkt $(\text{Radius}, \text{normaleLatitude}, \text{normaleLongitude})$ berührt.

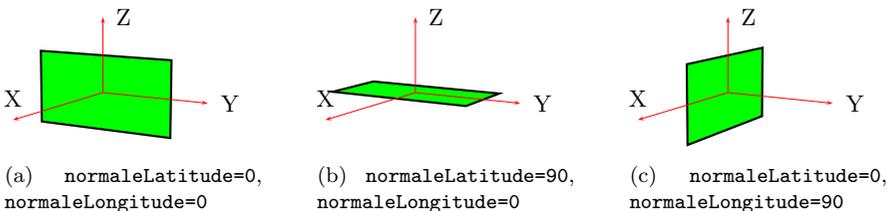


Abbildung 2: Bedeutung der Optionen `normaleLongitude` `normaleLatitude`

Im Beispiel von Tabelle 3 sind drei Rechtecke so angeordnet, dass sie eine Raumecke darstellen. Um diese besser hervorzuheben, kann in die einzelnen Rechtecke mit dem Makro `\QuadrillageThreeD` eine Netzstruktur eingezeichnet werden.

```
1 \QuadrillageThreeD[<Optionen>] (x,y,z) (XMin,YMin) (XMax,YMax)
```

Der Abstand der Linien kann über die Option `grille` beeinflusst werden.

Kreis

Ein Kreis ist durch seinen Radius definiert.

```
1 \CircleThreeD[<Optionen>](<x>,<y>,<z>){<Radius>}
```

Alternativ dazu lassen sich noch die Breitenkreise einer Kugel mit dem Makro `\SphereCercleThreeD` zeichnen. Die Lage eines Kreises im dreidimensionalen Raum wird analog zu einem Rechteck über die gleichen Optionen gesteuert.

Quader

Ein Quader ist ausgehend von seinem räumlichen Mittelpunkt durch seine drei orthogonalen Seitenvektoren gegeben. Deren Beträge entsprechen den halben Seitenlängen und sind sämtlich als Optionen `A,B,C` zu übergeben. Ein Quader, der beispielsweise mit `\CubeThreeD[A=5,B=7.5,C=5](0,0,0)` definiert ist, hat somit die Seitenlängen `a=10`, `b=15` und `c=10` in der jeweils gültigen Längeneinheit und wird mit seinem Mittelpunkt in den Koordinatenursprung gesetzt. Soll der Quader mit den oben angegebenen Seitenlängen mit einem seiner Eckpunkte in den Koordinatenursprung gesetzt werden, so kann dies durch `\CubeThreeD[A=5,B=7.5,C=5](5,7.5,5)` geschehen.

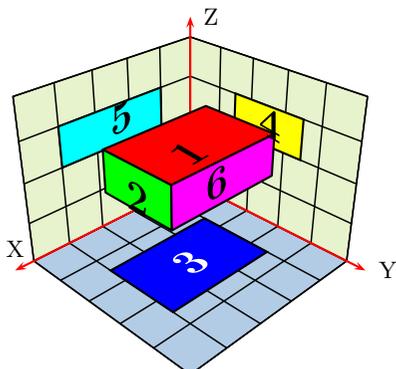
```
1 \cubeThreeD[<Optionen>](<x,y,z>)
```

Im Gegensatz zu den vorhergehenden Grafikobjekten, handelt es sich hier um einen Körper mit unterschiedlichen Seitenflächen, deren Färbung durch entsprechende Parameter geändert werden kann. Abbildung 3 zeigt ein Beispiel, wobei die entsprechenden Farbnummern auf der Option `CubeColorFace###` zugeordnet sind.

Der Würfel ist lediglich ein Spezialfall des Quaders und hat bei *pst-vue3d* insofern eine Bedeutung, als ein eigenes Makro `DieThreeD` für den „echten“ Würfel existiert, der standardmäßig in der hier abgebildeten Form erscheint. Durch entsprechende Rotationsoptionen kann aber auch jede beliebige andere Seite gezeigt werden.

Pyramide

Die senkrechte, viereckige Pyramide oder ein entsprechender Pyramidenstumpf sind definiert durch ihre Grundfläche und ihre Höhe. Erstere ist wieder



```

1 \psset{%
2   CubeColorFaceOne=1 0 0, %
3   CubeColorFaceTwo=0 1 0, %
4   CubeColorFaceThree=0 0 1,%
5   CubeColorFaceFour=1 1 0, %
6     yellow
7   CubeColorFaceFive=0 1 1, %
8     cyan
9   CubeColorFaceSix=1 0 1 %
10    pink
11 }

```

Abbildung 3: Quader mit seinen Parallelprojektionen und den zugeordneten Farben

ausgehend vom Mittelpunkt des Rechtecks durch die zwei halben Seitenlängen A, B gegeben, die anderen Punkte werden intern durch die zwei gegebenen berechnen berechnet.

```

1 \PyramideThreeD[<Optionen>](<x,y,z>){<Höhe>}

```

Die Farbe der Seiten kann wie beim Würfel individuell festgelegt werden. Zuständig sind die Optionen `ColorFaceA`, `ColorFaceB`, `ColorFaceC`, `ColorFaceD`, `ColorFaceE`. Letztere ist die Farbe für die Grund- und Deckfläche.

Der Kegelstumpf wird einfach über eine Option ($0 < \text{fracHeight} \leq 1$) realisiert.

Tetraeder

Der Tetraeder ist durch den Radius des Innenkreises der Seitenflächen gegeben, wobei der Mittelpunkt der unteren Fläche den räumlichen Bezugspunkt darstellt. Der Radius wird über die Option `Rtetraedre` festgelegt und ist mit dem Wert 5 vordefiniert.

```

1 \TetraedreThreeD[<Optionen>](<x,y,z>)

```

Im Gegensatz zur Pyramide können die einzelnen Seitenflächen nicht getrennten Farben zugeordnet werden. Ohne weitere Angaben zur Rotation ist nur eine Seitenfläche des Tetraeders sichtbar.

Kegel

Der senkrechte Kegel beziehungsweise Kegelstumpf sind gegeben durch den Radius der Grundfläche und die Höhe des Gesamtkegels. Der Mittelpunkt der Grundfläche ist wieder der räumliche Bezugspunkt des Kegels. Standardmäßig wird ein Gitternetz zur besseren optischen Darstellung gezeichnet. Dies kann „unterdrückt“ werden, wenn die Linienfarbe gleich der Füllfarbe gesetzt wird.

```
1 \ConeThreeD[<Optionen>](<x,y,z>){<Radius>}{<Höhe>}
```

Über eine Option ($0 < \text{fracHeight} \leq 1$) kann ein Kegelstumpf realisiert werden.

Zylinder

Der senkrechte Zylinder ist gegeben durch den Radius der Grundfläche und die Höhe. Der Mittelpunkt der Grundfläche ist wieder der räumliche Bezugspunkt des Kegels. Standardmäßig wird wie beim Kegel ein Gitternetz zur besseren optischen Darstellung gezeichnet.

```
1 \CylindreThreeD[<Optionen>](<x,y,z>){<Radius>}{<Höhe>}
```

Kugel

Die Kugel ist gegeben durch ihren Radius, der standardmäßig der räumliche Bezugspunkt der Kugel ist.

```
1 \SphereThreeD[<Optionen>](<x,y,z>){<Radius>}
```

Wie beim Kegel wird zur besseren optischen Darstellung ein Gitternetz gezeichnet und zusätzlich leicht nach vorne gekippt. Für die Kugel gibt es eine Vielzahl an weiteren Makros:

- o `\SphereInverseThreeD`: Kugel wird leicht nach hinten gekippt.

- `\SphereCercleThreeD`: Einzeichnen eines Breitengrades.
- `\SphereMeridienThreeD`: Einzeichnen eines Längengrades.
- `\DemiSphereThreeDThreeD`: obere Halbkugel kann über Rotationsoptionen in jede beliebige andere Richtung gebracht werden.
- `\SphereCreuseThreeD`: zeichnet eine Halbkugel mit einem innenliegenden Gitternetz, welche ebenfalls über die Rotationsoptionen in andere Richtungen gebracht werden kann.
- `\PortionSphereThreeD`: zeichnet nur einen Teil der Kugel, der durch die Parameter `DeltaTHETA`, `DeltaPHI`, `PortionSphereTHETA` und `PortionSpherePHI` festgelegt werden kann.

Anordnung des Koordinatensystems

Hier sind im Wesentlichen die beiden Optionen `Dobs` und `Decrans` zu beachten, welche zusammen mit den beiden Winkeln `THETA` und `PHI` und den Rotationsangaben `RotX`, `RotY` und `RotZ` eine beliebige Sicht auf die grafischen Objekte ermöglichen.

Der Programm-Code der Abbildung 4(d) ist relativ einfach, was hier nicht über die Komplexität der einzelnen Makros hinwegtäuschen darf.

```

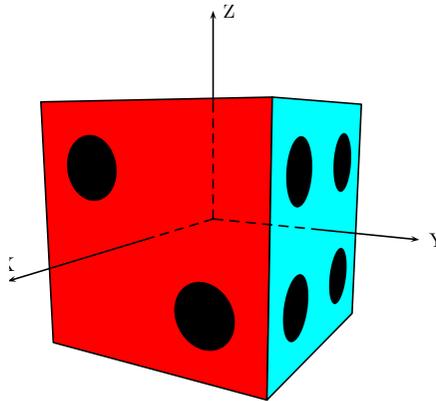
1 \begin{pspicture}(-3.6,-3.25)(4,3.75)
2   \DieThreeD[A=20,B=20,C=20](0,0,0)%
3   \AxesDie
4 \end{pspicture}

```

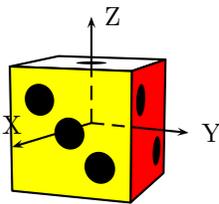
Die Standardwerte für die Abbildungsfaktoren sind `Dobs=100` und `Decrans=10`. `Decrans` ist letztlich nichts weiter als eine Vergrößerung des Originalobjekts, wobei zugehörige Strecken parallel zueinander bleiben. Mit dem Faktor `Dobs` wird die perspektivische Darstellung beeinflusst, was bedeutet, dass zugehörige Strecken nicht parallel zueinander bleiben. Dies lässt sich sehr gut an den Abbildungen erkennen.

Lage eines Würfels im Raum

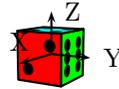
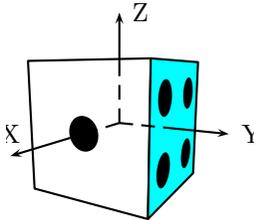
Ein Würfel kann grundsätzlich frei im dreidimensionalen Raum platziert werden, wobei die Kantenlänge eines Würfels hier 10 Längeneinheiten betragen soll. Der relative Mittelpunkt eines Würfels `\Die` hat dann die Koordinaten



(a) Standardansicht



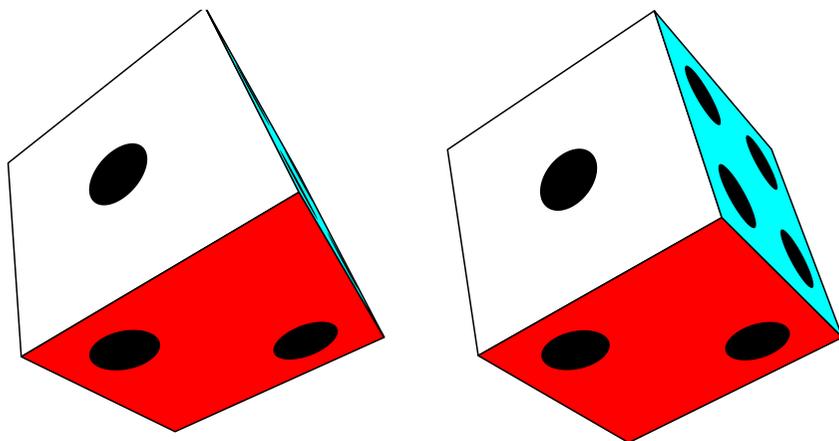
(b) Dobs=250, RotZ=10(c) Decran=4, RotY=90



(d) Dobs=250, Decran=4, RotX=90

Abbildung 4: 3D-Koordinaten und die Bedeutung der Parameter, die die Sicht beeinflussen.

$M(5|5|5)$. Zum besseren Verständnis ist ein zusätzliches Gitter mit den diagonalen Punkten $P_1(0|0|0)$ und $P_2(40|40|40)$ in das 3D-Koordinatensystem eingezeichnet (siehe Abbildung 6(a)). Diese Abbildung ergibt sich unter anderem durch die gewählten Parameter $\backslash\text{psset}\{\text{THETA}=30, \text{PHI}=30, \text{Dobs}=200, \text{Decran}=12\}$. Um in diese Darstellung hineinzuzoomen, wird wie in Abbildung 6(b) gezeigt, auf die Standardwerte zurückgegangen ($\text{THETA}=30, \text{PHI}=10, \text{Dobs}=100, \text{Decran}=10$).



(a) Dobs=60,Decran=10

(b) Dobs=100,Decran=20

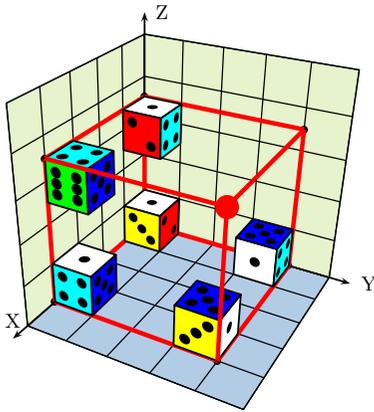
Abbildung 5: Bedeutung der Optionen Decran und Dobs.

Die Würfel aus Abbildung 6 lassen sich ohne Weiteres von hinten betrachten, wenn man den horizontalen Betrachtungswinkel Θ um 180° verändert. Das entsprechende Ergebnis ist in Abbildung 7 zu sehen, aus der ersichtlich ist, dass man von dem im Ursprung liegenden Würfel tatsächlich die Rückseite mit den entsprechenden Augenzahlen sieht.

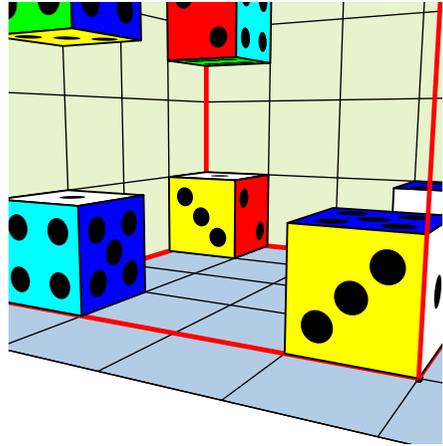
Beispiele

pst-vue3d kann für faktisch alle Bereiche, die dreidimensionale Abbildungen benötigen, eingesetzt werden. Sicher wird man für umfangreiche Abbildungen professionelle Programme einsetzen, doch immer dann, wenn man für Veröffentlichungen oder in der Lehre komplexe und dennoch relativ übersichtliche Abbildungen einsetzen will, lohnt sich die Anwendung von *pst-vue3d*.

Abbildung 8 zeigt, wie man mit relativ wenigen Befehlen Teile einer Kugel in Form einer Explosionszeichnung darstellen kann, während Abbildung 9 als Spielerei abgetan werden kann, wenn auch gerade im Bereich der Statik in Grundlagenvorlesungen derartige Darstellungen benötigt werden.



(a) Würfel in einem Würfel



(b) Änderung der Ansicht

Abbildung 6: 3D-Ansichten

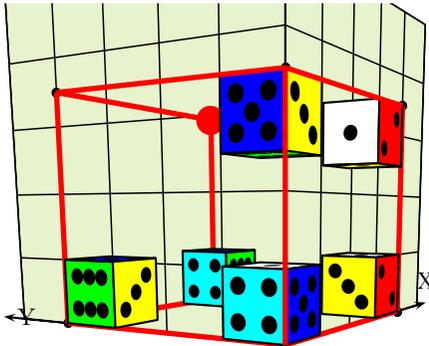


Abbildung 7: 180° Drehung der Abbildung 6

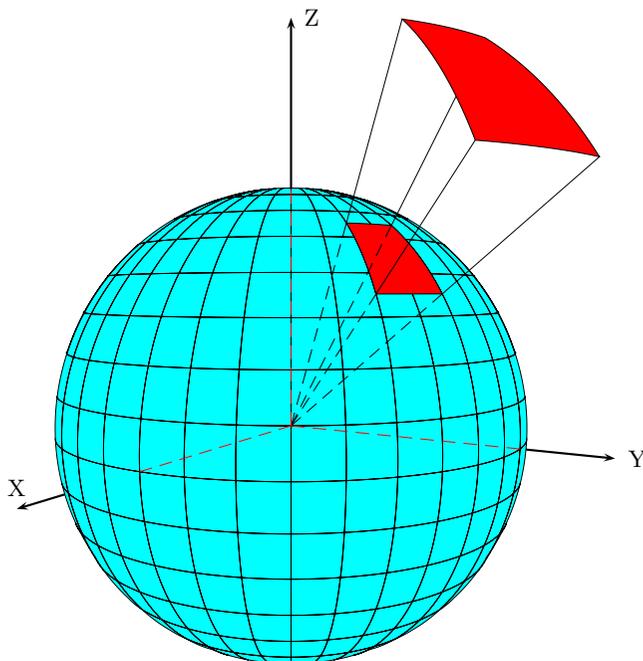


Abbildung 8: Teil einer Kugel hervorheben

```

1 \begin{pspicture}(-3.75,-3.25)(5,5.5)
2   \AxesThreeD(45,40,50)
3   \psset{PortionSphereTHETA=60,PortionSpherePHI=45,linewidth=0.1
4     pt}
5   \SphereThreeD[fillstyle=solid,fillcolor=cyan,linewidth=0.1pt
6     ](0,0,0){30}
7   \PortionSphereThreeD[fillstyle=solid,fillcolor=red](0,0,0){30}
8   \LineThreeD[SphericalCoord,linestyle=dashed](0,0,0)(30,70,35)
9   \LineThreeD[SphericalCoord,linestyle=dashed](0,0,0)(30,50,35)
10  \LineThreeD[SphericalCoord,linestyle=dashed](0,0,0)(30,70,55)
11  \LineThreeD[SphericalCoord,linestyle=dashed](0,0,0)(30,50,55)
12  \LineThreeD[SphericalCoord](30,70,35)(50,70,35)
13  \LineThreeD[SphericalCoord](30,50,35)(50,50,35)
14  \LineThreeD[SphericalCoord](30,70,55)(50,70,55)

```

```

13 \LineThreeD[SphericalCoor](30,50,55)(50,50,55)
14 \PortionSphereThreeD[fillstyle=solid,fillcolor=red](0,0,0){50}
15 \psset{linestyle=dashed,linecolor=red,linewidth=\pslinewidth}
16 \LineThreeD(0,0,0)(30,0,0)
17 \LineThreeD(0,0,0)(0,30,0)
18 \LineThreeD(0,0,0)(0,0,30)
19 \end{pspicture}

```

Sämtliche Objekte können beliebig gedreht und verschoben werden. Wenn ein Objekt von einem anderen teilweise verdeckt wird, dann ist auf die Reihenfolge der Eingabe zu achten; das am weitesten hinten liegende Objekt ist zuerst einzugeben. Verändert man beispielweise die Ansicht der Brücke aus Abbildung 9 zu `\psset{PHI=0,THETA=0,Dobs=15}`, was einer frontalen Sicht entspricht, dann wird hier das Problem der verdeckten Linien offensichtlich. In diesem Fall müsste die Reihenfolge der Makros aus dem folgenden Listing verändert werden.

```

1 \psset{dimen=middle}
2 \newcommand{\cables}{%
3 \multido{\iY=-2+4}{2}{%
4 \parametricplot[linewidth=3\pslinewidth]{-5}{5}{%
5 \variablesTroisD
6 /Xabscisse t def
7 /Yordonnee \iY\space def
8 /Zcote 0.1 Xabscisse dup mul mul def
9 tx@3DDict begin
10 formulesTroisD
11 Xi Yi
12 end
13 }
14 \multido{\nCable=-4.5+0.5}{20}{%
15 \Rmul\cote\nCable\nCable
16 \Rmul\cote\cote{0.1}
17 \LineThreeD(\nCable,\iY,\cote)(\nCable,\iY,0)
18 }%
19 }%
20 \LineThreeD[linewidth=3\pslinewidth](-9,-2,0)(-5,-2,2.5)
21 \LineThreeD[linewidth=3\pslinewidth](9,-2,0)(5,-2,2.5)
22 \LineThreeD[linewidth=3\pslinewidth](-9,2,0)(-5,2,2.5)
23 \LineThreeD[linewidth=3\pslinewidth](9,2,0)(5,2,2.5)
24 }
25 \psset{PHI=30,THETA=45,Dobs=15,Decran=10}

```

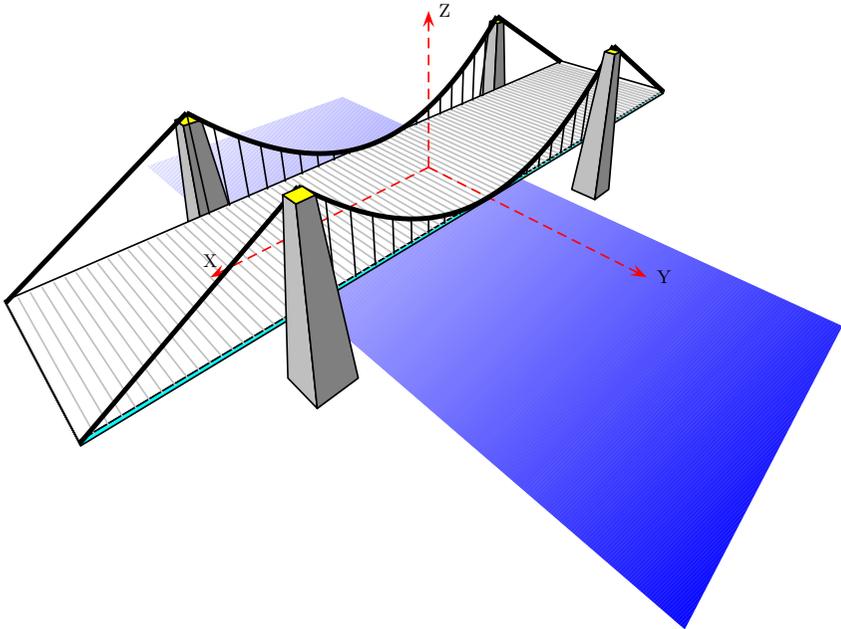


Abbildung 9: Pont du Riviere ...

```

26 \begin{pspicture}(-7.5,-8.5)(7.5,4.5)
27   \FrameThreeD[normaleLongitude=90,normaleLatitude=90,% the
      river
28     linestyle=none,fillstyle=gradient,gradbegin=white,%
29     gradend=blue,gradangle=45](0,0,-2)(-4,-9)(4,9)
30   {\psset{fracHeight=0.62, fillstyle=solid, %
31     ColorFaceA=lightgray,ColorFaceB=gray}
32   \PyramideThreeD[A=.4,B=.4,C=.4](-5,-2,-2){7}% Bridge
33   \PyramideThreeD[A=.5,B=.5,C=.5](5,-2,-2){7}}
34   \CubeThreeD[normaleLongitude=90,% the street
35     normaleLatitude=90,fillstyle=solid,A=9,B=2,C=0.05](0,0,0)
36   \multido{\nL=-8.8+0.2}{89}{%
37     \LineThreeD[linecolor=lightgray](\nL,-2,0)(\nL,2,0)
38   }
39   \AxesThreeD[linestyle=dashed,arrowsize=0.2,linecolor=red](6)
40   \cables

```

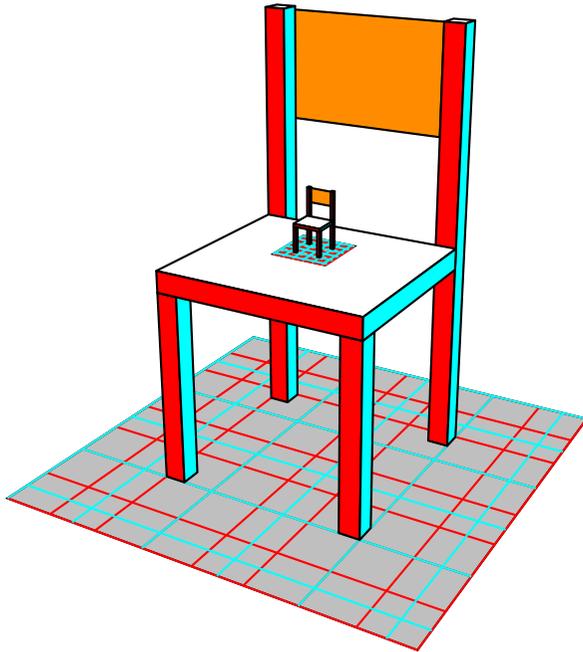


Abbildung 10: Ein Stuhl für einen Stuhl ...

```

41 \psset{fracHeight=0.62, fillstyle=solid, %
42   ColorFaceA=lightgray,ColorFaceB=gray}
43 \PyramideThreeD[A=.4,B=.4,C=.4](-5,2,-2){7}% Bridge
44 \PyramideThreeD[A=.5,B=.5,C=.5](5,2,-2){7}
45 \end{pspicture}

```

Literatur

- [1] Denis Girou: *pst-gr3d*; CTAN: graphics/pstricks/contrib/pst-gr3d.tex; 2001.
- [2] Manuel Luque: *pst-vue3d*; CTAN: graphics/pstricks/contrib/pst-vue3d; 2003.
- [3] Herbert Voß: *pst-3dplot*; CTAN: graphics/pstricks/contrib/pst-3dplot; 2003.
- [4] Timothy Van Zandt: *pst-3d*; ftp://ftp.dante.de/pub/tex/graphics/pstricks/generic/pst-3d.tex; 1999.

Leserbriefe

Zur Rezension „ \LaTeX für Dummies“

Hilmar Preuße

In „Die \TeX nische Komödie 3/2003“ wurde meine Rezension des Buches „ \LaTeX für Dummies“ veröffentlicht. Darin schrieb ich: „Dem Buch liegt eine CD-ROM bei, die eine ‚Kopie‘ einer \TeX -Live zu sein scheint. Leider wusste der DANTE-Vorstand nichts von einer Autorisierung der Kopie.“ Da die \TeX -Live laut Lizenzbestimmungen frei verteilt werden darf, ist das auch so nicht notwendig. Sorry!

Zu kritisieren ist, dass auf der CD selber nicht steht, dass es sich um eine (modifizierte) \TeX -Live handelt. Unrechtmäßig ist auch der Aufdruck des Verlages auf der CD: „© 2002 by mitp-Verlag/Bonn, ein Geschäftsbereich der verlag moderne industrie Buch AG & Co. KG/Landsberg. Alle Rechte vorbehalten. Kein Verleih! Keine unerlaubte Vervielfältigung, Vermietung, Aufführung, Änderung!“

Dieses Statement ist ein Verstoß gegen die Lizenzbestimmungen der \TeX -Live 7. Nur im Buch auf Seite 437 gibt den Hinweis, dass es sich um eine (modifizierte) \TeX -Live 7 handelt. Auch hier kein Hinweis, dass diese frei kopiert werden darf.

Spielplan

Termine

- 30. 8.–3. 9. 2004** TUG 2004
International Conference on T_EX, XML and Digital Ty-
pography
Democritus University of Thrace
Xanthi, Greece
<http://obelix.ee.duth.gr/tug2004/>
Kontakt: Apostolos Syropoulos
- 29.–30. 10. 2004** Herbsttagung und 31. Mitgliederversammlung von
DANTE e.V.
Regionales Rechenzentrum von Niedersachsen (RRZN)
Universität Hannover
- 7.–11. 3. 2005** EuroT_EX 2005 + GUTenberg 2005 + DANTE 2005
(mit 32. Mitgliederversammlung von DANTE e.V.)
„2² Jahre DANTE e.V. + GUTenberg“
Abbaye des Prémontrés
F-54705 Pont-à-Mousson, Frankreich

E t φ θ Δ ε η π ω 8 2 ω L P R 9 B Θ κ Α Ψ Γ Ι Α Σ Δ

l Δ Z C K L O Ψ e 6 O A T L m 7 σ Γ b a λ ε v 3 n Y

Δ N U 4 Ω ω l e 5 o B X h 3 φ x δ ω π Y K Σ 7 L 2

Graphists, programmers and TEXnicians,
all of you with a child's heart,
come and join us at the

**International
Conference on
TEX, XML and
Digital
Typography
TUG2004**

*in Xanthi, Greece,
between August 30
and September 3,
2004!*



*Hosted by the Greek TEX
Friends Group in Xanthi,
Greece, in collaboration
with the Democritus
University of Thrace.*

*For more information,
write to us at:*

tug2004@ocean1.ee.duth.gr

or visit our Web site:

obelix.ee.duth.gr/tug2004

STATHIS

o δ A k W X e z Q 1 n o ε f x m d ⊕ τ ρ ∞ H o

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von T_EX-Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://www.dante.de/events/stammtische/>.

Aachen

Torsten Bronger
bronger@physik.rwth-aachen.de
Gaststätte Knossos
Templergraben 28
Zweiter Donnerstag im Monat, 19.00 Uhr

Berlin

Rolf Niepraschk
Tel.: 030/3481316
rolf.niepraschk@ptb.de
Gasthaus Pali-Eck
Koppenstr. 41
Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

Martin Schröder
Tel.: 0421/2239425
martin@oneiros.de
Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr

Chemnitz

Ralf König
Tel.: 0341/4115800
ralf.koenig@s1998.tu-chemnitz.de
Studenten-Klub „bspw“
Vettersstrasse 70/72, 2.OG
Dritter Mittwoch im Monat, 18.00 Uhr

Darmstadt

Karlheinz Geyer
karlheinz.geyer@LHSystems.com
Restaurant „Bölle“
Darmstadt, Böllenfalltor
Nieder-Ramstädter-Straße 251/Klappacher
Straße
Erster Freitag im Monat, ab 19.30 Uhr

Dresden

Carsten Vogel
lego@wh10.tu-dresden.de
Studentenwohnheim, Borsbergstraße 34,
Dresden, Ortsteil Striesen
ca. alle 8 Wochen, Donnerstag, 19.00 Uhr

Erlangen

Walter Schmidt, Peter Seitz
w.a.schmidt@gmx.net
Gaststätte „Erlanger Gärtla“
Marquardsenstrasse 1
Dritter Dienstag im Monat, 19.00 Uhr

Freiburg

Heiko Oberdiek
Tel.: 0761/43405
oberdiek@uni-freiburg.de
Wechselnder Ort
Dritter Donnerstag im Monat, 19.30 Uhr

Hannover

Mark Heisterkamp
heisterkamp@rrzn.uni-hannover.de
Seminarraum RRZN
Schloßwender Straße 5
Zweiter Mittwoch von geraden Monaten,
18.30 Uhr

Heidelberg

Luzia Dietsche
Tel.: 06221/544527
luzia.dietsche@urz.uni-heidelberg.de
China-Restaurant „Palast“
Lessingstraße 36
Letzter Mittwoch im Monat, 20.00 Uhr

Karlsruhe

Klaus Braune
Tel.: 0721/6084031
braune@rz.uni-karlsruhe.de
Universität Karlsruhe, Rechenzentrum
Zirkel 2, 3. OG, Raum 316
Erster Donnerstag im Monat, 19.30 Uhr

Köln

Bruno Hopp
b.hopp@lepkes-frings.de
Institut für Kristallographie
Zülpicher Straße 49b
Letzter Mittwoch im Monat, 19.30 Uhr

Konstanz

Matthias Weisgerber, Hraban Ramm
 weisgerb@fmi.uni-konstanz.de,
 hraban@fieee.net
Restaurant Rheingold
Spanierstraße 3
unregelmäßig

München

Michael Niedermair
 m.g.n@gmx.de
Wirtshaus „Löwe am Markt“
Dreifaltigkeitsplatz 4
Erster Dienstag im Monat, 19.00 Uhr

Münster

Johannes Reese
 reese@linguist.de
Gaststätte „Sabroso“
Mauritzstraße 19
nach Vereinbarung

Stuttgart

Bernd Raichle
 bernd.raichle@gmx.de
Bar e Ristorante „Valle“
Geschwister-Scholl-Str. 3
Zweiter Dienstag im Monat, 19.30 Uhr

Wuppertal

Andreas Schrell
 Tel.: 02 02/50 63 81
 schrell@wupperonline.de
Restaurant Croatia „Haus Johannisberg“
Südstraße 10
an der Schwimmoper Wuppertal-Elberfeld
Zweiter Donnerstag im Monat, 19.30 Uhr

Zürich

Johannes Reese
 reese@spw.unizh.ch
nach Vereinbarung

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10.00–12.00 Uhr)
Fax: 0 62 21/16 79 06
E-Mail: dante@dante.de

Konten: Volksbank Rhein-Neckar eG
BLZ 670 900 00
Kontonummer 2 310 007
IBAN DE67 6709 0000 0002 3100 07
SWIFT-BIC GENODE61MA2

Postbank Karlsruhe (Auslandsüberweisungen)
BLZ 660 100 75
Kontonummer 213 400 757
IBAN DE93 6601 0075 0213 4007 57
SWIFT-BIC PBNKDEFF

Präsidium

Präsident:	Volker RW Schaa	president@dante.de
Vizepräsident:	Klaus Höppner	vice-president@dante.de
Schatzmeister:	Tobias Sterzl	treasurer@dante.de
Schriftführer:	Günter Partosch	secretary@dante.de
Beisitzer:	Thomas Koch	
	Bernd Raichle	advisor@dante.de

Server

ftp: [ftp.dante.de](ftp://ftp.dante.de) [134.100.9.51]
E-Mail: ftpmail@dante.de
WWW: <http://www.dante.de/>

Autoren/Organisatoren

- Torsten Bronger** [87]
Lochnerstraße 7
52064 Aachen
bronger@physik.rwth-aachen.de
- Mark Heisterkamp** [33]
Regionales Rechenzentrum für Niedersach-
sen (RRZN)
Universität Hannover
Schloßwender Str. 5
30159 Hannover
- Klaus Höppner** [4, 6]
siehe Seite 126
- Informatik-AG Kepler-Seminar** [9]
Stuttgart
www.sbb-stiftung.de/ags/info1.html
- Uwe Kern** [35]
Lessingstraße 14
65189 Wiesbaden
u.kern@web.de
- Gerd Neugebauer** [3]
Im Lerchelsböhl 5
64521 Groß-Gerau
gene@gerd-neugebauer.de
- Hilmar Preuße** [121]
Max-Planck-Str. 2
64807 Dieburg
hille42@web.de
- Volker RW Schaa** [4, 16, 33]
siehe Seite 126
- Apostolos Syropoulos** [122]
366, 28th October Str.
GR-671 00 Xanthi, Hellas
apostolo@ocean1.ee.duth.gr
- Till Tantau** [54]
International Computer Science Institute
1947 Center St, Suite 600
Berkeley, California 94704
USA
- Herbert Voß** [81, 101]
Wasgenstr. 21
14129 Berlin
voss@perce.de

Die T_EXnische Komödie

16. Jahrgang Heft 2/2004 Juni 2004

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Umstellung der Kreditkartenakzeptanz bei DANTE e.V.
- 9 Tagungsbericht Darmstadt 2004: Im Zeichen der 15
- 16 Eine Bildergalerie zum $(2^{2^{2^0}} - 1)$. Geburtstag
- 33 Einladung zur Herbsttagung und 31. Mitgliederversammlung von DANTE e.V.

Bretter, die die Welt bedeuten

- 35 Farbspielereien in L^AT_EX mit dem `xcolor`-Paket
- 54 Strahlende Präsentationen mit L^AT_EX
- 81 Farbige Mathematik
- 87 Zeichnen von Sternkarten mit PP3
- 101 3D-Ansichten mit `pst-vue3d`

Leserbriefe

- 121 Zur Rezension „L^AT_EX für Dummies“

Spielplan

- 122 Termine
- 123 TUG2004
- 124 Stammtische

Adressen

- 127 Autoren/Organisatoren