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, German, 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}
```

> **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 !

EXAMPLE | 2

For language with declensions

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}
  }
```

> **TIP**
> - The first line in the configuration is the default set of names when no variant is specified. It is recommended, though not required.
> - There is also an equivalent but shorter name `\setcrefname` for this command.

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.

Article contractions are by default enabled for German, but if you don't want this, you may use the package option `disable german article contractions`.

Using abbreviation form

There is another usage of `\crefthevariantname` (or equivalently, `\setcrefname`): setting the abbreviation form. This allows you to use both the full form and the abbreviation form within the same document. For example, you may set

```
\setcrefname{theorem}
  {
    {theorem}{theorems}
    , abbr. = {th.}{ths.}
  }
```

Then, when referencing, you can use the command option `abbr.` to enable abbreviation:

```
\crefthe[abbr.]{⟨label⟩}
```

> **ATTENTION**
>
> Please note that `abbr.` has more priority than the option(s) for specifying declensions. This means that when `abbr.` is enabled, the declensions are simply ignored.

/ 6 /
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.

For languages with declensions, such as German, there is also a `\Crefthevariantname` (or equivalently, `\setCrefname`), corresponding to the above `\crefthevariantname`.

`\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 would have to define the `\Crefthename`s separately, for example as:

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

/ 7 /
On the coordination of gender and number

In many languages, adjectives and verbs need to be properly inflected for the gender and singular or plural of the noun. Take French as example, you may see the following expressions:

- La proposition précédent**e**/suivant**e**... (The previous/next proposition...)
- La proposition est illustré**e** par... (The proposition is illustrated by...)

Thus if you later decide to change this proposition (feminine) into a theorem (masculine), you would have to modify the adjectives and verbs accordingly. Similar things happens in many other languages, and it is not quite realistic to fully automate such modifications.

The name-only relatives `\namecrefthe`, `\nameCrefthe` etc. are designed for this purpose. They do not automatically solve these kinds of problems, but you may later search for the ⟨*label*⟩ of the environment that changes, and modify the surrounding context manually — in other words, these commands serve as recorders: if you write the word like "theorem" or "proposition" directly in the text, then later it would be unclear which part of your text needs to be update when the environment itself changes.

There is another case. For example, if you wish to say "ce théorème" ("this theorem"), the pre-defined set of names that contains definite articles would be somewhat redundant. In such situation, you may use the command option `noun` to show only the noun:

```
\namecref[noun]{⟨label⟩}
```

But then of course you shall need to write the word "Ce/Cet/Cette" (these are all different forms of "This" in French) by yourself — and if you later decide to change the type of the environment to which this ⟨*label*⟩ refers, don't forget to change this word accordingly.

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.

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.

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`.

Acknowledgement

There are so many people I wish to express my gratitude to during the development of this package, including:

- Patrick Bideault, without whose advice and careful explanation, this package would probably never have been born. Merci !
- Enrico Gregorio, Phelype Oleinik and Joseph Wright (in alphabetical order). They made valuable suggestions in the early development of this package and even edited some of the code themselves.
- Jonathan P. Spratte, without whose help the German support would probably take much longer to complete.

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.
- 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.