# 6.2 Subsumption

Determining whether one concept (or expression) is a kind of another concept (or expression) is the fundamental capability enabled by SNOMED CT. For example, answering the question 'Which patients have an infectious disease?' involves finding all the patients with *any kind of* infectious disease (e. g. viral pneumonia, tuberculosis).

Subsumption occurs when one clinical meaning is a subtype of another clinical meaning, and testing for this is called 'subsumption testing'. If clinical meaning X is a subtype of clinical meaning Y, then Y is said to 'subsume' X and X is 'subsumed by' Y.

Subsumption testing between concepts is represented using a stated or implied |is a| relationship. For example, 75570004 |viral pneumonia| is a 40733004 |infectious disease| and therefore 40733004 |infectious disease| subsumes 75570004 |viral pneumonia|, and 75570004 |viral pneumonia| is subsumed by 40733004 |infectious disease|.

Subsumption testing between expressions tests to see if the *candidate expression* (often recorded in a patient record) is subsumed by a *predicate expression* (typically part of the query being run across the patient record). For example:

*Candidate expression*: 75570004 |viral pneumonia|

*Predicate expression*: 40733004 |infectious disease|:

363698007 |finding site| = 39607008 |lung structure|

In this case, the candidate expression *is* subsumed by the predicate expression.

Subsumption testing can be represented using the SNOMED CT Expression Constraint Language using the '<' (descendantOf) or '<<' (descendantOrSelfOf) operators. For example, the expression constraint:

<< 40733004 |infectious disease|

is satisfied by any expression that is subsumed by 40733004 |infectious disease|.

There are a variety of ways to implement subsumption testing. These are summarized in the Implementation sub-section below.

## Example

A typical example using subsumption would be an audit within a hospital, reviewing all patients with an infectious disease. In this scenario, the following simple query could be executed to find all the patients whose health record contains a diagnosis that is subsumed by the concept 40733004 |i nfectious disease|:

SELECT distinct patientID

FROM health_records

WHERE diagnosis = (<< 40733004 |infectious disease|)

If the health records contained the following data:

| patientID | date | diagnosis |
|---|---|---|
| 634711 | 16th January 2015 | 71620000\|fracture of femur\| |
| 634711 | 25th January 2015 | 415353009\|rotavirus food poisoning\| |
| 634711 | 3rd February 2015 | 66308002\|fracture of humerus\| |
| 158775 | 7th January 2015 | 40468003\|hepatitis A\| |
| 889125 | 7th January 2015 | 75570004 \|viral pneumonia\| |
| 456872 | 15th January 2015 | 22298006\|myocardial infarction\| |
| 456872 | 15th January 2015 | 195967001\|asthma\| |

Then this query would return the following list of patients:

- 634711 (because 415353009 |rotavirus food poisoning| is a subtype of 40733004 |infectious disease|)
- 158775 (because 40468003|hepatitis A| is a subtype of 40733004 |infectious disease|)
- 889125 (because 75570004 |viral pneumonia| is a subtype of 40733004 |infectious disease|)

Note that patient 456872 would not be returned by this query as neither 22298006 |myocardial infarction| or 195967001 |asthma| are subtypes of 40733004 |infectious disease|.

## Implementation

## Testing Subsumption Between Concepts

Rapid and efficient computation of whether a concept |is a| subtype descendant of another concept is essential for testing subsumption between expressions. A variety of approaches exist for testing subsumption. When the candidate and predicate expressions are both precoordinated concepts, subsumption testing can use the published relationships from the SNOMED CT release files. Approaches for testing subsumption between precoordinated concepts include:

- Exhaustive testing of subtype relationships

In this approach, every possible sequence of |is a| relationships are recursively tested from the candidate concept until the predicate concept is reached or until all possible paths have been exhausted.

- Semantic type identifiers and hierarchy flags

In this approach, flags are added to each concept to indicate the set of high-level concept nodes of which that concept is a subtype. A concept can only subsume concepts that include the same set of high-level concept flags. This reduces the number of tests that need to be performed to recursively test the subtype relationships.

- Use of proprietary database features

In this approach, a database is used which supports the recursive testing of a chain of hierarchical relationships.

- Branch numbering

In this approach, a depth first tree walk is performed that applies an incremental number to each concept. A second tree walk then allocates one or more branch number ranges to each concept, which contains the number of all of their descendants.

- Precomputed transitive closure table

In this approach, a comprehensive list of all supertypes of each concept is created by recursively traversing all |is a| relationships and adding each stated and inferred subtype relationship to a table.

- Using a Description Logic Reasoner

In this approach, a description logic reasoner (e.g. Snorocket, ELK, Fact++) is used to determine whether one concept is subsumed by another.

In most environments, the recommended approach is to either use a precomputed transitive closure table or a description logic reasoner. However, where disk capacity or distribution bandwidth are limiting factors, branch numbering provides an efficient alternative approach. For more information on these approaches, please refer to Subtype search scope restriction in the SNOMED CT Terminology Services Guide.

## Testing Subsumption Between Expressions

When either the candidate expression (in the patient data) or the predicate expression in the query are postcoordinated (or both), techniques based on description logic are needed to perform subsumption testing. Approaches for testing subsumption between postcoordinated expressions include:

- Comparing normal form expressions

In this approach, the predicate expression is transformed to short normal form and the candidate expression is transformed to long normal form. The two normal form expressions are then tested for subsumption by checking that each focus concept in the predicate expression subsumes at least one focus concept in the candidate expression, each attribute group in the predicate expression subsumes at least one attribute group in the candidate expression and each ungrouped attribute in the predicate expression subsumes at least one attribute in the candidate expression.

- Using a Description Logic Reasoner

In this approach, a description logic reasoner (e.g. Snorocket, ELK, Fact++) is used to determine whether one expression is subsumed by another.

Where available, the recommended approach is to use a description logic reasoner to calculate subsumption between expressions. However, comparing normal form expressions provides an alternative approach when a reasoner is not available. For more information on these approaches, please refer to Expression Retrieval and Normal Forms in the SNOMED CT Terminology Services Guide.

## Case Studies

A number of vendor products use the SNOMED CT hierarchy to support subsumption testing in their analytics services, including the Cerner Millennium Terminology (CMT) package and Epic's decision support and reporting tools. Terminology servers that provide the ability to perform subsumption testing include B2i Healthcare's Snow Owl® terminology server. The UK Terminology Centre's Data Migration Workbench also uses subsumption testing in its query tool, and its case mix and caseload trends analysis tools.