Mind your Language!
Controlled Language for Inference Purposes

Jana Z. Sukkarieh
Computational Linguistics Group
The Clarendon Institute, University of Oxford
England, OX1 2HG
jana.sukkarieh@clg.ox.ac.uk

Abstract

The Knowledge Representation (KR) community and the Natural Language Processing community, in our opinion, have common goals yet finding a language that is expressive enough and capable of efficient reasoning is yet a challenge. We have claimed elsewhere that having a Natural Language(NL)-like KR may be a step towards solving that challenge. The NL-like KR we looked at defines a controlled subset of English that is not trivial and exhibits powerful reasoning properties. **Controlled Language for Inference Purposes (CLIP)** is a dialect of English that was considered while developing a domain-independent knowledge-based system. The system takes as input 'clippy' utterances, U_i, and uses a NL-like KR called McLogic to deduce plausible inferences from U_i and give a justification for these deductions.

1 Introduction

1.1 A Broad Picture

A very good medium humans communicate with, state their problems in and express how they solve these problems is Natural Language (NL). Hence, one wishes to use NL as a Knowledge Representation(KR) in knowledge-based sytems. NL is very expressive-a desiderata that the KR community seek- but NL is ambiguous- a characteristic that is definitely not suitable for a KR. Therefore, in our work we looked at a Quasi-NL KR ((Sukkarieh and Pulman, 1999), (Sukkarieh, 2001b), (Sukkarieh, 2001a)). A KR language that supports precise and rigorous formal reasoning yet with the same practical advantages, from the user point of view, and friendly character that NL has. In addition to this friendly character, the KR that we have looked at is efficiently capable of deductive inferences that different semanticists (see FraCas test suite[1] (Cooper et al., 1996)) have agreed upon as the best test for any NL understanding system. The work we have done (Sukkarieh, 2001a) was motivated by the belief that the best test for the semantic capacity of any Natural Language Understanding (NLU) system is its capability to reason.

The **NL-like KR** and **the inference test suite** *define* the controlled language that we will look at in this paper, namely, **Controlled Language for Inference Purposes(CLIP)**. CLIP is a computer processable, **semantically-driven** sublanguage of English. Moreover, as the name suggests, it is an **inference-driven** sublanguage. Hence, it seems natural to describe the KR[2] first then the inferences we are looking at second and then CLIP. Before that, we "review" the properties of controlled languages, in particular, semantically-driven computer processable ones.

1.2 International Interchange of Ideas

The search for a common scientific language for communication among different countries started when Latin stopped serving that purpose. Uni-

[1] http://www.cogsci.ed.ac.uk/ fracas/

[2] acting at the same time as a semantic representation.

versal Languages were an attempt (Knowlson, 1975), (Slaughter, 1982) for, as the name suggests, a universal use. In the twentieth century, Esperanto and C. K. Ogden's Basic English (1929-1934) were viewed as candidate common languages. British American Scientific International Commercial (Basic) is a simple form of the English language with 850 words (650 nouns and 200 verbs) only. It was proposed as an international use for day-to-day communication. It has been used in courses for teaching foreigners (and non-foreigners) as a basis of written and spoken English. Restricted languages appeared later especially for technical writing on an international level, like Caterpillar Fundamental English (CFE) or Simplified English (SE). These languages can be referred to as languages for practical business. CFE was developed, in 1971, by the Caterpillar Tractor company. It is a 900-word vocabulary that is used for writing product documentation on agricultural vehicles for worldwide distribution. Similarly, SE was devised, in 1979, by the Douglas Aircraft Company. It is a 2,000 word dictionary that is used for writing its technical manuals. Later, researchers addressed constraints on such languages and computer systems were developed to check for conformance, for example, (AEC, 1995),(R. Wojcik, 1990), (Wojcik and Holmback, 1996). An international interchange of ideas for technical or business purposes meant devising writing systems that contain limited choices of standard English words and their senses and which restrict acceptable sentence structure and/or general document structure and presentation. Their main goal is to simplify written communication and to decrease the occurrence of ambiguities or avoid them all together.

1.3 Computer Processable Controlled Languages

Computer Processable Controlled Languages (CPCLs) are dialects of English that have the same properties of the writing systems mentioned above and "... more: dialects that are restricted as to be capable of being completely syntactically and semantically analysed by a natural language processing system..." (Pulman, 1996). Pulman's Computer Processable English (Pulman and Rayner, 1994), (Macias and Pulman, 1995), Attempto (Fuchs and Schwitter, 1995), (Fuchs and Schwitter, 1996) and the semantically-derived subset of English of (Holt and Klein, 1999) are all examples of such languages. CPCLs as opposed to Computer Processable full languages have specific purposes or applications like knowledge representation purposes, specification and verification or machine translation. Usually, training is needed and some languages, obviously, are more habitable than others but we will not investigate this here. The idea of a semantic-driven language, that is, defining a controlled language via the well-formedness of its translations into a logic originated in (Macias and Pulman, 1995) and (Holt and Klein, 1999) followed them. With the semantically-derived sublanguage (Holt and Klein, 1999), the logic seems to be the start point. Attempto (Fuchs and Schwitter, 1996), on the other hand, starts with the language and then look at the logic. In our case CLIP and McLogic go in tandem as we will see in section 3 where we show that CLIP is not trivial. In the next section we zoom in a bit to look briefly at the two main ingredients that define CLIP, namely, a Quasi-NL KR and a set of inferences.

2 Main Ingredients

2.1 NL-like KR: McLogic at a Glance

McLogic is an extension of a Knowledge Representation defined by McAllester et al (McAllester and Givan, 1992), (McAllester et al., 1991). Other NL-like Knowledge representations are described in (Hwang and Schubert, 1993), (Schubert and Hwang, 2000) and (Iwanska, 1996), (Iwanska, 2000) but McAllester et. al have a tractable inference relation equipped with their logic and was never, as far as we know, incorporated into a Natural Language Processing (NLP) system before we started our work. The basic form of McLogic, we call it McLogic$_0$, and some extensions built on top of it are presented next.

McLogic$_0$

The basic notion of $McLogic_0$ is a **class expression**, that is, an expression that denotes a set. In general, any monadic predicate symbol of classical syntax can be used as a class expression. Furthermore, for any class expression *s* and binary relation *R* one can construct the class ex-

presssions *(R (some s))* and *(R (every s))*. For example, if *like* is a binary relation symbol and *woman* is a class symbol, then one can construct the class expressions *(like (some woman))* and *(like (every woman))*. These denote the set of all entities in the domain 'who like some woman' and the set of all those 'who like all women' respectively. Examples of **formulae** in McLogic could be: $(every\ actor\ handsome)$,
$(some\ researcher\ poor)$,
$(some\ human\ exists)$,
and $(some\ academic\ rich)$.

An important feature of McLogic$_0$ is that it is very much **"English-like"**. By that we mean it is easy to read, natural-looking 'word' order[3].

Including Some Extensions: $McLogic_i$, where $i > 0$

The NL-like property and the efficiency of the inference are the main *controlling* factors to the extension. In the extension, the two logical notions are the same, namely, class expressions and formulae. In addition to these two, we have introduced the notion of a function symbol that we incorporated into the syntax and so as not to complicate the logic, we specified the result of applying a function to a class expression to be again a class expression. We won't go into details here. A grammar for the syntax of McLogic class expressions and formulae is given below, followed by examples. In the grammar, R is a binary relation or the inverse of a binary relation. R^3 is 3-ary relation or the inverse of a 3-ary relation, TimeExpression is a time class expression (denoting an interval of time), Q_i for $i = 1, 2$ is either $some$ or $every$, Q_i for $i = 3, 4$ are cardinal class expressions (denoting a cardinal), s and t are class expressions that are neither time class expressions nor cardinal class expressions. C-expr represents the set of class expressions and F represents the set of formulae:

[3]with the exception of Lambda expressions which are not very English! Consider $(John\ \lambda x.(x\ (like\ x)))$ that represents 'John likes himself'. However, the last issue could be solved by defining a formal macro, $himself$, that would serve as an intermediate representation. We won't go into this here. Please See details in (Sukkarieh, 2001a).

(10) $\frac{(some\ C\ exists)}{(some\ C\ C)}$ (12) $\frac{(some\ C\ W)}{(some\ W\ C)}$

(11) $\frac{(some\ C\ W)}{(some\ C\ exists)}$ (13) $\frac{(every\ C\ W),(every\ W\ Z)}{(every\ C\ Z)}$

Table 1: Example Inference rules for McLogic$_0$. C, W and Z range over class expressions.

C-expr ::=	F ::=
c a constant symbol |	$(some\ s\ t)$ |
s a predicate symbol |	$(every\ s\ t)$ |
$(R(some\ s))$ |	$(Q_3 * s\ t)$ |
$(R(every\ s))$ |	negation of F |
x a variable symbol |	Bool. combinations of F
$\lambda x.\phi(x)$ |	
$s + t$ |	
$s\#Mod$ |	
$s\$t$ |	
$(R^3(Q_1\ s)\ (Q_2\ t))$ |	
$(R^3(Q_3 * s)\ (Q_4 * t))$ |	
$(R^3(Q_1\ s)\ (Q_4 * t))$ |	
$(R^3(Q_3 * s)\ (Q_2\ t))$ |	
$Q_3 * s$ |	
$TimeExpression$	

What we have done, basically, is allow a wider range of class expressions that denote sets of entities and sets of sets of entities but at the same time not complicate the formulae. Example:

$(Pooh\ bear)\ and\ (Eeyore\ donkey)\ and\ (Piglet\ pig)\ and\ (Tigger\ tiger).$
$(Piglet\ (organise\ (some\ party)).$
$(every\ animal\ (bring(some\ present))).$
$(Eeyore\ (get\ (some\ small + present))).$
$(Eeyore\ (irritate(more - than - one * animal)))\ and$
$(two * bee\ (gave\ (Eeyore)\ (some\ sting))).$
$((most * animal\ laughed)\ after$
$(Eeyore(sting^{-1}(by(some\ entity))))).$
$not(Eeyore\ laugh).$

The proof theory of McLogic is a set of inference rules of the form $\frac{Premise_1 ... Premise_i}{Conclusion}$. See table 1 for examples. We extended the proof theory by adding more rules motivated by the inferences that we describe, briefly, next.

2.2 Structurally-Based Inferences

We are concerned with NL inferences but not with implicatures nor suppositions. Moreover, we do not deal with defeasible reasoning, abductive or inductive reasoning and so on. In our work (Sukkarieh, 2001a), we focus on deductive (valid)

inferences that depend on the properties of NL constructs. Entailments from an utterance U, or several utterances U_i that seem "natural", in other words, that people do entail when they hear U. For example, in the following, D_1 are deduced from Scenario S_1:

- S_1:

  (1) a. some cat sat on some mat.
  b. The cat has whiskers.

  $\Downarrow$

  D_1: some cat exists,
  some mat exists,
  some cat sat on some mat,
  some cat has whiskers (cat_1 has whiskers),
  some whiskers exist.
  $\not\Downarrow$
  whiskers sat on some mat

We define a **structurally-based inference** to be one that depends on the specific semantic properties of the syntactic categories of sentences in NL. For example,

- S_2: most cats are feline animals.
  $\Downarrow$
  D_2: most cats are feline,
  most cats are animals.

- S_3: Smith and Jones signed the contract.
  $\Downarrow$
  D_3: Smith signed the contract.

D_2 depend on the monotonicity properties of generalised quantifiers and D_3 on those of conjoined Noun Phrases. Our aim is a general inference component that is part of a Knowledge-based system that uses McLogic as the internal KR. Our reference, guide and the best benchmark we could find is the Fracas Test Suite[4] which is a domain-independent collection of structurally-based inferences. Having, briefly, described the logic and the inferences, how do these fit into the picture of defining CLIP.

3 Inferences, McLogic $\Longleftrightarrow$ CLIP

Definition 1 ***CLIP*** *is a sublanguage of English with the following properties:*

- *It is syntactically and semantically* ***processable by a computer***

- *Each sentence in CLIP has* ***a well-formed translation in McLogic***.

- ***The ambiguities in a sentence are controlled in a way that the interpretation of that sentence allows inferences required in FraCas D16 (Cooper et al., 1996)***

- *The* ***vocabulary*** *is controlled only as far as the syntactic category.*

The word CLIP implicitly 'clips' a part of the 'whole', that is, dialect or sublanguage not full English. Here is an example:

> **Calvin:** Susie understands some comic books. Many comic books deal with serious issues. All superheroes face tough social dilemmas. It is not true that a comic book is an escapist fantasy. Every comic book is a sophisticated social critique.
>
> **Hobbes:** Most comic books are incredibly stupid. Every character conveys a spoken or graphic ethical message to the reader before some evil spirit wins and rules.

McLogic_0 is the basic building block for CLIP. To start with, an English sentence belongs to CLIP if, and only if, it has a well-formed translation in McLogic_0. Further, we extended McLogic_0 to account for more English constituents motivated by the structurally-based inferences in the FraCas test suite. Inferences with their corresponding properties, premises and conclusion add to the expressivity of the dialect. To emphasize the above idea, we consider some kind of recursive view:

> **Base Case:** McLogic_0
> **Recursive Step:** $McLogic_n$ depends on $McLogic_{n-1}$

However, it is not an accumulative one-way hierarchy of languages since the English language motivates the extension. Besides, the view that McLogic could be extended indefinitely is not plausible. In other words, one can say that McLogic may be viewed as a representation language that can be extended as much as the language needs and hence, it may not really be seen as something that could control the English input. One can add as much as one likes! Though this may be true in theory, the extension is "controlled" by several parameters like the inference process, its efficiency and minimising the addition of inference rules, the ease in the

[4] http://www.cogsci.ed.ac.uk/ fracas/

extension itself, the ease in the translation to McLogic which is to be kept as Quasi-NL as possible. It is a back-and-forth process on 'What controls what' with minimum extension to McLogic. Consequently, McLogic and CLIP go in tandem. To summarise, consider figure 1 below.

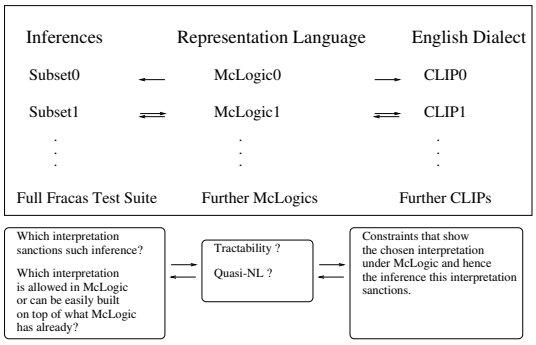

Figure 1: Reasoning $\longleftrightarrow$ Representation Language $\longleftrightarrow$ English Sublanguages

In the figure, the 3 components, namely, the set of inferences, McLogic and the controlled language acted as 3 parameters controlling the augmentation or the extension. McLogic_0 defines both Subset_0 and CLIP_0. Subset_0 consists of all inferences in the Fracas Test suite whose premises and conclusion are expressable as well-formed formulae in McLogic_0 and that the proof theory sanctions. Subset1 is a bigger subset of inferences that include Subset_0 and defined by McLogic_1 and CLIP_1. The aim eventually is to have a logic that is capable of doing the full FraCas test suite.

3.1 Specification of CLIP

Systems vary in describing the way they control the written texts. It could vary from deciding about punctuation, symbols, acronyms, listing the words allowed and/or forbidden, limit the use of tense and/or aspect of verbs, limit the meaning of a word, the use of specific words to a designated part of speech, ambiguous attachments, coordinated or subordinated structures, complex sentences, long-distance dependencies, agreement errors, anaphora use and so on. In summary, it is a set of rules concerning terminology or vocabulary and grammar. In case of CLIP, the specification is basically the well-formedness of translations in McLogic and some constraints due to the inferences. The grammar is specified by the fact that CLIP is processable by a machine. To specify CLIP, we are going to describe what McLogic can express. For lack of space, the presentation will be kept to a minimum and a lot of interesting motivation are going to be missed (please see (Sukkarieh, 2001a) for translation between CLIP and McLogic and for a more detailed motivation on the extension of McLogic_0 by certain English constituents). In the following, we describe CLIP_0, McLogic_i for some $i > 0$. Before that, we "imagine" a simpler language than CLIP_0 that is capable of powerful reasoning while its English is rudimentary. We could call it subCLIP_0, as it is a very restricted sublanguage of CLIP_0 but we will just refer to it by the reasoning task associated with it, as the following shows.

The Controlled Language of Categorical Syllogisms

We state some definitions about syllogisms in order to show what kind of controlled English they define.

Definition 2 *A **syllogism** is a deductive argument in which a conclusion is inferred from two premises. The premises and conclusion are called propositions.*

Definition 3 *A **standard-form categorical syllogism** is one with all propositions having a standard form and they are arranged in a standard order. For example,*

No hero is a coward
Some soldier is a coward

Some soldier is not a hero

Definition 4 *A **standard-form categorical proposition** is any of the form "All S is P", "Some S is P", "No S is P", "Some is not P".*

There are 256 distinct forms that standard-form syllogisms may assume. However, as far as their representation in McLogic and the inferences concerned, the 256 different forms are treated similarly. In McLogic, the standard-form propositions can be written as $(every\ S\ P)$, $(some\ S\ P)$, $not(some\ S\ P)$ and $(some\ S\ not(P))$ respectively, where S and P are monadic class expressions (or intersection of two monadic class expressions).

CLIP$_0$

Let V denote the translation from English to McLogic. V^{-1}(McLogic$_0$) will denote the English categories and sentences that have a corresponding translation. Note that NPs are not defined 'on their own'. Let Mc_0 be V^{-1}(McLogic$_0$):

$Mc_0 =$
- Proper Names,
- Countable Common Nouns
- Adjectives,
- Intransitive and Transitive verbs
- PPs as :
- complements for a predicative verb
- VPs either:
- transitive or intransitive or
- predicative
- Declarative sentences of the form:
- 'some N VP',
- 'every N VP' or
- 'Name VP'
- Yes/No questions of the form:
- 'Is some N VP',
- 'Is every N VP' or
- 'Is Name VP',
- 'does/did some N VP_{base}',
- 'does/did every N VP_{base}'.
- VP_{base} is a verb phrase
- where the verb occurs in its base form.

The general inference component that we developed sanctioned the following benchmark tasks:

Bench. Reasoning Tasks =
- the Grocer Puzzle (GP)
- the Schubert's Steamroller (SS)
- the Rich Old Man Puzzle (ROLP)

Figures 2, 3 and 4 (see Appendix A) state (GP), (SS), and (ROLP) respectively, in CLIP$_0$ as well as in McLogic$_0$. SS was presented in 1978 by Lenhart Schubert as a challenge to automated-deduction systems (Stickel, 1986).

CLIP$_1$

English Category	Example with McLogic translation
Numerals, NL Determiners	Cardinals: $(ten * bookold)$
Generalised Quantifiers	less than 10 , most, more than 5 books : $(most * bookold)$
Ditransitive verbs	Mary gave a present to Betty $(mary\ (gave(some\ present))\ betty)$
Simple Plurals	some books are old $(more\_than\_one * book\ old)$
Nominal modification by Adj	a happy man : $happy + man$
Nominal modification by PPs	some man with some red hat laugh $(some\ man + (with\ (some\ red + hat))\ laugh)$
Relative Clauses (Subject)	every student who laughs succeeds $(every\ (student + laugh\ succeed)$
Relative clauses (Object)	John whom Mary likes is a musician $(john + \lambda x.(mary\ (like\ (x)))\ musician)$
Verb Phrase modification	Sam drives fast : $(Sam\ drive\#fast)$
Passives	Some student was given a book by John $(some\ student\ (give^{-1}((some\ book)\ (john))))$
Comparatives	every bird is smaller than every horse $(every\ bird\ (smaller\_than(every\ horse)))$
	Some man borrowed less than ten books $(some\ man\ (borrow(less\_than\_ten * book)))$
	Smith owns a smaller computer than John $(Smith\ (own\ (more\_than\_one * computer+\ (more-than\ (some\ computer+\ \lambda x.(Jones\ (own\ (some\ x))))))))$.

The table above summarises the English constituents that have well-formed translations into McLogic. It is important to note that NPs, as we said above, like 'some man' and 'every woman' cannot be represented in McLogic$_0$. The reason is that there is no independent meaning for $(some\ S)$ and $(every\ S)$. However, with the introduction of other determiners, we can have a uniform representation for quantifiers. One can show that 'some' and 'every' can be treated as the rest. See a proof in Appendix B.

It is also important to mention that the additional control factors mentioned before imposed additional constraints on CLIP. We only give two examples here:

☞ Scope of quantifiers (a constraint imposed by McLogic$_0$):
Consider the following two examples,

(2) a. CLIP:every man likes a woman.
b. McLogic:
$(every\ man\ (like(some\ woman)))$.

Hence, the interpretation of the above sentence is where the universal quantifier has wider scope, for each man there is a woman whom the man likes. While in the following example the existential quantifier has wider scope, there exists a woman who likes all men.

(3) a. CLIP: a woman likes every man.
b. McLogic:
$(some\ woman\ (like(every\ man))$.

The interpretation in McLogic$_0$ basically respects the order in which the quantifiers appear in a sentence. McLogic$_0$ does not use 'every' and 'some' as quantifiers but only like combinators. For example, if you say $cos\ sin^{-1}\ sin\ \frac{\pi}{2}$ you can only consider them in order. Hence, the earlier the quantifier appears, the larger scope it has. When we extended McLogic$_0$, we kept the English quantifiers as combinators like in McLogic$_0$ rather than quantifiers and hence their scope is determined as they appear:

(4) a. every student likes most teachers.

b. $(every\ \ student\ \ (like(most\ *\ teacher)))$.

The interpretation is such that every entity who is a student likes most entities who are teachers.

☞ Collective and distributive reading (Constraint imposed by minimising the addition of inference rules to the proof theory):
In McLogic, a sentence like 'Smith and Jones sign five contracts' is translated into $(smith\ (sign(five * contract)))$ and $(Jones\ (sign(five * contract)))$ and hence it is considered to have a distributive reading. A distributive reading is favoured for the fact that in this case the only rules of inference required are those common for generalised quantifiers. However, if a collective interpretation is allowed, we need to consider additional inference rules, at least rules of arithmetic.

4 Conclusion

4.1 Transition from Categorical Language to CLIP

A standard form categorical syllogism may be thought of as being free from all obscurities and irrelevancies. Needless to say, of course, arguments do not always occur thus refined in a "state of nature" (Copi, 1990). In the following, we list some cases in which the terms used in the propositions do not qualify it to be in standard-form, as defined above. We then explore how logicians deal with them and how we dealt with them.

- Propositions with unique entities. 'John is P' and 'The man is P' are two examples. The logicians' way is to write these as [all John is P] and [all man is P], that is, considering that John or the table denotes a singleton set that is a subset of P. That is exactly what we do and how the representation in McLogic works. Similarly, sentences like 'John is not P' or 'The table is not antique', that is [S is not P], can be written as [no S is P] where S denotes the unit class whose only member is the object S.

- Propositions for which P is not a class term but an attribute or a property. 'All flowers are beautiful' and 'All students are ready to work hard' are two examples. In the usual method of standard-form syllogisms these have to be translated into something like 'all flowers are beauties' and 'all students are persons ready to work hard'. In our case, they can be left as they are since CLIP is not as controlled as standard-form propositions and since McLogic treats attributes and properties as class expressions (or class term in the syllogisms language).

- Sentences that have as a main verb other than "to be". For example, 'some people drink' which logicians translate into 'some people are drinkers' to be in standard-form. Again, as long as the verb is transitive or ditransitive then it is represented in McLogic and can be included without any change.

- Sentences in which the standard-form ingredients are present but not put in the standard-form, for example, there is no word that indicate quantities as in 'Computers are machines'. This can be translated into "All computers are machines". Bare plurals in object position are tricky. In CLIP, it is encouraged that the determiner should be specified like logicians suggest. If not then 'some' is assumed as a determiner (unless otherwise stated or known from context).

- Categorical propositions that have their quantifiers different than 'all', 'some' and 'no'. If the quantifier is 'every' or 'any' then it is easily taken care of as it can be replaced by 'all' in logicians' dictionary. In our case, there is no need to replace 'every'. Saying that 'John can borrow any book from the library' cannot be replaced by 'John can borrow every book/all books from the library'. The quantifier 'any' depends on context and sometimes mean 'all' and sometimes 'some'. The quantifiers, 'a', 'an' were treated existentially and one should use a universal quantifier explicitly when one needs to. The definite description quantifier, 'the', may designate a unique entity as in 'the

king is stupid'[5] or all members of a class as in 'the elephant has a trunk'. Negative statements like 'Not any S is P' is replaced by 'no S is P' or 'It is not true that some S is P'; 'Not every S is P' can be either left as it is or replaced by 'some S is not P'.

- Using say 4 terms with two of them being synonyms. To be able to transform into a standard-form syllogism, synonyms must be replaced by the same term.

Not all syllogisms can be transformed into standard-form ones. Natural language well-known intricacies prevent the application of this simple reasoning task even if the task is not required through a machine. Examples of such syllogisms can be seen in the arguments of (Sukkarieh, 2001b).

4.2 Wrapping it all!

The aim was to build a KR system. Instead of starting with real English and ask ourselves which logic is suitable and how the translation is to be done, we started with a logic, we asked ourselves what can we do with it, which English sentences are translated into it, and then back to the logic again and asking how to extend it to cover such and such English and so on. All this keeping in mind that the main motivation is the inference process and the kind of inferences. It is important to mention also that the purpose a CPCL serves and the kind of user that it addresses tailor the specification of the CPCL. For example, if the user (or application) is a translator then s/he will be interested to know more about the restrictions on the senses of the words say, while if the user is a checker then more information about part-of-speech and grammatical rules are needed. As we aim for a general **reasoning** component, the users of such system vary and the specification of CLIP will be tailored depending on specific users, possibly in forms of different user manuals for different users. We have already used CLIP and McLogic to describe a non-trivial **specification** task written in the Z language (again you can see that in Chapter 7 of (Sukkarieh, 2001a)). That gave us more insight on how to tailor the language for specification purposes. Finally, even with this quite restricted dialect, some problems that causes ambiguity a user may still not be aware of. For example, saying that 'John is an English student' could mean that 'John is English' (as nationality) or that 'John is reading for an English degree (at the university)'[6]. Therefore, some problems may not be obvious for someone writing a specification for a controlled language.

[5] 'the king' being referential to an entity already in context does not make any difference.

[6] An example pointed out to me by Pf. Hess while giving a talk at the Univ. of Zurich.

References

The European Association of Aerospace Industries (AECMA), 1995. *A guide for Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintenance Language*, 1 edition, September.

R. Cooper, D. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, and S. G. Pulman. 1996. The fracas consortium, deliverable d16. With additional contributions from Briscoe, T. and Maier, H. and Konrad, K.

I. M. Copi. 1990. *Introduction to Logic*. Macmillan Publishing Co., Inc.- New York and Collier Macmillan Publishers- London, 8 edition.

N. E. Fuchs and R. Schwitter. 1995. Attempto controlled natural language for requirements specifications. In *Seventh ILPS 95 on Logic Programming Environments*, December.

N. E. Fuchs and R. Schwitter. 1996. Attempto controlled english. In *The First International Workshop on Controlled Language Applications CLAW 96*, March.

A. Holt and E. Klein. 1999. A semantically-derived subset of english for hardware verification. In *Proceedings of the 37th Annual Meeting for Association for Computationl Linguistics*, pages 451–456. Association for Computational Linguistics.

C. H. Hwang and L. K. Schubert. 1993. Episodic logic: A comprehensive, natural representation for language understanding. *Minds and Machines*, 3(4):381–419.

L. M. Iwanska. 1996. Natural language is a representational language. In *Knowledge Representation Systems based on Natural Language*, pages 44–70. AAAI Press.

L. M. Iwanska, 2000. *Natural Language Processing and Knowledge Representation*, chapter Natural Language is a Powerful Knowledge Representation system, pages 7–64. MIT press.

J. Knowlson. 1975. *Universal Language Schemas in England and France 1600-1800*. University of Toronto Press.

B. Macias and S.G. Pulman. 1995. A method for controlling the production of specifications in natural language. *The Computer Journal*, 38(4):310–318.

D. McAllester and R. Givan. 1992. Natural language syntax and first-order inference. *Artificial Intelligence*, 56:1–20.

D. McAllester, R. Givan, and S. Shalaby. 1991. Natural language based inference procedures applied to schubert's steamroller. *AAAI*.

S. Pulman and M. Rayner. 1994. Computer processable controlled language.

S. Pulman. 1996. Controlled language for knowledge representation. In *Proceedings of the First International Workshop on Controlled Language Applications*, pages 233–242. First Internatinal Workshop on Controlled Language Applications.

K. Holzhauser R. Wojcik, J. Hoard. 1990. The boeing simplified english checker. In *International Conference Human Machine Interaction and Artificial Intelligence in Aeronautics and Space*, pages 43–57, Centre d'Etude et de Recherche de Toulouse.

L. K. Schubert and C. H. Hwang, 2000. *Natural Language Processing and Knowledge Representation*, chapter Episodic Logic Meets Little Red Riding Hood-A comprehensive Natural Representation for Language Understanding. AAAI Press; Cambridge, Mass.: MIT.

M. M. Slaughter. 1982. *Universal Languages and Scientific Taxonomy*. Cambridge University Press.

M. E. Stickel. 1986. Schubert's steamroller problem: Formulations and solutions. *Journal of Automated Reasoning*, 2:89–101.

J. Sukkarieh and S. G. Pulman. 1999. Computer processable english and mclogic. In *H. Bunt et al. (eds) Proceedings of the Third International Workshop on Computational Semantics*, Tilburg, The Netherlands.

J. Sukkarieh. 2001a. *Natural Language for Knowledge Representation*. Ph.D. thesis, Computer Lab. University of Cambridge. http://www.ling-phil.ox.ac.uk/staff/people/staff/jana/jana.htm.

J. Sukkarieh. 2001b. Quasi-nl knowledge representation for structurally-based inferences. In *P. Blackburn and M. Kolhase (eds) Proceedings of the Third International Workshop on Inference in Computational Semantics*, Siena, Italy.

R. Wojcik and H. Holmback. 1996. Getting a controlled language off the ground at boeing. In *CLAW 1996, Proceedings of the First International Workshop on Controlled Language Applications*. Pittsburgh: Language Techonologies Institute, Carnegie Mellon University.

Appendix A

✍ Description of Schubert's Steamroller

Given:
Every wolf is an animal.
$(every\ wolf\ animal)$.
Every fox is an animal.
$(every\ fox\ animal)$.
Every bird is an animal.
$(every\ bird\ animal)$.
Every caterpillar is an animal.
$(every\ caterpillar\ animal)$.
Every snail is an animal.
$(every\ snail\ animal)$.
Some wolf exists.
$(some\ wolf\ exists)$.
Some fox exists.
$(some\ fox\ exists)$.
Some bird exists.
$(some\ bird\ exists)$.
Some caterpillar exists.
$(some\ caterpillar\ exists)$.
Some snail exists.
$(some\ snail\ exists)$.
Every grain is a plant.
$(every\ grain\ plant)$.
Some grain exists.
$(some\ grain\ exists)$.
Every caterpillar is smaller than every bird (is).
$(every\ caterpillar\ (smaller\_than(every\ bird)))$.
Every snail is smaller than every bird (is).
$(every\ snail\ (smaller\_than(every\ bird)))$.
Every bird is smaller than every fox (is).
$(every\ bird\ (smaller\_than(every\ fox)))$.
Every fox is smaller than every wolf (is).
$(every\ fox\ (smaller\_than(every\ wolf)))$.
It_is_not_true_that some wolf eats some fox.
$(not(some\ wolf\ (eat(some\ fox))))$.
It_is_not_true_that some wolf eats some grain.
$(not(some\ wolf\ (eat(some\ grain))))$.
Every bird eats every caterpillar.
$(every\ bird\ (eat(every\ caterpillar)))$.
It_is_not_true_that some bird eats some snail.
$(not(some\ bird\ (eat(some\ snail))))$.
Every caterpillar eats some plant.
$(every\ caterpillar\ (eat(some\ plant)))$.
Every snail eats some plant.
$(every\ snail\ (eat(some\ plant)))$.
Every animal eats every plant or every animal
$(every\ animal\ (eat(every\ plant)))$\$
that is smaller than itself and eats some plant.
$lambda(X\ (every\ X\ (eat(every\ animal+$
$((smaller\_than(some\ X))+(eat(some\ plant)))))))$.
Show that :
Some animal eats an animal that eats some grain
$(some\ animal\ (eat(some\ animal+(eat(some\ grain)))))$.

✍ The Grocer Puzzle

Given:

every honest industrious man is healthy

$(every\ honest+industrious+man\ healthy)$

It_is_not_true_that some grocer is healthy

$not(some\ grocer\ healthy)$

every industrious grocer is honest

$(every\ industrious + grocer\ honest)$

every cyclist is industrious

$(every\ cyclist\ industrious)$

every unhealthy cyclist is dishonest

$(every\ unhealthy + cyclist\ dishonest)$

every grocer is a man

$(every\ grocer\ man)$

every cyclist is a man

$(every\ cyclist\ man)$

Show that: It_is_not_true that some grocer is a cyclist

$not(some\ grocer\ cyclist)$

✍ Description of the Rich Old Man Puzzle

Given:
every person who is not active is rich.
$(every\ person + not(active)\ rich)$.
every old poet is talented.
$(every\ old + poet\ talented)$.
every studious man is a magistrate.
$(every\ studious + man\ magistrate)$
It is not the case that some active politician is a snufftaker
$\neg(some\ active + politician\ snufftaker)$.
every studious man who is not rich is a poet.
$(every\ studious + man + not(rich)\ poet)$.
every fat person who is a politician is a snufftaker.
$(every\ fat + person + politician\ snufftaker)$.
every man who is a poet is a person.
$(every\ man + poet\ person)$.
every old magistrate is a politician.
$(every\ old + magistrate\ politician)$.
every talented person is fat.
$(every\ talented + person\ fat)$.
Show that: every studious old man is rich.
$(every\ studious + old + man\ rich)$.

Appendix B: Uniform Treatment of Generalised Quantifiers

NPs, like 'some man' and 'every woman' cannot be represented compositionally in McLogic$_0$. The reason is that there is no independent meaning for $(some\ S)$ and $(every\ S)$. However, with the introduction of other determiners, it seems that we can have a uniform representation for quantifiers. One can show that 'some' and 'every' can be treated as the rest. We review what we mean by cardinal class expressions.

Definition 5 *A cardinal class expression, N, represents a positive integer N and denotes the set $V(N) = \{X|X$ is a set of N objects$\}$.*

Note that a cardinal class expression does not have a meaning independently of entities (in some specified domain). This is motivated by the way one introduces a (abstract) number for a child, it is always associated with objects.
Having defined a cardinal class expression, the operator, $*$ can be defined:

Definition 6 *Given a cardinal class expression N and a class expression that is not a cardinal class expression s, then $N * s$ is defined to be $V(N) \cap P(V(s))$ where $P(V(s))$ is the power set of the denotation of s. In other words, $N * t$ is interpreted as $\{X|X \subseteq t\ \wedge |X| = N\}$, where $|X|$ is the cardinality of the set X.*

all English determiners followed by their nominals are represented as $N * t$ where N is a cardinal class expression and t is a class expression representing the nominal. In the following, we prove that this holds for every and some.
First, the existential singular quantifier 'some' can be considered to denote

$$\{X \in P(s)\ for\ any\ s\ that\ appear\ in\ D/card(X) > 0\}.$$

In this case $some * s = s$ and

$(some * s\ t)$ is true iff $(some\ s\ t)$ is true.

Proof To prove the last claim, assume that $(some * s\ t)$ is true. This implies that $\exists K \in some * s$ such that $K \subset t$. Let $x \in K$ then $x \in s$ since K is a set of s's and $x \in t$ since$K \subset t$. Hence, $x \in s \cap t$ and $(some\ s\ t)$ is true. Conversely, assume $s \cap t \neq \emptyset$ and let $x \in s \cap t$ then $\exists K = \{x\}$ such that $K \in some * s$ and$K \subset t$. Hence, $(some * s\ t)$ is true.

In the same way, 'every' could denote

$$\{X \in P(s)\ for\ any\ s\ that\ appear\ in\ D/card(X) = card(s)\} =$$

$$\{s/s\ is\ a\ class\ expression\}.$$

In this case, $every * s = s$ and

$(every * s\ t)$ is true iff $(every\ s\ t)$ is true.

Again, the proof of the last claim is obvious.

Proof $(every * s\ t)$ is true implies that $\exists K \in every * s$ such that $K \subset t$. But $K = s$. Hence $(every\ s\ t)$ is true. Conversely, if $s \subset t$ then $\exists K = s$ such that $k \in every * s$ and $K \subset t$.

Hence, to simplify things and keep a uniform representation we could drop the operator, $*$, from the representation of any generalised quantifier and be able to represent NPs. However, we decided not to drop the, $*$, from the representation since the inference rules are well-behaved this way. The above was an evidence that NPs have an independent representation and hence are part of CLIP. Moreover, it is important to note that we defined quantifiers in a way so that they do not fall into what quantifiers of FOL do. In $\forall x.raven(x)...$, x spans any entity in the domain. This does not happen in our case since quantifiers are defined in terms of classes and not entities in the domain. Finally, 'many', 'several' and so on are not equivalent to 'some'. We have used $most$, $many$ that are not usually used in a KR. However, their interpretation is like the interpretation of cardinals and cardinals are used in a KR. The reason we used symbols like $many$, $several$, $most$, $more\_than\_one$, is because our aim is to stick to NL-like representation as much as possible.