

prosperを使おう : LATEXでプレゼン資料を作成

池田, 大輔
九州大学情報基盤センター研究部

渡部, 善隆
九州大学情報基盤センター研究部

<https://doi.org/10.15017/1470408>

出版情報 : 九州大学情報基盤センター広報 : 全国共同利用版. 2 (1), pp.41-62, 2002-03. 九州大学情報基盤センター
バージョン :
権利関係 :

prosper を使おう — LaTeX でプレゼン資料を作成 —

池田大輔*

渡部善隆†

1 はじめに

prosper は Frédéric Goualard によって作られた LaTeX のクラスファイルで、プレゼンテーションの資料作成に使います。LaTeX で dvi ファイルを生成した後、トランスペアレンシー用

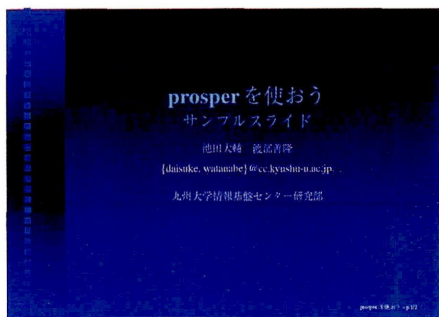


図 1: タイトルページのサンプル

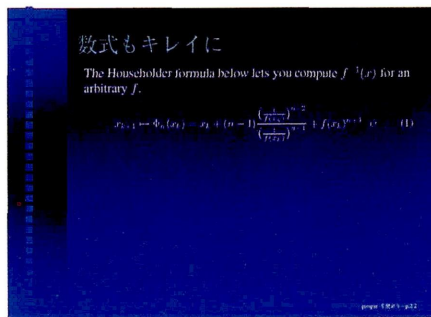


図 2: 本体部分のサンプル

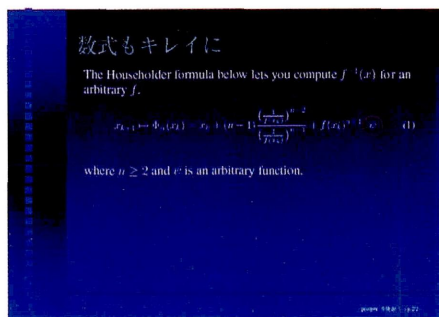


図 3: 説明と赤い円を追加

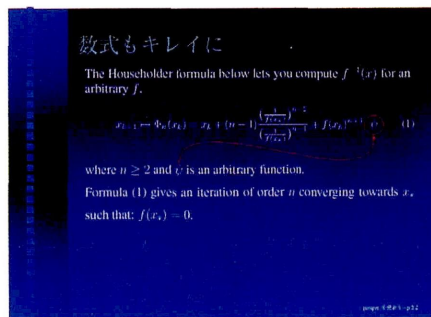


図 4: さらに矢印と説明を追加

*情報基盤センター研究部 <mailto:daisuke@cc.kyushu-u.ac.jp>

†情報基盤センター研究部 <mailto:watanabe@cc.kyushu-u.ac.jp>

の PostScript ファイルやビデオプロジェクトに投影して使用する Portable Document Format (PDF) ファイルに変換して使用します。

図 1 から図 4 は prosper を使って作った PostScript のサンプルです。図 1 はタイトルページです。ビデオプロジェクト上では、図 3 と図 4 は、図 2 上に追加されていくように見えます。

本稿では、図 2 から図 4 をまとめて 1 枚のスライドと呼び、個々の図をページと呼ぶことにします。

prosper の特徴は、 \LaTeX から受けついだものと、PDF というファイル形式から受けついだものがあります。 \LaTeX から継承したものとしては、例えば、 \LaTeX のクラスファイルであるため数式の記述能力に優れている (図 2~4 参照) 上に、 \LaTeX で論文を書く人は論文で使った数式をそのままコピーして使えます。バージョン管理システム CVS や RCS を利用している人にとっては、ソースコードがテキストであるということも大きな利点でしょう。

PSTricks パッケージを使えば簡単な図形 (円、多角形、線、矢印など) も書けます。矢印は \LaTeX の要素 (数式やテキストなど) 間を結ぶことも可能です (図 4 参照)。

一方、PDF ファイルであることの利点として、ビデオプロジェクトに投影する時には PDF ファイルと Acrobat Reader があれば事足ります。したがって、非常に可搬性が高いと言えます¹。また、PDF ファイルにハイパーリンクを埋めこんで、ページ間を移動したり、1 行ずつアニメーション効果をつけて表示したりすることもできます。ハイパーリンクは \LaTeX の中に埋め込んで使います。

一方、欠点としては dvi ファイルから PDF への変換に手間がかかる、最終的な見栄えは変換するまで分からない、といったところでしょうか。また、既に \LaTeX が使える環境にある場合でも、最終的な出力形態である PostScript や PDF に変換するための変換ソフトをインストールしたり、設定しなおしたりする必要があるかもしれません。

本稿では、 \LaTeX や関連するコマンドについての基本的な知識を仮定した上で、prosper の使い方やインストールの方法などを説明します。

1.1 必要な環境

まず、 \LaTeX が使える必要があります。p \LaTeX や j \LaTeX でも構いません。prosper 特有の機能は PostScript や PDF の機能を利用して実現されるので、dvi ファイルの作成まではどのコマンドでも構わないと思います。また、これらの \LaTeX コマンドを実行する OS にも依存しないと思います。

prosper はいくつかのクラスファイルやスタイルファイルを内部で利用します。そのため、あらかじめ seminar パッケージと PSTricks パッケージと hyperref (6.69 より新しいバージョン) が必要です。これらのパッケージは te \LaTeX [4] をインストールすると、すべてインストールされます。te \LaTeX は Thomas Esser がメンテナンスしている \TeX のディストリビューションです。

前述のように、prosper を利用するときは、最終的に PostScript 形式か PDF 形式にしてからプレゼンテーションを行いません。dvi ファイルから PostScript にするために dvips (5.85 以降) が必要です。なお、現在 dvips は単体での配布はされなくなり、te \LaTeX や他の \TeX パッケージなどの一部として配布されるようになりました。

¹Acrobat Reader 5.0 は Windows, Macintosh 上で、4.0 は AIX, OSF/1, HU-UP, IRIX, Solaris, Linux で動作します。

PostScript ファイルから PDF ファイルに変換するためには、Adobe 社の Acrobat Distiller を用いるか、Ghostscript に付属している ps2pdf コマンドを使う必要があります。

ps2pdf を用いる場合は、Ghostscript のバージョン 6.0 以降がお勧めです。これより前のバージョンでは、ビデオプロジェクトに投影する場合、フォントが綺麗に表示できません²。

上記のコマンドやスタイルファイル等は、研究用システムのライブラリサーバ wisdom 上で使うことができます。このうち platex などのコマンドは /usr/local/bin にインストールしてあります。

```
% platex
This is pTeX, Version p3.0.1, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
% dvips
This is dvipsk 5.86 pl.5e Copyright 1996-2001 ASCII Corp.(www-ptex@ascii.co.jp)
% gs -v
Galaddin Ghostscript 5.50 (1998-9-16)
```

また、prosper そのものは /usr/local/share/texmf/tex/misc/prosper にインストールしてあります。通常は、これらのコマンドやクラスファイルは設定の変更なしに利用できると思います。

prosper のインストールの仕方は 5 節を参照してください。

本稿では prosper-1.00.4 を用います。L^AT_EX ソースから PostScript までの変換は FreeBSD 上の以下の環境で動作を確認しました。

```
% platex
This is pTeX, Version p2.1.11, based on TeX, Version 3.14159 (EUC) (Web2C 7.3.1)
% dvips
This is dvipsk 5.86 pl.5e Copyright 1996-2001 ASCII Corp.(www-ptex@ascii.co.jp)
based on dvipsk 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
% gs -v
GNU Ghostscript 5.50 (2000-2-13)
```

ただし、Ghostscript は日本語されたものです。この Ghostscript に付属の ps2pdf は、資料作成の途中の確認のために用いるものです。最終的には Windows 98 上の Acrobat 5.0 を利用して PDF を生成しました。

また、Windows2000 上の pTeX (Version p2.1.11, based on TeX, Version 3.14159 (SJIS) (Web2C 7.3.3)) と Ghostscript 7.03³ でも動作を確認しています。Windows 上で prosper を使うときの覚え書きが文献 [3] にあります。こちらは、今後徐々に内容を充実させていく予定です。

2 基本的な使い方

prosper を使った L^AT_EX ソースファイルは図 5 のようになります⁴。prosper を使うには、

²日本語を使う場合は、このバージョン 6.0 でも綺麗に表示できません。Acrobat を購入しましょう。

³このバージョンに付属の ps2pdf でも、日本語は綺麗に表示できていません。

⁴この図はマニュアル [6] に付属の画像ファイルを編集したものです。

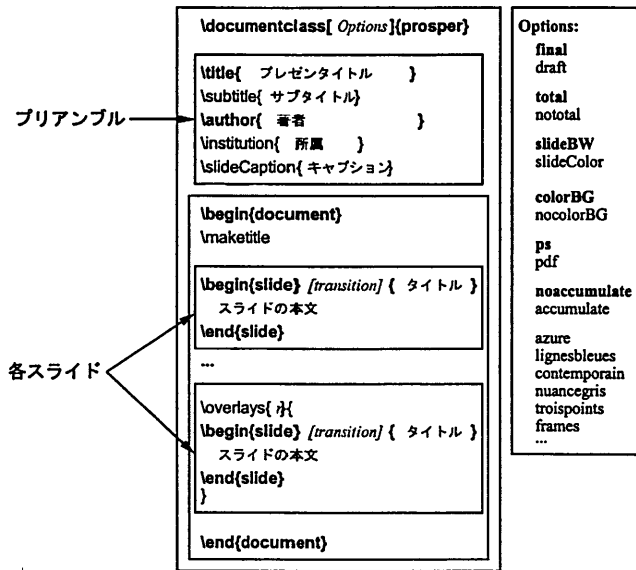


図 5: L^AT_EX ソースファイルの構造。右はオプション群で、空白で区切られたひとかたまりが一つのグループです。グループのうちゴチックで示されているのはデフォルトオプションです

`\documentclass` のクラスファイル名として “prosper” を指定します。

図 5 の右に示したように、prosper には `\documentclass` のオプションとして様々なものが用意されています。

2.1 オプション

ビデオプロジェクタに色を使った PDF ファイルを投影することを前提にすると、最低限必要なオプションは pdf と slideColor と colorBG です。あとは好みに応じてスタイルを指定するとよいでしょう。この節では、これらのオプションについて説明します。

最終的な出力形態を指定するオプションは二つ (ps と pdf) あり、ps がデフォルトです。つまり、オプション pdf を明示的に指定しないとプリントアウトのための PostScript ファイルを出力します。ビデオプロジェクタに投影して使用する場合は pdf を指定してください。

オプションに pdf を指定していて、かつ、PostScript ファイルからの変換プログラムに Acrobat を使う場合は distiller オプションも指定します。distiller オプションを指定せずに Acrobat を利用すると、エラーとなり PDF ファイルが生成されません⁵。

prosper では、背景の模様などをスタイルファイルとして準備してあります。どのようなスタイルがあるかは 3.1 節を参照してください。図 5 の右下 (“azure” で始まるところ) が、一部のスタイル名です。気に入ったスタイルファイルを指定してください。もちろん、スタイルファイルを指定しなくても構いません。

⁵ただ、こちらの Windows2000 上で確認したところでは distiller オプションをつけた場合は pdf オプションを指定するとエラーになる場合がありました。

ただし、選択したスタイルによっては `itemize` 環境や後述する `itemstep` 環境 (3.2 節参照) を使う場合に、prosper の配布物にある `img` ディレクトリ以下の画像ファイルが必要になります⁶。例えば、スタイルを指定しないときに利用される default スタイルを使うと、`itemize` 環境などで使われる行頭文字として `red-bullet-on-white.ps` が必要です。これらの画像ファイルがインストールされてない場合には `azure`, `contemporain`, `frames`, `lignesbleues`, `nuancegris`, `troispoints` のうちどれかを指定してください。

2.2 プリアンプル

タイトルや著者名は、通常通り `\title` と `\author` コマンドを使います。

```
\title{prosper を使おう}
\subtitle{\LaTeX を使ったプレゼンテーション}
\author{池田 大輔}
\email{daisuke@cc.kyushu-u.ac.jp}
\institution{九州大学情報基盤センター}
```

これらに加えサブタイトル (`\subtitle`)、メールアドレス (`\email`)、所属 (`\institution`) が使えます。これらのコマンドは `\maketitle` により有効になり、最初のスライドとして出力されます (図 1 参照)。

複数の著者がいる場合、通常の \LaTeX ソースでは `\author` 中に `\and` を用いますが、prosper では使用できません。

`\slideCaption{caption}` は、各スライドの右下に小さなフォントで “caption” が表示されます。デフォルトでページ番号もキャプションの一部として表示されます。`\ColorFoot{col}` で、スライドキャプションの色を変更できます。

本文は、通常の \LaTeX と同様に `document` 環境内に記述します。

2.3 スライドページ

各スライドを生成するには `slide` 環境を使います。

```
\begin{slide}{prosper の使い方}
  \begin{itemize}
    \item タイトルページ
    \item スライドページ
      \begin{itemize}
        \item 入れ子にできます
        \item インデントされて表示されます
      \end{itemize}
    \item スライドページ
  \end{itemize}
\end{slide}
```

⁶これらの画像ファイルは各 `\item` の行頭文字として使われます。

各スライドは、スライドタイトルと本文からなり、スライドタイトルは `slide` 環境の引数として与えます。このように入力したスライドは、図 6 のように出力されます。この例は背景など

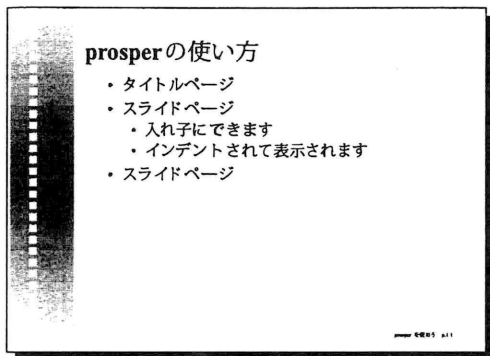


図 6: スライドのサンプル

のスタイルとして `azure` スタイル (3.1 節参照) を利用しています。これを含めて、以下の例では以下のようなオプションを指定します。

```
\documentclass [pdf, distiller, azure] {prosper}
```

これ以降のサンプルはすべて基本的に同じオプションによる出力です。

図 6 ではタイトルが青、各行の行頭文字が緑で表示されます⁷。また、右下には小さく “prosper を使おう - p1./1” と表示されています。これは `\title` に与えた文字とページ数から自動的に生成されます。

`slide` 環境の中では、通常の \LaTeX の環境やコマンドが使えますが、項目を連ねる場合は `itemize` 環境か後述する `itemstep` 環境 (3.2 節参照) を使うとよいでしょう。図 6 では `itemize` 環境を入れ子にして使っています。入れ子にされた `itemize` 環境内の各 `\item` は、インデントが大きくなって表示されます。使用するスタイルによっては、インデントの深さにより行頭文字が変化します。

`prosper` で作成した PDF ファイルを Acrobat Reader で開いたとき、タイトルとして指定した文字列が「しおり」(bookmark) として表示されます。ただし、日本語のタイトルは Acrobat で生成した PDF ファイルでも化けてしまうようです。日本語をタイトルに含める場合の対処方法については文献 [2] を参照してください。

`prosper` では、各スライドページを表示するときの表示方法をいくつか選択できます。選択できる表示方法は以下の通りです⁸。

Replace デフォルトの表示方法で、単に次のページにおきかわる。

Split スライドの中心に向かって、上からと下からと現われる。

⁷印刷の都合で白黒しか分かりませんが。

⁸文章による説明では分かりにくいと思いますので、実際にそれぞれの効果を試してみてください。

Blinds ブラインドのように複数の線を基準に、上下から現われる。

Box 左右から表示されていく。

Wipe 左から表示されていく。

Dissolve 溶けていくように現れる。

Glitter Dissolve と同じようだが、左から現われるような感じ。

これらの指定は `slide` 環境のオプションとして、以下のようにします。

```
\begin{slide}[Glitter]{スライドタイトル}
% ここに本文
\end{slide}
```

デフォルトの表示方法を変更するには `\DefaultTransition{trans}` をプリアンブルで指定します。

2.4 コンパイルおよび実行

`prosper` をクラスファイルに指定した \LaTeX のソースファイルができたら、あとはコンパイルして所望の形式に変換します。最終的には PostScript にしてプリントアウトするか、PDF にしてパソコンからビデオプロジェクタに投影して使用します。

まず \LaTeX のソースファイルを `latex` コマンド⁹で `dvi` 形式に変換します。

```
% platex file.tex # dvi ファイルを作成
% dvips -o file.ps file.dvi # PostScript を作成
```

このようにして作った `dvi` ファイルから `dvi` 形式からは `dvips` などのコマンドで PostScript に変換します。プリントアウトする場合は、この PostScript ファイルをプリンタに出力します。

PDF にする場合は Windows 上の Acrobat か Ghostscript に付属の `ps2pdf` コマンドで変換します。

```
% ps2pdf file.ps
```

他のコマンド、例えば `dvi` ファイルから直接 PDF ファイルを生成する `dvipdfm` は使用できません。

PDF ファイルを使ってプレゼンテーションを行なう場合は Acrobat Reader を全画面モード (Full Screen モード) で使用します。Acrobat Reader でファイルを開いた後、表示メニューから [全画面表示] を選択するか `Ctrl-L` を押して全画面モードに移行します。通常の画面に戻るには再度 `Ctrl-L` を押します。

⁹あるいは `platex` や `jlatex` などのコマンドを使います。

ページを移動するには矢印キーを使います。マウスクリックにより移動を行ないたい場合は Acrobat Reader で [編集] → [環境設定] → [全画面表示] から [クリックで次のページを表示] にチェックを入れます。

3 テクニックを駆使して良いプレゼンを

ここまでの説明で prosper の最低限の機能は使えると思います。しかし、良いプレゼンテーションを行なうためには、見映えよく配置したり、1枚のスライドに次々と内容が現われるようにしたりと、まだまだ工夫が必要です。

3.1 背景

prosper には、標準で 10 種類の背景 (スタイル) を使うことができます。PPRxxx.sty というスタイルファイルがあるとき、\documentclass のオプションに xxx と指定して用います。

```
\documentclass[pdf, distiller, slideColor, colorBG, azure]{prosper}
% azure が背景などを含めたスタイルのオプション
```

標準で添付されているスタイルファイルは図 7 から図 15 のようになります¹⁰。また、配布物の contrib ディレクトリに prosper の作者以外の人が作成し、寄贈してくれたスタイルファイルがあります。

```
% ls contrib
CVS/
PPRAlcatel.sty  PPRpascal.sty  gyom.tex      rico.tex
PPRricostyle.sty PPRricostyle.sty img/
PPRgyom.sty     gyom.ps        rico.ps
```

PPRxxx.sty という形のファイルがスタイルファイルです。PPRgyom.sty と PPRricostyle.sty に関しては、 \LaTeX と PostScript のサンプルもあります。PPRricostyle.sty を使う場合は contrib/img ディレクトリにある画像ファイルも必要となります。

prosper 本体には含まれていませんが、prosper のダウンロードページ¹¹ から PPRblends.sty というスタイルファイルも取得可能です。

3.2 重ね合せとアニメーション

トランスペアレンシーを使ったプレゼンテーションでよく使う技として、複数のトランスペアレンシーを重ねる方法があります。PowerPoint では、アニメーションとして同じような効果をだすことができます。prosper でも同様のことを PDF ファイルにすることで簡単に行なうことができます。

¹⁰スタイルを指定しない場合に使われる default スタイルは載せていません。

¹¹<http://prosper.sourceforge.net/>

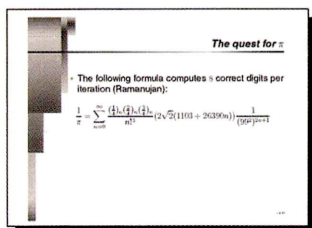


図 7: frames

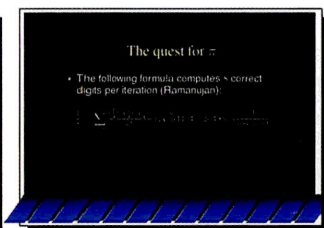


図 8: lignesbleues

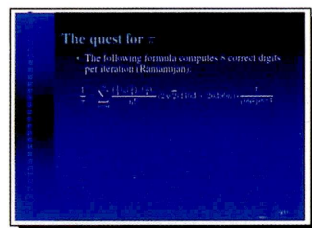


図 9: azure

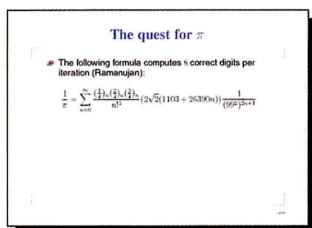


図 10: troispoints

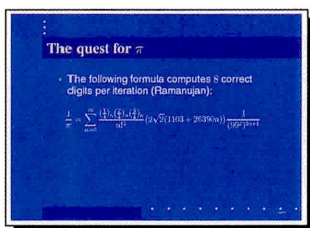


図 11: contemporain

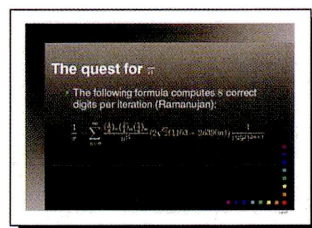


図 12: nuancegris

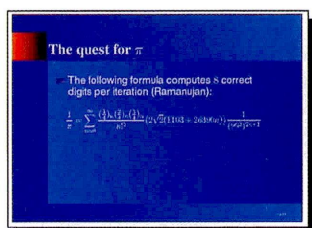


図 13: darkblue

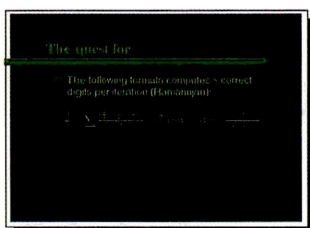


図 14: alienglow

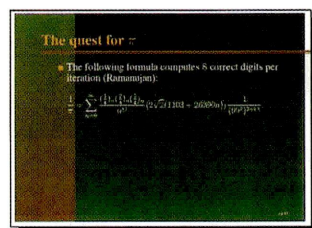


図 15: autumn

1枚のスライド上で1行ずつ表示する最も単純なアニメーションは、`prosper` で定義されている `itemstep` 環境で実現できます。`itemstep` 環境に限らず1ページのスライド上に次々と重ねて表示させたい場合は、そのページの `slide` 環境を、以下のように `\overlays{n}` で囲む必要があります。

```
\overlays{3}{
\begin{slide}{重ねて表示しよう}
  \begin{itemstep}
    \item 1行目
    \item 2行目
    \item 3行目
  \end{itemstep}
\end{slide}
}
```

n は1枚のスライドが何ページからなるかを指定する数です。上の例では3ページからなるので $n=3$ となります。このように入力すると図 16 から図 18 のように表示されます。

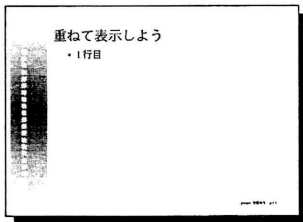


図 16: 最初の表示

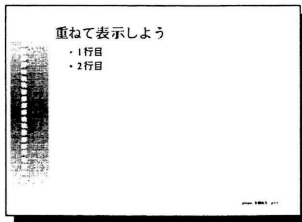


図 17: ↓キーで次の行

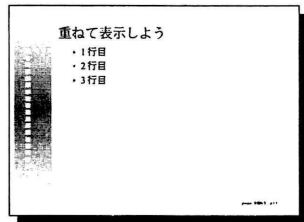


図 18: さらに↓キー

`itemstep` は入れ子にはできないことに注意してください。重ね合せのない単純な場合 (図 6 のような場合) は、`itemize` 環境を入れ子にすることで複数のインデントを持つスライドが生成できます。入れ子にした上で重ね合せをする場合は、以下のようにします。

```
\overlays{4}{
\begin{slide}{prosper の使い方}
  \begin{itemize}
    \fromSlide{1}{
      \item 1枚目
      \item ここも1枚目 } % ここまで1枚目
    \fromSlide{2}{
      \item ここは2枚目\fromSlide{3}{\untilSlide{3}{←ここは3枚目のみ}}
      \item ここは2枚目 } % ここまで2枚目
    \fromSlide{4}{
      \item ここは4枚目 } % ここまで4枚目
    \end{itemize}
  \end{slide}
}
```

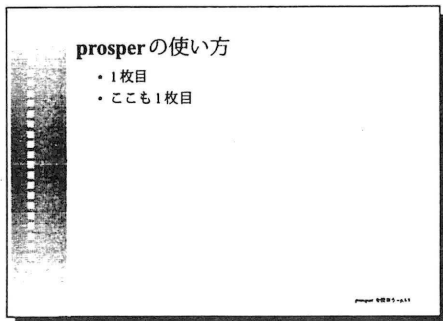


図 19: 最初の表示は 2 行

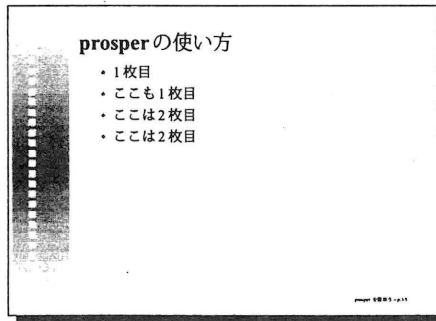


図 20: ↓キーで次の 2 行

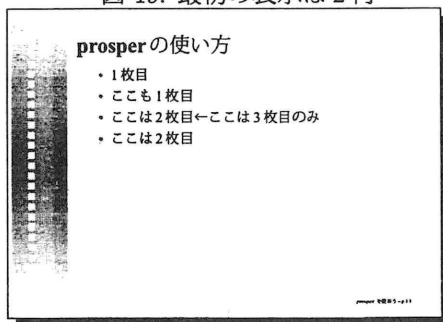


図 21: ↓キーで右側に追加

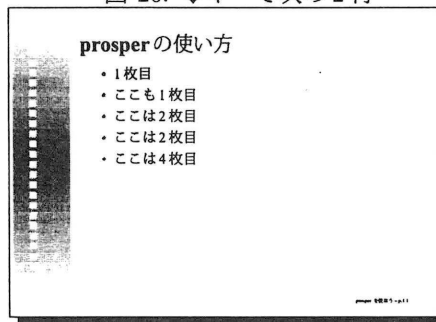


図 22: 図 21 で追加した分を消して、下に新たに追加

`\overlays{n}` としての `slide` 環境の中で `\fromSlide{p}{stuff}`, `\onlySlide{p}{stuff}`, `\untilSlide{p}{stuff}` を使えば, `stuff` を特定のページ上で表示させることが可能です。`\fromSlide{p}` は p 枚目から n 枚目まで, `\onlySlide{p}` は p 枚目のみ, `\untilSlide{p}` は 1 枚目から p 枚目まで表示します。この例では 4 ページで 1 つのスライドを構成しています (図 19 から図 22 参照)。単に `\item` ごとの表示ではないので, `itemstep` 環境ではなく `itemize` 環境を使っています。`\fromSlide{2}` の中で使っているように, これらのコマンドは入れ子にできます。

`\fromSlide{p}{stuff}`, `\onlySlide{p}{stuff}`, `\untilSlide{p}{stuff}` の各コマンド名に “*” をつけると, 幅 0 のボックスで `stuff` を出力します。これを使ってマニュアル [6] にあるように

```
\onlySlide*{1}{\includegraphics{example-1.eps}}%
\onlySlide*{2}{\includegraphics{example-2.eps}}%
\onlySlide*{3}{\includegraphics{example-3.eps}}%
```

と入力すると, 3 つの EPS ファイルが順に同じ位置に表示されます。EPS ファイル以外でも, 4 節で説明する図形が上の方法で同じ位置に出力できます。

この方法を応用して図 23 から図 25 のような表示をさせてみましょう。これは

```
\overlays{3}{%
\begin{slide}{重ねて表示しよう(2)}
\begin{itemstep}
\item overlays{\$n\$} で slide 環境を囲む\
\{from, only, until\}Slide で何枚目に表示するか指定\
\onlySlide*{2}{
⇒ ‘*’ をつけたら同じ位置に表示してくれる
}
\fromSlide{3}{\item itemstep 環境で簡単に 1 行ずつ表示}
\end{itemstep}
\end{slide}
}
```

と入力して実現します。

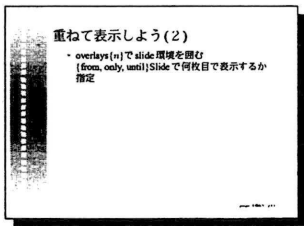


図 23: 最初の表示

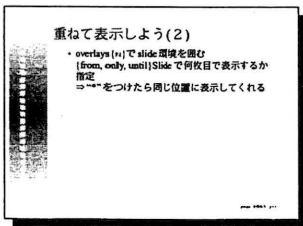


図 24: 注意書きを表示

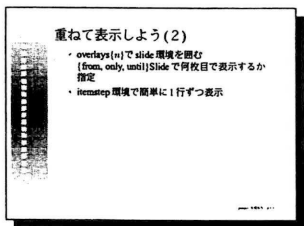


図 25: 注意書きを消して次を表示

図 23 から図 25 の `\onlySlide*{p}{stuff}` では、*stuff* の部分が単純ですのうまく表示できます。“うまくいかない” 場合というのは、不自然な空白などが生じる場合で、図 26 から図 28 がこれにあたります。これらの図では、一度しか表示しない部分 (`\onlySlide` の中身) を

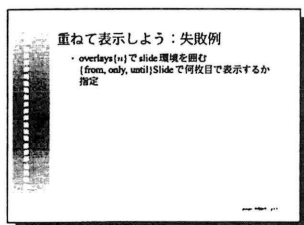


図 26: 最初の表示

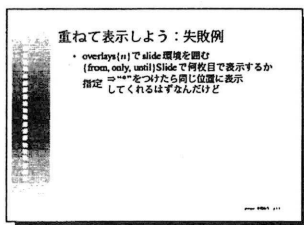


図 27: ここで空白 (「指定」の上) が生じた

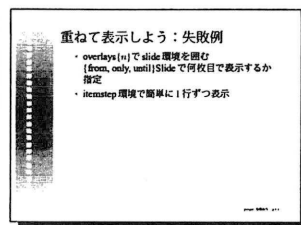


図 28: 空白が消えた

独立した行ではなく、すでにある行の右に表示させようとしています。ところが、図 27 中の「指定」の上に余分な空白が生じ、図 28 中ではこの空白が消えています。このようなページからなるスライドを連続的に表示すると、微妙な“動き”が生じてしまい、余計な注意をひくことになり好ましくありません。図 26 から図 28 は

```
\overlays{3}{%
\begin{slide}{重ねて表示しよう：失敗例}
  \begin{itemstep}
    \item overlays\{$n$\}で slide 環境を囲む\
      \{from, only, until\}Slide で何枚目で表示するか指定
    \onlySlide*{2}{
      \parbox{7cm}{→‘*’ をつけたら同じ位置に表示してくれるはずなんだけど}
    }
    \fromSlide{3}{\item itemstep 環境で簡単に 1 行ずつ表示}
  \end{itemstep}
\end{slide}
}
```

と入力しています。余計な空白の元凶は `\onlySlide*{p}{stuff}` の *stuff* が複数行からなるパラグラフになっていることです。そのため、最初のページにあった行を構成するために必要な縦方向の長さより多く長さを必要とします。

これを回避するには *stuff* を PSTricks のコマンドである `\rput` の中に押しこめてしまいます。

```

\overlays{3}{%
\begin{slide}{重ねて表示しよう：rput}
  \begin{itemstep}
  \item overlays{\$n\$}でslide環境を囲む\\
    \{from, only, until\}Slideで何枚目で表示するか指定
    \onlySlide*{2}{
      \rput[1](0,-.5em){\parbox{7cm}{⇒‘*’をつけたら同じ位置に表示してくれるはずな
        んだけど}}
    }
    \fromSlide{3}{\item itemstep環境で簡単に1行ずつ表示}
  \end{itemstep}
\end{slide}
}

```

\rputの一般形は\rput(x,y){stuff}となります。(x, y)は原点の指定で、通常はstuffの中央をどこに置くかを意味します。stuffの左側を原点に合わせるために、オプション[1]を指定しています。その他のオプションや詳しい使い方については、マニュアル[8]を参照してください。

このようにすると、図29から図31のようになり、余計な空白は生じません。

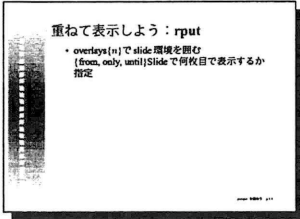


図 29: 最初の表示

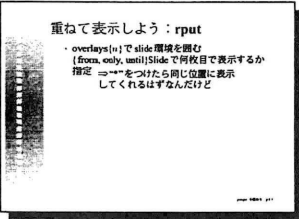


図 30: 空白は生じていない

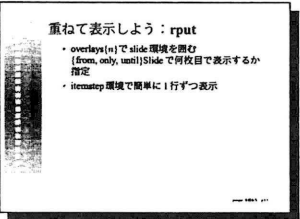


図 31: 注意書きがきえて最後の表示

3.3 スタイルのカスタマイズ

\slideCaption{caption}を利用すると、明示的に与えた“caption”の後にページ番号が表示されます。デフォルトでは“現在のページ番号/総ページ数”という形式ですが、\documentclassのオプションでnototalを指定すると、現在のスライド番号のみを表示します。

prosperにおける1枚のスライドは、タイトルと本文からなります。これらの書式(フォントと色)を変更するには、プリアンブルで\FontTitle{C}{BW}や\FontText{C}{BW}を用います。例えば、

```

\FontTitle{%
  \usefont{T1}{ptm}{m}{sl}\fontsize{22pt}{20pt}\selectfont\orange}{%
  \usefont{T1}{ptm}{m}{sl}\fontsize{22pt}{20pt}\selectfont\blue}
\FontText{%
  \mellow\usefont{T1}{phv}{m}{n}\fontsize{14.4pt}{14pt}\selectfont}{%
  \black\usefont{T1}{phv}{m}{n}\fontsize{14.4pt}{14pt}\selectfont}

```

のように用います。

itemize 環境や itemstep 環境の行頭文字を変えるには `\myitem{n}{def}` ($n = 1, 2, 3$) を使います。デフォルトでは、

```

\myitem1{{\scriptsize\green\raisebox{2pt}{\ensuremath{\bullet}}}}
\myitem2{{\scriptsize\green\raisebox{2pt}{\ensuremath{\bullet}}}}
\myitem3{{\scriptsize\green\raisebox{2pt}{\ensuremath{\bullet}}}}

```

という定義になっています。これを例えば、

```

\myitem1{\scalebox{.7}{\yellow\raisebox{2pt}{◆}}}
\myitem2{\scalebox{.5}{\yellow\raisebox{2pt}{▲}}}
\myitem3{\scalebox{.4}{\yellow\raisebox{2pt}{●}}}

```

などに変更して使います。この例では `\scalebox` を使っているので `graphics` か `graphicx` パッケージが必要です。

4 絵と図形

効果的なプレゼンテーションには、難しい数式や説明を並べるより、わかりやすい絵や図形がよい場合もあるでしょう。単に他のソフトで書いた絵を prosper で使うには、`graphics` や `graphicx` パッケージを使って EPS ファイルを取りこみます。これらのパッケージは、回転・縮小・拡大・色の機能も提供します。詳しくは文献 [1] を参照してください。

この節では、矢印やテキストを囲む枠などを L^AT_EX から使う方法を説明します。これらは prosper とは独立なパッケージで提供されている機能です。したがって prosper ではないクラスファイルでも使える機能です。

4.1 PSTricks パッケージ

PSTricks パッケージは T. Van Zandt による、PostScript の描画能力を L^AT_EX から使うためのパッケージです。そのため PostScript に描画できることは、ある程度なんでも L^AT_EX の中から描くことができます¹²。つまり、非常に多機能であり、すべてをここで説明することはでき

¹²しかし、例えば 3 次元の描画は苦手です。

ません¹³。この節では、プレゼンテーションでよく使うと思われる図形に絞ってこのパッケージの使い方を紹介します。より詳しい使い方はマニュアル [8] や文献 [5] を参照してください。

PSTricks パッケージでは、デフォルトで使える色として、グレースケールでは black, darkgray, gray, lightgray, white が、カラーでは red, green, blue, cyan, magenta, yellow が用意されています。これらのコマンドは、`\blue` (ここは青く印字されます) のように使います。

次に線や多角形を書きます。線は `\psline{arrows}(x_1,y_1)(x_2,y_2)...\(x,y)` で書きます。`{arrows}` を省略すると通常の線を引きますが、`{->}`, `{<-}`, `{<->}` などを指定することで矢印にすることができます。他にも `{|-|}`, `{o-o}`, `{[-]}` などが使えます。

線の太さ (`linewidth` でデフォルト 0.8pt) や線種 (`linestyle` でデフォルトは solid)、線の色 (`linecolor` でデフォルトは black) は、オプション引数で変えます。例えば、線の太さと線種を変えるときは、

```
\psline[linestyle=dotted, linewidth=3pt](x_1, y_1)(x_2, y_2)
```

とします。linestyle としては、他に none と dashed が使えます。

多角形は `\pspolygon(x_1,y_1)...\(x,y)` で、曲線は `\pscurve(x_1,y_1)...\(x,y)` を使います。

4.2 テキストボックス

最も単純なテキストボックスは `\psframebox{texts}` です。例えば

```
\begin{slide}{フレームボックス}
  \begin{itemize}
    \item \psframebox{1 行のフレームです}
    \item \psframebox{\parbox{4cm}{2 行以上の場合は\parbox で囲みます}}
    \item \psframebox[fillstyle=solid, fillcolor=lightgray]{薄く塗りつぶし}
  \end{itemize}
\end{slide}
```

と入力すると、図 32 のように表示されます。線の太さや塗りつぶし (`fillstyle` でデフォルトは none) をオプション引数で指定しています。

その他に、以下のようなテキストボックスが使えます。

- `\psdblframebox{texts}`
- `\psshadowbox{texts}`
- `\pscirclebox{texts}`
- `\psovalbox{texts}`

¹³ちなみに付属のマニュアル [8] は 100 ページほどあります。

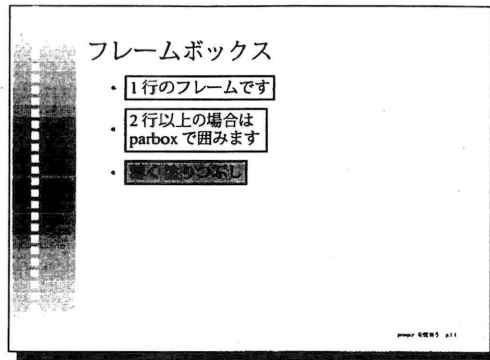


図 32: テキストボックス

4.3 ノードと矢印

この節では、複数の単語や図形に名前をつけ(ノードと呼びます)、ノード間に矢印や線を引く方法を説明します。既に説明した `\psline` でテキスト間の矢印を引こうとすると、テキストが配置された後の絶対的な位置を座標で指定しないとイケません。これは非常に面倒な作業が必要で、プレビューと座標の修正を何回か繰り返さないといけません。さらに、内容を修正したりすると、線の接続先の位置が変わることになり、再び座標を修正しないとイケません。

まず、プリアンブルで `\usepackage{pst-node}` を宣言して、ノード関連のコマンドが使える状態にします。

次に、線の接続先と接続元に名前をつけてノードにします。名前は `\rnode{name}{stuff}` とします。

少なくとも2つノードを作ったら、`\ncline{node1}{node2}` で2ノード間に線を引きます。例えば、

```
\usepackage{pst-node} % この宣言が必要
\begin{document}
\begin{slide}{ノード間の矢印 (ncline)}
  \begin{itemize}
    \item 1行目です。 \rnode{First}{ここに}矢印の先をもってきてください。
    \item[]
    \item 数式 \rnode{Eq1}{ $y = \int f(x) dx$ } でもよいです。
      \ncline{<->}{First}{Eq1}
    \end{itemize}
  \end{slide}
```

と入力すると、図 33 のように表示されます。ここで2つ目の `\item` は、線があることを分かりやすくするための空白行を作ります。

他に `\nccurve{node1}{node2}` や `\ncarc{node1}{node2}` も使えます。これらは、それぞれ直線の代わりに曲線と弧でノード間を接続します。

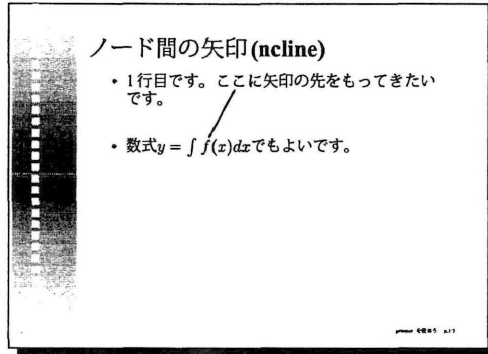


図 33: ノード間の矢印

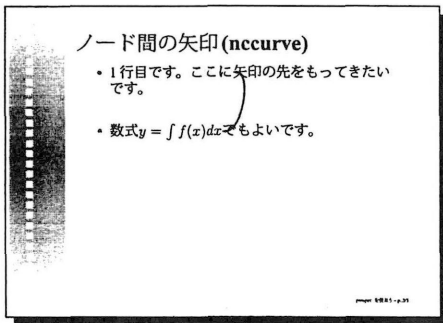


図 34: \nccurve

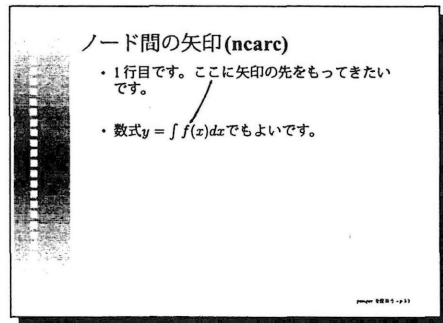


図 35: \ncarc

```

\begin{slide}{ノード間の矢印 (nccurve)}
  \begin{itemize}
    \item 1 行目です。 \rnode{First}{ここに}矢印の先をもってきてください。
    \item[]
    \item 数式 \rnode{Eq1}{ $y=\int f(x)dx$ }でもよいです。
      \nccurve[angleA=-90,angleB=90]{<->}{First}{Eq1}
    \end{itemize}
\end{slide}

\begin{slide}{ノード間の矢印 (ncarc)}
  \begin{itemize}
    \item 1 行目です。 \rnode{First}{ここに}矢印の先をもってきてください。
    \item[]
    \item 数式 \rnode{Eq1}{ $y=\int f(x)dx$ }でもよいです。
      \ncarc[arcangle=16]{<->}{First}{Eq1}
    \end{itemize}
\end{slide}

```

と入力すると、図 34 と図 35 のように表示されます。

`\nccurve` のオプション `[angleA=-90,angleB=90]` は、矢印をどちらからノード接続させるかを指定します。`\ncarc` のオプション `[arcangle=16]` は、弧の角度を指定します。

4.4 ハイパーリンク

`hyperref` は \LaTeX の `\ref` や `\cite` などを PDF のハイパーリンクに変換し、`section` 環境の引数で与えたセクションタイトルなどを PDF のしおり (bookmark) に変換するパッケージです。

`prosper` は内部で `hyperref` パッケージを呼びだしています。そのため `\usepackage` を宣言する必要はありません。`prosper` を利用する場合は `slide` 環境のタイトル部分が自動的にしおりになります。しかし、2.3 節に書いたようにタイトルに日本語が含まれると、しおりの部分が文字化けしてしまいます。これを解消する方法とそのための変換スクリプトが文献 [2] にあります。この変換スクリプトを `convert.pl` という名前で作成したとすると、

```
% perl convert.pl < input.ps > output.ps
```

として実行します。得られた PostScript ファイルを Acrobat で PDF にすると、しおりの部分が文字化けしません。

`hyperref` を使えば、スライド中に URL を埋め込むこともできます。`\href{URL}{text}` として使います。`text` の部分がスライド中に表示されます (図 36 参照)。Acrobat Reader でここをクリックすると、通常ブラウザが起動し URL で指定した Web ページやファイルが表示されます。

また、スライド中のある場所に名前をつけて、別スライドからそこへリンクを張ることもできます。名前をつけるのは `\hypertarget{Name}{text}` で行ないます。`text` の部分は数式で

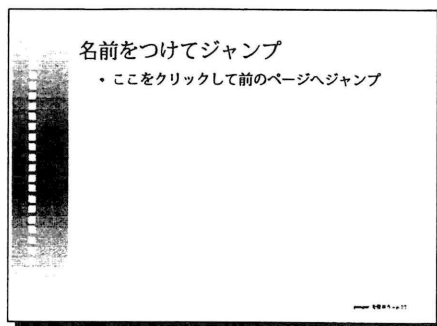
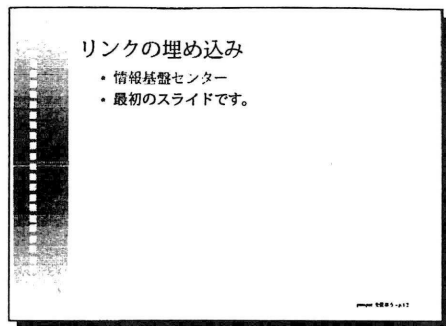


図 36: 1 行目がセンター Web ページへのリンク
 図 37: “ここ” が前のページへのリンク

も構いません。リンクを張るのは `\hyperlink{Name}{text}` で行ないます (図 37 参照)。共に `text` の部分が実際に表示されます。

図 36 と図 37 は以下のように入力しています。

```
\hypersetup{colorlinks=true} % プリアンプルで指定する
\begin{document}
\begin{slide}{リンクの埋め込み} % 最初のスライド
  \begin{itemize}
  \item \href{http://www.cc.kyushu-u.ac.jp}{情報基盤センター}
  \item 最初の\hypertarget{FirstPage}{スライド}です。
  \end{itemize}
\end{slide}
\begin{slide}{名前をつけてジャンプ} % 2 枚目のスライド
  \begin{itemize}
  \item \hyperlink{FirstPage}{ここ}をクリックして前のページへジャンプ
  \end{itemize}
\end{slide}
```

PDF ファイルを表示させた時にリンクの場所が分かるように `\hypersetup{colorlinks=true}` としています。デフォルトでは `false` となっています。

5 インストール

prosper の配布元は <http://prosper.sourceforge.net/> です。このページ右上の “Download” がダウンロードページへのリンクになっています。2002 年 3 月 10 日現在の最新版は `prosper-1.00.4.tar.gz`¹⁴です。

prosper のインストールするには、クラスファイルとスタイルファイルを適当なディレクトリにコピーするだけです。例えば

¹⁴<http://prdownloads.sourceforge.net/prosper/prosper-1.00.4.tar.gz>

```
# mkdir /usr/local/share/texmf/tex/latex/prosper/
# cp *.cls *.sty /usr/local/share/texmf/tex/latex/prosper/
```

とします。場合によっては `mktexlsr` を起動して `ls-R` ファイル¹⁵を更新する必要があるかもしれません。

`prosper` を使うには `LATEX` と `dvips` が必要です。 `dvips` は 5.85 以上のバージョンが必要です。またクラスファイルとしては、以下のものがが必要です。

- `seminar`
- `PSTricks`
- バージョン 6.69 以上の `hyperref`

これらのクラスファイルをインストールするためには、さらに別のクラスファイルが必要となる場合があります。 `teX` の最新版をインストールすることで、 `prosper` に必要なクラスファイルが一度にインストールされます¹⁶。

PDF 出力を `ps2pdf` コマンドで行なう場合は、バージョン 6.0 以降の `Ghostscript` も必要となります。英語のプレゼンテーション資料の場合は `ps2pdf` で作成した PDF ファイルを使えば、解像度に依存せず綺麗に出力されますが、日本語を使う場合は `Acrobat` を使うほうがよいと思います。

6 おわりに

本稿では `prosper` を使う上で基本的な事柄を説明しました。 `prosper` のマニュアル [6] そのものは、それほど量は少ないのですが、マニュアルに載っていない使い方や技は、まだまだありそうです。また、4 節の `PSTricks` は非常に高機能で、本稿ではそのほんの一部しか説明していません。マニュアル [8] はぶ厚いですが、豊富なサンプルが載っていますので、自分の描きたい図形を探し、サンプルコードを模倣するだけで十分使えると思います。

もともと `LATEX` は細い位置の調整を要するプレゼンテーション資料には向いていないと思います。やはり最後の調整は必要ですが、著者もスマートなやり方を知っているわけではなく、位置をずらしてプレビューする作業を繰り返すこととなります。使ってみて、便利な使い方やマクロなどがあれば、ぜひ著者にもお知らせください。

しかし、このような作業の繁雑さを考慮しても、使う価値は高いと思う人は多いと思います。数式が必要な人、UNIX で作業を完結させたい人¹⁷などにとっては特に使ってみる価値はあると思います。

参考文献

- [1] 牧嶋 直子、わくわく pLaTeX2e~graphics パッケージをあけてみよう~、九州大学大型計算機センター広報 Vol. 31, No. 1, pp. 19-33, 1998.

¹⁵このファイルは `kpathsea` ライブラリのためのファイル名データベースです。

¹⁶ただし `prosper` そのものは別にインストールする必要があります。

¹⁷または Windows を使いたくない人かもしれません。

- [2] 松本 隆太郎、hyperrefと日本語、<http://www.rmatsumoto.org/tex-ps-pdf/hyperref-ja.html>
- [3] 渡部 善隆、 Prosper の使い方 - A LaTeX class for writing transparencies -, <http://www.cc.kyushu-u.ac.jp/RD/watanabe/RESERCH/PROSPER/index.html>
- [4] T. Esser, The teTeX Homepage, <http://www.tug.org/teTeX/>
- [5] M. Goosens, S. Rahtz, F. Mittelbach 著: 鷺谷 好輝 訳、 LaTeX グラフィックスコンパニオン、アスキー、2000年。
- [6] F. Goualard, Manual for the prosper class (配布物の doc/prosper-doc.pdf).
- [7] S. Rahtz, Hypertext marks in L^AT_EX: the hyperref package, <http://www.tug.org/applications/hyperref/manual.html>.
- [8] T. Van Zandt, PSTricks: PostScript macros for Generic Tex. User's Guide, <http://www.tug.org/applications/PSTricks/>.