Expression Constraint Language - Specification and Guide

Exported on 11/22/2023
Table of Contents

1  1. Introduction......................................................................................................................... 8
  1.0.1 Background ......................................................................................................................... 8
  1.0.2 Purpose ............................................................................................................................... 8
  1.0.3 Scope .................................................................................................................................... 8
  1.0.4 History .................................................................................................................................. 9
  1.0.5 Audience ............................................................................................................................. 10
  1.0.6 Document Overview .......................................................................................................... 10

2  2. Use Cases .................................................................................................................................. 12
  2.1 2.1 Terminology Binding ........................................................................................................ 12
  2.2 2.2 Intensional Reference Set Definitions ............................................................................ 12
  2.3 2.3 SNOMED CT Content Queries ........................................................................................ 13
  2.4 2.4 SNOMED CT Concept Model .......................................................................................... 13

3  3. Requirements .......................................................................................................................... 14
  3.1 3.1 General SNOMED CT Language Requirements ............................................................. 14
  3.2 3.2 Expression Constraint and Query Requirements ........................................................... 14
  3.3 3.3 Concept Model Requirements ......................................................................................... 17

4  4. Logical Model .......................................................................................................................... 18
  4.1 4.1 Details ................................................................................................................................ 19

5  5. Syntax Specification ................................................................................................................ 21
  5.1 5.1 Brief Syntax (Normative) ................................................................................................ 21
  5.2 5.2 Long Syntax (Informative) ............................................................................................... 26
  5.3 5.3 Informative Comments ..................................................................................................... 32
  5.4 5.4 Order of Operation ........................................................................................................... 56
    5.4.1 Unary Operators ................................................................................................................ 56
    5.4.2 Binary Operators .............................................................................................................. 56
    5.4.3 Filter Constraints .............................................................................................................. 58
    5.4.4 History Supplements ....................................................................................................... 59
  5.5 5.5 Character Collation for Term Filters .............................................................................. 60

6  6. Examples .................................................................................................................................... 64
  6.1 6.1 Simple Expression Constraints ....................................................................................... 64
Expression Constraint Language – Expression Constraint Language - Specification and Guide

6.4.2 Attribute Conjunction and Disjunction ................................................................. 93
6.4.3 Attribute Group Conjunction and Disjunction ....................................................... 95
6.4.4 Attribute Value Conjunction and Disjunction ....................................................... 96
6.5 6.5 Exclusion and Not Equals ...................................................................................... 97
   6.5.1 Exclusion of Simple Expressions ........................................................................ 97
   6.5.2 Exclusion of Attribute Values ............................................................................ 97
   6.5.3 Not Equal to Attribute Value ............................................................................. 98
6.6 6.6 Constraint Comments ......................................................................................... 99
   6.6.1 Comments .......................................................................................................... 99
6.7 6.7 Nested Expression Constraints ........................................................................ 100
   6.7.1 Constraint Operators ....................................................................................... 100
   6.7.2 MemberOf Function ......................................................................................... 101
   6.7.3 Compound Expression Constraints .................................................................. 102
   6.7.4 Dotted Attributes ............................................................................................ 103
   6.7.5 Refinement ....................................................................................................... 104
   6.7.6 Attribute Values ................................................................................................ 105
6.8 6.8 Description Filters ............................................................................................. 105
   6.8.1 Overview ........................................................................................................... 105
   6.8.2 Term Filter ........................................................................................................ 105
   6.8.3 Language Filter ................................................................................................ 107
   6.8.4 Description Type Filter ..................................................................................... 108
   6.8.5 Dialect Filter ..................................................................................................... 109
   6.8.6 Acceptability Filter .......................................................................................... 110
   6.8.7 Filters with Negation ....................................................................................... 111
   6.8.8 Module Filter .................................................................................................... 112
   6.8.9 Effective Time Filter ....................................................................................... 113
   6.8.10 Active Filter .................................................................................................... 115
   6.8.11 Description Id Filter ....................................................................................... 115
6.9 6.9 Concept Filters .................................................................................................. 116
   6.9.1 Overview ........................................................................................................... 116
   6.9.2 Module Filter .................................................................................................... 118
   6.9.3 Effective Time Filter ....................................................................................... 118
   6.9.4 Active Filter ..................................................................................................... 120
6.10 6.10 Member Filters ............................................................................................... 121
The Expression Constraint Language is a formal syntax for representing SNOMED CT expression constraints. Expression constraints are computable rules used to define a bounded sets of clinical meanings represented by either precoordinated or postcoordinated expressions. Expression constraints can be used to restrict the valid values for a data element in an EHR, as the intensional definition of a concept-based reference set, as a machine processable query that identifies a set of matching expressions, or as a constraint that restricts the range of an attribute defined in the SNOMED CT concept model.

This document defines and describes the current version of the Expression Constraint Language - ECL v2.2.
1 1. Introduction

1.0.1 Background

SNOMED CT is a clinical terminology with global scope covering a wide range of clinical specialties and requirements. The use of SNOMED CT expressions in Electronic Health Records (EHRs) provides a standardized way to represent clinical meanings captured by clinicians and enables the automatic interpretation of these meanings. SNOMED CT expressions are a structured combination of one or more concept identifiers used to represent a clinical idea in a logical manner. The SNOMED CT Compositional Grammar\(^3\) provides a lightweight syntax for the representation of SNOMED CT expressions.

In contrast, a SNOMED CT Expression Constraint is a computable rule that can be used to define a bounded set of clinical meanings represented by either precoordinated or postcoordinated expressions. Expression constraints can be used as formal constraints on the content of a particular data element in an EHR, as the intensional definition of a concept-based reference set, as a machine processable query that identifies a set of matching precoordinated or postcoordinated expressions, or as a constraint that restricts the range of an attribute defined in the SNOMED CT concept model.

1.0.2 Purpose

The purpose of this document is to define and describe a formal language for representing SNOMED CT Expression Constraints. A SNOMED CT Expression Constraint is a computable rule that defines a bounded set of clinical meanings represented by either precoordinated or postcoordinated expressions. Two equivalent syntaxes are presented – a brief syntax, which is designed to be as compact as possible for interoperable communication between systems, and a long syntax, which introduces textual alternatives to the symbols from the brief syntax. This document also provides examples and guidance to assist in the implementation of this language.

1.0.3 Scope

This document presents the specification of an Expression Constraint Language, which can be used to represent SNOMED CT Expression Constraints. It includes a logical model of the language, two syntaxes, a set of example expression constraints and a summary of implementation considerations.

The Expression Constraint Language specified in this document is part of a consistent set of computer processable languages designed to support a variety of use cases involving the use of SNOMED CT. Other SNOMED CT computable languages include:

- Compositional Grammar\(^4\): designed to represent SNOMED CT expressions; and
- Template Syntax\(^5\): which allow slots to be added to expressions, expression constraints or queries that can be filled with specific values at a later time.

The compositional grammar is designed to provide a common foundation for the additional functionality added by the other languages.

This document does not include a full description of how to implement an expression constraint parser, classifier or interpreter. It does not describe how to transform an expression constraint into other languages, such as OWL, SPARQL or SQL; or how to determine whether two expression constraints are equivalent. It also does not describe how to implement an EHR which uses expression constraints to constrain or query its content, or a terminology

---

\(^3\) http://snomed.org/scg
\(^4\) http://snomed.org/scg
\(^5\) http://snomed.org/sts
server which uses expression constraints to query its content. Instead, it provides a specification, examples and
general guidance to assist in the implementation of expression constraints in any of these applications.

This document defines and describes the current version of the Expression Constraint Language - ECL 2.1.

1.0.4 History

Expression constraints have been used in projects and programs around the world for a number of years – for
example HL7 TermInfo\(^6\), and the NHS Logical Record Architecture\(^7\).

In 2013, a draft document on "SNOMED CT Expression Constraint Syntax Specification for Terminology Binding"
was developed as an assignment for the SNOMED CT Implementation Advisor (SIA) scheme.

In 2014, this work was revised and extended to support a wider range of relevant use cases to produce version 1.0 of
the Expression Constraint Language specification (2015). These updates included:

- Concrete values (e.g. integers, decimals and strings) are now permitted as attribute values. This is to provide
  alignment with the recent extensions to SNOMED CT Compositional Grammar;
- Cardinality constraints have been introduced, and as a result the optional operator (i.e. ~ ) is no longer
  provided;
- Attributes may now be preceded by a 'descendantOf' or 'descendantOrSelfOf' operator to indicate whether
  attribute descendants and/or the attribute itself should be used in the matching process;
- A reverse flag has been introduced, which allows relationships to be traversed in the reverse direction;
- Exclusion has been changed from a unary operator ('negation') to a binary operator ('minus');
- A wildcard character ('*') has been introduced to represent any concept in the substrate;
- A number of clarifications have been made, including the 'memberOf' operator and the default substrate
  upon which the expression constraints are executed.

An update to the Expression Constraint Language was then published in 2016 (version 1.1) to incorporate some
additional features requested by implementers of the language. These updates include:

- Two new operators 'childOf' and 'parentOf' were added to support querying immediate children and
  immediate parents of a concept during user interface design;
- A new 'dot notation' was introduced (as an alternative to the Reverse flag) to refer to an attribute value for a
  concept or expression;
- The ability for a constraint operator (e.g. 'descendantOf') to be applied to a nested expression constraint
  was added;
- The ability to add comments within the text of an expression constraint was added;
- Additional optional brackets were allowed around subexpressions; and
- The non-normative syntax (previously named the 'Full Syntax') was renamed to the 'Long Syntax'.

Early in 2017 version 1.2 was published, to include a new feature requested by implementers: namely, the ability for
the 'memberOf' function to be applied to a set of reference set concepts defined using an expression constraint. In
this version, the explanation of Operator Precedence was also moved from section 6.7 to section 5.4. Version 1.3 was
then published in mid 2017 to support a range of additional features - including allowing the refinement of
subexpression constraints, permitting the use of subexpression constraints to represent a set of valid attribute
names and simplifying the parsing of dotted expression constraints.

In mid 2020, version 1.4 was published to support boolean attribute values and to introduce the 'childOrSelfOf' and
'parentOrSelfOf' operators. Later that year, version 1.5 was published to support description filter constraints.
These constraints filter the result set, by matching only on concepts which have a description that satisfies the filter
criteria. Section 5.5 (Character Collation for Term Filters) and section 6.8 (Filter Constraints) were added in ECL
version 1.5.

---

\(^6\) http://snomed.org/hl7terminfo
\(^7\) https://isd.hscic.gov.uk/trud3/user/guest/group/0/pack/12
In 2021, version 1.6 added concept filters, which allow the result set to be filtered based on the definition status, module, effectiveTime and active status of each concept.

And then in early 2022, version 2.0 was published. Version 2.0 includes a number of significant features, including:

- History supplements, to supplement the results with relevant inactive concepts,
- Reference set member filters, to filter the rows of a reference set, based on the value of specified fields,
- Support for returning multiple fields of a reference set, including fields other than the referencedComponentId,
- Support for module, effectiveTime and active filters on descriptions, and
- Support for word-prefix-any-order and wildcard searches for string-based concrete attribute values (for consistency with term searches in a Description filter).

Most significantly, version 2.0 is the first version of ECL that is specifically designed to support querying over historical patient records, which may contain inactive codes.

In August 2022, version 2.1 was published to allow description filters to filter results using description identifiers, and to harmonise the dialect alias filter (see Appendix C(see page 193) with BCP-47 (Internet Best Current Practice Specification)8.

In November 2023 version 2.2 was published. This version added the ability to reference concepts using alternate identifiers, and also two convenience methods for finding the top (root) or bottom (leaf) concepts within a set.

For a list of previous PDF versions, please refer to Previous Versions(see page 211).

1.0.5 Audience

The target audiences of this document include:

- SNOMED National Release Centres;
- SNOMED CT designers and developers, including designers and developers of EHR systems, information models, data entry interfaces, storage systems, decision support systems, retrieval and analysis systems, communication standards and terminology services;
- SNOMED CT terminology developers, including concept model designers, content authors, map developers, subset and constraint developers and release process managers.

It should be noted that this document contains both technical and non-technical content. In particular, the detailed logical model and formal syntax is specifically focussed at more technical readers. Less technical readers are encouraged to read the introductory material (including the use cases and requirements) and the extensive set of examples that is presented. It should also be noted that even though complex expression constraints are possible, most expression constraints are likely to be very simple, such as those described in Simple Expression Constraints9.

1.0.6 Document Overview

This document defines the SNOMED CT Expression Constraint Language10 and describes how and where it may be implemented. Chapter 211 begins by describing the use cases in which it is anticipated that SNOMED CT Expression Constraint Language will be used. Chapter 312 then describes the requirements used to guide the definition of this language. In Chapter 413, the logical model of the Expression Constraint Language is presented, while in Chapter 5...
two syntaxes are defined using an ABNF serialisation of the logical model. Chapter 6 then presents some examples of expression constraints that conform to the SNOMED CT Expression Constraint syntaxes, and Chapter 7 discusses some implementation considerations. Appendix A – Examples Of Valid Expressions provides some examples of precoordinated and postcoordinated expressions that satisfy each of the expression constraints presented earlier in the document. Appendix B – Examples Of Invalid Expressions then provides some examples that do not satisfy these expression constraints. Appendix C – Dialect Aliases provides a list of example aliases that may be used to specify a particular dialect in an ECL filter constraint. Appendix D – ECL Quick Reference provides a quick reference to the key syntax features of the Expression Constraint Language. And finally, Appendix E – Reference Set Fields explains how reference set field names are used in ECL 2.0+.

14 https://confluence.ihtsdotools.org/display/WIPECL/SyntaxSpecification
15 https://confluence.ihtsdotools.org/display/WIPECL/Examples
16 https://confluence.ihtsdotools.org/display/WIPECL/ImplementationConsiderations
2 2. Use Cases

The SNOMED CT Expression Constraint Language enables the intensional definition of a bounded set of clinical meanings. This is important for a number of use cases, including:

- Terminology Binding[see page 12];
- Intensional Reference Set Definitions[see page 12];
- SNOMED CT Content Queries[see page 12]; and
- SNOMED CT Concept Model[see page 12].

In the following subsections, we describe each of these key use cases.

2.1 2.1 Terminology Binding

Most Electronic Health Records (EHRs) are designed and developed using one or more information models, which describe the information that is collected, stored, communicated and displayed. Some information models are designed for a specific proprietary system, while others are based on a common health information standard (e.g. HL7 FHIR resource, HL7 CDA template, ISO 13606 archetype). Information models may also be defined using a wide variety of representations (e.g. UML class diagram, database table design, Archetype Definition Language, or XML Schema). Irrespective of the purpose, design and representation of the information models, however, the use of clinical terminology is an important part of making the models complete and useful.

Terminology binding provides the links between the information model and the terminology. These links may be used to constrain the set of possible values which can populate a given coded data element in the information model, or they may define the meaning of an information model artefact using the terminology. Terminology binding is an important part of supporting the following clinical information system functions:

- Data capture;
- Retrieval and querying;
- Information model library management; and
- Semantic interoperability.

To enable terminology binding to be defined using intensional rules, a formal language must be used. The SNOMED CT Expression Constraint Language\(^\text{[17]}\) can be used in this way to define terminology bindings which constrain the set of possible coded values within an information model.

2.2 2.2 Intensional Reference Set Definitions

Reference sets are a flexible, extensible SNOMED CT file structure used to support a variety of requirements for the customization and enhancement of SNOMED CT content. These include the representation of subsets, language preferences, or maps to/from other code systems.

Some reference sets (using the Query Specification type) allow a serialised query to represent the membership of a subset of SNOMED CT components. A query contained in this reference set is executed against the content of SNOMED CT to produce a subset of concepts, descriptions or relationships. This query is referred to as an intensional definition of the subset. It can be run against future releases of SNOMED CT to generate a potentially different set of subset members. The members of the resulting subset may also be represented in an enumerated form as a Simple Reference Set. An enumerated representation of a subset is referred to as an extensional definition.

\(^{17}\)http://snomed.org/ecl
The SNOMED CT Expression Constraint Language\(^\text{18}\) can be used in this way to represent the intensional definition of a subset of SNOMED CT concepts that can be enumerated as a Simple Reference Set.

### 2.3 SNOMED CT Content Queries

SNOMED CT provides both hierarchies and formal concept definitions to allow a range of advanced query techniques. SNOMED CT queries can be performed over different sets of terminology artefacts (known as the substrate of the query), including:

- The precoordinated components distributed as part of the SNOMED CT international edition;
- The precoordinated components distributed by a local release centre as part of a national or local SNOMED CT edition;
- The postcoordinated expressions stored within an expression repository; or
- The SNOMED CT expressions stored within an Electronic Health Record (EHR).

The SNOMED CT Expression Constraint Language\(^\text{19}\) enables queries over SNOMED CT content to be expressed. These queries may be performed for a range of purposes, including the authoring and quality assurance of new SNOMED CT content, the design and development of extensional reference sets, and the design and display of SNOMED CT subsets in clinical user interfaces. While the language itself does not support querying over the full EHR content, the SNOMED CT Expression Constraint Language\(^\text{20}\) could be embedded within record-based query languages (such as SQL) to represent the terminological aspects of these queries.

### 2.4 SNOMED CT Concept Model

The SNOMED CT Concept Model is the set of rules that determines the permitted sets of attributes and values that may be applied to particular types of concepts. There are also additional rules on the cardinality and grouping of each type of attribute. The SNOMED CT Concept Model includes the definition of the domain and range of each attribute. The domain is the set of concepts which are permitted to be used as the source of the attribute, while the range is the set of concepts which are permitted to be used as the target of the attribute. For example, the domain of the attribute 363698007 | Finding site\(^\text{21}\) is the descendants and self of 404684003 | Clinical finding\(^\text{22}\), while the range is the descendants and self of 442083009 | Anatomical or acquired body structure\(^\text{23}\). The SNOMED CT Concept Model rules are represented in a computable form in the SNOMED CT Machine Readable Concept Model\(^\text{24}\).

---

\(^\text{18}\) http://snomed.org/ecl
\(^\text{19}\) http://snomed.org/ecl
\(^\text{20}\) http://snomed.org/ecl
\(^\text{21}\) http://snomed.info/id/363698007
\(^\text{22}\) http://snomed.info/id/404684003
\(^\text{23}\) http://snomed.info/id/442083009
\(^\text{24}\) http://snomed.org/mrcm
3 3. Requirements

In this chapter, we state the requirements of the SNOMED CT Expression Constraint Language\(^{25}\). These requirements are grouped into General SNOMED CT Language Requirements (see page 14) (which are shared by all SNOMED CT computable languages), Expression Constraint and Query Requirements (see page 14), and Concept Model Requirements (see page 17).

3.1 3.1 General SNOMED CT Language Requirements

The general SNOMED CT language requirements include:

**Requirement G.1**: Backward compatibility

The language must be backwardly compatible with any version of the language that has previously been adopted as an SNOMED International standard.

**Requirement G.2**: Consistency

Each logical feature of the language should have a single, consistent meaning across all the languages in the SNOMED CT family of languages. Each logical feature should also have a consistent set of syntax representations.

**Requirement G.3**: Sufficient and necessary

Each language must be sufficiently expressive to meet the requirements of the use cases for which it was designed. However, functionality without a corresponding use case will not be included, as this increases the complexity of implementation unnecessarily.

**Requirement G.4**: Machine processability

In order to facilitate the easy adoption by technical audiences, instances of each language must be able to be parsed into a logical representation using a machine processable syntax specification. This requirement will be met by defining the language syntax in ABNF.

**Requirement G.5**: Human readability

Non-technical stakeholders require that the language is as human readable as possible, while still meeting the other requirements. This is essential for both the clinical validation of expressions, as well as for the education and training required to author expressions.

3.2 3.2 Expression Constraint and Query Requirements

The general expression constraint language requirements include:

**Requirement E.1**: Able to be evaluated against SNOMED CT content

Expression constraints must be able to be evaluated against a specific set of SNOMED CT content (referred to as the substrate). When evaluated against a finite set of precoordinated concepts or postcoordinated SNOMED CT expressions, a finite subset of the substrate can be found which satisfies the expression constraint.

Please note that the substrate over which the expression constraint is evaluated is not explicitly defined within the expression constraint, and must therefore be established by some other means. By default, the assumed substrate is the set of active components from the snapshot release (in distribution normal form) of the SNOMED CT versioned edition currently loaded into the given tool.

**Requirement E.2**: Expression constraint functional requirements

25 http://snomed.org/ecl
The expression constraint language must support the following capabilities:

<table>
<thead>
<tr>
<th>Function</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept reference</td>
<td>The ability to reference a precoordinated SNOMED CT concept using its identifier and optional human-readable term.</td>
</tr>
<tr>
<td>Concept hierarchy</td>
<td>The ability to refer to a set of concepts which is exactly equal to the descendants, descendants and self, ancestors, or ancestors and self of a given concept.</td>
</tr>
<tr>
<td>Immediate children and parents</td>
<td>The ability to refer to a set of concepts which are either immediate children or immediate parents of a given concept (based on non-redundant relationships) (with or without the given concept itself).</td>
</tr>
<tr>
<td>Conjunction</td>
<td>The ability to connect two expression constraints, attribute groups or attribute sets via a logical AND operator.</td>
</tr>
<tr>
<td>Disjunction</td>
<td>The ability to connect two expression constraints, attribute groups or attribute sets via a logical OR operator.</td>
</tr>
<tr>
<td>Refinement</td>
<td>The ability to refine (or specialize) the meaning of an expression constraint using one or more attributes values.</td>
</tr>
<tr>
<td>Reverse</td>
<td>The ability to constrain the source concepts of a set of relationships, and refer to the destination concepts of these relationships.</td>
</tr>
<tr>
<td>Dotted attribute</td>
<td>The ability to refer to the value (or set of values) of an attribute that is included in the definition of a set of concepts.</td>
</tr>
<tr>
<td>Attribute group</td>
<td>The ability to group a collection of attributes which operate together as part of a refinement.</td>
</tr>
<tr>
<td>Attribute</td>
<td>The ability to specify an attribute name-value pair which further refines the meaning of the matching expressions.</td>
</tr>
<tr>
<td>Attribute descendants</td>
<td>The ability to define an attribute which may apply to either the descendants of the given attribute name, or the descendants and self of the given attribute name.</td>
</tr>
</tbody>
</table>

---

26 http://snomed.info/id/116680003
<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nesting</td>
<td>The ability to use an expression constraint to represent the valid set of attribute names and/or attribute values.</td>
</tr>
<tr>
<td>Concrete values</td>
<td>The ability to use integers, decimals, strings and booleans as attribute values.</td>
</tr>
<tr>
<td>Concrete value comparison</td>
<td>The ability to compare the attribute value of the matching expressions with the attribute value in the expression constraint using mathematical comparison operators (e.g. =, &lt;, &gt;, &lt;=, &gt;=, !=).</td>
</tr>
<tr>
<td>Member of</td>
<td>The ability to refer to a set of concepts that are referenced by members of a reference set (or set of reference sets).</td>
</tr>
<tr>
<td>Reference set field value selection</td>
<td>The ability to return the value of any non-metadata field of a reference set.</td>
</tr>
<tr>
<td>Exclusion</td>
<td>The ability to filter out a set of expressions from the result, by either removing expressions whose focus concept is in a specific set, or removing expressions whose attribute value matches a given value.</td>
</tr>
<tr>
<td>Any</td>
<td>The ability to refer to any concept in the substrate, without relying on the availability of a single root concept.</td>
</tr>
<tr>
<td>Description filter</td>
<td>The ability to filter the result set, based on the properties of each concept's descriptions. Expression constraints should be able to filter the concepts based on whether or not it has a description with a matching term, type, language, membership of a language reference set, and acceptability within that language reference set. Term matching approaches should include wildcard and word-prefix-any-order. Expression constraints should also be able to filter concepts based on the module, effectiveTime, active status and identifier of their descriptions.</td>
</tr>
<tr>
<td>Concept filter</td>
<td>The ability to filter the result set, based on the properties of each concept. Expression constraints should be able to restrict the definition status, module, effectiveTime and active status of matching concepts.</td>
</tr>
<tr>
<td>Member filter</td>
<td>The ability to filter rows of a reference set member, based on the value of specified fields.</td>
</tr>
<tr>
<td>History supplements</td>
<td>The ability to include inactive concepts that are associated with any active concept in a given result set, via an historical association reference set.</td>
</tr>
</tbody>
</table>
3.3 Concept Model Requirements

The SNOMED CT concept model requirements include:

**Requirement C.1:** The ability to express SNOMED CT concept model constraints

The language must support the ability to express SNOMED CT concept model constraints, such that the resulting expression constraint can be used to validate SNOMED CT concept definitions and postcoordinated expressions.

In particular, the language must support the ability to define the domain and cardinality of each attribute in the SNOMED CT concept model, and the range of all concept model **object** attributes (whose range is a set of SNOMED CT concepts). The domain of an attribute is the set of valid source concepts of relationships of that type. In most cases, this will be defined as the descendants and self of a given concept. The range of a concept model object attribute is the set of valid destination concepts of relationships of that type. This will be defined as the set of concepts that match a given expression constraint. The cardinality of an attribute constrains the number of times an active relationship of this type can be added to a concept in the SNOMED CT snapshot release (in necessary normal form). For more information about the SNOMED CT necessary normal form, please refer to 2.5. Generating Necessary Normal Form in the SNOMED CT OWL Guide (http://snomed.org/owl).

Please note that the range of a concept model **data** attribute (whose value is concrete) will be specified using a value list constraint from the SNOMED CT Template Syntax (http://snomed.org/sts).

---

27 https://confluence.ihtsdotools.org/display/WIPOWL/2.5.+Generating+Necessary+Normal+Form+Relationships+from+the+OWL+Refsets
28 https://confluence.ihtsdotools.org/display/DOCSTS/8.3.+Constrained+Replacement+Slots
4. Logical Model

A SNOMED CT Expression Constraint contains either a single focus concept, or a series of focus concepts joined by either conjunction, disjunction or exclusion. Each focus concept in an Expression Constraint is either a concept reference or a wildcard, and is normally preceded by either a constraint operator or a memberOf function. An Expression Constraint may also contain a refinement, which consists of grouped or ungrouped attributes (or both). Each attribute consists of the attribute name (optionally preceded by a cardinality, reverse flag and/or attribute operator) together with the value of the attribute. The attribute name is either a concept reference or a wild card. The attribute value is either an expression constraint or a concrete value (i.e. string, integer, decimal or boolean). Conjunction or disjunction can be applied at a variety of levels, including between expression constraints, refinements, attribute groups, and attributes. An expression constraint can also be followed by a dot and attribute name pair. One or more description filters may be applied to an expression constraint, which can include description identifier, module, effective time, active status, term, language, type, dialect and acceptability criteria. Similarly, one or more concept filters may be applied to an expression constraint, which can include definition status, module, effective time and active status criteria. Member filters may be applied to results of the memberOf function, and may include module, effective time, active status and specific refset field criteria. Finally, history supplements may be applied, which include an ECL query to specify the set of historical association reference sets to be used.

Figure 1 below illustrates the overall structure of an expression constraint using an abstract representation. Those parts of an expression constraint, which are in common with SNOMED CT Compositional Grammar expressions, are shown with dotted lines to emphasise the new features (using solid lines) in the Expression Constraint Language. Please note that no specific semantics should be attributed to each arrow in this abstract diagram.

Figure 2 below shows an example of an expression constraint with the main components marked. These components will be explained further in the subsequent sections of this document.

---

29 http://snomed.org/scg
30 http://snomed.org/ecl
The expression constraint in Figure 2 is satisfied by concepts which are clinical findings and members of the cardiology reference set and have an attribute group that either has an associated morphology of infarct (or descendant) or are associated with myocardial infarction (or descendant). In addition, all matching concepts must also have a description that matches the term "card", has a language of English, has a type of | Synonym and are preferred in the en-us language reference set. And matching concepts must be primitive, belong to the international core module, be published on or before 31st July 2021, and be active. The results of this expression constraint are then supplemented by any inactive concept that is associated with the active results via an historical association reference set.

4.1 Details

Figure 3 below provides a non-normative representation of the logical model of the SNOMED CT Expression Constraint Language using a UML class diagram. Please note that each of the classes in this diagram corresponds to a rule in the syntax specification defined in Chapter 5(see page 21). For a short description of each of these, please refer to Section 5.4(see page 32).
Figure 3: Logical Model of Expression Constraint Language
5 5. Syntax Specification

The following sections describe two syntaxes for use with the SNOMED CT Expression Constraint Language. These syntaxes are serialised representations of the logical model presented in the previous chapter, and are therefore logically equivalent.

The first of these syntaxes is referred to as the 'brief syntax' as it primarily uses a symbolic representation aimed to be as compact as possible. This syntax is considered to be the normative syntax, and is recommended for use in interoperable communications between systems.

The second syntax is referred to as the 'long syntax'. The long syntax introduces English-based textual alternatives to the symbols defined in the 'brief syntax', with the aim of increasing the human readability of the language. The textual alternatives provided in the 'long syntax' may (in theory) be translated into other languages to provide equivalent expression constraint representations that are human-readable by non-English speakers. Please note that the 'long syntax' (and any translations) is non-normative, and should only be used when a reliable mapping to the normative brief syntax is possible.

Please note that by default each expression constraint is evaluated against only the active components (and active members of each reference set) from the snapshot release (in distribution normal form) of a specified SNOMED CT versioned edition.

- 5.1 Brief Syntax (Normative)(see page 21)
- 5.2 Long Syntax (Informative)(see page 26)
- 5.3 Informative Comments(see page 32)
- 5.4 Order of Operation(see page 56)
- 5.5 Character Collation for Term Filters(see page 60)

5.1 5.1 Brief Syntax (Normative)

The following ABNF definition specifies the Brief Syntax of the SNOMED CT Expression Constraint Language. This ABNF syntax and the ANTLR syntax is maintained in the SNOMED Expression Constraint Language GitHub\(^{33}\) repository.

```
expressionConstraint = ws ( refinedExpressionConstraint / compoundExpressionConstraint / dottedExpressionConstraint / subExpressionConstraint ) ws
refinedExpressionConstraint = subExpressionConstraint ws ":" ws eclRefinement
compoundExpressionConstraint = conjunctionExpressionConstraint / disjunctionExpressionConstraint / exclusionExpressionConstraint
conjunctionExpressionConstraint = subExpressionConstraint 1*(ws conjunction ws subExpressionConstraint)
disjunctionExpressionConstraint = subExpressionConstraint 1*(ws disjunction ws subExpressionConstraint)
exclusionExpressionConstraint = subExpressionConstraint 1*(ws exclusion ws subExpressionConstraint)
dottedExpressionConstraint = subExpressionConstraint 1*(ws exclusion ws dottedExpressionConstraint)
dottedExpressionAttribute = dot ws eclAttributeName
subExpressionConstraint= [constraintOperator ws] ( ( [memberOf ws] (eclFocusConcept / "(" ws expressionConstraint ws ")") ) *(ws memberFilterConstraint)) / (eclFocusConcept / "(" ws expressionConstraint ws ")") ) *(ws (descriptionFilterConstraint / conceptFilterConstraint)) [ws
```

\(^{33}\)https://github.com/IHTSDO/snomed-expression-constraint-language
historySupplement

eclFocusConcept = eclConceptReference / wildCard / altIdentifier
dot = "."
memberOf = "^" [ ws "[" ws (refsetFieldNameSet / wildCard) ws "]" ]
refsetFieldNameSet = refsetFieldName * (ws "," ws refsetFieldName)
refsetFieldName = 1*alpha
eclConceptReference = conceptId [ ws "|" ws term ws "|""]
eclConceptReferenceSet = "(" ws eclConceptReference 1*(mws
eclConceptReference) ws ")"
conceptId = sctId
term = 1*nonwsNonPipe *( 1*SP 1*nonwsNonPipe )
altIdentifier = (QM altIdentifierSchemeAlias "#" altIdentifierCodeWithinQuotes
QM / altIdentifierSchemeAlias "#" altIdentifierCodeWithoutQuotes) [ ws ")" ws
term ws "]"
altIdentifierSchemeAlias = alpha *(dash / alpha / integerValue)
alIdentifierCodeWithinQuotes = 1*anyNonEscapedChar
altIdentifierCodeWithoutQuotes = 1*(alpha / digit / dash / "." / "/")
wildCard = "*"
constraintOperator = childOf / childOrSelfOf / descendantOrSelfOf /
descendantOf / parentOf / parentOrSelfOf / ancestorOrSelfOf /
top / bottom
descendantOf = "<"
descendantOrSelfOf = "<<"
childOf = "<!"
childOrSelfOf = "<<!"
ancestorOf = ">"
ancestorOrSelfOf = ">>"
parentOf = ">!"
parentOrSelfOf = ">>!"
top = "!!>
bottom = "!!<"
conjunction = (("a"/"A") ("n"/"N") ("d"/"D") mws) / ","
disjunction = ("o"/"O") ("r"/"R") mws
exclusion = ("m"/"M") ("i"/"I") ("n"/"N") ("u"/"U") ("s"/"S") mws
eclRefinement = subRefinement ws [conjunctionRefinementSet /
disjunctionRefinementSet]
conjunctionRefinementSet = 1*(ws conjunction ws subRefinement)
disjunctionRefinementSet = 1*(ws disjunction ws subRefinement)
subRefinement = eclAttributeSet / eclAttributeGroup / "(" ws eclRefinement ws ")"
eclAttributeSet = subAttributeSet ws [conjunctionAttributeSet /
disjunctionAttributeSet]
conjunctionAttributeSet = 1*(ws conjunction ws subAttributeSet)
disjunctionAttributeSet = 1*(ws disjunction ws subAttributeSet)
subAttributeSet = eclAttribute / "(" ws eclAttributeSet ws ")"
eclAttributeGroup = ["[" cardinality "]" ws ] [" ws eclAttributeSet ws "]"
eclAttribute = ["[" cardinality "]" ws ] [reverseFlag ws] eclAttributeName ws
(expressionComparisonOperator ws subExpressionConstraint /
numericComparisonOperator ws ")" numericValue / stringComparisonOperator ws
(typedSearchTerm / typedSearchTermSet) / booleanComparisonOperator ws
booleanValue)
cardinality = minValue to maxValue
minValue = nonNegativeIntegerValue
to = ".."
maxValue = nonNegativeIntegerValue / many
many = "+"
reverseFlag = "R"
eclAttributeName = subExpressionConstraint
equationComparisonOperator = "+=" / "+="
numericComparisonOperator = "+=" / "+=" / "+=" / "+=" / "+=
stringComparisonOperator = "\u2039" / "+="
booleanComparisonOperator = "\u2039" / "+="

idComparisonOperator = null / null
descriptionFilterConstraint = "{{" ws [ "d" / "D" ] ws descriptionFilter * ( (ws "," ws descriptionFilter) ws "}"

descriptionFilter = termFilter / languageFilter / typeFilter / dialectFilter / moduleFilter / effectiveTimeFilter / activeFilter / descriptionIdFilter
descriptionIdFilter = descriptionIdKeyword ws idComparisonOperator ws idComparisonOperator ws

descriptionIdKeyword = (\"i\"/\"I\") (\"d\"/\"D\")
descriptionId = sctId
descriptionIdSet = (\"F\"/\"F\" (\"d\"/\"D\")
termFilter = termKeyword ws stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet)
termKeyword = (\"t\"/\"T\") (\"e\"/\"E\") (\"r\"/\"R\") (\"m\"/\"M\")
typedSearchTerm = ( (matchKeyword ws ":") ws matchSearchTermSet ) ( wild ws :") ws wildSearchTermSet

typedSearchTermSet = (\"F\"/\"F\" (\"d\"/\"D\")
matchKeyword = (\"m\"/\"M\") (\"a\"/\"A\") (\"c\"/\"C\") (\"h\"/\"H\")
matchSearchTerm = 1*(nonwsNonEscapedChar / escapedChar)
matchSearchTermSet = QM ws matchSearchTerm *( (wildSearchTerm ws ")")
wildSearchTerm = 1*(anyNonEscapedChar / escapedChar)
wildSearchTermSet = QM ws wildSearchTerm *( (wildSearchTerm ws ")")
languageFilter = language ws booleanComparisonOperator ws (languageCode / languageCodeSet)
language = (\"l\"/\"L\") (\"a\"/\"A\") (\"n\"/\"N\") (\"g\"/\"G\") (\"u\"/\"U\") (\"a\"/\"A\") (\"g\"/\"G\") (\"e\"/\"E\")
languageCode = 2alpha
languageCodeSet = (\"F\"/\"F\" (\"d\"/\"D\")
typeFilter = typeFilter / typeTokenFilter
typeIdFilter = typeId ws booleanComparisonOperator ws (subExpressionConstraint / eclConceptReferenceSet)
typeId = (\"t\"/\"T\") (\"y\"/\"Y\") (\"p\"/\"P\") (\"e\"/\"E\") (\"i\"/\"I\") (\"d\"/\"D\")
typeTokenFilter = type booleanComparisonOperator ws (typeToken / typeTokenSet)
type = (\"t\"/\"T\") (\"y\"/\"Y\") (\"p\"/\"P\") (\"e\"/\"E\")
typeToken = synonym / fullySpecifiedName / definition
typeTokenSet = (\"F\"/\"F\" (\"d\"/\"D\")
synonym = (\"s\"/\"S\") (\"y\"/\"Y\") (\"n\"/\"N\")
fullySpecifiedName = (\"f\"/\"F\") (\"s\"/\"S\") (\"n\"/\"N\")
definition = (\"d\"/\"D\") (\"e\"/\"E\") (\"f\"/\"F\")
dialectFilter = (dialectIdFilter / dialectAliasFilter) [ ws acceptabilitySet ]
dialectIdFilter = dialectId ws booleanComparisonOperator ws
(subExpressionConstraint / dialectIdSet)
dialectId = ("d"/"D") ("i"/"I") ("a"/"A") ("l"/"L") ("e"/"E") ("c"/"C")
("t"/"T") ("i"/"I") ("d"/"D")
dialectAliasFilter = dialect ws booleanComparisonOperator ws (dialectAlias /
dialectAliasSet)
dialect = ("d"/"D") ("i"/"I") ("a"/"A") ("l"/"L") ("e"/"E") ("c"/"C")
("t"/"T")
dialectAlias = alpha *( dash / alpha / integerValue)
dialectAliasSet = "( ws dialectAlias [ws acceptabilitySet] *(mws dialectAlias
[ws acceptabilitySet] ) ws " )"
dialectIdSet = "( ws eclConceptReference [ws acceptabilitySet] *(mws
eclConceptReference [ws acceptabilitySet] ) ws " )"
acceptabilitySet = acceptabilityConceptReferenceSet / acceptabilityTokenSet
acceptabilityConceptReferenceSet = "( ws eclConceptReference *(mws
eclConceptReference) ws " )"
acceptabilityTokenSet = "( ws acceptabilityToken *(mws acceptabilityToken) ws " )"
acceptabilityToken = acceptable / preferred
acceptable = ("a"/"A") ("c"/"C") ("e"/"E") ("p"/"P") ("t"/"T")
preferred = ("p"/"P") ("r"/"R") ("e"/"E") ("f"/"F") ("t"/"T")
customFilterConstraint = "{ { ws ("c" / "C") ws customFilter *(ws "," ws
customFilter) ws " ) }"
customFilter = definitionStatusFilter / moduleFilter / effectiveTimeFilter /
activeFilter
definitionStatusFilter = definitionStatusIdFilter /
definitionStatusTokenFilter
definitionStatusIdFilter = definitionStatusIdKeyword ws booleanComparisonOperator ws (subExpressionConstraint /
eclConceptReferenceSet)
definitionStatusIdKeyword = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I")
("n"/"N")
("t"/"T")
definitionStatusTokenFilter = definitionStatusToken ws booleanComparisonOperator ws
definitionStatusTokenSet (definitionStatusToken /
definitionStatusTokenSet)
definitionStatusTokenSet = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I")
("n"/"N")
("t"/"T")
definitionStatusToken = primitiveToken / definedToken
definitionStatusTokenSet = "( ws definitionStatusToken *(mws
definitionStatusToken) ws " )"
primitiveToken = ("p"/"P") ("r"/"R") ("i"/"I") ("m"/"M")
("n"/"N") ("t"/"T")
("i"/"I") ("v"/"V")
definedToken = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") ("e"/"E")
("d"/"D")
moduleFilter = moduleIdKeyword ws booleanComparisonOperator ws (subExpressionConstraint /
eclConceptReferenceSet)
moduleIdKeyword = ("m"/"M") ("o"/"O") ("d"/"D") ("u"/"U")
("l"/"L") ("e"/"E")
("i"/"I") ("d"/"D")
effectiveTimeFilter = effectiveTimeKeyword ws timeComparisonOperator ws
(timeValue / timeValueSet)
effectiveTimeKeyword = ("e"/"E") ("f"/"F") ("f"/"F") ("e"/"E")
("c"/"C")
("t"/"T") ("i"/"I") ("v"/"V")
"e"/"E") ("t"/"T") ("i"/"I") ("m"/"M")"}
timeValue = QM [ year month day ] QM
timeValueSet = "(" ws timeValue *(mws timeValue) ws ")"
year = digitNonZero digit digit digit
digitNonZero = "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"
month = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" / "11" / "12"
day = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" / "11" / "12" / "13" / "14" / "15" / "16" / "17" / "18" / "19" / "20" / "21" / "22" / "23" / "24" / "25" / "26" / "27" / "28" / "29" / "30" / "31"
activeFilter = activeKeyword ws booleanComparisonOperator ws activeValue
activeKeyword = ("a"/"A") ("c"/"C") ("t"/"T") ("i"/"I") ("v"/"V") ("e"/"E")
activeValue = activeTrueValue / activeFalseValue
activeTrueValue = "1" / "true"
activeFalseValue = "0" / "false"
memberFilterConstraint = 
memberFilter = moduleFilter / effectiveTimeFilter / activeFilter / memberFieldFilter
memberFieldFilter = refsetFieldName ws (expressionComparisonOperator ws numericComparisonOperator ws "#" numericValue / stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet) / booleanComparisonOperator ws booleanValue / timeComparisonOperator ws timeValueSet)
historySupplement = "{" ws "+" ws historyKeyword [ historyProfileSuffix / ws historySubset] ws "}"
historyKeyword = ("h"/"H") ("i"/"I") ("s"/"S") ("t"/"T") ("o"/"O") ("r"/"R")
(memberFieldFilter)
numericValue = ["-"/"+"] (decimalValue / integerValue)
nonNegativeIntegerValue = (digitNonZero *digit) / zero
sctId = digitNonZero 5*17( digit )
ws = *( SP / HTAB / CR / LF / comment ) ; optional white space
mws = 1*( SP / HTAB / CR / LF / comment ) ; mandatory white space
comment = "/*" *(nonStarChar / starWithNonFSlash) "*/"
nonStarChar = SP / HTAB / CR / LF / %x21-29 / %x2B-7E / UTF8-2 / UTF8-3 / UTF8-4
starWithNonFSlash = %x2A nonFSlash
SP = %x20 ; space
HTAB = %x09 ; tab
CR = %x0D ; carriage return
LF = %x0A ; line feed
QM = %x22 ; quotation mark
BS = %x5C ; back slash
star = %x2A ; asterisk
digit = %x30-39
zero = %x30
digitNonZero = %x31-39
nonwsNonPipe = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4
anyNonEscapedChar = SP / HTAB / CR / LF / %x20-21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4
escapedChar = BS QM / BS BS
escapedWildChar = BS QM / BS BS / BS star
nonwsNonEscapedChar = %x21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4
alpha = %x41-5A / %x61-7A
dash = %x2D
UTF8-2 = %xC2-DF UTF8-tail
UTF8-3 = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) / %xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )
UTF8-4 = %xF0 %x90-BF 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) / %xF4 %x80-8F
2( UTF8-tail )
UTF8-tail = %x80-BF

5.2 5.2 Long Syntax (Informative)

The following ABNF definition specifies the Long Syntax the SNOMED CT Expression Constraint Language. Please note that all keywords are case insensitive. This ABNF syntax and the ANTLR syntax is maintained in the SNOMED Expression Constraint Language GitHub repository.

expressionConstraint = ws ( refinedExpressionConstraint / compoundExpressionConstraint / dottedExpressionConstraint / subExpressionConstraint ) ws
refinedExpressionConstraint = subExpressionConstraint ws ":" ws eclRefinement
compoundExpressionConstraint = conjunctionExpressionConstraint /
disjunctionExpressionConstraint / exclusionExpressionConstraint
conjunctionExpressionConstraint = subExpressionConstraint 1*(ws conjunction ws subExpressionConstraint)
disjunctionExpressionConstraint = subExpressionConstraint 1*(ws disjunction ws subExpressionConstraint)
exclusionExpressionConstraint = subExpressionConstraint ws exclusion ws subExpressionConstraint
dottedExpressionConstraint = subExpressionConstraint 1*(ws dottedExpressionAttribute)
dottedExpressionAttribute = dot ws eclAttributeName
subExpressionConstraint= [constraintOperator ws] ( ( [memberOf ws] (eclFocusConcept / "(" ws expressionConstraint ws ")") *(ws memberFilterConstraint)) / (eclFocusConcept / "/" ws expressionConstraint ws ")") ) *(ws (descriptionFilterConstraint / conceptFilterConstraint)) [ws historySupplement]
eclFocusConcept = eclConceptReference / wildCard / altIdentifier

34 http://snomed.org/ecl
35 https://github.com/IHTSDO/snomed-expression-constraint-language
dot = "."
memberOf = ( "^^" / ("m"/"M") ("e"/"E") ("m"/"M") ("b"/"B") ("e"/"E") ("r"/"R") ("o"/"O") ("f"/"F") ) [ ws "[" ws (refsetFieldNameSet / wildCard) ws "]" ]
refsetFieldNameSet = refsetFieldName *( ws "," ws refsetFieldName )
refsetFieldName = 1*alpha
eclConceptReference = conceptId [ ws "|" ws term ws "|""]
eclConceptReferenceSet = (" ws eclConceptReference 1*(mws
eclConceptReference) ws "]")
conceptId = sctId
term = 1*nonwsNonPipe *( 1*SP 1*nonwsNonPipe )
altIdentifier = (QM altIdentifierSchemeAlias "#" altIdentifierCodeWithinQuotes
QM / altIdentifierSchemeAlias "#" altIdentifierCodeWithoutQuotes) [ ws "]" ws
term ws "]")
altIdentifierSchemeAlias = alpha *(dash / alpha / integerValue)
altIdentifierCodeWithinQuotes = 1*alpha *(dash / alpha / integerValue)
altIdentifierCodeWithoutQuotes = 1*alpha *(dash / alpha / integerValue)
refsetFieldNameSet = refsetFieldName *( ws "," ws refsetFieldName )
refsetFieldName = 1*alpha
eclConceptReference = conceptId [ ws "]" ws term ws "|"

constraintOperator = childOf / childOrSelfOf / descendantOrSelfOf /
descendantOf / parentOf / parentOrSelfOf / ancestorOrSelfOf / ancestorOf /
top / bottom
descendantOf = ">
(descendantOrSelfOf = ">="
(childOf = ">
(childOrSelfOf = ">="
(ancestorOf = ">
(ancestorOrSelfOf = ">="
(top = ">="
(bottom = ">="
(conjunction = ("a"/"A") ("n"/"N") ("c"/"C") ("e"/"E") ("s"/"S") ("t"/"T") ("o"/"O") ("f"/"F")
(disjunction = ("o"/"O") ("r"/"R") ("n"/"N") ("w"/"W")
(exclusion = ("m"/"M") ("i"/"I") ("m"/"M") ("b"/"B")

expressionconstraintlanguage = eclAttributeSet ws [ conjunctionAttributeSet / disjunctionAttributeSet ]
conjunctionAttributeSet = 1*(ws conjunction ws subAttributeSet)
disjunctionAttributeSet = 1*(ws disjunction ws subAttributeSet)
subAttributeSet = eclAttribute / "(" ws eclAttributeSet ws ")"
eclAttributeGroup = ["[" cardinality "]" ws ] ["{" ws eclAttributeSet ws "}"]
eclAttribute = ["[" cardinality "]" ws ] [ reverseFlag ws ] eclAttributeName ws (expressionComparisonOperator ws subExpressionConstraint /)
umericComparisonOperator ws "#" numericValue / stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet) / booleanComparisonOperator ws booleanValue)
cardinality = minValue to maxValue
minValue = nonNegativeIntegerValue
to = ".." / (mws ("t"/"T") ("o"/"O") mws)
maxValue = nonNegativeIntegerValue / many
many = "*" / ( ("m"/"M") ("a"/"A") ("n"/"N") ("y"/"Y"))
reverseFlag = ( ("r"/"R") ("e"/"E") ("h"/"H") ("n"/"N") ("m"/"M") ("o"/"O") ("f"/"F") ) ) / "R"
eclAttributeName = subExpressionConstraint
expressionComparisonOperator = ":=" / ":!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / ":<" / ":<=" / ">=" / ">" timeComparisonOperator = ":=" / ":!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / ":<" / ":<=" / ">=" / ">" stringComparisonOperator = ":=" / ":!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / ":<" / ":<=" / ">=" / ">" booleanComparisonOperator = ":=" / ":!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / ":<" / ":<=" / ">=" / ">"
 descriptionFilterConstraint = {{" ws ["d" / "D"] ws descriptionFilter * (ws "," ws descriptionFilter ws ) ]
 descriptionFilter = termFilter / languageFilter / typeFilter / dialectFilter / moduleFilter / effectiveTimeFilter / activeFilter / descriptionIdFilter
descriptionIdFilter = descriptionIdKeyword ws idComparisonOperator ws descriptionIdSet
descriptionIdKeyword = ("i"/"I") ("d"/"D")
descriptionIdSet = (" ws descriptionId * (mws descriptionId) ws )
termFilter = termKeyword ws stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet)
termKeyword = ("t"/"T") ("e"/"E") ("r"/"R") ("m"/"M")
typedSearchTerm = ( [ matchKeyword ws ":" ws ] matchSearchTermSet ) / ( wild ws ":" ws wildSearchTermSet )
typedSearchTermSet = (" ws typedSearchTerm * (mws typedSearchTerm) ws )
 wild = ("w"/"W") ("n"/"N") ("i"/"I") ("l"/"L") ("d"/"D")
matchKeyword = ("n"/"N") ("a"/"A") ("t"/"T") ("c"/"C") ("h"/"H")
matchSearchTerm = 1* (nonwsNonEscapedChar / escapedChar)
matchSearchTermSet = QM ws matchSearchTerm * (mws matchSearchTerm) ws QM
wildSearchTerm = 1* (anyNonEscapedChar / escapedWildChar)
wildSearchTermSet = QM wildSearchTerm QM
languageFilter = language ws booleanComparisonOperator ws (languageCode / languageCodeSet)
language = ("l" / "L") ("a" / "A") ("n" / "N") ("g" / "G") ("u" / "U") ("a" / "A") ("e" / "E")
languageCode = 2alpha
languageCodeSet = "((\w+ languageCode \(ws mws languageCode\) ws "))"

typeFilter = typeIdFilter / typeTokenFilter
	typeIdFilter = typeId ws booleanComparisonOperator ws (subExpressionConstraint / eclConceptReferenceSet)
typeId = ("t" / "T") ("y" / "Y") ("p" / "P") ("e" / "E") ("i" / "I") ("d" / "D")
typeTokenFilter = type ws booleanComparisonOperator ws (typeToken / typeTokenSet)
type = ("t" / "T") ("y" / "Y") ("p" / "P") ("e" / "E")
typeToken = synonym / fullySpecifiedName / definition
typeTokenSet = "((\w+ typeToken \(ws mws typeToken\) ws "))"
synonym = ("s" / "S") ("y" / "Y") ("n" / "N") \[ ("o" / "O") ("n" / "N") ("y" / "Y") ("m" / "M") \]
fullySpecifiedName = ("f" / "F") ("s" / "S") ("n" / "N") \[ ("o" / "O") ("n" / "N") \[ ("y" / "Y") ("m" / "M") \]
definition = ("d" / "D") ("e" / "E") ("f" / "F") \[ ("i" / "I") ("n" / "N") ("i" / "I") \[ ("t" / "T") ("n" / "N") ("o" / "O") ("n" / "N") \]
dialectFilter = (dialectIdFilter / dialectAliasFilter) \[ ws acceptabilitySet \]
dialectIdFilter = dialectId ws booleanComparisonOperator ws (subExpressionConstraint / dialectIdSet)
dialectId = ("d" / "D") ("i" / "I") ("a" / "A") ("l" / "L") ("e" / "E") ("c" / "C")
(dialectAliasFilter = dialect ws booleanComparisonOperator ws (dialectAlias / dialectAliasSet)
dialect = ("d" / "D") ("i" / "I") ("a" / "A") ("l" / "L") ("e" / "E") ("c" / "C")
(dialectAlias = alpha \(\times\) (dash / alpha / integerValue)
dialectAliasSet = "((\w+ dialectAlias \[ws acceptabilitySet\] \(ws mws dialectAlias \[ws acceptabilitySet\] \) ws "))"
dialectIdSet = "((\w+ eclConceptReference \[ws acceptabilitySet\] \(ws mws eclConceptReference \[ws acceptabilitySet\] \) ws "))"
acceptabilitySet = acceptabilityConceptReferenceSet / acceptabilityTokenSet
acceptabilityConceptReferenceSet = "((\w+ eclConceptReference \[ws acceptabilitySet\] \(ws mws eclConceptReference \[ws acceptabilitySet\] \) ws "))"
acceptabilityTokenSet = "((\w+ acceptabilityToken \[ws acceptabilityToken\] \(ws mws acceptabilityToken\) ws "))"

acceptable = ("a" / "A") ("c" / "C") ("e" / "E") ("p" / "P") ("t" / "T")
acceptableToken = acceptable / preferred
preferred = ("p" / "P") ("r" / "R") ("f" / "F") ("e" / "E") ("r" / "R")
(preferred = ("r" / "R") ("e" / "E") ("d" / "D") \]
conceptFilterConstraint = 
conceptFilter = definitionStatusFilter / moduleFilter / effectiveTimeFilter / activeFilter
definitionStatusFilter = definitionStatusIdFilter /
definitionStatusTokenFilter = definitionStatusTokenIdFilter \nbooleanComparisonOperator ws \neclConceptReferenceSet

definitionStatusTokenIdFilter = \n("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") \n("a"/"A") ("t"/"T") ("u"/"U") ("s"/"S") \ndefinitionStatusTokenIdKeyword = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") \n("a"/"A") ("t"/"T") ("u"/"U") ("s"/"S") \ndefinitionStatusToken = primitiveToken / definedToken

definitionStatusToken = "(" ws definitionStatusToken *(mws \ndefinitionStatusToken) ws ")"

primitiveToken = ("p"/"P") ("r"/"R") ("i"/"I") ("m"/"M") ("t"/"T") \n("i"/"I") ("v"/"V") ("e"/"E")
definedToken = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") ("e"/"E") ("a"/"A") ("t"/"T") ("u"/"U") ("s"/"S")

moduleFilter = moduleIdKeyword \neclConceptReferenceSet

moduleIdKeyword = ("m"/"M") ("o"/"O") ("u"/"U") ("l"/"L") ("e"/"E") \n(moduleIdKeyword)\n
effectiveTimeFilter = effectiveTimeKeyword \ntimeComparisonOperator \ntimeValue \ntimeValueSet

effectiveTimeKeyword = ("e"/"E") ("f"/"F") ("i"/"I") ("v"/"V") ("e"/"E") ("m"/"M") \neffectiveTimeKeyword = ("e"/"E")
timeValue = QM [ year month day ] QM

timeValueSet = "(" ws timeValue *(mws timeValue) ws ")"

year = digitNonZero digit digit digit
digit = "0" / "1" / "2" / "3" / "4" / "5" / "6" / "7" / "8" / "9"

month = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" \n("11" / "12")
day = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" / "11" / "12" / "13" / "14" / "15" / "16" / "17" / "18" / "19" / "20" / "21" / "22" / "23" / "24" / "25" / "26" / "27" / "28" / "29" / "30" / "31"

activeFilter = activeKeyword \néclConceptReferenceSet

activeKeyword = ("a"/"A") ("c"/"C") ("n"/"N") ("e"/"E") ("t"/"T") ("i"/"I") ("v"/"V") ("e"/"E")

activeValue = activeTrueValue / activeFalseValue

activeTrueValue = "1" / "true"

activeFalseValue = "0" / "false"

memberFilter = moduleFilter / effectiveTimeFilter / activeFilter / memberFieldFilter

memberFieldFilter = refsetFieldName \nexpressionComparisonOperator \nnumericComparisonOperator \nstringComparisonOperator \nbooleanComparisonOperator \ntimeComparisonOperator \ntimeValueSet

historySupplement = "{" ws "+" ws historyKeyword [ historyProfileSuffix / ws historySubset ] ws "}"
5. Syntax Specification

- **historyKeyword** = ("h"/"H") ("i"/"I") ("s"/"S") ("t"/"T") ("o"/"O") ("r"/"R") ("y"/"Y")

- **historyProfileSuffix** = historyMinimumSuffix / historyModerateSuffix / historyMaximumSuffix

- **historyMinimumSuffix** = ("-"/"_") ("m"/"M") ("i"/"I") ("n"/"N")

- **historyModerateSuffix** = ("-"/"_") ("m"/"M") ("o"/"O") ("d"/"D")

- **historyMaximumSuffix** = ("-"/"_") ("m"/"M") ("a"/"A") ("x"/"X")

- **numericValue** = ["-"/"+"] (decimalValue / integerValue)

- **stringValue** = 1*(anyNonEscapedChar / escapedChar)

- **integerValue** = digitNonZero *digit / zero

- **decimalValue** = integerValue "." 1*digit

- **booleanValue** = true / false

- **sctId** = digitNonZero 5*17( digit )

- **ws** = *( SP / HTAB / CR / LF / comment ) ; optional white space

- **mws** = 1*( SP / HTAB / CR / LF / comment ) ; mandatory white space

- **comment** = "/*" *(nonStarChar / starWithNonFSlash) "*/"

- **nonStarChar** = SP / HTAB / CR / LF / %x21-29 / %x2B-7E /UTF8-2 / UTF8-3 / UTF8-4

- **starWithNonFSlash** = %x2A nonFSlash

- **nonFSlash** = SP / HTAB / CR / LF / %x21-2E / %x30-7E /UTF8-2 / UTF8-3 / UTF8-4

- **SP** = %x20 ; space

- **HTAB** = %x09 ; tab

- **CR** = %x0D ; carriage return

- **LF** = %x0A ; line feed

- **QM** = %x22 ; quotation mark

- **BS** = %x5C ; back slash

- **star** = %x2A ; asterisk

- **digit** = %x30-39

- **zero** = %x30

- **digitNonZero** = %x31-39

- **nonwsNonPipe** = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4

- **anyNonEscapedChar** = SP / HTAB / CR / LF / %x20-21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4

- **escapedChar** = BS QM / BS BS

- **escapedWildChar** = BS QM / BS BS / BS star

- **nonwsNonEscapedChar** = %x21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4

- **alpha** = %x41-5A / %x61-7A

- **dash** = %x2D

- **UTF8-2** = %xC2-DF UTF8-tail

- **UTF8-3** = %xE0 %xA0-BF UTF8-tail / %xE1-EC 2( UTF8-tail ) / %xED %x80-9F UTF8-tail / %xEE-EF 2( UTF8-tail )

- **UTF8-4** = %xF0 %x80-8F 2( UTF8-tail ) / %xF1-F3 3( UTF8-tail ) / %xF4 %x80-8F 2( UTF8-tail )

- **UTF8-tail** = %x80-BF
## 5.3 Informative Comments

This section provides a short description of each ABNF rule listed above. The related brief and long syntax rules are grouped together with the same description. Where the syntaxes are the same, the rule is listed once and preceded with the text "BS/LS". Where the brief and long syntaxes are different, both rules are listed separately and preceded with "BS" and "LS" respectively.

<table>
<thead>
<tr>
<th>BS/LS:</th>
<th>Rule Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>expressionConstraint</strong></td>
<td>( ws\ (\text{refinedExpressionConstraint} / \text{compoundExpressionConstraint} / \text{dottedExpressionConstraint} / \text{subExpressionConstraint}) ) ( ws)</td>
</tr>
</tbody>
</table>

An expression constraint is either a refined expression constraint, a compound expression constraint, a dotted expression constraint, or a sub expression constraint.

<table>
<thead>
<tr>
<th>BS/LS:</th>
<th>Rule Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>refinedExpressionConstraint</strong></td>
<td>( \text{subExpressionConstraint} \ ws ; \ ws \text{eclRefinement} )</td>
</tr>
</tbody>
</table>

A refined expression constraint includes a subexpression constraint followed by a refinement.

<table>
<thead>
<tr>
<th>BS/LS:</th>
<th>Rule Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>compoundExpressionConstraint</strong></td>
<td>( \text{conjunctionExpressionConstraint} / \text{disjunctionExpressionConstraint} / \text{exclusionExpressionConstraint} )</td>
</tr>
</tbody>
</table>

A compound expression constraint contains two or more expression constraints joined by either a conjunction, disjunction or exclusion. When potential ambiguity in binary operator precedence may occur, round brackets must be used to clearly disambiguate the order in which these operator are applied. Brackets are not required in expression constraints in which all binary operators are conjunctions, or all binary operators are disjunctions. Please note that unary operators (i.e. constraint operators and member of functions) are always applied before binary operators (i.e. conjunction, disjunction and exclusion).

<table>
<thead>
<tr>
<th>BS/LS:</th>
<th>Rule Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>conjunctionExpressionConstraint</strong></td>
<td>( \text{subExpressionConstraint} \ 1*(ws \text{conjunction} ws \text{subExpressionConstraint}) )</td>
</tr>
</tbody>
</table>

A conjunction expression constraint combines two or more expression constraints with a conjunction ("and") operator. More than one conjunction may be used without brackets. However any compound expression constraint (using a different binary operator) that appears within a conjunction expression constraint must be enclosed by brackets.

<table>
<thead>
<tr>
<th>BS/LS:</th>
<th>Rule Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>disjunctionExpressionConstraint</strong></td>
<td>( \text{subExpressionConstraint} \ 1*(ws \text{disjunction} ws \text{subExpressionConstraint}) )</td>
</tr>
</tbody>
</table>

A disjunction expression constraint combines two or more expression constraints with a disjunction ("or") operator. More than one disjunction may be used without brackets. However any compound expression constraint (using a different binary operator) that appears within a disjunction expression constraint must be enclosed by brackets.
### Expression Constraint Language - Specification and Guide

#### 5. Syntax Specification

**BS/LS: `exclusionExpressionConstraint`** = `subExpressionConstraint ws exclusion ws subExpressionConstraint`

An exclusion expression constraint combines two expression constraints with an exclusion ("minus") operator. A single exclusion operator may be used without brackets. However when the operands of the exclusion expression constraint are compound, these compound expression constraints must be enclosed by brackets.

**BS/LS: `dottedExpressionConstraint`** = `subExpressionConstraint 1*(ws dottedExpressionAttribute)`

A dotted expression constraint contains a sub expression constraint, followed by one or more dotted attributes. When a single dotted attribute is used, the result is the set of attribute values (for the given attribute name) of each concept that results from evaluating the subExpressionConstraint. When more than one dotted attribute is used, each dottedExpressionAttribute is sequentially evaluated (from left to right) against the given result set.

**BS/LS: `dottedExpressionAttribute`** = `dot ws eclAttributeName`

A dotted expression attribute consists of a 'dot', followed by an attribute name. Please note that the attribute name may be represented by any sub expression constraint.

**BS/LS: `subExpressionConstraint`** = `[constraintOperator ws] ( [memberOf ws] (eclFocusConcept / "(" ws expressionConstraint ws ")") *(ws memberFilterConstraint)) / (eclFocusConcept / "(" ws expressionConstraint ws ")") ) *(ws (descriptionFilterConstraint / conceptFilterConstraint)) [ws historySupplement]

A sub expression constraint optionally begins with a constraint operator and/or a memberOf function. It then includes either a single focus concept or an expression constraint (enclosed in brackets). If the memberOf function is applied, a member filter constraint may be used. A sub expression constraint may then optionally include one or more concept or description filter constraints, followed optionally by a history supplement.

Notes: A memberOf function should be used only when the eclFocusConcept or expressionConstraint refers to a reference set concept, a set of reference set concepts, or a wild card. When both a constraintOperator and a memberOf function are used, they are applied from the inside to out (i.e. from right to left) - see 5.4 Order of Operation[see page 56]. Therefore, if a constraintOperator is followed by a memberOf function, then the memberOf function is processed prior to the constraintOperator.

**BS/LS: `eclFocusConcept`** = `eclConceptReference / wildCard`

A focus concept is a concept reference or a wild card.

**BS/LS: `dot`** = "."
A dot connects an expression constraint with an attribute whose values are included in the result.

**BS:** `memberOf = "^" [ ws "[" ws (refsetFieldNameSet / wildCard) ws "]" ]`

**LS:** `memberOf = ( "^" / ("m"/"M") ("e"/"E") ("m"/"M") ("b"/"B") ("e"/"E") ("r"/"R") ("o"/"O") ("f"/"F") ) [ ws "]" ws (refsetFieldNameSet / wildCard) ws ")" ]`

By default, the 'memberOf' function returns the set of referenced components in the set of reference sets which follows. In the brief syntax, the memberOf function is represented using the "^" symbol. In the long syntax, the text "memberOf " (case insensitive and followed by at least one white space) is also allowed. If a set of reference set fields is listed in square brackets after the memberOf function, then the values of these fields are returned.

**BS/LS:** `refsetFieldNameSet = refsetFieldName *( ws "," ws refsetFieldName )`

A refsetFieldNameSet is a set of one or more reference set fields, separated by a comma and optional whitespace.

**BS/LS:** `refsetFieldName = 1*alpha`

A refsetFieldName is the set of alphabetic characters used to name a reference set field.

**BS/LS:** `eclConceptReference = conceptId [ws "|" ws term ws "]|"

A conceptReference is represented by a ConceptId, optionally followed by a term enclosed by a pair of "|" characters. Whitespace before or after the ConceptId is ignored as is any whitespace between the initial "|" characters and the first non-whitespace character in the term or between the last non-whitespace character and before second "|" character.

**BS/LS:** `eclConceptReferenceSet = "(" ws eclConceptReference 1*(mws eclConceptReference) ws ")"`

A concept reference set includes two or more concept references separated by mandatory white space and enclosed in brackets.

**BS/LS:** `conceptId = sctId`

The ConceptId must be a valid SNOMED CT identifier for a concept. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.

---

36 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
37 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
38 https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+identifier
39 https://confluence.ihtsdotools.org/display/DOCGLOSS/concept
Expression Constraint Language – Expression Constraint Language - Specification and Guide

5. Syntax Specification

### BS/LS: `term = 1*nonwsnonpipe *( 1*SP 1*nonwsnonpipe )`

The `term` must be the `term` from a SNOMED CT description that is associated with the concept identified by the preceding concept identifier. For example, the `term` could be the preferred description, or the preferred description associated with a particular translation. The `term` may include valid UTF-8 characters except for the pipe "

### BS: `wildCard = "*"`

**LS: `wildCard = "+" / ( ("a"/"A") ("n"/"N") ("y"/"Y") )**

A wild card represents any concept in the given substrate. In the brief syntax, a wildcard is represented using the "*" symbol. In the long syntax, the text "ANY" (case insensitive) is also allowed.

### BS/LS: `constraintOperator = childOf / childOrSelfOf / descendantOrSelfOf / descendantOf / parentOf / parentOrSelfOf / ancestorOrSelfOf / ancestorOf`

A constraint operator is either 'childOf', 'childOrSelfOf', 'descendantOrSelfOf', 'descendantOf', 'parentOf', 'parentOrSelfOf', 'ancestorOrSelfOf', or 'ancestorOf'.

### BS: `descendantOf = "<"`

**LS: `descendantOf = "<" / ( ("d"/"D") ("e"/"E") ("s"/"S") ("c"/"C") ("e"/"E") ("n"/"N") ("d"/"D") ("a"/"A") ("n"/"N") ("t"/"T") ("o"/"O") ("f"/"F") mws )**

The descendantOf operator returns the set of all subtypes of the given concept (or set of concepts). In the brief syntax, the descendantOf operator is represented using the symbol ". In the long syntax, the text "descendantOf" (case insensitive and followed by at least one white space) is also allowed.

### BS: `descendantOrSelfOf = "<<"`

**LS: `descendantOrSelfOf = "<<" / ( ("d"/"D") ("e"/"E") ("s"/"S") ("c"/"C") ("e"/"E") ("n"/"N") ("d"/"D") ("a"/"A") ("n"/"N") ("t"/"T") ("o"/"O") ("r"/"R") ("s"/"S") ("e"/"E") ("l"/"L") ("f"/"F") ("o"/"O") ("f"/"F") mws )**

---

40 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
41 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
42 https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+description
43 https://confluence.ihtsdotools.org/display/DOCGLOSS/concept
44 https://confluence.ihtsdotools.org/display/DOCGLOSS/concept+identifier
45 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
46 https://confluence.ihtsdotools.org/display/DOCGLOSS/description
47 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
48 https://confluence.ihtsdotools.org/display/DOCRELFMT/term+(field)
49 https://confluence.ihtsdotools.org/display/DOCRELFMT/UTF-8
The descendantOrSelfOf operator returns the set of all subtypes of the given concept (or set of concepts), plus the concept (or set of concepts) itself. In the brief syntax, the `descendantOrSelfOf` is represented using the symbols "<<". In the long syntax, the text "descendantOrSelfOf" (case insensitive and followed by at least one white space) is also allowed.

**BS:** `childOf = "<!"`

**LS:**
```
childOf = "<!" / ("c"/"C") ("h"/"H") ("i"/"I") ("l"/"L") ("d"/"D") ("o"/"O") ("f"/"F") mws )
```

The childOf operator returns the set of all immediate children of the given concept (or set of concepts). In the brief syntax, the `childOf` is represented using the symbols "<!". In the long syntax, the text "childOf" (case insensitive and followed by at least one white space) is also allowed.

**BS:** `childOrSelfOf = "<<!"`

**LS:**
```
childOrSelfOf = "<<!" / ("c"/"C") ("h"/"H") ("i"/"I") ("l"/"L") ("d"/"D") ("o"/"O") ("r"/"R") ("s"/"S") ("e"/"E") ("l"/"L") ("f"/"F") mws )
```

The childOrSelfOf operator returns the set of all immediate children of the given concept (or set of concepts), plus the concept (or set of concepts) itself. In the brief syntax, the `childOrSelfOf` is represented using the symbols "<<!". In the long syntax, the text "childOrSelfOf" (case insensitive and followed by at least one white space) is also allowed.

**BS:** `ancestorOf = ">"`

**LS:**
```
ancestorOf = ">" / ("a"/"A") ("n"/"N") ("c"/"C") ("e"/"E") ("s"/"S") ("t"/"T") ("o"/"O") ("r"/"R") ("i"/"I") ("f"/"F") mws )
```

The ancestorOf operator returns the set of all supertypes of the given concept (or set of concepts). In the brief syntax, the `ancestorOf` is represented using the symbol ">". In the long syntax, the text "ancestorOf" (case insensitive and followed by at least one white space) is also allowed.

**BS:** `ancestorOrSelfOf = ">>!"`

**LS:**
```
ancestorOrSelfOf = ">>!" / ("a"/"A") ("n"/"N") ("c"/"C") ("e"/"E") ("s"/"S") ("t"/"T") ("o"/"O") ("r"/"R") ("i"/"I") ("f"/"F") mws )
```

The ancestorOrSelfOf operator returns the set of all supertypes of the given concept (or set of concepts), plus the concept (or set of concepts) itself. In the brief syntax, the `ancestorOrSelfOf` is represented using the symbols ">>!". In the long syntax, the text "ancestorOrSelfOf" (case insensitive and followed by at least one white space) is also allowed.

**BS:** `parentOf = ">!"`

**LS:**
```
parentOf = ">!" / ("p"/"P") ("a"/"A") ("n"/"N") ("e"/"E") ("t"/"T") ("o"/"O") ("f"/"F") mws )
```

The parentOf operator returns the set of all parents of the given concept (or set of concepts), plus the concept (or set of concepts) itself. In the brief syntax, the `parentOf` is represented using the symbols ">!". In the long syntax, the text "parentOf" (case insensitive and followed by at least one white space) is also allowed.
The parentOf operator returns the set of all immediate parents of the given concept (or set of concepts). In the brief syntax, the parentOf operator is represented using the symbols ">!". In the long syntax, the text "parentOf" (case insensitive and followed by at least one white space) is also allowed.

BS: `parentOrSelfOf = ">>!"`
LS: `parentOrSelfOf = ">>!" / ("p"/"P") ("a"/"A") ("t"/"T") ("e"/"E") ("n"/"N") ("o"/"O") ("r"/"R") ("s"/"S") ("f"/"F") ("o"/"O") ("f"/"F") mws )`

The parentOrSelfOf operator returns the set of all immediate parents of the given concept (or set of concepts), plus the concept (or set of concepts) itself. In the brief syntax, the parentOrSelfOf operator is represented using the symbols ">>>!". In the long syntax, the text "parentOrSelfOf" (case insensitive and followed by at least one white space) is also allowed.

BS/LS: `conjunction = ("a"/"A") ("n"/"N") ("d"/"D") mws) / ","`

A conjunction is represented either by the word "and" (case insensitive and followed by at least one white space), or by a comma.

BS/LS: `disjunction = ("o"/"O") ("r"/"R") mws`

A disjunction is represented by the word "or" (case insensitive and followed by at least one white space).

BS/LS: `exclusion = ("m"/"M") ("l"/"L") ("u"/"U") ("s"/"S") mws`

The exclusion operator is represented by the word "minus" (case insensitive and followed by at least one white space).

BS/LS: `eclRefinement = subRefinement ws [conjunctionRefinementSet / disjunctionRefinementSet]`

A refinement contains all the grouped and ungrouped attributes that refine the set of clinical meanings satisfied by the expression constraint. Refinements may represent the conjunction or disjunction of two smaller refinements, and may optionally be placed in brackets. Where both conjunction and disjunction are used, brackets are mandatory to disambiguate the intended meaning.

BS/LS: `conjunctionRefinementSet = 1*(ws conjunction ws subRefinement)`

A conjunction refinement set consists of one or more conjunction operators, each followed by a subRefinement.

BS/LS: `disjunctionRefinementSet = 1*(ws disjunction ws subRefinement)`
A disjunction refinement set consists of one or more disjunction operators, each followed by a subRefinement.

BS/LS: subRefinement = eclAttributeSet / eclAttributeGroup / "(" ws eclRefinement ws ")"

A subRefinement is either an attribute set, an attribute group or a bracketed refinement.

BS/LS: eclAttributeSet = subAttributeSet ws [conjunctionAttributeSet / disjunctionAttributeSet]

An attribute set contains one or more attribute name-value pairs separated by a conjunction or disjunction operator. An attribute set may optionally be placed in brackets.

BS/LS: conjunctionAttributeSet = 1*(ws conjunction ws subAttributeSet)

A conjunction attribute set consists of one or more conjunction operators, each followed by a subAttributeSet.

BS/LS: disjunctionAttributeSet = 1*(ws disjunction ws subAttributeSet)

A disjunction attribute set consists of one or more disjunction operators, each followed by a subAttributeSet.

BS/LS: subAttributeSet = eclAttribute / "(" ws eclAttributeSet ws ")"

A subAttributeSet is either an attribute or a bracketed attribute set.

BS/LS: eclAttributeGroup = [ "[" cardinality "]" ws ] "{" ws eclAttributeSet ws "}"

An attribute group contains a collection of attributes that operate together as part of the refinement of the containing expression constraint. An attribute group may optionally be preceded by a cardinality. An attribute group cardinality indicates the minimum and maximum number of attribute groups that must satisfy the given attributeSet constraint for the expression constraint to be satisfied.

BS/LS: eclAttribute = [ "[" cardinality "]" ws ] [reverseFlag ws] eclAttributeName ws (expressionComparisonOperator ws subExpressionConstraint / numericComparisonOperator ws "#" numericValue / stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet) / booleanComparisonOperator ws booleanValue)

50 https://confluence.ihtsdotools.org/display/DOCGLOSS/attribute+name
51 https://confluence.ihtsdotools.org/display/DOCGLOSS/attribute+group
52 https://confluence.ihtsdotools.org/display/DOCGLOSS/refinement
53 https://confluence.ihtsdotools.org/display/DOCGLOSS/expression
An attribute is a name-value pair expressing a single refinement of the containing expression constraint. Either the attribute value must satisfy (or not) the given expression constraint, the attribute value is compared with a given numeric value (integer or decimal) using a numeric comparison operator, the attribute value must match (or not match) the given typedSearchTerm or typedSearchTermSet, or the attribute value must be equal to (or not equal to) the given boolean value. The attribute may optionally be preceded by a cardinality constraint and/or a reverse flag.

<table>
<thead>
<tr>
<th>BS/LS: cardinality = minValue to maxValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>The cardinality represents a constraint on the minimum and maximum number of times that the given attribute or attribute group may appear in a matching expression. The cardinality is enclosed in square brackets with the minimum cardinality appearing first, followed by a separator (two dots in the brief syntax), and then the maximum cardinality.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BS/LS: minValue = nonNegativeIntegerValue</th>
</tr>
</thead>
<tbody>
<tr>
<td>A value that represents the minimum number of times that an attribute or attribute group may appear. The minimum cardinality must always be less than or equal to the maximum cardinality.</td>
</tr>
</tbody>
</table>

| BS: to = ".." |
| LS: to = ".." / (mws "t"/"T") ("o"/"O") mws |
| In the brief syntax, the minimum and maximum cardinality are separated by two dots (i.e. ".."). In the long syntax, the text "to" (case insensitive with at least one white space before and after) is also allowed between the two cardinalities. |

<table>
<thead>
<tr>
<th>BS/LS: maxValue = nonNegativeIntegerValue / many</th>
</tr>
</thead>
<tbody>
<tr>
<td>A value that represents the maximum number of times that an attribute or attribute group may appear. A maximum cardinality of 'many' indicates that there is no limit on the number of times the attribute may appear.</td>
</tr>
</tbody>
</table>

| BS: many = "***" |
| LS: many = "***" / ( ("m"/"M") ("a"/"A") ("n"/"N") ("y"/"Y")) |
| In the brief syntax, a cardinality of 'many' is represented using the symbol "***". In the long syntax, the text "many" (case insensitive, with no trailing space) is also allowed. |

---

54 https://confluence.ihtsdotools.org/display/DOCGLOSS/attribute+name
55 https://confluence.ihtsdotools.org/display/DOCGLOSS/refinement
56 https://confluence.ihtsdotools.org/display/DOCGLOSS/expression

5. Syntax Specification – 39
When a reverse flag is used on an attribute, the matching relationships are traversed in the reverse of the normal direction. This means that the target concept of each relationship must match the focus concept to which the attribute is applied, while the source concept of the relationship must match the attribute value. In the brief syntax, the reverse flag is represented using the character "R" (in uppercase). In the long syntax, the text "reverseOf" (case insensitive) is also allowed.

The attribute name is the name of an attribute (or relationship type) to which a value is applied to refine the meaning of a containing expression constraint. The attribute name is represented using a subExpressionConstraint, as defined above.

Attributes whose value is a concept may be compared to an expression constraint using either equals ("=") or not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.

Attributes whose value is numeric (i.e. integer or decimal) may be compared to a specific concrete value using a variety of comparison operators, including equals ("=") , less than ("<"), less than or equals ("<="), greater than (">"), greater than or equals (">=") and not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.

Date and time values may be compared using a variety of comparison operators, , including equals ("=") , less than ("<"), less than or equals ("<="), greater than (">"), greater than or equals (">=") and not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.
Attributes whose value is a string may be compared to an expression constraint using either equals ("=") or not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.

| BS: booleanComparisonOperator = "=" / "!=" |
| LS: booleanComparisonOperator = "=" / "!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / "<>" |

Attributes whose value is a boolean may be compared to an expression constraint using either equals ("=") or not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.

| BS: idComparisonOperator = "=" / "!=" |
| LS: idComparisonOperator = "=" / "!=" / ("n"/"N") ("o"/"O") ("t"/"T") ws "=" / "<>" |

Filter criteria whose value is a SCTID may be compared to a SNOMED CT identifier using either equals ("=") or not equals ("!="). In the long syntax "<>" and "not =" (case insensitive) are also valid ways to represent not equals.

| BS/LS: descriptionFilterConstraint = "{{" ws ["d", / "D"] ws descriptionFilter *(ws "," ws descriptionFilter) ws "]]" |

A descriptionFilterConstraint is a constraint used to filter the concepts in the result set, according to whether or not the given conditions match at least one of the concept’s descriptions. A description filter constraint is always enclosed in double curly braces. Within these braces, it should (preferably) start with the letter 'D' followed by one or more description filters.

| BS/LS: descriptionFilter = termFilter / languageFilter / typeFilter / dialectFilter / moduleFilter / effectiveTimeFilter / activeFilter / descriptionIdFilter |

A description filter is either a term filter, a language filter, a type filter, a dialect filter, a module filter, an effective time filter, an active filter or a description id filter.

| BS/LS: descriptionIdFilter = descriptionIdKeyword ws idComparisonOperator ws (descriptionId / descriptionIdSet) |

A descriptionIdFilter starts with the 'id' keyword, followed by an id comparison operator and either a single description id or a set of description ids.

| BS/LS: descriptionIdKeyword = ("i"/"I") ("d"/"D") |

The description id keyword uses the text "id" (case insensitive)
BS/LS: descriptionId = sctId

The descriptionId must be a valid SNOMED CT identifier\(^7\) for a description\(^8\). The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.

BS/LS: descriptionIdSet = "(" ws descriptionId *(mws descriptionId) ws ")"

A description id set consists of one or more description ids separated by mandatory white space and enclosed in brackets.

BS/LS: termFilter = termKeyword ws stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet)

A termFilter starts with the 'term' keyword, followed by a string comparison operator and either a typed search term or a typed search term set (with optional white space between). For example: term = "respiratory".

BS/LS: termKeyword = ("t"/"T") ("e"/"E") ("r"/"R") ("m"/"M")

The term keyword uses the text "term" (case insensitive).

BS/LS: typedSearchTerm = ([ matchKeyword ws "." ws ] matchSearchTermSet) / ( wild ws "." ws wildSearchTermSet )

A typed search term is either a match search term set or a wild search term set. A match search term set is optionally preceded by the text "match" and a colon. A wild search term set must be preceded by the text "wild" and a colon.

BS/LS: typedSearchTermSet = "(" ws typedSearchTerm *(mws typedSearchTerm) ws ")"

A typed search term set consists of one or more typed search terms separated by mandatory white space and enclosed in brackets.

BS/LS: wild = ("w"/"W") ("i"/"I") ("l"/"L") ("d"/"D")

A wildcard search type is indicated by the word "wild" (case insensitive).

BS/LS: matchKeyword = ("m"/"M") ("a"/"A") ("t"/"T") ("c"/"C") ("h"/"H")

A word prefix any order search is indicated by the word "match" (case insensitive).

---

\(^7\)https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+identifier
\(^8\)https://confluence.ihtsdotools.org/display/DOCGLOSS/description
### BS/LS: `matchSearchTerm` = \(^1*(\text{nonwsNonEscapedChar} / \text{escapedChar})\)

A term used in a match search includes one or more of any non-whitespace printable character (other than double quotes or backslash) or an escaped character.

### BS/LS: `matchSearchTermSet` = `QM ws matchSearchTerm *(mws matchSearchTerm) ws QM`

A term set in a match search includes one or more terms separated by mandatory whitespace and enclosed in quotation marks.

### BS/LS: `wildSearchTerm` = \(^1*(\text{anyNonEscapedChar} / \text{escapedWildChar})\)

A term used in a wildcard search includes one or more printable characters (other than double quotes or backslash) or an escaped character.

### BS/LS: `wildSearchTermSet` = `QM wildSearchTerm QM`

A term set in a wildcard search includes a wildcard search term (optionally including whitespace) enclosed in quotation marks.

### BS/LS: `languageFilter` = `language ws booleanComparisonOperator ws (languageCode / languageCodeSet)`

A language filter specifies the languages that a matching description may use. A language filter starts with the 'language' keyword, followed by a boolean comparison operator and either a single language code or a set of language codes.

### BS/LS: `language` = (`"l"/"L") (`"a"/"A") (`"n"/"N") (`"g"/"G") (`"u"/"U") (`"a"/"A") (`"e"/"E")

The 'language' keyword uses the text "LANGUAGE" (case insensitive).

### BS/LS: `languageCode` = 2alpha

A language code is a 2 character alphanumeric string.

### BS/LS: `languageCodeSet` = `"(" ws languageCode *(mws languageCode) ws ")"`

A language code set is one or more language codes, separated by mandatory whitespace, and enclosed in brackets.

### BS/LS: `typeFilter` = `typeIdFilter / typeTokenFilter`
A type filter specifies the description types that a matching description may have. A type filter is either a typeId filter or a typeToken filter.

**BS/LS:** `typeId = typeId ws booleanComparisonOperator ws (subExpressionConstraint / eclConceptReferenceSet)`

A typeId filter starts with the 'typeId' keyword, followed by a boolean comparison operator, and either a subExpressionConstraint or a set of concept references.

**BS/LS:** `typeId = ("t"/"T") ("y"/"Y") ("p"/"P") ("e"/"E") ("i"/"I") ("d"/"D")`

The 'typeId' keyword uses the text "TYPEID" (case insensitive).

**BS/LS:** `typeTokenFilter = type ws booleanComparisonOperator ws (typeToken / typeTokenSet)`

A typeToken filter starts with the 'type' keyword, followed by a boolean comparison operator, and either a single type token or a set of type tokens.

**BS/LS:** `type = ("t"/"T") ("y"/"Y") ("p"/"P") ("e"/"E")`

The 'type' keyword uses the text "TYPE" (case insensitive).

**BS/LS:** `typeToken = synonym / fullySpecifiedName / definition`

A type token is either a 'synonym' token, a 'fully specified name' token or a 'definition' token.

**BS/LS:** `typeTokenSet = (" ws typeToken *(mws typeToken) ws ")`

A type token set is one or more type tokens, separated by mandatory whitespace and enclosed in brackets.

**BS:** `synonym = ("s"/"S") ("y"/"Y") ("n"/"N")`

**LS:** `synonym = ("s"/"S") ("y"/"Y") ("n"/"N") [ ("o"/"O") ("m"/"M") ]`

A 'synonym' token uses the text "SYN" (case insensitive). In the long syntax, the text "Synonym" (case insensitive) may be used instead.

**BS:** `fullySpecifiedName = ("f"/"F") ("s"/"S") ("n"/"N")`

**LS:** `fullySpecifiedName = ( ("f"/"F") ("s"/"S") ("n"/"N") ) / ( ("a"/"A") ("m"/"M") ("e"/"E") ("d"/"D") ("i"/"I") ) ("p"/"P") ("c"/"C") ("l"/"L") ) ("y"/"Y") ("s"/"S") ("p"/"P") ("e"/"E") ("i"/"I") ) ("f"/"F") ("i"/"I") ("e"/"E") )`

A 'synonym' token uses the text "SYN" (case insensitive). In the long syntax, the text "Synonym" (case insensitive) may be used instead.
A 'fully specified name' token uses the text "FSN" (case insensitive). In the long syntax, the text "FullySpecifiedName" (case insensitive) may be used instead.

**BS:** definition = ("d"/"D") ("e"/"E") ("f"/"F")

**LS:** definition = ("d"/"D") ("e"/"E") ("f"/"F") [ ("i"/"I") ("n"/"N") ("i"/"I") ("t"/"T") ("i"/"I") ("o"/"O") ("n"/"N") ]

A 'definition' token uses the text "DEF" (case insensitive). In the long syntax, the text "Definition" (case insensitive) may be used instead.

**BS/LS:** dialectFilter = (dialectIdFilter / dialectAliasFilter) [ ws acceptabilitySet ]

A dialect filter specifies the language reference sets to which a matching description must belong. A dialect filter consists of either a dialectId filter or a dialectAlias filter, optionally followed by a set of acceptability values.

**BS/LS:** dialectIdFilter = dialectId ws booleanComparisonOperator ws (subExpressionConstraint / dialectIdSet)

A dialectId filter starts with the 'dialectId' keyword, followed by a boolean comparison operator, and either a subExpressionConstraint or a set of dialectIds.

**BS/LS:** dialectId = ("d"/"D") ("i"/"I") ("a"/"A") ("l"/"L") ("e"/"E") ("c"/"C") ("t"/"T") ("i"/"I") ("d"/"D")

A 'dialectId' keyword uses the text "DIALECTID" (case insensitive).

**BS/LS:** dialectIdSet = (dialectIdSet / dialectId)

**BS/LS:** dialectAliasFilter = dialect ws booleanComparisonOperator ws (dialectAlias / dialectAliasSet)

A dialectAlias filter starts with the 'dialect' keyword, followed by a boolean comparison operator, and either a single dialect alias or a set of dialect aliases.

**BS/LS:** dialect = ("d"/"D") ("i"/"I") ("a"/"A") ("l"/"L") ("e"/"E") ("c"/"C") ("t"/"T")

A 'dialect' keyword uses the text "DIALECT" (case insensitive).

**BS/LS:** dialectAlias = alpha *( dash / alpha / integerValue)

A dialect alias consists of a single alphanumeric character followed by zero or more alphanumeric characters, integer values or dashes.

**BS/LS:** dialectAliasSet = "(" ws dialectAlias [ws acceptabilitySet] *(mws dialectAlias [ws acceptabilitySet]) ws ")"

A dialect alias set is one or more dialect aliases followed by an optional acceptability set, separated by mandatory white space, and enclosed in brackets.
### BS/LS: dialectIdSet

```
(" ws eclConceptReference [ws acceptabilitySet] *(mws eclConceptReference [ws acceptabilitySet]) ws ")"
```

A dialect id set is one or more concept references followed by an optional acceptability set, separated by mandatory white space, and enclosed in brackets.

### BS/LS: acceptabilitySet

```
acceptabilityConceptReferenceSet / acceptabilityTokenSet
```

An acceptability set specifies the acceptabilities that a matching description must have in the language reference set specified by the preceding dialect filter. An acceptability set is either a set of one or more concept references or an acceptabilityToken set.

### BS/LS: acceptabilityConceptReferenceSet

```
(" ws eclConceptReference *(mws eclConceptReference) ws ")"
```

An acceptability concept reference set is a set of one or more references to concepts that are a < 900000000000511003 |Acceptability|.

### BS/LS: acceptabilityTokenSet

```
(" ws acceptabilityToken *(mws acceptabilityToken) ws ")"
```

An acceptability token set is one or more acceptability tokens, separated by mandatory whitespace, and enclosed in brackets.

### BS/LS: acceptabilityToken

```
acceptable / preferred
```

An acceptability token is either an acceptable token and a preferred token.

### BS: acceptable

```
("a"/"A") ("c"/"C") ("e"/"E") ("p"/"P") ("t"/"T")
```

An acceptable token uses the text "ACCEPT" (case insensitive). In the long syntax, the text "Acceptable" (case insensitive) may be used instead.

### BS: preferred

```
("p"/"P") ("r"/"R") ("e"/"E") ("f"/"F") ("d"/"D")
```

A preferred token uses the text "PREFER" (case insensitive). In the long syntax, the text "Preferred" (case insensitive) may be used instead.

### BS/LS: conceptFilterConstraint

```
{{" ws (c" / "C") ws conceptFilter *(ws "," ws conceptFilter) ws "]"
```

A concept filter constraint is a set of one or more concept references, separated by mandatory whitespace, and enclosed in brackets.
A concept filter constraint is a constraint used to filter the concepts in the result set, according to whether or not the concept matches the given conditions. A concept filter constraint is always enclosed in double curly braces. Within these braces, it starts with the letter ‘C’ followed by one or more constraint filters.

**BS/LS:** `conceptFilter = definitionStatusFilter / moduleFilter / effectiveTimeFilter / activeFilter`

A concept filter is either a definition status filter, a module filter, an effective time filter or an active filter.

**BS/LS:** `definitionStatusFilter = definitionStatusIdFilter / definitionStatusTokenFilter`

A definition status filter is constraint that either filters the results of a query, based on each concept's definition status identifier or a token.

**BS/LS:** `definitionStatusIdFilter = definitionStatusIdKeyword ws booleanComparisonOperator ws (subExpressionConstraint / eclConceptReferenceset)`

A definition status filter is a constraint that filters the results of a query, based on whether or not each concept's definition status matches a given identifier. The filter starts with the keyword "definitionStatusId", followed by a boolean comparison operator and either a subexpression constraint or a set of concept references that are a subtype of 900000000000444006 | Definition status[^59].

**BS/LS:** `definitionStatusIdKeyword = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") ("t"/"T") ("u"/"U") ("s"/"S")`

The definition status id keyword is the text "definitionStatusId" (in any combination of upper or lower case).

**BS/LS:** `definitionStatusTokenFilter = definitionStatusKeyword ws booleanComparisonOperator ws (definitionStatusToken / definitionStatusTokenSet)`

A definition status filter is a constraint that filters the results of a query, based on whether or not each concept's definition status matches a given token.

**BS/LS:** `definitionStatusKeyword = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") ("t"/"T") ("u"/"U") ("s"/"S")`

The definition status keyword is the text "definitionStatus" (in any combination of upper or lower case).

[^59]: http://snomed.info/id/900000000000444006
### BS/LS: definitionStatusToken = primitiveToken / definedToken

A definition status token is either a primitive token or a defined token.

### BS/LS: definitionStatusTokenSet = "(" ws definitionStatusToken *(mws definitionStatusToken) ws ")"

A definition status token set consists of one or more definition status tokens separated by mandatory white space and enclosed in brackets.

### BS/LS: primitiveToken = ("p"/"P") ("r"/"R") ("i"/"I") ("m"/"M") ("i"/"I") ("t"/"T") ("v"/"V") ("e"/"E")

A primitive token represents the definition status using the text "primitive" (in any combination of upper and lower case characters).

### BS/LS: definedToken = ("d"/"D") ("e"/"E") ("f"/"F") ("i"/"I") ("n"/"N") ("e"/"E") ("d"/"D")

A defined token represents the definition status using the text "defined" (in any combination of upper and lower case characters).

### BS/LS: moduleFilter = moduleIdKeyword ws booleanComparisonOperator ws (subExpressionConstraint / eclConceptReferenceSet)

A module filter is a constraint that filters the results of a query based on the module to which each concept belongs. The filter starts with the keyword "moduleId", followed by a boolean comparison operator and either a subexpression constraint or a set of concept references that are a subtype of Module.

### BS/LS: moduleIdKeyword = ("m"/"M") ("o"/"O") ("d"/"D") ("u"/"U") ("l"/"L") ("e"/"E") ("i"/"I") ("d"/"D")

The module id keyword is the text "moduleId" (in any combination of upper or lower case).

### BS/LS: effectiveTimeFilter = effectiveTimeKeyword ws timeComparisonOperator ws ( timeValue / timeValeSet )

An effective time filter is a constraint that filters the results of a query based on the effective time assigned to each concept.
BS/LS: effectiveTimeKeyword = ("e"/"E") ("f"/"F") ("f"/"F") ("e"/"E") ("c"/"C") ("t"/"T") ("i"/"I") ("v"/"V") ("e"/"E") ("t"/"T") ("i"/"I") ("m"/"M") ("e"/"E")

The effective time keyword is the text "effectiveTime" (in any combination of upper or lower case).

BS/LS: timeValue = QM [ year month day ] QM

A time value is a 8 digit string that represents the year, month and day of a specific date.

BS/LS: timeValueSet = "(" ws timeValue *(mws timeValue) ws ")"

A time value set consists of one or more time values separated by mandatory white space and enclosed in brackets.

BS/LS: year = digitNonZero digit digit digit

A year is a 4 digit string starting with a non-zero digit.

BS/LS: month = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" / "11" / "12"

A month is a 2 digit string from "01" to "12" that represents a specific month of the year (e.g. "01" represents January)

BS/LS: day = "01" / "02" / "03" / "04" / "05" / "06" / "07" / "08" / "09" / "10" / "11" / "12" / "13" / "14" / "15" / "16" / "17" / "18" / "19" / "20" / "21" / "22" / "23" / "24" / "25" / "26" / "27" / "28" / "29" / "30" / "31"

A day is a 2 digit string from "01" to "31" that represents a specific day within a month of a year.

BS/LS: activeFilter = activeKeyword ws booleanComparisonOperator ws activeValue

An active filter is a constraint that filters the results of a query based on the active status of each concept

BS/LS: activeKeyword = ("a"/"A") ("c"/"C") ("t"/"T") ("i"/"I") ("v"/"V") ("e"/"E")

The active keyword is the text "active" (in any combination of upper or lower case).

BS/LS: activeValue = activeTrueValue / activeFalseValue
An active value represents the active status of a concept, and is either true (i.e. the concept is active) or false (i.e. the concept is inactive).

BS/LS: `activeTrueValue = "1" / "true"`

An active true value is a value that represents an active concept. This value is either "1" or "true".

BS/LS: `activeFalseValue = "0" / "false"`

An active false value is a value that represents an inactive concept. This value is either "0" or "false".

BS/LS: `memberFilterConstraint = \[ \{"m" / "M") ws memberFilter *(ws "," ws memberFilter) ws "}\]`

A member filter constraint is a constraint used to filter the rows in one or more result sets, according to values of particular fields. A member filter constraint is always surrounded by double curly braces. Within these braces, it starts with the letter 'M' followed by one or more member filters.

BS/LS: `memberFilter = moduleFilter / effectiveTimeFilter / activeFilter / memberFieldFilter`

A member filter is either a module filter, an effective time filter, an active filter, or a member field filter.

BS/LS: `memberFieldFilter = refsetFieldName ws (expressionComparisonOperator ws subExpressionConstraint / numericComparisonOperator ws "#" numericValue / stringComparisonOperator ws (typedSearchTerm / typedSearchTermSet) / booleanComparisonOperator ws booleanValue / timeComparisonOperator ws (timeValue / timeValueSet) )`

A member field filter always has three parts - (1) the reference set field name, (2) a comparison operator, and (3) the criteria on which to match the field's value. If the refset field is of type SNOMED CT concept, then an expression comparison operator is used, followed by a subexpression constraint. If the refset field is a numeric type, then a numeric comparison operator is used, followed by a hash symbol ("#") and a numeric value. If the refset field is of type string, then a string comparison operator is used, followed by a typed search term or a typed search term set. If the refset field is of type boolean, then a boolean comparison operator is used, followed by a boolean value. And if the refset field is of type dateTime, then a time comparison operator is used, followed by a time value or time value set.

BS/LS: `historySupplement = \[\{"ws "+" ws historyKeyword [ historyProfileSuffix / ws historySubset ] ws "}\]"`
A history supplement augments the results of the expression constraint with relevant inactive concepts. A history supplement is always surrounded by double curly braces. Within these braces, it starts with a plus symbol (i.e. "+"), followed by the history keyword. The history keyword is optionally followed by either a profile suffix, or a history subset.

**BS/LS:** `historyKeyword` = ("h"/"H") ("i"/"I") ("s"/"S") ("t"/"T") ("o"/"O") ("r"/"R") ("y"/"Y")

The history keyword is the word "HISTORY" (case insensitive).

**BS/LS:** `historyProfileSuffix` = historyMinimumSuffix / historyModerateSuffix / historyMaximumSuffix

A history profile suffix is either the suffix for history minimum, history moderate or history maximum.

**BS/LS:** `historyMinimumSuffix` = ("-"/"_")("m"/"M") ("i"/"I") ("n"/"N")

The history minimum suffix is ".MIN" (case insensitive). The suffix may start with either a hyphen (i.e. ".") or an underscore (i.e. ".").

**BS/LS:** `historyModerateSuffix` = ("-"/"_")("m"/"M") ("o"/"O") ("d"/"D")

The history moderate suffix is ".MOD" (case insensitive). The suffix may start with either a hyphen (i.e. ".") or an underscore (i.e. ".").

**BS/LS:** `historyMaximumSuffix` = ("-"/"_")("m"/"M") ("a"/"A") ("x"/"X")

The history maximum suffix is ".MAX" (case insensitive). The suffix may start with either a hyphen (i.e. ".") or an underscore (i.e. ".").

**BS/LS:** `historySubset` = (" ws expressionConstraint ws ")

A history subset is an expression constraint that defines a set of historical association reference sets, surrounded by round brackets. Only descendants of 900000000000522004 | Historical association reference set may be included in a history subset.

**BS/LS:** `numericValue` = ["-"/"+"](decimalValue / integerValue)

A numeric value is either an integer or a decimal. Positive numbers optionally start with a plus sign ("+"), while negative integers begin with a minus sign ("-").

**BS/LS:** `stringValue` = 1*(anyNonEscapedChar / escapedChar)

---

63 [http://snomed.info/id/900000000000522004](http://snomed.info/id/900000000000522004)
A string value includes one or more of any printable ASCII characters enclosed in quotation marks. Quotes and backslash characters within the string must be preceded by the escape character ("\").

**BS/LS:** `integerValue` = `digitNonZero` * `digit` / `zero`

An integer value is either starts with a non-zero digit followed by zero to many additional digits, or is the integer zero itself.

**BS/LS:** `decimalValue` = `integerValue` "." `1*digit`

A decimal value starts with an integer. This is followed by a decimal point and one to many digits.

**BS/LS:** `booleanValue` = `true` / `false`

A boolean value is either true or false.

**BS/LS:** `true` = ("t"/"T") ("r"/"R") ("u"/"U") ("e"/"E")

A boolean value of true is represented by the word "true" (case insensitive).

**BS/LS:** `false` = ("f"/"F") ("a"/"A") ("l"/"L") ("s"/"S") ("e"/"E")

A boolean value of false is represented by the word "false" (case insensitive).

**BS/LS:** `nonNegativeIntegerValue` = (`digitNonZero` * `digit`) / `zero`

A non-negative integer value (i.e. positive integers or zero), without a preceding plus sign ("+").

**BS/LS:** `sctId` = `digitNonZero` 5*17( `digit` )

A SNOMED CT id is used to represent an attribute id or a concept id. The initial digit may not be zero. The smallest number of digits is six, and the maximum is 18.

**BS/LS:** `ws` = *( `SP` / `HTAB` / `CR` / `LF` / `comment` )

---

64 https://confluence.ihtsdotools.org/display/DOCGLOSS/concept
Optional whitespace characters (space, tab, carriage return, linefeed or a comment) are ignored everywhere in the expression except:

1. Whitespace within a conceptId is an error.
   
   **Note:** Whitespace before or after the last digit of a valid Identifier is ignored.
2. Non-consecutive spaces within a term are treated as a significant character of the term.
   
   **Note:** Whitespace before the first or after the last non-whitespace character of a term is ignored
3. Whitespace within the quotation marks of a concrete value is treated as a significant character.

\[
\text{BS/LS: } \text{mws} = 1^* ( \text{SP} / \text{HTAB} / \text{CR} / \text{LF} / \text{comment} )
\]

Mandatory whitespace (i.e. space, tab, carriage return, linefeed or a comment) is required after certain keywords, including "And" and "Or".

\[
\text{BS/LS: } \text{comment} = "/" * ( \text{nonStarChar} / \text{starWithNonLSlash} ) "/"
\]

A comment, which provides additional human-readable details about the expression constraint. Comments begin with a forward slash directly followed by a star (i.e. "/"') and end with a star directly followed by a forward slash (i.e. "/"').

\[
\text{BS/LS: } \text{nonStarChar} = \text{SP} / \text{HTAB} / \text{CR} / \text{LF} / \%x21-29 / \%x2B-7E / \text{UTF8-2} / \text{UTF8-3} / \text{UTF8-4}
\]

A character that is not a star (i.e. not %x2A).

\[
\text{BS/LS: } \text{starWithNonLSlash} = \%x2A \text{nonLSlash}
\]

A star (i.e. "\") followed by a character that is not a forward slash (i.e. not "/").

\[
\text{BS/LS: } \text{nonLSlash} = \text{SP} / \text{HTAB} / \text{CR} / \text{LF} / \%x21-2E / \%x30-7E / \text{UTF8-2} / \text{UTF8-3} / \text{UTF8-4}
\]

A character that is not a forward slash (i.e. not "/").

\[
\text{BS/LS: } \text{SP} = \%x20
\]

Space character.

\[
\text{BS/LS: } \text{HTAB} = \%x09
\]
<table>
<thead>
<tr>
<th><strong>BS/LS:</strong></th>
<th><strong>Character</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CR</strong> = %x0D</td>
<td>Tab character.</td>
<td>Carriage return character.</td>
</tr>
<tr>
<td><strong>LF</strong> = %x0A</td>
<td>Carriage return character.</td>
<td>BS/LS: Carriage return character.</td>
</tr>
<tr>
<td><strong>QM</strong> = %x22</td>
<td>Quotation mark character.</td>
<td>BS/LS: Quotation mark character.</td>
</tr>
<tr>
<td><strong>BS</strong> = %x5C</td>
<td>Back slash</td>
<td>BS represents the backslash character &quot;&quot;.</td>
</tr>
<tr>
<td><strong>star</strong> = %x2A</td>
<td>Asterisk</td>
<td>BS/LS: Asterisk character.</td>
</tr>
<tr>
<td><strong>digit</strong> = %x30-39</td>
<td>Any digit 0 through 9.</td>
<td>BS/LS: Digit character.</td>
</tr>
<tr>
<td><strong>zero</strong> = %x30</td>
<td>The digit 0.</td>
<td>BS/LS: Zero character.</td>
</tr>
<tr>
<td><strong>digitNonZero</strong> = %x31-39</td>
<td>Digits 1 through 9, but excluding 0.</td>
<td>BS/LS: Digit non-zero character.</td>
</tr>
<tr>
<td><strong>nonwsnonpipe</strong> = %x21-7B / %x7D-7E / UTF8-2 / UTF8-3 / UTF8-4</td>
<td></td>
<td>BS/LS: Non ws non pipe character.</td>
</tr>
</tbody>
</table>

---

6a https://confluence.ihtsdotools.org/display/DOCGLOSS/concept+identifier
Non whitespace (and non pipe) includes printable ASCII characters (these are also valid UTF8 characters encoded as one octet) and also includes all UTF8 characters encoded as 2-3- or 4-octet sequences. It excludes space (which is %x20) and the pipe character ".

BS/LS: anyNonEscapedChar = SP / HTAB / CR / LF / %x20-21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4

anyNonEscapedChar includes any printable ASCII characters which do not need to be preceded by an escape character (i.e. "\"). This includes valid UTF8 characters encoded as one octet and all UTF8 characters encoded as 2, 3 or 4 octet sequences. It does, however, exclude the quotation mark ("), and the backslash (\). See RFC 3629 (UTF-8), a transformation format of ISO 10646 authored by the Network Working Group).

BS/LS: escapedChar = BS QM / BS BS

The double quotation mark and the backslash character must both be escaped within a string-based concrete value by preceding them with a back slash.

BS/LS: escapedWildChar = BS QM / BS BS / BS star

An escapedWildChar is one of the characters that must be escaped in a wildcard search term (i.e. " or \ or *), preceded by a backslash (i.e. \). The character sequence is therefore either " or \ or *.

BS/LS: nonwsNonEscapedChar = %x21 / %x23-5B / %x5D-7E / UTF8-2 / UTF8-3 / UTF8-4

A nonwsNonEscapedChar is any printable ASCII, UTF8-2, UTF8-3 or UTF8-4 character, excluding double quotes ("), backslash (\), and space ( )

BS/LS: alpha = %x41-5A / %x61-7A

An alpha is any uppercase or lowercase character from "A" to "Z" (and "a" to "z") inclusive.

BS/LS: dash = %x2D

A dash is a hyphen (i.e. ".-".

BS/LS: UTF8-2 = %xC2-DF UTF8-tail

UTF8 characters encoded as 2-octet sequences.

69 https://confluence.ihtsdotools.org/display/DOCRELFMT/UTF-8
70 https://confluence.ihtsdotools.org/display/DOCGLOSS/transformation
71 https://confluence.ihtsdotools.org/display/DOCGLOSS/ISO
5.4 5.4 Order of Operation

This section explains the correct order of operation for unary operators, binary operators, filters and supplements.

5.4.1 Unary Operators

Unary operators (e.g. descendantOf, descendantOrSelfOf, ancestorOf, ancestorOrSelfOf, memberOf) are applied from inside to out (i.e. from right to left). For example, when the following expression constraint is processed, the memberOf operator is applied first to the Example problem list concepts reference set, and then the descendants of the referenced components are determined.

\[
< \wedge 700043003 \mid \text{Example problem list concepts reference set}\]

5.4.2 Binary Operators

Whenever potential ambiguity in binary operator precedence may occur, round brackets must be used to clearly disambiguate the order in which these operators are applied. For example, the following expression constraint is not valid:

\[
< 19829001 \mid \text{Disorder of lung} \quad \text{OR} \quad 700043003 \mid \text{Example problem list concepts reference set} \quad \text{MINUS} \quad 450976002 \mid \text{Disorders and diseases reference set for GP/FP reason for encounter}\]

And must be expressed using brackets, as either:

\[
< 19829001 \mid \text{Disorder of lung} \quad \text{OR} \quad 700043003 \mid \text{Example problem list concepts reference set} \quad \text{MINUS} \quad 450976002 \mid \text{Disorders and diseases reference set for GP/FP reason for encounter}\]

---

72 http://snomed.info/id/700043003
73 http://snomed.info/id/19829001
74 http://snomed.info/id/700043003
75 http://snomed.info/id/450976002
5. Syntax Specification

When multiple exclusion operators (i.e. ‘minus’) are applied, brackets are similarly required. For example, the following expression constraint is not valid:

\(< 19829001 | \text{Disorder of lung} | 76 \text{ OR } ^{700043003 | \text{Example problem list concepts reference set} | 77 } \text{ MINUS } ^{450976002 | \text{Disorders and diseases reference set for GP/FP reason for encounter} | 78 }\) 

or:

\(< 19829001 | \text{Disorder of lung} | 79 \text{ OR } ^{700043003 | \text{Example problem list concepts reference set} | 80 } \text{ MINUS } ^{450976002 | \text{Disorders and diseases reference set for GP/FP reason for encounter} | 81 }\)

When multiple exclusion operators (i.e. ‘minus’) are applied, brackets are similarly required. For example, the following expression constraint is not valid:

\(< 19829001 | \text{Disorder of lung} | 82 \text{ MINUS } ^{700043003 | \text{Example problem list concepts reference set} | 83 } \text{ MINUS } ^{450976002 | \text{Disorders and diseases reference set for GP/FP reason for encounter} | 84 }\) 

And must be expressed using brackets, as either:

\(< 19829001 | \text{Disorder of lung} | 85 \text{ MINUS } ^{700043003 | \text{Example problem list concepts reference set} | 86 } \text{ MINUS } ^{450976002 | \text{Disorders and diseases reference set for GP/FP reason for encounter} | 87 }\)

or:

\(< 19829001 | \text{Disorder of lung} | 88 \text{ MINUS } ^{700043003 | \text{Example problem list concepts reference set} | 89 } \text{ MINUS } ^{450976002 | \text{Disorders and diseases reference set for GP/FP reason for encounter} | 90 }\)

However, when only a single binary operator is used, or when all binary operators are either conjunction (i.e. ‘and’) or disjunction (i.e. ‘or’), brackets are not required. For example, all of the following expression constraints are valid without brackets:

\(\text{57 http://snomed.info/id/19829001} \) 
\(\text{76 http://snomed.info/id/700043003} \) 
\(\text{77 http://snomed.info/id/450976002} \) 
\(\text{78 http://snomed.info/id/19829001} \) 
\(\text{79 http://snomed.info/id/700043003} \) 
\(\text{80 http://snomed.info/id/450976002} \) 
\(\text{81 http://snomed.info/id/19829001} \) 
\(\text{82 http://snomed.info/id/700043003} \) 
\(\text{83 http://snomed.info/id/450976002} \) 
\(\text{84 http://snomed.info/id/19829001} \) 
\(\text{85 http://snomed.info/id/700043003} \) 
\(\text{86 http://snomed.info/id/450976002} \) 
\(\text{87 http://snomed.info/id/19829001} \) 
\(\text{88 http://snomed.info/id/700043003} \) 
\(\text{89 http://snomed.info/id/450976002} \)
< 19829001 | Disorder of lung | Example problem list concepts reference set

< 19829001 | Disorder of lung | Example problem list concepts reference set

< 19829001 | Disorder of lung | Example problem list concepts reference set

< 19829001 | Disorder of lung | Example problem list concepts reference set

Please note that unary operators are always applied before binary operators.

5.4.3 Filter Constraints

Filter constraints (e.g. concept, description, or member filters) apply only to the sub-expression constraint part that is directly to the left of the filter.

For example, the following expression constraint will apply the term filter to only the descendants or self of 415582006 | Stenosis. This expression constraint will match descendants of 404684003 | Clinical finding with

---

91 http://snomed.info/id/19829001
92 http://snomed.info/id/700043003
93 http://snomed.info/id/19829001
94 http://snomed.info/id/700043003
95 http://snomed.info/id/19829001
96 http://snomed.info/id/700043003
97 http://snomed.info/id/19829001
98 http://snomed.info/id/700043003
99 http://snomed.info/id/450976002
100 http://snomed.info/id/19829001
101 http://snomed.info/id/700043003
102 http://snomed.info/id/450976002
103 http://snomed.info/id/415582006
104 http://snomed.info/id/404684003

---

5. Syntax Specification – 58
a finding site that is a descendant or self of 39057004 | Pulmonary valve structure106, and an associated morphology that is any descendant or self of 415582006 | Stenosis106 which has a description matching the term "insufficiency". Therefore, the concept 123801008 | Heart valve stenosis and regurgitation (disorder)107 will match this expression constraint because it has the associated morphology 708027006 | Valvular stenosis with valvular insufficiency108.

To apply a filter to a sub-expression constraint, which includes a refinement or binary operators, the subexpression must be enclosed in brackets. For example, the following expression constraint will find all the descendants of clinical finding, with a finding site that is a descendant or self of 39057004 | Pulmonary valve structure114 and an associated morphology that is a descendant or self of 415582006 | Stenosis115, and will then match only those clinical finding concepts that have a description that matches the term "insufficiency". Therefore, the concept 123801008 | Heart valve stenosis and regurgitation (disorder)116 will not match this expression constraint, as it does not have a description that matches the term "insufficiency".

5.4.4 History Supplements

History supplements are applied only to the sub-expression constraint part that is directly to its left, after any filter constraints on this sub-expression constraint part have been applied.

For example, the following expression constraint will match all concepts that are both an active member of the 734139008 | Anatomy structure and part association reference set122 and also either an active member of the 734138000 | Anatomy structure and entire association reference set123 or an inactive concept associated with an

---

105 http://snomed.info/id/39057004
106 http://snomed.info/id/415582006
107 http://snomed.info/id/123801008
108 http://snomed.info/id/708027006
109 http://snomed.info/id/404684003
110 http://snomed.info/id/363698007
111 http://snomed.info/id/39057004
112 http://snomed.info/id/116676008
113 http://snomed.info/id/415582006
114 http://snomed.info/id/39057004
115 http://snomed.info/id/415582006
116 http://snomed.info/id/123801008
117 http://snomed.info/id/404684003
118 http://snomed.info/id/363698007
119 http://snomed.info/id/39057004
120 http://snomed.info/id/116676008
121 http://snomed.info/id/415582006
122 http://snomed.info/id/734139008
123 http://snomed.info/id/734138000
active member of the Anatomy structure and entire association reference set \(^{124}\) via the 900000000000527005 \(\)SAME AS association reference set\(^{125}\). Because all active members of the Anatomy structure and part association reference set\(^{126}\) are active, there will be no inactive concepts in the result set.

\[
\begin{align*}
^\text{734139008} & \text{[Anatomy structure and part association reference set]} & \text{\(\text{127}\)} \\
\text{\text{AND}} & ^\text{734138000} & \text{[Anatomy structure and entire association reference set]} & \text{\(\text{128}\)} \\
\{\{ & + \text{HISTORY} & ( & 900000000000527005 & \text{[SAME AS association reference set]} & \text{\(\text{129}\)}) \}} \\
\end{align*}
\]

To apply the history supplement to the entire sub-expression constraint above, the sub-expression constraint must be enclosed in round brackets. For example, the following expression constraint will match concepts that are both members of the Anatomy structure and part association reference set\(^{130}\) and also members of the Anatomy structure and entire association reference set\(^{131}\); and it will also match on any inactive concept that is associated via a 900000000000527005 \(\)SAME AS association reference set\(^{132}\) to a member of both reference sets.

\[
\begin{align*}
{\{ & + \text{HISTORY} & ( & 900000000000527005 & \text{[SAME AS association reference set]} & \text{\(\text{129}\)}) \}} \\
\end{align*}
\]

5.5 Character Collation for Term Filters

This page is published as Draft for Trial Use. The recommendations on this page will be reviewed and may be updated following feedback from implementation experiences.

To promote consistency between implementations of ECL, the following collation principles are recommended:

- **Search and match** - The default behaviour of a system implementing ECL queries with term filters, is to use locale-specific asymmetric searching at the secondary comparison strength level - as specified in the Unicode Technical Standard #10 - Unicode Collation Algorithm\(^{136}\). This means that the search is, by default, case insensitive, with some language-specific character normalization behaviour.

  - **Asymmetric**: Asymmetric searches require characters in the query that are unmarked (i.e. the 'base letters') to match characters in the target that are either marked or unmarked (with the same base letter). However, a character in the query that is marked will only match a character in the target that is marked in the same way.

---

\(^{124}\) http://snomed.info/id/734138000
\(^{125}\) http://snomed.info/id/900000000000527005
\(^{126}\) http://snomed.info/id/734139008
\(^{127}\) http://snomed.info/id/734138000
\(^{128}\) http://snomed.info/id/734138000
\(^{129}\) http://snomed.info/id/900000000000527005
\(^{130}\) http://snomed.info/id/734139008
\(^{131}\) http://snomed.info/id/734138000
\(^{132}\) http://snomed.info/id/900000000000527005
\(^{133}\) http://snomed.info/id/734139008
\(^{134}\) http://snomed.info/id/734138000
\(^{135}\) http://snomed.info/id/900000000000527005
\(^{136}\) http://www.unicode.org/reports/tr10/#Asymmetric_Search_Secondary
• **Secondary strength**: Searches with a strength of secondary will only consider level 1 differences (e.g. "d" vs "e") and level 2 differences (e.g. "e" vs "é" in English). However, level 3 differences (e.g. "e" vs "E") are not considered. This provides the same effect as queries being case insensitive. For example, in English, "e" in the query will match both "e" and "E" in the target; and "E" in the query will similarly match both "e" and "E" in the target.

• **Language customizations** - Locale-based customizations of the standard are specified in the [Unicode Common Locale Data Repository (CLDR)](https://unicode.org/reports/tr35/tr35-collation.html#Root_Collation). The unicode CLDR specifies the characters that are considered to be 'marked' variants of the base letters, identical base letters, and/or contractions in each specified language. The description terms in the substrate should be indexed separately for each language supported. For example, the following search behaviour is expected in the locales specified below.

  - In **English**, **Swedish** and **Danish**, the following search behaviour is expected:

    **Note**: No customizations are made in these 3 locales for the characters used in these searches. Therefore, the [CLDR root collation order](https://unicode.org/reports/tr35/tr35-collation.html#Root_Collation) is used.

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>resume</td>
<td>resume, Resume, RESUME, résumé, rùsumè, Rùsumé, RÈSUMÈ, ...</td>
<td>-</td>
</tr>
<tr>
<td>Resume</td>
<td>resume, Resume, RESUME, résumé, rùsumè, Rùsumé, RÈSUMÈ, ...</td>
<td>-</td>
</tr>
<tr>
<td>résumé</td>
<td>résumé, Résumé, RÈSUMÈ, ...</td>
<td>resume, Resume, RESUME, ...</td>
</tr>
<tr>
<td>Résumé</td>
<td>résumé, Résumé, RÈSUMÈ, ...</td>
<td>resume, Resume, RESUME, ...</td>
</tr>
</tbody>
</table>

  - In **English**, the following search behaviour is expected (based on the [CLDR 'en' locale](https://github.com/unicode-org/cldr/blob/master/common/collation/en.xml), which uses the CLDR root collation order):

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>sjogren</td>
<td>sjogren, Sjogren, SJOgren, sjögren, Sjögren, SJOgren, sjögren, Sjögren, SJOgren, ...</td>
<td>-</td>
</tr>
<tr>
<td>sjögren</td>
<td>sjögren, Sjögren, SJOgren, sjögren, Sjögren, SJOgren, sjögren, Sjögren, SJOgren, ...</td>
<td>sjogren, Sjogren, SJOgren, sjögren, Sjögren, SJOgren, ...</td>
</tr>
<tr>
<td>Angstrom</td>
<td>angstrom, Angstrom, ANGSTROM, ångström, Ångström, ÅNGSTRÖM, ångstrøm, Ångstrøm, ÅNGSTRØM, ...</td>
<td>ångstrœm, Ångstrœm, ÅNGSTRÆM, ...</td>
</tr>
</tbody>
</table>

---

137 http://cldr.unicode.org/index/cldr-spec/collation-guidelines
138 https://unicode.org/reports/tr35/tr35-collation.html#Root_Collation
140 https://unicode.org/reports/tr35/tr35-collation.html#Root_Collation
<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
</table>
| Ångström    | ångström, Ångström, ÅNGSTRÖM, ... | angstrom, Angstrom, ANGSTROM, ångström, Ångström, ÅNGSTRÖM, ...
| Ångström    | ångström, Ångström, ÅNGSTRÖM, ... | angstrom, Angstrom, ANGSTROM, ångström, Ångström, ÅNGSTRÖM, ...
| aangström   | aangström, Aangström, AANGSTRÖM, ... | angstrom, Angstrom, ANGSTROM, ångström, Ångström, ÅNGSTRÖM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstrøm, Ångstrøm, ÅNGSTRØM, ...

In Swedish, the following search behaviour is expected (based on the customizations in the CLDR 'sv' locale\(^{141}\)):

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
</table>
| sjogren     | sjogren, Sjogren, SJOGREN, ... | sjögren, Sjögren, SJÖGREN, sjogren, Sjogren, SJØGREN, ...
| sjögren     | sjögren, Sjögren, SJÖGREN, sjögren, Sjögren, SJØGREN, ... | sjogren, Sjogren, SJOGREN, ...
| Angstrom    | angstrom, Angstrom, ANGSTROM, ... | ångström, Ångström, ÅNGSTRÖM, ångström, Ångström, ÅNGSTRÖM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstrøm, Ångstrøm, ÅNGSTRØM, ...
| Ångström    | ångström, Ångström, ÅNGSTRÖM, ångström, Ångström, ÅNGSTRÖM, ångstrøm, Ångstrøm, ÅNGSTRØM, ... | angstrom, Angstrom, ANGSTROM, aangström, Aangström, AANGSTRÖM, ...
| Ångstrøm    | ångstrøm, Ångstrøm, ÅNGSTRØM, ... | angstrom, Angstrom, ANGSTROM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstrøm, Ångstrøm, ÅNGSTRØM, ...

\(^{141}\)https://github.com/unicode-org/cldr/blob/master/common/collation/sv.xml
<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>aangstrøm</td>
<td>aangstrøm, Aangstrøm, AANGSTRØM, ...</td>
<td>angstrom, Angstrom, ANGSTROM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstroem, Ångstroem, ÅNGSTRØM, ...</td>
</tr>
</tbody>
</table>

And in **Danish**, the following search behaviour is expected (based on the customizations in the CLDR 'da' locale\(^{142}\)):

<table>
<thead>
<tr>
<th>Search Term</th>
<th>Target Matches</th>
<th>Target does NOT Match</th>
</tr>
</thead>
<tbody>
<tr>
<td>sjogren</td>
<td>sjogren, Sjogren, SJOGREN, ...</td>
<td>sjøgren, Sjøgren, SJØGREN, sjøgren, Sjøgren, SJØGREN, ...</td>
</tr>
<tr>
<td>sjøgren</td>
<td>sjøgren, Sjøgren, SJØGREN, ...</td>
<td>sjogren, Sjogren, SJOGREN, sjøgren, Sjøgren, SJØGREN, ...</td>
</tr>
<tr>
<td>Angstrøm</td>
<td>angstrom, Angstrom, ANGSTROM, ...</td>
<td>ångstrøm, Ångstrøm, ÅNGSTRØM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstroem, Ångstroem, ÅNGSTRØM, ...</td>
</tr>
<tr>
<td>Ångstrøm</td>
<td>ångstrøm, Ångstrøm, ÅNGSTRØM, aangstrøm, Aangstrøm, AANGSTRØM, ...</td>
<td>angstrom, Angstrom, ANGSTROM, ångstrøm, Ångstrøm, ÅNGSTRØM, ångstroem, Ångstroem, ÅNGSTRØM, ...</td>
</tr>
<tr>
<td>Ångstroem</td>
<td>ångstroem, Ångstroem, ÅNGSTRØM, aangstroem, Aangstroem, AANGSTRØM, ...</td>
<td>angstrom, Angstrom, ANGSTROM, ångstroem, Ångstroem, ÅNGSTRØM, ...</td>
</tr>
<tr>
<td>aangstroem</td>
<td>ångstroem, Ångstroem, ÅNGSTRØM, aangstroem, Aangstroem, AANGSTRØM, ...</td>
<td>angstrom, Angstrom, ANGSTROM, ångstroem, Ångstroem, ÅNGSTRØM, ...</td>
</tr>
</tbody>
</table>

\(^{142}\)https://github.com/unicode-org/cldr/blob/master/common/collation/da.xml
6 6. Examples

The examples in this section illustrate the syntaxes proposed in Section 5 (see page 21).

- 6.1 Simple Expression Constraints (see page 64)
- 6.2 Refinements (see page 72)
- 6.3 Cardinality (see page 82)
- 6.4 Conjunction and Disjunction (see page 90)
- 6.5 Exclusion and Not Equals (see page 97)
- 6.6 Constraint Comments (see page 99)
- 6.7 Nested Expression Constraints (see page 100)
- 6.8 Description Filters (see page 105)
- 6.9 Concept Filters (see page 116)
- 6.10 Member Filters (see page 121)
- 6.11 History Supplements (see page 124)
- 6.12 Top and Bottom (see page 129)

6.1 6.1 Simple Expression Constraints

The simplest type of expression constraint contains a single concept optionally preceded by an expression constraint operator and/or membership function. Expression constraint operators (e.g. descendant of) traverse the hierarchical relationships in SNOMED CT to return the set of concepts that are directly or transitively connected to the focus concept. Membership functions return the set of concepts referenced by a reference set.

In this section we consider some of these simple examples.

6.1.1 Self

If no expression constraint operator or membership function is applied, the expression constraint is satisfied only by the specified concept. For example, the expression constraint below is satisfied only by the concept 404684003 | Clinical finding143.

![404684003 | Clinical finding144]

Please note that this expression constraint is equivalent to an expression that looks the same but is written in SNOMED CT Compositional Grammar145.

6.1.2 Descendant of

A single 'less than' sign (i.e. "<") indicates that the expression constraint is satisfied by all descendants of the specified concept. The expression constraint below evaluates to the set of all subtypes (both direct children and transitive subtypes) of 404684003 | Clinical finding146, using the brief syntax.

---

143 http://snomed.info/id/404684003
144 http://snomed.info/id/404684003
145 http://snomed.org/scg
146 http://snomed.info/id/404684003
Using the long syntax, the above expression constraint may be represented as:

\[
\text{descendantOf} \ 404684003 \ | \text{Clinical finding}^{148}
\]

The descendantOf function is primarily used on concepts, which serve as the 'grouper' of a set of values (e.g. Clinical finding (finding)\(^{149}\), Severities (qualifier value)\(^{150}\), Unit (qualifier value)\(^{151}\)). The descendantOf function may also be applied to other concepts, or to nested expression constraints (as discussed in 6.7 Nested Expression Constraints (see page 100)).

### 6.1.3 Descendant or Self of

Two consecutive 'less than' signs (i.e. "<<") indicates that the expression constraint is satisfied by all descendants of the specified concept plus the specified concept itself. The expression constraint below evaluates to the set of descendants of \(73211009\ | \text{Diabetes mellitus}\)\(^{152}\), plus the concept \(73211009\ | \text{Diabetes mellitus}\)\(^{153}\) itself.

\[
<< \ 73211009 \ | \text{Diabetes mellitus}\]^{154}
\]

Using the long syntax, the above expression constraint may be represented as:

\[
\text{descendantOrSelfOf} \ 73211009 \ | \text{Diabetes mellitus}\]^{155}
\]

The descendantOrSelfOf function is primarily used for attribute values, which refer to a specific clinical value (e.g. \(73211009\ | \text{Diabetes mellitus}\)\(^{156}\), \(73761001\ | \text{Colonoscopy}\)\(^{157}\), \(385055001\ | \text{Tablet dose form}\)\(^{158}\)), but any specialization of this value is also acceptable. The descendantOrSelfOf function may also be applied to other concepts, or to nested expression constraints (as discussed in 6.7 Nested Expression Constraints (see page 100)).

---

147 http://snomed.info/id/404684003  
148 http://snomed.info/id/404684003  
149 http://snomed.info/id/404684003  
150 http://snomed.info/id/272141005  
151 http://snomed.info/id/258666001  
152 http://snomed.info/id/73211009  
153 http://snomed.info/id/73211009  
154 http://snomed.info/id/73211009  
155 http://snomed.info/id/73211009  
156 http://snomed.info/id/73211009  
157 http://snomed.info/id/73761001  
158 http://snomed.info/id/385055001
6.1.4 Child of

A 'less than' sign directly followed by an exclamation mark (i.e. "<!") indicates that the expression constraint is satisfied by the set of proximal children of the specified concept. The children of a concept are those concepts that are the source of a non-redundant 116680003 | is a relationship whose target is the given concept. The expression constraint below, represented using the brief syntax, evaluates to the set of immediate children of the concept 404684003 | Clinical finding.

<! 404684003 | Clinical finding

Using the long syntax, the above expression constraint may be represented as:

childOf 404684003 | Clinical finding

Please note that the childOf function may only be executed against a finite and pre-classified substrate, and that the results of this function are specific to the substrate used. The childOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints).

6.1.5 Child or Self of

Two consecutive 'less than' signs directly followed by an exclamation mark (i.e. "<<!") indicates that the expression constraint is satisfied by the set of proximal children of the specified concept plus the specified concept itself. The children of a concept are those concepts that are the source of a non-redundant 116680003 | is a relationship whose target is the given concept. The expression constraint below, represented using the brief syntax, evaluates to the set of immediate children of the concept 404684003 | Clinical finding, plus the concept 404684003 | Clinical finding itself.

<<! 404684003 | Clinical finding

Using the long syntax, the above expression constraint may be represented as:

childOrSelfOf 404684003 | Clinical finding

Please note that the childOrSelfOf function may only be executed against a finite and pre-classified substrate, and that the results of this function are specific to the substrate used. The childOrSelfOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints).
6.1.6 Ancestor of

A single ‘greater than’ sign (i.e. “>”) indicates that the expression constraint is satisfied by all ancestors of the specified concept. The expression constraint below, using the brief syntax, evaluates to the set of all supertypes (both direct parents and transitive supertypes) of 40541001 | Acute pulmonary edema:

> 40541001 | Acute pulmonary edema

Using the long syntax, the above expression constraint may be represented as:

ancestorOf 40541001 | Acute pulmonary edema

Please note that the ancestorOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints).

6.1.7 Ancestor or Self of

Two consecutive ‘greater than’ signs (i.e. “>>”) indicates that the expression constraint is satisfied by all ancestors of the specified concept plus the specified concept itself. The expression constraint below evaluates to the set of ancestors of 40541001 | Acute pulmonary edema, plus the concept 40541001 | Acute pulmonary edema.

>> 40541001 | Acute pulmonary edema

Using the long syntax, the above expression constraint may be represented as:

ancestorOrSelfOf 40541001 | Acute pulmonary edema

Please note that the ancestorOrSelfOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints).

6.1.8 Parent of

A ‘greater than’ sign directly followed by an exclamation mark (i.e. “>!”) indicates that the expression constraint is satisfied by the set of proximal parents of the specified concept. The parents of a concept are those concepts that

---

168 http://snomed.info/id/40541001
169 http://snomed.info/id/40541001
170 http://snomed.info/id/40541001
171 http://snomed.info/id/40541001
172 http://snomed.info/id/40541001
173 http://snomed.info/id/40541001
174 http://snomed.info/id/40541001
are the target of a non-redundant | is a | relationship whose source is the given concept. The expression constraint below, represented using the brief syntax, evaluates to the set of immediate parents of the concept 40541001 | Acute pulmonary edema | .

\[ >! 40541001 \mid \text{Acute pulmonary edema} \]

Using the long syntax, the above expression constraint may be represented as:

\[ \text{parentOf } 40541001 \mid \text{Acute pulmonary edema} \]

Please note that the parentOf function should only be executed against a finite and pre-classified substrate, and that the results of this function are specific to the substrate used. The parentOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints (see page 100)).

### 6.1.9 Parent or Self of

Two consecutive 'greater than' signs directly followed by an exclamation mark (i.e. ">>!") indicates that the expression constraint is satisfied by the set of proximal parents of the specified concept plus the specified concept itself. The parents of a concept are those concepts that are the target of a non-redundant | is a | relationship whose source is the given concept. The expression constraint below, represented using the brief syntax, evaluates to the set of immediate parents of the concept 40541001 | Acute pulmonary edema | , plus the concept 40541001 | Acute pulmonary edema | itself.

\[ >>! 40541001 \mid \text{Acute pulmonary edema} \]

Using the long syntax, the above expression constraint may be represented as:

\[ \text{parentOrSelfOf } 40541001 \mid \text{Acute pulmonary edema} \]

Please note that the parentOrSelfOf function should only be executed against a finite and pre-classified substrate, and that the results of this function are specific to the substrate used. The parentOrSelfOf function may also be applied to nested expression constraints (as discussed in 6.7 Nested Expression Constraints (see page 100)).
6.1.10 Member of

The memberOf function (by default) evaluates to the set of concepts that are referenced by the given reference set (i.e. the set of referencedComponentIds). Please note that this function may be applied only to reference sets whose referenced components are concepts. The SNOMED CT Expression Constraint Language does not support use of the memberOf function on reference sets whose referencedComponents are not concepts (i.e. descriptions or relationships).

The memberOf function is represented in the brief syntax using a 'caret' character (i.e. "^") and is usually followed by a single concept id for a concept-based reference set. For example, the following expression constraint is satisfied by the set of concepts which are members of 700043003 | Example problem list concepts reference set:

\[^700043003\]

Using the long syntax the expression constraint is represented as:

```
memberOf 700043003 | Example problem list concepts reference set
```

The expression constraints above both return the values in the referencedComponentId field of the given reference sets. However, it is also possible to specify one or more fields, whose values will be returned, by including the relevant field names in square brackets after the memberOf operator ("^" or "memberOf"). For example, the following expression constraint is equivalent to the brief syntax example above.

\[^\{referencedComponentId\}\] 700043003 | Example problem list concepts reference set

The value of other fields can also be returned by an expression constraint. For example, the following expression constraint will return the targetComponentId values (i.e. the 'Entire' anatomy concepts) from the 734138000 | Anatomy structure and entire association reference set.

\[^\{targetComponentId\}\] 734138000 | Anatomy structure and entire association reference set

It is also possible to return the values of more than one field in a reference set (e.g. a pair or tuple of values). For example, to return both the source and target of the 816210007 | SNOMED CT to MedDRA simple map reference set, the following expression constraint could be used:

\[^\{referencedComponentId, mapTarget\}\] 816210007 | SNOMED CT to MedDRA simple map reference set

---

184 http://snomed.info/id/700043003
185 http://snomed.info/id/700043003
186 http://snomed.info/id/700043003
187 http://snomed.info/id/700043003
188 http://snomed.info/id/734138000
189 http://snomed.info/id/734138000
190 http://snomed.info/id/816210007
191 http://snomed.info/id/816210007
To return all the non-metadata fields of a referenceSet (i.e. the values of the referencedComponentId and additional fields), a wildcard (i.e. "**" in the brief syntax, and "**" or "Any" in the long syntax) can be used. For example, the following expression constraint will return the referencedComponentId, mapGroup, mapPriority, mapRule, mapAdvice, mapTarget and correlationId for each row of the 447562003 | ICD-10 complex map reference set.

\[
^\ast [\ast] \ 447562003 \ 	ext{ICD-10 complex map reference set}
\]

For more information on the use of reference set field names in ECL, please refer to Appendix E - Reference Set Fields (see page 206).

Please note that it is also possible to apply the memberOf function to an expression constraint that returns a set of concept-based reference set concepts. For more information, please refer to 6.7 Nested Expression Constraints (see page 100).

And for information about applying filter constraints to reference set members, please refer to 6.10 Member Filters (see page 121).

6.1.11 Any

A single 'star' (i.e. "**") may be used in the place of a concept reference to represent any concept in the substrate. The expression constraint below evaluates to the set of all concepts in the given substrate.

\[
^\ast
\]

Using the long syntax, the above expression constraint may also be represented as:

\[
\text{ANY}
\]

This wildcard character (or 'ANY' keyword) may be used anywhere within an expression constraint that a concept reference may be used. In many situations, the wildcard is equivalent to the following expression constraint:

\[
<< \ 138875005 \ 	ext{SNOMED CT concept}
\]

However, some situations exist in which the concept 138875005 | SNOMED CT concept is not included in the substrate, and therefore cannot be used to determine the full set of concepts available. In other cases, the single character wildcard may serve as a convenient shortcut for the longer expression constraint above.

Please note that the following three expression constraints evaluate to the same set of concepts:

---

192 http://snomed.info/id/447562003
193 http://snomed.info/id/447562003
194 http://snomed.info/id/138875005
195 http://snomed.info/id/138875005
The two expression constraints below evaluate to all concepts in the substrate minus the root concept:

\[ < * \]
\[ <! * \]

And the two expression constraints below evaluate to all non-leaf concepts in the substrate:

\[ > * \]
\[ >! * \]

Finally, the expression constraint below evaluates to all concepts that are referenced by any reference set in the substrate:

\[ ^ * \]

### 6.1.12 Alternate Identifier

If an alternate identifier exists for a concept, in an identifier scheme other than SNOMED CT, then this can be used to refer to that concept (see page 0) (see 4.2.4 Identifier File Specification). For example there may be SNOMED CT concept for "Type of hemoglobin in blood at point in time" with an alternate identifier with the scheme alias "LOINC" and the code "54486-6":

\[ LOINC#54486-6 \]
The alternate identifier code can be surrounded by quotes.
The alternate identifier code can be surrounded by quotes.
The alternate identifier code can be surrounded by double quotes. Quoting may be necessary to ensure correct parsing of the constraint depending on the characters used in the code:

```
LOINC"#54486-6"
```

**Please note:** ECL is always evaluated against SNOMED CT concepts. When you run a query using alternate identifiers from another code system only concepts that are in the SNOMED CT representation of that other code system will be returned. For example the descendants of a concept in the SNOMED CT version of a code system is likely to be different from the code system of the alternate identifier.

---

6.2 6.2 Refinements

In this section, we illustrate how the set of matching concepts can be filtered using one or more simple attribute refinements. For more information on applying refinements to nested expression constraints, using nested attribute names and using nested attribute values, please refer to [6.7 Nested Expression Constraints](see page 100).

6.2.1 Attributes

Adding an attribute refinement to an expression constraint restricts the set of valid clinical meanings to only those whose defining attributes satisfy the given refinement condition. Similarly to [SNOMED CT Compositional Grammar](197), attribute refinements are placed after a 'colon' (i.e. ":") in the expression constraint.

The example below is satisfied only by the set of lung disorders, which have an associated morphology that is exactly equal to 79654002 | Edema(198).

---

197 http://snomed.org/scg
198 http://snomed.info/id/79654002
Using the long syntax, the above expression is represented as:

```
descendantOf 19829001 | Disorder of lung | 116676008 | Associated morphology = 79654002 | Edema
```

In many cases, however, the value of the matching attribute is allowed to be either the concept itself, or a descendant of that concept. In these cases, the descendantOrSelfOf operator is used prior to the concept representing the attribute value. For example, the expression constraint below (in brief and long syntaxes respectively) is satisfied only by the set of lung disorders, which have an associated morphology of `Edema` or any descendant of `Edema`.

```
< 19829001 | Disorder of lung : 116676008 | Associated morphology = << 79654002 | Edema
```

```
descendantOf 19829001 | Disorder of lung : 116676008 | Associated morphology = descendantOrSelfOf 79654002 | Edema
```

When more than one attribute is defined in an expression constraint, the attributes are normally separated by a comma. A comma between two attributes indicates a conjunction and implies that both attribute conditions must be true. For example, the expression constraint below, written in brief syntax, is satisfied only by the set of clinical findings, which have both a finding site of `Pulmonary valve structure` (or a subtype of `Pulmonary valve structure`) and an associated morphology of 'stenosis' (or a subtype of 'stenosis').
6. Examples

Please note that attribute refinements may also be used when the focus concept is '*' (or ANY). The following expression constraint represents any concept that has a 246075003 | Causative agent attribute whose value is 387517004 | Paracetamol.

\[
* : 246075003 | \text{Causative agent} = 387517004 | \text{Paracetamol}
\]

Using the long syntax, the above expression may also be represented as:

\[
\text{ANY} : 246075003 | \text{Causative agent} = 387517004 | \text{Paracetamol}
\]

6.2.2 Attribute Groups

Similarly to SNOMED CT compositional grammar, expression constraints use curly braces (i.e. "{}") to indicate that a set of attributes should be grouped together in an attribute group. For example, the expression constraint below is satisfied only by the set of clinical findings with an associated morphology of 'stenosis' (or descendant) at the finding site 'pulmonary valve structure' (or descendant), and also with an associated morphology of 'hypertrophy' (or descendant) at the finding site 'right ventricular structure' (or descendant).

\[
< 404684003 | \text{Clinical finding} :
\{ 363698007 | \text{Finding site} = 39057004 | \text{Pulmonary valve structure}, 116676008 | \text{Associated morphology} = 415582006 | \text{Stenosis} \},
\]
6. Examples

Using the 'long syntax', the above expression constraint is represented as:

```
descendantOf 404684003 |Clinical finding |
{ 363698007 |Finding site | = descendantOrSelfOf 39057004 |Pulmonary valve structure |
  116676008 |Associated morphology |
  , 116676008 |Associated morphology |
  = descendantOrSelfOf 415582006 |Stenosis |
  , 363698007 |Finding site |
  = descendantOrSelfOf 53085002 |Right ventricular structure |
  , 116676008 |Associated morphology |
  = descendantOrSelfOf 56246009 |Hypertrophy |
}
```

6.2.3 Attribute Constraint Operators

In some cases, an attribute concept has subtypes or supertypes in the concept model attribute hierarchy. Where this occurs, it is possible to indicate that an attribute condition may be satisfied by matching one of the subtypes or supertypes of the given attribute. This is done adding a constraint operator directly before the attribute name concept. For example, the expression constraint below will not only match clinical findings that are Associated with a type of Edema, but also those that are Due to, After or the Causative agent of a type of Edema. This result occurs because the 47429007 Associated with attribute concept has three subtypes: After, 255234002, 246075003, Causative agent, 42752001, and Due to.

```
<< 404684003 |Clinical finding |
<< 47429007 |Associated with |
<< 267038008 |Edema |
```

---

231 http://snomed.info/id/363698007
232 http://snomed.info/id/53085002
233 http://snomed.info/id/116676008
234 http://snomed.info/id/56246009
235 http://snomed.info/id/404684003
236 http://snomed.info/id/363698007
237 http://snomed.info/id/39057004
238 http://snomed.info/id/116676008
239 http://snomed.info/id/415582006
240 http://snomed.info/id/363698007
241 http://snomed.info/id/53085002
242 http://snomed.info/id/116676008
243 http://snomed.info/id/56246009
244 http://snomed.info/id/404684003
245 http://snomed.info/id/47429007
246 http://snomed.info/id/267038008
247 http://snomed.info/id/42752001
248 http://snomed.info/id/255234002
249 http://snomed.info/id/246075003
250 http://snomed.info/id/267038008
251 http://snomed.info/id/47429007
252 http://snomed.info/id/255234002
253 http://snomed.info/id/246075003
254 http://snomed.info/id/42752001
255 http://snomed.info/id/404684003
256 http://snomed.info/id/47429007
257 http://snomed.info/id/267038008
This expression constraint is represented in the long syntax as:

```
descendantOrSelfOf 404684003 |Clinical finding| 258:
descendantOrSelfOf 47429007 |Associated with| 259 = descendantOrSelfOf 267038008 |Edema| 260
```

Similarly, the expression constraint below will not only match clinical findings that are | Due to | 261 a type of | Edema | 262, but also those that have an | Associated with | 263 relationship whose value is a type of | Edema | 264.

```
<< 404684003 |Clinical finding| 265:
>> 246075003 |Causative agent| 266 = << 267038008 |Edema| 267
```

This expression constraint is represented in the long syntax as:

```
descendantOrSelfOf 404684003 |Clinical finding| 268:
ancestorOrSelfOf 246075003 |Causative agent| 269 = descendantOrSelfOf 267038008 |Edema| 270
```

### 6.2.4 Concrete Values

The revised SNOMED CT Compositional Grammar allows attributes to be given concrete values (e.g. Strings, Integers, Decimal, Boolean). The SNOMED CT Expression Constraint Language supports the ability to compare these attribute values with a given concrete value.

When numeric concrete values (i.e. Integers and Decimals) are compared, a set of standard mathematical operators may be used. These mathematical operators are:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>!=</td>
<td>Not equals</td>
</tr>
</tbody>
</table>

---

258 http://snomed.info/id/404684003  
259 http://snomed.info/id/47429007  
260 http://snomed.info/id/267038008  
261 http://snomed.info/id/42752001  
262 http://snomed.info/id/267038008  
263 http://snomed.info/id/47429007  
264 http://snomed.info/id/267038008  
265 http://snomed.info/id/404684003  
266 http://snomed.info/id/246075003  
267 http://snomed.info/id/267038008  
268 http://snomed.info/id/404684003  
269 http://snomed.info/id/246075003  
270 http://snomed.info/id/267038008  
271 http://snomed.org/scg  
272 http://snomed.org/ecl
### 6. Examples

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt;</code></td>
<td>Less than</td>
</tr>
<tr>
<td><code>&lt;=</code></td>
<td>Less than or equals</td>
</tr>
<tr>
<td><code>&gt;</code></td>
<td>Greater than</td>
</tr>
<tr>
<td><code>&gt;=</code></td>
<td>Greater than or equals</td>
</tr>
</tbody>
</table>

Please note that the 'not equals' operator may alternatively be represented as "" and "not =" (case insensitive) in the long syntax.

The following expression constraint is satisfied by oral medicinal products, which contain amoxicillin and have a presentation strength greater than or equal to 250 mg.

\[
< 763158003 | Medicinal product (product) 273 :
\]

\[
\text{411116001} | \text{Has manufactured dose form (attribute)} 274 = << \text{385268001} | \text{Oral dose form (dose form)} 275 ,
\]

\[
\{ << \text{127489000} | \text{Has active ingredient (attribute)} 276 = << \text{372687004} | \text{Amoxicillin (substance)} 277 ,
\]

\[
\text{1142135004} | \text{Has presentation strength numerator value (attribute)} 278 \geq \#250,
\]

\[
\text{732945000} | \text{Has presentation strength numerator unit (attribute)} 279 = \#258684004 \text{ milligram (qualifier value)} 280 \}
\]

Please note that, as per SNOMED CT Compositional Grammar, integer and decimal values are preceded by a hash character (e.g. "\#500"), while string values are surrounded by double quotes (e.g. "PANADOL").

To find those oral amoxicillin products that have a strength between 250 and 800 mg (inclusive), the following expression constraint may be used:

\[
< 763158003 | Medicinal product (product) 281 :
\]

\[
\text{411116001} | \text{Has manufactured dose form (attribute)} 282 = << \text{385268001} | \text{Oral dose form (dose form)} 283 ,
\]

\[
\{ << \text{127489000} | \text{Has active ingredient (attribute)} 284 = << \text{372687004} | \text{Amoxicillin (substance)} 285 ,
\]

\[
\text{1142135004} | \text{Has presentation strength numerator value (attribute)} 286 \geq \#250,
\]

\[
\text{1142135004} | \text{Has presentation strength numerator value (attribute)} 287 \leq \#800,
\]

---

273 http://snomed.info/id/763158003
274 http://snomed.info/id/411116001
275 http://snomed.info/id/385268001
276 http://snomed.info/id/127489000
277 http://snomed.info/id/372687004
278 http://snomed.info/id/1142135004
279 http://snomed.info/id/732945000
280 http://snomed.info/id/258684004
281 http://snomed.info/id/763158003
282 http://snomed.info/id/411116001
283 http://snomed.info/id/385268001
284 http://snomed.info/id/127489000
285 http://snomed.info/id/372687004
286 http://snomed.info/id/1142135004
287 http://snomed.info/id/1142135004
Concrete values of type string and boolean may also be included in an expression constraint, and compared using an 'equal to' (i.e. "=") or 'not equal to' (i.e. "!=") operator. The following expression constraint is satisfied only by products with a product name equal to "PANADOL" (see page 0).

\[
\text{< 373873005 | Pharmaceutical / biologic product}: \\
3460481009 | Has product name = "PANADOL"}
\]

The following expression constraint is satisfied only by products that are in the national benefit scheme (of the given country) (see page 0).

\[
\text{< 373873005 | Pharmaceutical / biologic product}: \\
859999999102 | Is in national benefit scheme = TRUE}
\]

### 6.2.5 Reverse Attributes

In most cases, an attribute refinement is satisfied by those concepts, which are the source concept of a defining relationship whose destination concept matches the attribute value. In some cases, however, it may be necessary to select the destination concept of a relationship and constrain the source concept to a given attribute value. To achieve this, an expression constraint indicates that an attribute is to be constrained in the reverse order using a 'reverse flag' (see page 0). In the brief syntax, the reverse flag is represented by preceding the name of the attribute with a capital letter 'R'.

For example, the expression constraint below finds the set of anatomical structures, which are the finding site of a type of bone fracture (e.g. 85050009 | Humerus, 71341001 | Femur).

\[
\text{< 91723000 | Anatomical structure}: \\
R 363698007 | Finding site = < 125605004 | Fracture of bone}
\]

The above expression constraint is represented in the long syntax as:

\[
732945000 | \text{Has presentation strength numerator unit (attribute)} = 258684004 | \text{milligram (qualifier value)}
\]

288 http://snomed.info/id/732945000
289 http://snomed.info/id/258684004
290 http://snomed.info/id/373873005
291 http://snomed.info/id/3460481009
292 http://snomed.info/id/373873005
293 http://snomed.org/fictid#859999999102
294 http://snomed.info/id/85050009
295 http://snomed.info/id/71341001
296 http://snomed.info/id/91723000
297 http://snomed.info/id/363698007
298 http://snomed.info/id/125605004

6. Examples – 78
6.2.6 Dotted Attributes

An alternative way of representing 'reversed attributes' is by applying the *dot notation* to represent them as *dotted attributes*. Using this alternative notation, "< 123456 123456 | X | Y >" represents the set of attribute values (i.e. destination concepts) of the attribute "Y" for descendants or self of concept "X". This is therefore equivalent to "* : R 234567 234567 | Y | X >" using the reverse flag.

The previous expression constraint (which finds the set of body sites for any subtype of bone fracture) has an equivalent representation using the 'dot notation' of:

< 91723000 | Anatomical structure | AND (< 125605004 | Fracture of bone | . 363698007 | Finding site | )

Because all values of 363698007 | Finding site must be < 91723000 | Anatomical structure (according to the SNOMED CT concept model), this expression constraint can be further simplified to:

< 125605004 | Fracture of bone . 363698007 | Finding site

The next example finds the set of substances, which are an active ingredient in any product containing amoxicillin.

< 105590001 | Substance : R < 127489000 | Has active ingredient = < 27658006 | Product containing amoxicillin

This expression constraint is represented in the long syntax as:

descendantOf 91723000 | Anatomical structure : reverseOf 363698007 | Finding site = descendantOf 125605004 | Fracture of bone
6. Examples

```
descendantOf 105590001 | Substance | 105590001 : ReverseOf descendantOrSelfOf 127489000 | Has active ingredient = descendantOf 27658006 | Product containing amoxicillin
```

An equivalent way of representing this constraint, using the ‘dot notation’ is:

```
< 105590001 | Substance | AND (< 27658006 | Product containing amoxicillin | 127489000 | Has active ingredient )
```

or (using the SNOMED CT concept model to simplify):

```
< 27658006 | Product containing amoxicillin | . | 127489000 | Has active ingredient
```

When more than one dot attribute is used in sequence, the dot notation is evaluated sequentially from left to right. For example, the following expression constraint represents the set of Finding sites of any concept that is Associated with a subtype of Disorder of lung.

```
< 19829001 | Disorder of lung | . < 47429007 | Associated with | 363698007 | Finding site
```

This expression constraint is evaluated by first finding the descendants of Disorder of lung, then finding the set of attribute values for these concepts (with an attribute type that is any subtype of Associated with), and then from these attribute value concepts, finding the value of any Finding sites attribute. Please note that the expression constraint above (with no brackets) is equivalent to the one below (with brackets added).

```
((< 19829001 | Disorder of lung | . | 47429007 | Associated with | 363698007 | Finding site
```

---

317 http://snomed.info/id/105590001
318 http://snomed.info/id/127489000
319 http://snomed.info/id/27658006
320 http://snomed.info/id/105590001
321 http://snomed.info/id/27658006
322 http://snomed.info/id/127489000
323 https://confluence.ihtsdotools.org/display/DOCGLOSS/SNOMED+CT+concept+model
324 http://snomed.info/id/27658006
325 http://snomed.info/id/127489000
326 http://snomed.info/id/363698007
327 http://snomed.info/id/47429007
328 http://snomed.info/id/19829001
329 http://snomed.info/id/19829001
330 http://snomed.info/id/47429007
331 http://snomed.info/id/363698007
332 http://snomed.info/id/19829001
333 http://snomed.info/id/47429007
334 http://snomed.info/id/363698007
335 http://snomed.info/id/19829001
336 http://snomed.info/id/47429007
337 http://snomed.info/id/363698007
6.2.7 Any Attribute Name and Value

A single 'star' (i.e. "*") may be used in the place of an attribute name to represent any attribute in the substrate. The expression constraint below evaluates to the set of clinical findings which have any attribute with a value of 79654002 |Edema|.

\[
< 404684003 |\text{Clinical finding} | * = 79654002 |\text{Edema} |
\]

Using the long syntax, the above expression constraint may also be represented as:

\[
\text{descendantOf} 404684003 |\text{Clinical finding} : \text{ANY} = 79654002 |\text{Edema} |
\]

The 'star' symbol (i.e. "*") may also be used to represent any attribute value (either with or without refinement). The following expression constraint evaluates to the set of clinical findings which have an associated morphology (with any value).

\[
< 404684003 |\text{Clinical finding} | 116676008 |\text{Associated morphology} | * = *
\]

Using the long syntax, the above expression constraint may also be represented as:

\[
\text{descendantOf} 404684003 |\text{Clinical finding} : 116676008 |\text{Associated morphology} | \text{ANY} = \text{ANY}
\]

Concrete values of type string are case sensitive and compared using the Unicode Collation Algorithm (http://www.unicode.org/reports/tr10/).

Please note that the concept 859999999102 |Is in national benefit scheme| is a fictitious attribute used here to illustrate boolean values.

It should be noted that using a reversed attribute joined by conjunction with a non-reversed attribute may lead to a nonsensical constraint (e.g. "<<a: {b=c, Rd=e}"). This is because the target concept of the reversed attribute must be matched with the source concept of the non-reversed attribute, which in turn must be the same as the source concept of the reversed attribute (being in the same attribute group). This would require the reversed attribute to be reflexive (i.e. the source and target concept to be the same).
6.3 6.3 Cardinality

6.3.1 Attribute cardinality

6.3.1.1 Overview

To support use cases such as the SNOMED CT concept model and terminology binding, expression constraints may
constrain the number of times an attribute can be included in an expression or concept definition represented in
the SNOMED CT distribution view (see page 0). This is done using a cardinality constraint, which consists of a minimum
cardinality and a maximum cardinality (written "[X..Y]"). A minimum cardinality of X constrains the valid clinical
meanings to those which have at least (i.e. >=) X non-redundant attributes that match the given attribute
criteria. A maximum cardinality of Y constrains the valid clinical meanings to those which have at most (i.e. <=) Y
non-redundant attributes that match the given attribute criteria. For example, a cardinality of "[1..5]"
indicates that all clinical meanings that satisfy the given expression constraint must have at least one and at most
five attributes that match the given attribute criteria.

The expression constraint below is satisfied only by products with one, two or three active ingredients.

```
< 373873005 | Pharmaceutical / biologic product | [1..3] 127489000 | Has active ingredient | 105590001 | Substance
```

Using the long syntax, this expression constraint may be represented as:

```
descendantOf 373873005 | Pharmaceutical / biologic product | [1 to 3] 127489000 | Has active ingredient | 105590001 | Substance
```

The following expression constraint is satisfied only by products which have exactly one active ingredient:

```
< 373873005 | Pharmaceutical / biologic product | [1..1] 127489000 | Has active ingredient | 105590001 | Substance
```

---

347 http://snomed.info/id/373873005
348 http://snomed.info/id/127489000
349 http://snomed.info/id/105590001
350 http://snomed.info/id/373873005
351 http://snomed.info/id/127489000
352 http://snomed.info/id/105590001
353 http://snomed.info/id/373873005
354 http://snomed.info/id/127489000
355 http://snomed.info/id/105590001
Unconstrained Cardinalities

A minimum cardinality of '0' indicates that there is no constraint on the minimum number of attributes that may match the given attribute criteria. For example, the following expression constraint is satisfied only by products with at most one active ingredient (i.e. the maximum cardinality is '1' and the minimum cardinality is unconstrained).

\[
< 373873005 \mid \text{Pharmaceutical / biologic product}\ : [0..1] 127489000 \mid \text{Has active ingredient} = \leq 105590001 \mid \text{Substance}
\]

Using the long syntax, this may be represented as:

\[
\text{descendantOf} \ 373873005 \mid \text{Pharmaceutical / biologic product} : [0 \ to \ 1] 127489000 \mid \text{Has active ingredient} = \leq \text{descendantOf} 105590001 \mid \text{Substance}
\]

A maximum cardinality of '*' (or 'many') indicates that there is no constraint on the maximum number of attributes that may match the given attribute criteria. For example, the following expression constraint is satisfied only by products that have at least one active ingredient (i.e. the minimum cardinality is '1' and the maximum cardinality is unconstrained).

\[
< 373873005 \mid \text{Pharmaceutical / biologic product} : [1..*] 127489000 \mid \text{Has active ingredient} = \leq 105590001 \mid \text{Substance}
\]

Using the long syntax, this may be represented as:

\[
\text{descendantOf} \ 373873005 \mid \text{Pharmaceutical / biologic product} : [1 \ to \ many] 127489000 \mid \text{Has active ingredient} = \leq \text{descendantOf} 105590001 \mid \text{Substance}
\]

A cardinality of [0..*] should therefore never be used as this indicates that the given attribute is not being constrained in any way, and is therefore a redundant part of the expression constraint.

---

356 http://snomed.info/id/373873005
357 http://snomed.info/id/127489000
358 http://snomed.info/id/105590001
359 http://snomed.info/id/373873005
360 http://snomed.info/id/127489000
361 http://snomed.info/id/105590001
362 http://snomed.info/id/373873005
363 http://snomed.info/id/127489000
364 http://snomed.info/id/105590001
6.3.1.3 Default Cardinalities

The default cardinality of each attribute, where not explicitly stated, is [1..*]. Therefore, the following two expression constraints are equivalent.

\[
< 373873005 | \text{Pharmaceutical / biologic product} | 127489000 | \text{Has active ingredient} = < 105590001 | \text{Substance}
\]

\[
< 373873005 | \text{Pharmaceutical / biologic product} : 127489000 | \text{Has active ingredient} = < 105590001 | \text{Substance}
\]

6.3.1.4 Non-redundant Attributes

As mentioned above, only non-redundant defining attributes are included in the cardinality count. Therefore, the following postcoordinated expression:

\[
\{ 116676008 | \text{Associated morphology} = 72704001 | \text{Fracture} , 363698007 | \text{Finding site} = 299701004 | \text{Bone of forearm} , 363698007 | \text{Finding site} = 62413002 | \text{Bone structure of radius} \}
\]

will successfully satisfy the expression constraint:

\[
< 404684003 | \text{Clinical finding} : \{ 363698007 | \text{Finding site} = < 91723000 | \text{Anatomical structure} \}
\]
This is because 299701004 | Bone of forearm 384 is a supertype of 62413002 | Bone structure of radius 385 and therefore the attribute 363698007 | Finding site 386 = 299701004 | Bone of forearm 387 | Bone of forearm is redundant.

6.3.1.5 Attribute Cardinality in Groups
When the attributes to which cardinality are applied can be grouped, but braces are not used in the expression constraint, the cardinality constrains the number of times the attribute may be included in any attribute group. For example, the following expression constraint is satisfied by any clinical finding whose definition has two or more non-redundant finding sites, irrespective of which attribute group they are contained in.

\[ < 404684003 | \text{Clinical finding} 388 : [2..*] 363698007 | \text{Finding site} 389 = < 91723000 | \text{Anatomical structure} 390 \]

In contrast, when braces are placed around an attribute with a given cardinality, there must exist at least one attribute group for which the given cardinality is satisfied by attributes in that group. For example, the following expression constraint is satisfied by any clinical finding whose definition contains an attribute group with two or more non-redundant finding sites.

\[ < 404684003 | \text{Clinical finding} 391 : \{ [2..*] 363698007 | \text{Finding site} 392 = < 91723000 | \text{Anatomical structure} 393 \} \]

6.3.2 Attribute Group Cardinality
Minimum and maximum cardinalities may also be applied to attribute groups. A minimum attribute group cardinality of \( X \) constrains the valid clinical meanings to those which have at least (i.e. \( \geq \)) \( X \) non-redundant attribute groups that match the given attribute group criteria. A maximum cardinality of \( Y \) constrains the valid clinical meanings to those which have at most (i.e. \( \leq \)) \( Y \) non-redundant attribute groups that match the given attribute group criteria. For example, a cardinality of \( [1..2] \) indicates that all clinical meanings that satisfy the given expression constraint must have at least one and at most two attribute groups that match the given attribute group criteria.

The expression constraint below is satisfied only by products with one, two or three attribute groups, which each contain at least one active ingredient relationship.

384 http://snomed.info/id/299701004
385 http://snomed.info/id/62413002
386 http://snomed.info/id/363698007
387 http://snomed.info/id/299701004
388 http://snomed.info/id/404684003
389 http://snomed.info/id/363698007
390 http://snomed.info/id/91723000
391 http://snomed.info/id/404684003
392 http://snomed.info/id/363698007
393 http://snomed.info/id/91723000
6. Examples

6.3.2.1 Unconstrained Cardinalities

As with attribute cardinalities, a minimum cardinality of ‘0’ indicates that there is no constraint on the minimum number of attribute groups that may match the given attribute group criteria. For example, the following expression constraint is satisfied only by products with at most one attribute group containing an active ingredient relationship (i.e. the maximum attribute group cardinality is ‘1’ and the minimum attribute group cardinality is unconstrained).

< 373873005 | Pharmaceutical / biologic product | [0..1] { [1..*] 1274890000 | Has active ingredient = < 105590001 | Substance }>

Using the long syntax, this may be represented as:

descendantOf 373873005 | Pharmaceutical / biologic product : [0..1] { [1..many] 1274890000 | Has active ingredient = descendantOf 105590001 | Substance }

394 http://snomed.info/id/373873005
395 http://snomed.info/id/1274890000
396 http://snomed.info/id/105590001
397 http://snomed.info/id/373873005
398 http://snomed.info/id/1274890000
399 http://snomed.info/id/105590001
400 http://snomed.info/id/373873005
401 http://snomed.info/id/1274890000
402 http://snomed.info/id/105590001
403 http://snomed.info/id/373873005
404 http://snomed.info/id/1274890000
405 http://snomed.info/id/105590001
406 http://snomed.info/id/373873005
407 http://snomed.info/id/1274890000
408 http://snomed.info/id/105590001
A maximum cardinality of '*' (or 'many') indicates that there is no constraint on the maximum number of attribute groups that may match the given attribute group criteria. For example, the following expression constraint is satisfied only by products that have at least one attribute group containing an active ingredient relationship (i.e. the minimum attribute group cardinality is '1' and the maximum attribute group cardinality is unconstrained).

\[
< 373873005 \text{ | Pharmaceutical / biologic product}^{409} : \\
[1..*] \{ 127489000 \text{ | Has active ingredient}^{410} = < 105590001 \text{ | Substance}^{411} \}
\]

Using the long syntax, this may be represented as:

\[
descendantOf 373873005 \text{ | Pharmaceutical / biologic product}^{412} : \\
[1 to *] \{ 127489000 \text{ | Has active ingredient}^{413} = descendantOf 105590001 \text{ | Substance}^{414} \}
\]

A cardinality of [0..*] should therefore never be used as this indicates that the given attribute group is not being constrained in any way, and is therefore a redundant part of the expression constraint.

### 6.3.2.2 Default Cardinalities

As with attribute cardinality, the default attribute group cardinality, where not explicitly stated, is [1..*]. Therefore, the following four expression constraints are equivalent.

\[
< 373873005 \text{ | Pharmaceutical / biologic product}^{415} : \\
\{ 127489000 \text{ | Has active ingredient}^{416} = < 105590001 \text{ | Substance}^{417} \}
\]

\[
< 373873005 \text{ | Pharmaceutical / biologic product}^{418} : \\
\{ [1..*] 127489000 \text{ | Has active ingredient}^{419} = < 105590001 \text{ | Substance}^{420} \}
\]

\[
< 373873005 \text{ | Pharmaceutical / biologic product}^{421} : \\
[1..*] \{ 127489000 \text{ | Has active ingredient}^{422} = < 105590001 \text{ | Substance}^{423} \}
\]

---

409 http://snomed.info/id/373873005
410 http://snomed.info/id/127489000
411 http://snomed.info/id/105590001
412 http://snomed.info/id/373873005
413 http://snomed.info/id/127489000
414 http://snomed.info/id/105590001
415 http://snomed.info/id/373873005
416 http://snomed.info/id/127489000
417 http://snomed.info/id/105590001
418 http://snomed.info/id/373873005
419 http://snomed.info/id/127489000
420 http://snomed.info/id/105590001
421 http://snomed.info/id/373873005
422 http://snomed.info/id/127489000
423 http://snomed.info/id/105590001
6.3.2.3 Non-redundant Attribute Groups

As mentioned above, only non-redundant defining attributes are included in the cardinality count. Therefore, the following postcoordinated expression:

\[
< 404684003 | \text{Clinical finding} | \begin{cases}
363698007 | \text{Finding site} = 299701004 | \text{Bone of forearm},
363698007 | \text{Finding site} = 62413002 | \text{Bone structure of radius}.
\end{cases}
\]

will successfully satisfy the expression constraint:

\[
< 404684003 | \text{Clinical finding} | \begin{cases}
363698007 | \text{Finding site} = 91723000 | \text{Anatomical structure}.
\end{cases}
\]

This is because \(299701004 | \text{Bone of forearm}\) is a supertype of \(62413002 | \text{Bone structure of radius}\) and therefore the attribute group "\(\begin{cases}
363698007 | \text{Finding site} = 299701004 | \text{Bone of forearm}.
\end{cases}" is redundant.

6.3.2.4 Attribute and Attribute Group Cardinalities

Attribute cardinalities and attribute group cardinalities can be used together to achieve a combined effect. For example, to represent the set of clinical findings which have no attribute groups that contain two or more finding site attributes (in the same attribute group), the following expression constraint can be used:

\[
< 373873005 | \text{Pharmaceutical / biologic product} | \begin{cases}
1..* | 127489000 | \text{Has active ingredient} = < 105590001 | \text{Substance}.
\end{cases}
\]
6.3.3 Reverse Cardinalities

When a cardinality constraint is applied to a reversed refinement, it constrains the number of source concepts (matching the given criteria) for which each destination concept may be relevant attribute value.

For example, the following expression constraint represents the substances, which are the active ingredient of exactly three products.

< 105590001 | Substance : [3..3] R 127489000 | Has active ingredient = *

If this expression constraint was executed against a simplified substrate containing the following seven relationships:

<table>
<thead>
<tr>
<th>Source concept</th>
<th>Attribute</th>
<th>Destination concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>412458007</td>
<td>Orphenadrine + aspirin + caffeine</td>
<td>127489000</td>
</tr>
<tr>
<td>412458007</td>
<td>Orphenadrine + aspirin + caffeine</td>
<td>127489000</td>
</tr>
<tr>
<td>412458007</td>
<td>Orphenadrine + aspirin + caffeine</td>
<td>127489000</td>
</tr>
<tr>
<td>412096001</td>
<td>Aspirin + codeine</td>
<td>127489000</td>
</tr>
</tbody>
</table>

### 6.4 Conjunction and Disjunction

#### 6.4.1 Compound Expression Constraints

Expression constraints can be built up from smaller parts using conjunction (i.e. AND) and disjunction (i.e. OR). The simplest example of this is where the conjunction or disjunction is used between two simple expressions. For example, the following expression constraint is satisfied only by clinical findings which are both a disorder of the lung and an edema of the trunk. This gives the same result as a mathematical intersection between the set of \(19829001 \mid \text{Disorder of lung}\) \(^{468}\) descendants and the set of \(301867009 \mid \text{Edema of trunk}\) \(^{469}\) descendants.

\[
< 19829001 \mid \text{Disorder of lung}^{470} \quad \text{AND} \quad < 301867009 \mid \text{Edema of trunk}^{471}
\]

Please note that all keywords are case insensitive, so the following two expression constraints are equivalent to the above:

---

456 http://snomed.info/id/412096001
457 http://snomed.info/id/127489000
458 http://snomed.info/id/387494007
459 http://snomed.info/id/424102008
460 http://snomed.info/id/127489000
461 http://snomed.info/id/387517004
462 http://snomed.info/id/424102008
463 http://snomed.info/id/127489000
464 http://snomed.info/id/387458008
465 http://snomed.info/id/387458008
466 http://snomed.org/tig
467 http://snomed.org/tig
468 http://snomed.info/id/19829001
469 http://snomed.info/id/301867009
470 http://snomed.info/id/19829001
471 http://snomed.info/id/301867009
6. Examples

The next expression constraint is satisfied only by clinical findings which are either a disorder of the lung or an edema of the trunk. This gives the same result as a mathematical union of the set of descendants and the set of descendants. For this reason, an OR operator will usually allow more valid clinical meanings than an AND operator.

Conjunction and disjunction operators may also be combined with the use of the 'member of' function, as shown below:

This expression constraint is satisfied only by concepts that belong to the hierarchy and are also members of the . When more than one conjunction or more than one disjunction is used, round brackets can be optionally applied. For example, the following expression constraints are all valid and equivalent to each other:

References:
472 http://snomed.info/id/19829001
473 http://snomed.info/id/301867009
474 http://snomed.info/id/19829001
475 http://snomed.info/id/301867009
476 http://snomed.info/id/19829001
477 http://snomed.info/id/301867009
478 http://snomed.info/id/19829001
479 http://snomed.info/id/301867009
480 http://snomed.info/id/19829001
481 http://snomed.info/id/700043003
482 http://snomed.info/id/19829001
483 http://snomed.info/id/700043003
484 http://snomed.info/id/19829001
485 http://snomed.info/id/301867009
486 http://snomed.info/id/700043003
However, where a conjunction and disjunction are both used together, it is mandatory to use round brackets to disambiguate the meaning of the expression constraint. For example, the following expression constraint is not valid:

\[
< 19829001 | \text{Disorder of lung} \quad 493 \quad \text{AND} \quad < 301867009 | \text{Edema of trunk} \quad 494 \quad \text{OR} \\
\uparrow 700043003 | \text{Example problem list concepts reference set} \quad 495
\]

And must be expressed (depending on the intended meaning) as either:

\[
(< 19829001 | \text{Disorder of lung} \quad 496 \quad \text{AND} \quad < 301867009 | \text{Edema of trunk} \quad 497 \quad \text{OR} \\
\uparrow 700043003 | \text{Example problem list concepts reference set} \quad 498)
\]

Or as:

\[
< 19829001 | \text{Disorder of lung} \quad 499 \quad \text{AND} \quad (< 301867009 | \text{Edema of trunk} \quad 500 \quad \text{OR} \\
\uparrow 700043003 | \text{Example problem list concepts reference set} \quad 501)
\]

---

487 http://snomed.info/id/19829001  
488 http://snomed.info/id/301867009  
489 http://snomed.info/id/700043003  
490 http://snomed.info/id/19829001  
491 http://snomed.info/id/301867009  
492 http://snomed.info/id/700043003  
493 http://snomed.info/id/19829001  
494 http://snomed.info/id/301867009  
495 http://snomed.info/id/700043003  
496 http://snomed.info/id/19829001  
497 http://snomed.info/id/301867009  
498 http://snomed.info/id/700043003  
499 http://snomed.info/id/19829001  
500 http://snomed.info/id/301867009  
501 http://snomed.info/id/700043003
6.4.2 Attribute Conjunction and Disjunction

Conjunction and disjunction may be used within refinements in a variety of ways. The most common way of using these operators in a refinement is to define the conjunction or disjunction of individual attributes.

For example, the expression constraint below, in which the comma between the two attributes represents conjunction, is satisfied only by clinical findings which have both a finding site of pulmonary valve structure (or subtype) and an associated morphology of stenosis (or subtype).

```
< 404684003 | Clinical finding |
  36398007 | Finding site |
  39057004 | Pulmonary valve structure |
  116676008 | Associated morphology |
  415582006 | Stenosis |
```

This expression constraint can equivalently be expressed as:

```
< 404684003 | Clinical finding |
  36398007 | Finding site |
  39057004 | Pulmonary valve structure |
  116676008 | Associated morphology |
  415582006 | Stenosis |
  509 | AND |
```

The following example uses the disjunction operator (OR) to represent the disjunction of two attributes. This constraint is satisfied only by clinical findings which have either an associated morphology of ‘infarct’ (or subtype) or are due to a myocardial infarction (or subtype).

```
< 404684003 | Clinical finding |
  116676008 | Associated morphology |
  55641003 | Infarct |
  22298006 | Myocardial infarction |
```

When more than one conjunction or more than one disjunction is used in a refinement, round brackets can be optionally applied. For example, the following expression constraints are all valid and equivalent to each other:

```
< 404684003 | Clinical finding |
  36398007 | Finding site |
  39057004 | Pulmonary valve structure |
  116676008 | Associated morphology |
  415582006 | Stenosis |
```

References:

502 http://snomed.info/id/404684003
503 http://snomed.info/id/36398007
504 http://snomed.info/id/39057004
505 http://snomed.info/id/116676008
506 http://snomed.info/id/415582006
507 http://snomed.info/id/404684003
508 http://snomed.info/id/36398007
509 http://snomed.info/id/39057004
510 http://snomed.info/id/116676008
511 http://snomed.info/id/415582006
512 http://snomed.info/id/404684003
513 http://snomed.info/id/116676008
514 http://snomed.info/id/55641003
515 http://snomed.info/id/42752001
516 http://snomed.info/id/22298006

6. Examples – 93
However, where a conjunction and disjunction are both used together in a refinement, it is mandatory to use brackets to disambiguate the meaning of the expression constraint.

For example, the following expression constraint is *not* valid:
6.4.3 Attribute Group Conjunction and Disjunction

Similarly, conjunction and disjunction may be defined between attribute groups. The following expression constraint is satisfied only by clinical findings which either have a finding site of pulmonary valve structure (or subtype) and an associated morphology of stenosis (or subtype), OR have a finding site of right ventricular structure (or subtype) and an associated morphology of hypertrophy (or subtype).

And must be expressed (depending on the intended meaning) as either:

Or as:

---

538 http://snomed.info/id/404684003
539 http://snomed.info/id/363698007
540 http://snomed.info/id/39057004
541 http://snomed.info/id/116676008
542 http://snomed.info/id/415582006
543 http://snomed.info/id/445238008
544 http://snomed.info/id/404684003
545 http://snomed.info/id/363698007
546 http://snomed.info/id/39057004
547 http://snomed.info/id/116676008
548 http://snomed.info/id/415582006
549 http://snomed.info/id/445238008
550 http://snomed.info/id/404684003
551 http://snomed.info/id/363698007
552 http://snomed.info/id/39057004
553 http://snomed.info/id/116676008
554 http://snomed.info/id/415582006
555 http://snomed.info/id/445238008
6.4.4 Attribute Value Conjunction and Disjunction

Conjunction and disjunction can also be applied to attribute values. The example below is satisfied only by members of the adverse drug reactions reference set for GP/FP health issue, which have a causative agent that is either a subtype of pharmaceutical / biologic product or a subtype of substance.

```
^ 450990004 | [Adverse drug reactions reference set for GP/FP health issue] OR
  246075003 | [Causative agent] = (< 373873005 | [Pharmaceutical / biologic product] OR < 105590001 | Substance)
```

Similarly, attribute values can also use conjunction. The following expression constraint is satisfied only by clinical findings with an associated morphology whose value is both a subtype (or self) of ulcer and a subtype (or self) of hemorrhage.

```
< 404684003 | [Clinical finding] : 116676008 | [Associated morphology] =
  (<< 56208002 | [Ulcer] AND << 50960005 | [Hemorrhage])
```

For more information about nested attribute values and nested compound expression constraints, please refer to 6.7 Nested Expression Constraints (see page 100).
6.5 Exclusion and Not Equals

6.5.1 Exclusion of Simple Expressions

Exclusion is supported in the SNOMED CT Expression Constraint Language by the binary operator ‘MINUS’. Exclusion works in a similar manner to mathematical subtraction. For example, the following expression constraint returns the set of lung disorders which are not a descendant or self of edema of the trunk.

\[
\text{Disorder of lung} \setminus \text{Edema of trunk}
\]

Logically, this expression constraint takes the set of descendants of ‘disorder of lung’ and subtracts the set of descendants of ‘edema of trunk’. Please note that the keyword ‘MINUS’ is case insensitive.

Exclusion can also be applied to the membership of a reference set. For example, the following expression constraint returns the set of lung disorders which are not members of the cardiology reference set. That is, the set of descendants or self of ‘disorder of lung’ minus the set of members of the ‘cardiology reference set’.

\[
\text{Disorder of lung} \setminus \text{Example problem list concepts reference set}
\]

Please note that when more than one exclusion operator is used, or when an exclusion operator is used together with a conjunction or disjunction, round brackets must be used to disambiguate the intended meaning.

6.5.2 Exclusion of Attribute Values

Attribute values, represented by compound expression constraints, may also contain exclusions. When this occurs, the expression constraint is satisfied by any concept or expression which has at least one attribute (of the given type) whose value is satisfied by the compound constraint defined in the attribute value. For example, the expression constraint below represents the set of clinical findings, which have an associated morphology that is a descendant or self of ulcer and a descendant or self of hemorrhage, but not a descendant or self of obstruction.

\[
\text{Clinical finding} \setminus \text{Associated morphology} = \left( \text{Ulcer} \setminus \text{Hemorrhage} \right) \setminus \text{Obstruction}
\]
6.5.3 Not Equal to Attribute Value

It is also possible to simply state that an attribute value should not fall in a particular range. The example below is satisfied only by clinical findings which have an associated morphology that is not a descendant (or self) of obstruction.

\[
\text{< 404684003 | Clinical finding, 585 : 116676008 | Associated morphology, 586 \neq \ll 26036001 | Obstruction, 587 }
\]

Using the long syntax, this expression constraint can be represented as:

\[
\text{descendantOf 404684003 | Clinical finding, 588 : 116676008 | Associated morphology, 589 NOT = descendantOrSelfOf 26036001 | Obstruction, 590}
\]

To prohibit an attribute from having a value in a particular range, a cardinality of \([0..0]\) must be used. For example, the following expression constraint represents the set of clinical findings which have exactly zero (i.e. they do not have any) associated morphologies that are a descendant or self of obstruction.

\[
\text{< 404684003 | Clinical finding, 591 : [0..0] 116676008 | Associated morphology, 592 = \ll 26036001 | Obstruction, 593}
\]

To prohibit an attribute from having a value outside a particular range, a cardinality of \([0..0]\) is used in conjunction with the 'not equal to' comparison operator. For example, the following expression constraint represents the set of clinical findings which have exactly zero associated morphologies that are not a descendant or self of obstruction. In other words, clinical findings for which all associated morphologies (if any exist) are descendants (or self) of obstruction.

\[
\text{< 404684003 | Clinical finding, 594 : [0..0] 116676008 | Associated morphology, 595 \neq \ll 26036001 | Obstruction, 596}
\]
If we also want to ensure that at least one associated morphology does exist (and all of these have a value which is a descendant or self of obstruction), then the following expression constraint can be used:

\[
< 404684003 | \text{Clinical finding} | [0..0] 116676008 | \text{Associated morphology} \neg= << 26036001 | \text{Obstruction} \land [1..*] 116676008 | \text{Associated morphology} = << 26036001 | \text{Obstruction}
\]

Note that the cardinality on the second attribute may be omitted, as \([1..*]\) is assumed by default.

### 6.6 Constraint Comments

#### 6.6.1 Comments

SNOMED CT Expression Constraints may also include comments inline within the constraint string to explain, describe or document different aspects of the expression constraints. Each comment begins with a forward slash directly followed by a star (i.e. "/*") and ends with a star directly followed by a forward slash (i.e. "/*/"). Comments may be placed anywhere in an expression constraint where whitespace (i.e. "ws") or mandatory whitespace (i.e. "mws") is allowed.

Comments have no effect on the machine processable interpretation of an expression constraint, as they should be ignored during evaluation. For example, the following two expression constraints (the first with comments, and the second without), will evaluate to exactly the same set of concepts:

```plaintext
/* Disorders of lung with edema */
< 19829001 | \text{Disorder of lung} : /* Descendants of disorder of lung */
116676008 | \text{Associated morphology} = << 79654002 | \text{Edema}
/* Where the associated morphology is edema or a subtype */
```

```plaintext
< 19829001 | \text{Disorder of lung} :
116676008 | \text{Associated morphology} = << 79654002 | \text{Edema}
```

A comment may include both stars and forward slashes. However a star may never be directly followed by a forward slash within the middle of a comment, as this combination denotes the end of the comment.

---

597 http://snomed.info/id/404684003
598 http://snomed.info/id/116676008
599 http://snomed.info/id/26036001
600 http://snomed.info/id/116676008
601 http://snomed.info/id/26036001
602 http://snomed.info/id/19829001
603 http://snomed.info/id/116676008
604 http://snomed.info/id/79654002
605 http://snomed.info/id/19829001
606 http://snomed.info/id/116676008
607 http://snomed.info/id/79654002

---

6. Examples – 99
6.7 Nested Expression Constraints

Expression constraints can be nested in a variety of ways to form nested expression constraints. These nested expression constraints use subexpressions, enclosed in round brackets, in the place of a simple concept reference.

Nested expression constraints can be created by:

- Applying constraint operators to an expression constraint
- Applying the memberOf function to an expression constraint
- Combining expression constraints using binary operators
- Adding dotted attributes to expression constraints
- Adding refinements to expression constraints
- Using expression constraints to represent valid attribute names
- Using expression constraints to represent valid attribute values

In this section, we describe each of these approaches to creating nested expression constraints.

6.7.1 Constraint Operators

When a constraint operator is applied to an expression constraint, the resulting set of matching expressions is the union of applying the constraint operator to each of its members.

For example, the following expression constraint represents all the members of the Example problem list concepts reference set plus the union of the descendants of each of these members.

\[
\text{<< (<} ^{700043003} \text{Example problem list concepts reference set})\]

Please note that the brackets in the above expression constraint are optional. In this particular case, removing the brackets does not change the meaning of the constraint.

As another example, the following expression constraint represents the set of all descendants of the Finding site of Fracture of bone.

\[
\text{< (} ^{125605004} \text{Fracture of bone} ^{363698007} \text{Finding site})\]

Because the Finding site of Fracture of bone is 272673000 Bone structure, the above expression constraint is equivalent to:

---

608 http://snomed.info/id/700043003
609 http://snomed.info/id/700043003
610 http://snomed.info/id/363698007
611 http://snomed.info/id/125605004
612 http://snomed.info/id/125605004
613 http://snomed.info/id/363698007
614 http://snomed.info/id/363698007
615 http://snomed.info/id/125605004
616 http://snomed.info/id/272673000
6. Examples

–

Please note that this is not the same as the expression constraint:

\[
< 272673000 | \text{Bone structure} >
\]

which refers to the set of values for any descendant of \( \text{Fracture of bone} \), and is instead equivalent to:

\[
( < 125605004 | \text{Fracture of bone} > \cdot 363698007 | \text{Finding site} >)
\]

See the subsection below on Dotted Attributes (see page 103) for more information about expression constraints of this form.

6.7.2 MemberOf Function

The memberOf function may also be applied to an expression constraint that returns a set of concept-based reference set concepts. When this is done, the nested expression constraint (to which the memberOf function is applied) must always be enclosed in round brackets.

For example, the expression constraint below is satisfied by the set of concepts which are members of any subtype of \( \text{GP/FP health issue reference set} \). In other words, it represents the union of applying the memberOf function to each of the descendants of \( \text{GP/FP health issue reference set} \).

\[
\wedge (< 450973005 | \text{GP/FP health issue reference set} >)
\]

The expression constraint above evaluates to the same set of concepts as applying the memberOf function to each individual subtype of \( \text{GP/FP health issue reference set} \) and then taking the union of these sets. Therefore, when applied to the 20170131 international edition of SNOMED CT, the above expression constraint evaluates to the same set of concepts as the following expression constraint.

---

617 http://snomed.info/id/272673000
618 http://snomed.info/id/125605004
619 http://snomed.info/id/363698007
620 http://snomed.info/id/125605004
621 http://snomed.info/id/363698007
622 http://snomed.info/id/125605004
623 http://snomed.info/id/363698007
624 http://snomed.info/id/450973005
625 http://snomed.info/id/450973005
626 http://snomed.info/id/450973005
627 http://snomed.info/id/450973005
6. Examples – 101
6.7.3 Compound Expression Constraints

When conjunction (i.e. AND), disjunction (i.e. OR) or exclusion (i.e. MINUS) are applied to one or more complex subexpression constraints, brackets are usually required to nest the subexpression constraints.

For example, the following expression constraint uses brackets around the first complex operand (\(< 404684003 \ | \ Clinical\ finding\ 363698007 \ | \ Finding\ site\ = \ <\ 39057004 \ | \ Pulmonary\ valve\ structure\)) to apply the 'AND' operator to two expression constraints.

\[
(\(< 404684003 \ | \ Clinical\ finding\ 363698007 \ | \ Finding\ site\ = \ <\ 39057004 \ | \ Pulmonary\ valve\ structure\)) \ AND \ ^700043003 \ | \ Example\ problem\ list\ concepts\ reference\ set
\]

An equivalent expression constraint can be achieved by swapping the order of the operands, as shown below.

\[
^700043003 \ | \ Example\ problem\ list\ concepts\ reference\ set \ AND \ (< 404684003 \ | \ Clinical\ finding\ 363698007 \ | \ Finding\ site\ = \ <\ 39057004 \ | \ Pulmonary\ valve\ structure\)
\]
Similarly, if both sides of the compound expression are complex expression constraints, then brackets may be required on both sides. For example:

\[
(\lt 404684003 \text{Clinical finding} \leq 363698007 \text{Finding site} \lt 39057004 \text{Pulmonary valve structure})
\]

\[
\text{AND } (\lt 64572001 \text{Disease} \leq 116676008 \text{Associated morphology} \lt 415582006 \text{Stenosis})
\]

6.7.4 Dotted Attributes

Dotted attributes can also be applied to a nested subexpression constraint. When this is done, the resulting subexpression represents the union of the values of the given dotted attribute for any expression that matches the given nested subexpression constraint.

For example, the following expression constraint represents the set of all substances that are the \text{Direct substance} of a \text{Specimen collection} procedure that is \text{Using device} equal to a subtype (or self) of \text{Catheter}.

\[
(\lt 17636008 \text{Specimen collection} \leq 424226004 \text{Using device} = \lt 19923001 \text{Catheter})
\]

When executed against the 20170131 international edition of SNOMED CT, the above expression constraint matches the following three concepts:

<table>
<thead>
<tr>
<th>Concept</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Urine</td>
<td>78014005</td>
</tr>
<tr>
<td>Blood</td>
<td>87612001</td>
</tr>
<tr>
<td>Arterial blood</td>
<td>4635002</td>
</tr>
</tbody>
</table>

647 http://snomed.info/id/404684003
648 http://snomed.info/id/363698007
649 http://snomed.info/id/39057004
650 http://snomed.info/id/64572001
651 http://snomed.info/id/116676008
652 http://snomed.info/id/415582006
653 http://snomed.info/id/363701004
654 http://snomed.info/id/17636008
655 http://snomed.info/id/424226004
656 http://snomed.info/id/19923001
657 http://snomed.info/id/17636008
658 http://snomed.info/id/424226004
659 http://snomed.info/id/19923001
660 http://snomed.info/id/363701004
661 http://snomed.info/id/78014005
662 http://snomed.info/id/87612001
663 http://snomed.info/id/4635002
6.7.5 Refinement

As mentioned in 6.2 Refinements (see page 72), it is possible to apply refinements to nested expression constraints. When a refinement is applied to a complex subexpression constraint, the subexpression constraint must be enclosed in brackets.

For example, the expression constraint below represents the set of all clinical findings and events which occur after some procedure.

\[
\langle< \text{Clinical finding} \rangle \text{ OR } \langle< \text{Event} \rangle \rangle
\]

\[
\langle< \langle< \text{Procedure} \rangle \rangle : \langle< \langle< \text{After} \rangle \rangle = \langle< \text{Procedure} \rangle \rangle \rangle
\]

**Attribute Names**

In some cases, the valid set of attribute names can be represented using an expression constraint. For example, the expression constraint below represents the set of bone fractures that have no additional defining attributes (besides | Finding site | and | Associated morphology |).

\[
\langle< \text{Fracture of bone} \rangle : \langle< \langle< \text{Concept model attribute} \rangle \langle< \langle< \text{Finding site} \rangle \rangle \rangle \rangle \rangle
\]

\[
\langle< \langle< \text{Associated morphology} \rangle \rangle = \ast
\]

Within this expression constraint, the subexpression:

\[
\langle< \langle< \text{Concept model attribute} \rangle \rangle \langle< \langle< \text{Finding site} \rangle \rangle \rangle \langle< \text{Associated morphology} \rangle \rangle
\]

represents the set of attributes that must match the given refinement condition (in this case, these attributes must not appear in the concept definition of matching concepts due to the cardinality of [0..0]).

---

664 http://snomed.info/id/404684003
665 http://snomed.info/id/272379006
666 http://snomed.info/id/255234002
667 http://snomed.info/id/71388002
668 http://snomed.info/id/363698007
669 http://snomed.info/id/116676008
670 http://snomed.info/id/125605004
671 http://snomed.info/id/410662002
672 http://snomed.info/id/363698007
673 http://snomed.info/id/116676008
674 http://snomed.info/id/410662002
675 http://snomed.info/id/363698007
676 http://snomed.info/id/116676008
677 http://snomed.info/id/363698007
678 http://snomed.info/id/116676008
679 http://snomed.info/id/410662002
680 http://snomed.info/id/363698007
681 http://snomed.info/id/116676008
682 http://snomed.info/id/410662002
683 http://snomed.info/id/363698007
684 http://snomed.info/id/116676008
685 http://snomed.info/id/272379006
686 http://snomed.info/id/255234002
687 http://snomed.info/id/71388002
688 http://snomed.info/id/363698007
689 http://snomed.info/id/116676008
690 http://snomed.info/id/125605004
691 http://snomed.info/id/410662002
692 http://snomed.info/id/363698007
693 http://snomed.info/id/116676008
694 http://snomed.info/id/410662002
695 http://snomed.info/id/363698007
696 http://snomed.info/id/116676008
6.7.6 Attribute Values

Similarly to the SNOMED CT Compositional Grammar, it is also possible to nest expression constraints within an attribute value. Please note that when the attribute value is a simple expression constraint (as per the above examples), brackets are not required around the value. However, when the attribute value is either an expression constraint with a refinement, or a compound expression constraint with a binary operator, then brackets must be placed around the attribute value. For example, the following expression constraint represents the set of clinical findings which are associated with another clinical finding that has an associated morphology of 'infarct' (or subtype).

\[
< \text{404684003 | Clinical finding}^{677} ; \\
\text{47429007 | Associated with}^{678} = (\text{< 404684003 | Clinical finding}^{679} ; \\
\text{116676008 | Associated morphology}^{680} = \ll < 55641003 | \text{Infarct}^{681})
\]

In this example, brackets are required around the nested attribute value " < 404684003 | Clinical finding".

6.8 Description Filters

In this section, we illustrate how description filters can be applied to expression constraints to further restrict the matching concepts.

6.8.1 Overview

Description filter constraints provide the ability to limit the set of concepts, that satisfy a given expression constraint, based on the descriptions associated with each concept. Only concepts that have at least one matching description for each filter criteria will be included in the set of matching concepts. Descriptions can be filtered based on their term, type, language, dialect, acceptability in a given dialect, module, effectiveTime, active status and description identifier. Description filters are specified inside double curly braces, and optionally being with the letter "D". Any filter that does not specify its type is, by default, assumed to be a description filter.

In the following sections, we explain each type of description filter criteria.

6.8.2 Term Filter

Term filters enable an expression constraint to match on only those concepts with an associated description whose term matches the given search term. For example, the following expression constraint is satisfied by SNOMED CT concepts with a description matching the search terms "heart" and "att". This expression constraint works like a term search performed in a SNOMED CT browser. Please note that the "D" (either upper or lower case) at the start of the filter indicates that this is a description filter constraint, rather than a concept filter constraint (see 6.9 Concept Filters).

677 http://snomed.info/id/404684003
678 http://snomed.info/id/47429007
679 http://snomed.info/id/404684003
680 http://snomed.info/id/116676008
681 http://snomed.info/id/55641003
682 http://snomed.info/id/404684003
683 http://snomed.info/id/116676008
684 http://snomed.info/id/55641003
Filters (see page 116). If the type of a filter constraint is not specified (as in most of the examples below), then it is assumed that the constraint is a description constraint.

By default, term filters match using a word-prefix-any-order match technique. This means that each string value in the search term must match the start of a word in the concept's description term, but that these words may appear in any order. This word-prefix-any-order match technique can be explicitly specified in the term filter, using the keyword "match:" before the search term. For example, the following four expression constraints are equivalent, and are each satisfied only by diseases with a description term that includes both a word starting with "heart" and a word starting with "att" (in any order).

```
< 64572001 | Disease | {{ term = "heart att" }}
```

```
< 64572001 | Disease | {{ term = "heart", term = "att" }}
```

```
< 64572001 | Disease | {{ term = match:"heart att" }}
```

```
< 64572001 | Disease | {{ term = "att heart" }}
```

To indicate that a matching description may match either one search term or another, a search term set may be used.

The example below matches only those diseases with a description term containing either a word starting with "heart" or a word starting with "card" (or both).

```
< 64572001 | Disease | {{ term = ("heart" "card") }}
```

The other technique that may be used is a wildcard search. This technique is specified using the keyword "wild:" and matches the search term in the expression constraint against the entire candidate description term (rather than just individual words). An asterisk ("*") is used as a wildcard to indicate that any (zero to many) characters may appear in the given position.

For example, the expression constraint below will match only diseases with a description term starting with "cardi" and ending with "opathy" with any number of characters between. This term filter would therefore match on terms such as "cardiopathy", "cardiomyopathy" and "cardiac channelopathy", but would not match on terms like "atrial cardiopathy" or "Cardiomyopathy (disorder)".

---

Please note that to perform an exact string match on an entire term, a wildcard search without a wildcard can be used (e.g. term=wild:"cardiopathy"). For example, the following expression constraint will match only diseases with a description term that exactly matches the full string "cardiopathy". This expression constraint will therefore match the concept 56265001 | Heart disease (disorder) (with synonym "Cardiopathy"), but will not match the concept 870575001 | Disorder of cardiac atrium (disorder) (with synonym "Atrial cardiopathy")

It is also possible to mix the match techniques in a search term set. For example, the expression constraint below will match those diseases with a description term that either contains a word starting with "gas", or ending with "itis" - e.g. "gastric flu", "gastritis", or "tonsillitis".

If more than one filter is applied, then all filters (surrounded in double braces) must match at least one description of a concept, for that concept to satisfy the constraint. The descriptions that match each of the filters can either be the same description, or different descriptions on the same concept.

The expression constraint below matches those diseases which have both a description that contains a word starting "eye" and a description that ends with "itis". For example, this constraint would match the concept 9826008 | Conjunctivitis (disorder) (with synonyms "Pink eye disease" and " Conjunctivitis") and the concept 15680481000119104 | Viral conjunctivitis of bilateral eyes (disorder) (with synonyms "Bilateral viral conjunctivitis" and "Viral conjunctivitis of both eyes"), but would not match the concept 45261009 | Viral conjunctivitis (disorder) (which does not have a synonym matching the word prefix "eye").

6.8.3 Language Filter
Language filters enable an expression constraint to match on only those concepts with a matching description in a specified language. Language filters use the keyword "language", followed by a comparison operator (e.g. "=" or "!
="), and the ISO 639-1 two-character language code (in upper or lowercase).

The expression constraint below matches only those diseases with a Swedish description containing the word prefix "hjärt" - e.g. 41884003 | hjärtpolyp from the Swedish Edition (20200531)
The expression constraint below matches only those diseases with a Swedish description containing the word prefix "hjärt" and an English description containing the word prefix "heart" - e.g. 84114007 | hjärtsvikt (with English synonym "Heart failure") from the Swedish Edition (20200531).

< 64572001 | Disease {{ term = "hjärt", language = sv }}

The following two expression constraints are equivalent, and both match only the subtypes of Heart disease, which have a Swedish synonym containing the word prefix "hjärt".

< 64572001 | Disease {{ term = "hjärt", language = sv }} {{ term = "heart", language = en }}

6.8.4 Description Type Filter

Type filters enable an expression constraint to match on only those concepts with a matching description of a specified type. Type filters may either use the keyword "type" with the values "fsn", "syn" or "def", or may use the keyword "typeld" with a concept value that is 900000000000446008 | Description type.

The following table lists the valid description type keywords in both the brief and full syntax, and their equivalent concept reference alternatives. Please note that the full syntax accepts both the brief and full syntax keywords. If additional description types are required, these must be specified in a filter using the 'typeld' keyword with the corresponding concept reference.

<table>
<thead>
<tr>
<th>Type Keyword</th>
<th>Typeld</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brief Syntax</td>
<td>Full Syntax</td>
</tr>
<tr>
<td>fsn</td>
<td>fullySpecifiedName</td>
</tr>
<tr>
<td>syn</td>
<td>synonym</td>
</tr>
<tr>
<td>def</td>
<td>definition</td>
</tr>
</tbody>
</table>

For example, the expression constraint below matches all the subtypes of Heart disease, that have a fully specified name containing the word prefix "heart".

< 56265001 | Heart disease {{ term = "heart", type = fsn }}

The following two expression constraints are equivalent, and both match only the subtypes of Heart disease, which have a Swedish synonym containing the word prefix "hjärt".

http://snomed.info/id/64572001
http://snomed.info/id/84114007
http://snomed.info/id/56265001
http://snomed.info/id/900000000000446008
http://snomed.info/id/64572001
http://snomed.info/id/64572001
http://snomed.info/id/64572001
http://snomed.info/id/64572001

The two equivalent expression constraints below match the subtypes of \textit{Heart disease}, which either have a synonym containing the word prefix "heart", or a fully specified name containing the word prefix "heart".

\begin{verbatim}
< 56265001 | Heart disease | { term = "hjärt", language = SV, type = syn }
< 56265001 | Heart disease | { term = "hjärta", language = sv, typeld = 90000000000013009 | synonym }
\end{verbatim}

\subsection*{6.8.5 Dialect Filter}

Dialect filters enable an expression constraint to match on only those concepts with a matching description in a specified language reference set. Dialect filters may either use the keyword "dialect" with a value that represents a valid alias for a specific language reference set, or may use the keyword "dialectId" with a concept value that is a reference set.

For example, the two equivalent expression constraints below will match all subtypes of \textit{Disease} that have a description in the Australian English language reference set.

\begin{verbatim}
< 64572001 | Disease | { dialect = en-au }
< 64572001 | Disease | { dialectId = 32570271000036106 | Australian English language reference set }
\end{verbatim}

The expression constraint below matches all diseases with a description in the New Zealand English language reference set that has a word starting with "cardio".

\begin{verbatim}
< 64572001 | Disease | { dialectId = 32570271000036106 | Australian English language reference set }
\end{verbatim}
6. Examples

In some situations, multiple language reference sets need to be used together to identify an appropriate set of concepts. A filter constraint may include a list of dialects to specify that a matching description may belong to any of the given language reference sets.

For example, the following expression constraint matches all diseases that have a description in either the en-nhs-clinical or en-nhs-pharmacy language reference sets, where that description contains a word starting with the prefix "card".

```
< 64572001 |Disease| { term = "card", dialect = en-nz }
```

6.8.6 Acceptability Filter

Acceptability filters enable an expression constraint to match on only those concepts with a matching description that has the specified acceptability in the specified language reference set. Acceptability filters must always be applied to a specified dialect. As such, they are represented by placing the required acceptability in brackets after the value of the dialect filter. Acceptabilities can be indicated using either one of the keywords below, or using a concept value that is a valid acceptability in both the brief and full syntax, and their equivalent concept reference alternatives. Please note that the full syntax accepts both the brief and full syntax keywords.

<table>
<thead>
<tr>
<th>Acceptability Keyword</th>
<th>Acceptability Id</th>
<th>Concept Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>prefer</code></td>
<td><code>preferred</code></td>
<td>90000000000548007</td>
</tr>
<tr>
<td><code>accept</code></td>
<td><code>acceptable</code></td>
<td>90000000000549004</td>
</tr>
</tbody>
</table>

For example, the following two expression constraints both match all descendants of disease with a description that matches the word prefix 'box', has the type 'synonym', and has an acceptability of 'preferred' in the en-us language reference set. In other words, this expression constraint matches diseases with a US English preferred term that uses the word prefix 'box'.

```
< 64572001 |Disease| { term = "box", type = syn, dialect = en-us (prefer) }
```
Multiple dialect filters may be used with different acceptabilities applied to each. For example, the expression constraint below matches on diseases, which have a synonym with word prefix "box" that is preferred in the en-nhs-clinical language reference set and is acceptable in the en-gb language reference set.

```
< 64572001 |Disease|727 {{ term = "box", type = syn, dialect = en-nhs-clinical (prefer), dialect = en-gb (accept) }}
```

To support alternative acceptabilities in more than one language reference set, a dialect set can be used. For example, the following two equivalent expression constraints match on diseases, which have a synonym with word prefix "box" that is either preferred in the en-gb language reference set or preferred in the en-nhs-clinical language reference set.

```
< 64572001 |Disease|728 {{ term = "box", type = syn, dialect = ( en-gb (prefer) en-nhs-clinical (prefer) ) }}
```

```
< 64572001 |Disease|729 {{ term = "box", type = syn, dialect = ( en-gb en-nhs-clinical ) (prefer) }}
```

### 6.8.7 Filters with Negation

Filters can use negation in a number of ways. The simplest approach is to use the 'not equal to' comparison operator (e.g. "!=") before the value.

For example, the following expression constraint matches on subtypes of |Fracture of bone|730 that do not use the word prefix "fracture" in their US English preferred term.

```
< 125605004 |Fracture of bone|731 {{ term != "fracture", type = syn, dialect = en-us (prefer) }}
```

If we remove the type and acceptability filters, as shown below, the remaining expression constraint matches on those subtypes of |Fracture of bone|732 which have any US English description that does not contain the word prefix "fracture". Concepts including |Fractured nasal bones|,733 (with synonym "Broken nose") will match the constraint below.

```
< 125605004 |Fracture of bone|732 {{ term != "fracture", type = syn, dialect = en-us (prefer) }}
```
To find the set of concepts, for which all descriptions match some specified criteria, the expression constraint must use the MINUS operation to exclude concepts that have a non-matching description. For example, the expression constraint below matches all subtypes of Fracture of bone, for which every description contains the word prefix “fracture”. Please note that the filter only applies to the descendants of Fracture of bone (i.e. the subexpression directly proceeding the filter).

This expression constraint can be simplified to the equivalent one below, using the wildcard character ‘*’ (which represents any concept in the substrate).

Using a similar principle, the expression constraint below matches all concepts that do not have a preferred term specified in the en-nz language reference set.

### 6.8.8 Module Filter

Description module filters enable an expression constraint to match on only those concepts with a matching description that belongs to a specified module. Module filters use the keyword “moduleId” with a concept reference that is Module.

For example, the expression constraint below matches all subtypes of Asthma with a description that belongs to the US National Library of Medicine maintained module.

And the expression constraint below matches all subtypes of Clinical finding with a definition that belongs to the international core module.
6.8.9 Effective Time Filter

Description effective time filters enable an expression constraint to match on only those concepts with a description that has an effectiveTime matching the specified criteria. Effective time filters may use any of the date comparison operators shown below:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>!=</td>
<td>Not equals</td>
</tr>
<tr>
<td>&lt;</td>
<td>Before the given date</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Before or on the given date</td>
</tr>
<tr>
<td>&gt;</td>
<td>After the given date</td>
</tr>
<tr>
<td>&gt;=</td>
<td>After or on the given date</td>
</tr>
</tbody>
</table>

Please note that the value of an effective time filter (if present) must be a 8 digit date, formatted according to ISO 8601’s basic calendar date format (i.e. YYYYMMDD). If the effectiveTime of the description in the substrate includes a time and/or time zone designator, these should be ignored when performing the comparison.

For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description that has an effective time of 31st January 2021.

< 125605004 | Fracture of bone {{ D effectiveTime = "20210131" }}

And the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description that has any effective time that is not 31st January 2021.

< 125605004 | Fracture of bone {{ D effectiveTime != "20210131" }}
Similarly, greater than, less than, greater than or equals and less than or equals operators may be used in an effectiveTime filter. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description that has an effectiveTime of 31st July 2019 or later (i.e. more recent).

\[
< 125605004 | \text{Fracture of bone} \text{ } \{ \text{D effectiveTime } \geq \text{"20190731"} \} \]
\]

And the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description that has an effective time of 31st July 2019 or earlier.

\[
< 125605004 | \text{Fracture of bone} \text{ } \{ \text{D effectiveTime } \leq \text{"20190731"} \} \]
\]

The effectiveTime filter can also use sets of effective times. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description that has an effectiveTime of either 31st January 2019, 31st July 2019, 31st January 2020, or 31st July 2020.

\[
< 125605004 | \text{Fracture of bone} \text{ } \{ \text{D effectiveTime } = (\text{"20190131" } \text{"20190731" } \text{"20200131" } \text{"20200731"}) \} \]
\]

And the expression constraint below matches all subtypes of 125605004 | Fracture of bone with a description, which does not have any of the following effective times: 31st January 2019, 31st July 2019, 31st January 2020 or 31st July 2020.

\[
< 125605004 | \text{Fracture of bone} \text{ } \{ \text{D effectiveTime } \neq (\text{"20190131" } \text{"20190731" } \text{"20200131" } \text{"20200731"}) \} \]
\]

To match concepts with unpublished descriptions, to which an effectiveTime has not been assigned, an effectiveTime value of "" can be used. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with a description to which an effectiveTime has not yet been assigned.

\[
< 125605004 | \text{Fracture of bone} \text{ } \{ \text{D effectiveTime } = \text{""} \} \]
\]

Please note that description effectiveTime filters, which use the comparison operators "<" and ">", will not match any descriptions with an effectiveTime = "".

\[751\text{ http://snomed.info/id/125605004} \]
\[752\text{ http://snomed.info/id/125605004} \]
\[753\text{ http://snomed.info/id/125605004} \]
\[754\text{ http://snomed.info/id/125605004} \]
\[755\text{ http://snomed.info/id/125605004} \]
\[756\text{ http://snomed.info/id/125605004} \]
\[757\text{ http://snomed.info/id/125605004} \]
\[758\text{ http://snomed.info/id/125605004} \]
\[759\text{ http://snomed.info/id/125605004} \]
\[760\text{ http://snomed.info/id/125605004} \]
6.8.10 Active Filter

Description active filters enable an expression constraint to match on only those concepts with a description that has a matching active status. Descriptions are either active (i.e. active = 1 or active = "true") or inactive (i.e. active = 0 or active = "false"). By default, only active descriptions are included in the substrate.

For example, the following expression constraints return all concepts in the International Patient Summary reference set, which have an active description.

\[
\begin{align*}
\wedge 816080008 & \mid \text{International Patient Summary} \mid \{ \{ \text{D active} = 1 \} \} \\
\wedge 816080008 & \mid \text{International Patient Summary} \mid \{ \{ \text{D active} = \text{true} \} \}
\end{align*}
\]

And the following expression constraints return all concepts in the International Patient Summary reference set, which have an inactive description.

\[
\begin{align*}
\wedge 816080008 & \mid \text{International Patient Summary} \mid \{ \{ \text{D active} = 0 \} \} \\
\wedge 816080008 & \mid \text{International Patient Summary} \mid \{ \{ \text{D active} = \text{false} \} \}
\end{align*}
\]

6.8.11 Description Id Filter

Description id filters enable an expression constraint to match on only those concepts with a description that has a matching description identifier. For example, the following expression constraint matches any concept, which has an associated description with the identifier "3032638017". The only concept that matches this expression constraint is 707444001 | Uncomplicated asthma (disorder).

\[
* \begin{align*}
\{ \{ \text{D id} = 3032638017 \} \}
\end{align*}
\]

Description id filters can also be applied to other expression constraints, to check whether the concept with the matching description id is in a given set of concepts. For example, the following expression constraint will match any descendant of 195967001 | Asthma (disorder), which has a description with identifier "3032638017". This can be used to check if the concept with the given description id is a descendant of 195967001 | Asthma (disorder).

---

761 http://snomed.info/id/816080008
762 http://snomed.info/id/816080008
763 http://snomed.info/id/816080008
764 http://snomed.info/id/816080008
765 http://snomed.info/id/707444001
766 http://snomed.info/id/195967001
767 http://snomed.info/id/195967001
6.9 6.9 Concept Filters

In this section, we illustrate how concept filters can be applied to expression constraints to further restrict the matching concepts.

6.9.1 Overview

Concept filter constraints provide the ability to limit the set of concepts that satisfy a given expression constraint, based on the properties of each concept. Only concepts with properties that match the criteria specified in the concept filter constraint will be included in the set of matching concepts. Concepts can be filtered based on their definition status, module, effectiveTime, and active status. In the following sections we explain each of these concept filter criteria.

Definition Status Filter

Definition status filters enable an expression constraint to match on only those concepts with a matching definition status. Definition status filters may either use the keyword 'definitionStatus' with the values "defined" or "primitive", or may use the keyword "definitionStatusId" with a concept value that is < 900000000000444006 | Definition status".  

---

768 http://snomed.info/id/195967001
769 http://snomed.info/id/195967001
770 http://snomed.info/id/195967001
771 https://confluence.ihtsdotools.org/display/ECL/6.11+History+Supplements
772 http://snomed.info/id/195967001
773 http://snomed.info/id/195967001
774 http://snomed.info/id/170644007
775 http://snomed.info/id/195967001
776 http://snomed.info/id/900000000000444006
The following table lists the valid definitionStatus tokens and their equivalent definitionStatusId concept reference alternatives. If additional definition statuses are required, these must be specified in a filter using the 'definitionStatusId' keyword with the corresponding concept reference.

<table>
<thead>
<tr>
<th>definitionStatus (token)</th>
<th>definitionStatusId (concept reference)</th>
</tr>
</thead>
<tbody>
<tr>
<td>primitive</td>
<td>900000000000074008</td>
</tr>
<tr>
<td>defined</td>
<td>900000000000073002</td>
</tr>
</tbody>
</table>

For example, the expression constraints below match all the primitive subtypes of |Heart disease|.

< 56265001 |Heart disease| {{ C definitionStatus = primitive }} |

< 56265001 |Heart disease| {{ C definitionStatusId = 900000000000074008 |Primitive | }} |

Similarly, the two expression constraints below match all the fully defined subtypes of |Heart disease|.

< 56265001 |Heart disease| {{ C definitionStatus = defined }} |

< 56265001 |Heart disease| {{ C definitionStatusId = 900000000000073002 |Defined | }} |

Please note that Concept filters and Description Filters(see page 105) can be used together to filter the results of an expression constraint based on both the properties of each concept and the properties of their descriptions. For example the following expression constraint matches all primitive subtypes of |Disease|, which have at least one description term that includes a word starting with "heart".

< 64572001 |Disease| {{ C definitionStatus = primitive }} {{ D term = "heart" }} |

777 http://snomed.info/id/56265001
778 http://snomed.info/id/56265001
779 http://snomed.info/id/56265001
780 http://snomed.info/id/900000000000074008
781 http://snomed.info/id/56265001
782 http://snomed.info/id/56265001
783 http://snomed.info/id/56265001
784 http://snomed.info/id/900000000000073002
785 http://snomed.info/id/64572001
786 http://snomed.info/id/64572001
6.9.2 Module Filter

Module filters enable an expression constraint to match on only those concepts that belong to a specified module. Module filters use the keyword "moduleId" with a concept reference that is `< 900000000000443000 | Module >`. For example, the expression constraint below matches all subtypes of `195967001 | Asthma` that belong to the US National Library of Medicine maintained module.

\[
< 195967001 | Asthma \quad \{(C \text{moduleId} = 731000124108 | US \text{National Library of Medicine maintained module})\}
\]

And the expression constraint below matches all primitive subtypes of `195967001 | Asthma` that belong to the international core module.

\[
< 195967001 | Asthma \quad \{(C \text{definitionStatus} = \text{primitive}, \text{moduleId} = 900000000000207008 | \text{SNOMED CT core module})\}
\]

6.9.3 Effective Time Filter

Effective time filters enable an expression constraint to match on only those concepts with an effectiveTime that matches the specified criteria. Effective time filters may use any of the date comparison operators shown below:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>!=</td>
<td>Not equals</td>
</tr>
<tr>
<td>&lt;</td>
<td>Before the given date</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Before or on the given date</td>
</tr>
<tr>
<td>&gt;</td>
<td>After the given date</td>
</tr>
<tr>
<td>&gt;=</td>
<td>After or on the given date</td>
</tr>
</tbody>
</table>

787 http://snomed.info/id/900000000000443000
788 http://snomed.info/id/195967001
789 http://snomed.info/id/195967001
790 http://snomed.info/id/731000124108
791 http://snomed.info/id/195967001
792 http://snomed.info/id/195967001
793 http://snomed.info/id/900000000000207008
Please note that the value of an effective time filter (if present) must be a 8 digit date, formatted according to ISO 8601’s basic calendar date format (i.e. YYYYMMDD). If the effectiveTime of the concept in the substrate includes a time and/or time zone designator, these should be ignored when performing the comparison.

For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with an effective time of 31st January 2021.

< 125605004 | Fracture of bone

And the following expression constraint matches all subtypes of 125605004 | Fracture of bone with any effective time that is not 31st January 2021.

< 125605004 | Fracture of bone

Similarly, greater than, less than, greater than or equals and less than or equals operators may be used in an effectiveTime filter. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with an effectiveTime of 31st July 2019 or later (i.e. more recent).

< 125605004 | Fracture of bone

And the following expression constraint matches all subtypes of 125605004 | Fracture of bone with an effective time of 31st July 2019 or earlier.

< 125605004 | Fracture of bone

The effectiveTime filter can also use sets of effective times. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone with an effectiveTime of either 31st January 2019, 31st July 2019, 31st January 2020, or 31st July 2020.

< 125605004 | Fracture of bone

And the expression constraint below matches all subtypes of 125605004 | Fracture of bone which does not have any of the following effective times: 31st January 2019, 31st July 2019, 31st January 2020 or 31st July 2020.

< 125605004 | Fracture of bone

---

794 http://snomed.info/id/125605004
795 http://snomed.info/id/125605004
796 http://snomed.info/id/125605004
797 http://snomed.info/id/125605004
798 http://snomed.info/id/125605004
799 http://snomed.info/id/125605004
800 http://snomed.info/id/125605004
801 http://snomed.info/id/125605004
802 http://snomed.info/id/125605004
803 http://snomed.info/id/125605004
804 http://snomed.info/id/125605004
To match unpublished concepts to which an effectiveTime has not been assigned, an effectiveTime value of "" can be used. For example, the following expression constraint matches all subtypes of 125605004 | Fracture of bone to which an effectiveTime has not yet been assigned.

< 125605004 | Fracture of bone | {{ C effectiveTime = "" }}

Please note that effectiveTime filters, which use the comparison operators "<" and ">"", will **not** match any concepts with an effectiveTime = "".

### 6.9.4 Active Filter

Active filters enable an expression constraint to match on only those concepts with a matching active status. Concepts are either active (i.e. active = 1 or active = "true") or inactive (i.e. active = 0 or active = "false"). By default, both active and inactive concepts are included in the substrate. This allows inactive members of a reference set to be retrieved (e.g. for historical reference sets, in which the referenced component is intended to be inactive). However, because only active relationships are included in the default substrate, as soon as a refinement or hierarchical operator is used, only active concepts are matched.

For example, the following expression constraints returns only active concepts in the International Patient Summary reference set.

^ 816080008 | International Patient Summary | {{ C active = 1 }}

And the following expression constraints return only inactive concepts in the International Patient Summary reference set.

^ 816080008 | International Patient Summary | {{ C active = 0 }}

^ 816080008 | International Patient Summary | {{ C active = false }}
Please note that module filters are not intended to replace the use of simple reference sets to organize content of a particular type. Module filters are instead intended to be used for purposes related to the management of extensions or editions.

6.10 6.10 Member Filters

In this section, we illustrate how filters can be applied to a set of reference set members to restrict the matching values.

6.10.1 Overview

Member filters provide the ability to filter the rows of a reference set, based on the value of specific fields in the reference set. These filters are specified inside double curly braces, and begin with the letter "M".

6.10.2 Member Field Filters

To apply a member filter to one or more reference sets, the fields of those reference sets are matched against specified criteria. Only reference set members whose field values match the given criteria will be included in the results.

For example, the following expression constraint will match all referencedComponentIds (i.e. SNOMED CT concept id) from the active ICD-10 complex map reference set rows, which map to the ICD-10 code "J45.9" (as a word prefix). When applied to the July 2021 international edition, this will match 59 concepts, including 195967001 | Asthma, 707447008 | Exacerbation of severe persistent asthma (disorder) and 401193004 | Asthma confirmed (situation).

```
^ 447562003 |ICD-10 complex map reference set| | {{ M mapTarget = "J45.9" }}
```

Please note that by default, a word-prefix-any-order match is performed. Therefore the following expression constraint will match on rows that have a mapTarget of "J45.0", "J45.1", ..., "J45.8", "J45.9" etc.

```
^ 447562003 |ICD-10 complex map reference set| | {{ M mapTarget = "J45" }}
```

Alternatively, a wildcard search can be performed, to achieve similar results. For example, the expression constraint below will match on rows that have a mapTarget starting with "J45" followed by zero or more other characters (e.g. "J45.0", "J45.1", ..., "J45.8", "J45.9")

```
^ 447562003 |ICD-10 complex map reference set| | {{ M mapTarget = wild:"J45*" }}
```

812 http://snomed.info/id/447562003
813 http://snomed.info/id/195967001
814 http://snomed.info/id/707447008
815 http://snomed.info/id/401193004
816 http://snomed.info/id/447562003
817 http://snomed.info/id/447562003
818 http://snomed.info/id/447562003
To achieve an exact string match, a wildcard search (without an ‘*’ can be used. For example, the expression constraint below will match only rows that have a mapTarget of "J45.9".

\[^\text{447562003}\mid\text{ICD-10 complex map reference set}\]{{ \text{M mapTarget = wild:"J45.9"} }}

For more information on wildcard and word-prefix-any-order searching, please refer to 6.8 Description Filters. Multiple field constraints can be applied within a reference set member filter. For example, the following expression constraint will return the referencedComponentId from the rows, which have a mapGroup of "2", a mapPriority of "1" and a mapTarget of "J45.9".

\[^\text{447562003}\mid\text{ICD-10 complex map reference set}\]{{ \text{M mapGroup = #2, mapPriority = #1, mapTarget = "J45.9"} }}

Other comparison operators may also be used, when defining field criteria. The available operators depend on the field’s datatype, as shown in the table below.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Comparison Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brief syntax</td>
</tr>
<tr>
<td>SCTID / Expression</td>
<td>= , !=</td>
</tr>
<tr>
<td>Integer / Decimal</td>
<td>= , != , &lt;= , &lt; , &gt;= , &gt;</td>
</tr>
<tr>
<td>String</td>
<td>= , !=</td>
</tr>
<tr>
<td>Boolean</td>
<td>= , !=</td>
</tr>
<tr>
<td>Time</td>
<td>= , != , &lt;= , &lt; , &gt;= , &gt;</td>
</tr>
</tbody>
</table>

In addition, reference set fields of type 'string' may be filtered using the same word-prefix-any-order and wildcard techniques used by the description term filters. For example, the following expression constraint will match all rows that have a mapGroup not equal to 2, a mapPriority less than 2, and a mapTarget that starts with the letter "J".

\[^\text{447562003}\mid\text{ICD-10 complex map reference set}\]{{ \text{M mapGroup ! = #2, mapPriority < #2, mapTarget = wild:"J*"} }}
Member filters can also be used in combination with the memberOf function to support the selection of other fields of a reference set (see 6.1 Simple Expression Constraints (see page 64)). For example, the following expression constraint returns the active SNOMED CT concept that is considered to be the same as the inactive concept 67415000 | Hay asthma.

```
^ [targetComponentId] 900000000000527005 | SAME AS association reference set
{{ M referencedComponentId = 67415000 | Hay asthma }}
```

For more information on the use of reference set field names in ECL, please refer to Appendix E - Reference Set Fields (see page 206).

For additional ways of specifying queries over the historical association reference sets, please refer to 6.11 History Supplements (see page 124).

### 6.10.3 Module Filter

Module filters enable an expression constraint to match on only those rows of a reference set that belong to a specified module. Module filters use the keyword "moduleId" with a concept reference that is < 900000000000443000 | Module.

For example, the expression constraint below matches all members of the 900000000000534007 | Module dependency reference set that belong to an Australian maintained module.

```
^ 900000000000534007 | Module dependency reference set
{{ M moduleId = << 32570231000036109 | Australian maintained module }}
```

### 6.10.4 Effective Time Filter

Effective time filters enable an expression constraint to match on only those rows of a reference set with an effectiveTime that matches the specified criteria.

For example, the following expression constraint matches all rows of the 816080008 | International Patient Summary which have been updated since 31st July 2021 (inclusive). Note that the referencedComponentId is the only field returned.

```
^ 816080008 | International Patient Summary
{{ M effectiveTime >= "20210731" }}
```

---

824 http://snomed.info/id/67415000
825 http://snomed.info/id/900000000000527005
826 http://snomed.info/id/67415000
827 http://snomed.info/id/900000000000443000
828 http://snomed.info/id/900000000000534007
829 http://snomed.info/id/900000000000534007
830 http://snomed.info/id/32570231000036109
831 http://snomed.info/id/816080008
832 http://snomed.info/id/816080008
6.10.5 Active Filter

Active filters enable an expression constraint to match on only those members of a reference set with a matching active status. Reference set rows are either active (i.e. active = 1 or active = "true") or inactive (i.e. active = 0 or active = "false"). By default, only active members of a reference set are included in the substrate.

For example, the following expression constraints returns the inactive members of the http://snomed.info/id/816080008 | International Patient Summary.

\[^{\text{816080008}}\text{|International Patient Summary}|^{\text{834}}\{\{M\text{ active }= 0\}\}\]

6.11 History Supplements

In this section, we illustrate how history supplements can be applied to an expression constraint to supplement the results with relevant inactive concepts. History supplements are specified inside double curly braces and begin with a plus sign (i.e. "+") followed by the word "HISTORY".

6.11.1 Background

When capturing new clinical data in an electronic health record (EHR), it is good practice to only allow active SNOMED CT concept identifiers to be recorded. However, SNOMED CT is a dynamic and evolving terminology that must remain consistent with current clinical practice and our evolving understanding of disease processes and treatments. As a result, content may change, become outdated, or need remodelling. As SNOMED CT evolves, concepts that were previously recorded in the EHR may subsequently be inactivated. For legal reasons, it is important that the concepts used at the time the data was recorded should persist in the health records. For this reason, the number of inactive SNOMED CT identifiers in an EHR may increase over time.

As most ECL queries typically return only active SNOMED CT concept identifiers, it may not be possible to retrieve health records containing inactive identifiers using a standard expression constraint. One solution to this challenge, is to execute the expression constraint over an old SNOMED CT edition, in which all required concepts were active. However, given that the logical definitions in SNOMED CT typically improve over time, it is generally accepted that the best ECL results can be obtained using the most recent edition. Therefore, a query approach utilising the most recent edition of SNOMED CT is preferred in many cases.

When a SNOMED CT concept is inactivated, the author first allocates an appropriate reason for the inactivation, and then links the inactivated concept to one or more replacements using historical association reference sets. These historical associations provide a clear understanding of the level of semantic equivalence between the inactivated concept and its replacements where they exist. Vendors can use these historical associations to supplement the active concepts in their query results, with inactive concepts which are linked via appropriate historical associations to the active query results.

On this page, we describe how 'history supplements' can be added to an ECL query, to augment the query results with relevant inactive concepts, and how the resulting queries can be used to retrieve a more complete set of matching health records.

---

833 http://snomed.info/id/816080008
834 http://snomed.info/id/816080008
6.11.2 History Supplements

6.11.2.1 Overview

The member filter syntax, described in 6.10 Member Filters (see page 121), can be used to augment the results of an expression constraint with a set of inactive concepts that are related via an historical association reference set. For example, the following expression constraint can be used to find all the active descendants (and self) of the concept 195967001 | Asthma | 835, plus any inactive concept that is linked to an active descendant (or self) of 195967001 | Asthma | 836 via a historical | SAME AS association reference set | 837 member.

```
<< 195967001 | Asthma | 838 OR
  ^ 9000000000000527005 | SAME AS association reference set | 839 {{ M targetComponentId = << 195967001 | Asthma | 840 }}
```

The ECL history supplement syntax can be used to simplify queries with this structure. For example, the above query can be expressed in a a shorter form as:

```
<< 195967001 | Asthma | 841 {{ + HISTORY ( 9000000000000527005 | SAME AS association reference set | 842 ) }}
```

6.11.2.2 Template

The general template (see page 0) for history supplements is shown below.

```
[[+ecl @ecl_query]] {{ + HISTORY ( [[+ecl @history_refset_query]] ) }}
```

This general template for history supplements is equivalent to the expanded version shown below. Please note that the first and last slot in this template have the same name, which indicates that they must be populated with the same value (which in this case is the ECL query being performed).

```
[[+ecl @ecl_query]] OR
  ^ [[+ecl @history_refset_query]] {{ M targetComponentId = [[+ecl @ecl_query]] }}
```

---

835 http://snomed.info/id/195967001
836 http://snomed.info/id/195967001
837 http://snomed.info/id/9000000000000527005
838 http://snomed.info/id/195967001
839 http://snomed.info/id/9000000000000527005
840 http://snomed.info/id/195967001
841 http://snomed.info/id/195967001
842 http://snomed.info/id/9000000000000527005
6.11.2.3 Profiles

To help implementers of clinical systems write suitable ECL queries that include an appropriate set of inactive concepts, three history supplement profiles are provided. These profiles are designed to support a range of use cases, depending on the level of precision and recall required for inactive content. The three history supplement profiles are described in the table below.

<table>
<thead>
<tr>
<th>History Profile</th>
<th>Purpose</th>
<th>Historical Association Reference Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY-MIN</td>
<td>Minimum: To support use cases requiring a high level of precision, only historical associations that have a one-to-one equivalence with their replacement are used. Example use case: Clinical decision support</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9000000000000527005</td>
</tr>
<tr>
<td>HISTORY-MOD</td>
<td>Moderate: To support use cases that must balance precision with recall, only historical associations that • Have a one-to-one equivalence with their replacement • Have a one-to-many equivalence with their replacement, or • Are replaced by a concept that represents the intended original meaning closely enough to be clinically useful are used. Example use cases: Clinical research, clinical audit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>9000000000000527005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9000000000000526001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9000000000000528000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1186924009</td>
</tr>
</tbody>
</table>

Please note that this history template does not support the 9000000000000525002 | MOVED FROM association reference set, as the referencedComponentId refers to the active concept, while the targetComponentId refers to the inactive concept (which is the opposite of typical historical associations). If supporting historical associations, it is recommended that these be added to the 9000000000000527005 | SAME AS association reference set, to ensure that the template pattern above can be consistently applied.

Also note that the 9000000000000524003 | MOVED TO association reference set can be ignored for the purposes of executing historical ECL queries.

843 http://snomed.info/id/9000000000000525002
844 http://snomed.info/id/9000000000000525002
845 http://snomed.info/id/9000000000000527005
846 http://snomed.info/id/9000000000000524003
847 http://snomed.info/id/9000000000000527005
848 http://snomed.info/id/9000000000000527005
849 http://snomed.info/id/9000000000000526001
850 http://snomed.info/id/9000000000000528000
851 http://snomed.info/id/1186924009
### 6. Examples

#### History Profile

<table>
<thead>
<tr>
<th>History Profile</th>
<th>Purpose</th>
<th>Historical Association Reference Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>HISTORY-MAX</td>
<td><strong>Maximum:</strong> To support use cases that require the highest level of recall, where precision is not as important, all possible historical associations are used. Example use case: Identifying patients for manual review.</td>
<td>&lt; 900000000000522004</td>
</tr>
</tbody>
</table>

For example, if a high level of precision is required, then the HISTORY-MIN profile may be used. The expression constraint below matches descendants or self of \[195967001\] Asthma, plus any inactive concept that is associated with a descendant or self of \[195967001\] Asthma in the 900000000000527005 | SAME AS association reference set or the 900000000000525002 | MOVED FROM association reference set.

```plaintext
<< 195967001 | Asthma | {{ + HISTORY-MIN }}
```

The above expression constraint is equivalent to the one below, with an expanded history supplement.

```plaintext
<< 195967001 | Asthma | {{ + HISTORY ( 900000000000527005 | SAME AS association reference set ) }}
```

Use cases that must balance the precision of associated inactive concepts with the level of recall, may use the HISTORY-MOD supplement. The following two expression constraint, which use the history supplement profile and the expanded history supplement respectively, are equivalent.

```plaintext
<< 195967001 | Asthma | {{ + HISTORY-MOD }}
```

```plaintext
<< 195967001 | Asthma | {{ + HISTORY ( 900000000000527005 | SAME AS association reference set ) OR 900000000000526001 | REPLACED BY association reference set OR 900000000000528000 | WAS A association reference set OR 1186924009 | PARTIALLY EQUIVALENT TO association reference set ) }}
```

---

852 http://snomed.info/id/9000000000000522004
853 http://snomed.info/id/195967001
854 http://snomed.info/id/195967001
855 http://snomed.info/id/900000000000527005
856 http://snomed.info/id/900000000000525002
857 http://snomed.info/id/195967001
858 http://snomed.info/id/195967001
859 http://snomed.info/id/900000000000527005
860 http://snomed.info/id/195967001
861 http://snomed.info/id/195967001
862 http://snomed.info/id/900000000000527005
863 http://snomed.info/id/900000000000526001
864 http://snomed.info/id/900000000000528000
865 http://snomed.info/id/1186924009

---

6. Examples – 127
And finally, use cases that require the highest level of recall, may use the HISTORY-MAX supplement profile. This profile uses all possible historical association reference sets to find any potentially relevant inactive concept. The following four expression constraints, which use (a) the history supplement profile, (b) the expanded history supplement, (c) the ANY wildcard symbol ('*'), and (d) the ‘history’ keyword on its own, are all equivalent. Please note that the 900000000000524003 | MOVED TO association reference set does not need to be included in the execution of this query, because the targetComponentId is assigned a namespace concept

\[
\text{195967001 | Asthma} \quad \{+ \text{HISTORY-MAX}\}
\]

\[
\text{195967001 | Asthma} \quad \{+ \text{HISTORY} \ (< \ 900000000000522004 \ | \text{Historical association reference set}\})\}
\]

\[
\text{195967001 | Asthma} \quad \{+ \text{HISTORY} (*)\}
\]

\[
\text{195967001 | Asthma} \quad \{+ \text{HISTORY}\}
\]

### 6.11.3 Use Case Examples

Here are two use cases that illustrate how these history supplements may be used in practice:

#### 6.11.3.1 Use Case 1

A clinical system is trying to count the number of patients who have had any type of referral to a service. The system attempts to use the following ECL query to find patient records with a matching procedure.

\[
\text{306206005 | Referral to service (procedure)}
\]

This query is successfully used to finds patient records containing active referral concepts, such as Referral to radiology service (procedure).

However, it is discovered that there are 738,090 patient records coded with the inactive SNOMED CT concept Refer to Radiology department (procedure), which should also be included in the patient count. The clinical system, therefore, adjusts its expression constraint query as shown below, to add a history supplement that includes all inactive concepts with the same meaning as one of the active referral concepts.

---

866 http://snomed.info/id/900000000000524003
867 http://snomed.info/id/195967001
868 http://snomed.info/id/195967001
869 http://snomed.info/id/900000000000522004
870 http://snomed.info/id/195967001
871 http://snomed.info/id/195967001
872 http://snomed.info/id/306206005
873 http://snomed.info/id/308461008
874 http://snomed.info/id/183598009

---
Because the expression constraint " " matches the active concept 308461008 | Referral to radiology service (procedure) | and a SAME AS association exists between the inactive concept 183598009 | Refer to Radiology department (procedure) | and the active concept 308461008 | Referral to radiology service (procedure) | , the above expression constraint will include the inactive concept 183598009 | Refer to Radiology department (procedure) | , and therefore successfully find the additional 738,090 patient records in which this inactive referral procedure is recorded.

6.11.3.2 Use Case 2

A clinician is trying to find all patients with any type of breast pain. Knowing that she will be reviewing the patient records prior to acting upon the information, she decides to use a maximal approach to searching historical records. She therefore uses the following ECL query:

She is delighted to see that patient records containing the inactive concept 315251009 | Pain of breast (finding) | are retrieved, as these are indeed relevant to her query. Behind the scenes, the clinical system was able to identify that this inactive concept may be relevant, because it is linked to the active concepts 1010235008 | Pain of left breast | and 1010237000 | Pain of right breast | (which are both a type of Pain of breast | ) via the POSSIBLY EQUIVALENT TO association reference set.

Note that this template uses the template syntax defined in the SNOMED CT Template Syntax specification, with the addition of an 'ECL' replacement type to indicate that the respective slot must be replaced by a valid ECL expression constraint. This extended template slot syntax is then used within an expression constraint to informally illustrate the pattern required when expanding a history supplement.

6.12 6.12 Top and Bottom

In this section we illustrate how a set of concepts can be filtered, using the top or bottom operators, to find the concepts that are the highest or lowest in the hierarchy within the set.
6.12.1 Top of set

Two consecutive exclamation marks followed by a 'greater than' sign (i.e. "!!>") indicates that the expression constraint is satisfied by the concepts from the results of the subexpression that have the highest position in the hierarchy, relative to one another. In other words the set of concepts, that is the result of the subexpression, will be filtered by concepts that have no ancestors within that set.

For example the following expression constraint represents the highest, or most general, concepts in the hierarchy within the set of 363698007 | Finding sites[^888] from the subtypes of 386617003 | Digestive system finding[^889].

\[ !!> ( < 386617003 | Digestive system finding[^890]. 363698007 | Finding site[^891]) \]

Using the long syntax, the above expression constraint may be represented as:

\[ top ( < 386617003 | Digestive system finding[^892]. 363698007 | Finding site[^893]) \]

An equivalent expression constraint without using the top operator can be written:

\[ ( < 386617003 | Digestive system finding[^894]. 363698007 | Finding site[^895]) \]
\[ MINUS ( < 386617003 | Digestive system finding[^896]. 363698007 | Finding site[^897]) \]

6.12.2 Bottom of set

Two consecutive exclamation marks followed by a 'less than' sign (i.e. "!!<") indicates that the expression constraint is satisfied by all concepts from the results of the subexpression that have the lowest position in the hierarchy, relative to one another. In other words the set of concepts, that is the result of the subexpression, will be filtered by concepts that have no descendants within that set.

For example, the following expression constraint represents the lowest, or most specific, concepts in the hierarchy within the set of concepts that are both ancestor-and-self of 427089005 | Diabetes mellitus due to cystic fibrosis[^898] and also within the 816080008 | International Patient Summary[^899] reference set:

\[ !!< ( < 427089005 | Diabetes mellitus due to cystic fibrosis[^898]. 816080008 | International Patient Summary[^899]) \]

[^888]: http://snomed.info/id/363698007
[^889]: http://snomed.info/id/386617003
[^890]: http://snomed.info/id/386617003
[^891]: http://snomed.info/id/363698007
[^892]: http://snomed.info/id/386617003
[^893]: http://snomed.info/id/363698007
[^894]: http://snomed.info/id/386617003
[^895]: http://snomed.info/id/363698007
[^896]: http://snomed.info/id/386617003
[^897]: http://snomed.info/id/363698007
[^898]: http://snomed.info/id/427089005
[^899]: http://snomed.info/id/816080008
6. Examples

6.12.3 Use Case Examples

Here are use cases that illustrate how the top and bottom operators may be used in practice:

6.12.3.1 Use Cases for Bottom

Not all clinical information systems use the same set of SNOMED CT concepts, for various reasons; many countries use their own national editions, countries without a national license may use a SNOMED CT freeset, national guidelines may dictate information model bindings that constrain the set of concepts used.

There are scenarios where it may be necessary to transform clinical records from one information system to another. For example a cross-border research project. If the source system has a broader or different set of concepts than the target system can use then a common strategy is to walk up the hierarchy to find the first common ancestor that is shared by both systems.

One example is development and use of value sets for cross-border sharing, as in MyHealth@EU, where not all participants have a SNOMED CT Affiliate License. The countries who are a member would like to use and share full-SNOMED value sets whereas non-members should only use concepts from the SNOMED CT freeset.

Another example is the NHS Emergency Care Data Set. This is a collection of UK nationally defined subsets for use in a specific context. The bottom operator could be used to transform a specific concept like 45133009 | Neurotoxic shellfish poisoning into a less specific concept that is within the 991411000000109 | Emergency care diagnosis simple reference set:

900 http://snomed.info/id/427089005
901 http://snomed.info/id/816080008
902 http://snomed.info/id/427089005
903 http://snomed.info/id/816080008
904 http://snomed.info/id/427089005
905 http://snomed.info/id/816080008
906 http://snomed.info/id/427089005
907 http://snomed.info/id/816080008
908 http://snomed.info/id/45133009
909 http://snomed.info/id/991411000000109
6. Examples

This would result in the set of concepts: 118940003 | Disorder of nervous system, 912 and 75258004 | Food poisoning, 913.
7. Implementation Considerations

When implementing the SNOMED CT Expression Constraint Language, the factors that need to be taken into consideration depend on what tasks are being performed. For example, implementations may require expression constraints to be authored, parsed, validated, executed, stored, displayed or exchanged.

The subsections below look at each of these tasks individually and provide a summary of the factors that should be considered prior to implementation. Please note that the guidance provided below is not a step-by-step how-to manual, but instead provides some general insights that we hope are helpful in implementing this language specification.

- 7.1 Authoring (see page 133)
- 7.2 Parsing (see page 135)
- 7.3 Validating (see page 136)
- 7.4 Executing (see page 136)
- 7.5 Storing (see page 136)
- 7.6 Displaying (see page 137)
- 7.7 Exchanging (see page 137)

7.1 7.1 Authoring

Authoring SNOMED CT Expression Constraints can be performed using two main techniques:

1. **Language-based authoring**: This technique involves the author constructing a SNOMED CT Expression Constraint using one of the syntaxes defined in Chapter 5.

2. **Form-based authoring**: This technique involves the author entering values into separate fields of a form, and the clinical system automatically composing the values together into a syntactically correct SNOMED CT Expression Constraint.

7.1.1 Language-Based Authoring

Language-based authoring is useful for situations in which ad hoc expression constraints must be defined which don’t necessarily conform to a consistent structure. For example, some expression constraints (e.g. those that define terminology bindings or predefined queries) may be authored by software developers during the design, development or customization of a clinical application. Other expression constraints (e.g. those used to define intentional reference sets or validation queries) may be defined by terminologists during the process of developing a SNOMED CT extension. Expression constraints may also be authored by users who wish to retrieve or analyse information stored in patient records using SNOMED CT (e.g. for clinical, epidemiological or research queries).

To use language-based authoring, the user must be familiar with the basic features of the Expression Constraint Language syntax. There are, however, a number of ways in which a tool can support the user while creating expression constraints, including:

- Validating the syntactical correctness of the expression constraint as it is authored;
- Checking the expression constraint for conformance against the concept model;
- Automatically populating or correcting the term associated with a concept reference;
- Providing integrated tools to search the SNOMED CT hierarchy for concept references to include in the expression constraint;
- Filtering the concept search to those concepts which are valid to use at the given point in the expression constraint (e.g. only showing attribute concepts, or those within the valid range of the given attribute); and
- Suggesting the set of valid operators or characters that may be used at a given point in the expression constraint;
7.1.2 Form-Based Authoring

Form-based authoring is particularly useful when non-technical users need to create constraints or queries which have a consistent structure. In these situations, it may be useful to either:

- Create an 'expression constraint template' in which the attribute values are populated with the values that the user enters into the associated fields of the form;
- Create a form-driven query tool to support a useful subset of possible query structures.

One scenario in which the first form-based approach may be used is when there is a terminology-based dependency between the values of two fields on a user interface. For example, Figure 4 illustrates a simplified Procedures form in which the coded value entered into the Procedure Type field must be a descendant of the coded value entered into the Procedure Category field. When a Procedure Category of "Surgery" (i.e. 387713003 | Surgical procedure

914) is selected, the expression constraint " < 387713003 | Surgical procedure

915" is used to populate the value list for the Procedure Type field.

![Figure 4: Authoring using expression constraint templates](image)

The second form-based authoring technique mentioned above is a form-driven query tool. Figure 5 below illustrates a very simple form-driven query tool, in which the user selects the required operator (e.g. 'ancestorOf', 'descendantOf', 'memberOf') and operand (e.g. 'Example Problem List') and then defines one or more attribute refinements.
7.2 Parsing

Parsing is the process of analysing a string of characters according to the rules of a formal grammar. Parsing a SNOMED CT Expression Constraint involves processing the expression constraint string using one of the ABNF syntax specifications defined in Chapter 5 (see page 21), and breaking it into its constituent parts. This creates a representation of the expression constraint that can be further processed. Parsing an expression constraint is required to perform syntactic validation, concept model validation or execution. It should be noted, when parsing, that all keywords in the language are case insensitive.

A number of parser development tools are available which can generate a parser from a context-free grammar written in ABNF, such as the one defined in this document. These tools include:

- APG
- aParse
- abnfgener

Please note, the ABNF syntax defined in this specification was tested using the APG Parser Generator (see page 0).

Other non-ABNF parser generators are also available which can be used with an alternate syntax representation – for example:

- ANTLR
- XText
- ACE

Some of these tools (e.g. XText and ACE) can also be used to generate authoring environments with features such as syntax highlighting and autocompletion.
Alternatively, an expression constraint parser can be created manually using a programming language such as Perl or C++.

7.3 Validating
SNOMED CT Expression Constraints can be automatically validated to ensure that they conform to a variety of rules, including:

- Expression constraints must conform to one of the syntaxes defined in Chapter 5 (see page 21). Syntactic validation can be performed using an expression parser, as described in Section 7.2 (see page 135);
- Expression constraints must conform to the concept model. This validation can be performed by comparing the parsed expression constraint against the rules defined in the SNOMED CT concept model;
- All concept references included in the expression constraint must be valid. In most cases this means that the concept references must refer to active concepts in the given version and edition of SNOMED CT;
- All concept references used to refer to attribute names must be a descendant of `246061005` | `Attribute`;
- All concept references to which a memberOf function is applied must be a descendant of `900000000000455006` | `Reference set`;
- All concept references to which a memberOf function is applied must contain only referencedComponentIds that refer to concepts.

Please note that some of these rules may not apply in all environments.

7.4 Executing
SNOMED CT Expression Constraints must be evaluated against a given SNOMED CT substrate in order to instantiate the matching set of concepts or expressions. There are a number of possible implementation strategies for the execution of SNOMED CT Expression Constraints, which depend in part on the storage format of the substrate. For example:

- Store SNOMED CT in a relational database, and translate each SNOMED CT Expression Constraint into one or more SQL statements;
- Store SNOMED CT in an RDF store, and translate each SNOMED CT Expression Constraint into a SPARQL query;
- Store SNOMED CT in an XML database, and translate each SNOMED CT Expression Constraint into one or more XQL statements;
- Write a bespoke query execution engine (e.g. in Java or C++) to return matching concepts or expressions.

Each of these strategies requires that the expression constraints are first parsed (and preferably validated) prior to execution.

7.5 Storing
Storing SNOMED CT Expression Constraints in an expression constraint library may be done for a variety of purposes, including:
• To enable expression constraints to be re-executed (without re-authoring) after updates are made to the SNOMED CT substrate or the expression constraint itself;
• To provide a library of terminology binding constraints against which record instances will be validated;
• To provide a library of concept model constraints against which terminology artefacts (e.g. extensions, expressions) will be validated;
• To provide a library of predefined queries that may be shared by multiple users;
• To provide a library of terminology binding constraints that may be shared within a standards community.

A library of SNOMED CT Expression Constraints may be implemented using a number of techniques, including:

- Creating a Query specification reference set that records the expression constraint as the 'query';
- Creating a customized RF2 reference set with one or more new attributes that allow the expression constraint string and relevant metadata to be recorded;
- Creating a table in a relational database to store the SNOMED CT Expression Constraint and associated metadata;
- Creating a text file with a consistent structural format to store the SNOMED CT Expression Constraint and associated metadata;

In many cases it is useful to assign a unique identifier to each expression constraint in the library, so that they can be indexed and referenced for faster retrieval.

7.6 Displaying

A number of options exist for displaying SNOMED CT Expression Constraints, including:

- Displaying the expression constraint using SNOMED CT Expression Constraint Language in its originally authored and stored form;
- Converting the expression constraint to use either all symbols (as per the Brief Syntax), or all human-readable operators (as per alternate text introduced in the Long Syntax);
- Enhancing the expression constraint by adding in terms that may have been omitted, or replacing the existing terms with either local-dialect Preferred Terms or Fully Specified Names;
- Hiding the SNOMED CT identifiers for each concept and displaying only the Preferred Terms;
- Enhancing the display by using different font colors for each different part of the expression constraint (e.g. identifiers, terms, vertical bars, and operators), and by using whitespace in a way that improves the readability of the expression;
- Automatically transforming the expression constraint into a human-readable string using a predefined algorithm. For example, a simple algorithm may convert the symbols to text and remove the concept identifiers – e.g. "Descendants of fracture of bone: Finding site = Descendants or self of arm". More sophisticated algorithms may use pattern matching and predefined templates to construct a more natural string;
- Representing the operators, operands and attribute values of the expression constraint by populating a structured form. This approach is primarily suited to expression constraints with a consistent template, where the form can be pre-designed.

Which of these options is most appropriate to use when displaying expression constraints, will depend on a number of factors, including the type of users that will be viewing the constraints, the scope of the required constraint functionality, and the capabilities of the system implementation.

7.7 Exchanging

SNOMED CT Expression Constraints can be shared between systems and users via a number of methods, including:
• Exchanging an expression constraint string which conforms to the Brief Syntax of the Expression Constraint Language\(^919\);

• Exchanging an expression constraint identifier, which can be unambiguously interpreted by the receiving system. If this approach is adopted it is recommended that an expression constraint repository is used to ensure that both the sending and receiving systems have a shared and consistent understanding of the meaning of each expression constraint.

Irrespective of the method used, it is recommended that the Brief Syntax of the SNOMED CT Expression Constraint Language\(^920\) be used as the normative syntax for the interoperable sharing of expression constraints.

\(^919\) http://snomed.org/ecl
\(^920\) http://snomed.org/ecl
8 Appendix A – Examples Of Valid Expressions

This appendix provides examples of expressions (both precoordinated and postcoordinated) which satisfy each of the expression constraints that were introduced in Chapter 6 (see page 64). This list of examples is not intended to be exhaustive, but rather to provide a representative sample to help clarify the meaning of each constraint. It is assumed that each particular usage of an expression constraint will clearly identify whether or not postcoordinated expressions are part of the valid substrate. Please refer to the SNOMED CT Languages Github repository for a set of text files containing each of these examples.

- A.1 Simple Expression Constraints - Valid Expressions (see page 139)
- A.2 Refinements - Valid Expressions (see page 142)
- A.3 Cardinality - Valid Expressions (see page 149)
- A.4 Conjunction and Disjunction - Valid Expressions (see page 155)
- A.5 Exclusion and Not Equals - Valid Expressions (see page 159)
- A.6 Nested Expression Constraints - Valid Expressions (see page 163)

### 8.1 A.1 Simple Expression Constraints - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Valid Expression</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>404684003</td>
<td>Clinical finding</td>
<td>404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>&lt;&lt; 73211009</td>
<td>Diabetes mellitus</td>
<td>73211009</td>
<td>Diabetes mellitus</td>
</tr>
<tr>
<td>64572001</td>
<td>Disease</td>
<td>64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>56265001</td>
<td>Heart disease</td>
<td>56265001</td>
<td>Heart disease</td>
</tr>
<tr>
<td>73211009</td>
<td>Diabetes mellitus type 1</td>
<td>73211009</td>
<td>Diabetes mellitus type 1</td>
</tr>
</tbody>
</table>

921 https://github.com/IHTSDO/SNOMEDCT-Languages
922 http://snomed.info/id/404684003
923 http://snomed.info/id/404684003
924 http://snomed.info/id/404684003
925 http://snomed.info/id/64572001
926 http://snomed.info/id/404684003
927 http://snomed.info/id/363698007
928 http://snomed.info/id/80891009
929 http://snomed.info/id/56265001
930 http://snomed.info/id/73211009
931 http://snomed.info/id/73211009
932 http://snomed.info/id/73211009
933 http://snomed.info/id/73211009
934 http://snomed.info/id/42752001
935 http://snomed.info/id/61823004
936 http://snomed.info/id/46635009
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`105401000119101</td>
<td>Diabetes mellitus due to pancreatic injury`&lt;sup&gt;936&lt;/sup&gt;</td>
</tr>
<tr>
<td>`&lt;! 404684003</td>
<td>Clinical finding`&lt;sup&gt;937&lt;/sup&gt;</td>
</tr>
<tr>
<td>`64572001</td>
<td>Disease`&lt;sup&gt;938&lt;/sup&gt;</td>
</tr>
<tr>
<td>`267038008</td>
<td>Edema`&lt;sup&gt;942&lt;/sup&gt;</td>
</tr>
<tr>
<td>`404684003</td>
<td>Clinical finding<code>&lt;sup&gt;939&lt;/sup&gt; : </code>116676008</td>
</tr>
<tr>
<td>`&gt; 40541001</td>
<td>Acute pulmonary edema`&lt;sup&gt;943&lt;/sup&gt;</td>
</tr>
<tr>
<td>`111273006</td>
<td>Acute respiratory disease`&lt;sup&gt;944&lt;/sup&gt;</td>
</tr>
<tr>
<td>`404684003</td>
<td>Clinical finding`&lt;sup&gt;950&lt;/sup&gt;</td>
</tr>
<tr>
<td>`138875005</td>
<td>SNOMED CT concept`&lt;sup&gt;951&lt;/sup&gt;</td>
</tr>
<tr>
<td>`&gt;&gt; 40541001</td>
<td>Acute pulmonary edema`&lt;sup&gt;952&lt;/sup&gt;</td>
</tr>
<tr>
<td>`40541001</td>
<td>Acute pulmonary edema`&lt;sup&gt;953&lt;/sup&gt;</td>
</tr>
<tr>
<td>`111273006</td>
<td>Acute respiratory disease`&lt;sup&gt;954&lt;/sup&gt;</td>
</tr>
<tr>
<td>`404684003</td>
<td>Clinical finding`&lt;sup&gt;955&lt;/sup&gt;</td>
</tr>
<tr>
<td>`64572001</td>
<td>Disease<code>&lt;sup&gt;956&lt;/sup&gt; : </code>263502005</td>
</tr>
<tr>
<td><code>{ </code>116676008</td>
<td>Associated morphology<code>&lt;sup&gt;959&lt;/sup&gt; = </code>40829002</td>
</tr>
<tr>
<td><code>{ </code>363698007</td>
<td>Finding site<code>&lt;sup&gt;961&lt;/sup&gt; = </code>39607008</td>
</tr>
<tr>
<td>SNOMED CT concept</td>
<td>Expression Constraint Language</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>138875005</td>
<td>SNOMED CT concept</td>
</tr>
<tr>
<td>111273006</td>
<td>Acute respiratory disease</td>
</tr>
<tr>
<td>19242006</td>
<td>Pulmonary edema</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>394659003</td>
<td>Acute coronary syndrome</td>
</tr>
<tr>
<td>194828000</td>
<td>Angina</td>
</tr>
<tr>
<td>29857009</td>
<td>Chest pain</td>
</tr>
<tr>
<td>404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>405813007</td>
<td>Procedure site - Direct</td>
</tr>
</tbody>
</table>

963 http://snomed.info/id/138875005
964 http://snomed.info/id/40541001
965 http://snomed.info/id/111273006
966 http://snomed.info/id/19829001
967 http://snomed.info/id/116676008
968 http://snomed.info/id/79654002
969 http://snomed.info/id/363698007
970 http://snomed.info/id/39607008
971 http://snomed.info/id/19242006
972 http://snomed.info/id/700043003
973 http://snomed.info/id/394659003
974 http://snomed.info/id/194828000
975 http://snomed.info/id/29857009
976 http://snomed.info/id/138875005
977 http://snomed.info/id/404684003
978 http://snomed.info/id/363698007
979 http://snomed.info/id/80891009
980 http://snomed.info/id/404684003
981 http://snomed.info/id/71388002
982 http://snomed.info/id/405813007
983 http://snomed.info/id/66754008
Where necessary, these examples make some assumptions about the membership of the example reference sets.

Please note that this makes the assumption that the given expression constraint is executed against a finite set of expressions that has been pre-classified (e.g. in an expression repository), and that after classification there are no intermediate expressions between this expression and 404684003 | Clinical finding |

Please note that this makes the assumption that the given expression constraint is executed against a finite set of expressions that has been pre-classified (e.g. in an expression repository), and that after classification there are no intermediate expressions between 40541001 | Acute pulmonary edema |

### 8.2  A.2 Refinements - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung</td>
<td></td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
<td></td>
</tr>
<tr>
<td>79654002</td>
<td>Edema</td>
<td></td>
</tr>
<tr>
<td>11468004</td>
<td>Postoperative pulmonary edema</td>
<td></td>
</tr>
<tr>
<td>210051003</td>
<td>Injury to heart and lung</td>
<td></td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
<td></td>
</tr>
<tr>
<td>79654002</td>
<td>Edema</td>
<td></td>
</tr>
<tr>
<td>276637009</td>
<td>Hemorrhagic pulmonary edema</td>
<td></td>
</tr>
</tbody>
</table>

---

984 http://snomed.info/id/322236009
985 http://snomed.info/id/373873005
986 http://snomed.info/id/127489000
986 http://snomed.info/id/412031009
988 http://snomed.info/id/404684003
989 http://snomed.info/id/40541001
990 http://snomed.info/id/19829001
991 http://snomed.info/id/116676008
992 http://snomed.info/id/79654002
993 http://snomed.info/id/11468004
994 http://snomed.info/id/210051003
995 http://snomed.info/id/116676008
996 http://snomed.info/id/79654002
997 http://snomed.info/id/276637009
<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>233709006</td>
<td>Toxic pulmonary edema</td>
</tr>
<tr>
<td>275504005</td>
<td>Lung cyst</td>
</tr>
<tr>
<td>363698007</td>
<td>Pulmonary valve stenosis</td>
</tr>
<tr>
<td>404684003</td>
<td>Clinical finding</td>
</tr>
</tbody>
</table>

### Examples

< 19829001 | Disorder of lung |
116676008 | Associated morphology |
79654002 | Edema |

= 116676008 | Associated morphology |

= << 415582006 | Stenosis |

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>233711002</td>
<td>Oxygen-induced pulmonary edema</td>
</tr>
<tr>
<td>19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>233711002</td>
<td>Oxygen-induced pulmonary edema</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
</tbody>
</table>

= 40829002 | Acute edema |

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>56786000</td>
<td>Pulmonic valve stenosis</td>
</tr>
<tr>
<td>56786000</td>
<td>Pulmonic valve stenosis</td>
</tr>
<tr>
<td>363698007</td>
<td>Pulmonic valve stenosis</td>
</tr>
<tr>
<td>90318009</td>
<td>Structure of anulus fibrosus of pulmonary artery</td>
</tr>
<tr>
<td>88015002</td>
<td>Partial stenosis</td>
</tr>
</tbody>
</table>

= 39057004 | Pulmonary valve structure |

Appendix A – Examples Of Valid Expressions – 143
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>: 246075003</td>
<td>Causative agent Paracetamol</td>
</tr>
<tr>
<td>387517004</td>
<td>Stenosis</td>
</tr>
<tr>
<td>1024</td>
<td>415582006</td>
</tr>
<tr>
<td>1025</td>
<td>246075003</td>
</tr>
<tr>
<td>1026</td>
<td>387517004</td>
</tr>
<tr>
<td>1027</td>
<td>295124009</td>
</tr>
<tr>
<td>1028</td>
<td>Paracetamol overdose</td>
</tr>
<tr>
<td>1029</td>
<td>404684003</td>
</tr>
<tr>
<td>1030</td>
<td>Clinical finding Paracetamol</td>
</tr>
<tr>
<td>1031</td>
<td>387517004</td>
</tr>
<tr>
<td>1032</td>
<td>292042007</td>
</tr>
<tr>
<td>1033</td>
<td>Adverse reaction to paracetamol</td>
</tr>
<tr>
<td></td>
<td>86299006</td>
</tr>
<tr>
<td>1034</td>
<td>Tetralogy of Fallot</td>
</tr>
<tr>
<td>1035</td>
<td>204351007</td>
</tr>
<tr>
<td>1036</td>
<td>Fallot's triology</td>
</tr>
<tr>
<td>1037</td>
<td>404684003</td>
</tr>
<tr>
<td>1038</td>
<td>Clinical finding Paracetamol</td>
</tr>
<tr>
<td>1039</td>
<td>387517004</td>
</tr>
<tr>
<td>1040</td>
<td>246075003</td>
</tr>
<tr>
<td>1041</td>
<td>387517004</td>
</tr>
<tr>
<td>1042</td>
<td>363698007</td>
</tr>
<tr>
<td>1043</td>
<td>295124009</td>
</tr>
<tr>
<td>1044</td>
<td>415582006</td>
</tr>
<tr>
<td>1045</td>
<td>363698007</td>
</tr>
<tr>
<td>1046</td>
<td>404684003</td>
</tr>
<tr>
<td>1047</td>
<td>363698007</td>
</tr>
<tr>
<td>1048</td>
<td>53085002</td>
</tr>
<tr>
<td>1049</td>
<td>363698007</td>
</tr>
<tr>
<td>1050</td>
<td>125521000</td>
</tr>
<tr>
<td>1051</td>
<td>404684003</td>
</tr>
<tr>
<td>1052</td>
<td>363698007</td>
</tr>
<tr>
<td>1053</td>
<td>30815002</td>
</tr>
<tr>
<td>1054</td>
<td>363698007</td>
</tr>
<tr>
<td>1055</td>
<td>125521000</td>
</tr>
<tr>
<td>1056</td>
<td>404684003</td>
</tr>
<tr>
<td>1057</td>
<td>363698007</td>
</tr>
<tr>
<td>1058</td>
<td>404684003</td>
</tr>
<tr>
<td>1059</td>
<td>363698007</td>
</tr>
<tr>
<td>1060</td>
<td>125521000</td>
</tr>
<tr>
<td>1061</td>
<td>404684003</td>
</tr>
<tr>
<td>1062</td>
<td>363698007</td>
</tr>
<tr>
<td>1063</td>
<td>404684003</td>
</tr>
<tr>
<td>1064</td>
<td>363698007</td>
</tr>
<tr>
<td>1065</td>
<td>125521000</td>
</tr>
<tr>
<td>1066</td>
<td>404684003</td>
</tr>
<tr>
<td>1067</td>
<td>363698007</td>
</tr>
<tr>
<td>1068</td>
<td>404684003</td>
</tr>
<tr>
<td>1069</td>
<td>363698007</td>
</tr>
<tr>
<td>1070</td>
<td>125521000</td>
</tr>
<tr>
<td>1071</td>
<td>404684003</td>
</tr>
<tr>
<td>1072</td>
<td>363698007</td>
</tr>
<tr>
<td>1073</td>
<td>404684003</td>
</tr>
<tr>
<td>1074</td>
<td>363698007</td>
</tr>
<tr>
<td>1075</td>
<td>125521000</td>
</tr>
<tr>
<td>1076</td>
<td>404684003</td>
</tr>
<tr>
<td>1077</td>
<td>363698007</td>
</tr>
</tbody>
</table>

1024 [http://snomed.info/id/116676008](http://snomed.info/id/116676008)
1025 [http://snomed.info/id/415582006](http://snomed.info/id/415582006)
1026 [http://snomed.info/id/246075003](http://snomed.info/id/246075003)
1027 [http://snomed.info/id/387517004](http://snomed.info/id/387517004)
1028 [http://snomed.info/id/295124009](http://snomed.info/id/295124009)
1029 [http://snomed.info/id/404684003](http://snomed.info/id/404684003)
1030 [http://snomed.info/id/387517004](http://snomed.info/id/387517004)
1031 [http://snomed.info/id/292042007](http://snomed.info/id/292042007)
1032 [http://snomed.info/id/404684003](http://snomed.info/id/404684003)
1033 [http://snomed.info/id/363698007](http://snomed.info/id/363698007)
1034 [http://snomed.info/id/363698007](http://snomed.info/id/363698007)
1035 [http://snomed.info/id/39057004](http://snomed.info/id/39057004)
1036 [http://snomed.info/id/116676008](http://snomed.info/id/116676008)
1037 [http://snomed.info/id/246075003](http://snomed.info/id/246075003)
1038 [http://snomed.info/id/363698007](http://snomed.info/id/363698007)
1039 [http://snomed.info/id/53085002](http://snomed.info/id/53085002)
1040 [http://snomed.info/id/116676008](http://snomed.info/id/116676008)
1041 [http://snomed.info/id/56246009](http://snomed.info/id/56246009)
1042 [http://snomed.info/id/86299006](http://snomed.info/id/86299006)
1043 [http://snomed.info/id/404684003](http://snomed.info/id/404684003)
1044 [http://snomed.info/id/363698007](http://snomed.info/id/363698007)
1045 [http://snomed.info/id/31689007](http://snomed.info/id/31689007)
1046 [http://snomed.info/id/116676008](http://snomed.info/id/116676008)
1047 [http://snomed.info/id/415582006](http://snomed.info/id/415582006)
1048 [http://snomed.info/id/363698007](http://snomed.info/id/363698007)
1049 [http://snomed.info/id/53085002](http://snomed.info/id/53085002)
1050 [http://snomed.info/id/116676008](http://snomed.info/id/116676008)
1051 [http://snomed.info/id/125521000](http://snomed.info/id/125521000)
1052 [http://snomed.info/id/204351007](http://snomed.info/id/204351007)
< 404684003 | Clinical finding 1053 : 
<< 47429007 | Associated with 1054 = << 
267038008 | Edema 1055 

230580009 | Myxedema neuropathy 1056 

95356008 | Mucosal ulcer 1057 : 
42752001 | Due to 1058 = 19242006 | Pulmonary edema 1059 

< 27658006 | Amoxicillin 1060 : 
411116001 | Has dose form 1061 = 
<< 385055001 | Tablet dose form 1062 , 
{ 179999999100 | Has basis of strength 1063 } = 
( 219999999102 | Amoxicillin only 1064 : 
189999999103 | Has strength magnitude 1065 >= 200, 
199999999101 | Has strength unit 1066 = 258684004 | mg 1067 ) 

374644001 | Amoxicillin trihydrate 200 mg tablet 1068 

27658006 | Amoxicillin 1069 : 
411116001 | Has dose form 1070 = 
421026006 | Oral tablet 1071 , 
{ 127489000 | Has active ingredient 1072 = 
96068000 | Amoxicillin trihydrate 1073 } 
179999999100 | Has basis of strength 1074 = ( 
219999999102 | Amoxicillin only 1075 : 
189999999103 | Has strength magnitude 1076 = #500, 
199999999101 | Has strength unit 1077 = 258684004 | mg 1078 ) 

1053 http://snomed.info/id/404684003 
1054 http://snomed.info/id/47429007 
1055 http://snomed.info/id/267038008 
1056 http://snomed.info/id/230580009 
1057 http://snomed.info/id/95356008 
1058 http://snomed.info/id/42752001 
1059 http://snomed.info/id/19242006 
1060 http://snomed.info/id/27658006 
1061 http://snomed.info/id/411116001 
1062 http://snomed.info/id/385055001 
1063 http://snomed.org/fictid#179999999100 
1064 http://snomed.org/fictid#219999999102 
1065 http://snomed.org/fictid#189999999103 
1066 http://snomed.org/fictid#199999999101 
1067 http://snomed.info/id/258684004 
1068 http://snomed.info/id/374644001 
1069 http://snomed.info/id/27658006 
1070 http://snomed.info/id/411116001 
1071 http://snomed.info/id/421026006 
1072 http://snomed.info/id/127489000 
1073 http://snomed.info/id/96068000 
1074 http://snomed.org/fictid#179999999100 
1075 http://snomed.org/fictid#219999999102 
1076 http://snomed.org/fictid#189999999103 
1077 http://snomed.org/fictid#199999999101 
1078 http://snomed.info/id/258684004
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 27658006</td>
<td>Amoxicillin</td>
</tr>
<tr>
<td>411116001</td>
<td>Has dose form</td>
</tr>
<tr>
<td>`{ 179999999100</td>
<td>Has basis of strength</td>
</tr>
<tr>
<td>189999999103</td>
<td>Has strength magnitude</td>
</tr>
<tr>
<td>199999999101</td>
<td>Has strength unit</td>
</tr>
<tr>
<td>258684004</td>
<td>mg</td>
</tr>
<tr>
<td>373873005</td>
<td>Pharmaceutical / biologic product</td>
</tr>
<tr>
<td>209999999104</td>
<td>Has trade name</td>
</tr>
<tr>
<td>27658006</td>
<td>Amoxicillin</td>
</tr>
<tr>
<td>411116001</td>
<td>Has dose form</td>
</tr>
<tr>
<td>`{ 179999999100</td>
<td>Has basis of strength</td>
</tr>
<tr>
<td>189999999103</td>
<td>Has strength magnitude</td>
</tr>
<tr>
<td>199999999101</td>
<td>Has strength unit</td>
</tr>
<tr>
<td>373873005</td>
<td>Pharmaceutical / biologic product</td>
</tr>
<tr>
<td>209999999104</td>
<td>Has trade name</td>
</tr>
</tbody>
</table>

---

1079 http://snomed.info/id/27658006
1080 http://snomed.info/id/411116001
1081 http://snomed.info/id/385055001
1082 http://snomed.org/fictid#179999999100
1083 http://snomed.org/fictid#219999999102
1084 http://snomed.org/fictid#189999999103
1085 http://snomed.org/fictid#199999999101
1086 http://snomed.info/id/258684004
1087 http://snomed.info/id/374646004
1088 http://snomed.info/id/27658006
1089 http://snomed.info/id/411116001
1090 http://snomed.info/id/421026006
1091 http://snomed.info/id/127489000
1092 http://snomed.info/id/412031009
1093 http://snomed.org/fictid#179999999100
1094 http://snomed.org/fictid#219999999102
1095 http://snomed.info/id/373873005
1096 http://snomed.info/id/258684004
1097 http://snomed.info/id/374646004
1098 http://snomed.info/id/27658006
1099 http://snomed.info/id/411116001
1100 http://snomed.info/id/373873005
1101 http://snomed.info/id/127489000
1102 http://snomed.info/id/412031009
1103 http://snomed.org/fictid#209999999104

---

Appendix A – Examples Of Valid Expressions – 146
<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 91723000</td>
<td>Anatomical structure: Humerus, Right, Left</td>
</tr>
<tr>
<td>R 363698007</td>
<td>Finding site: Fracture of bone</td>
</tr>
<tr>
<td>`&lt; 125605004</td>
<td>Fracture of bone, Left</td>
</tr>
<tr>
<td>71341001</td>
<td>Femur, Right</td>
</tr>
<tr>
<td>85050009</td>
<td>Humerus, 272741003, Laterality, 7771000, Left</td>
</tr>
<tr>
<td>85050009</td>
<td>Humerus, 272741003, Laterality, 7771000, Left</td>
</tr>
<tr>
<td>71341001</td>
<td>Femur, Right</td>
</tr>
<tr>
<td>`&lt; 125605004</td>
<td>Fracture of bone, Left</td>
</tr>
<tr>
<td>85050009</td>
<td>Humerus, 272741003, Laterality, 7771000, Left</td>
</tr>
<tr>
<td>85050009</td>
<td>Humerus, 272741003, Laterality, 7771000, Left</td>
</tr>
<tr>
<td>71341001</td>
<td>Femur, Right</td>
</tr>
<tr>
<td>`&lt; 105590001</td>
<td>Substance, Has active ingredient: Amoxicillin</td>
</tr>
<tr>
<td>R &lt;&lt; 127489000</td>
<td>Product containing amoxicillin</td>
</tr>
<tr>
<td>`&lt; 27658006</td>
<td></td>
</tr>
<tr>
<td>395938000</td>
<td>Clavulanate potassium</td>
</tr>
</tbody>
</table>

1104 http://snomed.info/id/91723000
1105 http://snomed.info/id/363698007
1106 http://snomed.info/id/125605004
1107 http://snomed.info/id/85050009
1108 http://snomed.info/id/85050009
1109 http://snomed.info/id/272741003
1110 http://snomed.info/id/7771000
1111 http://snomed.info/id/71341001
1112 http://snomed.info/id/71341001
1113 http://snomed.info/id/272741003
1114 http://snomed.info/id/24028007
1115 http://snomed.info/id/125605004
1116 http://snomed.info/id/363698007
1117 http://snomed.info/id/85050009
1118 http://snomed.info/id/85050009
1119 http://snomed.info/id/272741003
1120 http://snomed.info/id/7771000
1121 http://snomed.info/id/71341001
1122 http://snomed.info/id/71341001
1123 http://snomed.info/id/71341001
1124 http://snomed.info/id/24028007
1125 http://snomed.info/id/127489000
1126 http://snomed.info/id/105590001
1127 http://snomed.info/id/27658006
1128 http://snomed.info/id/395938000
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Concept Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 27658006</td>
<td>Product containing amoxicillin</td>
<td>387137007</td>
</tr>
<tr>
<td>`&lt;= 1129</td>
<td>Clavulanate potassium</td>
<td>387137007</td>
</tr>
<tr>
<td>`&lt;= 1134</td>
<td>Omeprazole</td>
<td>1129</td>
</tr>
<tr>
<td>`&lt; 27658006</td>
<td>Product containing amoxicillin</td>
<td>387137007</td>
</tr>
<tr>
<td>`&lt;= 1135</td>
<td>Clavulanate potassium</td>
<td>387137007</td>
</tr>
<tr>
<td>`&lt;= 1136</td>
<td>Edema</td>
<td>19242006</td>
</tr>
<tr>
<td>`&lt;= 1137</td>
<td>Edema</td>
<td>19242006</td>
</tr>
<tr>
<td>`&lt;= 1138</td>
<td>Edema</td>
<td>19242006</td>
</tr>
<tr>
<td>`&lt;= 1139</td>
<td>Edema</td>
<td>19242006</td>
</tr>
<tr>
<td>`&lt; 263225007</td>
<td>Hip fracture</td>
<td>263225007</td>
</tr>
</tbody>
</table>

---

1129 http://snomed.info/id/387137007
1130 http://snomed.info/id/27658006
1131 http://snomed.info/id/127489000
1132 http://snomed.info/id/395938000
1133 http://snomed.info/id/387137007
1134 http://snomed.info/id/404684003
1135 http://snomed.info/id/79654002
1136 http://snomed.info/id/19242006
1137 http://snomed.info/id/404684003
1138 http://snomed.info/id/116676008
1139 http://snomed.info/id/79654002
1140 http://snomed.info/id/404684003
1141 http://snomed.info/id/116676008
1142 http://snomed.info/id/404684003
1143 http://snomed.info/id/116676008
1144 http://snomed.info/id/116676008
1145 http://snomed.info/id/79654002
1146 http://snomed.info/id/263225007
1147 http://snomed.info/id/404684003
1148 http://snomed.info/id/116676008
1149 http://snomed.info/id/72704001
Please note that some of these examples are based on a hypothetical drug concept model. These examples are not intended to reflect any specific drug model. SNOMED CT identifiers with the '9999999' namespace were created for example only, and should not be used in a production environment.

8.3 A.3 Cardinality - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Valid Expression</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>322236009</td>
<td>Paracetamol 500mg tablet</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
<td>1151</td>
<td>=</td>
</tr>
<tr>
<td>1150</td>
<td>: [1..3]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 373873005 | Pharmaceutical / biologic product | 373873005 | :
| 1154 | : |
| 1155 | = |
| 127489000 | Has active ingredient | 1159 | = |
| 412031009 | Paracetamol or derivative | 1160 |
| 322236009 | Paracetamol 500mg tablet | 1153 |
| 370166004 | Aspirin 325mg tablet | 1166 |
| 373873005 | Pharmaceutical / biologic product | 373873005 | :
| 1167 | : |
| 127489000 | Has active ingredient | 1160 |
| 412031009 | Paracetamol or derivative | 1160 |
| 127489000 | Has active ingredient | 1161 | = |
| 387494007 | Codeine |
| 404826002 | Benzocaine + butamben + tetracaine hydrochloride | 412031009 | Paracetamol or derivative |
| 1157 |

1150 http://snomed.info/id/373873005
1151 http://snomed.info/id/127489000
1152 http://snomed.info/id/105590001
1153 http://snomed.info/id/322236009
1154 http://snomed.info/id/373873005
1155 http://snomed.info/id/127489000
1156 http://snomed.info/id/412031009
1157 http://snomed.info/id/404826002
1158 http://snomed.info/id/373873005
1159 http://snomed.info/id/127489000
1160 http://snomed.info/id/412031009
1161 http://snomed.info/id/127489000
1162 http://snomed.info/id/387494007
1163 http://snomed.info/id/373873005
1164 http://snomed.info/id/370166004
1165 http://snomed.info/id/373873005
### Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Semantics</th>
</tr>
</thead>
<tbody>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td>7947003</td>
<td>Aspirin</td>
</tr>
</tbody>
</table>

References:

1164 http://snomed.info/id/127489000
1165 http://snomed.info/id/105590001
1166 http://snomed.info/id/127489000
1167 http://snomed.info/id/127489000
1168 http://snomed.info/id/127489000
1169 http://snomed.info/id/127489000
1170 http://snomed.info/id/370166004
1171 http://snomed.info/id/127489000
1172 http://snomed.info/id/105590001
1173 http://snomed.org/fictid#279999999108
1174 http://snomed.info/id/3738730005
1175 http://snomed.info/id/127489000
1176 http://snomed.info/id/127489000
1177 http://snomed.info/id/370166004
1178 http://snomed.info/id/3738730005
1179 http://snomed.info/id/127489000
1180 http://snomed.info/id/105590001
1181 http://snomed.info/id/7947003
1182 http://snomed.info/id/3738730005
1183 http://snomed.info/id/127489000
1184 http://snomed.info/id/127489000
1185 http://snomed.info/id/127489000
1186 http://snomed.info/id/255641001
1187 http://snomed.info/id/127489000
1188 http://snomed.info/id/387458008
1189 http://snomed.info/id/437867004
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>&lt; 91723000</td>
<td>Anatomical structure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>&lt; 91723000</td>
<td>Anatomical structure</td>
</tr>
</tbody>
</table>

1190 http://snomed.info/id/404684003
1191 http://snomed.info/id/363698007
1192 http://snomed.info/id/91723000
1193 http://snomed.info/id/125596004
1194 http://snomed.info/id/404684003
1195 http://snomed.info/id/116676008
1196 http://snomed.info/id/72704001
1197 http://snomed.info/id/363698007
1198 http://snomed.info/id/299701004
1199 http://snomed.info/id/62413002
1200 http://snomed.info/id/702468001
1201 http://snomed.info/id/363698007
1202 http://snomed.info/id/91723000
1203 http://snomed.info/id/72704001
1204 http://snomed.info/id/86299006
1205 http://snomed.info/id/404684003
1206 http://snomed.info/id/363698007
1207 http://snomed.info/id/116676008
1208 http://snomed.info/id/72704001
1209 http://snomed.info/id/363698007
1210 http://snomed.info/id/116676008
1211 http://snomed.info/id/72704001
1212 http://snomed.info/id/363698007
1213 http://snomed.info/id/702468001
Appendix A – Examples Of Valid Expressions

< 404684003 | Clinical finding | 1214 |
\{ 2..* \} 363698007 | finding site |
1215 =
< 91723000 | Anatomical structure | 1216

- 64572001 | Disease | 1217 |
\{ 116676008 | Associated morphology |
1218 = 396351009 | Congenital septal defect |
363698007 | Finding site |
1220 = 25943004 | Structure of atrioventricular node |
363698007 | Finding site |
1222 = 113262008 | Thoracic aorta structure |
\{ 116676008 | Associated morphology |
1224 = 90141005 | Congenital hypertrophy |
363698007 | Finding site |
1226 = 244384009 | Entire right ventricle |
1227

< 373873005 | Pharmaceutical / biologic product | 1228 |
\{ 1..3 \} \{ 1..* \}
127489000 | Has active ingredient |
1229 =< 105590001 | Substance |
1230

322236009 | Paracetamol 500mg tablet | 1231

373873005 | Pharmaceutical / biologic product | 1232 |
\{ 127489000 | Has active ingredient |
1233 = 412031009 | Paracetamol or derivative |
1234

404826002 | Benzocaine + butamben + tetracaine hydrochloride | 1235

373873005 | Pharmaceutical / biologic product | 1236 |
\{ 127489000 | Has active ingredient |

---

1214 http://snomed.info/id/404684003
1215 http://snomed.info/id/363698007
1216 http://snomed.info/id/91723000
1217 http://snomed.info/id/64572001
1218 http://snomed.info/id/116676008
1219 http://snomed.info/id/396351009
1220 http://snomed.info/id/363698007
1221 http://snomed.info/id/25943004
1222 http://snomed.info/id/363698007
1223 http://snomed.info/id/113262008
1224 http://snomed.info/id/116676008
1225 http://snomed.info/id/90141005
1226 http://snomed.info/id/363698007
1227 http://snomed.info/id/244384009
1228 http://snomed.info/id/373873005
1229 http://snomed.info/id/127489000
1230 http://snomed.info/id/412031009
1231 http://snomed.info/id/322236009
1232 http://snomed.info/id/373873005
1233 http://snomed.info/id/127489000
1234 http://snomed.info/id/412031009
1235 http://snomed.info/id/404826002
1236 http://snomed.info/id/373873005
Appendix A – Examples Of Valid Expressions

1237 \(=\) 412031009 \(\text{Paracetamol or derivative}\),
1238 \{ 127489000 \(\text{Has active ingredient}\) \}
1239 \(=\) 387494007 \(\text{Codeine}\)

\(< 373873005 \text{ Pharmaceutical / biologic product }^{1241} : [0..1] \{ \n\begin{align*}
127489000 & \text{ Has active ingredient }^{1242} \\
1242 & = < 105590001 \text{ Substance }^{1243}
\end{align*} \}
\)

\(11111527999999108 \text{ Inert tablet }^{1244}
370166004 \text{ Aspirin 325mg tablet }^{1248}

\(< 373873005 \text{ Pharmaceutical / biologic product }^{1249} : [1..*] \{ \n\begin{align*}
127489000 & \text{ Has active ingredient }^{1250} \\
1250 & = < 105590001 \text{ Substance }^{1251}
\end{align*} \}
\)

\(370166004 \text{ Aspirin 325mg tablet }^{1252}
373873005 \text{ Pharmaceutical / biologic product }^{1253} :
1254 \(=\) 412031009 \(\text{Paracetamol or derivative}\),
1255 \{ 127489000 \(\text{Has active ingredient}\) \},
1256 \(=\) 387494007 \(\text{Codeine}\)
| Expression Constraint Language – Expression Constraint Language - Specification and Guide |

| 1258 | Clinical finding
| 1259 | Finding site
| 1260 | Anatomical structure
| 1261 | Injury of elbow

| 1262 | Clinical finding
| 1263 | Finding site
| 1264 | Bone of forearm
| 1265 | Bone structure of radius

| 1266 | Clinical finding
| 1267 | Finding site
| 1268 | Bone structure of radius

| 1269 | Clinical finding
| 1270 | Finding site
| 1271 | Associated morphology
| 1272 | Pulmonary valve structure
| 1273 | Tetralogy of Fallot
| 1274 | Tetralogy of Fallot
| 1275 | Stenosis

**Appendix A – Examples Of Valid Expressions**

- `<404684003|Clinical finding: {1..1} [363698007|Finding site] 91723000|Anatomical structure}`
- `<404684003|Clinical finding: [0..0] {2..*} [363698007|Finding site] 91723000|Anatomical structure>`

---

The SNOMED CT identifiers created with the '9999999' namespace are for example only, and should not be used in a production environment.

As mentioned earlier, only non-redundant defining attributes are included in the cardinality count.

Because `<a href="http://snomed.info/id/62413002" title="62413002 | Bone structure of radius |" class="external-link" rel="nofollow">62413002</a>` is a subtype of `<a href="http://snomed.info/id/299701004" title="299701004 | Bone of forearm |" class="external-link" rel="nofollow">299701004</a>`, the refinement `<a href="http://snomed.info/id/363698007" title="363698007 | Finding site |" class="external-link" rel="nofollow">363698007</a> = `<a href="http://snomed.info/id/299701004" title="299701004 | Bone of forearm |" class="external-link" rel="nofollow">299701004</a>` is redundant.
### 8.4 A.4 Conjunction and Disjunction - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Valid Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precoordinated</td>
</tr>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>1276</td>
<td>AND</td>
</tr>
<tr>
<td>&lt; 301867009</td>
<td>Edema of trunk</td>
</tr>
<tr>
<td>1277</td>
<td></td>
</tr>
<tr>
<td>1278</td>
<td></td>
</tr>
<tr>
<td>61233003</td>
<td>Silo-fillers&amp;#39; disease</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung</td>
<td>363358000</td>
</tr>
<tr>
<td>1285</td>
<td>OR</td>
<td>1287</td>
</tr>
<tr>
<td>&lt; 301867009</td>
<td>Edema of trunk</td>
<td>19242006</td>
</tr>
<tr>
<td>1286</td>
<td></td>
<td>1291</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung</td>
<td>100100011</td>
</tr>
<tr>
<td>1292</td>
<td>AND</td>
<td>9102</td>
</tr>
<tr>
<td>^ 700043003</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

1276 http://snomed.info/id/19829001  
1277 http://snomed.info/id/301867009  
1278 http://snomed.info/id/233709006  
1279 http://snomed.info/id/233709006  
1280 http://snomed.info/id/116676008  
1281 http://snomed.info/id/40829002  
1282 http://snomed.info/id/363698007  
1283 http://snomed.info/id/278985004  
1284 http://snomed.info/id/61233003  
1285 http://snomed.info/id/19829001  
1286 http://snomed.info/id/301867009  
1287 http://snomed.info/id/363358000  
1288 http://snomed.info/id/233709006  
1289 http://snomed.info/id/116676008  
1290 http://snomed.info/id/40829002  
1291 http://snomed.info/id/19242006  
1292 http://snomed.info/id/19829001
<table>
<thead>
<tr>
<th>problem list concepts reference set</th>
<th>pulmonary infarction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinical finding</td>
<td>&lt; 404684003</td>
</tr>
<tr>
<td>Finding site $^{1296}$</td>
<td>363698007</td>
</tr>
<tr>
<td>AND</td>
<td>90318009</td>
</tr>
<tr>
<td>Associated morphology $^{1298}$</td>
<td>88015002</td>
</tr>
<tr>
<td>=</td>
<td>116676008</td>
</tr>
<tr>
<td>Stenosis $^{1299}$</td>
<td>56786000</td>
</tr>
<tr>
<td>363698007</td>
<td>Pulmonary valve structure$^{1297}$</td>
</tr>
<tr>
<td>1300</td>
<td>Tetralogy of Fallot$^{1306}$</td>
</tr>
<tr>
<td>39057004</td>
<td>Rheumatic pulmonary valve stenosis$^{1294}$</td>
</tr>
<tr>
<td>1301</td>
<td>415582006</td>
</tr>
<tr>
<td>56786000</td>
<td>363698007</td>
</tr>
<tr>
<td>1302</td>
<td>Finding site$^{1302}$</td>
</tr>
</tbody>
</table>

1293 http://snomed.info/id/700043003
1294 http://snomed.info/id/1001000119102
1295 http://snomed.info/id/404684003
1296 http://snomed.info/id/363698007
1297 http://snomed.info/id/39057004
1298 http://snomed.info/id/116676008
1299 http://snomed.info/id/415582006
1300 http://snomed.info/id/91442002
1301 http://snomed.info/id/56786000
1302 http://snomed.info/id/363698007
1303 http://snomed.info/id/90318009
1304 http://snomed.info/id/116676008
1305 http://snomed.info/id/88015002
1306 http://snomed.info/id/86299006
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Clinical finding</th>
<th>Renal infarct</th>
<th>Sudden cardiac death</th>
</tr>
</thead>
<tbody>
<tr>
<td>404684003</td>
<td>45456005</td>
<td>95281009</td>
</tr>
<tr>
<td>Associated morphology</td>
<td>=</td>
<td>Due to</td>
</tr>
<tr>
<td>116676008</td>
<td>55641003</td>
<td>42752001</td>
</tr>
<tr>
<td>OR</td>
<td>42752001</td>
<td>95281009</td>
</tr>
<tr>
<td>703326006</td>
<td>22298006</td>
<td>Myocardial infarction</td>
</tr>
<tr>
<td>Mitral regurgitation due to acute myocardial infarction</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1307 http://snomed.info/id/404684003
1308 http://snomed.info/id/116676008
1309 http://snomed.info/id/55641003
1310 http://snomed.info/id/42752001
1311 http://snomed.info/id/22298006
1312 http://snomed.info/id/45456005
1313 http://snomed.info/id/95281009
1314 http://snomed.info/id/42752001
1315 http://snomed.info/id/22298006
1316 http://snomed.info/id/703326006
Expression Constraint Language – Expression Constraint Language - Specification and Guide

Appendix A – Examples Of Valid Expressions – 158

85971001 | Rheumatic pulmonary valve stenosis with insufficiency

1317

{ 363698007 | Finding site
    1318  =  << 39057004 | Pulmonary valve structure
    116676008 | Associated morphology
    1319  =  << 415582006 | Stenosis
    1320 } OR
{ 363698007 | Finding site
    1321  =  << 53085002 | Right ventricular structure
    116676008 | Associated morphology
    1322  =  << 56246009 | Hypertrophy
    1323 } OR

86299006 | Tetralogy of Fallot

1332

56786000 | Pulmonic valve stenosis

1327

363698007 | Finding site
    1328  =  90318009 | Structure of anulus fibrosus of pulmonary artery
    116676008 | Associated morphology
    1329
    88015002 | Partial stenosis
    1330

56786000 | Pulmonic valve stenosis

1327

363698007 | Finding site
    1328  =  90318009 | Structure of anulus fibrosus of pulmonary artery
    116676008 | Associated morphology
    1329
    88015002 | Partial stenosis
    1330

85971001 | Rheumatic pulmonary valve stenosis with insufficiency

1326

363698007 | Finding site
    1327  =  < 404684003 | Clinical finding
    1317

{ 363698007 | Finding site
    1318  =  << 39057004 | Pulmonary valve structure
    116676008 | Associated morphology
    1319  =  << 415582006 | Stenosis
    1320 } OR
{ 363698007 | Finding site
    1321  =  << 53085002 | Right ventricular structure
    116676008 | Associated morphology
    1322  =  << 56246009 | Hypertrophy
    1323 } OR

86299006 | Tetralogy of Fallot

1332

56786000 | Pulmonic valve stenosis

1327

363698007 | Finding site
    1328  =  90318009 | Structure of anulus fibrosus of pulmonary artery
    116676008 | Associated morphology
    1329
    88015002 | Partial stenosis
    1330

^ 450990004 | Adverse drug reactions reference set for GP/FP health

294811002 | Corticotrophic hormone allergy

1337

1317 http://snomed.info/id/404684003
1318 http://snomed.info/id/363698007
1319 http://snomed.info/id/39057004
1320 http://snomed.info/id/116676008
1321 http://snomed.info/id/415582006
1322 http://snomed.info/id/363698007
1323 http://snomed.info/id/53085002
1324 http://snomed.info/id/116676008
1325 http://snomed.info/id/56246009
1326 http://snomed.info/id/85971001
1327 http://snomed.info/id/56786000
1328 http://snomed.info/id/363698007
1329 http://snomed.info/id/90318009
1330 http://snomed.info/id/116676008
1331 http://snomed.info/id/88015002
1332 http://snomed.info/id/86299006
1337 http://snomed.info/id/294811002
Where necessary, these examples make some assumptions about the membership of the example reference sets.

### Appendix A – Examples Of Valid Expressions

#### 8.5 A.5 Exclusion and Not Equals - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Valid Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precoordinated</td>
<td>Postcoordinated</td>
</tr>
</tbody>
</table>

1333 http://snomed.info/id/450990004
1334 http://snomed.info/id/246075003
1335 http://snomed.info/id/373873005
1336 http://snomed.info/id/105590001
1338 http://snomed.info/id/293584003
1339 http://snomed.info/id/293585002
1340 http://snomed.info/id/404684003
1341 http://snomed.info/id/116676008
1342 http://snomed.info/id/56208002
1343 http://snomed.info/id/50960005
1344 http://snomed.info/id/64572001
1345 http://snomed.info/id/116676008
1346 http://snomed.info/id/55075001
1347 http://snomed.info/id/363698007
1348 http://snomed.info/id/14374004
1349 http://snomed.info/id/14374004
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt;&lt; 19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>MINUS</td>
<td>301867009</td>
</tr>
<tr>
<td>`372146004</td>
<td>Acute chest syndrome</td>
</tr>
<tr>
<td>`413839001</td>
<td>Chronic lung disease</td>
</tr>
<tr>
<td>`27819004</td>
<td>Pulmonary ossification</td>
</tr>
<tr>
<td>`116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>`18115005</td>
<td>Pathologic calcification</td>
</tr>
<tr>
<td>`363698007</td>
<td>Finding site</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt;&lt; 19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>MINUS</td>
<td>`700043003</td>
</tr>
<tr>
<td>`233613009</td>
<td>Fungal pneumonia</td>
</tr>
<tr>
<td>`27819004</td>
<td>Pulmonary ossification</td>
</tr>
<tr>
<td>`116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>`18115005</td>
<td>Pathologic calcification</td>
</tr>
<tr>
<td>`363698007</td>
<td>Finding site</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>`(&lt;&lt; 56208002</td>
<td>Ulcer</td>
</tr>
<tr>
<td>AND</td>
<td>`50960005</td>
</tr>
<tr>
<td>)</td>
<td>`26036001</td>
</tr>
<tr>
<td>MINUS</td>
<td>`15902003</td>
</tr>
<tr>
<td>`64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>`116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>`55075001</td>
<td>Bleeding ulcer</td>
</tr>
<tr>
<td>`363698007</td>
<td>Finding site</td>
</tr>
</tbody>
</table>

1350 http://snomed.info/id/19829001
1351 http://snomed.info/id/301867009
1352 http://snomed.info/id/372146004
1353 http://snomed.info/id/27819004
1354 http://snomed.info/id/116676008
1355 http://snomed.info/id/18115005
1356 http://snomed.info/id/363698007
1357 http://snomed.info/id/31094006
1358 http://snomed.info/id/116676008
1359 http://snomed.info/id/18115005
1360 http://snomed.info/id/363698007
1361 http://snomed.info/id/27819004
1362 http://snomed.info/id/116676008
1363 http://snomed.info/id/363698007
1364 http://snomed.info/id/700043003
1365 http://snomed.info/id/2233613009
1366 http://snomed.info/id/116676008
1367 http://snomed.info/id/18115005
1368 http://snomed.info/id/363698007
1369 http://snomed.info/id/31094006
1370 http://snomed.info/id/404684003
1371 http://snomed.info/id/116676008
1372 http://snomed.info/id/56208002
1373 http://snomed.info/id/50960005
1374 http://snomed.info/id/116676008
1375 http://snomed.info/id/26036001
1376 http://snomed.info/id/15902003
1377 http://snomed.info/id/64572001
1378 http://snomed.info/id/363698007
1379 http://snomed.info/id/14374004
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td></td>
<td>233613009</td>
</tr>
<tr>
<td></td>
<td>46708007</td>
</tr>
<tr>
<td></td>
<td>64572001</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|                                                          |                                                    | 422897007 | Finding site | 1389 = Vascular structure of stomach | 1390 |}
|                                                          | 404684003 | Clinical finding | 1392 : [0..0] 116676008 | Associated morphology | 1393 = << 26036001 | Obstruction | 1394 |
|                                                          | 233613009 | Fungal pneumonia | 1395 |
|                                                          | 15902003 | Gastric ulcer with hemorrhage | 1401 |
|                                                          | 64572001 | Disease | 1396 : 116676008 | Associated morphology | 1397 = 55075001 | Bleeding ulcer | 1398 |
|                                                          |                                                    | Finding site | 1399 = Structure of lymphatic vessel of oesophagus | 1400 |}

---

1378 http://snomed.info/id/404684003  
1379 http://snomed.info/id/116676008  
1380 http://snomed.info/id/26036001  
1381 http://snomed.info/id/233613009  
1382 http://snomed.info/id/64572001  
1383 http://snomed.info/id/116676008  
1384 http://snomed.info/id/26036001  
1385 http://snomed.info/id/363698007  
1386 http://snomed.info/id/422897007  
1387 http://snomed.info/id/116676008  
1388 http://snomed.info/id/45771005  
1389 http://snomed.info/id/363698007  
1390 http://snomed.info/id/422897007  
1391 http://snomed.info/id/46708007  
1392 http://snomed.info/id/404684003  
1393 http://snomed.info/id/116676008  
1394 http://snomed.info/id/26036001  
1395 http://snomed.info/id/363698007  
1396 http://snomed.info/id/64572001  
1397 http://snomed.info/id/116676008  
1398 http://snomed.info/id/55075001  
1399 http://snomed.info/id/363698007  
1400 http://snomed.info/id/14374004  
1401 http://snomed.info/id/15902003
<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>26036001</td>
<td>Obstruction</td>
</tr>
<tr>
<td>84906002</td>
<td>Local cyanosis</td>
</tr>
<tr>
<td>244815007</td>
<td>Pyloric obstruction</td>
</tr>
<tr>
<td>1402</td>
<td>64572001</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>1405</td>
<td>1407</td>
</tr>
<tr>
<td>244815007</td>
<td>Pyloric obstruction</td>
</tr>
<tr>
<td>1411</td>
<td>64572001</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>1412</td>
<td>1413</td>
</tr>
<tr>
<td>1417</td>
<td>244815007</td>
</tr>
<tr>
<td>1418</td>
<td>64572001</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>1419</td>
<td>1420</td>
</tr>
</tbody>
</table>

Where necessary, these examples make some assumptions about the membership of the example reference sets.

---

1402 http://snomed.info/id/404684003
1403 http://snomed.info/id/116676008
1404 http://snomed.info/id/26036001
1405 http://snomed.info/id/244815007
1406 http://snomed.info/id/64572001
1407 http://snomed.info/id/116676008
1408 http://snomed.info/id/26036001
1409 http://snomed.info/id/363698007
1410 http://snomed.info/id/314600001
1411 http://snomed.info/id/84906002
1412 http://snomed.info/id/404684003
1413 http://snomed.info/id/116676008
1414 http://snomed.info/id/26036001
1415 http://snomed.info/id/116676008
1416 http://snomed.info/id/26036001
1417 http://snomed.info/id/244815007
1418 http://snomed.info/id/64572001
1419 http://snomed.info/id/116676008
1420 http://snomed.info/id/26036001
1421 http://snomed.info/id/363698007
1422 http://snomed.info/id/314600001
## 8.6 A.6 Nested Expression Constraints - Valid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;&lt;&lt; ((^{700043003})</td>
<td>Example problem list concepts reference set(^{[1423]})</td>
<td>394659003</td>
</tr>
<tr>
<td></td>
<td></td>
<td>194828000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>371807002</td>
</tr>
<tr>
<td>(^{&lt;} (^{450973005})</td>
<td>GP/FP health issue reference set(^{[1430]})</td>
<td>140004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>297009</td>
</tr>
<tr>
<td>(&lt; (^{404684003})</td>
<td>Clinical finding(^{[1433]})</td>
<td>204351007</td>
</tr>
<tr>
<td>:</td>
<td>Finding site(^{[1434]})</td>
<td>457652006</td>
</tr>
<tr>
<td>(=&lt; (^{39057004})</td>
<td>Pulmonary valve structure(^{[1435]})</td>
<td></td>
</tr>
<tr>
<td>AND (^{700043003})</td>
<td>Example problem list concepts reference set(^{[1436]})</td>
<td></td>
</tr>
</tbody>
</table>

\(^{1423}\) http://snomed.info/id/700043003
\(^{1424}\) http://snomed.info/id/394659003
\(^{1425}\) http://snomed.info/id/194828000
\(^{1426}\) http://snomed.info/id/255234002
\(^{1427}\) http://snomed.info/id/22298006
\(^{1428}\) http://snomed.info/id/194828000
\(^{1429}\) http://snomed.info/id/22298006
\(^{1430}\) http://snomed.info/id/404684003
\(^{1431}\) http://snomed.info/id/363698007
\(^{1432}\) http://snomed.info/id/39057004
\(^{1433}\) http://snomed.info/id/457652006
(< 404684003 Clinical finding |<1439 363698007 Finding site |<1440 39057004 Pulmonary valve structure |<1441 ) AND (< 64572001 Disease |<1442 116676008 Associated morphology |<1443 = << 415582006 Stenosis |<1444 )

204351007 Fallot’s trilogy |1445
56786000 Pulmonic valve stenosis |1451

19036004 Rheumatic heart valve stenosis |1446 : ( 363698007 Finding site |1447 = 39057004 Pulmonary valve structure |1448 , 116676008 Associated morphology |1449 = 415582006 Stenosis |1450 )

(< < 17636008 Specimen collection |1452 : 424226004 Using device |1453 = << 19923001 Catheter |1454 ) . 363701004 Direct substance |1455

78014005 Urine |1456
87612001 Blood |1457

(< < 404684003 Clinical finding |<1458 OR = << 272379006 Event (event) |1459 ) :
255234002 After |1460 = << 71388002 Procedure (procedure) |1461

235948002 Postoperative acute pancreatitis |1462

64572001 Disease |1463 : ( 370135005 Pathological process |1464 = 441862004 Infectious process |1465 , 255234002 After |1466 = 387713003 Surgical procedure |1467

1439 http://snomed.info/id/404684003
1440 http://snomed.info/id/363698007
1441 http://snomed.info/id/39057004
1442 http://snomed.info/id/64572001
1443 http://snomed.info/id/116676008
1444 http://snomed.info/id/415582006
1445 http://snomed.info/id/204351007
1446 http://snomed.info/id/19036004
1447 http://snomed.info/id/363698007
1448 http://snomed.info/id/39057004
1449 http://snomed.info/id/116676008
1450 http://snomed.info/id/415582006
1451 http://snomed.info/id/56786000
1452 http://snomed.info/id/17636008
1453 http://snomed.info/id/424226004
1454 http://snomed.info/id/19923001
1455 http://snomed.info/id/363701004
1456 http://snomed.info/id/78014005
1457 http://snomed.info/id/87612001
1458 http://snomed.info/id/404684003
1459 http://snomed.info/id/272379006
1460 http://snomed.info/id/255234002
1461 http://snomed.info/id/71388002
1462 http://snomed.info/id/235948002
1463 http://snomed.info/id/64572001
1464 http://snomed.info/id/370135005
1465 http://snomed.info/id/441862004
1466 http://snomed.info/id/255234002
### Appendix A – Examples Of Valid Expressions

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>441795000</td>
<td>Infected seroma after surgical procedure</td>
<td>1470</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
<td>1468 = 112633009</td>
</tr>
<tr>
<td>1467</td>
<td>Surgical wound</td>
<td>1469</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt; 125605004</td>
<td>Fracture of bone</td>
<td>1471</td>
</tr>
<tr>
<td>[0..0]</td>
<td>Concept model attribute</td>
<td>1472</td>
</tr>
<tr>
<td>MINUS</td>
<td>Finding site</td>
<td>1473</td>
</tr>
<tr>
<td>MINUS</td>
<td>Associated morphology</td>
<td>1474</td>
</tr>
<tr>
<td>=</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>125605004</td>
<td>Fracture of bone</td>
<td>1475</td>
</tr>
<tr>
<td>439987009</td>
<td>Open fracture of bone</td>
<td>1481</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>64572001</td>
<td>Disease</td>
<td>1476</td>
</tr>
<tr>
<td>{ 363698007</td>
<td>Finding site</td>
<td>1477</td>
</tr>
<tr>
<td>=</td>
<td>71341001</td>
<td>Bone structure of femur</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
<td>1479</td>
</tr>
<tr>
<td>=</td>
<td>20946005</td>
<td>Fracture, closed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 404684003</td>
<td>Clinical finding</td>
<td>1482</td>
</tr>
<tr>
<td>47429007</td>
<td>Associated with</td>
<td>1483</td>
</tr>
<tr>
<td>=</td>
<td>{ &lt;</td>
<td>404684003</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
<td>1485</td>
</tr>
<tr>
<td>=</td>
<td>&lt; 55641003</td>
<td>Infarct</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>71023004</td>
<td>Pericarditis secondary to acute myocardial infarction</td>
<td>1487</td>
</tr>
<tr>
<td>3238004</td>
<td>Pericarditis (disorder)</td>
<td>1488</td>
</tr>
<tr>
<td>47429007</td>
<td>Associated with</td>
<td>1489</td>
</tr>
<tr>
<td>=</td>
<td>57054005</td>
<td>Acute myocardial infarction</td>
</tr>
</tbody>
</table>

Where necessary, these examples make some assumptions about the membership of the example reference sets.
Appendix B – Examples Of Invalid Expressions

This appendix provides examples of expressions (both precoordinated and postcoordinated) which do not satisfy the given expression constraints from Chapter 6 (see page 64). This list of examples is not intended to be exhaustive, but rather to provide a useful sample to help clarify the meaning of these constraint. Please refer to the SNOMED CT Languages Github repository for a set of text files containing each of these examples.

- B.1 Simple Expression Constraints - Invalid Expressions
- B.2 Refinements - Invalid Expressions
- B.3 Cardinality - Invalid Expressions
- B.4 Conjunction and Disjunction - Invalid Expressions
- B.5 Exclusion and Not Equals - Invalid Expressions
- B.6 Nested Expression Constraints - Invalid Expressions

9.1 B.1 Simple Expression Constraints - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>INVALID Expression</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>404684003</td>
<td>Clinical finding 1492</td>
<td>56265001</td>
<td>Heart disease 1493</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71388002</td>
<td>Procedure 1497</td>
</tr>
<tr>
<td>&lt; 404684003</td>
<td>Clinical finding 1498</td>
<td>404684003</td>
<td>Clinical finding 1499</td>
</tr>
<tr>
<td></td>
<td></td>
<td>71388002</td>
<td>Procedure 1503</td>
</tr>
<tr>
<td>&lt;&lt; 73211009</td>
<td>Diabetes mellitus 1504</td>
<td>71388002</td>
<td>Procedure 1505</td>
</tr>
</tbody>
</table>

---

1491 https://github.com/IHTSDO/SNOMEDCT-Languages
1492 http://snomed.info/id/404684003
1493 http://snomed.info/id/56265001
1494 http://snomed.info/id/404684003
1495 http://snomed.info/id/363698007
1496 http://snomed.info/id/80891009
1497 http://snomed.info/id/71388002
1498 http://snomed.info/id/404684003
1499 http://snomed.info/id/404684003
1500 http://snomed.info/id/404684003
1501 http://snomed.info/id/405813007
1502 http://snomed.info/id/80891009
1503 http://snomed.info/id/71388002
1504 http://snomed.info/id/73211009
1505 http://snomed.info/id/71388002
1506 http://snomed.info/id/404684003
1507 http://snomed.info/id/363698007
Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; ! 404684003</td>
<td>Clinical finding 1510</td>
</tr>
<tr>
<td>362969004</td>
<td>Disorder of endocrine system 1509</td>
</tr>
<tr>
<td>113331007</td>
<td>Structure of endocrine system 1508</td>
</tr>
<tr>
<td>404684003</td>
<td>Clinical finding 1511</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology 1513</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site 1515</td>
</tr>
<tr>
<td>79654002</td>
<td>Edema 1514</td>
</tr>
<tr>
<td>80891009</td>
<td>Heart structure 1516</td>
</tr>
<tr>
<td>233709006</td>
<td>Toxic pulmonary edema 1517</td>
</tr>
<tr>
<td>&gt; 40541001</td>
<td>Acute pulmonary edema 1518</td>
</tr>
<tr>
<td>40541001</td>
<td>Acute pulmonary edema 1519</td>
</tr>
<tr>
<td>246112005</td>
<td>Severity 1521</td>
</tr>
<tr>
<td>24484000</td>
<td>Severe 1522</td>
</tr>
<tr>
<td>233709006</td>
<td>Toxic pulmonary edema 1523</td>
</tr>
<tr>
<td>304527002</td>
<td>Acute asthma 1524</td>
</tr>
<tr>
<td>&gt;&gt; 40541001</td>
<td>Acute pulmonary edema 1525</td>
</tr>
<tr>
<td>233709006</td>
<td>Toxic pulmonary edema 1526</td>
</tr>
<tr>
<td>304527002</td>
<td>Acute asthma 1530</td>
</tr>
<tr>
<td>246112005</td>
<td>Severity 1527</td>
</tr>
<tr>
<td>24484000</td>
<td>Severe 1528</td>
</tr>
<tr>
<td>246112005</td>
<td>Severity 1529</td>
</tr>
<tr>
<td>24484000</td>
<td>Severe 1529</td>
</tr>
</tbody>
</table>

1509 http://snomed.info/id/362969004
1508 http://snomed.info/id/113331007
1510 http://snomed.info/id/404684003
1511 http://snomed.info/id/404684003
1512 http://snomed.info/id/116676008
1513 http://snomed.info/id/79654002
1514 http://snomed.info/id/363698007
1515 http://snomed.info/id/80891009
1516 http://snomed.info/id/233709006
1517 http://snomed.info/id/40541001
1518 http://snomed.info/id/40541001
1519 http://snomed.info/id/40541001
1520 http://snomed.info/id/246112005
1521 http://snomed.info/id/24484000
1522 http://snomed.info/id/233709006
1523 http://snomed.info/id/304527002
1524 http://snomed.info/id/246112005
1525 http://snomed.info/id/24484000
1526 http://snomed.info/id/233709006
1527 http://snomed.info/id/40541001
1528 http://snomed.info/id/246112005
1529 http://snomed.info/id/24484000
1530 http://snomed.info/id/304527002
Expression Constraint Language – Expression Constraint Language - Specification and Guide

### 9.2 B.2 Refinements - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>INVALID Expression</th>
</tr>
</thead>
</table>

Where necessary, these examples make some assumptions about the membership of the example reference sets.

Please note that this makes the assumption that the given expression constraint is executed against a finite set of expressions that has been pre-classified (e.g. in an expression repository), and that after classification there is at least one intermediate expression between this expression and Clinical finding.

Please note that this makes the assumption that the given expression constraint is executed against a finite set of expressions that has been pre-classified (e.g. in an expression repository), and that after classification there is at least one intermediate expression between Acute pulmonary edema and this expression.

---

1531 http://snomed.info/id/40541001
1532 http://snomed.info/id/404684003
1533 http://snomed.info/id/64572001
1534 http://snomed.info/id/263502005
1535 http://snomed.info/id/424124008
1536 http://snomed.info/id/267038008
1537 http://snomed.info/id/6143009
1538 http://snomed.info/id/71388002
1539 http://snomed.info/id/80891009
1540 http://snomed.info/id/75367002
1541 http://snomed.info/id/405813007
1542 http://snomed.info/id/80891009
1543 http://snomed.info/id/404684003
1544 http://snomed.info/id/40541001

Appendix B – Examples Of Invalid Expressions – 168
## Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>79654002</td>
<td>Edema</td>
</tr>
<tr>
<td>73452002</td>
<td>Abscess of lung</td>
</tr>
<tr>
<td>233711002</td>
<td>Oxygen-induced pulmonary edema</td>
</tr>
<tr>
<td>6141006</td>
<td>Retinal edema</td>
</tr>
<tr>
<td>6141006</td>
<td>Retinal edema</td>
</tr>
<tr>
<td>6141006</td>
<td>Retinal edema</td>
</tr>
</tbody>
</table>

---

1545 http://snomed.info/id/19829001
1546 http://snomed.info/id/116676008
1547 http://snomed.info/id/79654002
1548 http://snomed.info/id/19829001
1549 http://snomed.info/id/116676008
1550 http://snomed.info/id/44132006
1551 http://snomed.info/id/19829001
1552 http://snomed.info/id/73452002
1553 http://snomed.info/id/19829001
1554 http://snomed.info/id/116676008
1555 http://snomed.info/id/40829002
1556 http://snomed.info/id/233711002
1557 http://snomed.info/id/19829001
1558 http://snomed.info/id/79654002
1559 http://snomed.info/id/116676008
1560 http://snomed.info/id/6141006
1561 http://snomed.info/id/19829001
1562 http://snomed.info/id/116676008
1563 http://snomed.info/id/103619005
1564 http://snomed.info/id/73452002
1565 http://snomed.info/id/19829001
1566 http://snomed.info/id/116676008
1567 http://snomed.info/id/44132006
1568 http://snomed.info/id/6141006
<table>
<thead>
<tr>
<th>Expression Constraint Language – Expression Constraint Language - Specification and Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Appendix B – Examples Of Invalid Expressions</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>&lt; 404684003</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>404684003</td>
</tr>
<tr>
<td>448643005</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>431238002</td>
</tr>
<tr>
<td>46093004</td>
</tr>
</tbody>
</table>

1569 http://snomed.info/id/404684003
1570 http://snomed.info/id/363698007
1571 http://snomed.info/id/39057004
1572 http://snomed.info/id/116676008
1573 http://snomed.info/id/415582006
1574 http://snomed.info/id/404684003
1575 http://snomed.info/id/448643005
1576 http://snomed.info/id/116676008
1577 http://snomed.info/id/44132006
1578 http://snomed.info/id/448643005
1579 http://snomed.info/id/404684003
1580 http://snomed.info/id/363698007
1581 http://snomed.info/id/81853006
1582 http://snomed.info/id/116676008
1583 http://snomed.info/id/415582006
1584 http://snomed.info/id/341238002
1585 http://snomed.info/id/246075003
1586 http://snomed.info/id/387517004
1587 http://snomed.info/id/46093004
1588 http://snomed.info/id/404684003
1589 http://snomed.info/id/246075003
1590 http://snomed.info/id/372687004
< 404684003 | Clinical finding  
{ 363698007 | Finding site 39057004 | Pulmonary valve structure  
116676008 | Associated morphology  
415582006 | Stenosis }  
{ 363698007 | Finding site 53085002 | Right ventricular structure  
116676008 | Associated morphology  
415582006 | Stenosis }  

404684003 | Clinical finding  
{ 363698007 | Finding site 39057004 | Pulmonary valve structure  
116676008 | Associated morphology  
56246009 | Hypertrophy }  
{ 363698007 | Finding site 53085002 | Right ventricular structure  
116676008 | Associated morphology  
415582006 | Stenosis }

56786000 | Pulmonary valve stenosis

1591 http://snomed.info/id/404684003
1592 http://snomed.info/id/363698007
1593 http://snomed.info/id/39057004
1594 http://snomed.info/id/116676008
1595 http://snomed.info/id/415582006
1596 http://snomed.info/id/363698007
1597 http://snomed.info/id/53085002
1598 http://snomed.info/id/116676008
1599 http://snomed.info/id/56246009
1600 http://snomed.info/id/404684003
1601 http://snomed.info/id/404684003
1602 http://snomed.info/id/363698007
1603 http://snomed.info/id/39057004
1604 http://snomed.info/id/116676008
1605 http://snomed.info/id/56246009
1606 http://snomed.info/id/363698007
1607 http://snomed.info/id/53085002
1608 http://snomed.info/id/116676008
1609 http://snomed.info/id/415582006
1610 http://snomed.info/id/56786000

Appendix B – Examples Of Invalid Expressions – 171
<table>
<thead>
<tr>
<th>expression</th>
<th>concept</th>
<th>code</th>
<th>definition</th>
<th>code</th>
<th>concept</th>
<th>code</th>
<th>definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt; 404684003</td>
<td>Clinical finding</td>
<td>404684003</td>
<td>Mucosal ulcer</td>
<td>95356008</td>
<td>Mucosal ulcer</td>
<td>95356008</td>
<td>Due to</td>
</tr>
</tbody>
</table>

1611 http://snomed.info/id/404684003
1612 http://snomed.info/id/47429007
1613 http://snomed.info/id/267038008
1614 http://snomed.info/id/404684003
1615 http://snomed.info/id/95356008
1616 http://snomed.info/id/42752001
1617 http://snomed.info/id/59901004
### Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 27658006</td>
<td>Amoxicillin</td>
</tr>
<tr>
<td>411116001</td>
<td>Has dose form</td>
</tr>
<tr>
<td>= &lt;&lt;</td>
<td>385055001</td>
</tr>
<tr>
<td>37423002</td>
<td>Amoxicillin trihydrate 125 mg chewable tablet</td>
</tr>
<tr>
<td>269999991</td>
<td>27658006 Amoxicillin</td>
</tr>
<tr>
<td>1627</td>
<td>Has dose form</td>
</tr>
<tr>
<td>411116001</td>
<td>421026006 Oral tablet</td>
</tr>
<tr>
<td>1628</td>
<td>179999999100 Has basis of strength</td>
</tr>
<tr>
<td>1629</td>
<td>219999999102 Amoxicillin only</td>
</tr>
<tr>
<td>1630</td>
<td>189999999103 Has strength magnitude</td>
</tr>
<tr>
<td>1631</td>
<td>199999999101 Has strength unit</td>
</tr>
<tr>
<td>1632</td>
<td>258684004 mg</td>
</tr>
<tr>
<td>1633</td>
<td>374233002 Amoxicillin trihydrate 125 mg chewable tablet</td>
</tr>
<tr>
<td>1634</td>
<td></td>
</tr>
</tbody>
</table>

---

1618 [http://snomed.info/id/27658006](http://snomed.info/id/27658006)
1619 [http://snomed.info/id/411116001](http://snomed.info/id/411116001)
1620 [http://snomed.info/id/385055001](http://snomed.info/id/385055001)
1621 [http://snomed.org/fictid#179999999100](http://snomed.org/fictid#179999999100)
1622 [http://snomed.org/fictid#219999999102](http://snomed.org/fictid#219999999102)
1623 [http://snomed.org/fictid#189999999103](http://snomed.org/fictid#189999999103)
1624 [http://snomed.org/fictid#199999999101](http://snomed.org/fictid#199999999101)
1625 [http://snomed.info/id/258684004](http://snomed.info/id/258684004)
1626 [http://snomed.info/id/269999999100](http://snomed.info/id/269999999100)
1627 [http://snomed.info/id/27658006](http://snomed.info/id/27658006)
1628 [http://snomed.info/id/411116001](http://snomed.info/id/411116001)
1629 [http://snomed.info/id/421026006](http://snomed.info/id/421026006)
1630 [http://snomed.info/id/179999999100](http://snomed.info/id/179999999100)
1631 [http://snomed.info/id/219999999102](http://snomed.info/id/219999999102)
1632 [http://snomed.info/id/189999999103](http://snomed.info/id/189999999103)
1633 [http://snomed.info/id/199999999101](http://snomed.info/id/199999999101)
1634 [http://snomed.info/id/258684004](http://snomed.info/id/258684004)
1635 [http://snomed.info/id/374233002](http://snomed.info/id/374233002)
< 27658006 | Amoxicillin |
1636 :
411116001 | Has dose form |
1637 = <<
385055001 | Tablet dose form |
1638 ,
{ 17999999910 |
0 | Has basis of strength |
1639 = {
21999999910 |
2 | Amoxicillin only |
1640 :
18999999910 | Has strength magnitude |
1641 >= #500,
18999999910 | Has strength magnitude |
1642 <= #800,
19999999910 | Has strength unit |
1643 = 258684004 | mg |
1644 }}
### Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Constraint</th>
<th>Description</th>
</tr>
</thead>
</table>
| `< 373873005 | Pharmaceutical / biologic product | Has trade name | "PANADOL"
| 209999999104 | Pharmaceutical / biologic product | Has trade name | "PANADEINE"
| 373873005 | Pharmaceutical / biologic product | Has active ingredient | Paracetamol or derivative
| 127489000 | Paracetamol or derivative | Has trade name | "PANADEINE"
| 34080009 | Malleus structure | Finding site | "Malleus structure"
| 125605004 | Fracture of bone | Laterality | Left
| 10200004 | Liver structure | Laterality | Right

1654 http://snomed.info/id/374647008
1655 http://snomed.info/id/373873005
1656 http://snomed.info/id/209999999104
1657 http://snomed.info/id/373873005
1658 http://snomed.info/id/373873005
1659 http://snomed.info/id/127489000
1660 http://snomed.info/id/412031009
1661 http://snomed.info/id/209999999104
1662 http://snomed.info/id/322236009
1663 http://snomed.info/id/91723000
1664 http://snomed.info/id/363698009
1665 http://snomed.info/id/125605004
1666 http://snomed.info/id/34080009
1667 http://snomed.info/id/34080009
1668 http://snomed.info/id/272741003
1669 http://snomed.info/id/7771000
1670 http://snomed.info/id/10200004
1671 http://snomed.info/id/10200004
1672 http://snomed.info/id/272741003
1673 http://snomed.info/id/24028007
### Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Invalid Expression</th>
<th>Corrected Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 125605004</td>
<td>Fracture of bone</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>34080009</td>
<td>Malleus structure</td>
</tr>
<tr>
<td>272741003</td>
<td>Laterality</td>
</tr>
<tr>
<td>77710000</td>
<td>Left</td>
</tr>
<tr>
<td>10200004</td>
<td>Liver structure</td>
</tr>
<tr>
<td></td>
<td>10200004</td>
</tr>
<tr>
<td></td>
<td>272741003</td>
</tr>
<tr>
<td></td>
<td>24028007</td>
</tr>
<tr>
<td></td>
<td>`&lt; 105590001</td>
</tr>
<tr>
<td></td>
<td>R &lt;= 127489000</td>
</tr>
<tr>
<td></td>
<td>=</td>
</tr>
<tr>
<td>387517004</td>
<td>Paracetamol</td>
</tr>
<tr>
<td></td>
<td>24999999910</td>
</tr>
<tr>
<td></td>
<td>127489000</td>
</tr>
<tr>
<td></td>
<td>105590001</td>
</tr>
<tr>
<td>373873005</td>
<td>Pharmaceutical / biologic product</td>
</tr>
<tr>
<td></td>
<td>127489000</td>
</tr>
<tr>
<td></td>
<td>372687004</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td></td>
<td>105590001</td>
</tr>
<tr>
<td></td>
<td>373873005</td>
</tr>
<tr>
<td></td>
<td>127489000</td>
</tr>
<tr>
<td></td>
<td>126109000</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td>105590001</td>
<td>Substance</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td>373873005</td>
<td>Pharmaceutical / biologic product</td>
</tr>
<tr>
<td></td>
<td>127489000</td>
</tr>
<tr>
<td></td>
<td>372687004</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
</tr>
</tbody>
</table>

1674 http://snomed.info/id/125605004
1675 http://snomed.info/id/363698007
1676 http://snomed.info/id/34080009
1677 http://snomed.info/id/272741003
1678 http://snomed.info/id/77710000
1679 http://snomed.info/id/10200004
1680 http://snomed.info/id/10200004
1681 http://snomed.info/id/272741003
1682 http://snomed.info/id/24028007
1683 http://snomed.info/id/105590001
1684 http://snomed.info/id/127489000
1685 http://snomed.info/id/77710000
1686 http://snomed.info/id/10200004
1687 http://snomed.info/id/10200004
1688 http://snomed.info/id/272741003
1689 http://snomed.info/id/24028007
1690 http://snomed.info/id/105590001
1691 http://snomed.info/id/127489000
1692 http://snomed.info/id/27658006
1693 http://snomed.info/id/105590001
1694 http://snomed.info/id/373873005
1695 http://snomed.info/id/127489000
1696 http://snomed.info/id/126109000
1697 http://snomed.info/id/105590001
1698 http://snomed.info/id/127489000
1699 http://snomed.info/id/27658006
1700 http://snomed.info/id/105590001
1701 http://snomed.info/id/373873005
1702 http://snomed.info/id/127489000
1703 http://snomed.info/id/126109000
Please note that some of these examples are based on a hypothetical drug concept model.

The SNOMED CT identifiers created with the '9999999' namespace are for example only, and should not be used in a production environment.

9.3 B.3 Cardinality - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>INVALID Expression</th>
</tr>
</thead>
</table>

1698 http://snomed.info/id/387517004
1699 http://snomed.info/id/404684003
1700 http://snomed.info/id/79654002
1701 http://snomed.info/id/263225007
1702 http://snomed.info/id/404684003
1703 http://snomed.info/id/116676008
1704 http://snomed.info/id/72704001
1705 http://snomed.info/id/385933006
1706 http://snomed.info/id/363698007
1707 http://snomed.info/id/80891009
1708 http://snomed.info/id/372687004
1709 http://snomed.info/id/116676008
1710 http://snomed.info/id/363698007
1711 http://snomed.info/id/80891009
1712 http://snomed.info/id/246075003
1713 http://snomed.info/id/372687004
1714 http://snomed.info/id/404684003
1715 http://snomed.info/id/372687004

Appendix B – Examples Of Invalid Expressions – 177
## Appendix B – Examples Of Invalid Expressions

### Precoordinated

<table>
<thead>
<tr>
<th>Expressed as</th>
<th>Description</th>
<th>Specified References</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>1716</td>
</tr>
<tr>
<td>[1..3] 127489000</td>
<td>Has active ingredient</td>
<td>1717</td>
</tr>
<tr>
<td>&lt; 105590001</td>
<td>Substance</td>
<td>1718</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expressed as</th>
<th>Description</th>
<th>Specified References</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>1730</td>
</tr>
<tr>
<td>[1..1] 127489000</td>
<td>Has active ingredient</td>
<td>1731</td>
</tr>
<tr>
<td>&lt; 105590001</td>
<td>Substance</td>
<td>1732</td>
</tr>
</tbody>
</table>

### Postcoordinated

<table>
<thead>
<tr>
<th>Expressed as</th>
<th>Description</th>
<th>Specified References</th>
</tr>
</thead>
<tbody>
<tr>
<td>279999999108</td>
<td>Inert tablet</td>
<td>1719</td>
</tr>
<tr>
<td>437867004</td>
<td>Chlorphenamine + dextromethorphan + paracetamol + pseudoephedrine</td>
<td>1729</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expressed as</th>
<th>Description</th>
<th>Specified References</th>
</tr>
</thead>
<tbody>
<tr>
<td>412556009</td>
<td>Doxylamine</td>
<td>1728</td>
</tr>
</tbody>
</table>

### References

1716 http://snomed.info/id/373873005  
1717 http://snomed.info/id/127489000  
1718 http://snomed.info/id/105590001  
1719 http://snomed.info/fictid#279999999108  
1720 http://snomed.info/id/373873005  
1721 http://snomed.info/id/127489000  
1722 http://snomed.info/id/412031009  
1723 http://snomed.info/id/387494007  
1724 http://snomed.info/id/127489000  
1725 http://snomed.info/id/255641001  
1726 http://snomed.info/id/127489000  
1727 http://snomed.info/id/44068004  
1728 http://snomed.info/id/437867004  
1729 http://snomed.info/id/373873005  
1730 http://snomed.info/id/127489000  
1731 http://snomed.info/id/105590001  
1732 http://snomed.info/fictid#279999999108  
1733 http://snomed.info/id/373873005  
1734 http://snomed.info/id/127489000  
1735 http://snomed.info/id/412031009  
1736 http://snomed.info/id/387494007  
1737 http://snomed.info/id/412556009  
1738 http://snomed.info/id/437867004
### Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
<th>Snomed ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>373873005</td>
</tr>
<tr>
<td>[0..1] 127489000</td>
<td>Has active ingredient</td>
<td>127489000</td>
</tr>
<tr>
<td>`&lt; 105590001</td>
<td>Substance</td>
<td>105590001</td>
</tr>
<tr>
<td>Paracetamol + codeine</td>
<td>412556009</td>
<td></td>
</tr>
<tr>
<td>`&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>373873005</td>
</tr>
<tr>
<td>[1..*] 127489000</td>
<td>Has active ingredient</td>
<td>127489000</td>
</tr>
<tr>
<td>`&lt; 105590001</td>
<td>Substance</td>
<td>105590001</td>
</tr>
<tr>
<td>373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>373873005</td>
</tr>
<tr>
<td>127489000</td>
<td>Has active ingredient</td>
<td>127489000</td>
</tr>
<tr>
<td>412031009</td>
<td>Paracetamol or derivative</td>
<td>412031009</td>
</tr>
<tr>
<td>`&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>373873005</td>
</tr>
<tr>
<td>411116001</td>
<td>Has dose form</td>
<td>411116001</td>
</tr>
<tr>
<td>385055001</td>
<td>Tablet</td>
<td>385055001</td>
</tr>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
<td>404684003</td>
</tr>
<tr>
<td>[1..1] 363698007</td>
<td>Finding site</td>
<td>363698007</td>
</tr>
<tr>
<td>Fracture of radius and ulna</td>
<td>75857000</td>
<td></td>
</tr>
<tr>
<td>`&lt; 40733004</td>
<td>Infectious disease</td>
<td>40733004</td>
</tr>
<tr>
<td>279999999108</td>
<td>Inert tablet</td>
<td>279999999108</td>
</tr>
<tr>
<td>404684003</td>
<td>Clinical finding</td>
<td>404684003</td>
</tr>
<tr>
<td>[1..*] 116676008</td>
<td>Associated morphology</td>
<td>116676008</td>
</tr>
<tr>
<td>72704001</td>
<td>Fracture</td>
<td>72704001</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
<td>363698007</td>
</tr>
<tr>
<td>62413002</td>
<td>Bone structure of radius</td>
<td>62413002</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
<td>363698007</td>
</tr>
<tr>
<td>23416004</td>
<td>Bone structure of ulna</td>
<td>23416004</td>
</tr>
</tbody>
</table>

1740 http://snomed.info/id/373873005
1741 http://snomed.info/id/127489000
1742 http://snomed.info/id/105590001
1743 http://snomed.info/id/412556009
1744 http://snomed.info/id/373873005
1745 http://snomed.info/id/127489000
1746 http://snomed.info/id/412031009
1747 http://snomed.info/id/385055001
1748 http://snomed.info/id/279999999108
1749 http://snomed.info/id/404684003
1750 http://snomed.info/id/363698007
1751 http://snomed.info/id/91723000
1752 http://snomed.info/id/40733004
1753 http://snomed.info/id/373873005
1754 http://snomed.info/id/411116001
1755 http://snomed.info/id/385055001
1756 http://snomed.info/id/404684003
1757 http://snomed.info/id/363698007
1758 http://snomed.info/id/91723000
1759 http://snomed.info/id/75857000
1760 http://snomed.info/id/116676008
1761 http://snomed.info/id/116676008
1762 http://snomed.info/id/72704001
1763 http://snomed.info/id/363698007
1764 http://snomed.info/id/62413002
1765 http://snomed.info/id/363698007
1766 http://snomed.info/id/23416004
1767 http://snomed.info/id/40733004
### Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td><code>2.*</code></td>
<td>Arm fracture</td>
</tr>
<tr>
<td>`&lt; 91723000</td>
<td>Anatomical structure</td>
</tr>
<tr>
<td><code>768</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Fracture of radius and ulna</td>
</tr>
<tr>
<td><code>1768</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Fracture of radius and ulna</td>
</tr>
<tr>
<td><code>1778</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td><code>2.*</code></td>
<td>Infectious disease</td>
</tr>
<tr>
<td>`&lt; 91723000</td>
<td>Anatomical structure</td>
</tr>
<tr>
<td><code>1769</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Infectious disease</td>
</tr>
<tr>
<td><code>1779</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td><code>2.*</code></td>
<td>Infectious disease</td>
</tr>
<tr>
<td>`&lt; 91723000</td>
<td>Anatomical structure</td>
</tr>
<tr>
<td><code>1780</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital septal defect</td>
</tr>
<tr>
<td><code>1781</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital hypertrophy</td>
</tr>
<tr>
<td><code>1782</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital septal defect</td>
</tr>
<tr>
<td><code>1783</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital hypertrophy</td>
</tr>
<tr>
<td><code>1784</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital septal defect</td>
</tr>
<tr>
<td><code>1785</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital hypertrophy</td>
</tr>
<tr>
<td><code>1786</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital septal defect</td>
</tr>
<tr>
<td><code>1787</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital hypertrophy</td>
</tr>
<tr>
<td><code>1788</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital septal defect</td>
</tr>
<tr>
<td><code>1789</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>`{ [2.*]</td>
<td>Congenital hypertrophy</td>
</tr>
<tr>
<td><code>1790</code></td>
<td></td>
</tr>
</tbody>
</table>


---

1768 http://snomed.info/id/404684003
1769 http://snomed.info/id/363698007
1770 http://snomed.info/id/91723000
1771 http://snomed.info/id/23406007
1772 http://snomed.info/id/404684003
1773 http://snomed.info/id/116676008
1774 http://snomed.info/id/72704001
1775 http://snomed.info/id/363698007
1776 http://snomed.info/id/702468001
1777 http://snomed.info/id/363698007
1778 http://snomed.info/id/75857000
1779 http://snomed.info/id/64572001
1780 http://snomed.info/id/116676008
1781 http://snomed.info/id/75857000
1782 http://snomed.info/id/64572001
1783 http://snomed.info/id/116676008
1784 http://snomed.info/id/363698007
1785 http://snomed.info/id/363698007
1786 http://snomed.info/id/113262008
1787 http://snomed.info/id/116676008
1788 http://snomed.info/id/90141005
1789 http://snomed.info/id/363698007
1790 http://snomed.info/id/244384009
Appendix B – Examples Of Invalid Expressions

< 373873005 | Pharmaceutical / biologic product | Has active ingredient | 1791 : [1..3] { [1..7] 127489000 | Has active ingredient | 1792 = < 105590001 | Substance | 1793 }

279999999108 | Inert tablet | 1784

437867004 | Chlorphenamine + dextromethorphan + paracetamol + pseudoephedrine | 1804

373873005 | Pharmaceutical / biologic product | Has active ingredient | 1795 : { 127489000 | Has active ingredient | 1796 = 412031009 | Paracetamol or derivative | 1797 }, { 127489000 | Has active ingredient | 1798 = 387494007 | Codeine | 1799 }, { 127489000 | Has active ingredient | 1800 = 255641001 | Caffeine | 1801 }, { 127489000 | Has active ingredient | 1802 = 44068004 | Doxylamine | 1803 }

< 373873005 | Pharmaceutical / biologic product | Has active ingredient | 1805 : [0..1] { 127489000 | Has active ingredient | 1806 = < 105590001 | Substance | 1807 }

412556009 | Paracetamol + codeine | 1808

373873005 | Pharmaceutical / biologic product | Has active ingredient | 1809 : { 127489000 | Has active ingredient | 1810 = 412031009 | Paracetamol or derivative | 1811 }, { 127489000 | Has active ingredient | 1812 = 387494007 | Codeine | 1813 }

1791 http://snomed.info/id/373873005
1792 http://snomed.info/id/127489000
1793 http://snomed.info/id/105590001
1794 http://snomed.org/fictid#279999999108
1795 http://snomed.info/id/373873005
1796 http://snomed.info/id/127489000
1797 http://snomed.info/id/412031009
1798 http://snomed.info/id/127489000
1799 http://snomed.info/id/387494007
1800 http://snomed.info/id/127489000
1801 http://snomed.info/id/255641001
1802 http://snomed.info/id/127489000
1803 http://snomed.info/id/44068004
1804 http://snomed.info/id/437867004
1805 http://snomed.info/id/373873005
1806 http://snomed.info/id/127489000
1807 http://snomed.info/id/105590001
1808 http://snomed.info/id/412556009
1809 http://snomed.info/id/373873005
1810 http://snomed.info/id/127489000
1811 http://snomed.info/id/412031009
1812 http://snomed.info/id/127489000
1813 http://snomed.info/id/387494007
<table>
<thead>
<tr>
<th>Expression</th>
<th>Constraint</th>
<th>Language</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>`&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>1814 :</td>
<td>[1..*] { 127489000</td>
<td>Has active ingredient</td>
</tr>
<tr>
<td>`&lt; 373873005</td>
<td>Pharmaceutical / biologic product</td>
<td>1818 :</td>
<td>411116001</td>
<td>Has dose form</td>
</tr>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
<td>1821 :</td>
<td>{ 363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
<td>1825 :</td>
<td>{ 116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>`&lt; 404684003</td>
<td>Clinical finding</td>
<td>1835 :</td>
<td>{ 0..0</td>
<td>{ 2..*} 363698007</td>
</tr>
<tr>
<td>`&lt; 64572001</td>
<td>Disease</td>
<td>1838 :</td>
<td>{ 116676008</td>
<td>Associated morphology</td>
</tr>
</tbody>
</table>

1814 http://snomed.info/id/373873005
1815 http://snomed.info/id/127489000
1816 http://snomed.info/id/105590001
1817 http://snomed.org/fictid#279999999108
1818 http://snomed.info/id/373873005
1819 http://snomed.info/id/411116001
1820 http://snomed.info/id/385055001
1821 http://snomed.info/id/404684003
1822 http://snomed.info/id/363698007
1823 http://snomed.info/id/91723000
1824 http://snomed.info/id/75857000
1825 http://snomed.info/id/404684003
1826 http://snomed.info/id/116676008
1827 http://snomed.info/id/72704001
1828 http://snomed.info/id/363698007
1829 http://snomed.info/id/62413002
1830 http://snomed.info/id/116676008
1831 http://snomed.info/id/72704001
1832 http://snomed.info/id/363698007
1833 http://snomed.info/id/23416004
1834 http://snomed.info/id/40733004
1835 http://snomed.info/id/404684003
1836 http://snomed.info/id/363698007
1837 http://snomed.info/id/91723000
1838 http://snomed.info/id/64572001
1839 http://snomed.info/id/116676008
### 9.4 B.4 Conjunction and Disjunction - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>INVALID Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Precoordinated</strong></td>
<td></td>
</tr>
<tr>
<td>&lt; 19829001</td>
<td>Disorder of lung AND Edema of trunk</td>
</tr>
<tr>
<td>&lt; 301867009</td>
<td></td>
</tr>
<tr>
<td>73452002</td>
<td>Abscess of lung AND Abdominal wall edema</td>
</tr>
<tr>
<td>248508001</td>
<td>Abdominal wall edema</td>
</tr>
<tr>
<td><strong>Postcoordinated</strong></td>
<td></td>
</tr>
<tr>
<td>248508001</td>
<td>Abdominal wall edema</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>40829002</td>
<td>Acute edema</td>
</tr>
</tbody>
</table>

---

1840 http://snomed.info/id/396351009
1841 http://snomed.info/id/363698007
1842 http://snomed.info/id/25943004
1843 http://snomed.info/id/363698007
1844 http://snomed.info/id/113262008
1845 http://snomed.info/id/116676008
1846 http://snomed.info/id/90141005
1847 http://snomed.info/id/363698007
1848 http://snomed.info/id/244384009
1849 http://snomed.info/id/19829001
1850 http://snomed.info/id/301867009
1851 http://snomed.info/id/73452002
1852 http://snomed.info/id/248508001
1853 http://snomed.info/id/116676008
1854 http://snomed.info/id/40829002
1855 http://snomed.info/id/248508001
Appendix B – Examples Of Invalid Expressions

< 19829001 | Disorder of lung | 1856 OR < 301867009 | Edema of trunk | 1857

19829001 | Disorder of lung | 1858
301867009 | Edema of trunk | 1862
128121009 | Disorder of trunk | 1863
128121009 | Disorder of trunk | 1859
116676008 | Associated morphology | 1860 = Abscess | 1861
301867009 | Edema of trunk | 1862
128121009 | Disorder of trunk | 1863
116676008 | Associated morphology | 1868 = Abscess | 1869

< 19829001 | Disorder of lung | 1864 AND ^ 700043003 | Example problem list concepts reference set | 1865

73452002 | Abscess of lung | 1866
19829001 | Disorder of lung | 1867
116676008 | Associated morphology | 1868 = Abscess | 1869

< 404684003 | Clinical finding | 1870 : 363698007 | Finding site | 1871 = << 39057004 | Pulmonary valve structure | 1872

116676008 | Associated morphology | 1873 = << 415582006 | Stenosis | 1874
301104003 | Pulmonary valve finding | 1875
60573004 | Aortic valve stenosis | 1879
404684003 | Clinical finding | 1876
116676008 | Associated morphology | 1877 = Partial stenosis | 1878
301104003 | Pulmonary valve finding | 1875
60573004 | Aortic valve stenosis | 1879
404684003 | Clinical finding | 1876
116676008 | Associated morphology | 1877 = Partial stenosis | 1878

< 404684003 | Clinical finding | 1880 : 116676008 | Associated morphology | 1880 = << 55641003 | Infarct | 1882 OR

368009 | Heart valve disorder | 1885

95281009 | Sudden cardiac death | 1886 : Due to | 1887 =

1856 http://snomed.info/id/19829001
1857 http://snomed.info/id/301867009
1858 http://snomed.info/id/19829001
1859 http://snomed.info/id/128121009
1860 http://snomed.info/id/116676008
1861 http://snomed.info/id/44132006
1862 http://snomed.info/id/301867009
1863 http://snomed.info/id/128121009
1864 http://snomed.info/id/19829001
1865 http://snomed.info/id/700043003
1866 http://snomed.info/id/73452002
1867 http://snomed.info/id/19829001
1868 http://snomed.info/id/116676008
1869 http://snomed.info/id/44132006
1870 http://snomed.info/id/404684003
1871 http://snomed.info/id/363698007
1872 http://snomed.info/id/39057004
1873 http://snomed.info/id/116676008
1874 http://snomed.info/id/415582006
1875 http://snomed.info/id/301104003
1876 http://snomed.info/id/404684003
1877 http://snomed.info/id/116676008
1878 http://snomed.info/id/88015002
1879 http://snomed.info/id/60573004
1880 http://snomed.info/id/404684003
1881 http://snomed.info/id/116676008
1882 http://snomed.info/id/55641003
1883 http://snomed.info/id/368009
1884 http://snomed.info/id/95281009
1885 http://snomed.info/id/42752001
1886 http://snomed.info/id/42752001
<table>
<thead>
<tr>
<th>Expr.</th>
<th>Due to 42752001</th>
<th>Myocardial infarction 1884</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expr.</td>
<td>Cardiac abnormality due to heart abscess 1889</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Acute congestive heart failure 1888</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Clinical finding 404684003</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Myocardial infarction 461089003</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Cardiac abnormality due to heart abscess 1889</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Acute congestive heart failure 1888</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Clinical finding 404684003</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Myocardial infarction 461089003</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Cardiac abnormality due to heart abscess 1889</td>
<td></td>
</tr>
<tr>
<td>Expr.</td>
<td>Acute congestive heart failure 1888</td>
<td></td>
</tr>
</tbody>
</table>

Appendix B – Examples Of Invalid Expressions – 185
10629471000119106 | Allergic rhinitis caused by mould

< 404684003 | Clinical finding
116676008 | Associated morphology

196652006 | Acute duodenal ulcer

64572001 | Disease
116676008 | Associated morphology

10629471000119106 | Allergic rhinitis caused by mould

Where necessary, these examples make some assumptions about the membership of the example reference sets.

9.5 B.5 Exclusion and Not Equals - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>INVALID Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precoordinated</td>
<td>Postcoordinated</td>
</tr>
<tr>
<td>&lt; 19829001 MINUS</td>
<td>19829001</td>
</tr>
<tr>
<td>Disorder of lung</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>1925</td>
<td>1925</td>
</tr>
<tr>
<td>Acute gastrointestinal hemorrhage</td>
<td>Acute gastrointestinal hemorrhage</td>
</tr>
<tr>
<td>1927</td>
<td>1928</td>
</tr>
<tr>
<td>Edema of trunk</td>
<td>Acute edema</td>
</tr>
<tr>
<td>1926</td>
<td>1930</td>
</tr>
<tr>
<td>Pulmonary edema</td>
<td>Finding site</td>
</tr>
<tr>
<td>1933</td>
<td>1931</td>
</tr>
</tbody>
</table>
Appendix B – Examples Of Invalid Expressions – 187

Expression Constraint Language – Expression Constraint Language - Specification and Guide

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;= 19829001</td>
<td>Disorder of lung</td>
</tr>
<tr>
<td>MINUS</td>
<td></td>
</tr>
<tr>
<td>^ 700043003</td>
<td>Example problem list concepts reference set</td>
</tr>
<tr>
<td>67599009</td>
<td>Pulmonary congestion</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>3341006</td>
<td>Right lung structure</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 404684003</td>
<td>Clinical finding</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>397825006</td>
<td>Gastric ulcer</td>
</tr>
<tr>
<td>235670001</td>
<td>Gastric stomal obstruction</td>
</tr>
<tr>
<td>81060008</td>
<td>Intestinal obstruction</td>
</tr>
<tr>
<td>56265001</td>
<td>Heart disease</td>
</tr>
<tr>
<td>64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>26036001</td>
<td>Obstruction</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>422897007</td>
<td>Vascular structure of stomach</td>
</tr>
<tr>
<td>56265001</td>
<td>Heart disease</td>
</tr>
<tr>
<td>116676008</td>
<td>Associated morphology</td>
</tr>
<tr>
<td>26036001</td>
<td>Obstruction</td>
</tr>
<tr>
<td>363698007</td>
<td>Finding site</td>
</tr>
<tr>
<td>422897007</td>
<td>Vascular structure of stomach</td>
</tr>
</tbody>
</table>

References:
- 1934 http://snomed.info/id/19829001
- 1935 http://snomed.info/id/700043003
- 1936 http://snomed.info/id/67599009
- 1937 http://snomed.info/id/363698007
- 1938 http://snomed.info/id/3341006
- 1939 http://snomed.info/id/116676008
- 1940 http://snomed.info/id/404684003
- 1941 http://snomed.info/id/56208002
- 1942 http://snomed.info/id/50960005
- 1943 http://snomed.info/id/26036001
- 1944 http://snomed.info/id/397825006
- 1945 http://snomed.info/id/64572001
- 1946 http://snomed.info/id/116676008
- 1947 http://snomed.info/id/26036001
- 1948 http://snomed.info/id/363698007
- 1949 http://snomed.info/id/422897007
- 1950 http://snomed.info/id/56265001

Appendix B – Examples Of Invalid Expressions – 187
< 404684003 | Clinical finding$^{1960}$ | [0..0] 116676008 | Associated morphology$^{1961}$ = << 26036001 | Obstruction$^{1962}$

81060008 | Intestinal obstruction$^{1963}$

234059001 | Venous stenosis$^{1973}$

< 404684003 | Clinical finding$^{1974}$ | [0..0] 116676008 | Associated morphology$^{1975}$ != << 26036001 | Obstruction$^{1976}$

196652006 | Acute duodenal ulcer$^{1977}$

64572001 | Disease$^{1964}$ : 116676008 | Associated morphology$^{1965}$ = 26036001 | Obstruction$^{1966}$ 363698007 | Finding site$^{1967}$ = 422897007 | Vascular structure of stomach$^{1968}$

{ 116676008 | Associated morphology$^{1969}$ = 45771005 | Acute bleeding ulcer$^{1970}$

363698007 | Finding site$^{1971}$ = 422897007 | Vascular structure of stomach$^{1972}$

64572001 | Disease$^{1978}$ : 116676008 | Associated morphology$^{1979}$ = 26036001 | Obstruction$^{1980}$ 363698007 | Finding site$^{1981}$ = 422897007 | Vascular structure of stomach$^{1982}$

{ 116676008 | Associated morphology$^{1983}$ = 45771005 | Acute bleeding ulcer$^{1984}$

363698007 | Finding site$^{1985}$ = 422897007 | Vascular structure of stomach$^{1986}$

1960 http://snomed.info/id/404684003
1961 http://snomed.info/id/116676008
1962 http://snomed.info/id/26036001
1963 http://snomed.info/id/81060008
1964 http://snomed.info/id/64572001
1965 http://snomed.info/id/116676008
1966 http://snomed.info/id/26036001
1967 http://snomed.info/id/363698007
1968 http://snomed.info/id/422897007
1969 http://snomed.info/id/116676008
1970 http://snomed.info/id/45771005
1971 http://snomed.info/id/363698007
1972 http://snomed.info/id/422897007
1973 http://snomed.info/id/234059001
1974 http://snomed.info/id/404684003
1975 http://snomed.info/id/116676008
1976 http://snomed.info/id/26036001
1977 http://snomed.info/id/196652006
1978 http://snomed.info/id/64572001
1979 http://snomed.info/id/116676008
1980 http://snomed.info/id/26036001
1981 http://snomed.info/id/363698007
1982 http://snomed.info/id/422897007
1983 http://snomed.info/id/116676008
1984 http://snomed.info/id/45771005
1985 http://snomed.info/id/363698007
1986 http://snomed.info/id/422897007
### 8377001 | Hernia, with obstruction | 1987

\< 404684003 | Clinical finding: [0..0] 116676008 | Associated morphology: \< 26036001 | Obstruction | AND | \< [1..*] 116676008 | Associated morphology: \< 26036001 | Obstruction | 1992

### 196652006 | Acute duodenal ulcer | 1993


### 8377001 | Hernia, with obstruction | 2003

\56265001 | Heart disease | 2009


---

1987 http://snomed.info/id/8377001
1988 http://snomed.info/id/404684003
1989 http://snomed.info/id/116676008
1990 http://snomed.info/id/26036001
1991 http://snomed.info/id/116676008
1992 http://snomed.info/id/26036001
1993 http://snomed.info/id/196652006
1994 http://snomed.info/id/64572001
1995 http://snomed.info/id/116676008
1996 http://snomed.info/id/26036001
1997 http://snomed.info/id/363698007
1998 http://snomed.info/id/422897007
1999 http://snomed.info/id/56265001
2000 http://snomed.info/id/45771005
2001 http://snomed.info/id/363698007
2002 http://snomed.info/id/422897007
2003 http://snomed.info/id/8377001
2004 http://snomed.info/id/64572001
2005 http://snomed.info/id/116676008
2006 http://snomed.info/id/45771005
2007 http://snomed.info/id/363698007
2008 http://snomed.info/id/422897007
2009 http://snomed.info/id/56265001

Appendix B – Examples Of Invalid Expressions – 189
## 9.6 B.6 Nested Expression Constraints - Invalid Expressions

<table>
<thead>
<tr>
<th>Expression Constraint</th>
<th>Valid Expression</th>
<th>Precoordinated</th>
<th>Postcoordinated</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt; (^ 700043003</td>
<td>Example problem list concepts reference set(2010)</td>
<td>6143009</td>
<td>Diabetic education(^ {2011})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75367002</td>
<td>Blood pressure(^ {2015})</td>
</tr>
<tr>
<td>^ (&lt; 450973005</td>
<td>GP/FP health issue reference set(2016)</td>
<td>80146002</td>
<td>Appendectomy(^ {2017})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>305342007</td>
<td>Admission to ward(^ {2018})</td>
</tr>
<tr>
<td>(&lt; 404684003</td>
<td>Clinical finding(^ {2019})</td>
<td>125605004</td>
<td>Fracture of bone(^ {2023})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>195967001</td>
<td>Asthma(^ {2027})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

2010 http://snomed.info/id/700043003  
2011 http://snomed.info/id/6143009  
2012 http://snomed.info/id/71388002  
2013 http://snomed.info/id/405813007  
2014 http://snomed.info/id/80891009  
2015 http://snomed.info/id/450973005  
2016 http://snomed.info/id/39057004  
2017 http://snomed.info/id/17401000  
2018 http://snomed.info/id/363698007  
2019 http://snomed.info/id/404684003  
2020 http://snomed.info/id/363698007  
2021 http://snomed.info/id/700043003  
2022 http://snomed.info/id/125605004  
2023 http://snomed.info/id/404684003  
2024 http://snomed.info/id/363698007  
2025 http://snomed.info/id/17401000  
2026 http://snomed.info/id/195967001
### Appendix B – Examples Of Invalid Expressions

- **Expression:**
  \[
  (< 404684003|\text{Clinical finding}|2028 : 363698007|\text{Finding site}|2029 = << 39057004|\text{Pulmonary valve structure}|2030 )
  \]
  \[
  \text{AND} ( < 64572001|\text{Disease}|2031 : 116676008|\text{Associated morphology}|2032 = << 415582006|\text{Stenosis}|2033 )
  \]

  - **Result:**
    - 76107001 | **Spinal stenosis** | 2038

  - **Expression:**
    \[
    (<< 17636008|\text{Specimen collection}|2042 : 424226004|\text{Using device}|2043 = << 19923001|\text{Catheter}|2044) \]
    \[
    363701004 | \text{Direct substance} | 2045
    \]
    \[
    \text{OR} (<< 17636008|\text{Specimen collection}|2042 : 424226004|\text{Using device}|2043 = << 19923001|\text{Catheter}|2044)
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2051 \text{ OR} << 272379006|\text{Event (event)}|2052 )
    \]
    \[
    255234002 | \text{After} | 2053 = << 71388002|\text{Procedure (procedure)}|2054
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2051 : 272379006|\text{Event (event)}|2052 = << 71388002|\text{Procedure (procedure)}|2054
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2051 : 363698007|\text{Finding site}|2029 = << 39057004|\text{Pulmonary valve structure}|2030 )
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2028 : 363698007|\text{Finding site}|2029 = << 39057004|\text{Pulmonary valve structure}|2030 )
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2028 : 363698007|\text{Finding site}|2029 = << 39057004|\text{Pulmonary valve structure}|2030 )
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055

  - **Expression:**
    \[
    (< 404684003|\text{Clinical finding}|2028 : 363698007|\text{Finding site}|2029 = << 39057004|\text{Pulmonary valve structure}|2030 )
    \]

  - **Result:**
    - 293690005 | **Peppermint oil allergy** | 2055
## Appendix B – Examples Of Invalid Expressions

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>82510005</td>
<td>Posttraumatic vertigo</td>
</tr>
<tr>
<td>704333004</td>
<td>Pathological fracture of hand due to osteoporosis</td>
</tr>
<tr>
<td>722571004</td>
<td>Linear fracture of skull due to birth trauma</td>
</tr>
<tr>
<td>3238004</td>
<td>Pericarditis</td>
</tr>
<tr>
<td>64572001</td>
<td>Disease</td>
</tr>
</tbody>
</table>

Where necessary, these examples make some assumptions about the membership of the example reference sets.
10 Appendix C - Dialect Aliases

This appendix provides a list of example aliases that may be used to specify a particular dialect in an ECL filter constraint. Please refer to the 'Dialect Filter' section on 6.8 Description Filters for more information on how these dialect aliases are used in ECL.

All dialect aliases should follow the ABNF syntax shown below. This format is designed to be compatible with BCP-47 (Internet Best Current Practice Specification), which ensures alignment with a range of other specifications - e.g. HTTP "accept-language" headers, and the HL7 FHIR "designation.language" data element.

\[
\text{dialectAlias} = ( \text{language} [\text{"-" script}] [\text{"-" region}] [\text{"-" privateuse}]) / \text{privateuse} \\
\text{language} = \text{alpha alpha} \quad ; \text{ISO 639-1 code (List of codes)} \\
\text{script} = \text{alpha alpha alpha alpha} \quad ; \text{ISO 15924 code (List of codes)} \\
\text{region} = \text{alpha alpha} \quad ; \text{ISO 3166-1 code (List of codes)} \\
\text{privateuse} = "x" 1*("." 1*8(alpha / digit)) \quad ; \text{the clinical scope or context of use}
\]

The table below lists the valid 'dialect' filter values and their equivalent 'dialectId' filter values, for a selection of known language reference sets. To request the addition of a new dialect alias, please use the 'Feedback' button on the bottom of this page.

<table>
<thead>
<tr>
<th>dialect</th>
<th>dialectId</th>
</tr>
</thead>
<tbody>
<tr>
<td>da-dk</td>
<td>554461000005103</td>
</tr>
<tr>
<td>en-au</td>
<td>32570271000036106</td>
</tr>
<tr>
<td>en-ca</td>
<td>19491000087109</td>
</tr>
<tr>
<td>en-gb</td>
<td>90000000000508004</td>
</tr>
<tr>
<td>en-ie</td>
<td>21000220103</td>
</tr>
<tr>
<td>en-nz</td>
<td>271000210107</td>
</tr>
<tr>
<td>en-nz-x-pat</td>
<td>281000210109</td>
</tr>
<tr>
<td>en-us</td>
<td>90000000000509007</td>
</tr>
<tr>
<td>en-x-gmdn</td>
<td>608771002</td>
</tr>
</tbody>
</table>

---

2080 https://www.rfc-editor.org/rfc/rfc5646.html
<table>
<thead>
<tr>
<th>dialect</th>
<th>dialectId</th>
</tr>
</thead>
<tbody>
<tr>
<td>en-x-nhs-clinical</td>
<td>999001261000000100</td>
</tr>
<tr>
<td>en-x-nhs-dmd</td>
<td>999000671000001103</td>
</tr>
<tr>
<td>en-x-nhs-pharmacy</td>
<td>999000691000001104</td>
</tr>
<tr>
<td>en-gb-x-drug</td>
<td>999000681000001101</td>
</tr>
<tr>
<td>en-gb-x-ext</td>
<td>999001251000000103</td>
</tr>
<tr>
<td>es</td>
<td>450828004</td>
</tr>
<tr>
<td>es-uy</td>
<td>5641000179103</td>
</tr>
<tr>
<td>et-ee</td>
<td>71000181105</td>
</tr>
<tr>
<td>de</td>
<td>722130004</td>
</tr>
<tr>
<td>fr</td>
<td>722131000</td>
</tr>
<tr>
<td>fr-be</td>
<td>21000172104</td>
</tr>
<tr>
<td>fr-ca</td>
<td>20581000087109</td>
</tr>
<tr>
<td>ja</td>
<td>722129009</td>
</tr>
<tr>
<td>mi</td>
<td>291000210106</td>
</tr>
<tr>
<td>nl-be</td>
<td>31000172101</td>
</tr>
<tr>
<td>nl-nl</td>
<td>31000146106</td>
</tr>
<tr>
<td>nb-no</td>
<td>61000202103</td>
</tr>
<tr>
<td>dialect</td>
<td>dialectId</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>nn-no</td>
<td>91000202106 [Norwegian Nynorsk language reference set]</td>
</tr>
<tr>
<td>sv-se</td>
<td>46011000052107 [Swedish language reference set]</td>
</tr>
<tr>
<td>zh</td>
<td>722128001 [Chinese language reference set]</td>
</tr>
</tbody>
</table>
11 Appendix D - ECL Quick Reference

This section provides a quick reference to the key syntax features of the Expression Constraint Language.

11.1 Syntax Overview

The following table summarises the key symbols used in the Expression Constraint Language's brief syntax, with the ECL version in which each symbol was introduced. For more information about the version history of ECL, please refer to the 'History' section in 1. Introduction (see page 8).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name</th>
<th>Version</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pipe</td>
<td>1.0</td>
<td>Used on either side of a concept’s term for human readability</td>
</tr>
<tr>
<td>*</td>
<td>Any</td>
<td>1.0</td>
<td>Retrieves all concepts in the substrate</td>
</tr>
<tr>
<td>^</td>
<td>Member of</td>
<td>1.0</td>
<td>Retrieves the referencedComponentId of all (active) members of a reference set (or set of reference sets)</td>
</tr>
<tr>
<td>^ [ A, B]</td>
<td>Member of (with field selection)</td>
<td>2.0</td>
<td>Retrieves the values of fields A and B of all (active) members of a reference set (or set of reference sets) that match the included Member filters (if applicable)</td>
</tr>
<tr>
<td>&lt;</td>
<td>Descendant of</td>
<td>1.0</td>
<td>Retrieves all descendants (subtypes) of the specified concept excluding the concept itself</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Descendant or self of</td>
<td>1.0</td>
<td>Retrieves all descendants (subtypes) of the specified concept including the concept itself</td>
</tr>
<tr>
<td>&lt;!</td>
<td>Child of</td>
<td>1.1</td>
<td>Retrieves all children (immediate subtypes) of the specified concept excluding the concept itself</td>
</tr>
<tr>
<td>&lt;&lt;=</td>
<td>Child or self of</td>
<td>1.4</td>
<td>Retrieves all children (immediate subtypes) of the specified concept including the concept itself</td>
</tr>
<tr>
<td>&gt;</td>
<td>Ancestor of</td>
<td>1.0</td>
<td>Retrieves all ancestors (supertypes) of the specified concept excluding the concept itself</td>
</tr>
<tr>
<td>Symbol</td>
<td>Name</td>
<td>Version</td>
<td>Notes</td>
</tr>
<tr>
<td>--------</td>
<td>---------------------------</td>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>&gt;&gt;</td>
<td>Ancestor or self of</td>
<td>1.0</td>
<td>Retrieves all ancestors (supertypes) of the specified concept including the concept itself</td>
</tr>
<tr>
<td>&gt;!</td>
<td>Parent of</td>
<td>1.1</td>
<td>Retrieves all parents (immediate supertypes) of the specified concept excluding the concept itself</td>
</tr>
<tr>
<td>&gt;&gt;!</td>
<td>Parent or self of</td>
<td>1.4</td>
<td>Retrieves all parents (immediate supertypes) of the specified concept including the concept itself</td>
</tr>
<tr>
<td>!!&gt;</td>
<td>Top of set</td>
<td>2.2</td>
<td>Filters the results set, by matching only on concepts that have no ancestors within the set</td>
</tr>
<tr>
<td>!!&lt;</td>
<td>Bottom of set</td>
<td>2.2</td>
<td>Filters the results set, by matching only on concepts that have no descendants within the set</td>
</tr>
<tr>
<td>A#B</td>
<td>Alternate identifier</td>
<td>2.2</td>
<td>Retrieves a single concept based on an alternate identifier, where A is the identifier scheme alias and B is the identifier code</td>
</tr>
<tr>
<td>AND</td>
<td>Conjunction</td>
<td>1.0</td>
<td>Retrieves the intersection of the results of each sub-expressions</td>
</tr>
<tr>
<td>OR</td>
<td>Disjunction</td>
<td>1.0</td>
<td>Retrieves the union of the results of each sub-expressions</td>
</tr>
<tr>
<td>MINUS</td>
<td>Exclusion</td>
<td>1.0</td>
<td>Retrieves the members of the first expression and excludes the members returned by the second expression</td>
</tr>
<tr>
<td>:</td>
<td>Refinement</td>
<td>1.0</td>
<td>Used before one or more attribute-value pairs to refine the set of concepts retrieved</td>
</tr>
<tr>
<td>[1..3]</td>
<td>Cardinality</td>
<td>1.0</td>
<td>Used to indicate the minimum and maximum number of occurrences of attributes or relationship groups</td>
</tr>
</tbody>
</table>
### 11.2 Examples

The following table provides some examples of each of the key syntax features of the Expression Constraint Language.

**Notes:**

1. In the table above:
   - `'id'` represents a single SNOMED CT concept identifier,
   - `'term'` represents a term associated with the concept identified by `'id'`,
   - `'x'`, `'y'` and `'v'` each represent either a single concept or a set of concepts defined using an expression constraint,
• 'z' represents either a single concept or a set of concepts that are a subtype of \texttt{900000000000455006} \texttt{| Reference set}\textsuperscript{2084},

• 'a' and 'b' each represent either a single concept or a set of concepts that are a subtype of \texttt{410662002} \texttt{| Concept model attribute}\textsuperscript{2085}, and

• 'min' and 'max' are two numeric values that represent the minimum and maximum cardinality allowed.

2. The default substrate, to which expression constraints are applied, includes all concepts, active relationships, active descriptions and active reference set members of a chosen SNOMED CT versioned edition.

### Simple expression constraints

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Evaluation Notes</th>
<th>Example</th>
<th>Example Expansion Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>**id</td>
<td>term</td>
<td>**</td>
<td>Only the concept with the identifier 'id'</td>
</tr>
<tr>
<td>**</td>
<td>**</td>
<td>**</td>
<td>Any concept in the given substrate</td>
</tr>
<tr>
<td>^z</td>
<td>The set of concepts which are members of the reference sets in z</td>
<td>128477000 \texttt{</td>
<td>Abscess</td>
</tr>
<tr>
<td>\texttt{^}723264001 \texttt{</td>
<td>Lateralizable body structure reference set</td>
<td>}\textsuperscript{2088}</td>
<td>181216001 \texttt{</td>
</tr>
<tr>
<td>65784005 \texttt{</td>
<td>Structure of fundus of eye</td>
<td>}\textsuperscript{2090}</td>
<td>181216001 \texttt{</td>
</tr>
<tr>
<td>\texttt{^}128477000 \texttt{</td>
<td>Abscess</td>
<td>}\textsuperscript{2086}</td>
<td>128477000 \texttt{</td>
</tr>
<tr>
<td>\texttt{^}73211009 \texttt{</td>
<td>Diabetes mellitus</td>
<td>}\textsuperscript{2091}</td>
<td>46635009 \texttt{</td>
</tr>
<tr>
<td>8801005 \texttt{</td>
<td>Secondary diabetes mellitus</td>
<td>}\textsuperscript{2093}</td>
<td>181216001 \texttt{</td>
</tr>
</tbody>
</table>

---

2084 \url{http://snomed.info/id/900000000000455006}
2085 \url{http://snomed.info/id/410662002}
2086 \url{http://snomed.info/id/128477000}
2087 \url{http://snomed.info/id/128477000}
2088 \url{http://snomed.info/id/723264001}
2089 \url{http://snomed.info/id/181216001}
2090 \url{http://snomed.info/id/65784005}
2091 \url{http://snomed.info/id/73211009}
2092 \url{http://snomed.info/id/46635009}
2093 \url{http://snomed.info/id/8801005}
### Appendix D - ECL Quick Reference

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt; x</td>
<td>The set of all descendants (both direct and indirect) of x, plus x itself</td>
<td>73211009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73211009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>46635009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8801005</td>
</tr>
<tr>
<td>&lt;! x</td>
<td>The set of all immediate children of x</td>
<td>362965005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49601007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>362969004</td>
</tr>
<tr>
<td>&lt;&lt;! x</td>
<td>The set of all immediate children of x, plus x itself</td>
<td>362965005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>49601007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>362969004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>362965005</td>
</tr>
<tr>
<td>&gt; x</td>
<td>The set of all ancestors (both direct and indirect) of x</td>
<td>279420009</td>
</tr>
<tr>
<td></td>
<td></td>
<td>106076001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>297968009</td>
</tr>
</tbody>
</table>
### Expression Constraint Language - Specification and Guide

#### Appendix D - ECL Quick Reference

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Evaluation Notes</th>
<th>Example</th>
<th>Example Expansion Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gg x )</td>
<td>The set of all ancestors (both direct and indirect) of ( x ), plus ( x ) itself</td>
<td>( 279420009 ) [Hematoma of skin]</td>
<td></td>
</tr>
<tr>
<td>( &gt;! x )</td>
<td>The set of all immediate parents of ( x )</td>
<td>( 22298006 ) [Myocardial infarction]</td>
<td></td>
</tr>
<tr>
<td>( \gg! x )</td>
<td>The set of all immediate parents of ( x ), plus ( x ) itself</td>
<td>( 22298006 ) [Myocardial infarction]</td>
<td></td>
</tr>
</tbody>
</table>

#### Conjunction, Disjunction and Exclusion

**Syntax**

\( x \) **AND** \( y \)

**Evaluation Notes**

The set of concepts that are both in \( x \) and in \( y \) (i.e. the intersection of \( x \) and \( y \))

**Example**

\( < 19829001 \) [Disorder of lung] **AND** \( < 87628006 \) [Bacterial infectious disease]

**Example Expansion Concepts**

\( 430395005 \) [Pneumonia caused by Gram negative bacteria]
### Expression Constraint Language - Specification and Guide

#### Appendix D - ECL Quick Reference

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Evaluation Notes</th>
<th>Example</th>
<th>Example Expansion Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>x OR y</td>
<td>The set of concepts that are either in x or in y (i.e. the union of x and y)</td>
<td>&lt; 73452002</td>
<td>Pulmonary tuberculosis^{2122}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OR &lt; 275504005</td>
<td>Tuberculous abscess of lung^{2125}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 75444003</td>
<td>Congenital cystic lung^{2126}</td>
</tr>
<tr>
<td>x MINUS y</td>
<td>The set of concepts that are in x but are not in y (i.e. x excluding concepts in y)</td>
<td>&lt; 29303009</td>
<td>Electrocardiographic procedure^{2127}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MINUS &lt; 75444003</td>
<td>Fetal electrocardiogram^{2128}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>447114004</td>
<td>12 lead electrocardiogram during exercise^{2129}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>252417001</td>
<td>24 Hour electrocardiogram^{2130}</td>
</tr>
<tr>
<td>x : a = y</td>
<td>The set of concepts in x, which have a necessary relationship with an attribute in a and a value in y</td>
<td>&lt; 385494008</td>
<td>Hematoma^{2131}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>: &lt;&lt; 370135005 Pathological process^{2132} = &lt;&lt; 441862004 Infectious process^{2133}</td>
<td>Infected hematoma^{2134}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>698573001</td>
<td>Infection of wound hematoma^{2135}</td>
</tr>
</tbody>
</table>

---

{^{2122} http://snomed.info/id/154283005}
{^{2123} http://snomed.info/id/73452002}
{^{2124} http://snomed.info/id/275504005}
{^{2125} http://snomed.info/id/44654007}
{^{2126} http://snomed.info/id/87119009}
{^{2127} http://snomed.info/id/29303009}
{^{2128} http://snomed.info/id/75444003}
{^{2129} http://snomed.info/id/447114004}
{^{2130} http://snomed.info/id/252417001}
{^{2131} http://snomed.info/id/385494008}
{^{2132} http://snomed.info/id/370135005}
{^{2133} http://snomed.info/id/441862004}
{^{2134} http://snomed.info/id/698573001}
{^{2135} http://snomed.info/id/444109008}
The set of concepts in \( x \), which have both a necessary relationship with an attribute in \( a \) and a value in \( y \), and also have a necessary relationship (either the same one or a different one) with an attribute in \( b \) and a value in \( v \):

\[
x : a = y, b = v
\]

The set of concepts in \( x \), which have a role group that contains both a necessary relationship with an attribute in \( a \) and a value in \( y \), and also have a necessary relationship (either the same one or a different one) with an attribute in \( b \) and a value in \( v \):

\[
x : \{ a = y, b = v \}
\]

---

**Cardinality**

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Evaluation Notes</th>
<th>Example</th>
<th>Example Expansion Concepts</th>
</tr>
</thead>
</table>

---

2136 http://snomed.info/id/71388002
2137 http://snomed.info/id/363704007
2138 http://snomed.info/id/69695003
2139 http://snomed.info/id/405815000
2140 http://snomed.info/id/86174004
2141 http://snomed.info/id/708987006
2142 http://snomed.info/id/57922004
2143 http://snomed.info/id/71388002
2144 http://snomed.info/id/405813007
2145 http://snomed.info/id/10200004
2146 http://snomed.info/id/260686004
2147 http://snomed.info/id/129433002
2148 http://snomed.info/id/773252007
2149 http://snomed.info/id/20933000
2150 http://snomed.info/id/773252007
2151 http://snomed.info/id/20933000
### Expression Constraint Language

**Appendix D - ECL Quick Reference**

#### x : [min .. max] 

<table>
<thead>
<tr>
<th>a = y</th>
</tr>
</thead>
</table>

The set of concepts in `x`, which have between `min` and `max` necessary relationships with an attribute in `a` and a value in `y`.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt; 373873005</code></td>
<td><a href="http://snomed.info/id/373873005">Pharmaceutical / biologic product</a></td>
</tr>
<tr>
<td><code>127489000</code></td>
<td>Has active ingredient <a href="http://snomed.info/id/105590001">Substance</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>786732006</code></td>
<td><a href="http://snomed.info/id/2153">Product containing only brompheniramine and codeine and phenylpropanolamine</a></td>
</tr>
<tr>
<td><code>787979009</code></td>
<td><a href="http://snomed.info/id/2154">Product containing cyanocobalamin and folic acid and pyridoxine</a></td>
</tr>
</tbody>
</table>

#### x : [min .. max] 

{ a = y }

The set of concepts in `x`, which have between `min` and `max` role groups that contain a necessary relationship with an attribute in `a` and a value in `y`.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>&lt; 404684003</code></td>
<td><a href="http://snomed.info/id/404684003">Clinical finding</a></td>
</tr>
<tr>
<td><code>363698007</code></td>
<td>Finding site <a href="http://snomed.info/id/72704001">Fracture</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concept</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>271577005</code></td>
<td><a href="http://snomed.info/id/2159">Fracture of shaft of tibia and fibula</a></td>
</tr>
<tr>
<td><code>5857000</code></td>
<td>Fracture of radius AND ulna <a href="http://snomed.info/id/2160">Fracture</a></td>
</tr>
</tbody>
</table>

### Reversed Attributes

#### Syntax

<table>
<thead>
<tr>
<th>Evaluation Notes</th>
<th>Example</th>
<th>Example Expansion Concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

2150 http://snomed.info/id/373873005  
2151 http://snomed.info/id/127489000  
2152 http://snomed.info/id/105590001  
2153 http://snomed.info/id/786732006  
2154 http://snomed.info/id/787979009  
2155 http://snomed.info/id/404684003  
2156 http://snomed.info/id/363698007  
2157 http://snomed.info/id/116676008  
2158 http://snomed.info/id/72704001  
2159 http://snomed.info/id/271577005  
2160 http://snomed.info/id/75857000
| y: R a = x | The set of concepts in y, which are the destination (ie attribute value) of a necessary relationship on a source concept in x with an attribute in a | < 91723000 | [Anatomical structure] 2161 | : R 363698007 | [Finding site] 2162 | = < 445945000 | [Infectious disease associated with acquired immune deficiency syndrome] 2163 |
| x . a | The set of attribute values (ie destination concepts) of all necessary relationships on a source concept in x with an attribute in a | < 27658006 | [Product containing amoxicillin] 2166 | . 127489000 | [Has active ingredient] 2167 |

Appendix D - ECL Quick Reference – 205
12 Appendix E - Reference Set Fields

In the SNOMED CT Release File Specification (http://snomed.org/rfs), SNOMED International specifies a set of reference set types\footnote{https://confluence.ihtsdotools.org/display/DOCRELFMT/5.2+Reference+Set+Types} with their own specific properties (e.g. an attribute value type reference set). Each reference set that is developed to conform to a specified type is defined as a subtype of the associated reference set concept (e.g. 90000000000480006 | Attribute value type reference set\footnote{http://snomed.info/id/90000000000480006}). All reference sets of a given type are populated with members using the same data structure - with the same set of field names in the same order. SNOMED International uses these reference set type data structures (as defined in the Release File Specification\footnote{http://snomed.info/id/90000000000455006}) as the release file format for all reference sets of that type.

All reference set type\footnote{http://snomed.info/id/90000000000455006} concepts are a subtype of 90000000000455006 | Reference set\footnote{http://snomed.info/id/90000000000456007}, and have an associated set of reference set descriptors in the | Reference set descriptor reference set\footnote{http://snomed.info/id/90000000000456007}. Some reference set type concepts are organised under one or more reference set groups (e.g. 723564002 | MRCM reference set\footnote{http://snomed.info/id/723564002}), which represent a group of reference set types (often with different data structures).

In the Expression Constraint Language (v2.0+) reference set field names are used to indicate which field values to return, and to filter reference set members based on specific field criteria. The first (non-metadata) field in every reference set (in order '0') must always be 'referencedComponentId'. For reference sets, which are a subtype of an international reference set type, the additional field names defined in the SNOMED CT Release File Specification\footnote{http://snomed.info/id/723564002} must be used. In all other cases, the additional field names may use any latin-script alphabetic character (a-z or A-Z) defined by the owner of the corresponding reference set type concept. Owners of a reference set type are encouraged to explicitly document these field names, keep them unchanged and publish a machine readable representation of these (following the format used below). In the absence of this, the column name from the corresponding RF2 file (with all whitespace removed) will be used.

The international reference set types and their corresponding list of field names to be used in ECL v2.0+ are shown in the table below (for information only). A normative, computable representation of this table is attached below the table. Please note that this file may be extended by implementers with national or local reference set types.

<table>
<thead>
<tr>
<th>Content Reference Set Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reference Set Type</strong></td>
</tr>
<tr>
<td>4466090009</td>
</tr>
<tr>
<td>7336190002</td>
</tr>
<tr>
<td>90000000000480006</td>
</tr>
</tbody>
</table>

\footnote{2171 http://snomed.info/id/900000000000480006}
### Language Reference Set Types

<table>
<thead>
<tr>
<th>Reference Set Type</th>
<th>Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000000000000521006</td>
<td>referencedComponentId,targetComponentId</td>
</tr>
<tr>
<td>733618005</td>
<td>referencedComponentId,targetComponentId,order</td>
</tr>
<tr>
<td>9000000000000516008</td>
<td>referencedComponentId,annotation</td>
</tr>
<tr>
<td>9000000000000512005</td>
<td>referencedComponentId,query</td>
</tr>
<tr>
<td>447258008</td>
<td>referencedComponentId,order,linkedTold</td>
</tr>
<tr>
<td>762676003</td>
<td>referencedComponentId,owlExpression</td>
</tr>
<tr>
<td>1119417006</td>
<td>referencedComponentId,expression,substrate</td>
</tr>
</tbody>
</table>

### Map Reference Set Types

<table>
<thead>
<tr>
<th>Reference Set Type</th>
<th>Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>9000000000000506000</td>
<td>referencedComponentId,acceptabilityId</td>
</tr>
<tr>
<td>9000000000000496009</td>
<td>referencedComponentId,mapTarget</td>
</tr>
</tbody>
</table>

2181 http://snomed.info/id/9000000000000521006
2182 http://snomed.info/id/733618005
2183 http://snomed.info/id/9000000000000516008
2184 http://snomed.info/id/9000000000000512005
2185 http://snomed.info/id/447258008
2186 http://snomed.info/id/762676003
2187 http://snomed.info/id/1119417006
2188 http://snomed.info/id/9000000000000506000
2189 http://snomed.info/id/9000000000000496009
### Appendix E - Reference Set Fields

<table>
<thead>
<tr>
<th>Reference Set Type</th>
<th>Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple map to SNOMED CT type reference set</td>
<td>referencedComponentId,mapSource</td>
</tr>
<tr>
<td>Complex map from SNOMED CT type reference set</td>
<td>referencedComponentId,mapGroup,mapPriority,mapRule,mapAdvice,mapTarget,correlationId</td>
</tr>
<tr>
<td>Extended map from SNOMED CT type reference set</td>
<td>referencedComponentId,mapGroup,mapPriority,mapRule,mapAdvice,mapTarget,correlationId,mapCategoryId</td>
</tr>
<tr>
<td>Map to SNOMED CT with correlation and origin type reference set</td>
<td>referencedComponentId,mapSource,attributeId,correlationId,conceptOriginId</td>
</tr>
<tr>
<td>Code to expression type reference set reference set</td>
<td>referencedComponentId,mapSource,expression,definitionStatusId,correlationId,conceptOriginId</td>
</tr>
<tr>
<td>Simple map with correlation from SNOMED CT type reference set</td>
<td>referencedComponentId,mapTarget,correlationId</td>
</tr>
<tr>
<td>Simple map with correlation to SNOMED CT type reference set</td>
<td>referencedComponentId,mapSource,correlationId</td>
</tr>
<tr>
<td>Simple map with correlation from SNOMED CT to SNOMED CT type reference set</td>
<td>referencedComponentId,mapTarget,correlationId</td>
</tr>
</tbody>
</table>

### Metadata Reference Set Types

<table>
<thead>
<tr>
<th>Reference Set Type</th>
<th>Field Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference set descriptor type reference set</td>
<td>referencedComponentId,attributeDescription,attributeType,attributeOrder</td>
</tr>
</tbody>
</table>

---

2190 http://snomed.info/id/1187636009
2191 http://snomed.info/id/447250001
2192 http://snomed.info/id/609331003
2193 http://snomed.info/id/705111002
2194 http://snomed.info/id/705109006
2195 http://snomed.info/id/1193542003
2196 http://snomed.info/id/1193543008
2197 http://snomed.info/id/1193544002
2198 http://snomed.info/id/900000000000456007
<table>
<thead>
<tr>
<th>Reference Set ID</th>
<th>Description</th>
<th>Example Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>900000000000534007</td>
<td>Module dependency type reference set</td>
<td>referencedComponentId, sourceEffectiveTime, targetEffectiveTime</td>
</tr>
<tr>
<td>900000000000538005</td>
<td>Description format type reference set</td>
<td>referencedComponentId, descriptionFormat, descriptionLength</td>
</tr>
<tr>
<td>723589008</td>
<td>MRCM domain type reference set</td>
<td>referencedComponentId, domainConstraint, parentDomain, proximalPrimitiveConstraint, proximalPrimitiveRefinement, domainTemplateForPrecoordination, domainTemplateForPostcoordination, guideURL</td>
</tr>
<tr>
<td>723604009</td>
<td>MRCM attribute domain type reference set</td>
<td>referencedComponentId, domainId, grouped, attributeCardinality, attributeInGroupCardinality, ruleStrengthId, contentTypeId</td>
</tr>
<tr>
<td>723592007</td>
<td>MRCM attribute range type reference set</td>
<td>referencedComponentId, rangeConstraint, attributeRule, ruleStrengthId, contentTypeId</td>
</tr>
<tr>
<td>723563008</td>
<td>MRCM module scope type reference set</td>
<td>referencedComponentId, mrcmRuleRefsetId</td>
</tr>
</tbody>
</table>
13 References

6. SNOMED CT Languages Github Repository, [https://github.com/IHTSDO/SNOMEDCT-Languages](https://github.com/IHTSDO/SNOMEDCT-Languages)
14 Previous Versions

This page provides access to downloadable PDF copies of the current and previous versions of this document. The most recent version should also be available via a link on the front page of the document.
15 Recent Updates

15.1 The most recently updated pages in this document are listed below

- Expression Constraint Language - Specification and Guide (see page 7)
  - 2 minutes ago • updated by Kai Kewley
  - view change

- 1. Introduction (see page 8)
  - 20 minutes ago • updated by Kai Kewley
  - view change

- 5.1 Brief Syntax (Normative) (see page 21)
  - 27 minutes ago • updated by Kai Kewley
  - view change

- 5.2 Long Syntax (Informative) (see page 26)
  - 29 minutes ago • updated by Kai Kewley
  - view change

- 6.1 Simple Expression Constraints (see page 64)
  - 2023-Aug-24 • updated by Kai Kewley
  - view change

- 6.12 Top and Bottom (see page 129)
  - 2023-Jul-17 • updated by Kai Kewley
  - view change

- Appendix D - ECL Quick Reference (see page 196)
  - 2023-Jul-12 • updated by Kai Kewley
  - view change

- 1. Introduction (see page 8)
  - 2022-Sep-01 • updated by Linda Bird
  - view change

- 6.11 History Supplements (see page 124)
  - 2022-Aug-31 • updated by Anne Randorff Højen
  - view change

- Expression Constraint Language - Specification and Guide (see page 7)
  - 2022-Aug-24 • updated by Linda Bird
  - view change

- Appendix C - Dialect Aliases (see page 193)
  - 2022-Aug-11 • updated by Linda Bird
  - view change

- 6.8 Description Filters (see page 105)

2205 https://confluence.ihtsdotools.org/display/~kewley
2206 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=26840518&selectedPageVersions=50&selectedPageVersions=51
2207 https://confluence.ihtsdotools.org/display/~kewley
2208 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=28739382&selectedPageVersions=17&selectedPageVersions=18
2209 https://confluence.ihtsdotools.org/display/~kewley
2210 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=28739405&selectedPageVersions=17&selectedPageVersions=18
2211 https://confluence.ihtsdotools.org/display/~kewley
2212 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=28739406&selectedPageVersions=15&selectedPageVersions=16
2213 https://confluence.ihtsdotools.org/display/~kewley
2214 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=28739409&selectedPageVersions=13&selectedPageVersions=14
2215 https://confluence.ihtsdotools.org/display/~kewley
2216 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=212338972&selectedPageVersions=1&selectedPageVersions=2
2217 https://confluence.ihtsdotools.org/display/~kewley
2218 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=115883413&selectedPageVersions=5&selectedPageVersions=6
2219 https://confluence.ihtsdotools.org/display/~ibird
2220 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=28739382&selectedPageVersions=16&selectedPageVersions=17
2221 https://confluence.ihtsdotools.org/display/~ahojen
2222 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=142144136&selectedPageVersions=1&selectedPageVersions=2
2223 https://confluence.ihtsdotools.org/display/~ibird
2224 https://confluence.ihtsdotools.org/pages/difffpagesbyversion.action?pageId=26840518&selectedPageVersions=49&selectedPageVersions=50
2225 https://confluence.ihtsdotools.org/display/~ibird
Recent Updates – 213