A (basic) command-redefinition package for SILE

The **redefine** package can be used to easily redefine a command under a new name.

Sometimes one may want to redefine a command (e.g. a font switching hook for some other command, etc.), but would also want to restore the initial command definition afterwards at some point, or to invoke the original definition from the newly redefined one.

This package is just some sort of quick "hack" in order to do it in an easy way. It's far from perfect, it likely has implications if users starts saving and restoring commands in a disordered way, but it can do the magic in some fairly reasonable symmetric cases.

The first syntax below allows one to change the definition of command ⟨*name*⟩ to new ⟨*content*⟩, but saving the previous definition to ⟨*saved-name*⟩:

\redefine[command=⟨*name*⟩**, as=**⟨*saved-name*⟩**]{**⟨*content*⟩**}**

From now on, invoking \⟨*name*⟩ will result in the new definition to be applied, while \⟨*save-name*⟩ will invoke the previous definition, whatever it was.

Of course, be sure to use a unique save name – otherwise, if overwriting an existing command, you will get a warning, at your own risks...

If invoked without ⟨*content*⟩, the redefinition will just define an alias to the current command:

\redefine[command=⟨*name*⟩**, as=**⟨*saved-name*⟩**]**

The following syntax allows one to restore command ⟨*name*⟩ to whatever was saved in ⟨*saved-name*⟩, and to clear the latter:

\redefine[command=⟨*name*⟩**, from=**⟨*saved-name*⟩**]**

So now on \⟨*name*⟩ is restored to whatever was saved, and \⟨*saved-name*⟩ is no longer defined. Again, if the saved name corresponds to some existing command in a broader scope, things may break.