Manual of `lingproblems`

Michał Boroń

April 2021

1 Setup

If you do not install the package on your local computer to make it accessible for all the files, TEX automatically looks for the `.sty` file in your local folder in which you store your main `.tex` file. It means that you should save `.sty` file in that folder. In this case, you need to include the following in your TEX documents (which means that the file `lingproblems.sty` is in the same directory as your `.tex` file):

```
\usepackage{./lingproblems}
```

If you want to install this package on your local machine and use it globally, I would recommend typing in your favourite web search machine: *how to install .sty package* and eventually attaching the name of your favourite editor. The question is for sure answered.

The number of possibilities for each operating system and each editor is huge, although the general way to do it, is to make a new directory with the name of the package, i.e. `lingproblems`, in the path where other directories of other packages are stored.[1] Then copy the `.sty` file to the newly created directory. After that, you will need to update the whole folder with these directories. On UNIX-based systems it can be done with

```
$ sudo mktexlsr
```

or something similar. Then you can normally import the package with the following command:

```
\usepackage{lingproblems}
```

2 The environment `lingproblem`

This environment allows adding a new linguistic problem. The format of such a problem is based on the format used at the International Linguistics Olympiad (ILO), a worldwide contest in linguistics for high-school students held annually. Every problem consists of a problem statement, assignments and a short information about the language the problem is about. Furthermore, a single problem might include an introduction as well as a short explanation of some unusual phonemes or glyphs used in the problem. Students usually do not need any prior knowledge about the language to complete the assignments.

At an instance of the environment `lingproblem`, we generally pass two arguments: the name of the problem and the author of the problem. They will be later used in the formatting of the whole statement. The first argument is the title for the problem and the second is the author's name. Let us look at an example.

[1] On my Linux it is: `/usr/share/texlive/texmf-dist/tex/latex/`

```
\begin{lingproblem}{Cyrillic script}{Andrei Smirnov}
Here comes your problem statement.
\end{lingproblem}
```

The preceding outputs the following (without the box):

Cyrillic script

Here comes your problem statement.

Beyond that, there is an option not to input and print the name of the problem. This can be achieved by using `lingproblem*` instead. Please note that you will still need to provide an author or leave the curly braces empty.

```
\begin{lingproblem*}{Andrei Smirnov}
Here comes your problem statement.
\end{lingproblem*}
```

The preceding outputs the following (without the box):

Here comes your problem statement.

2.1 The environment `tasks`

In this environment (inside the environment `lingproblem`) the assignments are formulated. Every assignment (or task, simply) is initiated with a command `\task{}`. It generates a bullet of the following form: **(a)**. Bold, small letters of the Latin alphabet are used. Again, it follows some tradition of the ILO. You type the assignment as the argument to this command.

There is also an option to include a task with an indicated number of points. You can achieve this by using `\taskp{}{}` instead, whereby you write the number of points as the second argument.

```
\begin{lingproblem}{Cyrillic script}{Andrei Smirnov}
Here comes your problem statement.

\begin{tasks}
\task{The first task.}
\taskp{The second task.
This time with an indicated number of points.}{10 points}

\end{tasks}

\end{lingproblem}
```

The preceding outputs the following (without the box):

Cyrillic script

Here comes your problem statement.

(a) The first task.

(b) The second task. This time with an indicated number of points. [*10 points*]

2.2 The environment `langinfo`

As mentioned above, you generally want to add some information about the language the problem is about. To do that, you can use this environment. It automatically generates the danger symbol ⚠ and add the author of the problem at the end in italics. This a general standard at the International Linguistics Olympiad.

```
\begin{lingproblem}{Cyrillic script}{Andrei Smirnov}
Here comes your problem statement.

\begin{tasks}
\task{The first task.}
\end{tasks}

\begin{langinfo}
This language is a member of the F family.
It is spoken in X by Y speakers.

These are some unusual phonemes used in the problem: \'a.
\end{langinfo}

\end{lingproblem}
```

The preceding will produce (without the box):

Cyrillic script

Here comes your problem statement.

(a) The first task.

⚠ This language is a member of the F family. It is spoken in X by Y speakers. These are some unusual phonemes used in the problem: á. —*Andrei Smirnov*

You can also omit the author at the end by using the environment `langinfo*` instead. In this case, the second argument of `\begin{problem}{}{}` will not be used at all, so you can leave it empty.

```
\begin{lingproblem}{Cyrillic script}{}
Here comes your problem statement.

\begin{tasks}
\task{The first task.}
\end{tasks}

\begin{langinfo*}
This language is a member of the F family.
It is spoken in X by Y speakers.
These are some unusual phonemes used in the problem: \'a.
\end{langinfo*}

\end{lingproblem}
```

The preceding outputs the following (without the box):

Cyrillic script

Here comes your problem statement.

(a) The first task.

⚠ This language is a member of the F family. It is spoken in X by Y speakers. These are some unusual phonemes used in the problem: á.

3 Example

Please view `example.pdf` and `example.tex` which demonstrate the usage of this package on an example problem.