INTERNATIONAL STANDARD

ISO 24623-1

First edition
2018-04

Language resource management — Corpus query lingua franca (CQLF) —

Part 1: Metamodel

Gestion des ressources linguistiques — Corpus query lingua franca (CQLF) —

Partie 1: Métamodèle

Reference number
ISO 24623-1:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 37, *Language and terminology*, Subcommittee SC 4, *Language resource management*.

A list of all parts in the ISO 24623 series can be found on the ISO website. Additional parts on single-stream and multi-stream ontology architectures are planned to be developed in the future.

Introduction

A range of standards relating to language resource management, with the Linguistic Annotation Framework (ISO 24612) at the centre, have been developed. These standards are mostly designed to regulate the representation aspect of language data – they look at the data from the point of view of preparation and curation. This document complements this perspective by that of the end-user, that is to say, from the point of view of processing and querying.

The corpus linguistic community has, by now, developed several corpus query languages (QLs), and there is a particularly large number of them if "dialects" and forks are included. There are two main reasons for this abundance. Firstly, there are socio-economic and organizational factors, with separate query systems having been created by isolated projects with un-coordinated funding, many of them eventually developing their own set of followers. Secondly, query systems are typically sensitive to the format of the data and are often designed with a specific purpose in mind. For example, systems for querying parallel audio and transcription streams with multiple speakers have different characteristics from systems designed to query purely textual data with a single layer of morphosyntactic description. Dependency and hierarchical annotations demand yet another set of solutions. All of this results in the richness of alternatives or near-alternatives on the one hand, and in the lack of interoperability among the variants on the other. As a consequence, a "wrong" choice at the beginning of a project can bury months of research by exposing inadequacies in the initial decision after the project has become mature enough to move to new extended functionality and towards addressing more complex information needs.

This document codifies, in a modular way, the best existing practices followed in the design of corpus query languages. Its theoretical aim is to provide a basis for the investigation of the relationships between language resource architecture and corpus query language properties. The practical aim of the Corpus Query *Lingua Franca* (henceforth CQLF) is to provide linguists and language technology practitioners with a clear and coherent basis for making informed choices concerning data architectures and the query languages appropriate to them.

Language resource management — Corpus query lingua franca (CQLF) —

Part 1: Metamodel

1 Scope

This document describes the abstract metamodel designed to accommodate any corpus query language (QL) and providing a basis for coarse-grained classification. The metamodel consists of several components referred to as CQLF classes, levels, and modules, and is illustrated with examples from the Single-stream class (where a single data stream is used to organize the relevant data structures). Within this class, this document discusses three CQLF levels (Linear, Complex and Concurrent), as well as their subdivisions into modules, dictated by functional and modelling criteria.

This document does not provide a way to specify further details beyond the above-mentioned divisions, and neither does it contain within its scope QLs designed to query more than one concurrent data stream, as in multimodal corpora or in parallel corpora (such QLs can still be classified according to the criteria suggested here for less expressive QLs).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 24611, *Language resource management — Morpho-syntactic annotation framework (MAF)*

ISO 24612, *Language resource management — Linguistic annotation framework (LAF)*

ISO 24615-1, *Language resource management — Syntactic annotation framework (SynAF) — Part 1: Syntactic model*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— IEC Electropedia: available at https://www.electropedia.org/

— ISO Online browsing platform: available at https://www.iso.org/obp

3.1
annotation
information added to *primary data* (3.9), independent of its representation

[SOURCE: ISO 24612:2012, 2.3, modified — "linguistic" at the beginning of the definition was deleted.]

3.1.1
concurrent annotations
multiple, potentially conflicting *annotations* (3.1) describing, entirely or partly, the same *character span* (3.2) or an overlapping sequence of character spans

Note 1 to entry: Concurrent annotations may be expected to conflict in several ways: content-wise (with different tags for the same character span), structure-wise (assuming different structural arrangements within the targeted character spans), and also in terms of segment edges (which is typically due to structurally conflicting claims concerning the encompassing character spans). Concurrent annotations typically come from different sources (e.g. tools or human annotators) or result from different settings (e.g. different parsing models or segmentation rules) within a single tool. When encoded in XML, concurrent annotations are typically expressed by means of stand-off techniques.

3.1.2
dependency annotation
annotation (3.1) that encodes the dependency relations between *character spans* (3.2)

Note 1 to entry: An example of a dependency relation (see ISO 24615-1:2014, 3.5) is one between a verb and its subject or direct object, between an attributive adjective and its head noun, or between a preposition and the head of its dependent noun phrase. Dependency relations may be defined at the word-level alone, or may involve higher-level syntactic constructs, in which case it is possible to speak of mixed hierarchical-dependency annotations.

3.1.3
hierarchical annotation
annotation (3.1) that encodes the relationship of dominance (often also precedence) necessary to define syntactic trees over *character spans* (3.2)

Note 1 to entry: Annotating hierarchical relationships requires only the relation of dominance to be indicated. Precedence is typically implicit in the ordering of character spans.

3.1.4
segmentation annotation
annotation (3.1) that delimits linguistic elements that appear in the *primary data* (3.9)

Note 1 to entry: These elements include (1) continuous segments (appearing contiguously in the primary data), (2) super- and sub-segments, where groups of segments will comprise the parts of a larger segment (e.g. contiguous word segments typically comprise a sentence segment), (3) discontinuous segments (linking continuous segments) and (4) landmarks (e.g. time stamps) that note a point in the primary data. In current practice, segmental information may or may not appear in the document containing the primary data itself.

[SOURCE: ISO 24612:2012, 2.5]

3.1.5
simple annotation
annotation (3.1) that constitutes a single information package whose interpretation is not dependent on other annotations

Note 1 to entry: This definition is intended to distinguish the simplest ("tabular") kind of annotation from more complex relational structures (providing hierarchical, dependency, or alignment information); simple annotations are the only kind of annotations present at the linear level of complexity.

3.1.6
stand-off annotation
annotation (3.1) that can be layered over *primary data* (3.9) but is separated from the data stream that it targets

Note 1 to entry: Stand-off annotations refer to specific locations in the primary data, by addressing the character offsets, elements or coordinates to which the annotation applies. They can be serialized as separate documents, but do not have to be. Multiple stand-off annotation documents for a given type of annotation can refer to the same primary document (e.g. two different part of speech annotations for a given text). It is also possible to construct hierarchies of stand-off annotation layers, where layer *n* can reference layers 0..*n*−1.

[SOURCE: ISO 24612:2012, 2.7, modified — The definition and note were modified.]

3.2
character span
sequence of characters, identified by start and end offsets, to which an annotation may be applied

Note 1 to entry: This definition is a relaxed version of the definition in ISO 24615-1:2014, 3.16, the difference lying in the use of “may be applied” over “is applied”. Compare also the definition of “region” in ISO 24612:2012, 2.10.

3.3
character span containment
relation obtaining between *character spans* (3.2) of *primary data* (3.9) in which character span A contains character span B if the initial offset of span A is equal to or higher than that of span B, and the final offset of span A is smaller than or equal to that of span B

Note 1 to entry: The relation of character span containment is used for stating a relationship between two or more character spans or simple annotations, without the need to utilize tree-based concepts and mechanisms. Instead of tree traversal, operators such as *contains*, *in* or *within* are typically used for character span containment queries.

3.4
corpus query language
formal language designed to retrieve specific information from (large) language data collections, and thereby incorporate certain abstractions over commonly shared data models that make it possible for the user (or user agents) to address parts of those data models

3.5
CQLF class
top-level division in the CQLF data model

Note 1 to entry: The CQLF Metamodel distinguishes two classes: Single-stream (where the annotation structure is built upon a single data stream, typically a character stream) and Multi-stream (corresponding to e.g. multi-modal corpora or parallel corpora).

3.6
CQLF implementation
query language that has been analysed with respect to the criteria described by the CQLF Metamodel, and thus has been “located” in the proposed feature matrix as “conformant with CQLF”

3.7
CQLF level
part of the matrix of QL properties, defined in terms of the general features of the assumed corpus data models, and consequently the set of properties of a corpus query language that is used to address these features

Note 1 to entry: The CQLF Metamodel distinguishes three levels of complexity within the Single-stream class: Linear, Complex and Concurrent.

3.8
CQLF module
subcomponent of a CQLF level, defined with reference to a specified data-model characteristic

Note 1 to entry: CQLF Metamodel currently distinguishes three modules within CQLF Level 1, Linear (plain-text, segmentation, and simple annotation), and three modules within CQLF Level 2, Complex (hierarchical, dependency, and containment).

3.9
primary data
electronic representation of language data

3.10
token
non-empty contiguous sequence of graphemes or phonemes in a document

[SOURCE: ISO 24611:2012, 3.21, modified — The note was deleted.]

4 Aims

The CQLF Metamodel is intended to establish a frame and a basis for establishing the potential extent and the limits of interoperability between different corpus query systems. It aims to provide a single matrix of a few well-defined properties in which any corpus QL can be located for the purpose of coarse-grained comparison with the others. Further parts of the standard elaborate on these properties and flesh out the relationships among them. A long-term goal of CQLF is, additionally, to help reduce the gap between end users with a linguistic or literary background and powerful search environments.

While CQLF as a whole might be expected to mediate between individual corpus QLs as an interlingua, such initiatives raise a host of problems, ranging from low-level technical descriptions (e.g. the inability to preserve information when translating between regular expressions on the one hand and wildcards on the other) to issues of epistemology ("Does the result of the query reformulated in QL 2 address exactly the information need expressed in QL 1?"). The immediate goal of CQLF is therefore more modest: to serve as the target space within which QLs can be located with respect to their basic properties. It can thus also serve as a measure of compatibility and interoperability, but without the added claim to provide QL-to-QL mappings. A robust CQLF-based bi-directional mapping system in action (with the epistemological burden appropriately controlled) would be an interesting challenge in the long run, but it makes a lot of sense to start smaller, and in the spirit of other standards, lay the ground for a pivot-based system with monodirectional mapping from various corpus QLs into a representation defined by a superset of their individual properties.

The metamodel presented here circumscribes the outer limits of QL compatibility. CQLF may be used as a set of guidelines to be applied in the development of a new QL or in the enrichment of an existing QL with new functionality. It is likely that information about the extent and points of conformance of a given QL with CQLF will also be useful to corpus linguists for the purpose of identifying the QL suitable for the task that they are faced with.

5 Assumptions

The metamodel described here builds on models defined in other standards. In particular, its infrastructure includes:

— the general data model for corpus and annotation description defined by ISO 24612 (LAF), together with the more detailed models defined by ISO 24611 (MAF) and ISO 24615-1 (SynAF);

— a common repository of data content and data containers (see ISO 12620).

The simple data model for Single-stream architectures (as opposed to, e.g., multi-modal corpora or parallel corpora) recognizes a minimum of two layers at which text can be queried: the layer of characters and the layer of labelled abstractions over characters (interpreted by the annotator or tools). The sequences in the former layer can be defined by means of character-based regular expressions, whereas sequences in the latter can be defined using ISO/IEC 14977 (EBNF).

An additional assumption made in the present specification is that tokenization is merely a special kind of segmentation annotation, upon which hierarchies of other annotation layers can be built.

6 CQLF Metamodel

CQLF is designed to be a modular construction with several components. Each component is characterised with respect to some aspect of the data models describing corpus objects that are the target or context of queries. A schematic view of the components of a CQLF Metamodel is presented in Figure 1. The top-level components are referred to as CQLF Classes and correspond to the major division into data models built upon a single data stream vs. those which use more than one data stream (be it binary or text-based) in parallel. This document illustrates an instantiation of the metamodel for the Single-stream class (to be introduced below), which consists of three CQLF Levels that correspond to the major kinds of data organization in linguistic corpora.

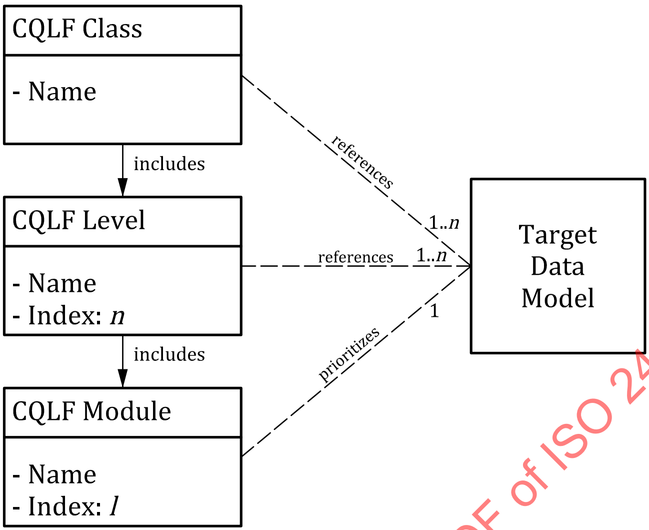


Figure 1 — Components of the CQLF Metamodel

The *index* variables of CQLF Level and Module are convenience details that allow for easier reference. The identification of levels and modules, while grounded in formal properties of data models and in functional characteristics of query languages, is also partially utilitarian, where it neglects some distinctions that could otherwise be recognised, in order to provide a simpler mechanism for determining conformance with the overall model and for stating the most important similarities and differences between corpus QLs.

The relationships sketched in Figure 1, together with the basic divisions made on the basis of the existing corpus QLs, yield a basic taxonomy presented in the diagram in Figure 2.

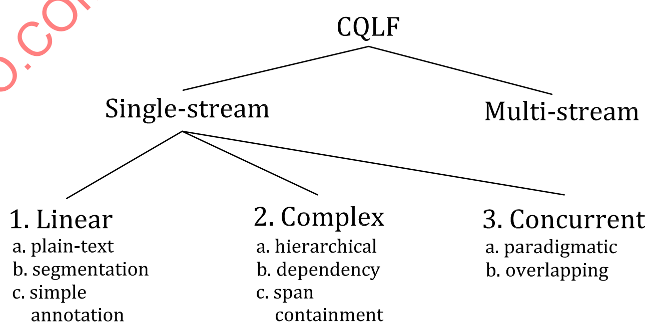


Figure 2 — Partial CQLF taxonomy of query language properties

Within the Single-stream class, each consecutive level introduces a more complex search dimension. The level system is based on the distinction between different major types of data organization and consequently different types of annotations (modelled by various parts of the LAF family of standards), as well as queries that correspond to them.

- **Level 1 (Linear)** addresses, in any combination, plain-text search (1a) as well as search in segmented data (segmentation annotations; 1b), and in simple annotations (1c) attached to particular segments:
  - at this level, annotations (if present) form a single layer of objects that exhaustively or partially describe the primary data stream;

  — tokenization is treated as a level of segmental annotation;

— **Level 2 (Complex)** addresses search in data annotated with hierarchical structures (2a) and/or dependency information (2b), or querying simple annotations by means of containment-based queries (2c):

  — the hierarchical module (2a) concerns tree-based representations (such as those used for phrase-structure description), whereas the dependency module (2b) focuses on the identification of relationships in which words function as nodes linked by directed arcs;

  — (2a) and (2b) can be composed into a mixed representation, with dependency relationships imposed on phrase structure;

  — (2c) addresses simplified hierarchical relationships encoded as character span containment; this can mimic phrase structure (with extra devices for resolving cases of mutual containment, from which the hierarchy cannot be derived) and is often non-recursive (as illustrated by e.g. CQP, see Reference [3]);

— **Level 3 (Concurrent)** addresses search in multiple concurrent (overlapping, intersecting and often conflicting) annotations built upon a single data stream:

  — the Paradigmatic module (3a) addresses prototypical cases where different annotation layers provide data packages describing the same location;

  — the Overlapping module (3b) addresses concurrent annotations that are built upon character spans which differ in their start and/or end offsets;

  — concurrent tokenizations are a special case, understood as mismatching segmental annotations at the highest level of granularity.

The relevant target data models shall minimally include the following:

— for Level 1: ISO 24612 (LAF), ISO 24611 (MAF);

— for Level 2: ISO 24612 (LAF), ISO 24615-1 (SynAF-Metamodel);

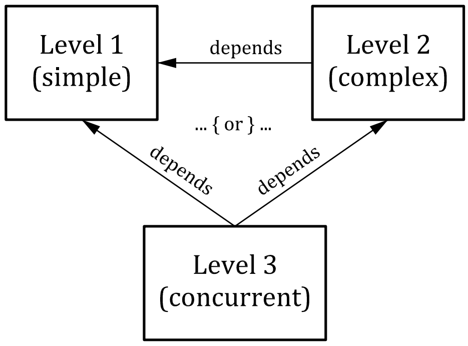
— for Level 3: the target data models *may* include ISO 24612 (LAF) and other complex models (note that LAF does not provide recommendations for putting together concurrent annotations other than by creating a super-graph over the graphs describing individual stand-off annotations).

The above list is the baseline for assessing the position of a given QL within the matrix proposed here. CQLF is meant to be maximally inclusive within the normative skeleton presented above.

Note that in order to keep the number of levels small and manageable, some concessions have been made: plain-text search has been classified together with search within segments (whether token-level or larger) and search in simple annotations due to the fact that some matching mechanisms and syntactic operators are shared by them.

In a similar vein, hierarchical and dependency-based description have been grouped together because they again share many mechanisms [with mixed models well attested: for example, the Tiger XML (see Reference [4]) model is hierarchy-based, with “secondary edges” introducing dependencies between constituents]. Containment-based relationships can be viewed as quasi-hierarchical (hierarchy can in many cases be derived from containment).

Both Level 2 (Linear) and Level 3 (Concurrent) depend on simple annotations (alternatively, on segmental annotations), hence on Level 1. Level 3 may, but does not need to, build on Level 2. These interdependencies are illustrated in Figure 3.



NOTE "{or}" is an Object Constraint Language (OCL) way of expressing the relevant disjunction: Level 3 may depend either on Level 1 or on Level 2, or on both of them at the same time.

Figure 3 — Dependencies among CQLF levels

By definition, hierarchical and dependency annotations involve character spans (anonymous or with simple annotations), either as terminal leaves in trees, or as objects over which relations are defined. Similarly, in order to create representations in which annotations of the same data stream are aligned (or contrasted), there must exist some kind of character span (even if sentence- or paragraph-sized) that is a member of the relevant relation. Levels 2 (Complex) and 3 (Concurrent) therefore assume the existence of segmental annotations (and by a minimal extension simple annotations), and additionally allow for plain-text search.

7 Conformance

The aim of the conformance statements for the CQLF Metamodel is to classify the existing corpus QLs into major categories, corresponding to the CQLF components (classes, levels and modules). This classification is intended to be maximally inclusive and coarse-grained, and is a prelude to a more thorough feature-by-feature classification in future documents.

In order to claim conformance with the CQLF Metamodel, it is enough for a QL to qualify as conformant at any leaf node of the CQLF component hierarchy illustrated in Figure 2, as long as the following conditions are also satisfied.

— To claim conformance at Level 1 (Linear) of the CQLF Metamodel, a corpus QL shall provide support for plain-text search and/or search in segmentation annotations and/or search in simple (non-hierarchical and non-dependency) annotations.

— To claim conformance at Level 2 (Complex) of the CQLF Metamodel, a corpus QL shall provide support for querying hierarchical annotations and/or dependency annotations and/or annotations (segmentation or simple) related by a containment relationship, and it shall be conformant at Level 1.

— To claim conformance at Level 3 of the CQLF Metamodel, a corpus QL shall provide support for querying the relationships/alignment between simple annotations (minimally) or between a mixture of simple annotations, hierarchical annotations, and dependency annotations, and it shall be conformant at Level 1.

It is essential to draw a distinction between CQLF conformance and CQLF Metamodel conformance; the former shall contain a list of references to the (sets of) features. This reference will be in the form of a list of Data Categories or Data Category Selections (see ISO 12620).

Note that this document does not specify conformance conditions for CQLF Class "Multi-stream".

Example conformance statements are cited in Annex A.

Annex A
(informative)

Example CQLF conformance statements

The criteria provided by the CQLF Metamodel are designed to be maximally inclusive. This Annex contains an example of how the metamodel conformance conditions apply, together with several examples of queries from various QLs.

It is expected that conformance claims will be formulated by the creators/maintainers of the given QL (or query system on the whole), who will be able to substantiate these claims by means of commented query examples. This Annex is therefore to be taken merely as an example of how the conformance procedure is expected to be applied.

It should be recalled that, formally, a CQLF Metamodel conformance statement is expressed as a set of paths across the hierarchy shown in Figure 3. Each of the paths for the Single-stream class is addressed in the sections below. For the sake of compactness, information items concerning the versioning of the particular corpus QLs and the (often long) source/documentation URLs are skipped.

A.1 Single-stream → 1. Linear → plain-text

EXAMPLE

[FRAME]	look for all plain text spans corresponding to string A	
[SIMPLE USE CASE]	look for all plain text spans corresponding to string A	
[QUERY]	hasSyntax:	$ grep "this" fileName
	isFormulatedIn:	grep
	comment:	this query finds "this" in raw text

A.2 Single-stream → 1. Linear → segmentation

EXAMPLE 1

[FRAME]	look for all objects A occurring a specific number of times in a row (= [[min. | max. | exactly] N | from N to M] times)	
[SIMPLE USE CASE]	look for all objects A occurring at least N times in a row	
[QUERY]	hasSyntax:	[orth = ja]{2,}
	isFormulatedIn:	Poliqarp
	comment:	this query matches a sequence of tokens "ja" containing at least two "ja"

EXAMPLE 2

[FRAME 1]	look for all object groups A with a specific distance between the first and the last of them (=[[exactly|min.|max.] N | from N to M] tokens)	
[FRAME 2]	look for all objects A followed by an object B (= look for all object sequences A)	
[COMPLEX USE CASE]	look for all object sequences A with a maximum distance of N tokens between the first and the last of them	
[QUERY]	hasSyntax:	NEAR (Haus;Mutter;Vater;10)
	isFormulatedIn:	DDC Query Language
	comment:	this query returns all hits that contain “Haus” and “Vater”, and there are up to 10 words between them and one of those words is “Mutter”

A.3 Single-stream → 1. Linear → simple annotation

EXAMPLE

[FRAME]	look for all objects annotated with feature A	
[SIMPLE USE CASE]	look for all objects annotated with feature A	
[QUERY]	hasSyntax:	MORPH(PRN rel dat)
	isFormulatedIn:	Cosmas II
	comment:	this query finds all relative pronouns in the dative case

A.4 Single-stream → 2. Complex → hierarchical

EXAMPLE

[FRAME]	look for all nodes A dominating node B	
[SIMPLE USE CASE]	look for all nodes A dominating node B	
[QUERY]	hasSyntax:	//NP//DT
	isFormulatedIn:	Fangorn
	comment:	this query finds a noun phrase which dominates a determiner

A.5 Single-stream → 2. Complex → dependency

EXAMPLE

[FRAME]	look for all relations labelled with A (= Look for all objects B connected by relation A to objects C)	
[SIMPLE USE CASE]	look for all relations labelled with A (= Look for all objects B connected by relation A to objects C)	
[QUERY]	hasSyntax:	pos = /P.*/ & pos = /V.FIN/ & #2 - > dep[func = "sbj"] #1
	isFormulatedIn:	ANNIS
	comment:	this query searches dependencies for pronominal subjects (func = "sbj") of finite verbs

A.6 Single-stream → 2. Complex → containment

EXAMPLE

[FRAME]	look for all [spans | span sequences] A containing a [span | span sequence] B	
[SIMPLE USE CASE]	look for all multi-token spans A containing span sequence B (startsWith, endsWith, isAround or fullMatch relation).	
[QUERY]	hasSyntax:	<s/> containing []* [tag = "N.* "] []* [tag = ”N.*”] []*
	isFormulatedIn:	Sketch Engine
	comment:	this query looks for all sentences containing more than one noun

A.7 Single-stream → 3. Concurrent → paradigmatic

EXAMPLE

[FRAME]	look for all objects annotated with feature set A, consisting of features coming from different sources	
[SIMPLE USE CASE]	look for all objects annotated with feature A from source B, ling. level C and feature D from source E, ling. level F	
[QUERY]	hasSyntax:	[tt/p = "+FIN" &/m = number:sg & mate/m = person:2]
	isFormulatedIn:	PoliqarpPlus
	comment:	this query finds all finite verbs in the 2nd person singular