INTERNATIONAL STANDARD

ISO/IEC 24707

Second edition
2018-07

Information technology — Common Logic (CL) — A framework for a family of logic-based languages

Technologies de l'information — Logique Commune (CL) — Cadre pour une famille des langages logique-basés

Reference number
ISO/IEC 24707:2018(E)

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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/IEC JTC 1, *Information technology*, SC 32, *Data management and interchange*.

This second edition cancels and replaces the first edition (ISO/IEC 24707:2007), which has been technically revised.

The main changes compared to the previous edition are as follows:

— the list of syntactic errors that have already been identified in the Defect Report has been fixed;

— the XML syntax in Annex C has been corrected and completed;

— a more general approach to annotation of CL-texts has been made;

— semantics has been modified to allow the existence of definitional extensions in CL;

— support for circular imports;

— semantics of CL-module have been clarified;

— clarification of the distinction between segregated and non-segregated dialects;

— clarification of conformance conditions has been made.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Common Logic is a logic framework intended for information exchange and transmission. The framework allows for a variety of different syntactic forms, called dialects, all translatable by a semantics-preserving transformation to a common XML-based syntax.

Common Logic has some novel features, chief among them being a syntax which permits "higher-order" constructions, such as quantification over classes or relations while preserving a first-order model theory, and a semantics which allows theories to describe intentional entities such as classes or properties. It also has provision for the use of datatypes and for naming, importing and transmitting content on the World Wide Web using XML.

Information technology — Common Logic (CL) — A framework for a family of logic-based languages

1 Scope

This document specifies a family of logic languages designed for use in the representation and interchange of information and data among disparate computer systems.

The following features are essential to the design of this document.

— Languages in the family have declarative semantics. It is possible to understand the meaning of expressions in these languages without appeal to an interpreter for manipulating those expressions.

— Languages in the family are logically comprehensive – at its most general, they provide for the expression of arbitrary first-order logical sentences.

— Languages in the family are translatable by a semantics-preserving transformation to a common XML-based syntax, facilitating interchange of information among heterogeneous computer systems.

The following are within the scope of this document:

— representation of information in ontologies and knowledge bases;

— specification of expressions that are the input or output of inference engines;

— formal interpretations of the symbols in the language.

The following are outside the scope of this document:

— specification of proof theory or inference rules;

— specification of translators between the notations of heterogeneous computer systems;

— computer-based operational methods of providing relationships between symbols in the logical "universe of discourse" and individuals in the "real world".

This document describes Common Logic's syntax and semantics.

This document defines an abstract syntax and an associated model-theoretic semantics for a specific extension of first-order logic. The intent is that the content of any system using first-order logic can be represented in this document. The purpose is to facilitate interchange of first-order logic-based information between systems.

Issues relating to computability using this document (including efficiency, optimization, etc.) are not addressed.

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/IEC 2382:2015, *Information technology — Vocabulary*

ISO/IEC 10646:2014, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 14977:1996, *Information technology — Syntactic metalanguage — Extended BNF*

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 http://www.electropedia.org/

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

3.1
axiom
any *sentence* (3.15), statement or text which is assumed to be true, or from which others are derived, or by which they are entailed

Note 1 to entry: In a computational setting, an axiom is a sentence which is never posed as a goal to be proved, but only used to prove other sentences.

3.2
conceptual graph
CG
graphical or textual display of symbols arranged according to the style of *conceptual graph theory* (3.3)

3.3
conceptual graph theory
form of first-order logic which represents existential quantification and conjunction via the assertion of logical constructs called concepts and relations which are arranged in an abstract or visually displayed graph

3.4
CLIF
text-based first-order formalism using a LISP-like list notation

Note 1 to entry: This is one of the concrete syntaxes for Common Logic (described in Annex A).

Note 2 to entry: CLIF is a KIF-based syntax that is used for illustration purposes in this document. KIF, introduced by Mike Genesereth[3], originated with the Knowledge Sharing Effort sponsored by the U.S. DARPA. The name "KIF" is not used for this syntax in order to distinguish it from the commonly used KIF dialects. No assumptions are made in this document with respect to KIF semantics; in particular, no equivalence between CLIF and KIF is intended.

Note 3 to entry: Historically, CLIF was an acronym for Common Logic Interchange Format. However, CLIF does not hold a privileged position among *Common Logic dialects* (3.7), as the expanded name suggests. Further, XCL is the recommended interchange format on the Web.

3.5
Conceptual Graph Interchange Format
CGIF
text version of *conceptual graphs* (3.2)

Note 1 to entry: Sometimes, this may refer to an example of a character string that conforms to Annex B, intended to convey exactly the same structure and semantics as an equivalent conceptual graph.

3.6
denotation
relationship holding between a name or expression and the thing to which it refers

Note 1 to entry: Also used with "of" to mean the thing being named, i.e. the referent of a name or expression.

3.7
Common Logic dialect
concrete instance of Common Logic syntax that shares (at least some of) the uniform semantics of Common Logic

Note 1 to entry: A dialect may be textual or graphical or possibly some other form. A dialect, by definition, is also a conforming language (see 7.1 for further details).

3.8
eXtensible Common Logic Markup Language
XCL
XML-based syntax for Common Logic

3.9
individual
<of an interpretation> one element of the *universe of discourse* (3.17) of an *interpretation* (3.12)

Note 1 to entry: The universe of discourse of an interpretation is the set of all of its individuals.

3.10
internationalized resource identifier
IRI
string of Unicode characters intended for use as an Internet network identifier syntax which can accommodate a wide variety of international character forms

3.11
inscription
structure of symbols that may be either linear or graphic

3.12
interpretation
formal specification of the meanings of the names in a vocabulary of a *Common Logic dialect* (3.7) in terms of a *universe of reference* (3.18)

Note 1 to entry: A Common Logic interpretation, in turn, determines the semantic values of all complex expressions of the dialect, in particular, the truth values of its *sentences* (3.15), statements and texts.

Note 2 to entry: See 6.2 for a more precise description of how an interpretation is defined.

3.13
operator
distinguished syntactic role played by a specified component within a functional *term* (3.16)

Note 1 to entry: The *denotation* (3.6) of a functional term in an *interpretation* (3.12) is determined by the functional extension of the denotation of the operator together with the denotations of the arguments.

3.14
predicate
<Common Logic> distinguished syntactic role played by exactly one component within a simple *sentence* (3.15)

Note 1 to entry: The truth value of a simple sentence in an *interpretation* (3.12) is determined by the relational extension of the *denotation* (3.6) of the predicate together with the denotations of the arguments.

3.15
sentence
<Common Logic> expression in the syntactic form of a traditional first-order logical formula

EXAMPLE A simple sentence (see 6.1.1.15), Boolean sentence (see 6.1.1.14) or quantification (see 6.1.1.13).

3.16
term
<Common Logic> expression which denotes an *individual* (3.9), consisting of either a name or, recursively, a functional term applied to a sequence of arguments, which are themselves terms

Note 1 to entry: Languages for traditional first-order logic specifically exclude *predicate* (3.14) quantifiers and the use of the same name in both predicate and argument position in simple *sentences* (3.15), both of which are permitted (though not required) in Common Logic. Languages for traditional first-order logic fall within the category of presupposing CL dialect, with a discourse presupposition of "non-discourse" for all names used as function *operators* (3.13) or predicates, and "discourse" for all names used as the arguments of functional terms or simple sentences or as bindings.

3.17
universe of discourse
domain of discourse
set of all the *individuals* (3.9) in an *interpretation* (3.12), i.e. the set over which the quantifiers range

Note 1 to entry: Required to be a subset of the *universe of reference* (3.18) and may be identical to it.

3.18
universe of reference
set of all the things needed to define the meanings of logical expressions in an *interpretation* (3.12)

Note 1 to entry: Required to be a superset of the *universe of discourse* (3.17) and may be identical to it.

4 Symbols and abbreviated terms

4.1 Symbols

fun_I	mapping from UR_I to functions from UD_I* to UD_I
I	interpretation in the model-theoretic sense
int_I	mapping from names in a vocabulary V to UR_I; informally, a means of associating names in V to referents in UR_I
rel_I	mapping from UR_I to subsets of UD_I*
seq_I	mapping from sequence markers in V to UD_I*
λ	lexicon, which consists of a vocabulary, a set of sequence markers (Smark), and a set of titles (Ttl)
V	vocabulary, which is a set of names
Smark	set of sequence markers
Ttl	set of titles
UD_I	universe of discourse; a non-empty set of individuals that an interpretation I is "about" and over which the quantifiers are understood to range
UR_I	universe of reference, i.e. the set of all referents of names in an interpretation I
X*	set of finite sequences of the elements of X, for any set X. Thus, $X^* = \{<x_1,...,x_n> \mid x_1,...,x_n \in X\}$, for any $n \geq 0$. Note that the empty sequence is in X*, for any X.

4.2 Abbreviated terms

CL	Common Logic
DF	display form
EBNF	Extended Backus-Naur Format (as in ISO/IEC 14977)
FO	first-order
KIF	Knowledge Interchange Format
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
TFOL	traditional first-order logic
XML	eXtensible Markup Language

5 Requirements and design overview

5.1 Requirements

5.1.1 Common Logic should include full first-order logic with equality

Common Logic abstract syntax and semantics shall provide for the full range of first-order syntactic forms with their usual meanings. Any traditional first-order syntax will be directly translatable into Common Logic without loss of information or alteration of meaning.

5.1.2 Common Logic should provide a general-purpose syntax for communicating logical expressions

a) There should be a single XML syntax for communicating Common Logic content on the Web.

b) Common Logic languages should be able to express various commonly used “syntactic sugarings” for logical forms or commonly used patterns of logical sentences.

c) The Common Logic abstract syntax should relate to existing conventions; in particular, it should be capable of rendering any content expressible in RDF, RDFS or OWL.

d) There should be at least one human-readable presentation syntax defined which can be used to express the entire language.

5.1.3 Common Logic should be easy and natural for use on the Web

a) The XML syntax should be compatible with the published specifications for XML, IRI syntax, XML Schema, Unicode and other conventions relevant to transmission of information on the Web.

b) IRIs should be usable as names in the language.

c) IRIs should be usable to title texts and label expressions, in order to facilitate Web operations such as retrieval, importation and cross-reference.

5.1.4 Common Logic should support open networks

a) Transmission of content between Common Logic-aware agents should not require negotiation about syntactic roles of symbols or translations between syntactic roles.

b) Any piece of Common Logic text should have the same meaning, and support the same entailments, everywhere on the network. Every name should have the same logical meaning at every node of the network.

c) No agent should be able to limit the ability of another agent to refer to anything or to make assertions about anything.

d) The language should support ways to refer to a local universe of discourse and be able to relate it to other such universes.

e) Users of Common Logic should be free to invent new names and use them in published Common Logic content.

5.1.5 Common Logic should not make arbitrary assumptions about semantics

a) Common Logic does not make gratuitous or arbitrary assumptions about logical relationships between different expressions.

b) If possible, Common Logic agents should express these assumptions in Common Logic directly.

5.2 A family of languages

This subclause describes what is meant by a "family" of languages and gives some of the rationale behind the development of Common Logic.

Following the convention whereby any language has a grammar, then Common Logic is a family of languages rather than a single language. Different Common Logic languages, referred to in this document as dialects, may differ sharply in their surface syntax, but they have a single uniform semantics and can all be transcribed into the common abstract syntax. Membership in the family is defined by being inter-translatable with the other dialects while preserving meaning, rather than by having any particular syntactic form. Several existing logical notations and languages, therefore, can be considered to be Common Logic dialects.

A Common Logic dialect called CLIF based on KIF (see Annex A) is used in giving examples throughout this document. CLIF can be considered an updated and simplified form of KIF 3.0[3] and hence a separate language in its own right. Conceptual graphs[1] are also a well-known form of first-order logic for machine processing; the CGIF language is specified in Annex B. An exactly conformant XML Dialect, called XCL, is specified in Annex C, for the purpose of satisfying requirements 5.1.2 a) and 5.1.3 a).

6 Common Logic abstract syntax and semantics

6.1 Common Logic abstract syntax

6.1.1 Abstract syntax categories

6.1.1.1 Terms, sequence markers, sentences, statements and texts are well-formed expressions.

6.1.1.2 A text is a text construction, domain restriction, or importation.

6.1.1.3 A text construction contains a set, list or bag of sentences, statements and/or texts. A Common Logic text may be a sequence, a set or a bag of sentences, statements and/or texts; dialects may specify which is intended or leave this undefined. Re-orderings and repetitions of arguments of a text construction are semantically irrelevant. However, applications which transmit or re-publish Common Logic text shall preserve the structure of text constructions, since other applications are allowed to utilize the structure for other purposes, such as indexing. If a dialect imposes conditions on text constructions, then these conditions shall be preserved by conforming applications. A text construction may be empty.

6.1.1.4 A domain restriction consists of a term and a text called the *body text*. The term indicates the "local" universe of discourse in which the text is understood.

6.1.1.5 An importation contains a title. The intention is that the title provides an identifier to an external Common Logic text, and that the importation re-asserts that external text in the importing text.

6.1.1.6 An axiom is a statement, sentence or text.

6.1.1.7 A statement is either a discourse statement or a titling.

6.1.1.8 A discourse statement is either an out-of-discourse statement or an in-discourse statement.

6.1.1.9 An out-of-discourse statement contains a sequence of terms.

6.1.1.10 An in-discourse statement contains a sequence of terms.

6.1.1.11 A titling contains a name and a text. The titling assigns a title to a text. Titles are often IRIs, which identify the text as resource.

6.1.1.12 A sentence is either a quantified sentence or a Boolean sentence or a simple sentence.

6.1.1.13 A quantified sentence has (i) a type, called a *quantifier*, (ii) a finite, nonrepeating sequence of interpretable names called the *binding sequence*, each element of which is called a *binding* of the quantified sentence, and (iii) a sentence called the *body* of the quantified sentence. The abstract syntax distinguishes the *universal* and the *existential* types of quantified sentence. A name which occurs in the binding sequence is said to be *bound in* the body. Any name or sequence marker which is not bound in the body is said to be *free in* the body.

6.1.1.14 A Boolean sentence has a type, called a *connective*, and a number of sentences called the *components* of the Boolean sentence. The number depends on the particular type. The abstract syntax distinguishes five types of Boolean sentences: *conjunctions* and *disjunctions,* which have any number of components, *implications* and *biconditionals*, which have exactly two components, and *negations*, which have exactly one component. The two components of an implication fill different roles; one is the antecedent and the other is the consequent.

6.1.1.15 A simple sentence is either an *equation* containing two *arguments*, which are terms, or is an atomic sentence, which consists of a term, called the *predicate*, and a term sequence called the *argument sequence*, the elements of which are called *arguments* of the atomic sentence.

6.1.1.16 A term is either a name or a functional term. A term may have an attached comment. Further, every name is a term.

6.1.1.17 A functional term consists of a term, called the *operator*, and a term sequence called the *argument sequence*, the elements of which are called *arguments* of the functional term.

6.1.1.18 A term sequence is a finite sequence of terms and/or sequence markers. Term sequences may be empty, but a functional term with an empty argument sequence shall not be identified with its operator, and an atomic sentence with an empty argument sequence shall not be identified with its predicate.

6.1.1.19 A *lexicon* is a set of names (i.e. the vocabulary of the lexicon), a set of sequence markers, and a set of titles.

6.1.1.20 Irregular sentences in a concrete syntax are parsed into the abstract syntax as propositions (i.e. nullary atomic sentences) with a new name for the predicate. In this way, irregular sentences can be nested within texts, statement and (otherwise) regular compound sentences, and the semantics of the resulting expressions is determined as usual from Table 2.

6.1.1.21 A comment is a piece of data. Any number of comments may be attached to well-formed expressions that are texts, statements, sentences or functional terms, but not to names, sequence markers or other comments. No particular restrictions are placed on the nature of Common Logic comments; in particular, a comment may be Common Logic text. Particular dialects may impose conditions on the form of comments.

6.1 completely describes the abstract syntactic structure of Common Logic. Any fully conformant Common Logic dialect **shall** provide an unambiguous syntactic representation for each of the above types of well-formed expressions.

Sentence types are commonly indicated by the inclusion of explicit text strings, such as “forall” for universal sentence and “and” for conjunction. However, no conditions are imposed on how the various syntactic categories are represented in the surface forms of a dialect. In particular, expressions in a dialect are not required to consist of character strings.

6.1.2 Metamodel of the Common Logic abstract syntax

6.1.2.1 Names and sequence markers

The class of names and sequence markers of a Common Logic language is the classes obtained from strings using the following operators:

— *Voc* : *String* → *V*

— *Seqmark* : *String* → *Smark*

— *Titling* : *String* → *Ttl*

— *Binder* = *V* ∪ *Smark*

6.1.2.2 Terms and term sequences

The class of terms of a Common Logic language is the class Term that includes all names in V and all functional terms. The class of functional terms of a Common Logic language is the class FunctionalTerm obtained by the recursive application of the operator *Func* to pairs made up of one term and one term sequence. The class of term sequences of a Common Logic language is the class TermSequence that includes all finite sequences of terms and/or sequence markers.

— *Func:Term* × *TermSequence* → *FunctionalTerm*

— *Term* = *V* ∪ *FunctionalTerm*

— *TSeq:*(*Term* ∪ *Smark*) × ... × (*Term* ∪ *Smark*) → *TermSequence*

6.1.2.3 Sentences

The class of sentences of a Common Logic language is the class Sentence that includes all simple sentences (including equations, if applicable) formed by the application of the operation(s) Atomic (and Id) from well-formed terms and term sequences and all compound sentences formed by the recursive application of the set of operations Neg, Conj, Disj, Cond, BiCond, EQuant, and UQuant that satisfy the following conditions:

— each operation is one-to-one;

— the range of the operations is pairwise disjoint and disjoint from the set of terms of λ.

— $\mathit{Atomic}:\mathit{Term} \times \mathit{TermSequence} \rightarrow \mathit{Sentence}$

— $\mathit{Id}:\mathit{Term} \times \mathit{Term} \rightarrow \mathit{Sentence}$

— $\mathit{Neg}:\mathit{Sentence} \rightarrow \mathit{Sentence}$

— $\mathit{Conj} : \mathit{Sentence} \times \ldots \times \mathit{Sentence} \rightarrow \mathit{Sentence}$

— $\mathit{Disj} : \mathit{Sentence} \times \ldots \times \mathit{Sentence} \rightarrow \mathit{Sentence}$

— $\mathit{Cond} : \mathit{Sentence} \times \mathit{Sentence} \rightarrow \mathit{Sentence}$

— $\mathit{BiCond}:\mathit{Sentence} \times \mathit{Sentence} \rightarrow \mathit{Sentence}$

— $\mathit{EQuant}:\mathit{Binder} \times \ldots \times \mathit{Binder} \times \mathit{Sentence} \rightarrow \mathit{Sentence}, n \geq 0$

— $\mathit{UQuant}:\mathit{Binder} \times \ldots \times \mathit{Binder} \times \mathit{Sentence} \rightarrow \mathit{Sentence}, n \geq 0$

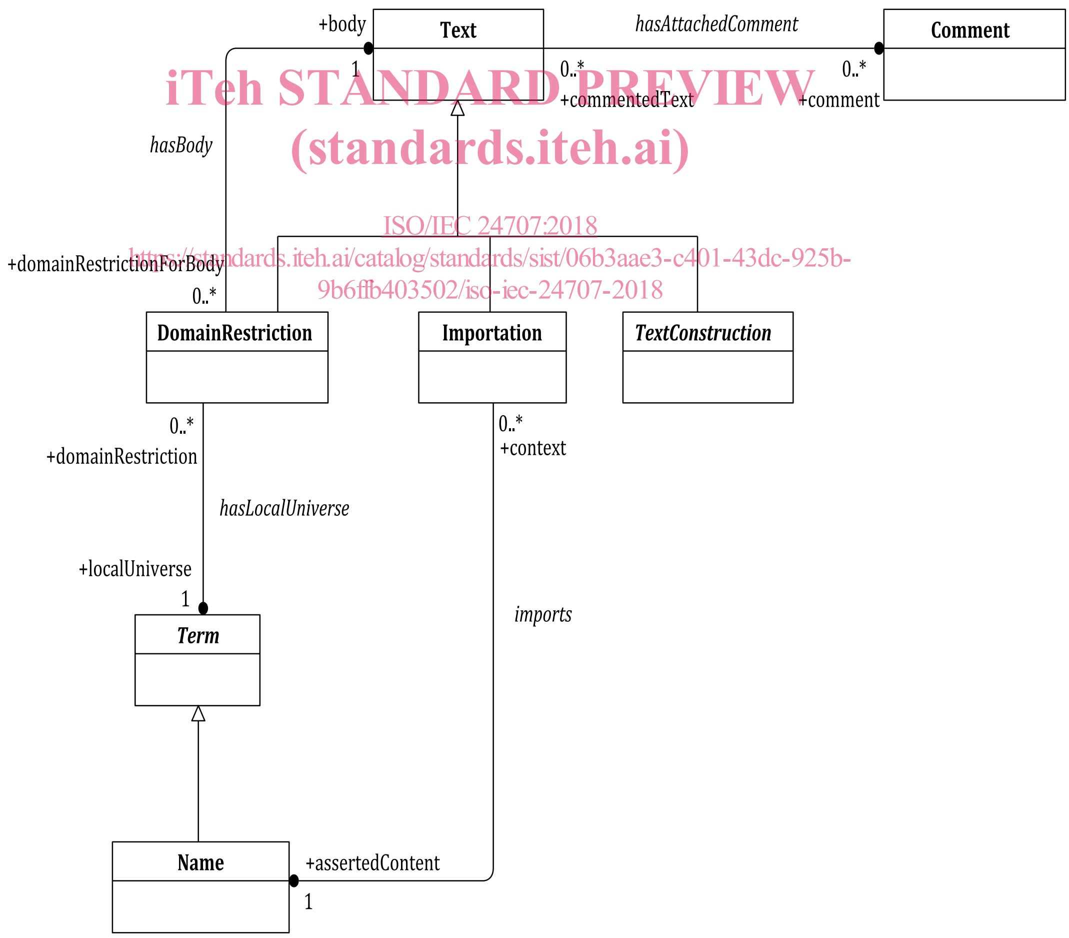

Figure 1 — Abstract syntax of texts

