SBVR to NL translation pattern

Processing SBVR to Natural language:

Step 1:
The SBVR docuement is read and transformed into a dictionary. Here each concept is used as a key and the value is the list of rules that apply to the concept.

Step 2:
Baed on mututal references and the occurance of multiple concepts withtin 1 rule, concepts are devided into groups. These groups combine concepts that lie close together. All concepts are part of at least 1 group, but can also be part of multiple groups.

Step 3:
We order the groups, using a simple greedy algorithm, so that consecutive groups contain a common element, if possible.

Step 4:
For each group a paragraph is written according to the transformation patterns described in the next section "Processing a group into a paragraph".

Step 5:
A text file is created for the output and each paragraph is written to the file.

Processing a group into a paragraph:

Step 1:
The firtst step is to loop over every concept in the group.

Step 2:
For every concept, we loop over every rule. Here we filter out the rules that contain concepts that are not part of the group. The remaining rules are processed further.

Step 3:
For each rule, we find out the type of rule, the indentifier, (eg. 'is component of', 'is of type', ..., etc.) and the target concepts of the rule. Based on the identifier and the target we distinquish between 3 different types of rules: normal rules, generals of a generalization set, specifics of a generalization set.

Step 4:
Process each rule based on the type of rule

> Step 4.1:
> Process the normal rules
>
> > Step 4.1.1:
> > The identifier is rewritten, to support use in a more natural scentence, in the following ways:

Original notation	New notation

Genral concept: {concept}	Is a(c) {concept}
Necessity: .. requires {c + c}	Needs {c + c}
Necessity: .. is a component of {c+c}	Is a component of {c+c}
Possibility: .. is a component of {c+c}	Can be a component of {c+c}
Necessity: .. is composed of {c+c}	Is composed of {c+c}
Possibility: .. is composed of {c+c}	Can be composed of {c+}
Necessity: .. characterizes {c + c}	Characterizes {c + c}
Possibility: .. characterizes {c + c}	Can characterize {c + c}
Necessity: .. is characterized by {c + c}	Is characterized by {c + c}
Possibility: .. is characterized by {c + c}	Can be characterized by {c + c}
Necessity: .. is a member of {c + c}	Is a member of {c + c}
Possibility: .. is a member of {c + c}	Can be a member of {c + c}
Necessity: .. has as members {c + c}	Has as members {c + c}
Possibility: .. has as members {c + c}	Can have as members {c + c}
Necessity: .. is required by {c + c}	Is required by {c + c}
Possibility: .. is required by {c + c}	Can be required by {c + c}
Necessity: .. is associated to {c + c}	Is associated to {c + c}
Possibility: .. is associated to {c + c}	Can be associated to {c + c}
Necessity: .. is connected to {c + c}	Is connected to {c + c}
Possibility: .. is connected to {c + c}	Can be connected to {c + c}

Step 4.1.2:
A scentence is constructed and added to a list of scentences:
A(n) {concept} {identifier} {target}

Step 4.2:
Process the generals of a generalization set

Step 4.2.1
A scentence is constructed based wheter the set is disjoint and complete. The constructed scentece is then put in the paragraph. The constructed scentence can be deduced from the original scentece and gets translated in the following way:

Original scentence	New scentence
... exactly one ...	Every {concept} is either a(n) {targets}.
... at least one ...	Every {concept} is at least one {targets}, but can also be multiple at the same time.
... at most one ...	A(n) {concept} can be a(n) {targets}, or another possibility, but only 1 at the same time.
... is a ...	A(n) {concept} can be a(n) {targets}, or another possibility, it can also be multiple at the same time.

Step 4.3:
The rules for a specific of a genreralization set are not used in the praragraph.

Step 5:
Scentences constructed in step 4.1 get combined together if they have the same concept and identifier. These new scentences are stored.

Step 6:
Scentences constructe in step 4.1 get combined together if they have the same identifier and targets. These new scentences are stored.

Step 7:
The scentences constructed in steps 5 and 6 are combined with scenteces that remained unchanged in steps 5 and 6. All these scentences are added to the paragraph.