How to use hyplain — version 1.0

Many (well, at least more than twelve) years ago, LaTeX was not as usable as it is today. So we were forced to use AMS-TeX for typesetting mathematical papers.

At the very beginning there were two plain formats to use, the original one and the one (called plainit) for Italian hyphenation. With the advent of TeX3 we could use only one and define some switch for selecting the language.

I found a package called HyMaster that could be used for this purpose; it was developed by Daniel Flipo and Laurent Siebenmann for French. Since Italian typesetting rules do not require the contortions of the French, it was simple to adapt only one file for canning a format file.

With the advent of LaTeX2$_\varepsilon$, everything was forgotten. Until recently, when a user in a discussion forum showed some code that I recognized as my modifications to HyMaster! He proposed some procedure to build the format which was very complicated (you see, at that time we were using OzTeX; nowadays, teTeX or MiKTeX require different actions). So I tried to rethink to the business.

Come usare hyplain

Parecchi (be', almeno dodici) anni fa, LaTeX non era così facile da usare come è oggi. Per comporre testi matematici eravamo quindi costretti a usare AMS-TeX.

All'inizio c'erano due formati plain da usare, l'originale e quello (chiamato plainit) per la sillabazione italiana. Con l'avvento di TeX3, si poté usare un solo formato con la possibilità di cambiare le regole di sillabazione con un comando.

Trovai un pacchetto chiamato HyMaster che poteva essere usato per questo scopo; era stato sviluppato da Daniel Flipo e Laurent Siebenmann per il francese. Siccome la tipografia italiana non richiede le contorsioni di quella francese, fu semplice adattare un solo *file* per costruire un formato.

Con l'avvento di LaTeX2$_\varepsilon$, tutto fu dimenticato. Finché, recentemente, un utente di un gruppo di discussione esibì del codice che riconobbi come le mie modifiche a HyMaster! Proponeva una procedura piuttosto complicata per costruire il formato (a quel tempo usavamo OzTeX; ora teTeX e MiKTeX richiedono azioni diverse). Perciò ho provato a ripensare alla faccenda.

As you can see, hyphenation in US English and in Italian behave properly. The text above has been typeset with more hyphens than usual.

The package consists of three files: `hyplain.tex`, `hyrules.tex` and `hylang.tex`. Only the third one needs to be modified by the users who want to define languages to employ.

The basic definition of a language is given by a command

```
\definelanguage{xx}{YY}{filename}
```

where `xx` is the ISO abbreviation for the language (`en` for English, `it` for Italian, `de` for German), `YY` is the ISO abbreviation for the nation and `filename` is the name of the hyphenation pattern file. Thus

```
\definelanguage{it}{IT}{ithyph}
```

defines Italian.

There is also `\definedialect` for defining a language which shares hyphenation patterns with another; for example

```
\definedialect{de}{AT}{de}{DE}
```

defines "German for Austria", assuming that

```
\definelanguage{de}{DE}{dehypht}
```

has already been given.

After a `\definelanguage` command, `\refinelanguage` should be given; it has four arguments

`#1` the two-letter language abbreviation,
`#2` the two-letter nation abbreviation,
`#3` the code to be executed when the language comes into action,

`#4` the code to be executed when another language is called.

This command should always contain in the third argument the left and right hyphenation minima: for example, we set

```
\refinelanguage{en}{US}{\hyphenmins{2}{3}}{}
```

in order to enforce the usual conventions for American English; as another example, we set

```
\refinelanguage{it}{IT}{\hyphenmins{2}{2}\lccode‘\’=‘\’}{\lccode‘\’=0 }
```

because the lower case code of the right quote should not be zero for correct Italian hyphenation. We need to "undo" the effect since we don't know whether the user calls the language switch inside a group or not. The command

```
\hyphenmins{x}{y}
```

is an abbreviation for `\lefthyphenmin=x\righthyphenmin=y` and it is not necessary to "undo" it.

The commands `\definedialect` and `\refinedialect` (analog to `\refinelanguage`) can be given also in the user's document where, of course, it is too late to define new languages with their hyphenation patterns. Anyway, users can simulate new language definitions by defining a dialect to language `\zz_ZZ`, which is a fallback without hyphenation tables.

If one wants to extend the functionalities of a language, there is also the command

```
\addto{xx}{YY}{do}{undo}
```

which can be used, for example, in the following way

```
\addto{it}{IT}{\frenchspacing}{\nonfrenchspacing}
```

in order to enforce French spacing for Italian (in Italy).

Finally, there is a command to choose a defined language (or even an undefined one, try it)

```
\selectlanguage{xx}{YY}
```

which selects language `xx` for `YY`. Users can add aliases for this with

```
\addalias{\command}{xx}{YY}
```

(see `hylang.tex` for examples). After saying

```
\addalias{\IT}{it}{IT}
```

you can simply say `\IT` to switch to Italian. This alias is actually predefined in the present version of `hylang.tex` as well as `\US` for US English.

Installation

Put the three files in some directory and process with

```
tex -ini hyplain
```

and after this put the produced format in some suitable place of the TEX tree. After this

```
tex -fmt hyplain filename
```

will typeset `filename.tex` using the new format. This is for a *nix system and a Web2C distribution like teTEX or TEXLive; for MiKTEX find the way, I don't use it. You can use also `pdftex` and you can use the newly created `hyplain.fmt` to build AMS-TEX over it. For `pdftex` you should use

```
pdftex -ini hypdfplain.ini
```

and then

```
pdftex -fmt hypdfplain filename
```

If you want to access the extended features of ε-TEX, then do

```
pdftex -ini '*hypdfplain.ini'
```

Enrico Gregorio
October 17, 2006