6. SNOMED CT Language Templates

In this section, we explain how the SNOMED CT template syntax can be applied to the computable languages to create template languages. In particular, we combine the Template Syntax with Compositional Grammar to create the Expression Template Language. A similar process can be used to combine the Template Syntax with the Expression Constraint Language to create an Expression Constraint Template Language.

SNOMED CT template languages are created by:

1. Combining the base language (to which the slots are added) with the SNOMED CT template syntax;
2. Adding any additional rules referenced by the SNOMED CT template syntax (e.g. the Expression Constraint Language to represent slot value constraints);
3. Removing any duplicate rules (e.g. rules that are repeated in both the base language and the expression constraint language;
4. Adding references to the template syntax in the appropriate rules of the base language to support the inclusion of slots. This involves:
    a. Renaming the first rule in the base language to add the word "Template" (e.g. from "expression" to "expressionTemplate");
    b. Adding the rule **tokenReplacementSlot** as an alternative wherever a token is referenced (e.g. "definitionStatus / tokenReplacementSlot");
    c. Adding the rules **conceptReplacementSlot** and **expressionReplacementSlot** as alternatives within the **conceptReference** rule;
    d. Adding the rule **concreteValueReplacementSlot** as an alternative attributeValue; and
    e. Adding the rule **templateInformationSlot** before each focus concept, each attribute group and each attribute name value pair.

For an example of how this process is applied to SNOMED CT compositional grammar to create expression templates, please refer to the Expression Template Language syntax in the next section.