2.4.4 Combined queries

As shown above, you can query for individual tokens. But what if you need more detail, maybe from different layers? Let us consider the following examples:

1. The Italian token *io* used by only male informants
2. Different spelling forms of the German token *was* used as a relative pronoun
3. The French token *est-ce* followed by *que*

Before looking at how to build those queries, let us describe their structure:

1. The first example describes a specific value for a token that is embedded in a message that fulfills specific criteria. We thus have two conditions on different layers (token and message).
2. The second example describes a series of spelling variants as they are found on the token level. The same token has to have a specific value as a normalisation and a specific part of speech. We thus look only at tokens but with two different attributes (annotations for lemma and PoS).
3. In the third example we look only at the layer of tokens, but we look at two different tokens that are in a specific relationship to one another, i.e. one is following the other directly.

The queries for these examples are the following:

1. To find *io* written by males, we query for: `tok="io"& gender = "m"& #2_i_#1`. That reads as: a token with the contents *io* and the gender *m*. The second attribute has to include the first (**_i_**).
2. In the second example we are looking for different spelling variants for the standard spelling *was*. There are two attributes that can be used to this aim. Manually processed messages have the standard spelling in the layer *gloss*, automatically processed messages use the TreeTagger annotation *tt_lem* for lemma. If you are more interested in precision (i.e. if you only want to find samples that are most likely correctly annotated), your best guess is to use the annotation *gloss*. If, on the other hand, you focus on recall (ie. you want as many results as possible even if there might be some wrong annotations) you better use *tt_lem*. The according query would then look like: `tt_lem=/was/ & tt_pos="PRELS" & #1_=_#2`. If we look at that in detail, we see the lemma *was* as a first attribute and the annotation for the relative pronoun as a second. Those two annotatations are on the same level and have to cover the same token (**_=_**).
3. In the third example we look for two tokens, one directly following the other. Here, we could use one of the normalisations, too, i.e. `mftb_lem` (the tagger used for French) or we could use the token. This choice depends on what we want to find. If we are after the spelling *est-ce que* used by the informant, we query for `tok=/…/`. If, on the other hand, we want to include unconventional spellings like *sq*, we have to use `mftb_lem=/…/`. Let us use the first option, which gives us the following query: `tok="est-ce" & tok="que" & #1 . #2`, which we can read as: a first token *est-ce* and a second token *que*. The expression `#1 . #2` means the first token has to directly precede the second one.

That much for the examples. But how can you remember all of these options? You do not have to, since ANNIS offers you lots of support in creation the queries.