ccref - cross referencing with proper definite articles

Jinwen XU

October 2021, in Paris

Abstract

The package ccref provides a command \ccref parallel to cleveref's \cref for handling definite articles properly (especially for the article contractions in some European languages).

/ 1 / The motivation

By default, with cleveref's \cref to reference theorem-like environments, the names do not contain definite articles. While this might be acceptable for English, it is certainly not good enough for languages such as French, Italian, Portuguese, Spanish, etc. – in these cases there shall be grammatical errors and would give you a strong feeling that it is machine-generated.

However, even if we manually add the definite articles to the names, there would still be other problems. As an example, if we define the French names to be:

```
\crefname{theorem}{le théorème}{les théorèmes}
\crefname{proposition}{la proposition}{les propositions}
```

then when one writes (which means "*We can deduce this from ...*")

```
On peut le déduire de \cref{thm1,thm2,prop3}.
```

the result would be:

> On peut le déduire **de les** théorèmes 1 et 2 et **la** proposition 3.

which is wrong, as the correct result should be:

> On peut le déduire **des** théorèmes 1 et 2 et **de la** proposition 3.

\cref cannot handle such cases automatically — and that is when \ccref comes into play.

/ 2 / The usage

2.1 | How to load it ?

Simply load the package with:

```
\usepackage{ccref}
```

Corresponding to: ccref 2021/10/30

Tip

- Since ccref uses cleveref internally, it should usually be placed at the last of your preamble, and notably, after varioref and hyperref.
- To handle article contractions correctly, `\ccref` shall detect the current language, thus you need to use packages such as babel or polyglossia to set your languages, and use commands like `\selectlanguage` to select them appropriately.

2.2 | How to use it ?

Then you can use the command `\ccref` as follows:

- `\ccref[`⟨*prep*⟩`]{`⟨*labels*⟩`}`
  - This will pass the preposition ⟨*prep*⟩ to the definite articles that follows. Its behavior depends on the current language (for example, in Spanish, ⟨*prep*⟩ is passed only to the first definite article, while in French it is passed to everyone).
- `\ccref-[`⟨*prep*⟩`]{`⟨*labels*⟩`}` and `\ccref+[`⟨*prep*⟩`]{`⟨*labels*⟩`}`
  - In case the automatic version does not meet your needs, here are two manual ones. The `-` version passes the preposition ⟨*prep*⟩ only to the first definite article, while the `+` version passes ⟨*prep*⟩ to every definite article.

Tip

There is also a stared version `\ccref*` for generating the same referencing text but without creating hyperlinks.

However, before using it, you should first define the `\crefname`s carefully. The definite article in `\crefname`s needs to be marked manually using `\ccmarkart`, for example:

```
\crefname{theorem}{\ccmarkart{le} théorème}{\ccmarkart{les} théorèmes}
```

/ 3 /

Example

Let us come back to the example at the beginning, now you can do this:

```
\crefname{theorem}{\ccmarkart{le} théorème}
                  {\ccmarkart{les} théorèmes}
\crefname{proposition}{\ccmarkart{la} proposition}
                      {\ccmarkart{les} propositions}
```

And the sentence shall be written as:

```
On peut le déduire \ccref[de]{thm1,thm2,prop3}.
```

which would result in (provided that you have done `\selectlanguage{french}`):

> On peut le déduire **des** théorèmes 1 et 2 et **de la** proposition 3.

Voilà !

/ 4 /

Known issues

- ccref currently only works for French, Italian, Portuguese (both European and Brazilian) and Spanish, certainly more would be added to this list.
- The current mechanism does not work for German. However, the author has planed to adopt a more refined approach in later versions in order to support the various situations in German.
- In case that the initial letter of ⟨*prep*⟩ is capitalized, ccref cannot yet handle the case changes automatically. However, this should be a rare occurrence.
- The names of theorem-like environments are not provided — for the moment you need to define them all by yourself. However, users are encouraged to use the ProjLib toolkit, which already handles everything for you.