Reasoning with output from parsing using world knowledge

Anthony Hunter*and Lutz Marten†

June 15, 2000

Keywords: Default logic; Non-monotonic reasoning; Semantic knowledge; Natural language understanding; World knowledge; Commonsense reasoning; Inconsistency handling

Abstract

Many parsers take natural langauge as input, and give some form of parse, in some formal language, as output. However, most parsers use little or no world knowledge to check the commonsense of the output or to facilitate inferencing from the output. Here we look at how world knowledge can be used for commonsense checking and commonsense inferencing with output from parsing. In the framework that we present in this paper, we assume that the output of the parser is in the form of a predicate-argument structure. We encode this output of parsing in classical first-order logic. Commonsense checking involves classical consistency checking of the output with world knowledge whereas commonsense inferencing involves classical thereom proving with world knowledge. The encodation is based on treating each symbol used in the language of the predicate-argument structures as a classical n-ary function symbol, where $n \geq 0$. The encodation also assumes a set of predicate symbols for capturing relationships between predicate-argument structures output by parsers and for capturing world knowledge. We also use these predicates to derive explanations of the meaning of predicate-argument structures output by parsers and hence of the input to parsers. In order to address the need for context-sensitive reasoning with world knowledge, the whole framework is formalized in default logic, which incorporates classical logic. In the following, we provide some motivation for this framework, definitions of the syntax, proof theory, and semantics, and consider, in more detail, formalization of commonsense reasoning with output from parsing.

1 Introduction

In handling natural language in computational linguistics and language engineering, there is a need for formalizing commonsense reasoning. This should draw on world knowledge, and should address the uncertainty, incompleteness, and inconsistencies, inherent in representing and reasoning with such knowledge.

World knowledge is increasingly important in computational linguisitics and language engineering. However, it is a difficult kind of knowledge to represent and reason with. Existing approaches to formalizing world knowledge have tended to combine it with lexical knowledge and have used languages with limited expressibility and in particular they have not addressed the context-dependent nature of the knowledge.

*Department of Computer Science, University College London, Gower Street, London WC1 6BT.
†Department of Linguistics, School of Oriental and African Studies, Thornhaugh Street, London WC1H 0XG

Here we attempt to clarify the nature of lexical and world knowledge. In particular, we focus on the role of world knowledge in reasoning with the output of parsers. For this, we make the following assumptions about lexical and world knowledge. Note, these are only provisional delineations. Later we will define these more fully.

Lexical knowledge is knowledge such as morphology and syntactic categories for words. For each word, it provides constraints on how each word can combine with other words to build phrases and sentences.

World knowledge is knowledge about the relationship between predicate-argument structures (ie. output of the parser) and the real-world. It therefore provides a route for deriving an understanding of the output of the parser (and therefore of the original natural language string input to the parser).

We also assume that the output from the parser is implicitly a parse tree where the phrase structure has been identified. Though, we assume that the parser does not worry about issues of commonsense, so for example, the parser would allow `The cat barked` or `London is in the Alps`. Using these assumptions, the roles of a world knowledgebase may include:

- Providing information for understanding the output of a parser.
- Checking the commonsense of the output of a parser.
- Deriving inferences from the output of a parser.

For our approach to reasoning with output from parsing using world knowledge, we attempt to formalise the following three commonsense activities:

Commonsense checking Here we check that the output from a parser is (1) consistent with the world knowledge and (2) whether it is exceptional, and thereby potentially incorrect, with respect to the world knowledge. For example for (1) `Birmingham in France` is inconsistent with a world knoweldgebase containing knowledge that `Birmingham is in England and not in France`, and for (2) `the bath in the kitchen` is exceptional with respect to the world knowledge that `baths are usually in bathrooms`.

Commonsense inferencing Here we infer formulae that follow from the output of the parser in the context of the world knowledge. For example, from `John ran fast`, we want to infer `John ran`, or from `the hat is under the box` and `the box is under the table`, we want to infer `the hat is under the table`. We may regard inferencing as being done in response to a query or a known information need by the user.

Commonsense explaining Here we explain the meaning of the output of the parse and of inferences derived from the output in the context of the lexical and world knowledge. For example, from `the handbook is heavy`, we want to obtain explanations including `a handbook is a type of book`, `a book is made of paper`, and `the weight of the book is greater than that for normal books`. We may regard explaining as being done in response to a user request for an explanation, or done when there is an expectation by the system of a problem of understanding by the user.

Both lexical and world knowledge is context-dependent knowledge. For example, if we know that a newspaper article is about `oil` by looking at the keywords, it is usually reasonable to derive that it is about `petroleum`, with exceptions such as in contexts about `cooking`. Given the uncertainty involved in this form of context-dependent reasoning, it is a form of default knowledge.

Viewing lexical and world knowledge as default knowledge is part of a general trend of viewing linguistic knowledge as default knowledge [Tho97]. This need for lexical and world knowledge raises significant knowledge representation and reasoning questions. Indeed, developing lexical and world knowledgebases, with automated reasoning technology, has proved to be a complex, and as yet, unresolved problem.

In order to address the need for context-sensitive reasoning with world knowledge, we present a framework formalized in default logic — a context-sensitive logic that incorporates classical logic. In the following, we provide some motivation for this framework, definitions of the syntax, proof theory, and semantics, and consider, in more detail, formalization of commonsense reasoning with output from parsing.

2 Background to lexical and world knowledge

Perhaps the most significant example of a general purpose system for lexical knowledge is WordNet [BFGM91, Mil95]. This is a semantic network containing lexical knowledge on over 90,000 word senses and it is now found to be an increasingly important resource on synonyms, generalizations, and specializations of words, for language engineering applications, such as information retrieval and natural language processing [Voo94, Fel98]. In WordNet, each set of words that are regarded as strict synonyms (i.e. the words can be interchanged in a sentence[1]) is called a synset. The following is an example of a synset.

{`Molotov-cocktail, petrol-bomb, gasoline-bomb`}

Whilst WordNet separates different meanings of the same word by putting the same word in more than one synset, there is no explicit machinery for determining in which context a particular wordsense should be used. Moreover, there is no logical reasoning with the relations in the semantic network.

A more sophisticated system being developed with a deeper knowledgebase on several thousand wordsenses is FrameNet [FA98]. This is a frame-based representation with logical reasoning limited to inheritance, where one frame is an elaboration of another, and composition, where one frame is built-up of other frames as the parts. As with WordNet there is no explicit machinery for determining in which context a particular wordsense should be used.

A richer knowledge representation and reasoning framework is the generative lexicon (GLT) [PB93, Pus95]. This is a form of frame-based representation that is based on specifying the following features for a word:

- Argument structure: providing conventional information about the syntax of a word.
- Event structure: describing a word in terms of states, processes and transitions.
- Qualia structure: describing a word's meaning (or qualia) in terms of the following aspects of the concept it captures:

[1] Of course, the notion of a strict synonym is in general artificial, since very few words can be interchanged in any sentence that they appear in.

  - Constitutive: the relation between the concept and its constituent parts.
  - Formal: information that distinguishes the concept within a larger domain.
  - Telic: information on the purpose and function of the concept.
  - Agentive: factors involved in the origin or generation of the concept.

- Lexical inheritance: determines the relation of a word to other words in a lexicon.

The aim of GLT is to provide richer semantic information about words in addition to basic syntactic and semantic information. In particular, to allow words to be used creatively; they assume new senses in novel contexts. Whilst GLT is a rich framework, the approach is limited in context-dependent reasoning, supporting only a form of inheritance, and in particular it does not support uncertainty management. Furthermore, it lacks a complete definition and the route to implementation remains open. We will return to GLT in Section 9 when we will compare it with our framework.

A much more ambitious approach to capturing world knowledge is CYC. This is a large commonsense knowledgebase intended to be used in a wide range of applications such as reasoning with natural language [LG90, Len95]. It contains information on diverse topics such as bank accounts, animals, transport, and so on, about the world. It includes information such as `animals live for a single solid period of time`, `nothing can be in two places at once`, and `animals don't like pain`. Such information is not usually documented since it is too common, yet for knowledge-based systems that are viable in applications, it could be very useful. The emphasis of the CYC project has been on building up the knowldgebase rather than development of a formal foundation. Problems with CYC include the lack of context-dependent reasoning and formal uncertainty management.

Some machine-readable lexicons have been developed in formal knowledge representation frameworks, supported by logical reasoning. Key formal frameworks are DATR [EG89a, EG89b, EG96], Laurel [Cop92], and persistent default unification (PDU) [LBAC95, LCB96]. These are forms of typed feature structure, a generalization of a form of inheritance hierarchy system (for an introduction to typed feature structures see [Car92]), that offer an efficient and lucid representation of lexical knowledge. However, they are limited formalisms in terms of expressibility and inferential capability, in particular with respect to context-dependent reasoning, and reasoning with inferences.

We will compare our approach with that of typed feature structures in Section 9. However, it is helpful to make a few general comments on typed feature structures here. Implicitly, a typed feature structure is a directed acyclic graph where a set of formulae is attached to each node. The only way to prohibit a default value from a parent node appearing as an entry for a feature at a child node is by an explicit value appearing for that feature at the child node. For example, suppose there is a node for the word `mammal`, with an entry `locomotion = walking` and suppose there is a node for the word `whale` that inherits entries from `mammal` except those that are explicitly lisited with different values such as `locomotion = swimming`. So the only basis for preferring one entry over another is via a simple form of specificity principle: There is no consideration of the general context in which each word is examined, and so no ability to manage the uncertainty that arises. In particular, there is no facility for reasoning in the absence of information or equivalently there is no facility for reasoning that is controlled by ensuring the consistency of specified constraints with assumptions and derivations. Also, there is no facility for preferences based on reasoning about exceptions or for explicit priorities. In addition, there is no integrated logical reasoning with the entries. Rather it is a one-step process whereby a subset of formulae is selected from the knowledgebase. In this way, all preferred inferences need to be made explicit rather than derived through some inference process that involves a sequence of logical steps. This means a lexical knowledgebase based on a typed feature structures behaves more as a sophisticated database than a logical knowledgebase. For discussion of applications for DATR see [EG96], for Laurel see [BCP94], and for PDU see [LBAC95].

To augment the defeasible reasoning in PDU, a form of non-monotonic conditional logic, called commonsense entailment has been integrated [LC98, LC99]. However, preference in commonsense entailment [AM91] is restricted to the specificity principle, and this can force counter-intutive reasoning and inefficient representation of defaults (for examples see section 9.2). Much more significantly is that the non-monotonic reasoning is only defined in terms of manipulating sets of models, and given the cost of storing many models it seems very unlikely that this could offer a computationally viable implementation. Also, there are some more general weaknesses of this logic for capturing default knowledge (for a review of this aspect see [Del98] which includes a comparison with a range of conditional logics for non-monotonic reasoning). We should add that there is no perfect non-monotonic logic — rather each non-monotonic logic has pros and cons for each application. We believe that for the application in this paper, default logic is an optimal choice. We argue why in the following sections. We will compare our approach with commonsense entailment in Section 9.

There are some other proposals to use logics for lexical knowledge representation and reasoning though none offer an explicit framework for reasoning with output from parsing. An epistemic logic has been used as the basis of a framework for reasoning about context for resolving lexical ambiguity [Buv96]. Unfortunately, the reasoning is monotonic and so offers no context-dependent default reasoning. A new variant of default logic has been proposed for reasoning about ambiguity according to context [Poe96]. However, there is no mechanism for determining contexts for a given ambiguous word, and the approach is limited in reasoning about contexts. There are also open questions about the behaviour of the default logic with regard to automated reasoning. Further monotonic formalisms for handling related problems in reasoning about semantic ambiguity, underspecification and context are discussed in [DP96, IV98].

Another modal logic for disambiguation has been proposed in [Fer97] and this does provide default reasoning. This framework is interesting as an abstract framework. But, there are open questions with regard to its practical application. Again the reasoning about contexts is limited and there is no implicit or explicit prioritization mechanism over context-dependent rules for disambiguation. There is a mechanism for determining a context for a given ambiguous word, but this is also limited to explicitly listing the presence or absence of individual words in the sequence of words to the left and right of the ambiguous word for each disambiguation rule. This does not take advantage of the structure that exists in the knowledge.

Finally, default logic has been used as the basis of a framework for capturing context-dependent reasoning with lexical knowledge [Hun99]. Default logic is a first-order logic offering a more expressive formalisation than inheritance hierarchies: (1) First-order formulae about words can be inferred; (2) Preferences over formulae can be based on specificity, reasoning about exceptions, or explicit priorities; (3) Contexts can be reasoned with as first-order formulae; and (4) Contexts can be derived as default inferences. In this framework, the word for which further information is sought is called the query word. The context for a query word is derived from further words, such as words in the same sentence as the query word. These further words are used with a form of decision tree called a context classification tree to identify which contexts hold for the query word. These contexts can be used in default logic to identify lexical knowledge about the query word such as synonyms, antonyms, specializations, meronyms, and more sophisticated first-order semantic knowledge. Standard machine learning algorithm are used to generate the context classification trees. Whilst this framework indicates the value of using default logic for lexical knowledge, there is little consideration of world knowledge and moreover there is a lack of a general framework for integrating lexical and world knowledge for commonsense reasoning with the output of a parser.

In conclusion, we find key problems with current approaches. First, current approaches mix lexical and world knowledge under the banner of lexical knowledge. We believe that separating them will clarify the role of each of them, and thereby allow different formalisms and technologies for each of them and so facilitate engineering them. Second, current approaches do not adequately address the problems of uncertainty, incompleteness and inconsistency, inherent in lexical and world knowledge. In particular, they do not adequately support context-sensitive reasoning with output from parsing. Third, the different types of commonsense reasoning required for handling natural language needs to be clarified. For example, there has been no detailed proposals making commitments on when and how we switch between consistency checking and inferencing.

3 Our perspective on lexical and world knowledge

Although our goal in this paper is to provide an implementable model of the interaction of output from parsing and world knowledge, based on default logic, there are a number of more theoretical assumptions we make about the nature of language, and the knowledge (human) users have in order to use natural language. In this section we make more precise some of the key ideas behind our framework.

3.1 Lexical and world knowledge

In our framework, we restrict the role of the lexicon to providing certain kinds of information required for parsing. The initial step in parsing is to segmentize the input stream and to match segments against lexical items stored in a lexical knowledgebase. If the match is successful, the activated lexical item provides input to combinatorial parsing ("syntax"). In particular, information on the syntactic side of the lexicon consists exclusively of instructions to the combinatorial system, which constructs parse trees in accordance with lexical input.

The lexicon therefore includes information on combinatorial properties of lexical items, their preferred location in the parse tree, and certain locality restrictions. On the other hand, the lexicon does not contain any lexical semantics. In other words, our aim is to locate all lexical meaning, including "inferential roles" and relations such as synonymy and antonmy ("lexical" semantic relations) in the world knowldgebase. Thus, our lexicon includes effectively no semantic information, since we model all lexical semantics in world knowledge.

Following the arguments of Fodor et al [Fod98, FL98], the motivation for this move is (at least) twofold: (1) we want to keep lexical items atomic — if we were to decompose, say, `kill` into a statement based on a small set of constructs, we could not refer to a concept `kill` and reason with it, whereas our world knowlegebase is such that we do want to reason about `kill`; and (2) we do not want to duplicate knowledge — for example, we do not want to state in our lexicon that, say, `bachelors` are unmarried men, only to include in our world knowldgebases that `bachelors` are, among other things, unmarried men.

The role of the world knowledge is for the representation of, and reasoning with, concepts. So semantic information associated with individual lexical items feeds directly into the reasoning about concepts. (Though we should stress that in our framework the focus of the world knowledge is on the predicate-argument structures, rather than individual words in a sequence (see below)). Information about concepts emerge as inferences from the world knowledge.

Before turning to the question of how this world knowledgebase is structured, we take a closer look at the output of the parser in order to see what sort of input we supply to our commonsense reasoning system.

3.2 Output from the parser

As indicated above, we assume that a syntactic parser takes lexical input in the order in which it occurs in an input sentence and exploits lexical information to build some kind of parse tree. It is these parse trees that are the vehicles for semantic interpretation of the input sentences. In this paper we are not concerned with the question of how these parse trees are interpreted, or for that matter if they are interpreted at all, independent of world knowledge. Rather we assume that world knowledge is in principle available to the interpretation process at any stage after the lexicon, and it is this interaction which we address.

There are at least three areas where world knowledge is relevant for interpretation; reference assignment, disambiguation, and enrichment. However, the model we present here is in several ways restricted. First, although world knowledge and parsing are in principle interleaved, we concentrate here in the use of world knowledge on completed parse trees. In other words, interpretation is "serial". This is an assumption which simplifies the issues and might turn out to be inappropriate. The assumption is fine for us here, though, because we do not address problems of reference assignment or of resolution of structural ambiguities. Where needed, we take these tasks to be completed.

This means that we are here mainly concerned with enrichment. More concretely, we address the question of how concepts are identified and interpreted in their sentential context. This leaves us with two areas of underspecification in the output of the parser which need world knowledge enrichment in interpretation: the formation of concepts in working memory formed from concepts encoded in the world knowledgebase on the one hand and the sentential context on the other, and secondly, the establishment of predicate-argument structures. This later point might be somewhat controversial — we assume that the parser does not give fully specified predicate-argument structures as output. Rather, distinctions between arguments and adjuncts are established with recourse to world knowledge. In fact, since both concept formation and the establishment of predicate-argument structures involve the application of the default logic we develop in the next section, they are for us just subcases of the same process.

To sum up, in our approach we take predicate-argument structures output from a parser as input. The predicate-argument strutures will be underspecified representations of the meaning of the sentences input to the parser. By underspecified, we mean that whilst the predicate-argument structures have an operational semantics, they lack sufficient meaning with respect to real-world concepts. The output of our system will be inferences from the world knowledgebase together with each predicate-argument structure and these inferences will be intended to decrease the underspecification. We do not address questions of reference assignment, definiteness, or quantification. With this in mind, we can now turn to the question of how to model concepts.

3.3 Concepts

A concept is a mental representation we have of say **dog**, **king**, **fish**, **red**, **love** or **hate**. Everything, we know about say **dog**, **king**, **fish**, **red**, **love** or **hate** is found under the appropriate concept. There are no restrictions on the kinds of information found under a concept. For example, information associated with the concept **John Wayne** might include information about the visual representation of the man, of the way he speaks, and of the kinds of movies he appeared in. Information under **Issey Miyake** might include information about an olfactoric representation. Furthermore, there is no restriction to only include "definitional" information. Rather under a concept, definitional, descriptive, associative, emotive, sensory, etc information can be freely entered. We further assume that there are many concepts, in particular that there are many more concepts than words (cf. Sperber and Wilson [SW97]). Since a concept is mental representation, all we can hope to do in a computer is to have information about concept — in other words, a model of a concept. We believe this view is consistent with that of [Car99, Fod98].

Against this theoretical background we propose to model concepts by employing default logic — a set of default rules that provides this representation of concepts is called world knowledge. In world knowledge, a concept is described by general statements with exceptions. This may include short simple statements with many potential exceptions. It may also include longer complex statements with fewer potential exceptions. The more general statements will be associated with broader wordsenses. The more specific statements will be associated with narrower wordsenses. This approach is contrary to that of GLT where a minimum number of wordsenses for each word is maintained, and then further wordsenses are forced using type-coercion when combined with other words.

Now, an important practical question is how do we construct a world knowledgebase. We can approach this by considering a couple of examples for modelling the concepts **water** and **love**. We will just present these informally though in our framework we use default logic.

water	**love**
Fish swim in water	Love is a feeling
Humans can swim in water	Love is positive
Humans can drink water	Love is stronger than liking
Water comes in bottles for drinking	Love is often directed to another person
Water comes in pools for swimming	Love can also be directed to things
Water comes in oceans, seas, and rivers	Love is often irrational
Some people refers to it as H_2O	Love often goes with being nice
Humans use water for washing	Love is often associated with poetry
Water feels wet	Love is often associated with romance

It can be seen that the organization of concepts is not especially systematic, efficient, or clear-cut. The same information can be found under several concepts (eg. "fish swim in water" appears in the concept **fish**, **swim**, and **water**). However, we can see that information about a concept is represented by a set of statements. In our framework, each of these statements is represented in default logic, and so to reiterate, information about individual concepts emerges as inferences (classical logic formulae) from the default rules in the world knowledgebase.

3.4 Commonsense reasoning

In Section 1, we introduced the requirement for commonsense reasoning, together with the notions of commonsense checking, commonsense inferencing, and commonsense explaining. Now that we have clarified our perspective on lexical and world knowledge, we return to the nature of commonsense reasoning with the output of a parser. Two key aims of our approach are:

- to allow as much semantic information about a string of words to be derived irrespective of whether all the words are known, used consistently, used unusually, or whether the whole string makes sense with respect to the world knowledge;
- to structure our knowledgebase so that we have both general and specialized defaults describing concepts — in this way even with a minimal understanding about a string of words some inferences might be drawn.

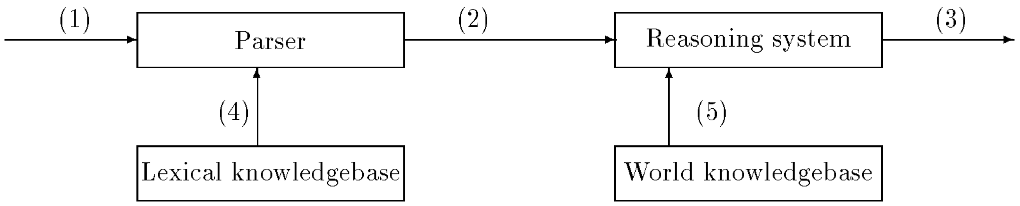

Figure 1: A simple dataflow diagram to indicate the relationships between the parser, the reasoning system, and the lexical and world knowledgebases where: (1) is natural language input to the parser; (2) is output from the parser in the form of predicate-argument structures; (3) are results of commonsense checking, inferencing and explaining; (4) is lexical knowledge provided to the parser; and (5) is world knowledge provided to the reasoning system.

Our framework for commonsense reasoning with world knowledge is logic-based. We will give a formal definition of the language for representing knowledge and of the proof theory for reasoning with that knowledge. In Figure 1, we give an overview of how information flows between the various components in a system.

The final question we consider in this section is that of the correctness of the commonsense reasoning system. By correctness, we mean that for any input to the commonsense reasoning system, the output (the inferences) is acceptable with respect to the input. As we will define in the following sections, each input is a predicate-argument structure obtained from a parse tree, and the output is a set of classical logic formulae. Clearly in practice there are a great many acceptable input/output pairs, and there is no systematic way of generating them. Hence, the correctness of a commonsense reasoning system can only be verified empirically.

In the next Section, we provide an overview of default logic, and then in Section 5, we formalize our framework for commonsense reasoning using default logic.

4 Overview of default logic

Default logic was proposed by Reiter [Rei80], and good reviews are available (see for example [Bes89, Bre91, Ant97, BDK97, Sch98]).

In default logic, knowledge is represented as a **default theory**, which consists of a set of first-order formulae and a set of default rules for representing default information. A **default rule** is of the following form, where α, β and γ are classical formulae,

$$\frac{\alpha : \beta}{\gamma}$$

The inference rules are those of classical logic plus a special mechanism to deal with default rules: Basically, if α is inferred, and $\neg\beta$ cannot be inferred, then infer γ. For this, α is called the pre-condition, β is called the justification, and γ is called the consequent. In this paper, if α and β are large formulae, then we may present a default rules as follows:

$$\frac{\alpha \atop : \beta}{\gamma}$$

Informally, an **extension** is a maximally consistent set of inferences (classical formulae) that follow from a default theory. Default logic incorporates classical logic. Hence, all classical inferences from the classical information in a default theory is derivable (if there is an extension). The default theory then augments these classical inferences by default inferences derivable using the default rules.

More formally, we introduce the operator Γ that indicates what conclusions are to be associated with a given set E of formulae, where E is some set of classical formulae. Let (D, W) be a default theory, where D is a set of default rules and W is a set of classical formulae. Let Cn be the function that for a set of formulae returns the set of classical consequences of those formulae. For this, $\Gamma(E)$ is the smallest set of classical formulae such that the following three conditions are satisfied.

1. $W \subseteq \Gamma(E)$
2. $\Gamma(E) = Cn(\Gamma(E))$
3. For each default in D, where α is the pre-condition, β is the justification, and γ is the consequent, the following holds:

$$\text{if } \alpha \in \Gamma(E) \text{ and } \neg\beta \notin E \text{ then } \gamma \in \Gamma(E)$$

Let us call E the satisfaction set, and $\Gamma(E)$ the putative extension. Once $\Gamma(E)$ has been identified, E is an extension of (D, W) iff $E = \Gamma(E)$. If E is an extension, then the first condition ensures that the set of classical formulae W is also in the extension, the second condition ensures the extension is closed under classical consequence, and the third condition ensures that for each default rule, if the pre-condition is in the extension, and the justification is consistent with the extension, then the consequent is in the extension.

We can view E as the set of formulae for which we are ensuring consistency with the justification of each default rule that we are attempting to apply. We can view $\Gamma(E)$ as the set of putative conclusions of a default theory: It contains W, it is closed under classical consequence, and for each default that is applicable (i.e. the precondition is $\Gamma(E)$ and the justification is satisfiable with E), then the consequent is in $\Gamma(E)$. We ask for the smallest $\Gamma(E)$ to ensure that each default rule that is applied is grounded. This means that it is not the case that one or more default rules are self-supporting. For example, a single default rule is self-supporting if the pre-condition is satisfied using the consequent. The test $E = \Gamma(E)$ ensures that the set of formulae for which the justifications are checked for consistency coincides with the set of putative conclusions of the default theory. If $E \subset \Gamma(E)$, then not all applied rules had their justification checked with $\Gamma(E)$. If $\Gamma(E) \subset E$, then the rules are checked with more than is necessary.

Default rules can be presented with unbound variables in the precondition, justification, or consequent. These default rules are schema and so can be bound with any term in the default theory. The scope of each unbound variable is limited to the rule in which it occurs. In this way, each schema is just an abbreviation for a set of default rules without unbound variables.

To illustrate the context-dependent reasoning with default logic, consider the default theory composed of the following set of default rules. Note `X` is unbound, and so these default rules are schema that can be bound with either of the constants `fred` or `sid` appearing in the example.

$$\frac{\texttt{Bird(X)} \ : \ \neg(\texttt{Ostrich(X)} \vee \texttt{Penguin(X)})}{\texttt{Fly(X)}}$$

$$\frac{\texttt{Penguin(X)} \ : \ \neg\texttt{Fly(X))}}{\neg\texttt{Fly(X)}}$$

$$\frac{\texttt{Ostrich(X)} \ : \ \neg\texttt{Fly(X))}}{\neg\texttt{Fly(X)}}$$

and the following set of atoms,

$$\begin{array}{c}\texttt{Bird(fred)} \\ \texttt{Bird(sid)} \\ \texttt{Ostrich(sid)}\end{array}$$

From this default theory, we obtain just one extension which contains `Fly(fred)` and ¬`Fly(sid)`.

Basing the framework on default logic brings advantages. Default logic provides a rich (expressive) and lucid representation for context-dependent reasoning and handling of exceptions. It is an efficient representation in terms of space, and it is a well-understood formalism for representing uncertain information, and it has strong theoretical foundations. In addition, there are prototype implementations of inference engines for default logic that can be used for developing default logic knowledge-bases [Nie94, LS95, Sch95, SBN96, BS98, NS98].

Examples of using default logic in computational linguistics and language engineering include for reasoning about presuppositions [Mer91, Ger95], for anaphoric resolution [Qua93], for reasoning about the notion of "aboutness" for text handling [Hun95, Hun96], and for context-dependent reasoning with lexical knowledge [Hun97, Hun99].

5 Framework for reasoning with output from parsing

This framework for reasoning with output from parsing is based on classical first-order logic and default logic. Indeed, recall default logic includes classical logic. In this section, we delineate the use of classical logic in the framework, and introduce term substitution.

5.1 Classical logic in the framework

We assume the output from the parser is in the form of a predicate-argument structure. In this paper, we restrict the outputs to being ground (so we do not consider any form of variable or quantifier in the predicate-argument structure). Each predicate symbol and function symbol in the language of the predicate-argument structure is represented as a function symbol in our framework.

Definition 5.1 *Let S be a set of logical symbols used in predicate-argument structures. Let F be a set of function symbols. For each symbol α in S, there is a n-place function symbol α in F for each $n \geq 0$. Recall that a constant is a zero-place function. Let V be a set of variable symbols. Let T be the set of terms formed from F in the usual inductive definition for classical logic:*

If $\alpha \in F$ and $\beta_1, ..., \beta_n \in T$, then $\alpha(\beta_1, ..., \beta_n) \in T$
If $\alpha \in F$, then $\alpha \in T$
If $\alpha \in V$, then $\alpha \in T$

We also assume the usual definition of the notion of subterm.

If S is the set {`black, is, the_cat, sat`}, then T includes:

```
black(the_cat)
is(the_cat,black)
sat(the_cat)
```

Definition 5.2 *Let* $terms : T \longrightarrow \wp(T)$ *be a function such that (1) if* $\alpha \in F$ *or* V*, then* $terms(\alpha) = \{\alpha\}$*, and (2)* $terms(\alpha(\beta_1, .., \beta_n)) = terms(\beta_1) \cup ... \cup terms(\beta_n)$*. So* $\phi \in terms(\psi)$ *iff* ϕ *is a subterm of* ψ*.*

Definition 5.3 *For* $\alpha \in T$*, if* $\phi \in terms(\alpha)$*, let* $\alpha[\phi/\psi]$ *be the result of replacing all occurrences of* ϕ *in* α *with* ψ*. If* $\phi \notin terms(\alpha)$*, then* $\alpha[\phi/\psi]$ *is* α*.*

Example 5.1 *If* $\texttt{a(c(b(g,h)))} \in T$*, then* $\texttt{a}(\texttt{c}(\texttt{b}(\texttt{g},\texttt{h})))[\texttt{c}(\texttt{b}(\texttt{g},\texttt{h}))/\texttt{d}(\texttt{g},\texttt{h})]$ *is* `a(d(g,h))`*.*

We are developing a general framework in this paper, and so we aim to minimize the assumptions we have to make about the output from the parser. Essentially, the minimum we require is a predicate-argument structure output by the parser. In order to focus the presentation in this paper, we have not considered issues such as quantification or anaphoric resolution. However, we can potentially exploit the output from more sophisticated parsers that give information on issues such quantification, co-reference, co-ordination, and negation.

From the predicate-argument structure, we identify an element of T to represent the predicate-argument structure. For example, it could be recursively defined using types such as in LDSNL [KMG00], or using the lexical head and lexical arguments as in X-bar syntax [Jac77, Gor95]. The choice of an element of T is not necessarily unique.

Consider the input `John saw the book under the table` to the parser. The output for this could be represented as either of the following terms.

```
saw(John,under(the_book,the_table))
saw(John,the_book,under(the_table))
```

Definition 5.4 *Let* P *be a set of predicate symbols and let* A *be a set of atoms formed from* P*, and* T *using the usual inductive definition for classical logic:*

$$\textit{If } \alpha \in P \textit{ and } \beta_1, .., \beta_n \in T, \textit{ then } \alpha(\beta_1, .., \beta_n) \in A.$$

In this paper we will use the notational system of function symbols starting with lower case letters and predicate symbols starting with upper case letters.

If `Holds` is a predicate symbol, then assuming the example T given above, atoms include:

```
Holds(black(the_cat))
Holds(is(the_cat, black))
Holds(sat(the_cat))
```

Definition 5.5 *Let $\vee$, $\wedge$, $\neg$, and $\rightarrow$, be the usual logical symbols. Let L be the usual set of logical formulae that can be formed from the set of atoms A, the logical connectives, and the logical quantifiers, $\forall$ and $\exists$ (universal and existential quantifiers, respectively) using the usual inductive definition for classical logic.*

In order to formalize the role of the world knowledge, and of the reasoning, we adopt the following classes of predicate symbols.

World predicate symbols The world knowledgebase provides understanding of terms using a vocabulary including `PhysicalObject`, `Action`, ... We do not assume a controlled vocabulary. Rather any symbol in S (ie. the symbols that can occur in the output of the parser) can be a predicate symbol in the world knowledge. So for example `Animal`, `Cat`, `Red`, `Sank`, `Happy` and `Human` are all predicate symbols. Whilst we do not have a controlled vocabulary, we do assume some axioms for some predicates in the world knowledge, there are many possibilities for axioms including transitvity, reflexivity, asymmetry, linearity, or some more specialized property, for some predicate symbols, or constraints between a number of predicate symbols.

Structural predicate symbols Further predicate symbols are needed to represent the output of the parser and to reason with the structure of that output. These include the monadic predicate `Holds`, where `Holds`(α) is an inference if α is the output of the parser or if α is a commonsense inference from the output of the parser. This class of predicates also includes the binary predicate `Sub`, where any occurrence of the first term as a subterm can be substituted by the second term (see Section 5.2), and the monadic predicate `PartHolds`, where if `Holds`(α) is an inference, and β is a subterm of α, then `PartHolds`(β) is an inference, and the binary predicate `PartOf` $\subseteq T \times T$ where the second argument is a subterm of the first argument. We introduce further structural predicate symbols later.

The `PartHolds` and `PartOf` predicates offer the significant ability to manipulate the output of the parse. They are used to help minimize the assumptions made about the parsing. Note, for all α, α', if `PartOf`(α, α') holds, and `PartHolds`(α) holds, then `PartHolds`(α').

Example 5.2 *Let* `Holds(a(b,c(d)))` *hold. So* `PartHolds` *relations include*

$$\begin{array}{c}\texttt{PartHolds}(\texttt{a})\\ \texttt{PartHolds}(\texttt{c}(\texttt{d}))\\ \texttt{PartHolds}(\texttt{d})\end{array}$$

Similarly, `PartOf` *relations include:*

$$\begin{array}{c}\texttt{PartOf}(\texttt{a}(\texttt{b},\texttt{c}(\texttt{d})),\texttt{c}(\texttt{d}))\\ \texttt{PartOf}(\texttt{a}(\texttt{b},\texttt{c}(\texttt{d})),\texttt{b})\end{array}$$

We can consider structural predicates as being meta-level relations, and world predicates as object-level predicates. However, in taking this view, we don't make any general assumptions (such as with regard to negation) about translations between the object-level and meta-level. Though we can use default rules to encode translation rules. We discuss this later.

5.2 Term substitution for reasoning

In addition to the proof theory, we also want to use term subsitution for generating new terms from existing terms. Consider the natural language input `John banked the plane`. There are good reasons to represent this as `banked(John,the_plane)`. However, we may wish to also represent it as `turned(in(the(air(John,the_plane))))`, and so we may equate `John banked the plane` with `John turned the plane in the air`. We can use default rules in world knowledge to make them interchangeable. In other words, via reasoning we will allow term substitution in certain contexts.

Definition 5.6 *If* `Holds(`α`)` *is an inference from the world knowledgebase and* $\phi \in terms(\alpha)$ *and* $\mathtt{Sub}(\phi, \psi)$ *is an inference from the world knowledgebase and* $\beta = \alpha[\phi/\psi]$*, then* `Holds(`$\beta$`)` *is an inference from the world knowledgebase.*

Definition 5.7 *If* `PartHolds(`α`)` *is an inference from the world knowledgebase, and* $\phi \in terms(\alpha)$*, and* $\mathtt{Sub}(\phi, \psi)$ *is an inference from the world knowledgebase, and* $\beta = \alpha[\phi/\psi]$*, then* `PartHolds(`$\beta$`)` *is an inference from the world knowledgebase.*

So returning to the input `John banked the plane`, which could be output as $\mathtt{banked}(\mathtt{John}, \mathtt{the\_plane})$ by the parser.

$$\frac{\mathtt{Aircraft(Y)\ :\ Pilot(X)}}{\mathtt{Sub}(\mathtt{banked}(\mathtt{X}, \mathtt{Y}), \mathtt{turned}(\mathtt{in}(\mathtt{the\_air}(\mathtt{X}, \mathtt{Y}))))} \qquad (1)$$

Assuming the fact $\mathtt{Aircraft}(\mathtt{the\_plane})$, and using the Facilitation Default rule (1), we obtain the following as an inference.

$$\mathtt{Sub}(\mathtt{banked}(\mathtt{John}, \mathtt{the\_plane}), \mathtt{turned}(\mathtt{in}(\mathtt{the\_air}(\mathtt{John}, \mathtt{the\_plane}))))$$

So we can obtain both $\mathtt{Holds}(\mathtt{banked}(\mathtt{John}, \mathtt{the\_plane}))$ and $\mathtt{Holds}(\mathtt{turned}(\mathtt{in}(\mathtt{the\_air}(\mathtt{John}, \mathtt{the\_plane}))))$ as inferences.

5.3 Knowledgebases for reasoning

We divide the default rules into sets according to the predicate symbols used in the consequent.

Facilitation defaults This is the set of default rules for deriving inferences involving structural predicate symbols (such as `Holds`, `PartOf`, `PartHolds`, and `Sub`).

Translation defaults This is the set of default rules for deriving inferences involving world predicate symbols from `Holds` and `PartHolds` inferences.

World defaults This is the set of default rules that represents information about the world that is normally correct in some contexts. These rules are for deriving inferences involving the world predicate symbols.

In addition, we assume the following set of classical formulae.

World facts This is a set of classical formulae that represents information about the world that we assume is correct.

The choice of the actual formulae in each of these sets depends on the application of the commonsense reasoning system. Using these sets of formulae, we define the world knowledge as follows:

Definition 5.8 *The world knowledge is a default theory* (Δ, Γ) *where* Δ *is the union of Facilitation defaults, Translation defaults, and World defaults, and* Γ *is the World facts.*

Whilst most of the examples and discussion in this paper are focussed on commonsense reasoning with parses of sentences, we can also consider individual words, or phrases, or sequences of sentences in the same way.

5.4 Semantics for reasoning

Before we consider the semantics for our framework, we need to decouple two types of semantics, and to provide links with the notion of concepts discussed in Section 3.3:

Logico-semantics This type of semantics usually provides a formal conceptualization of some class of models together with a means to evaluate individual formulae with respect to the models, and to determine whether a particular formula is entailed by a set of assumptions. In an abstract way, logico-semantics provides a declarative understanding of formulae, and the behaviour of formulae. Relationships between logic-semantics and real-world concepts seem to be quite secondary in general. This is particularly so if we use logic as a meta-level formalism for describing linguistic information, as in the case for structural predicates.

Lingua-semantics This type of semantics usually provides a means to describe or delineate a word or string of words using real-world concepts. The ways that such a semantics is delivered ranges from using other strings of words through to establishing some direct link with real-world objects or phenomena. In Section 3.3, we described modelling of concepts, and of how we capture concepts in default logic. In the rest of this paper, we will see how we can use world defaults to model concepts, and that the inferences from the world knowledge gives information about the concepts in some sentential context, and that this information constitutes a form of lingua-semantics.

A key challenge is to develop and use logico-semantics to deliver lingua-semantics in an automated reasoning systems.

In our framework, we can regard the logico-semantics as essentially being classical. So for each classical formula in the set of inferences, we have some evaluation of that formula in terms of true and false. This approach to logico-semantics naturally extends to the class of default theories that we use in this paper (for details of semantics for default logic see [Eth88]).

However, in our framework we can partition the interpretations given to the predicate symbols, so that the world predicates and structural predicates are regarded differently.

Interpreting structural predicates Essentially, structural predicates are meta-level predicates. The objects in the model are just terms formed from the symbols in the language output from the parser.

Interpreting world predicates Essentially, world predicates are object-level predicates. The objects in the model represent entities in the real-world. Each function symbol is interpreted as a mapping from a set of tuples of objects to a set of objects. Each predicate symbol is a subset of objects and so represents a subset of entities in the real-world.

We can consider all the predicate symbols as contributing towards delivering lingua-semantics for a given input to the parser. However, the world predicates provide the richest route to giving meaning or understanding to the output of the parser.

6 Commonsense checking

Commonsense checking involves (1) checking consistency of an output from a parser with the world knowledge; and (2) checking normality of an output from a parser by reference to the world knowledge. The aim of commonsense checking is to flag inconsistencies and irregularities. This may take the form of meta-level statements such as $\texttt{InconsistentWithWorldKnowledge}(\alpha)$ where α is an output from the parser.

6.1 Checking consistency

Suppose α is the output of the parser, then $\texttt{Holds}(\alpha)$ is an inference. Furthermore, suppose β is a subterm in α, then $\texttt{PartHolds}(\beta)$ is an inference. For commonsense checking, we take each of $\texttt{Holds}(\alpha)$ and $\texttt{PartHolds}(\beta)$ and check their consistency with the world and lexical knowledge according to the following definition.

Definition 6.1 *Let (Δ, Γ) be a default theory for world knowledge. Also let $\alpha \in T$ be the output of parsing and for $\beta \in T$, let $\texttt{PartHolds}(\beta)$ hold. The output of the parsing is consistent with the world knowledge iff the following two conditions hold:*

(1) there is an extension of the default theory $(\Delta, \Gamma \cup \{\texttt{Holds}(\alpha)\})$
(2) there is an extension of the default theory $(\Delta, \Gamma \cup \{\texttt{PartHolds}(\beta)\})$

By considering PartHolds relations, we are examining the internal consistency of the output of the parser. It offers a more direct route to examining the contents of a sentence than looking at all the Holds inference that could be obtained from the output of the parser.

In Checking example 1, we consider consistency checking with facts in the world knowledgebase, and then in Checking example 2, we consider consistency checking with some temporal knowledge.

6.1.1 Checking example 1

In this example, we assume $\texttt{CityAndCountry} \subseteq \texttt{City} \times \texttt{Country}$, and City is a set of cities and Country is a set of countries. We assume world facts for subsequent examples can be defined similarly. Suppose the world facts includes:

$$\begin{array}{c}\texttt{CityAndCountry(London,England)}\\ \neg\texttt{CityAndCountry(London,France)}\\ \texttt{City(London)}\\ \texttt{City(Birmingham)}\\ \texttt{Country(England)}\\ \texttt{Country(France)}\end{array}$$

Now consider the input `John lives in London in England`. Let the parser output be as follows.

$$\mathtt{lives(John, in(London(in(England))))}$$

Since `Holds(lives(John,in(London(in(England)))))` and so `PartHolds(in(London(in(England)))` are inferences, we have `CityAndCountry(London,England)` as an inference from the Translation Default Rule (2), and this inference is consistent with the world facts. In other words, there is an extension of the world knowledge with `PartHolds(in(London(in(England))`.

$$\frac{\mathtt{PartHolds(in(X(in(Y)))) \wedge City(X) \wedge Country(Y)\ :\ CityAndCountry(X,Y)}}{\mathtt{CityAndCountry(X,Y)}} \tag{2}$$

Now consider the input `John lives in Birmingham in France` giving the following output from the parser:

$$\mathtt{lives(John, in(Birmingham(in(France))))}$$

Since `PartHolds(in(Birmingham(in(France))))` is an inference, from the Translation Default Rule (2) we have `CityAndCountry(Birmingham,France)` as an inference and this is inconsistent with the world facts, assuming that we have ¬`CityAndCountry(Birmingham,France)`. Either ¬`CityAndCountry(Birmingham,France)` is explicit in the world facts, or we apply the closed world assumption [Rei78] to the world facts. The closed world assumption allows for the derivation of a negative literal $\neg\alpha$ from a set of literals Φ if α cannot be derived from Φ.

6.1.2 Checking example 2

Now we consider more complicated inconsistencies that can arise. The following sentence appears to imply a temporally impossible situation.

`John lived at the house for one year during March`

Suppose we obtain the following parse:

`lived(John,at(the_house),for(one_year(during(March))))`

From this, we obtain the following inference:

`Holds(lived(John,at(the_house),for(one_year(during(March)))))`

We then obtain `PartHolds(one_year(during(March)))`. Assume that `TimePeriod(one_year)` and `TimePeriod(March)` are in the world facts.

$$\frac{\mathtt{PartHolds(X(during(Y)) \wedge TimePeriod(X) \wedge TimePeriod(Y)\ :\ PartHolds(during(X,Y))}}{\mathtt{PartHolds(during(X,Y))}} \tag{3}$$

$$\frac{\texttt{PartHolds(X(during(Y)))} \ : \ \texttt{PartHolds(in(X,Y))}}{\texttt{PartHolds(in(X,Y))}} \tag{4}$$

Then using the Facilitation Default Rules (3) and (4), we obtain the following inferences.

$$\texttt{PartHolds(during(one\_year,March))}$$
$$\texttt{PartHolds(in(one\_year,March))}$$

$$\frac{\texttt{PartHolds(in(X,Y))} \wedge \texttt{TimePeriod(X)} \wedge \texttt{TimePeriod(Y)} \ : \ \texttt{Subinterval(X,Y)}}{\texttt{Subinterval(X,Y)}} \tag{5}$$

From the Translation Default Rule (5), we obtain `Subinterval(one_year,March)`. Now we obtain a contradiction, assuming that ¬`Subinterval(one_year,March)` is in the world facts.

This approach is intended to find inconsistencies in the components of a sentence, rather than evaluate a sentence as a whole. So consider the sentence `He cannot have lived at the house for one year during March.` Here again we obtain the inconsistency regarding the component `for one year during March`, but this does not mean that the sentence as a whole is inconsistent.

6.2 Checking normality

Describing some information as normal or abnormal is highly subjective and context-dependent. However, it is possible to loosely categorize sentences, in the context of the sentence itself, as containing information that appears normal (either redundant or regular), or abnormal, by appealing to commonsense use of world knowledge, as described as follows:

Regular information This is information that appears normal when compared with world knowledge. For example, `John buttered the toast with a knife` or `John buttered the toast with margarine`.

Redundant information This is information that appears normal but contains some redundancy when compared with world knowledge. For example `John is singing in the bath in the bathroom` or `John buttered the toast with butter`. Since `baths` are normally in `bathrooms` and `buttering toast` normally involves `butter` both sentences use redundant words.

Abnormal information This is information that appears abnormal, or unusual, when compared with world knowledge. For example, `John is singing in the bath in the kitchen` or `John buttered the toast with a fork`. Here a `bath` is not normally in a `kitchen`, and similarly `toast` is not normally `buttered with a fork`.

In information theoretic terms, we can view three kinds of information as follows:

- Regular information is information that is one of a number of options of roughly equal probability of holding and so for the reader of the sentence to the parser, it is unlikely to have been guessed to be true and so the reader would only know about it if it was explicitly communicated in the sentence.

- Redundant information is information that has a very high probability of holding and so for the reader of the sentence input to the parser, it could have been reliably guessed to be true without being explicitly communicated in the sentence.
- Abnormal information is information that has a very low probability of holding, and there are other options with significantly higher probability of holding. So for the reader of the sentence to the parser, it is unlikely to have been guessed to be true and so the reader would only know about it if it was explicitly communicated in the sentence.

For checking normality, we assume a set of Normality Default Rules — these are a particular subset of the Default Rules. The special character of Normality Default Rules is that the inferences are likely to be true with a very high probability. Remember, in general, an inference from a default rule is only the "best possible" inference in the current context. It doesn't necessarily mean that the inference is likely to be true with a very high probability.

To use these with a particular output of the parser, say α, we take an inferentially weaker formlua β, together with the Normality Default Rules, and see if we obtain the same information as is contained in α.

- If we infer predicates from the Normality Default Rules that contradict with those that follow directly from the original sentence, then the original sentence contains abnormal information.
- If we infer predicates from the Normality Default Rules that also follow directly from the original sentence, then the original sentence contains redundant information.
- If none of the predicates inferred from the Normality Default Rules are contained in nor contradict the set of inferences obtained directly from the original sentence, then the sentence contains regular information.

As an example of a Normality Default Rule, consider the following, Default Rule (6), where we effectively state that with very high probability if `X butters the_toast`, then `X does so with a_knife`.

$$\frac{\texttt{Butter}(\texttt{X},\texttt{the\_toast})\ :\ \texttt{Butter}(\texttt{X},\texttt{the\_toast},\texttt{with}(\texttt{a\_knife}))}{\texttt{Butter}(\texttt{X},\texttt{the\_toast},\texttt{with}(\texttt{a\_knife}))} \tag{6}$$

We can capture regular information in default rules by giving the options as a disjunction of possibilities in the consequent. Consider for example the default Rule (7) below, where the consequent gives all the possibilities that Y is likely to be. Of course, with lower probability, it could be something else, but the exceptional cases can be accommodated by the justification being unsatisfiable in the context.

$$\frac{\begin{array}{c}\texttt{Butter}(\texttt{X},\texttt{Y}) \wedge \texttt{Person}(\texttt{X}) \wedge \neg\texttt{Instrument}(\texttt{Y}) \\ :\ (\texttt{Y} = \texttt{toast}) \vee (\texttt{Y} = \texttt{bread}) \vee (\texttt{Y} = \texttt{potato}) \vee (\texttt{Y} = \texttt{cracker}) \vee (\texttt{Y} = \texttt{crumpet})\end{array}}{(\texttt{Y} = \texttt{toast}) \vee (\texttt{Y} = \texttt{bread}) \vee (\texttt{Y} = \texttt{potato}) \vee (\texttt{Y} = \texttt{cracker}) \vee (\texttt{Y} = \texttt{crumpet})} \tag{7}$$

In this way, we can capture the alternatives that normally arise. Such defaults can also play a useful role in commonsense inferencing and commonsense explaining.

6.2.1 Checking example 3

First assume the Normality Default Rules include the Facilitation Default Rule (9) and the World Default Rule (6). Now consider the output `butter(John,the_toast,with(a_knife))` from the parser. Here the verb `butter` is represented as a three-place function where the first argument is the subject, the second is the object, and the third is the adjunct. So from the output from the parser we have the following.

$$\mathtt{Holds(butter(John, the\_toast, with(a\_knife)))}$$

Using the Translation default Rule (8), we obtain `Butter(john,the_toast,with(a_knife))` from the original sentence.

$$\frac{\mathtt{Holds(butter(X,Y,Z))\ :\ Butter(X,Y,Z)}}{\mathtt{Butter(X,Y,Z)}} \tag{8}$$

Now assuming the world facts `Human(John)` and `Food(the_toast)`, using the Facilitation Default rule (9) we obtain `Holds(butter(John,the_toast))`. The inference `Holds(butter(John,the_toast))` is inferentially weaker than the original `Holds(butter(John,the_toast,with(a_knife)))`. So we use it to reason with the Normality Default Rules.

$$\frac{\mathtt{Holds(butter(X,Y,Z))} \wedge \mathtt{PartOf(X,X')} \wedge \mathtt{PartOf(Y,Y')} \wedge \mathtt{Human(X')} \wedge \mathtt{Food(Y')}\ :\ \mathtt{Holds(butter(X,Y))}}{\mathtt{Holds(butter(X,Y))}} \tag{9}$$

Then using the Translation default Rule (10), we obtain `Butter(John,the_toast))`.

$$\frac{\mathtt{Holds(butter(X,Y))\ :\ Butter(X,Y)}}{\mathtt{Butter(X,Y)}} \tag{10}$$

Then using the World Default Rule (6), we obtain `Butter(John,the_toast,with(a_knife))`. So the original sentence contains redundant information.

6.2.2 Checking example 4

Consider the output `butter(John,the_toast,with(a_spade))` from the parser. So from the output from the parser we have the following.

```
Holds(butter(John,the_toast,with(a_spade)))
```

Using Translation default Rule (8), we get `Butter(John,the_toast,with(a_spade))` from the original sentence, and using Facilitation Default rule (9), we get `Holds(butter(John,the_toast))`.

Now using the Translation default Rule (10), we obtain `Butter(John,the_toast)`, and then using the World Default Rule (6), we obtain `Butter(John,the_toast,with(a_knife))`. Now assume that the following is in the world facts.

$$\forall \texttt{X,Y}\ \texttt{Butter(X,Y,with(a\_knife))} \leftrightarrow \neg\texttt{Butter(X,Y,with(a\_spade))}$$

So in this case we obtain a contradiction, and so the original sentence contains abnormal information.

7 Commonsense inferencing

Essentially, commonsense inferencing is formalised as the derivation of extensions of default theories representing world knowledge.

Definition 7.1 *Let (Δ, Γ) be the world knowledge, and $\alpha \in T$ is the output of parsing. For $\beta \in T$, β is a commonsense inference from $(\Delta, \Gamma \cup \{\texttt{Holds}(\alpha)\})$ iff there is an extension of $(\Delta, \Gamma \cup \{\texttt{Holds}(\alpha)\})$ and β is in the extension.*

7.0.3 Inferencing example 1

Consider the input `Bill saw the man with the telescope`. There are a number of parses of this including:

```
saw(Bill,the_man(with(the_telescope)))
saw(Bill,the_man,with(the_telescope))
saw(Bill(with(the_telescope)),the_man)
```

Assume that in the world facts, we have `Human(Bill)` and `Human(the_man)`. Using the following Facilitation Default Rule (11), we obtain `Hold(saw(Bill,the_man)))` as a commonsense inference.

$$\frac{\begin{array}{c}[\texttt{Holds}(\texttt{saw}(\texttt{X},\texttt{Y}(\texttt{with}(\texttt{Z})))) \vee \texttt{Holds}(\texttt{saw}(\texttt{X},\texttt{Y},(\texttt{with}(\texttt{Z})))) \vee \texttt{Holds}(\texttt{saw}(\texttt{X}(\texttt{with}(\texttt{Z})),\texttt{Y}))] \\ \wedge\ \texttt{Human}(\texttt{X}) \wedge \texttt{Human}(\texttt{Y}) \\ : \texttt{Holds}(\texttt{saw}(\texttt{X},\texttt{Y}))\end{array}}{\texttt{Holds}(\texttt{saw}(\texttt{X},\texttt{Y}))} \tag{11}$$

So even though there might be ambiguity over the meaning of the whole sentences, all these parses agree that `Bill saw the man`.

7.0.4 Inferencing example 2

Now consider the inputs `the hat under the box` and `the box under the table`. This gives the following pair of inferences:

$$\begin{array}{c}\texttt{Holds}(\texttt{under}(\texttt{the\_hat},\texttt{the\_box})) \\ \texttt{Holds}(\texttt{under}(\texttt{the\_box},\texttt{the\_table}))\end{array}$$

Using the Facilitation Default rule (12), we obtain `Holds(under(the_hat,the_table))` as a commonsense inference.

$$\frac{\texttt{Holds(under(X,Y))} \wedge \texttt{Holds(under(Y,Z))} \;:\; \texttt{Holds(under(X,Z))}}{\texttt{Holds(under(X,Z))}} \qquad (12)$$

So Facilitation Default Rule (12) gives a form of transitivity for the word `under`.

7.0.5 Inferencing example 3

Consider the input `John walked the dog round the wood`. Suppose this is as follows

```
walked(John,the_dog,round(the_wood))
```

Assuming `Journey(round(the_wood))` and `Mammal(the_dog)` are in the world facts, we obtain the following commonsense inferences using Facilitation Default rule (13).

```
Holds(walked(John,round(the_wood))
Holds(walked(the_dog,round(the_wood))
```

$$\frac{\begin{array}{c}\texttt{Holds(walked(X,Y,Z))} \wedge \texttt{Journey(Z)} \wedge \texttt{Mammal(Y)} \\ :\; \neg\texttt{Baby(Y)} \wedge \texttt{Holds(walked(X,Z))} \wedge \texttt{Holds(walked(Y,Z))}\end{array}}{\texttt{Holds(walked(X,Z))} \wedge \texttt{Holds(walked(Y,Z))}} \qquad (13)$$

So we obtain `John walked round the wood` and `The dog walked round the wood`.

Now consider the input `John walked the baby round the wood`. Suppose this is parsed as follows.

```
walked(John,the_baby,round(the_wood))
```

Suppose also that `Baby(the_baby)` is in the world facts. So the Facilitation Default Rule (13) cannot be applied.

$$\frac{\begin{array}{c}\texttt{Holds(walked(X,Y,Z))} \wedge \texttt{Journey(Z)} \\ :\; \texttt{Holds(went(X,Y,Z))} \wedge \texttt{Holds(went(X,Z))} \wedge \texttt{Holds(went(Y,Z))}\end{array}}{\texttt{Holds(went(X,Y,Z))} \wedge \texttt{Holds(went(X,Z))} \wedge \texttt{Holds(went(Y,Z))}} \qquad (14)$$

However, from the Facilitation Default Rule (14) we obtain the commonsense inferences.

```
Holds(went(John,the_baby,round(the_wood)))
Holds(went(John,round(the_wood)))
Holds(went(the_baby,round(the_wood)))
```

Now consider the input `John walked the baby round in the pram`. Suppose this is parsed as `walked(john,the_baby,round(in(the_pram)))` Assuming `¬Journey(round(in(the_pram)))` is in the world facts, then from the Facilitation Default rule (15) we can derive the commonsense inference `Holds(went(the_baby),round(in(the_pram)))`.

$$\frac{\mathtt{Holds(walked(X,Y,Z))} \wedge \neg\mathtt{Journey(Z)} \quad : \ \mathtt{Holds(went(Y,Z))}}{\mathtt{Holds(went(Y,Z))}} \tag{15}$$

Hence we obtain `The baby went round in the pram`. But we don't get `John went round in the pram`.

8 Commonsense explaining

Commonsense explaining is done in three ways: (1) deriving `Holds` inferences; (2) deriving inferences using the world knowledge predicate symbols; and (3) rewriting terms in `Holds` inferences using term substitution.

8.1 Deriving `Holds` inferences

Deriving `Holds` predicates provides a rewrite of the output of the parser. So if α is output by the parser, and with respect to the world knowledge, α implies β, then `Holds`(β) is an inference. This kind of explaining therefore incorporates the commonsense inferencing covered in Section 7. It provides a simplification of the output taking a subset of the words in the output and/or replacing some words with alternative words where the alternative words might be more meaningful to the user of the system. By replacing words with alternative words, the inference might be longer (ie. contains more words) than the original `Holds` inference. This overlaps with the approach of term substitution discussed in Section 8.3.

8.1.1 Explaining example 1

In Inferencing example 3, the derivation of `Holds(walked(John,the_baby))` from `John walked the baby round the woods` could be part of an explanation. In addition, we may wish to expand the explanation by infering more complex formulae such as by using the Facilitation Default Rule (16).

$$\frac{\mathtt{Holds(walked(X,Y,Z))} \wedge \mathtt{Journey(Z)} \wedge \mathtt{Baby(Y)} \ : \ \neg\mathtt{Holds(walked(Y,Z))}}{\neg\mathtt{Holds(walked(Y,Z))}} \tag{16}$$

So if `Journey(round(the_wood))` and `Baby(the_baby)` are in the world facts, we obtain

$$\neg\mathtt{Holds(walked(the\_baby,round(the\_wood)))}$$

and thereby exclude the possibility that `the baby walked round the wood`.

8.2 Deriving inferences using world predicate symbols

Here we use world default rules to obtain inferences using world predicate symbols. These inferences provide explanations of words or combinations of words in terms of the semantics of these predicate symbols.

8.2.1 Explaining example 2

Consider the following world default rule,

$$\frac{\texttt{PartHolds(handbook)} \ : \ \texttt{Book(handbook)}}{\texttt{Book(handbook)}} \tag{17}$$

where the following is a world fact,

$$\begin{aligned}\texttt{Book(handbook)} \leftrightarrow \ & \texttt{IsA(handbook,book)} \\ & \wedge \texttt{PhysicalObject(book)} \\ & \wedge \texttt{MadeOf(book,paper)}\end{aligned}$$

Suppose we have the output `is(the_handbook,heavy)` from the parser. From this, we infer, `PartHolds(handbook)`. With this inference, together with the World Default Rule (17), we obtain the following inferences:

$$\begin{gathered}\texttt{IsA(handbook, book)} \\ \texttt{PhysicalObject(book)} \\ \texttt{MadeOf(book, paper)}\end{gathered}$$

In this way, the occurrence of `handbook` in the output from the parser can be used to obtained context-sensitive world knowledge about `handbook`, and this context-sensitive knowledge provides a form of explanation.

8.2.2 Explaining example 3

Consider the sentence `She almost killed him`. This sentence has a few quite different interpretations. One of these interpretation can be partly described by the following default rule.

$$\frac{\texttt{Holds(almost(killed(X, Y)))} \ : \ \neg\texttt{Caused(X,an\_accident)} \wedge \neg\texttt{Caused(X,a\_violent\_act)}}{\texttt{Is\_very\_angry\_with(X, Y)}} \tag{18}$$

Other interpretations of `almost killed` can be captured by similar default rules.

8.3 Rewriting terms using term substitution

By rewriting terms using term substitution, unclear or unknown terms can be replaced by more meaningful terms (from the user's perspective). Note, this kind of explaining may involve replacing terms by terms with more symbols.

8.3.1 Explaining example 4

Consider the verb `flies` occuring in a phrase. Suppose this is represented by the n-ary function symbol $\texttt{flies}(\alpha_1, ..., \alpha_n)$ where $\alpha_1, ..., \alpha_n$ are terms. Possible term substitutions that we could obtain in general using the Facilitation Default Rules include:

$$\begin{gathered}\texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(through\_the\_air(}\alpha_1, ..., \alpha_n\texttt{)))} \\ \texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(quickly(}\alpha_1, ..., \alpha_n\texttt{)))} \\ \texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(through\_a\_trajectory(}\alpha_1, ..., \alpha_n\texttt{)))} \\ \texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(swiftly(}\alpha_1, ..., \alpha_n\texttt{)))}\end{gathered}$$

However, in the more specialized context of time being the topic of the phrase, then we could obtain the more restricted set of terms.

$$\begin{gathered}\texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(quickly(}\alpha_1, ..., \alpha_n\texttt{)))} \\ \texttt{Sub(flies(}\alpha_1, ..., \alpha_n\texttt{), moves(swiftly(}\alpha_1, ..., \alpha_n\texttt{)))}\end{gathered}$$

Using the Default Rule (19) to generate the term substitution inference, we can explain a sentence like `This helicopter flies` by rewriting it into a semantically equivalent form but with different terms.

$$\texttt{moves(through\_the\_air(this\_helicopter))}$$

Note, in the following default rules we have used the extra notation $\overline{\texttt{T}}$, where $\overline{\texttt{T}}$ denotes a tuple of zero or more further variables. This allows us to obviate the need to duplicate each rule for each value of n for the n-ary function $\texttt{flies}(\alpha_1, ..., \alpha_n)$.

$$\frac{\texttt{PhysicalObject(X)} \ : \ \texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(through\_the\_air(X},\overline{\texttt{T}}\texttt{)))}}{\texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(through\_the\_air(X},\overline{\texttt{T}}\texttt{)))}} \tag{19}$$

$$\frac{\texttt{PhysicalObject(X)} \ : \ \texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(through\_a\_trajectory(X},\overline{\texttt{T}}\texttt{)))}}{\texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(through\_a\_trajectory(X},\overline{\texttt{T}}\texttt{)))}} \tag{20}$$

$$\frac{\neg\texttt{Aircraft(X)} \ : \ \texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(quickly(X},\overline{\texttt{T}}\texttt{)))}}{\texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(quickly(X},\overline{\texttt{T}}\texttt{)))}} \tag{21}$$

$$\frac{\neg\texttt{Aircraft(X)} \ : \ \texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(swiftly(X},\overline{\texttt{T}}\texttt{)))}}{\texttt{Sub(flies(X},\overline{\texttt{T}}\texttt{),moves(swiftly(X},\overline{\texttt{T}}\texttt{)))}} \tag{22}$$

Using default rules to generate the term substitution inference, we can explain a sentence like `This helicopter flies to the island` by rewriting it into a semantically equivalent form:

$$\texttt{moves(through\_the\_air(this\_helicopter,to(the\_island)))}$$

As another example we can explain a sentence like `time flies` by rewriting it into a semantically equivalent form `moves(quickly(time))` but with different terms. Now consider the natural language input `time flies like an arrow`. The output from the parser could be represented by the term `flies(time,like(an_arrow))`. From this, we can obtain the term `flies(time,like(an_arrow))`, and hence `moves(quickly(time,like(an_arrow)))`.

8.3.2 Explaining example 5

Consider the sentence `Mary frapped John`. Here the word `frapped` doesn't appear in any English dictionary. Maybe the word is a typographical error, or maybe it is a new word. Yet from the morphology of the word and from the understanding of the rest of the sentence, it seems that frapped is some kind of action that can be viewed as a relationship between Mary and John. So from this, we would like to infer that it is a specialization of the word `does-something`, and so capture that `Mary does something to John`.

$$\frac{\mathtt{Holds(X(Y,Z))} \wedge \mathtt{Human(Y)} \wedge \mathtt{Human(Z)} \wedge \mathtt{UnknownWord(X)} : \ \mathtt{Action(X)}}{\mathtt{Action(X)} \wedge \mathtt{Sub(X(Y,Z), does\_something(Y,Z))}} \tag{23}$$

Here we assume if a word α does not appear in our lexicon, then $\mathtt{UnknownWord}(\alpha)$ holds. We assume `Unknown` is a structural predicate symbol.

9 Comparison with typed feature structures

Now that we have presented our framework, we will return to considering weaknesses of other approaches that address similar issues to us. In particular, we will focus on approaches based on typed feature structures. In this section, we assume the reader is familiar with [Cop92, AM91, LBAC95, LC99].

9.1 Default inferencing without reasoning with feature values

First we consider a typed feature system with default unification such as the Laurel system [Cop92]. Entries for each feature can be regarded as "all or nothing". If $\mathtt{F} = \top$, then the entry for the feature `F` can be regarded as the value "nothing", otherwise the entry is some value `a`. This value is then all that is known about `F` in that feature structure. This all or nothing behaviour is reflected in default unification, where $\sqcap^<$ denotes default unification with the feature on the left is a non-defeasible feature, and the feature on the right is a defeasible feature:

$$([\mathtt{F} = \mathtt{a}] \sqcap^< [\mathtt{F} = \top]) = [\mathtt{F} = \mathtt{a}]$$

$$([\mathtt{F} = \top] \sqcap^< [\mathtt{F} = \mathtt{a}]) = [\mathtt{F} = \mathtt{a}]$$

$$([\mathtt{F} = \mathtt{a}] \sqcap^< [\mathtt{F} = \mathtt{b}]) = [\mathtt{F} = \mathtt{a}]$$

So if either the non-defeasible feature structure or the defeasible feature structure contains nothing then the entry for the feature structure appears in the result of default unification. However, if both feature structures contain something, then the entry for the non-defeasible feature excludes the entry for the defeasible feature in the default unification. This holds even if in some sense $\alpha \wedge \beta$ is consistent. The default unification proceeds without reasoning with the feature values. This means the approach is ignoring information that is available.

This problem also means that a knowledgebase has to be developed in a global way — new items of knowledge cannot be added without considering where to slot it in the overall typed feature structure since the system does not offer logical reasoning. This is a general problem of typed feature structure systems including PDU [LBAC95] and is directly related to the problems raised in Section 2 about viewing a typed feature structure as a database — information about a concept is inheritied from the more general concepts without consideration of the context of the concept and without consideration of the meaning of the values being inherited. In a sense, typed feature structures can be viewed as a meta-level language where the value entered for each feature is just treated as an atom.

9.2 Default inferencing without reasoning with the context of sentences

Consider the example `Mary enjoyed the book` discussed in [LC99]. From this sentence, we could obtain the defeasible inference `Read(enjoyment_event,Mary,the_book)` using a typed feature structure system such as PDU [LBAC95]. The problem with this is that this defeasible inference should be defeated if for example during the course of a dialogue it is apparent that Mary is a goat, in which case we would more likely infer something like `Eat(enjoyment_event,Mary,the_book)`. In handling this kind of example, Lascarides and Copestake [LC99] have identified the lack of expressibility with a typed feature structure approach, and suggested a two stage process whereby defeasible inferences made by a PDU system can be defeated by a Commonsense Entailment knowledgebase. In this way, semantic or world knowledge is shared between a typed feature structure and a conditional logic knowledgebase. This seems to be a suboptimal solution to the problem.

We believe that they don't need to obtain the inference `Read(enjoyment_event,Mary,the_book)` in order to obtain a predicate-argument structure as output from the parser. Therefore determing in what sense Mary enjoyed the book should be left until reasoning with the World Knowledgebase. If all this reasoning is done with the World Knowledgebase, then it is not necessary to split world knowledge between a typed feature structure and a conditional logic knowledgebase. Rather it seems more lucid and efficient to put all this knowledge into one formalism. This will help developing and maintaining the knowledge.

We now consider how we can handle this example in our framework. Assuming the output of the parser gives `Enjoyed(Mary,the_book)`, and if we have the following World facts,

$$\begin{gathered}
\texttt{Book(the_book)} \\
\texttt{Literate(X)} \leftrightarrow \neg\texttt{Illiterate(X)} \\
\texttt{AllPagesBlank(X)} \rightarrow \neg\texttt{Readable(X)} \\
\texttt{MadeOfMarzipan(X)} \rightarrow \neg\texttt{Readable(X)}
\end{gathered}$$

then we obtain `Read(Mary,the_book)` as an inference from the Default Rule (24).

$$\frac{\texttt{Enjoyed(X,Y)} \wedge \texttt{Book(Y)} \ : \ \texttt{Literate(X)} \wedge \texttt{Readable(Y)}}{\texttt{Read(X,Y)}} \tag{24}$$

$$\frac{\texttt{Enjoyed(X,Y)} \wedge \texttt{Goat(X)} \wedge \texttt{Book(Y)} \ : \ \texttt{Eat(X,Y)} \wedge \neg\texttt{FairyTaleGoat(X)}}{\texttt{Eat(X,Y)}} \tag{25}$$

However, if we also have `Goat(Mary)` in the World Facts, then we get `Eat(Mary,the_book)` using the Default Rule (25). This approach seems much simpler and lucid than that proposed in [LC99].

9.3 Weakness of Commonsense Entailment for default reasoning

The logic of Commonsense Entailment [AM91] has some nice properties including the following, where we assume Δ is a knowledgebase of formulae in this logic:

- If a default formula $\alpha > \beta$ is in Δ, and α is an inference from Δ, and β is consistent with Δ, then β is a (defeasible) inference from Δ.
- If there are two default formulae $\alpha_1 > \beta_1$ and $\alpha_2 > \beta_2$ in Δ such that α_1 and α_2 are inferences from Δ, and β_1 and β_2 conflict, and α_1 is more specific than α_2 (ie. $\alpha_1 \rightarrow \alpha_2$ is an inference from Δ and $\alpha_2 \rightarrow \alpha_1$ is not an inference from Δ) then β_1 is derived in preference to β_2 from Δ.
- The language allows default statements of the form $\alpha > \beta$ as inferences from Δ and these default statements can contain nested occurrences of default statements.

However, no non-monotonic logic is ideal for all applications. With regard to the application outlined in this paper — commonsense reasoning with the predicate-argument structure output of a parser using world knowledge — Commonsense Entailment is inferior to default logic. We outline the weaknesses of Commonsense Entailment as follows.

First, there is only a semantic definition for drawing defeasible inferences. This semantic definition involves selecting preferred models. For anything other than very small examples, this is likely to be computational unviable, since there is need to construct a potentially large number of models concurrently. This may explain why there are no prototype implementations of automated reasoning technology based on Commonsense Entailment. This situation contrasts with the increasingly sophisticated automated reasoning for default logic now being made available.

Second, the representation of default knowledge is in some key respects weaker than that of default logic. Preference over defaults can only be based on specificty. So for example, if we have two defaults, say $\alpha_1 > \beta_1$ and $\alpha_2 > \beta_2$ in a knowledgebase Δ, where $\alpha_1 \rightarrow \alpha_2$ is not an inference from Δ and $\alpha_2 \rightarrow \alpha_1$ is not an inference from Δ, then in order to represent a preference over these formulae, say a preference for the first over the second, we need to add another rule $\alpha_1 \wedge \alpha_2 > \beta_1$ to Δ. For anything other than very small knowledgebases, this can cause an overwhelming increase in the number of formulae in the knowledgebase, together with associated maintenance problems.

Third, another representational weakness of Commonsense Entailment is that it cannot represent the equivalent of semi-normal defaults found in default logic. A semi-normal default rule is a default rule where the justification is a conjunction of the formula used as the consequent with some other formula. To illustrate the need for semi-normal defaults consider the following semi-normal default rule:

$$\frac{\texttt{HasMotives(X)} \ : \ \texttt{IsSuspect(X)} \wedge \texttt{IsGuilty(X)}}{\texttt{IsSuspect(X)}} \tag{26}$$

A normal default rules is a default rule where the same formula is used as the justification and consequent. A normal default rule cannot capture the information in Default Rule (26). Consider the attempt at presenting Default Rule (26) as a normal default in Default Rule (27). This seems unacceptable since the consequent is now strengthed to force the inference of `IsGuilty(X)` for an indvidual `X`.

$$\frac{\texttt{HasMotives(X)} \ : \ \texttt{IsSuspect(X)} \wedge \texttt{IsGuilty(X)}}{\texttt{IsSuspect(X)} \wedge \texttt{IsGuilty(X)}} \tag{27}$$

Since Commonsense Entailment only allows conflict on the basis of the consequent — it does not support any counterpart to the justification found in default logic — it is not able to formalize this example in an analogous way to Default Rule (26).

Fourth, Commonsense Entailment excludes the possibility of multiple extensions. This means Commonsense Entailment takes a skeptical view of ambiguity. Rather than express multiple viewpoints (credulous views) on ambiguous information it only infers (defeasible) facts that could hold in any credulous extension.

In conclusion, Commonsense Entailment does not seem to offer a practical solution to handling world knowledge.

9.4 Overgeneralization and undergeneralization in GLT

GLT suffers from the general problems of typed feature structures that we have discussed including the amount of semantic (or world) information inferred is limited in expressibility and no account is taken of the wider context of the word(s) being considered. There are a number of further weakenesses of GLT. We focus here on overgeneralization and undergeneralization in forming concepts (for a discussion see [HM99]).

This example is based on example 45 (p122) of GLT [Pus95]. Consider the following two statements. Here, there are two overlapping senses of the word `bake`. In the first, we may regard `bake` as causing a *change of state* in `the_potatoes`, whereas in the second, we may regard `bake` as *creating* `the_cake`.

$$\texttt{bake}(\texttt{John}, \texttt{the\_potatoes})$$

$$\texttt{bake}(\texttt{John}, \texttt{the\_cake})$$

In GLT, the lexicon contains the *change of state* sense of `bake`, and then the second sense is obtained by `cake` forcing type coercion giving the *creating* sense. So the qualia `bake_a_cake` is obtained from the combination of the qulia for `bake` and for `cake`.

Unfortunately, GLT suffers from undergeneralization and overgeneralization. This stems from the definition of qualia unification (page 124 of [Pus95]) which requires a qualia value to be the same in both qualias (undergeneralization) and the unified qualia obtained by taking the "unique lower bound of the two qualia" (overgeneralization).

The undergeneralization eminates from the absence of any inferencing being involved in the process of "matching" two qualia. So for example if `Q1 = [F = a]` and `Q2 = [F = b]`, then qualia unification between `Q1` and `Q2` is prohibited since $\texttt{a} \neq \texttt{b}$. However, suppose in our world knowledge we can infer $\texttt{a} \leftrightarrow \texttt{b}$, or even $\texttt{a} \rightarrow \texttt{b}$, then it would seem desirable that this pre-condition for qualia unification is satisfied.

The overgeneralization eminates from the process of taking the unique lower bound of the two qualia when for example we could obtain the same feature structure for `bake a potato` and `bake a flower`. Yet really we want information from the semantic or world knowledge to differentiate these two verb phrases. Trying to reduce the semantics of `bake` to one feature structure forces the inferences to just being very general, whereas we really want a range of items of world knowledge that allow us to go from very general inferences to very specific inferences, during context-sensitive reasoning, thereby allowing similarities and differences between concepts to be inferred.

To summarize, in GLT to obtain extra senses of `bake` from the co-composition with complements requires careful consideration of the entries in the qualia of all nouns it is likely to co-occur, hence defeating the intended modularity of the approach. Furthermore, basing co-composition on $\sqcap$ is too weak since relatively few inferences can be obtained by such a reasoning mechanism.

As we have indicated in Section 3, we take a different approach to GLT. In the following defaults, all word senses give `heats` as a property of `bake`. More specialized aspects of `bake` include `create` if it is a `cake` being baked.

$$\frac{\mathtt{Bake(X,Y)} \ : \ \mathtt{PhysicalObject(Y)}}{\mathtt{Heats(X,Y)}} \tag{28}$$

$$\frac{\mathtt{Bake(X,Y)} \wedge \mathtt{Food(Y)} \ : \ \mathtt{Inedible(Y)}}{\mathtt{MakeEdible(X,Y)}} \tag{29}$$

$$\frac{\mathtt{Bake(X,Y)} \wedge \mathtt{Cake(Y)} \ : \ \mathtt{Create(X,Y)}}{\mathtt{Create(X,Y)}} \tag{30}$$

$$\frac{\mathtt{Bake(X,Y)} \wedge \mathtt{Food(Y)} \ : \ \mathtt{MakeTastier(Y)}}{\mathtt{MakeTastier(X,Y)}} \tag{31}$$

So from `Bake(John,the_potatoes)` and `Food(the_potatoes)`, we get `Heats(John,the_potatoes)`, `MakeEdible(John,the_potatoes)`, and `MakeTastier(John,the_potatoes)`. In this way, default logic allows us to incrementally build a conceptualization of `bake` in a sentence with both general and specific inferences selected according to the wider context.

10 Discussion

In this paper, we have proposed a framework for reasoning with the output of parsers by using world knowledge. This framework, being based on default logic, offers context-sensitive reasoning with uncertainty management. In this framework, the roles of lexical and world knowledge has been separated and clarified. Furthermore, the kinds of reasoning with output has been clarified by considering commonsense checking, commonsense inferencing, and commonsense explaining. This framework is richer and more expressive than alternative approaches to capturing lexical and world knowledge. It is based on default logic, the most extensively studied approach to default reasoning, with many intuitive features and formal properties. Furthermore there are a range of technologies for supporting the use of default logic. Moreover, default logic seems ideally suited to applications in commonsense reasoning with output from parsers.

It is worth remembering that there is no loss is expressibility in restricting the output of the parser to be represented by terms, as opposed to classical formulae, since if we have n-ary functions, we have the full power of classical logic. In addition, predicate and function symbols are interchangable in the sense that any n-place function can be represented by an n+1 place predicate symbol, and any n-place predicate can be represented by an n-place function with codomain containing just true and false.

The framework presented in this paper is very expressive, and so there are a number of ways of representing the the same thing. We saw this in particular in Section 8 where for example we could rewrite the output from the parser using different words either by `Holds` inference or by term substitution. It is not yet clear whether any of these choices are preferable. We will attempt to address this in our next research goal where we are attempting to construct a useful default theory for understanding sentences from a limited vocabulary.

Another research goal is to develop techniques for semi-automated generation of default thories for lexical and world knowledge. For an application, it is possible that a relatively large number of default rules would be required for an acceptable level of performance. To address this viability problem, we are investigating a number of avenues: (1) Using inductive logic programming ([Mug92, Cus97, Cus98, TM98, DN99]) and statistical techniques (such as [CHS93]), to generate formulae for a default theory for a domain; (2) Using co-locational data with decision tree learning based on ID3 or C4.5, or co-locational data with unsupervised learning techniques such as in [Yar95]; and (3) Using machine-readable dictionaries and thesauri [NFE90, Mei93]. Whilst none of these techniques, except [DN99], use a hypothesis language that is as expressive as default logic, they are potentially very useful in identifying relationships and generating rules that could be incorporated within default logic.

Further developments in the theory for formalizing commonsense reasoning with outputs from parsers should draw upon research in commonsense reasoning in artificial intelligence. This field is central to artificial intelligence. It aims to find frameworks for more comprehensive reasoning about the world based on a more sophisticated modelling of notions such as time, action and causality. Consider for example given the information that `John made a cup of tea`. There are various complex ramifications of this information including:

```
John boiled some water and poured the water on some tea.
John probably poured some water in the kettle and switched the kettle on.
John probably used some water, some electricity, and some tea.
Switching the electricity on caused some electrons to flow in the lead.
The electrons flowing caused the elctricity meter to turn.
```

Problems raised in attempting to mechanize such reasoning, led to calls for models of the physics of the world around us [MH69, Hay79]. For more recent coverage of commonsense reasoning in artificial intelligence see [San94, Sha97, MS98].

There are, in addition, linguistic applications where the expressivity and richness of the framework proposed here might prove to be necessary. Recall that we have restricted our attention in this paper to cases where world knowledge interacts with the output from parsers, and that we have furthermore assumed that there is no prioritization of reasoning tasks. However, for some disambiguation, hearers make interpretative decisions before the complete sentence is processed, and these decisions are partly based on contextual and world knowledge. This means that our default theory should be interfaced more closely with the parsing process, rather than just with an output string. Furthermore, as has been argued from the perspective of Dynamic Syntax [KMG00], the context dependence of parsing excedes cases of disambiguation, and influences the process of building semantic representations from words throughout. In our framework, we have included unbound variables, as well as variables ranging over tuples of variables, so that our default rules can be used as rule schemata. However, this formalization can be extended to model the incremental nature of deriving inferences from increasing natural language input. Thus, for example, default rules can be employed to model the difference between the following two sentences:

```
While John was knitting the cup fell down

While John was knitting the jumper fell down
```

In the first sentence, `knitting` is interpreted as intransitive, and `the cup` in interpreted as the subject of the main clause. In contrast, in the second sentence, `the jumper` can be interpreted as the object of `knitting`, so that a garden-path effect results, since there is no subject for the main clause. The difference between the two examples results from the difference between the plausibility of knitting cups and knitting jumpers, which, for us, is a difference in world knowledge. For cases like these, contextual reasoning does not interact with output from parsing, but rather interacts with words before a sentence is built, thereby influencing parsing decisions about syntactic structure. Our framework can be employed for cases like these if rules are allowed to be used with incomplete structures and individual words.

The second assumption we make here is that reasoning tasks are not prioritized. However, in a fully worked out application, probably not all derivable inferences are equally relevant on a given occasion, and some inferences might not even be derived at all. Although the details of implementing prioritization are yet to be worked out, we believe that the distinction between lexical and world knowledge we propose here is a step in the right direction, since from this perspective all inferences are potentially contextually determined and this expressible within a uniform analysis. While this perspective makes the task of implementing some version of prioritization more important, it also provides an elegant formalization of the general context dependence of the establishment of linguistic meaning.

In summary, we have developed here a general framework for reasoning with information from natural language and world knowledge, where we have assumed that the interpretation of natural langauge expressions is largely inferential and context sensitive. Our framework provides an explicit and rich formalization in which context specific inference associated with some linguistic expression on a given occasion can be expressed. The framework is flexible, so that is can be used for a number of more specific applications, while at the same time, it is based on defulat logic, so that it can be integrated with other systems of knowledge representation.

Acknowledgements

The authors would like to thank Ruth Kempson for helpful feedback on an earlier draft of this paper.

References

[AM91] N Asher and M Morreau. Commonsense entailment: A modal theory of nonmonotonic reasoning. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI91)*, pages 387–392. Morgan Kaufmann, 1991.

[Ant97] G Antoniou. *Non-monotonic Reasoning.* MIT Press, 1997.

[BCP94] T Briscoe, A Copestake, and V de Paiva, editors. *Inheritance, Defaults and the Lexicon.* Studies in Natural Language Processing. Cambridge University Prees, 1994.

[BDK97] G Brewka, J Dix, and K Konolige. *Non-monotonic Reasoning: An Overview.* CLSI Publications, 1997.

[Bes89] Ph Besnard. *An Introduction to Default Logic.* Springer, 1989.

[BFGM91] R Beckworth, C Fellbaum, D Gross, and G Miller. WordNet: A lexical database organized on psycholinguistic principles. In U Zernik, editor, *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, pages 211–226. Lawrence Erlbaum Associates, 1991.

[Bre91] G Brewka. *Common-sense Reasoning.* Cambridge University Press, 1991.

[BS98] S Bruning and T Schaub. Prolog technology for default reasoning: roof theory and compilation techniques. *Artificial Intelligence*, 106:1–75, 1998.

[Buv96] S Buvac. Resolving lexical ambiguity using a formal theory of context. In K van Deemter and S Peters, editors, *Semantic Ambiguity and Underspecification*, pages 101–124. CSLI Publications, 1996.

[Car92] B Carpenter. *The Logic of Typed Feature Structures*. Cambridge University Press, 1992.

[Car99] R Carston. *Thoughts and Utterances: The Pragmatics of Verbal Communication*. Blackwells, 1999.

[CHS93] J Cussens, A Hunter, and A Srinivasan. Generating explicit orderings for non-monotonic logics. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)*, pages 420–425. MIT Press, 1993.

[Cop92] A Copestake. The representation of lexical semantic information. Technical report, University of Sussex, 1992. CSRP 280.

[Cus97] J Cussens. Parts-of-speech tagging using progol. In *Proceedings of the Inductive Logic Programming Conference(ILP'97)*, 1997.

[Cus98] J Cussens. Using prior probabilities and density estimation for relational classification. In *Proceedings of the Inductive Logic Programming Conference(ILP'98)*, 1998.

[Del98] J Delgrande. Conditional logics for defeasible reasoning. In *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 2, pages 135–174. Kluwer, 1998.

[DN99] B Duval and P Nicolas. Learning default theories. In Anthony Hunter and Simon Parsons, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 1638 of *Lecture Notes in Computer Science*, pages 148–159, 1999.

[DP96] K van Deemter and S Peters, editors. *Semantic Ambiguity and Underspecification*, volume 55 of *CLSI Lecture Notes*. CLSI Publications, 1996.

[EG89a] R Evans and G Gazdar. Inference in DATR. In *Proceedings of the 4th Conference of European Chapter of the Association of Computational Linguistics (EACL89)*, pages 66–71, 1989.

[EG89b] R Evans and G Gazdar. The semantics of DATR. In A Cohn, editor, *Proceedings of the Seventh Conference of the Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB'89)*, pages 79–87. Pitman, 1989.

[EG96] R Evans and G Gazdar. DATR: A language for lexical knowledge representation. *Computational Linguistics*, 22:167–216, 1996.

[Eth88] D Etherington. *Reasoning with Incomplete Information*. Pitman, 1988.

[FA98] C Fillmore and B Atkins. Framenet and lexicongraphic relevance. In *Proceedings of the First International Conference on Language Resources and Evaluation*, 1998.

[Fel98] C Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[Fer97] T Fernando. A modal logic non-determinstic disambiguation. In *Proceedings of the Amsterdam Colloquium*, pages 121–126, 1997.

[FL98] J Fodor and E Lepore. The emptiness of the lexicon: critcial reflection on j. putejovsky's the generative lexicon. *Linguistic Inquiry*, 25:269–288, 1998.

[Fod98] J Fodor. *Concepts: Where Cognitive Science Went Wrong*. Clarendon Press (OUP), 1998.

[Ger95] P Gervas. *Logical considerations in the interpretation of presuppositional sentences*. PhD thesis, Department of Computer Science, Imperial College, 1995.

[Gor95] P Gorrell. *Syntac and Parsing*. Cambridge University Press, 1995.

[Hay79] P Hayes. Naive physics manifesto. In D Michie, editor, *Expert Systems in the Microelectronic Age*. Edinburgh University Press, 1979.

[HM99] A Hunter and L Marten. Context-sensitive reasoning with world knowledge. In *Working Papers in Linguistics*, volume 9, pages 373–386. SOAS, 1999.

[Hun95] A. Hunter. Using default logic in information retrieval. In C Froidvaux and J Kohlas, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 946 of *Lecture Notes in Computer Science*, pages 235–242. Springer, 1995.

[Hun96] A Hunter. Intelligent text handling using default logic. In *Proceedings of the IEEE Conference on Tools with Artificial Intelligence*, pages 34–40. IEEE Computer Society Press, 1996.

[Hun97] A. Hunter. Using default logic for lexical knowledge. In *Qualitative and Quantitative Practical Reasoning*, volume 1244 of *Lecture Notes in Computer Science*, pages 86–95. Springer, 1997.

[Hun99] A Hunter. Context-dependent reasoning with lexical knowledge using default logic. Technical report, Department of Computer Science, University College, London, 1999.

[IV98] N Ide and J Veronis. Introduction to the special issue on word sense disambiguation. *Computational Linguistics*, 24(1):1–40, 1998.

[Jac77] R Jackenndoff. *X-Bar Syntax*. MIT Press, 1977.

[KMG00] R Kempson, W Meyer-Viol, and D Gabbay. *The Deductive Flow of Natural Language*. Blackwell, 2000. (in press).

[LBAC95] A Lascarides, T Briscoe, N Asher, and A Copestake. Order independent and persisitent typed default unification. *Linguistics and Philiosophy*, 19(1):1–90, 1995.

[LC98] A Lascarides and A Copestake. Default representation in constraint-based frameworks. *Journal of Linguistics*, 34(2):387–414, 1998.

[LC99] A Lascarides and A Copestake. Pragmatics and word meaning. *Computational Linguistics*, 25(1):55–105, 1999.

[LCB96] A Lascarides, A Copestake, and T Briscoe. Ambiguity and coherence. *Journal of Semantics*, 13(1):41–65, 1996.

[Len95] D Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[LG90] D Lenat and R Guha. *Building Large Knowledge-based Systems*. Addison-Wesley, 1990.

[LS95] T Linke and T Schaub. Lemma handling in default logic theorem provers. In *Symbolic and Qualitative Approaches to Reasoning and Uncertainty*, volume 946 of *Lecture Notes in Computer Science*, pages 285–292. Springer, 1995.

[Mei93] W Meijs. Exploring lexical knowledge. In C Souter and E Atwell, editors, *Corpus-based Computational Linguistics*, pages 249–260. Rodopi, 1993.

[Mer91] B Mercer. Presuppositions and default reasoning: A study in lexical pragmatics. In J Pustejovsky and S Bergler, editors, *Lexical Semantics and Knowledge Representation*, volume 627 of *Lecture Notes in Artificial Intelligence*, pages 321–340. Springer, 1991.

[MH69] J McCarthy and P Hayes. Some philosophical problems from the standpoint of artifical intelligence. In B Meltzer and D Michie, editors, *Machine Intelligence 4*. Edinburgh University Press, 1969.

[Mil95] G Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[MS98] R Miller and M Shanahan. Working papers of commonsense'98: The fourth symposium on logical formalization of commonsense reasoning. Technical report, Queen Mary and Westfield College, London, 1998.

[Mug92] S Muggleton. *Inductive Logic Programming*. Academic Press, 1992.

[NFE90] J Nutter, E Fox, and M Evens. Building a lexicon from machine-readable dictionaries for improved information retrieval. *Literary and Linguistic Computing*, 5(2):129–137, 1990.

[Nie94] I Niemelä. A decision method for non-monotonic reasoning based on autoepistemic reasoning. In *Proceedings of the Fourth International Conference Principles of Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufmann, 1994.

[NS98] P Nicolas and T Schaub. The XRay system: An implementation platform for local query answering in default logics. In A Hunter and S Parsons, editors, *Applications of Uncertainty Formalisms*, Lecture Notes in Computer Science. Springer, 1998.

[PB93] J Pustejovsky and B Boguraev. Lexical knowledge and natural language processing. *Artificial Intelligence*, 63:193–223, 1993.

[Poe96] M Poesio. Semantic ambiguity and perceived ambiguity. In K van Deemter and S Peters, editors, *Semantic Ambiguity and Underspecification*, pages 159–202. CSLI Publications, 1996.

[Pus95] J Pustejovsky. *The Generative Lexicon*. MIT Press, 1995.

[Qua93] J Quantz. A preferential default description logic for disambiguation in natural language processing. In G Brewka and C Witteveen, editors, *Proceedings of the Dutch-German Workshop on Nonmonotonic Reasoning Techniques and their Applications*, pages 1–13. RWTH Aachen, 1993.

[Rei78] R Reiter. On closed world databases. In *Logic and Databases*, pages 55–76. Plenum Press, 1978.

[Rei80] R Reiter. Default logic. *Artificial Intelligence*, 13:81–132, 1980.

[San94] E Sandewall. *Features and Fluents: The Representation of Dynamical Systems (Volume 1)*. Oxford University Press, 1994.

[SBN96] T Schaub, S Brüning, and P Nicolas. XRay: A prolog technology theorem prover for default reasoning: A system description. In M McRobbie and J Slaney, editors, *Proceedings of the Conference on Automated Deduction*, volume 1104 of *Lecture Notes in Artificial Intelligence*, pages 293–297. Springer, 1996.

[Sch95] T Schaub. A new methodology for query-answering in default logics via structure-oriented theorem proving. *Journal of Automated Reasoning*, 15:95–165, 1995.

[Sch98] T Schaub. *The Automation of Reasoning with Incomplete Information: From Semantic Foundation to Efficient Computation.* Springer, 1998.

[Sha97] M Shanahan. *Solving the Frame Problem.* MIT Press, 1997.

[SW97] D Sperber and D Wilson. The mapping between the mental and the public lexicon. Technical report, Department of Linguistics, University College London, 1997.

[Tho97] R Thomason. Non-monotonicity in linguistics. In J van Benthem and A ter Meulen, editors, *Handbook of Logic and Language*, pages 777–831. Elsevier, 1997.

[TM98] C Thompson and R Mooney. Semantic lexicon acquisition for learning natural language interfaces. Technical Report TR AI98-273, AI Lab, University of Texas at Austin, 1998.

[Voo94] E Voorhees. Query expansion using lexical-semantic relations. In W Croft and C van Rijsbergen, editors, *Proceedings of the Seventeenth International ACM-SIGIR Conference on Research and Developement in Information Retrieval*, pages 61–69, 1994.

[Yar95] D Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 32nd Annual Meeting of the Association of Computational Lingustics*, pages 189–196, 1995.