probsoln.sty v2.0: LaTeX Package to help create problem sheets

N.L.C. Talbot

19th Dec 2000

Contents

1 Introduction

The package `probsoln.sty` is designed for teachers or lecturers who want to create problem sheets for their students. This package was designed with specifically mathematics problems in mind, but can be used for other subjects as well. The idea is to create a file containing a large number of problems with their solutions which can be read in by LaTeX, and then select a number of problems to typeset. This means that once the database has been set up, each year you can easily create a new problem sheet that is sufficiently different from the previous year, thus preventing the temptation of current students seeking out the previous year's students, and checking out their answers. There is also an option that can be passed to the package to determine whether or not the solutions should be printed. In this way, one file can either produce the student's version or the teacher's version.

2 Package Options

The following options may be passed to this package:

`answers`	Show the answers
`noanswers`	Don't show the answers (default)

3 Commands Provided

3.1 Creating a New Problem

```
\newproblem[⟨nargs⟩]{⟨label⟩}{⟨problem⟩}{⟨solution⟩}
```

\newproblem A new problem is defined using the command `\newproblem`. This does not print anything, but merely stores the problem. The argument ⟨*label*⟩ is a unique string that is assigned to this problem so that it can be used later. The argument ⟨*problem*⟩ is normal LaTeX code that should be used to typeset the problem. The argument ⟨*solution*⟩ is normal LaTeX code that should be used to typeset the solution, if required. For example:

```
\newproblem{quaddiff}{%
%This is the problem
\begin{displaymath}
f(x) = x^2 + 3x + 4
\end{displaymath}
}{%
%This is the solution
\begin{displaymath}
f'(x) = 2x + 3
\end{displaymath}
}
```

The optional argument ⟨*nargs*⟩ specifies the number of parameters this problem will take. By default this value is 0, but any value from 1 to 9 may be used. Each parameter is refered to by `#1`, `#2`, . . . , `#9`. For example, the following problem takes one parameter:

```
\newproblem[1]{sindiff}{%
\(f(x) = \sin(#1x)\)
}{%
\(f'(x) = #1\cos(#1x)\)
}
```

The `quaddiff` problem shown above can be made more generic by using parameters:

```
\newcount\ctr
\newproblem[3]{diff:quad}{%
\(f(x) =
\ifnum#1=0
\else
\ifnum#1=1\else#1\fi x^2
\fi
\ifnum#2=0
\else
\ifnum#2>0 \ifnum#1=0 \else + \fi \fi
\ifnum#2=1\else#2\fi x
\fi
\ifnum#3=0
\else
\ifnum#3>0 \ifnum#2=0 \ifnum#1=0 \else + \fi \else + \fi\fi
#3
\fi\)
}{%
\(f'(x) =
\ifnum#1=0
\else
\ctr=2
\multiply\ctr by #1
\the\ctr x
\fi
\ifnum#2=0
\else
\ifnum#2>0 \ifnum#1=0 \else + \fi \fi
#2
\fi
 % print 0 if both #1 and #2 are 0
\ifnum#1=0 \ifnum#2=0 0 \fi\fi
\)
}
```

The three parameters correspond to the coefficients. Note that they must all be integers since TEX only performs integer arithmetic.

To generate a database, simply create a `.tex` file where all the problems are defined using `\newproblem`, and either `\input` it at the start of your document if you want to use specific problems (see Section 3.2), or pass it to `\selectrandomly` (see Section 3.3).

3.2 Displaying a Problem

`\useproblem{⟨label⟩}`

`\useproblem`

Once a problem has be defined using `\newproblem`, it can be typeset using the command `\useproblem`. If the problem was defined to take arguments, the arguments to the problem should come after the label. In the case of the `sindiff` example above, the command `\useproblem{sindiff}{2}` would produce the following:

$f(x) = \sin(2x)$

Solution: $f'(x) = 2\cos(2x)$

whereas the command `\useproblem{diff:quad}{3}{0}{-2}` would produce:

$f(x) = 3x^2 - 2$

Solution: $f'(x) = 6x$

Suppose all the above problems are defined in the file `probs.tex`, then the following code will create a problem sheet with four questions in it:

```
\documentclass{article}

\usepackage{probsoln}
\input{probs}
```

```
\begin{document}
Differentiate the following functions with respect to $x$:
\begin{enumerate}
\item \useproblem{quaddiff}
\item \useproblem{sindiff}{4}
\item \useproblem{diff:quad}{2}{3}{1}
\item \useproblem{diff:quad}{0}{1}{2}
\end{enumerate}
\end{document}
```

The answer sheet can then be generated by passing the option `answers` to the `probsoln` package.

3.3 Selecting Problems at Random

`\selectrandomly{`⟨*filename*⟩`}{`⟨*n*⟩`}`

`\selectrandomly` The command `\selectrandomly` will select ⟨*n*⟩ problems that are defined in the file ⟨*filename*⟩. Each problem is preceeded by a `\item`, so the command `\selectrandomly` should occur within one of the list-like environments, such as `enumerate`. For example:

```
\begin{enumerate}
\selectrandomly{easy.tex}{4}
\end{enumerate}
```

will result in four numbered problems, selected at random from the file `easy.tex`. (The `.tex` extension may be omitted.)

Multiple `\selectrandomly` commands may be used. For example:

```
\begin{enumerate}
\item Differentiate the following functions with respect to $x$:

\begin{enumerate}
\selectrandomly{samples/easy.tex}{3}
\selectrandomly{samples/args.tex}{1}
\end{enumerate}

\selectrandomly{samples/implicit.tex}{1}
\selectrandomly{samples/1stprncpl.tex}{1}

\end{enumerate}
```

This will result in a total of 6 problems, numbered 1(a), 1(b), 1(c), 1(d), 2 and 3.

If a randomly selected problem requires arguments, a message similar to the following will be displayed:

```
Problem diff:quad requires 3 argument(s), please specify (e.g. {5}{3}):
```

Enter the required arguments, where each argument is enclosed in braces ({ }).

3.4 Other Commands

`\PSNrandseed` The command `\PSNrandseed{`⟨*n*⟩`}` specifies the seed for the random number generator. For example, `\PSNrandseed{\year}` will produce a different set of random problems each year, whereas `\PSNrandseed{\time}` will produce a different set of problems each time you LaTeX the problem (as long as you leave at least a minute between runs.)

`\showanswers` The command `\showanswers` will show the solutions from that point on. May be localised by placing within a group.

`\hideanswers` The command `\hideanswers` will hide the solutions from that point on. May be localised by placing within a group.

`\solutionname` By default, the solution is preceeded by the text: **Solution:**. This can be changed by redefining the macro `\solutionname`.

`showanswers` The boolean variable `showanswers` is defined to be true if the answers are shown and false otherwise. You can therefore do something like:

```
\ifthenelse{\boolean{showanswers}}{\textbf{Solution Sheet}}{}
```

and **Solution Sheet** will be printed only if the answers are displayed. (For more information on `\ifthenelse` and `\boolean` see the documentation for the `ifthen` package by David Carlisle.)

4 Error Messages

```
! Package probsoln Error: Label ... already used.
```

Each label identifier used in `\newproblem` must be unique. Check to make sure you haven't used the same label more than once. Also check to make sure you haven't `\input`ed or randomly selected from the same file more than once. (Or `\input`ed and randomly selected from the same file.)

```
! Package probsoln Error: Label ... undefined.
```

You need to define a problem before you can use it. Check to make sure you haven't mis-spelt it.

```
! Package probsoln Error: Requested number too large.
```

You have asked for more problems than are defined within the specified file. All problems in that file will be selected.

5 Contact Details

Dr Nicola Talbot
School of Computing Sciences
University of East Anglia
Norwich. NR4 7TJ
http://theoval.cmp.uea.ac.uk/~nlct/