crefthe

Cross referencing with proper definite articles

Jinwen XU

July 2023, in Beijing

Abstract

The package crefthe provides a command `\crefthe` 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 — that is when `\crefthe` comes into play.

/ 2 / The usage

2.1 | How to load it ?

Simply load the package with:

```
\usepackage{crefthe}
```

Corresponding to: crefthe 2023/07/20

Tip

- Since crefthe 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, `\crefthe` 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 ?

Before everything, you need to define the names, which can be done with `\crefthename`. Its syntax is similar to `\crefname`, but now you can specify the definite articles, for example:

```
\crefthename{theorem}[le]{théorème}[les]{théorèmes}
```

Tip

The `\crefthename`s should be placed in your preamble, otherwise the `\cref` formats will not be set. These names can, however, be reset within the document body.

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

`\crefthe[⟨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).

`\crefthe-[⟨prep⟩]{⟨labels⟩}` and `\crefthe+[⟨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 `\crefthe*` for generating the same referencing text without creating hyperlinks.
- The name-only relatives are also available: `\namecrefthe` and `\namecrefsthe`.

/ 3 / Example

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

```
\crefthename{theorem}[le]{théorème}[les]{théorèmes}
\crefthename{proposition}[la]{proposition}[les]{propositions}
```

And the sentence shall be written as:

```
On peut le déduire \crefthe[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 /
What about German

In German, there are four declensions: nominative (`Nominativ`), genitive (`Genitiv`), dative (`Dativ`) and accusative (`Akkusativ`), making the previously presented mechanism insufficient. For such situation, we introduce the command `\crefthevariantname` to specify the referencing name for the correspond environment. Below is an example of usage:

```
\crefthevariantname{theorem}
  {
    {Satz}{Sätze}
    , Nominativ = [der]{Satz}[die]{Sätze}
    , Genitiv  = [des]{Satzes}[der]{Sätze}
    , Dativ    = [dem]{Satz}[den]{Sätzen}
    , Akkusativ = [den]{Satz}[die]{Sätze}
  }
```

The first line in the configuration is the default set of names when no variant is specified. It is recommended, though not required.

After this, you may refer to a theorem via

```
\crefthe[⟨prep⟩,variant=⟨declension⟩]{⟨label⟩}
```

You may also use the shortcuts (`nom.`, `gen.`, `dat.` and `akk.`), such as:

```
\crefthe[⟨prep⟩,Nom]{⟨label⟩}   \crefthe[⟨prep⟩,Nom.]{⟨label⟩}
\crefthe[⟨prep⟩,nom]{⟨label⟩}   \crefthe[⟨prep⟩,nom.]{⟨label⟩}
```

These four are all equivalent and you may choose one to use according to your preference.

/ 5 /
Regarding the upper and lower cases

As in cleveref, these commands have corresponding uppercased version: `\Crefthename`, `\Crefthe`, `\nameCrefthe` and `\nameCrefsthe`, similar to `\Crefname`, `\Cref`, `\nameCref` and `\nameCrefs`, reserved for using at the beginning of a sentence. `\Crefthe` (and the name-only relatives) can handle case changing automatically: for example, with `\Crefthe [À]{thm1,thm2,prop3}`, you will get something like

> **Aux** théorèmes 1 et 2 et **à la** proposition 3

Of course you will have to define the `\Crefthename`s separately, for example as:

```
\Crefthename{theorem}[Le]{théorème}[Les]{théorèmes}
\Crefthename{proposition}[La]{proposition}[Les]{propositions}
```

/ 6 /
For writing multi-language documents

To place hyperlinks at the correct place, `\crefthename` touches the corresponding format macro `\crefformat` internally, which makes the format language-dependent. If you are writing multi-language documents, you may consider putting `\crefthename` inside your language configuration so as to reset it each time you select a new language.

/ 7 /
Dealing with existed `cref` / `Cref` names

For those preset names defined with `\crefname` / `\Crefname`, such as the ones for `chapter`, `section` and `subsection`, etc., the optional argument of `\crefthe` / `\Crefthe` shall be useless: names defined with `\crefthename` / `\Crefthename` have a `\crefthemark` marking the definite articles, which also deals with the given preposition ; without `\crefthemark`, the preposition is simply ignored.

`\crefthepatchname{`⟨*counters*⟩`}` is provided for this purpose. With this, an empty mark shall be added before the existed singular and plural names. However, it is still recommended to simply redefine the names with `\crefthename` / `\Crefthename`, especially in multilingual documents.

/ 8 /
The relationship with cleveref

crefthe loads cleveref automatically and passes related options to it. All its commands, used without optional arguments, degenerate to those in cleveref. For example, `\crefthe{...}` will produce the same result as `\cref{...}`, and `\crefthename` shall behave the same as `\crefname` if the definite articles are not specified. That said, you may safely use the command `\crefthe` everywhere in your document without causing extra trouble.

With the package option `overwrite`, user commands in cleveref will be replaced by those offered here, thus you can simply write `\cref` for `\crefthe` — and similarly with `\Cref`, `\crefname` and `\Crefname`.

/ 9 /
Known issues

- crefthe currently works for French, German, Italian, Portuguese (European and Brazilian) and Spanish — certainly more languages would be added to this list in the future.
- For now, you cannot specify the masculine/masculine/... of words. This means, for instance, that you cannot make the suffix adjective automatically adopted to the word when using `\namecref` — as an example, if you wish to say “la proposition suivante” (in French), you would have to write the feminine adjective manually. The situation should be improved with the introduction of a new key-value configuration system, which is currently on the todo-list. Meanwhile, you may consider the package zref-clever, which has a much more powerful and sophisticated interface for configuring cross referencing.
- The names of theorem-like environments are not provided here, you need to define them by yourself. However, users are encouraged to use the ProjLib toolkit (more specifically, the internal package create-theorem), which already handles everything for you.

If you run into any issues or have ideas for improvement, feel free to discuss on:

`https://github.com/Jinwen-XU/crefthe/issues`

or email me via `ProjLib@outlook.com`.