

5.1. Normative Specification

The following ABNF definition specifies the SNOMED CT template syntax.

```
templateSlot = templateReplacementSlot / templateInformationSlot

templateReplacementSlot = conceptReplacementSlot / expressionReplacementSlot / tokenReplacementSlot /
concreteValueReplacementSlot

conceptReplacementSlot = "[[" ws "+" ws "id" ws [ "(" ws expressionConstraint ws ")" ws ] slotName ws "]"

expressionReplacementSlot = "[[" ws "+" ws [ "scg" ws ] [ "(" ws expressionConstraint ws ")" ws ] slotName ws "]"

tokenReplacementSlot = "[[" ws "+" ws "tok" ws [ "(" ws slotTokenSet ws ")" ws ] slotName ws "]"

concreteValueReplacementSlot = stringReplacementSlot / integerReplacementSlot / decimalReplacementSlot

stringReplacementSlot = "[[" ws "+" ws "str" ws [ "(" ws slotStringSet ws ")" ws ] slotName ws "]"

integerReplacementSlot = "[[" ws "+" ws "int" ws [ "(" ws slotIntegerSet ws ")" ws ] slotName ws "]"

decimalReplacementSlot = "[[" ws "+" ws "dec" ws [ "(" ws slotDecimalSet ws ")" ws ] slotName ws "]"

slotTokenSet = slotToken *(mws slotToken)

slotStringSet = slotString *(mws slotString)

slotIntegerSet = ( "#" integerValue / slotIntegerRange ) *(mws ( "#" integerValue / slotIntegerRange ))

slotDecimalSet = ( "#" decimalValue / slotDecimalRange ) *(mws ( "#" decimalValue / slotDecimalRange ))

slotIntegerRange = ( slotIntegerMinimum to [ slotIntegerMaximum ] ) / ( to slotIntegerMaximum )

slotIntegerMinimum = [ exclusiveMinimum ] "#" integerValue

slotIntegerMaximum = [ exclusiveMaximum ] "#" integerValue

slotDecimalRange = ( slotDecimalMinimum to [ slotDecimalMaximum ] ) / ( to slotDecimalMaximum )

slotDecimalMinimum = [ exclusiveMinimum ] "#" DecimalValue

slotDecimalMaximum = [ exclusiveMaximum ] "#" DecimalValue

exclusiveMinimum = ">"

exclusiveMaximum = "<"

slotName = "@" ( nonQuoteStringValue / slotString )

slotToken = definitionStatus / memberOf / constraintOperator / conjunction / disjunction / exclusion / reverseFlag /
expressionComparisonOperator / numericComparisonOperator / stringComparisonOperator

slotString = QM stringValue QM

nonQuoteStringValue = *(%x21 / %x23-26 / %x28-3F / %x41-5A / %x5C / %x5E-7E) ; string with no ws, quotes, at or
square brackets

templateInformationSlot = "[[" ws slotInformation ws "]"

slotInformation = [ cardinality ws ] slotName ws ]
```

Please note that some of the rules referenced above are defined elsewhere. In particular, *definitionStatus* is defined in [SNOMED CT compositional grammar](#), and *expressionConstraint*, *cardinality*, *to*, *QM*, *stringValue*, *integerValue*, *decimalValue*, *ws*, *mws*, *memberOf*, *constraintOperator*, *conjunction*, *disjunction*, *exclusion*, *reverseFlag*, *expressionComparisonOperator*, *numericComparisonOperator* and *stringComparisonOperator* are all defined in the [expression constraint language](#) syntax. These rule definitions must therefore be combined with the above syntax before it can be used.