Importation Closure that is Robust to Circular Dependencies

Tara Athan[1]

Athan Services, W Lafayette, IN, USA
taraathan@gmail.com
WWW home page: http://athant.com

Abstract. Any approach to the integration of distributed knowledge bases (KBs) through importation must make design decisions regarding circular importation dependencies: are they disallowed, ignored, or allowed? If they are allowed, how is the resulting infinite loop of importation handled in practice? We present preliminary results of Importations (http://athant.com/projects/Importation/), an exploration of the ramifications of embracing circular importation dependencies, with particular application to design decisions in the revision of ISO Common Logic (CL).

1 Introduction

The distributed, modular representation of knowledge (rules, ontologies, facts) in widely-available form such as on the Web is becoming increasingly common. Correspondingly, there is a growing need to merge, or import, such modules (in CL called "texts") to create more comprehensive KBs. As authorship and maintenance becomes more distributed and less-closely coordinated, the likelihood of circular importation dependencies among texts increases.

While intentional circular dependencies are not a recommended practice in KB engineering, a robust reasoning engine should be able to recover gracefully from such occurrences. Languages may even be designed to provide unambiguous semantics for these situations. In particular, the CL community has recently adressed this issue[2] in regard to the proposed revision of the ISO CL[1].

This study is intended to provide further input regarding the ramifications of design decisions for the syntax and semantics of importation. Practical concerns include:

- Minimizing the burden involved in syntactic validation of texts with importations.
- Avoiding ambiguity in the semantics of texts with importations.
- Avoiding semantics that is non-intuitive.
- Ensuring that the set of expressions that must be handled for practical purposes, such as reasoning, remains finite.
- Minimizing the computational effort of resolving importations.

A typical practice in handling circular importation dependencies is to simply ignore an importation the second time it occurs, and in many settings this is a safe strategy that minimizes the effort of both syntactic validation and semantic evaluation and it

keeps things finite. However, in some applications the importation of the same text in different contexts creates different semantic effects. The result might depend on which importation was performed first, leading to ambiguity.

ISO CL is an example of a practical case where the importation of the same text in different contexts can create different semantic effects. One of the aims of CL is to allow the integration of segregated texts, e.g. where the vocabularies for individuals, functions and relations are disjoint, and nonsegregated texts with a so-called "higher-order syntax", where the same name may be used to denote an individual and refer to a function and/or relation.

CL handles the integration of such disparate representations formally through a modification of the interpretation that is applied to an imported text. The modification, called restriction, has the effect of shrinking (i.e. restricting) the domain of discourse of the interpretation so that quantification occurs only over the intended set of entities. Such a restriction can have different semantic effects, depending on the composition of the restricted domain of discourse. Therefore, ignoring a second importation of a text may not produce the desired semantics.

It has been shown[3] that domain restriction of a text, in the particular language studied there, is logically equivalent to a well-defined rewriting of the text, including modifying the sentences inside quantifications as well as adding domain membership facts to avoid *free logic* behaviours. Such a rewriting can be considered a special case of a more general "text operator". Other examples of text operators include a translation operator to convert a text from one KR language into another.

1.1 Propositions and Some Weird Things

The language that is created here to study circular importation dependencies contains the following components

- propositional statements, which encapsulate a text in an arbitrary knowledge representation (KR) language;
- text operators, as a model for CL domain restriction as well as translations;
- titling and importation statements modelled after the abstract semantics proposed for the CL revision[3];
- text construction statements, which form collections of texts.

Importation is handled at a metalogical level and it never mingles with the underlying logical language. For example, we do not allow conditional importation that depends on the interpretation of some logical sentence. Because of this, we may defer all logical considerations until after the resolution of importations, and this is why there is no need to include, e.g., logical connectives in the language.

2 Background

A semantics for importations of distributed KBs was recently proposed for CL in [2]. Two statements are used to implement importation, a titling statement, which assigns

a name from the vocabulary as the title of a text, and an importation statement, which invokes a text title.

The proposed CL semantics includes a relationship, called the title mapping, as a part of an interpretation. A titling statement with a name and text is true iff the image of that name under the titling mapping is indeed that text, verbatim.

The proposed CL semantics for importation statements is based on the concept of a corpus - a (possibly empty, finite or infinite) set of finite texts. The importation statement itself always interprets to true, but its semantic effect is to add a new text to the set of texts that must interpret to true if the corpus is to be considered satisfied by an interpretation. Intuititively, a set of new texts is generated from the old by replacing importation statements[1] that do not occur within a titling statement by their images under the title mapping, a process called *resolution*. These new texts are added to the corpus. Then if any of the new texts contain importation statements, the resolution process is repeated. At each resolution step a finite number of (finite) texts is added to the corpus, but the texts already in the corpus are not deleted or changed in any way. Thus importation resolution is a monotonic process.

In this approach, importation statements may occur within titled texts, and it is permitted to have circular importation dependencies, the simplest being a text that imports itself. However, if the importation dependencies are circular, then the importation closure process as defined above does not terminate in any finite number of steps.

The precise definition in [2] of the importation closure is the minimal fixed point of this corpus-extension procedure that contains the original corpus. It was shown there that such a minimal fixed point of such an importation closure exists for any interpretation. Although the proof was carried out for the language considered there, $\mathcal{L}$, a simplified language inspired by the CL abstract syntax and semantics, and the particular CL dialect called CLIF[1], the same proof applies to the even simpler language we consider below.

Note that the importation closure depends, in general, on the intepretation because the text susbstituted for an importation statement is determined by the title mapping of the interpretation, not the titling statements in the corpus. The connection between title mapping and titling statements is established semantically

In comparison to importation, titling and reification mechanisms of other KR languages:

- unlike the OWL importation mechanism, resolution of importation in our approach does not syntactically affect the text in which it appears. Also it does not introduce a “text” or “ontology” entity into the universe of reference.
- unlike the current CL semantics, our titling approach does not affect the denotation of the name that is used as a title. By analogy consider that ’Wherever You Go, There You Are.’ is the title of a book, as well as the name of a proposition. Similarly, names in our vocabulary have dual, independent use as titles and as names of propositions.

[1] Alternative formulations of this transformation are (1) replace all importations in a text in the corpus simultaneously (2) replace each importation individually, leaving the others as they are, producing, in general, a set of new texts for each text resolved.

We first define the syntax, then the semantics of a family $\mathcal{L}_0$ of languages. Some examples of circular importation dependencies are presented, and we then examine conditions under which a finite corpus will always be logically equivalent to some finite corpus having a certain closure property that will allow us to use that finite corpus for reasoning purposes.

3 Syntax of $\mathcal{L}_0$

3.1 Lexical items

An $\mathcal{L}_0$ language consists of the following lexical items:

- Logical lexical items: the text construction operator 'txt'; a set of text operators 'F_i', $i \in \mathcal{I}$; the titling operator '*title* '; the importation operator '*import* ';
- Auxiliary lexical items: the parentheses: '(', ')'; the Unicode SPACE character (U+0200).
- Names: a set V of strings of unicode text characters (i.e., no whitespace) other than those above.

3.2 Grammar

An *expression* of $\mathcal{L}_0$ is any (possibly empty) string of lexical items of $\mathcal{L}_0$. The basic syntactic categories of expressions, which are partially overlapping, are *names*, *propositions*, *statements*, and *texts*. Names were defined in the previous section 3.1.

We now define all other syntactic categories, including the subcategories of *import statement*, *text titling*, and *text constructions*, of $\mathcal{L}_0$ simultaneously[2]:

We will use N for names, S for statements, Γ, Δ for texts, and E for arbitrary expressions (with and without indices).

- A propositional statement is a name, N enclosed in parentheses (N).
- A statement is either a propositional statement, an import statement, or a text titling statement.
- A text is either
  - a statement,
  - a text construction $(txt\ \Gamma_1 \ldots \Gamma_n)$, where $\Gamma_1, \ldots, \Gamma_n$ $(0 \leq n)$ are texts , or
  - a text operator applied to a text $(F_i\ \Gamma)$ where Γ is a text.
- A text titling is $(\textit{title}\ N\ \Gamma)$ where Γ is a text and N is a name.
- An import statement is $(\textit{import}\ \ N)$ where N is a name.

A title mapping ttl is a total function from V of $\mathcal{L}_0$ to the set of texts of $\mathcal{L}_0$.

A corpus of a $\mathcal{L}_0$ language is a set of texts in that language. A corpus may be empty, finite or infinite. Although it is not necessary, and in some cases not possible, to serialize a corpus, for the examples in this paper we use the notation $\{\Gamma_1\Gamma_2...\Gamma_n\}$ for a finite corpus, where Γ_i are texts, and the texts are separated by white space and enclosed in curly braces, which is similar to standard mathematical set notation. The curly braces are not part of $\mathcal{L}_0$.

[2] a LISP-like Cambridge Polish prefix notation is used for operator application

4 Semantics of $\mathcal{L}_0$

The semantics of languges in $\mathcal{L}_0$ are described, but not specified, in this section. In particular, no specific connection is established between a text and the result of applying a text operator to that text, or the interpretation of that resulting text.

An *interpretation* I consists of

- a subset TP_I of V;
- a subset TX_I of text expressions;
- a title mapping ttl_I.

Let I be an interpretation. Let E be an expression of $\mathcal{L}$. *The interpretation of* E *under* I, $I(E)$, is defined in Table 1.

Table 1. $\mathcal{L}_0$ Semantics

	If E is	Then $I(E)$ =
R1	a proposition: (N)	true if $N \in TP_I$; otherwise false
R2	a text construction: $(txt \;\; E_1 \ldots E_n)$	true if $(txt \;\; E_1 \ldots E_n) \in TX_I$; otherwise false
R3	an application of a text operator: $(F_i \; \Gamma)$	true if $(F_i \; \Gamma) \in TX_I$; otherwise false
R4	a text titling $(title \;\; N \; \Gamma)$	true if $ttl_I(N) = \Gamma$; otherwise false
R5	an import statement $(import \;\; N)$	is always true

The importation closure $\mathcal{C}'_I$ of a corpus $\mathcal{C}$ under a particular interpretation I is defined as discussed in the previous section 2. In general, the importation closures of a text of $\mathcal{L}_0$ with circular importation dependencies will not be finite.

A corpus $\mathcal{C}$ is satisfied by an interpretation I if for every text Γ in the importation closure $\mathcal{C}'_I$ of the text under the interpretation I, $I(\Gamma)$ is true. When I satisfies $\mathcal{C}$, we write $I(\mathcal{C})$ is true; otherwise, $I(\mathcal{C})$ is false. A model M of a corpus $\mathcal{C}$ is a satisfying interpretation ($M(\mathcal{C}) =$ true).

One corpus $\mathcal{C}_1$ entails another corpus $\mathcal{C}_2$ if the latter is satisfied whenever the former is satisfied ($I(\mathcal{C}_2)$ is true whenever $I(\mathcal{C}_1)$ is true). Two corpora are logically equivalent if they entail each other ($I(\mathcal{C}_1) = I(\mathcal{C}_2)$ for all I).

Note that if two interpretations I and J have the same title mapping $I = J$, then the importation closures of a particular corpus $\mathcal{C}$ are the same, $\mathcal{C}'_I = \mathcal{C}'_J$. However, the specification above is somewhat too weak to draw conclusions, and we will need to further restrict the semantics of titling.

It is expected that any actual KR language that adopts this importation mechanism will have a distinct, and typically compositional, specification of semantics, and such semantics will embue the language with certain algebraic properties. In the following subsections, esp. 4.1, 4.5, we develop algebraic properties that we assume to hold for all $\mathcal{L}_0$ languages.

4.1 Titling Semantics

We suppose that the title mapping affects only the semantics of texts containing titling statements. In particular, we suppose that individual titling statements may be extracted from text operators and text construction statements as follows:

A corpus
$\{(\ldots(txt\ldots(\textit{title}\ N\ \Gamma)\ldots)\ldots)\}$
is fully-equivalent to the corpus
$\{(\ldots(txt\ \ldots\ldots))$
$(\ldots(\textit{title}\ N\ \Gamma)\ldots)\}$
Similarly, a corpus
$\{(\ldots(\text{foo}\ldots(\textit{title}\ N\ \Gamma)\ldots)\ldots)\}$
is equivalent to the corpus
$\{(\ldots(\text{foo}\ldots\ldots))$
$(\ldots(\text{foo}(\textit{title}\ N\ \Gamma))\ldots)\}$
Therefore, it is sufficient to consider texts of only two kinds: (type I) those that contain no titling statements, and (type II) those that contain only individual titling statements, nested within some finite number of text operators and (unary) text constructions. We call the set of all type I(II) texts in a particular language $\mathcal{T}_I$ ($\mathcal{T}_{II}$).[3]

When an importation of a type II text into a type I text is resolved, the titling statement may be extracted as above, giving a new type II text while leaving the original type I text unchanged except for the removal of the importation statement. Note that the interpretation $I(S)$ of a type I statement S is independent of the title mapping ttl_I of the interpretation.

A titling-model of a corpus C is an interpretation J that satisfies (at least) the type II statements of the importation closure C'_J. A corpus is *title-satisfiable* if it has at least one titling-model.

4.2 Self-Contained Corpora

If a corpus contains an import statement, say for name N, and does not have a corresponding titling statement to restrain the identity mapping of N, it may still be possible to draw conclusions based on the explicit statements and importations that are well-defined. However, the algorithms for addressing these cases become more complex, so for this study we wish to distinguish between these cases. To this end, we introduce a property of "self-containment", indicating a corpus contains enough titling statements to uniquely define all the importations it "needs".

Ideally, self-containment would be syntactically defined. However, the best we can manage at this point is a semantic definition of self-containment because of our non-compositional semantics.

Let a corpus $\mathcal{C}$ be semantically self-contained iff

- $\mathcal{C}$ is title-satisfiable;

[3] The algebraic conditions on type II texts are restated more precisely in different notation in subsection 4.5.

– there exists a unique importation closure $\hat{\mathcal{C}'}$ (called the *canonical importation closure*) for all titling-models.

The uniqueness of the r closure arises from the presence of "enough" titling statements - too few titling statements leads to multiple importation closures for titling-models, while its existence arises from the absence of "too many" titling statements, esp. contradictory ones that lead to lack of title-satisfiability.

4.3 Cover

When corpora have infinite importation closures, practical reasoning is only possible when entailments can be determined from some finite set of texts. To this end, we define the concept of a *cover*.

Let corpus $\tilde{\mathcal{C}}$ be a cover of a corpus $\mathcal{C}$ iff

– $\tilde{\mathcal{C}}$ is logically equivalent to $\mathcal{C}$ and
– for every interpretation I, $\tilde{\mathcal{C}}$ is satisfied by I iff $I(\Gamma)$ is true for every text $\Gamma \in \tilde{\mathcal{C}}$.

In other words, a cover captures the semantics of the corpus in a different, and potentially smaller, package than the importation closure. Reasoning may be carried out practically using covers, provided they exist and are finite for the corpora of interest.

The second item in the definition of a cover makes use of a compositional truth value of a corpus, without considering the importation closure. We now define a notation for this.

Given any interpretation I of $\mathcal{L}_0$, there is a mapping I_S, which we will call the superficial interpretation mapping, from corpora of $\mathcal{L}_0$ into truth values such that $I_S(\mathcal{C})$ is true iff $I(\Gamma)$ is true for every text Γ in $\mathcal{C}$.

The importation closure of a corpus contains the original corpus. Therefore, a superficial interpretation $I_S(\mathcal{C})$ of true is necessary, but not sufficient, for an interpretation I to satisfy a corpus $\mathcal{C}$. In this notation, the satisfaction condition for all corpora may be stated as follows.

Given any interpretation I of $\mathcal{L}_0$ and any corpus $\mathcal{C}$ of $\mathcal{L}_0$, then $I(\mathcal{C}) = I_S(\mathcal{C}')$ where C'_I is the importation closure of $\mathcal{C}$ under I.

If $\tilde{\mathcal{C}}$ is a cover of $corp$, the second condition of the definition implies $I(\tilde{\mathcal{C}}) = I_S(\tilde{\mathcal{C}})$, while the first condition implies $I(\tilde{\mathcal{C}}) = I(\mathcal{C})$, and so we have an equivalent statement of cover:

A corpus $\tilde{\mathcal{C}}$ is a cover of a corpus $\mathcal{C}$ iff for every interpretation I,
$I(\mathcal{C}) = I(\tilde{\mathcal{C}}) = I_S(\tilde{\mathcal{C}})$

In order to develop algorithms for determining a cover of a corpus, we need to know what are the characteristics of covers. In particular, we note that the second condition in the definition of a cover is a property independent of the corpus the cover "covers". Because any cover is a cover of itself, being a cover (of something) is equivalent to meeting the condition
$I(\tilde{\mathcal{C}}) = I_S(\tilde{\mathcal{C}})$

We now show that being a cover is equivalent to the condition: for every non-satisfying interpretation I, there is a text $\gamma_I \in \mathcal{C}$ such that $I(\gamma_I)$ is false.

First we consider that case that $I(\mathcal{C})$ is true. Then by definition, any text Γ in $\mathcal{C}'_I$, $I(\Gamma)$ is true. But $\mathcal{C}'_I$ contains $\mathcal{C}$, therefore $I_S(\mathcal{C})$ is true, and hence $I(\mathcal{C}) = I_S(\mathcal{C})$. This holds for any corpus, not just covers.

Now suppose that $I(\mathcal{C})$ is false, i.e., I doesn't satisfy $\mathcal{C}$. If the condition above is satisfied, then there is some text Γ in $\mathcal{C}$ such that $I(\Gamma)$ is false, and thus $I(\mathcal{C})$ is false and hence $I(\mathcal{C}) = I_S(\mathcal{C})$. Conversely, suppose $\mathcal{C}$ is a cover. Then $I_S(\mathcal{C})$ is false, which can only be the case if there exists some text Γ in $\mathcal{C}$ such that $I(\Gamma)$ is false.

Our basic goals for this paper are

- to determine conditions on a $\mathcal{L}_0$ language such that for any finite corpus, even one with an infinite importation closure, there exists a finite cover;
- to describe an algorithm for finding a cover of a corpus for languages that satisfy these conditions.

We will accomplish this with the help of some semantics-preserving transformations, as defined in the next section.

4.4 Semantics Preserving Transforations

We first define a variant form of logical equivalence of texts, which we call superficial-logical-equivalence (SL-equivalence or SLE).

With respect to a given $\mathcal{L}_0$ language, let

- two texts Γ, Δ be *SL-equivalent* ($\Gamma \equiv_{SL} \Delta$) iff $I(\Gamma) = I(\Delta)$ for every interpretation I;
- two corpora $\mathcal{C}_1$, $\mathcal{C}_2$ be *SL-equivalent* ($\mathcal{C}_1 \equiv_{SL} \mathcal{C}_2$) iff $I_S(\mathcal{C}_1) = I_S(\mathcal{C}_2)$ for every interpretation I;

It is noteworthy that logical equivalence and SL-equivalence are not comparable characteristics; there exists pairs of texts that are logically equivalent but not SL-equivalent and v.v. (see Example 5.1. A stronger equivalence relation between texts can be defined by combining the two: a pair of texts (or corpora) Γ , Δ are fully equivalent if they are both logically and SL-equivalent, and we write $\Gamma \equiv_F \Delta$.

A corpus $\mathcal{C}$ containing two fully-equivalent texts Γ, Δ is fully-equivalent to the corpus $\bar{\mathcal{C}}$ obtained from $\mathcal{C}$ by deleting Δ.

A text transformation that alwys generates an ouput text that is fully equivalent to the input text is called a semantics-preserving transformation.

4.5 Algebra-like Structures of the Syntax

Corpora are sets, and so are subject to the binary operations of union and intersection with their usual properties. The set of all corpora in a $\mathcal{L}_0$ language is, naturally, closed under these two operations. There are four equivalence relations to consider: syntactic equality, and superficial-, logical- and full-equivalence, and there is a set of equivalence classes of corpora for each of the latter three relations. There is a set of operators on corpora corresponding to importation closure under a particular interpretation of a $\mathcal{L}_0$ language. These operators are also well-defined on logical and full equivalence classes

of corpora. The subset of corpora that are self-contained also has a unique operator corresponding to importation closure under any titling-model. Because the semantics of corpora are defined compositionally in terms of the interpretation of texts (of the importation closure), certain properties hold for all $\mathcal{L}_0$ languages; e.g., a corpus entails its subsets, the union of unsatisfiable corpora is unsatisfiable, and so on.

At a more fine-grained level, we may examine algebraic-like structures involving texts. We may consider individual texts, or ordered tuples of texts and the same four equivalence relations apply as were considered for corpora. Concatenation is a binary operator acting on tuples, and text construction is a polyadic operator (which we will denote by Q) on texts ($Q : \mathcal{T}^* \rightarrow \mathcal{T}$) and a unary operator on text tuples. The semantics of text construction is not defined compositionally, so for general $\mathcal{L}_0$ we cannot assume algebraic properties of text construction, such as commutativity, associativity or distributivity of construction over concatenation. We will consider subsets of the $\mathcal{L}_0$ languages that have certain algebraic properties.

The most restrictive assumption that could be made about the semantics of text construction is

$\{Q(\Gamma_1, \ldots, \Gamma_n)\} \equiv_F \{\Gamma_1 \ \ldots \ \Gamma_n\}$

where $\Gamma_i \in \mathcal{T}^*, 1 \leq i \leq n$, which would make text construction equivalent to corpus construction, giving text construction a semantics similar to polyadic conjunction.

Let $\mathcal{L}_0^+$ be the subset of $\mathcal{L}_0$ languages that satisfy this *conjunctive-text-construction* property. In $\mathcal{L}_0^+$, the set of full-equivalence classes of texts is an Abelian algebra loop (closed, associative, commutative, with identity element) under Q as a binary operation.[4]

The minimal assumption that is imposed on all $\mathcal{L}_0$ languages, as described above in subsection 4.1, for all $\mathcal{L}_0$ confers on text construction a semantics like polyadic conjunction only on type II texts, and also has the semantics similar to binary conjunction when a type II text embedded in a tuple of arbitrary texts. In particular, for all $\mathcal{L}_0$ languages,

$\{Q(\Gamma_1, \ldots, \Gamma_n)\} \equiv_F \{\Gamma_1 \ \ldots \ \Gamma_n\}$

where $\Gamma_i \in \mathcal{T}_{II}^*, 1 \leq i \leq n$ and

$\{Q(\ldots \Gamma_1, \Delta, \ldots \Gamma_2))\} \equiv_F \{\Delta, Q(\ldots \Gamma_1, \ldots \Gamma_2)\}$

where $\Delta \in \mathcal{T}_{II}$ and $\ldots \Gamma_i \in \mathcal{T}^*, i = 1, 2$.

Similarly to text construction, the semantics of text operators, $F \in \mathcal{F}$, $F : \mathcal{T} \rightarrow \mathcal{T}$, is not defined compositionally. To achieve the separability of type I and type II texts, it is imposed above in subsection 4.1, for all $\mathcal{L}_0$ languages, that text operators are distributive over polyadic text construction on type II texts, as well as a type II text embedded in a text tuple. That is, for all $\mathcal{L}_0$ languages,

$F(Q(\ldots \Gamma_1, \Delta, \ldots \Gamma_2)) \equiv_F Q(F(\Delta), F(Q(\ldots \Gamma_1, \ldots \Gamma_2)))$

where $F \in \mathcal{F}$, $\Delta \in \mathcal{T}_{II}$ and $\ldots \Gamma_i \in \mathcal{T}^*, i = 1, 2$.

We let $\mathcal{L}_0^{\Omega}$ be the subset of $\mathcal{L}_0$ languages where text operator distributivity holds over all texts. That is,

$F(Q(\Gamma_1, \ldots, \Gamma_n)) \equiv_F Q(F(\Gamma_1), \ldots \ F(\Gamma_n))$

where $F \in \mathcal{F}$, and $\Gamma_i \in \mathcal{T}$, $1 \leq i \leq n$.

[4] Polyadic text construction is a shortcut for repitition of the binary construction in this case.

Further let $\mathcal{L}_0^{\&\Omega} = \mathcal{L}_0^+ \cap \mathcal{L}_0^\Omega$; in $\mathcal{L}_0^{\&\Omega}$ languages, the set of full-equivalence classes of texts could be considered an Abelian algebra loop with distributive operators (Abelian Ω-loop), a generalization of the well-studied algebraic structure "group with operators" or Ω-group.

5 Examples

The following examples correspond to existing[2] as well as new Test Cases for the resolution of circular importation dependencies.

5.1 Noncomparability of Logical- and SL-Equivalence

In $\mathcal{L}_0^+$, the texts $(txt\ (\textit{import}\ \text{foo}))$ and $(txt\ )$ are not logically equivalent, but they are SL-equivalent. On the other hand, the texts

$(txt\ (\textit{title}\ \text{foo}\ (A))\ (\textit{import}\ \text{foo}))$

$(txt\ (\textit{title}\ \text{foo}\ (A))\ (A))$

are logically equivalent in $\mathcal{L}_0^+$ but not SL-equivalent. This demonstrates that the concepts of logical and SL-equivalence are not comparable, in general.

5.2 No Cover if Not Self-Contained

Texts that are not semantically self-contained will, in general, have no cover (finite or otherwise). For example, the corpus

$\{(txt\ \textit{import}\ \text{foo})\}$

has no cover. The importation closure is still defined for every interpretation, but its content is determined by the image of "foo" in the title mapping of the interpretation, and this can vary arbitrarily.

5.3 Simple Circular Importation

Consider a self-contained corpus consisting of two texts

$\{(\textit{title}\ \text{foo}\ (F_0\ (\textit{import}\ \text{foo})))$

$(\textit{import}\ \text{foo})\}$

The importation closure contains

$(F_0(\textit{import}\ \text{foo}))$

$(F_0(F_0(\textit{import}\ \text{foo})))$

$(F_0(F_0(F_0(\textit{import}\ \text{foo}))))$

$\ldots$

as well as the original texts. Similar examples can be constructed with three or more texts where multiple operators are composed as follows:.

$\{(\textit{title}\ \text{foo}\ (F_0\ (\textit{import}\ \text{bar})))$

$\{(\textit{title}\ \text{bar}\ (F_0\ (\textit{import}\ \text{foo})))$

$(\textit{import}\ \text{foo})\}$

This example demonstrates that circular importation generates an infinite importation closure from a finite corpus.

Examining the results of this example from the perspective of algebra-like structures, we focus on the interaction of elements of $\mathcal{F}'_F$, full-equivalence classes on the closure of the set of text operators $\mathcal{F}$ under the composition operation. Text operators F_0, F_1 are fully-equivalent iff

$(F_0\ \Gamma) \equiv_F (F_1\ \Gamma)$ for every $\Gamma \in \mathcal{T}$.

We can achieve a finite cover for the corpora in this example if the closure under composition of a finite set of $\mathcal{F}'_F$ is finite and is itself a subset of $\mathcal{F}$, a kind of compactness property. Let $\bar{\mathcal{L}}_0$ denote the subset of $\mathcal{L}_0$ languages that satisfy this *compositional-compactness* property.

5.4 Concatenation of Multiple Operators appied to a Circular Import

A more complex circular importation dependency is demonstrated here:

$\{$(*title* foo $(txt(F_0\ ($*import* foo$))\ (F_1\ ($*import* foo$))\ (F_2\ ($*import* foo$))))$

(*import* foo$)\}$

This example demonstrates the need for distributivity of text operators over text construction ($\mathcal{L}_0^{\Omega}$) if we are to make progress on determining a finite cover for such a corpus. Assuming distributivity of the text operators over text construction, the importation resolution generates arbitrary finite compositions of the text operators F_0, F_1 and F_2. If the language has the compositional-compactness property, each member of this infinite set of operator compositions is fully-equivalent to some member of a finite subset of $\mathcal{F}'$, and thus a finite cover of the original corpus may be obtained.

5.5 Concatenation of Proposition and Operator applied to a Circular Import

Consider the corpus consisting of two texts

$\{$(*title* foo $(txt\ (A)\ (F_0\ ($*import* foo$))))$

(*import* foo$)\}$

Importation resolution yields

$(txt\ (A)\ (F_0\ ($*import* foo$)))$

$(txt\ (A)\ (F_0\ (txt\ (A)\ (F_0\ ($*import* foo$)))))$

$\ldots$

We again make use of the distributive property of text operators over text construction:

$(txt\ (A)\ (txt\ (F_0\ (A))\ (F_0\ (F_0\ ($*import* foo$)))$

Further resolution of importation produces

$(txt\ (A)\ (txt\ (F_0\ (A))\ (txt\ (F_0\ (F_0\ (A)))\ (F_0\ (F_0\ (F_0\ ($*import* foo$)))))))$

$\ldots$

We may take advantage of the compositional-compactness property to rewrite this as

$(txt\ (A)\ (txt\ (F_0\ (A))\ (txt\ (F_1\ (A))\ (F_2\ ($*import* foo$)))))$

However, in the absence of at least some aspects of conjunctive semantics for text construction there is little more that can be done towards deriving a finite cover.

Let us suppose that text constructions can be flattened,

$Q(\ldots \Gamma_1, Q(\ldots \Gamma_2), \ldots \Gamma_3) \equiv_F Q(\ldots \Gamma_1, \ldots \Gamma_2, \ldots \Gamma_3)$

for all $\Gamma_i \in \mathcal{F}, 1 \leq i \leq n$. Then the text above may be transformed to the fully-equivalent

$(txt\ (A)\,(F_0\,(A))\,(F_1\,(A))\,(F_2\,(\textit{import}\ \text{foo})))$.
Further suppose that the semantics of text construction is independent of the order of texts or duplication of texts in the tuple. Then application of compositional-compactness implies that there are a finite number of full-equivalence classes for texts of this form, given that the F_i are generated by composition of F_0 with itself.

6 Summary of Conditions for Applicability of Cover Determination Algorithm

Let $\bar{\mathcal{L}}_0^{+\Omega} = \bar{\mathcal{L}}_0 \cap \mathcal{L}_0^+ \cap \mathcal{L}_0^\Omega$ be the subset of $\mathcal{L}_0$ with the following properties.[5]

- ($\bar{\mathcal{L}}_0$) The language is compositionally-compact: the closure under composition of a finite subset of $\mathcal{F}'_F$ is finite and is itself a subset of $\mathcal{F}_F$, where $\mathcal{F}'_F$ is the closure, under composition, of the set $\mathcal{F}_F$ of fully-equivalent classes of text operators ($\mathcal{F}$) in the language;
- ($\mathcal{L}_0^\Omega$) Text operators distribute over text construction:
  $F(Q(\Gamma_1, \ldots, \Gamma_n)) \equiv_F Q(F(\Gamma_1), \ldots\ F(\Gamma_n))$
  where $F \in \mathcal{F}$, and $\Gamma_i \in \mathcal{T}$, $1 \leq i \leq n$;
- ($\mathcal{L}_0^+$)
  - Text constructions can be flattened (associativity, identity):
    $Q(\ldots\Gamma_1, Q(\ldots\Gamma_2), \ldots\Gamma_3) \equiv_F Q(\ldots\Gamma_1, \ldots\Gamma_2, \ldots\Gamma_3)$
    for all $\Gamma_i \in \mathcal{F}, 1 \leq i \leq n$;
  - Text construction is independent of the order (commutativity):
    $Q(\Gamma, \Delta) \equiv_F Q(\Delta, \Gamma)$ for all $\Gamma, \Delta \in \mathcal{F}$.
  - Text construction is independent of duplication (idempotency):
    $Q(\Gamma, \Gamma) \equiv_F Q(\Gamma)$ for all $\Gamma \in \mathcal{F}$.

In any $\bar{\mathcal{L}}_0^{+\Omega}$ language, we define an elementary statement to be a single proposition, titling or importation statement. Further, let an elementary text to be an elementary statement or the application of finite set of text operators and/or unary text construction to at most one elementary statement.

Each elementary text is a member of a full-equivalence class of elementary texts, and by the assumed closure of composition, each elementary text may be represented using at most one text operator.

By application of the assumed algebraic properties to a given text in a $\bar{\mathcal{L}}_0^{+\Omega}$ language, the text may be converted to a fully-equivalent text having the form of a text construction over some finite set of elementary texts. This representation is similar to a normal form in logic, so we call it a normal form of the text (it is not necessarily unique). A normal form $\bar{\mathcal{C}}$ of a corpus $\mathcal{C}$ is a set of texts such that each text of the original corpus $\mathcal{C}$ has a normal form in $\bar{\mathcal{C}}$.

Given a self-contained corpus $\mathcal{C}$, there is a unique importation closure $\hat{\mathcal{C}}'$ for all title models. Let $\bar{\mathcal{C}}'$ be a normal form of $\hat{\mathcal{C}}'$. Because of the compositional-compactness property, only a finite number of text operators are needed to represent all FE-classes of texts in $\bar{\mathcal{C}}'$.

[5] $\bar{\mathcal{L}}_0^{+\Omega}$ languages are Abelian Ω-loops where every finitely-generated sub-Ω-loop is compact relative to the discrete topology of the full-equivalence relation (i.e. has a finite number of FE-equivalence classes).

7 Data Model

We describe here a data model providing a compact representation of a normal-form text or FE-class from a particular $\bar{\mathcal{L}}_0^{+\Omega}$ language. The representation is designed to facilitate explanation of the algorithm for determining the cover of a self-contained corpus from such languages.

7.1 Corpus Table

A corpus is represented in one or two tables or arrays (if split, one for the type I texts and another for the type II texts). Each FE-class of the corpus is represented in a set of rows, one row for each elementary text in a normal form. A row is associated with an FE-class by its "classID" field, which contains an identifier.

In each row of the corpus table, the remainder of the fields contain a compact representation of an elementary text. The "txt" field contains a binary value indicating the presence of a unary text construction operator. The "op" field contains an identifier for a text operator, if present, with null or a special value to indicate its absence. The final fields describe the elementary statement, providing either the name of a proposition ("prop"), the title invoked by an importation statement ("import"), or the primary key of a titling statement("titleID"), to be discussed in the next section.

7.2 Titling Table

The texts associated with all titling statements that occur within the corpus under consideration are represented in the titling table, whether or not they are *active* (are asserted in the type II corpus table). Each row in the table corresponds to one elementary text in a normal form of the titled text.

It is possible for the same title to be assigned to two different texts within a corpus. There are certain circumstances when such a corpus may even be satisfiable, such as the second titling statement occurring within another titling whose text is never imported, or a text operator is applied that translates to a non-conflicting title. Therefore we cannot rely on the text title (the "title" field of the table) as a primary key for titling statements; the "id" field contains an identifier that creates the primary key for titling statements (optionally jointly with the title.)[6]

The remaing fields of the row contain a representation of the elementary text in the same form introduced in the previous section for the corpus table(s).

7.3 Representation of Example 5.5

The corpus from Example 5.5 is represented in the corpus Table 2 and titling Table 3.

[6] Alternatively the table schema could be normalized by factoring out the first column of the titling table into another table where the title name associated with each titling statement identifier is recorded. This table is particularly important in the more general case, not implemented here, where a text operator may modify the title of a titling statement.

Table 2. Example 5.5 Corpus Table (combined types I and II) - Initial State

classID	txt	op	prop	import	titleID
1	0				foo1
2	0			foo	

Table 3. Example 5.5 Titling Tables

title	id	txt	op	prop	import	titleID
foo	foo1	1		A		
foo	foo1	1	F0		foo	

8 Determination of Cover

We first state the algorithm procedurally and then illustrate for a particular example.

8.1 Overview of Cover Determination Algorithm

The procedure for updating the corpus table is described for the case of resolution of importations individually and assuming titling statements are unaffected by text operators:

- Construct the corpus and title tables, providing separate titleID's for each syntactically distinct titling statement.
- Locate an importation statement in some class in the corpus type I table.
- Determine if the text to be imported is well-defined. If there are multiple titling statements for this title asserted in the corpus type II table, the corpus is unsatisfiable. If there is no asserted titling statement for this title, defer the resolution of this importation until later - a titling statement for this title may be imported into the corpus elsewhere.
- Perform an importation resolution and update the corpus table.
  - copy the class containing the selected importation into a new set of rows, providing a new, unused classID.
  - locate the row, in the new class, corresponding to the selected importation to be resolved.
  - Update the text construction field in this row by binary OR between the existing vaue and the value in the FE-class to be imported.
  - Update the text operator field with the FE-class identifier obtained (e.g. from table lookup) for the composition of the existing operator applied to the operator in the class to be imported.
  - Compare the new FE-class to existing classes, and discard if it is equal to any existing class. The order of rows is not significant.
- Repeat the resolution and update process, applying only once to each individual importation statement in the corpus. Note that the order of application is immaterial except in the case of ill-defined imported texts. If all texts to be imported are well-defined, a fixed point will be reached.

– If a fixed point cannot be achieved due to ill-defined import(s), throw an error that a cover does not exist due to missing or conflicting titling statement(s), as appropriate.

We present the cover determination algorithm by illustration as it would apply to Example 5.5. The second text in the original corpus has an importation statement (row 2 of the corpus table) which invokes the name "foo". There is no text operation applied to the importation, so no name translation need be performed.

There is only one active titling statement (titleID "foo1") in the corpus table and it applies to the name "foo". Therefore the text to be imported is well-defined. Now if we carry out the importation resolution explicitly and normalize we obtain the text

$(txt\ (A)\ (F_0\ (\textit{import}\ \text{foo})))$

We then tentatively add the representation of this text to the corpus table. In terms of the tabular representation, this is a sort of nonlinear join.

Table 4. Example 5.5 Corpus Table - After Step 1(new rows only)

classID	txt	op	prop	import	titleID
s3	1		A		
3	1	F0		foo	

We compare the new text with existing texts in the corpus. In this case, the new text is different and thus is retained. Further, the new text has an importation so resolution is repeated. We assume, for illustration purposes, that F0 $\circ$ F0 = F0. In general there will be a finite set of FE-classes of operators containing F0^n, $n \geq 1$.

Table 5. Example 5.5 Corpus Table - After Step 2 (new rows only)

classID	txt	op	prop	import	titleID
4	1	F0	A		
4	1	F0		foo	

The new class is again different than existing classes and is retained. It also has an importation so resolution is repeated. However, this time the new class is the same as the class it was derived from. The procedure has reached a fixed point and terminates.

9 Discussion

The following feedback to the ISO CL language design was developed on the basis of the analysis above:

– This approach allows a design decision regarding whether all importations in a corpus should be resolved simultaneously, or individually. The algorithm reveals:

- Individual importation resolution is more convenient to implement.
- For simultaneous resolution, there are cases where the algorithm as defined above fails to determine a cover when one does exist. For example, with the two text corpus
  $\{(\textit{title}\ \text{bar}\ (\textit{title}\ \text{foo}\ (A)))$
  $(\mathit{txt}\ (\textit{import}\ \text{foo})\ (\textit{import}\ \text{bar}))\}$
  simultaneous resolution of the two importations in the second text fails because of a missing titling statement for "foo". Individual resolution introduces the missing titling statement when the importation for "bar" is resolved.

– There is much redundancy in copying an entire text whenever an importation resolution is performed, which could be eliminated if text construction has conjunction-like semantics, so that every corpus can be expanded to a fully-equivalent corpus containing only elementary texts. This would not only reduce the size of the cover; it would make the termination criterion easier to check, since only single rows would need to be compared. However, conjunction-like semantics implies monotonicity; this simplification would not be applicable to nonmonotonic logics, such as defeasible logics.

10 Conclusion

We have investigated a family of propositionally-based languages containing generic text operators and a titling/importation mechanism, identifying a set of characteristics that allow a practical implementation of importation closure, in preparation for reasoning, that is robust in the presence of circular importation dependencies. The study has demonstrated its utility by providing feedback to the revision of a standard KR language, ISO CL. The extension of the analysis to the case of text operators that modify titling statements warrants further consideration.

Acknowledgements Thanks to Fabian Neuhaus for valuable suggestions on this paper.

References

[1] ISOIEC. *Information technology Common Logic (CL): a framework for a family of logic-based languages*.

[2] Fabian Neuhaus and Tara Athan. *CL Semantics Strawman @ONLINE*. Apr. 2013. URL: http://philebus.tamu.edu/pipermail/cl/attachments/20130405/153ad554/attachment-0001.pdf.

[3] Fabian Neuhaus and Pat Hayes. "Common Logic and the Horatio problem". In: *Appl. Ontol.* 7.2 (Apr. 2012), pp. 211–231. ISSN: 1570-5838. URL: http://dl.acm.org/citation.cfm?id=2351667.2351672.