FORMALIZING CONTEXT
(Expanded Notes)

John McCarthy and Saša Buvač
Computer Science Department
Stanford University
Stanford
California 94305-2140.
{jmc, buvac}@sail.stanford.edu

Abstract

These notes discuss formalizing contexts as first class objects. The basic relation is $ist(c, p)$. It asserts that the *proposition* p is true in the *context* c. The most important formulas relate the propositions true in different contexts. Introducing contexts as formal objects will permit axiomatizations in limited contexts to be expanded to *transcend* the original limitations. This seems necessary to provide AI programs using logic with certain capabilities that human fact representation and human reasoning possess. Fully implementing *transcendence* seems to require further extensions to mathematical logic, i.e. beyond the nonmonotonic inference methods first invented in AI and now studied as a new domain of logic.

1 Introduction

These notes contain some of the reasoning behind the proposals of [McC87] to introduce contexts as formal objects. The present proposals are incomplete and tentative. In particular the formulas are not what we will eventually want, and we will feel free to use formulas in discussions of different applications that aren't always compatible with each other. This is an expanded and revised version of [McC93].

Our object is to introduce contexts as abstract mathematical entities with properties useful in artificial intelligence. Our attitude is therefore a computer science or engineering attitude. If one takes a psychological or philosophical attitude, one can examine the phenomenon of contextual dependence of an utterance or a belief. However, it seems to us unlikely that this study will result in a unique conclusion about *what context is.* Instead, as is usual in AI, various notions will be found useful.

One major AI goal of this formalization is to allow simple axioms for common sense phenomena, e.g. axioms for static blocks world situations, to be *lifted* to contexts involving fewer assumptions, e.g. to contexts in which situations change. This is necessary if the axioms are to be included in general common sense databases that can be used by any programs needing to know about the phenomenon covered but which may be concerned with other matters as well. Rules for lifting are described in section 4 and an example is given.

A second goal is to treat the context associated with a particular circumstance, e.g. the context of a conversation in which terms have particular meanings that they wouldn't have in the language in general.

The most ambitious goal is to make AI systems which are never permanently stuck with the concepts they use at a given time because they can always *transcend* the context they are in—if they are smart enough or are told how to do so. To this end, formulas $ist(c, p)$ are always considered as themselves asserted within a context, i.e. we have something like $ist(c', ist(c, p))$. The regress is infinite, but we will show that it is harmless.

The main formulas are sentences of the form

$$c' : \quad ist(c, p),$$

which are to be taken as assertions that the proposition p is true in the context c, itself asserted in an outer context c'. (We have adopted Guha's [Guh91] notation rather than that of [McC87], because he built his into Cyc, and it was easy for us to change ours.) For now, propositions may be identified with sentences in English or in various logical languages, but we may later take them in the sense of [McC79b] as abstractions with possibly different identity conditions. We will use both logical sentences and English sentences in the examples, according to whichever is more convenient.

Contexts are abstract objects. We don't offer a definition, but we will offer some examples. Some contexts will be *rich* objects, like situations in situation calculus. For example, the context associated with a conversation is rich; we cannot list all the common assumptions of the participants. Thus we don't purport to describe such contexts completely; we only say something about them. On the other hand, the contexts associated with certain microtheories are poor and can be completely described.

Here are some examples.

$$c0 : \quad ist(\textit{context-of}(\text{“Sherlock Holmes stories”}), \text{“Holmes is a detective”})$$

asserts that it is true in the context of the Sherlock Holmes stories that Holmes is a detective. We use English quotations here, because the formal notation is still undecided. Here $c0$ is considered to be an *outer context*. In the context *context-of*("Sherlock Holmes stories"), Holmes's mother's maiden name does not have a value. We also have

$$c0 : \quad ist(\textit{context-of}(\text{“U.S. legal history”}), \text{“Holmes is a Supreme Court Justice”}).$$

Since the outer context is taken to be the same as above, we will omit it in subsequent formulas until it becomes relevant again. In this context, Holmes's mother's maiden name has a value, namely Jackson, and it would still have that value even if no-one today knew it.

$ist(c1, at(jmc, Stanford))$ is the assertion that John McCarthy is at Stanford University in a context in which it is given that *jmc* stands for the author of this paper and that *Stanford* stands for Stanford University. The context $c1$ may be one in which the symbol *at* is taken in the sense of being regularly at a place, rather than meaning momentarily at the place. In another context $c2$, $at(jmc, Stanford)$ may mean physical presence at Stanford at a certain instant. Programs based on the theory should use the appropriate meaning automatically.

Besides the sentence $ist(c, p)$, we also want the term $value(c, term)$ where $term$ is a term. For example, we may need $value(c, time)$, when c is a context that has a time, e.g. a context usable for making assertions about a particular situation. The interpretation of $value(c, term)$ involves a problem that doesn't arise with $ist(c, p)$. Namely, the space in which terms take values may itself be context dependent. However, many applications will not require this generality and will allow the domain of terms to be regarded as fixed.

Here's another example of the value of a term depending on context:

$$c0: \quad value(context\text{-}of(\text{“Sherlock Holmes stories”}), \text{“number of Holmes's wives”}) = 0$$

whereas

$$c0: \quad value(context\text{-}of(\text{“U.S. legal history”}), \text{“number of Holmes's wives”}) = 1.$$

We can consider $setof\text{-}wives(Holmes)$ as a term for which the set of possible values depends on context. In the case of the Supreme Court justice, the set consists of real women, whereas in the Sherlock Holmes case, it consists of fictitious women.

The remainder of this paper is organized as follows. In §2 we give examples of some elementary relations among contexts. The basic operations of contextual reasoning, entering and exiting contexts, are introduced in §3. In §4 we focus on *lifting axioms*—axioms relating what is true in one context based on what is true in another context. Building on the basic notions of entering/exiting contexts and lifting axioms, §5 shows how contexts can be used to reason in the style of natural deduction. To illustrate short term applicability of contexts, §6 demonstrates how the context formalism aids in the integration of databases which were not originally intended to be used together. In §7 we treat contexts associated with particular circumstances, namely those that come up in a conversation. The transcending of the context an AI system is in, as discussed in §8, might result in AI systems which are never permanently stuck with the concepts they use at a particular time. In §9, we argue that all sentences will always be context dependent, and thus it is not possible to define an absolute outermost context. Returning to applications, in §10 we sketch how contexts can be used to represent mental states and revise the beliefs of an agent. We conclude with a some remarks in §11. Most of the ideas and results in §2–§4 and §8–§11 were first reported in [McC93].

2 Relations among Contexts

There are many useful relations among contexts and also context valued functions. Here are some.

1. $specialize\text{-}time(t, c)$ is a context related to c in which the time is specialized to have the value t. We may have the relation

$$c0: \quad ist(specialize\text{-}time(t, c), at(jmc, Stanford)) \equiv ist(c, at\text{-}time(t, at(jmc, Stanford))).$$

Here $\mathit{at\text{-}time}(t,p)$ is the assertion that the proposition p holds at time t. We call this a *lifting* relation. It is convenient to write $\mathit{at\text{-}time}(t, \mathit{foo}(x,y,z))$ rather than $\mathit{foo}(x,y,z,t)$, because this lets us drop t in certain contexts. Many expressions are also better represented using modifiers expressed by functions rather than by using predicates and functions with many arguments. Actions give immediate examples, e.g. $\mathit{slowly}(\mathit{on\text{-}foot}(\mathit{go}))$ rather than $\mathit{go}(\mathit{on\text{-}foot}, \mathit{slowly})$.

Instead of using the function *specialize-time*, it may be convenient to use a predicate *specializes-time* and an axiom

$$c0: \quad \mathit{specializes\text{-}time}(t, c1, c2) \wedge \mathit{ist}(c1, p) \supset \mathit{ist}(c2, \mathit{at\text{-}time}(t, p)).$$

This would permit different contexts $c1$ all of which specialize $c2$ to a particular time.

There are also relations concerned with specializing places and with specializing speakers and hearers. Such relations permit lifting sentences containing pronouns to contexts not presuming specific places and persons.

2. If q is a proposition and c is a context, then $\mathit{assuming}(p, c)$ is another context like c in which p is assumed, where "assumed" is taken in the natural deduction sense. We investigate this further in §5.

3. There is a general relation *specializes* between contexts. We say $\mathit{specializes}(c1, c2)$ when $c2$ involves no more assumptions than $c1$. We have nonmonotonic relations

$$\mathit{specializes}(c1, c2) \wedge \neg ab1(p, c1, c2) \wedge \mathit{ist}(c1, p) \supset \mathit{ist}(c2, p).$$

and

$$\mathit{specializes}(c1, c2) \wedge \neg ab2(p, c1, c2) \wedge \mathit{ist}(c2, p) \supset \mathit{ist}(c1, p).$$

This gives nonmonotonic inheritance of *ist* in both from the subcontext to the supercontext and vice versa. More useful is the case when the sentences must change when lifted. Then we need to state that and every *proposition* meaningful in $c1$ is translatable into one meaningful in $c2$. See §4 for an example.

4. A major set of relations that need to be expressed are those between the context of a particular conversation and a subsequent written report about the situation in which the conversation took place. References to persons and objects are *decontextualized* in the report, and sentences like those given above can be used to express their relations.

5. Consider a wire with a signal on it which may have the value 0 or 1. We can associate a context with this wire that depends on time. Call it $c_{\mathit{wire}117}(t)$. Suppose at time 331, the value of this signal is 0. We can write this

$$\mathit{ist}(c_{\mathit{wire}117}(331), \mathit{signal} = 0).$$

Suppose the meaning of the signal is that the door of the microwave oven is open or closed according to whether the signal on *wire*117 is 0 or 1. We can then write the lifting relation

$$(\forall\, t)(\mathit{ist}(c_{\mathit{wire}117}(t), \mathit{signal} = 0) \equiv \mathit{door\text{-}open}(t).$$

The idea is that we can introduce contexts associated with particular parts of a circuit or other system, each with its special language, and lift sentences from this context to sentences meaningful for the system as a whole.

3 Entering and Exiting Contexts

Suppose we have the formula $c0 : \ ist(c, p)$. We can then *enter* the context c and infer the formula $c : \ p$. Conversely, if we have the formula $c : \ p$ we can infer $c0 : \ ist(c, p)$ by *exiting* the context c. We don't always want to be explicit about the sequence of all the contexts that were entered, but the logic needs to be such that the system always exits into the context it was in before entering. The enter and exit operations can be thought of as the push and pop operations on a stack. In the logic presented in [BBM] the sequence of contexts that has been entered is always explicitly stated.

We can regard $ist(c, p)$ as analogous to $c \supset p$, and the operation of entering c as analogous to *assuming* c in a system of *natural deduction* as invented by Gentzen and described in many logic texts. Indeed a context is a generalization of a collection of assumptions, but there are important differences. For example, contexts contain linguistic *assumptions* as well as declarative and a context may correspond to an infinite and only partially known collection of assumptions. Moreover, because relations among contexts are expressed as sentences in the language, $ist(c, p)$ allows inferences within the language that could only be done at the meta-level of the usual natural deduction systems.

There are various ways of handling the reasoning step of entering a context. The way most analogous to the usual natural deduction systems is to have an operation *enter* c. Having done this, one could then write any p for which one already had $ist(c, p)$. However, it seems more convenient in an interactive theorem proving to use the style of Jussi Ketonen's EKL interactive theorem prover [KW84]. In the style of that system, if one had $ist(c, p)$, one could immediately write p, and the system would keep track of the dependence on c. To avoid ambiguity as to where an occurrence of $ist(\ , p)$ came from, one might have to refer to a line number in the derivation. Having obtained p by entering c and then inferring some sentence q, one can *leave* c and get $ist(c, q)$. In natural deduction, this would be called discharging the assumption c.

Human natural language risks ambiguity by not always specifying such assumptions, relying on the hearer or reader to guess what contexts makes sense. The hearer employs a *principle of charity* and chooses an interpretation that assumes the speaker is making sense. In AI usage we probably don't usually want computers to make assertions that depend on principles of charity for their interpretation.

We are presently doubtful that the reasoning we will want our programs to do on their own will correspond closely to using an interactive theorem prover. Therefore, it isn't clear whether the above ideas for implementing entering and leaving contexts will be what we want.

Sentences of the form $ist(c, p)$ can themselves be true in contexts, e.g. we can have $ist(c0, ist(c1, p))$. In this draft, we will ignore the fact that if we want to stay in first order logic, we should reify assertions and write something like $ist(c0, Ist(c1, p))$, where $Ist(c, p)$ is a term rather than a wff. Actually the same problem arises for p itself; the occurrence of p in $ist(c, p)$ might have to be syntactically distinct from the occurrence of p standing by itself. Alternatively to reifying assertions we could use a modified logic; this approach was investigated in [BBM].

4 Lifting Axioms

Lifting axioms are axioms which relate the truth in one context to the truth in another context. Lifting is the process of inferring what is true in one context based on what is true in another context by the means of lifting axioms. We treat lifting as an informal notion in the sense that we never introduce a lifting operator. In this section we give an example of lifting.

Consider a context *above-theory*, which expresses a static theory of the blocks world predicates *on* and *above*. In reasoning about the predicates themselves it is convenient not to make them depend on situations or on a time parameter. However, we need to *lift* the results of *above-theory* to outer contexts that do involve situations or times.

To describe *above-theory*, we may write informally

(1) $\quad above\text{-}theory: \quad (\forall xy)(on(x,y) \supset above(x,y))$

(2) $\quad above\text{-}theory: \quad (\forall xyz)(above(x,y) \wedge above(y,z) \supset above(x,z))$

etc.

which stands for

(3) $\quad c0: \quad ist(above\text{-}theory, (\forall xy)(on(x,y) \supset above(x,y)))$

etc.

Constant $c0$ denotes an outer context. Section §8 has more about $c0$. In the following formulas, we put the context in which the formula is true to the left followed by a colon.

We want to use the *above-theory* in a context *blocks* which contains the theory of blocks world expressed using situation calculus. (We assume that situations are a disjoint sort, and that the variable s ranges over the situation sort.) In the context *blocks* predicates *on* and *above* have a third argument denoting a situation. Thus the context *blocks* needs to relate its three-argument predicates $on(x,y,s)$ and $above(x,y,s)$ to two-argument predicates $on(x,y)$ and $above(x,y)$ of the *above-theory* context. This is done by introducing a context of a particular situation, $spec\text{-}sit(s)$. A context $spec\text{-}sit(s)$ is associated with each situation s, such that

(4) $\quad blocks: \quad (\forall xys)(on(x,y,s) \equiv ist(spec\text{-}sit(s), on(x,y))),$

(5) $\quad blocks: \quad (\forall xys)(above(x,y,s) \equiv ist(spec\text{-}sit(s), above(x,y))),$

etc.

In order to get relations between $on(x,y,s)$ and $above(x,y,s)$, we will now import *above-theory* into the *blocks* context. The importation of *above-theory* is expressed by the axiom

(6) $\quad c0: \quad (\forall p) ist(above\text{-}theory, p) \supset ist(blocks, (\forall s)(ist(spec\text{-}sit(s), p))),$

asserting that the facts of *above-theory* all hold in the contexts associated with every situation. The following relation between $on(x,y,s)$ and $above(x,y,s)$ follows from the above axioms.

Theorem (above):

$$blocks: \quad (\forall sxy)(on(x,y,s) \supset above(x,y,s)).$$

The example given is so small that it would be simpler to give the relations among the three-argument predicates directly, but imagine that *above-theory* was much larger than is given here.

We proceed to derive the above theorem.

Proof (above): We begin by assuming

(7) $blocks: \quad on(x,y,s),$

asserting that block x is on block y in a specific situation s. Together with the universally instantiated form of the $\Rightarrow$ direction of formula 4 we get

(8) $blocks: \quad ist(spec\text{-}sit(s), on(x,y)).$

Now we enter $spec\text{-}sit(s)$ and get

(9) $spec\text{-}sit(s): \quad on(x,y).$

From (3) and (6) we conclude

(10) $c0: \quad ist(blocks, (\forall s) ist(spec\text{-}sit(s), (\forall xy) on(x,y) \supset above(x,y))).$

Therefore, by entering *blocks* we have

(11) $blocks: \quad (\forall s) ist(spec\text{-}sit(s), (\forall xy) on(x,y) \supset above(x,y)).$

By universal instantiation it follows that

(12) $blocks: \quad ist(spec\text{-}sit(s), (\forall xy) on(x,y) \supset above(x,y)).$

Entering $spec\text{-}sit(s)$ gives

(13) $spec\text{-}sit(s): \quad (\forall xy) on(x,y) \supset above(x,y).$

By logic, formulas 9 and 13 give

(14) $spec\text{-}sit(s): \quad above(x,y).$

We can now either continue reasoning in $spec\text{-}sit(s)$ or exit $spec\text{-}sit(s)$ and get

(15) $blocks: \quad ist(spec\text{-}sit(s), above(x,y)).$

Together with the universally instantiated form of the $\Leftarrow$ direction of formula 5 we get

(16) $blocks: \quad above(x,y,s).$

By the deduction theorem we can discharge the initial assumption to obtain

(17) $blocks: \quad on(x, y, s) \supset above(x, y, s).$

Finally, by universal generalization it follows that

(18) $blocks: \quad (\forall sxy) on(x, y, s) \supset above(x, y, s).$

$\square_{\textbf{above}}$

In this derivation we used a function giving a context *spec-sit*(s) which depends on the situation parameter s. Contexts depending on parameters will surely present problems requiring more study.

Besides that, the careful reader of the derivation will wonder what system of logic permits the manipulations involved, especially the substitution of sentences for variables followed by the immediate use of the results of the substitution. There are various systems that can be used, e.g. quasi-quotation as used in the Lisp or KIF, use of back-quotes, or the ideas of [McC79b]. Furthermore, the drafts of this paper have motivated a number of researchers to develop logics of context in which the above would be a valid derivation; these include [BM93, Nay94, AS, BBM]. However, at present we are more attached to the derivation than to any specific logical system.

As a further example, consider rules for lifting statements like those of section 1 to one in which we can express statements about Justice Holmes's opinion of the Sherlock Holmes stories.

5 Natural Deduction via Context

The formal theory of context can be used to represent inference and reason in the style of natural deduction. This requires lifting axioms (or lifting rules) to treat the context which a reasoning system is in as a formal object. If p is a sentence and we are in some context c, we define a new context *assuming*(c, p) so that it validates the following rules:

importation $\quad c : p \supset q \;\vdash\; assuming(c, p) : q$

discharge $\quad assuming(c, p) : q \;\vdash\; c : p \supset q$

Note that these rules can be replaced by lifting axioms. Thus **importation** is replaced by

(19) $(\forall cpq)(ist(c, p \supset q) \supset ist(assuming(c, p), q))$

To make the presentation simpler we write them in the rule form. An interesting rule which can be derived from the above is

assumption $\quad \vdash assuming(c, p) : p$

In analogy to the restriction to the rule of $\forall$ introduction in formal systems of natural deduction, we will have to restrict the rule of universal generalization to ensure that the variable being generalized does not occur free in any of the $assuming(c, p)$ terms of the current context.

To illustrate the rules we now give a natural-deduction style proof of the **above** theorem, which was introduced in §4. This theorem involves the lifting of the theory of above into the context of situation calculus. The proof should be compared to the Hilbert style proof which was given in §4.

Proof (above): We begin with the $\Rightarrow$ direction of formula 4

$$(20)\quad blocks: \quad (\forall xys)(on(x,y,s) \supset ist(spec\text{-}sit(s), on(x,y)))$$

It follows by universal instantiation that

$$(21)\quad blocks: \quad on(x,y,s) \supset ist(spec\text{-}sit(s), on(x,y))$$

By the **importation** rule we get

$$(22)\quad assuming(blocks, on(x,y,s)): \quad ist(spec\text{-}sit(s), on(x,y))$$

Therefore, by entering the $spec\text{-}sit(s)$ context we get

$$(23)\quad spec\text{-}sit(s): \quad on(x,y)$$

Now, from formulas 3 and 6 it follows that

$$(24)\quad c0: \quad ist(blocks, (\forall s) ist(spec\text{-}sit(s), (\forall xy)(on(x,y) \supset above(x,y))))$$

By entering *blocks* we get

$$(25)\quad blocks: \quad (\forall s) ist(spec\text{-}sit(s), (\forall xy)(on(x,y) \supset above(x,y)))$$

By instantiating the universal quantifier over situations we get

$$(26)\quad blocks: \quad ist(spec\text{-}sit(s), (\forall xy)(on(x,y) \supset above(x,y)))$$

Therefore, by propositional logic we have

$$(27)\quad blocks: \quad on(x,y,s) \supset ist(spec\text{-}sit(s), (\forall xy)(on(x,y) \supset above(x,y)))$$

Therefore, by the **importation** rule we get

$$(28)\quad assuming(blocks, on(x,y,s)): \quad ist(spec\text{-}sit(s), (\forall xy)(on(x,y) \supset above(x,y)))$$

Now, by entering the $spec\text{-}sit(s)$ context we get

$$(29)\quad spec\text{-}sit(s): \quad (\forall xy)(on(x,y) \supset above(x,y))$$

By logic from formulas 23 and 29 it follows that

(30) $\mathit{spec\text{-}sit}(s): \quad \mathit{above}(x, y)$

By exiting the $\mathit{spec\text{-}sit}(s)$ context we get

(31) $\mathit{assuming}(\mathit{blocks}, \mathit{on}(x, y, s)): \quad \mathit{ist}(\mathit{spec\text{-}sit}(s), \mathit{above}(x, y))$

The $\Leftarrow$ direction of formula 5

(32) $\mathit{blocks}: \quad (\forall xys)\mathit{ist}(\mathit{spec\text{-}sit}(s), \mathit{above}(x, y)) \supset \mathit{above}(x, y, s)$

By propositional logic we have

(33) $\mathit{blocks}: \quad \mathit{on}(x, y, s) \supset (\forall xys)\mathit{ist}(\mathit{spec\text{-}sit}(s), \mathit{above}(x, y)) \supset \mathit{above}(x, y, s)$

Together with the **importation** rule the above formula allows us to infer

(34) $\mathit{assuming}(\mathit{blocks}, \mathit{on}(x, y, s)): \quad (\forall sxy)\mathit{ist}(\mathit{spec\text{-}sit}(s), \mathit{above}(x, y)) \supset \mathit{above}(x, y, s)$

By logic from (31) and (34) we get

(35) $\mathit{assuming}(\mathit{blocks}, \mathit{on}(x, y, s)): \quad \mathit{above}(x, y, s)$

Using the rule **discharge** it follows that

(36) $\mathit{blocks}: \quad \mathit{on}(x, y, s) \supset \mathit{above}(x, y, s)$

Therefore, by universal generalization we obtain what was to be proved

(37) $\mathit{blocks}: \quad (\forall sxy)\mathit{on}(x, y, s) \supset \mathit{above}(x, y, s)$

$\square_{\textbf{above}}$

In the above proof we have entered the context $\mathit{assuming}(c, p)$ in a number of instances. This creates an interesting example because it might not be obvious in which context the term $\mathit{assuming}(c, p)$ is to be interpreted. However, since the logic needs to keep track of which contexts were entered in the process of reasoning, the answer becomes obvious: the term $\mathit{assuming}(c, p)$ will be interpreted in the next outer context (see §3 for discussion on sequences of contexts).

5.1 Postponing Preconditions via *assuming*

We conclude by noting that the *assuming* function, as defined in this section, is also useful for formalizing a number of other phenomena. Examine a naive formalism for reasoning about action where the preconditions for flying are given by the formula

(38) $c: \quad \textit{have-ticket}(x) \wedge \textit{clothed}(x) \supset \textit{can-fly}(x).$

In common sense reasoning we want the ability to *postpone* dealing with the precondition of being clothed. This can be done by considering a context which assumes that one is clothed $\textit{assuming}(c, \textit{clothed}(x))$. By the **importation** rule and the formula 38 we get

(39) $\textit{assuming}(c, \textit{clothed}(x)): \quad \textit{have-ticket}(x) \supset \textit{can-fly}(x).$

Thus in the context $\textit{assuming}(c, \textit{clothed}(x))$ we do not need to consider the precondition of being clothed in order to infer that one can fly.

Note that we are only developing an ontology for representing this phenomena, and are not dealing with pragmatic issues like which context a reasoning system will start in, and how the system will decide to consider a context making an additional assumption. In fact, from a pragmatic viewpoint the above process might need to be completely reversed. The reasoning system may realize that its current problem solving context c is making a particular assumption p that needs to be discharged. Then it will need to consider a context c' such that $c = \textit{assuming}(c', p)$.

The *assuming* function is also needed for representing discourse. In §7 we show how it is used to handle replies to a query; in that section we call the *assuming* function "*reply*".

6 Integrating Databases

We see the use of formalized contexts as one of the essential tools for reaching human level intelligence by logic based methods. However, formalized contexts have shorter term applications. In this section we deal with one short term application: we show how two data bases, which were not originally intended to be used together, can be integrated by lifting their contents into a wider context. We proceed with an example.

6.1 The GE, Navy, and Air Force Example

Here's a hypothetical example. Imagine that the Navy, the Air Force and General Electric have separately developed standards for databases containing facts about prices of jet engines and parts. But these standards are not the same. Suppose that associated with each item is a price. Suppose further

1. For GE, the price is a retail price not including spare parts.
2. For the Navy, the price is the Government's purchase price including spare parts.
3. For the Air Force, the price includes additional inventory costs. It includes spare parts but a different assortment than the Navy's.

Now suppose that associated with each database are many programs that use the information. For example, General Electric can compute the cost of equipment packages taking into account discounts. The Navy can compute the economic ordering quantity for use when supplies get low.

Suppose now that some plan requires that unexpectedly a certain item made by General Electric is required in large quantity by both the Navy and the Air Force and deliveries and purchases from various General Electric warehouses have to be scheduled in co-ordination. The context in which the reasoning is done requires the *lifting* of various information from the contexts of the separate databases to the reasoning context. In the course of this lifting, the sentences representing the information are translated into new forms appropriate for the reasoning context.

6.2 A Simple Formalization

In this simple case, assume that the Air Force and Navy data bases need to be updated on the new General Electric prices. The GE database lists the list price, i.e. the price at which GE is selling the engine. The Navy database lists the price which Navy *will need to pay* for the engine and its assortment of spare parts, if it decides to use GE.

In order to reason with multiple databases, c_{ps}, an ad hoc context for reasoning about the particular problem, may be required. The problem solving context c_{ps} contains objects denoting the General Electric context c_{GE}, the Navy context c_{N}, and the Air Force context c_{AF}. This enables us to talk about facts which are contained in the corresponding databases. If for example the GE database contains a fact $\mathit{price}(\mathit{FX\text{-}22\text{-}engine}) = \$3600K$ then the sentence $\mathit{ist}(c_{\mathrm{GE}}, \mathit{price}(\mathit{FX\text{-}22\text{-}engine}) = \$3600K)$ is true in c_{ps}.

Different data bases might make different assumptions. For instance, prices of engines in some contexts might or might not include spare parts or warranties. We need the ability to represent this information in c_{ps}. Function *spares*, when given a product and a context, returns the spares which the given context assumes necessary and thus includes in the price of the product. For example, $\mathit{spares}(c_{\mathrm{NAVY}}, x)$ is the set of spares that Navy assumes will be included in the price of the product x. Function *warranty*, when given a product and a context, returns the name of the warranty assumed for the product in the given context. For example, $\mathit{warranty}(c_{\mathrm{NAVY}}, x)$ is the name of the warranty which Navy assumes is included in the price of the product x. In this note we are treating warranty in the same manner as we would treat spare parts or additional optional features. It would be the responsibility of another formalization to "understand" the warranty and give axioms describing the exact obligations that GE has to its clients. Note that information about spares and warranties assumed by the Navy will probably not be contained in the Navy data base. (Otherwise, we would use $\mathit{value}(c_{\mathrm{NAVY}}, \mathit{spares}(x))$ rather than $\mathit{spares}(c_{\mathrm{NAVY}}, x)$ to refer to the spares that Navy assumes will be included in the price of the product x.) Rather, this information is kept in in some manual. But for these data bases to be used jointly, the spares information needs to be included; we assume that it is described declaratively in c_{ps}. Finally, the vocabulary of c_{ps} also has a function *GE-price*, which to every object assigns its corresponding price in dollars.

In the problem solving context c_{ps} we also represent the fact that GE lists engine prices without any spares, while Navy assumes spare parts to be included in the price of a product. This is done by lifting axioms, which define how the notion of price in different databases translates into the problem solving context:

$$(40)\quad c_{\text{ps}}: \quad (\forall x) value(c_{\text{GE}}, price(x)) = \textit{GE-price}(x)$$

$$(41)\quad c_{\text{ps}}: \quad (\forall x) value(c_{\text{NAVY}}, price(x)) = \textit{GE-price}(x) + \textit{GE-price}(spares(c_{\text{NAVY}}, x)) + \textit{GE-price}(warranty(c_{\text{NAVY}}, x))$$

expressing that the price listed in the Navy data base is the price of the engine, some bag of spares, and the particular warranty that are assumed by the Navy.

$$(42)\quad c_{\text{ps}}: \quad (\forall x) value(c_{\text{AF}}, price(x)) = f(x, \textit{GE-price}(x), \textit{GE-price}(spares(c_{\text{AF}}, x)), \textit{GE-price}(warranty(c_{\text{AF}}, x)))$$

where f is some function which determines the total price of an item and spares, also taking into account the inventory cost. Note that f might not be precisely known, in which case we might decide to only give some approximate bounds on f.

Now we work out an example. Assume that we are given the prices as listed in the GE data base; i.e. the following formulas hold in c_{GE}:

$$(43)\quad c_{\text{GE}}: \quad price(\textit{FX-22-engine}) = \$3600K$$

$$(44)\quad c_{\text{GE}}: \quad price(\textit{FX-22-engine-fan-blades}) = \$5K$$

$$(45)\quad c_{\text{GE}}: \quad price(\textit{FX-22-engine-two-year-warranty}) = \$6K$$

Information about spares and warranties will not be found in the c_{GE} data base and will probably require looking up in some manual or description of the the data base. We need to enter this information into the the problem solving context:

$$(46)\quad c_{\text{ps}}: \quad spares(c_{\text{NAVY}}, \textit{FX-22-engine}) = \textit{FX-22-engine-fan-blades}$$

$$(47)\quad c_{\text{ps}}: \quad warranty(c_{\text{NAVY}}, \textit{FX-22-engine}) = \textit{FX-22-engine-two-year-warranty}$$

Then we can compute the price of the FX-22 jet engine for the Navy. The following formula is a theorem, i.e. it follows from the above axioms.

Theorem (engine price):

$$c_{\text{NAVY}}: \quad price(\textit{FX-22-engine}) = \$3611K$$

In order to compute this price for the Air Force, the inventory cost given by function f would need to be known.

Proof (engine price): First we exit the c_{GE} context thus rewriting formulas 43, 44, and 45 as

$$(48)\quad c_{\mathrm{ps}}: \quad value(c_{\mathrm{GE}}, price(\textit{FX-22-engine})) = \$3600K$$

$$(49)\quad c_{\mathrm{ps}}: \quad value(c_{\mathrm{GE}}, price(\textit{FX-22-engine-fan-blades})) = \$5K$$

$$(50)\quad c_{\mathrm{ps}}: \quad value(c_{\mathrm{GE}}, price(\textit{FX-22-engine-two-year-warranty})) = \$6K$$

From formulas 40 and 48 it follows that

$$(51)\quad c_{\mathrm{ps}}: \quad \textit{GE-price}(\textit{FX-22-engine}) = \$3600K$$

From formulas 40 and 49 it follows that

$$(52)\quad c_{\mathrm{ps}}: \quad \textit{GE-price}(\textit{FX-22-engine-fan-blades}) = \$5K$$

Therefore, using formula 46, we get

$$(53)\quad c_{\mathrm{ps}}: \quad \textit{GE-price}(spares(c_{\mathrm{NAVY}}, \textit{FX-22-engine})) = \$5K$$

In a similar fashion, from formulas 40, 47 and 50 we can conclude that

$$(54)\quad c_{\mathrm{ps}}: \quad \textit{GE-price}(warranty(c_{\mathrm{NAVY}}, \textit{FX-22-engine})) = \$6K$$

From formulas 51, 53, and 54 if follows that

$$(55)\quad c_{\mathrm{ps}}: \quad \textit{GE-price}(\textit{FX-22-engine}) + \textit{GE-price}(spares(c_{\mathrm{NAVY}}, \textit{FX-22-engine})) + \textit{GE-price}(warranty(c_{\mathrm{NAVY}}, x)) = \$3611K$$

Then, using formula 41 we can conclude that

$$(56)\quad c_{\mathrm{ps}}: \quad value(c_{\mathrm{NAVY}}, price(\textit{FX-22-engine})) = \$3611K$$

By entering c_{NAVY} we get

$$(57)\quad c_{\mathrm{NAVY}}: \quad price(\textit{FX-22-engine}) = \$3611K$$

$\square_{\textbf{engine-price}}$

In the above proof we are assuming that all constants denote the same object in all contexts, i.e. that all constants are *rigid designators*. Consequently constants can be substituted for universally quantified variables by the universal instantiation rule. Generalizing the proof is straight forward if we drop this assumption.

6.3 Formalization for Bargaining

Now assume that the air force database contains the price air force *plans to pay* for a product, i.e. the *included in the budget.* Like before, the GE database contain the list price, which will probably be higher than the air force budget price. This formalization is suited for use by some bargaining agents or programs. The bargaining agent for the air force will through negotiation attempt to convince the GE agent to lower the GE list price to the air force budget price (or some price that would be acceptable to the air force).

The bargaining agents will work in some problem solving context c_{ps}. This context contains constants denoting the various data bases which will be relevant to the bargaining; in our case these will be the General Electric context c_{GE}, and the Air Force context c_{AF}. Context c_{ps} contains functions which represent the budget price and the list price of a product. Function *manufacturer-price*, when given a context of a manufacturer and a product, returns the price at which the product is offered for sale by the manufacturer; functions *budget-price*, when given a context of a customer and a product, returns the price which the customer is willing to pay for the product. Like in the previous example, c_{ps} can represent the spares associated with an engine. Function *spares*, when given a product and an object, returns the spares which the given context assumes necessary and thus included in the price of the product.

The air force and GE will need to bargain in order to negotiate a price which is acceptable to both parties. However, since unlike GE, the air force assumes that the price will include a set of spare parts, the lifting axioms will be needed to adjust the prices in the two data bases to ensure that both the budget price and the list price pertain to the same package. The lifting axioms are:

(58) $c_{\mathrm{ps}} : (\forall x) value(c_{\mathrm{GE}}, price(x)) = manufacturer\text{-}price(c_{\mathrm{GE}}, x)$

(59) $c_{\mathrm{ps}} : (\forall x) value(c_{\mathrm{AF}}, price(x)) = budget\text{-}price(c_{\mathrm{AF}}, x) + budget\text{-}price(c_{\mathrm{AF}}, spares(c_{\mathrm{AF}}, x))$

The lifting axioms will enable us to derive the *budget-price* and *manufacturer-price* prices in c_{ps}, both of which pertain to the engine only, excluding any spares. These can then be used by the bargaining programs to negotiate a price and administrate a sale.

Note again the difference between this formalization and the previous one. In the previous subsection the *price* function in both data bases referred to the price which was actually being paid for a product. Therefore, the lifting axioms were used to directly infer the price in one data base based on the price listed in another. In this example, on the other hand, given the list price the lifting axioms can not be used to work out the budget price. The lifting axioms simply ensure that both the list price and the budget price talk only about the engine, and do not implicitly assume the inclusion of spare parts.

6.4 Treating *value* as an Abbreviation

It will be possible to define *value* as an abbreviation in a context language which contains the *ist*. We first deal with the case where all contexts have the same domains. We define *value* as an abbreviation:

(60) $value(c, x) = y \quad \equiv \quad (\forall z) y = z \equiv ist(c, x = z)$

Eliminating the *value* abbreviation, the above formulas are equivalent to:

(61) $c_{\mathrm{ps}} : (\forall xy) ist(c_{\mathrm{GE}}, y = price(x)) \equiv y = GE\text{-}price(x)$

(62) $c_{\mathrm{ps}} : (\forall xy) ist(c_{\mathrm{NAVY}}, y = price(x)) \equiv y = GE\text{-}price(x) + GE\text{-}price(spares(c_{\mathrm{NAVY}}, x))$

(63) $c_{\mathrm{ps}} : (\forall xy) ist(c_{\mathrm{AF}}, y = price(x)) \equiv y = f(x, GE\text{-}price(x), GE\text{-}price(spares(c_{\mathrm{AF}}, x)))$

If the domains of all the contexts are not the same, then the above formulas are not intuitively correct. Instead, a domain precondition needs to be added to all the formulas. For example instead of formula 61, we would write

(64) $c_{\mathrm{ps}} : (\forall xy)(in(c,x) \wedge in(c,y)) \supset (ist(c_{\mathrm{GE}}, y = price(x)) \equiv y = GE\text{-}price(x))$

where $in(c,x)$ is true iff the object denoted by x is in the domain of the context denoted by c. [1]

Note however, if we simply change the abbreviation of *value* to

(65) $value(c,x) = y \quad \equiv \quad in(c,x) \supset (\forall z)(in(c,z) \supset (y = z \equiv ist(c, x = z)))$

then the axioms involving *value* (axioms 40-42 and 58-59) will still be true. In other words, the previous formalizations remain unaltered. To verify this, note that substituting this new definition for value (given in formula 65) into formula 40 gives us formula 64, rather then formula 61.

We also need to assert that the problem solving context c_{ps} contains all the objects present in the other contexts which are involved in the particular problem solving process. In some outer context $c0$ we would write:

(66) $c0 : (\forall c) involved\text{-}in\text{-}ps(c) \supset (\forall x)(in(c,x) \supset in(c_{\mathrm{ps}}, x))$.

In both cases mentioned above, the rules of entering and exiting a context for the *value* function will follow from the general rules enter and exit for the *ist*.

7 Representing Discourse

Formal theory of context is needed to provide a representation of the context associated with a particular circumstance. In this section we illustrate this by showing how our formalism can be used to represent the context of a conversation in which terms have particular meanings that they wouldn't have in the language in general. We examine question/answer conversations which are simply sequence of questions and answers. In this simple model we allow two types of questions:

[1] The main implication connective in this formula will probably not be classical. However this is a technical point which we address elsewhere.

propositional questions are used to inquire whether a proposition is true or false; they require a yes or no answer. In the language we introduce a special proposition *yes* which is used to answer these questions.

qualitative questions are used to find the objects of which a formula holds; in the language we introduce a unary predicate *answer* which holds of these objects.

In order to know what is being communicated in a discourse, as well as reason about a discourse in general, we need a way of representing the discourse. To do this in the framework of the formal theory of context, we identify a new class of contexts, the *discourse contexts*. Discourse contexts are not only characterized by the sentences which are true in them but also by the intended meaning of their predicates, which might vary from one discourse context to the next.

We represent a discourse with a sequence of discourse contexts, each of which in turn represents the discourse state after an utterance in the discourse. Our attention is focused only on discourses which are sequences of questions and replies: $[q^1, r^1, q^2, r^2, \ldots, q^n, r^n]$. Thus, we can represent such a discourse with a sequence of discourse contexts:

$$[c_{\mathrm{d}}, \mathit{query}(c_{\mathrm{d}}, q^1), \mathit{reply}(\mathit{query}(c_{\mathrm{d}}, q^1), r^1), \ldots$$

$$\ldots, \mathit{reply}(\mathit{query}(\mathit{reply}(\cdots \mathit{reply}(\mathit{query}(c_{\mathrm{d}}, q^1), r^1) \cdots, r^{n-1}), q^n), r^n)]$$

s.t. (i) ${c_{\mathrm{d}}}^0$ is some discourse context in which the initial question (q^1) was asked; (ii) the function *query* takes a question ϕ and some discourse context c_{d} (representing the discourse state before the question ϕ) and returns the discourse context representing the discourse state after asking the question ϕ in c_{d}; (iii) the function *reply* takes a reply ϕ and some discourse context c_{d} (representing the discourse state before before replying ϕ) and returns the discourse context representing the discourse state after replying ϕ in c_{d}. In order to reason about the discourse we now only need the properties of the functions *query* and *reply*. These will be made precise in the next subsection.

Since we are not concerned with solving the syntactic and semantic problems addressed by the natural language community, we are assuming the system is given the discourse utterances in the form of logical formulas. This assumption is in line with [McC90a]; in McCarthy's terminology we would say that the discourse has been processed by both the parser and the understander to produce a logical theory. Note that the process of producing this theory is not precisely defined, and it is not completely clear how much common sense information is needed to generate it. It might turn out that producing such a theory requires the solution of the problem we had set out to solve. But for the time being let us take a positive perspective and assume the discourse theory is given.

Note that our simple model will not capture a number of other aspects of a discourse state, as have been studied by computational linguists [GS86]. For example, we have completely ignored all pragmatic issues which are in fact considered central to discourse analysis. Extending the notion of the discourse state is part of our plan for future research.

7.1 The Logic of *query* and *reply*

In this section we give the properties of the functions *query* and *reply*, which are central for representing question/answer discourses. Since the *query* and *reply* functions are treated in the style of situation calculus, we do not need to change our basic theory of context, but simply give the axioms that formalize the two functions.

Intuitively, the *query* function will set up a context in which the reply to the question will be interpreted. For example, the context resulting in asking some proposition p will have the property that *yes* in that context will be interpreted as p. Thus *query* only changes the semantic state of the discourse context. The *reply* function will do a simple update of information: the formulas true in the context resulting in replying p in c_{d} will be exactly those formulas which are conditionally true on p in c_{d}. Thus the *reply* function only changes the epistemic state of the discourse context. We now make these notions more precise.

The following axioms characterize the functions *query* and *reply*.

interpretation axiom (propositional) if ϕ is a closed formula, then

$$ist(query(\kappa, \phi), \phi \equiv yes)$$

frame axiom (propositional) if ϕ is a closed formula, and *yes* does not occur in ψ, then

$$ist(\kappa, \psi) \supset ist(query(\kappa, \phi), \psi)$$

interpretation axiom (qualitative) if x is the only variable occurring free in ϕ, then

$$ist(query(\kappa, \phi(x)), \phi(x) \equiv answer(x))$$

frame axiom (qualitative) if x is the only variable occurring free in ϕ, and *answer* does not occur in ψ, then

$$ist(\kappa, \psi) \supset ist(query(\kappa, \phi(x)), \psi)$$

reply axiom $\quad ist(reply(\kappa, \phi), \psi) \equiv ist(\kappa, \phi \supset \psi)$

We proceed to illustrate the axioms and their use with an example.

7.2 Example: Air Force–GE Discourse

We examine the following hypothetical discourse taking place between the Air Force and General Electric:

1. AF: Will you bid on the engine for the FX22?

2. GE: Yes.

3. AF: What is your bid?
4. GE: \$4M.
5. AF: Does that include spares?
6. GE: Yes.

We transcribe the above discourse in our logic as a sequence of discourse contexts, s.t.

$c1 = query(c, \textit{will-bid-on}(engine(\mathrm{FX22})))$

$c2 = reply(c1, yes)$

$c3 = query(c2, price(engine(\mathrm{FX22}), x))$

$c4 = reply(c3, answer(\$4M))$

$c5 = query(c4, price(x) \equiv ist(c_{\mathrm{kb}}, \textit{price-including-spares}))$

$c6 = reply(c5, yes)$

where c is the initial discourse context. To simplify presentation, in this section we take *price* to be a predicate; in §4 we have illustrated how it can be treated as a function by using *value* instead of *ist*.

7.3 Deriving Properties of the Air Force–GE Discourse

We now show some properties of the discourse which can be derived with our logic.

7.3.1 First Question: Propositional Case

The discourse begins with a propositional question. We show how they modify the discourse state.

Theorem ($c2$): $ist(c2, \textit{will-bid-on}(engine(\mathrm{FX22})))$

Proof ($c2$): Instantiating the first axiom for the propositional questions, we get

$$ist(query(c, \textit{will-bid-on}(engine(\mathrm{FX22}))), \textit{will-bid-on}(engine(\mathrm{FX22})) \equiv yes)$$

which, by definition of $c1$, can be written as

$$ist(c1, \textit{will-bid-on}(engine(\mathrm{FX22})) \equiv yes)$$

Instantiating the axiom for reply we have

$$ist(reply(c1, yes), \textit{will-bid-on}(engine(\mathrm{FX22}))) \equiv ist(c1, yes \supset \textit{will-bid-on}(engine(\mathrm{FX22})))$$

and it follows from the two lines above that

$$ist(reply(c1, yes), \textit{will-bid-on}(engine(\mathrm{FX22})))$$

which by definition of $c2$ we can write as

$$ist(c2, \textit{will-bid-on}(engine(\mathrm{FX22})))$$

$\square_{\mathbf{c2}}$

7.3.2 Second Question: Qualitative Case

The reasoning for this qualitative question is similar to the propositional question.

Theorem ($c4$): $ist(c4, price(engine(\mathrm{FX22}), \$4M))$

Proof ($c4$): We begin with an instance of the first axiom for qualitative questions

$$ist(query(c2, price(engine(\mathrm{FX22}), x)), price(engine(\mathrm{FX22}), x) \equiv answer(x))$$

which, by definition of $c3$, can be written as

$$ist(c3, price(engine(\mathrm{FX22}), x) \equiv answer(x))$$

Instantiating the axiom for reply we have

$$ist(reply(c3, answer(\$4M)), price(engine(\mathrm{FX22}), \$4M)) \equiv$$

$$\equiv ist(c3, answer(\$4M) \supset price(engine(\mathrm{FX22}), \$4M))$$

and it follows from the two lines above that

$$ist(reply(c3, answer(\$4M)), price(engine(\mathrm{FX22}), \$4M))$$

which by definition of $c4$ we can write as

$$ist(c4, price(engine(\mathrm{FX22}), \$4M))$$

$\Box_{\mathbf{c4}}$

Due to the frame axioms, the conclusion established in the first question

$$ist(c2, \textit{will-bid-on}(engine(\mathrm{FX22})))$$

also holds in context $c4$.

Theorem (frame): $ist(c2, \textit{will-bid-on}(engine(\mathrm{FX22})))$

Proof (frame): We first instantiate the second axiom for qualitative questions to get

$$ist(c2, \textit{will-bid-on}(engine(\mathrm{FX22}))) \supset$$

$$\supset ist(query(c2, price(engine(\mathrm{FX22}), x)), \textit{will-bid-on}(engine(\mathrm{FX22})))$$

The two lines above imply

$$ist(query(c2, price(engine(\mathrm{FX22}), x)), \textit{will-bid-on}(engine(\mathrm{FX22})))$$

which, by definition of $c3$, can be written as

$$ist(c3, \textit{will-bid-on}(\textit{engine}(\mathrm{FX22})))$$

Now we apply the following instance of the reply axiom

$$ist(\textit{reply}(c3, \textit{answer}(\$4M)), \textit{will-bid-on}(\textit{engine}(\mathrm{FX22}))) \equiv$$

$$\equiv ist(c3, \textit{answer}(\$4M) \supset \textit{will-bid-on}(\textit{engine}(\mathrm{FX22})))$$

to get

$$ist(\textit{reply}(c3, \textit{answer}(\$4M)), \textit{will-bid-on}(\textit{engine}(\mathrm{FX22})))$$

which, by definition of $c4$, can be written as

$$ist(c4, \textit{will-bid-on}(\textit{engine}(\mathrm{FX22})))$$

$\Box_{\mathbf{frame}}$

7.3.3 Third Question: Dealing with Ambiguity

We are assuming that the predicate *price* is ambiguous in the discourse contexts since it can be ambiguously interpreted as either *price-including-spares* or as *price-not-including-spares* in some knowledge base. In the third question the predicate is disambiguated for context $c6$. This will allow us to prove that the GE bid on the FX22 engine is \$4M including spare parts. Note that we will have to state the above in the kb context because the discourse contexts are not expressive enough to distinguish between the price including spares and the price excluding spares (which in fact was the source of ambiguity).

Theorem (kb): $ist(c_{\mathrm{kb}}, \textit{price-including-spares}(\textit{engine}(\mathrm{FX22}), \$4\mathrm{M}))$

Proof (kb): By reasoning similar to the first question, we can conclude

$$ist(c6, \textit{price}(x, y)) \equiv ist(c_{\mathrm{kb}}, \textit{price-including-spares}(x, y))$$

From the frame axioms we get

$$ist(c6, \textit{price}(\textit{engine}(\mathrm{FX22}), \$4M))$$

similarly to the frame derivation in the second question. Now the theorem follows from the above formulas. $\Box_{\mathbf{kb}}$

8 Transcending Contexts

Human intelligence involves an ability that no-one has yet undertaken to put in computer programs—namely the ability to *transcend* the context of one's beliefs.

That objects fall would be expected to be as thoroughly built into human mental structure as any belief could be. Nevertheless, long before space travel became possible, the possibility of weightlessness was contemplated. It wasn't easy, and Jules Verne got it wrong when he thought that there would be a turn-over point on the way to the moon when the travellers, who had been experiencing a pull towards the earth would suddenly experience a pull towards the moon.

In fact, this ability is required for something less than full intelligence. We need it to be able to comprehend someone else's discovery even if we can't make the discovery ourselves. To use the terminology of [MH69], it is needed for the *epistemological* part of intelligence, leaving aside the heuristic.

We want to regard the system as being at any time within an implicit outer context; we have used $c0$ in this paper. Thus a sentence p that the program believes without qualification is regarded as equivalent to $ist(c0, p)$, and the program can therefore infer $ist(c0, p)$ from p, thus *transcending* the context $c0$. Performing this operation again should give us a new outer context, call it c_{-1}. This process can be continued indefinitely. We might even consider continuing the process transfinitely, for example, in order to have sentences that refer to the process of successive transcendence. However, I have no present use for that.

However, if the only mechanism we had is the one described in the previous paragraph, transcendence would be pointless. The new sentences would just be more elaborate versions of the old. The point of transcendence arises when we want the transcending context to relax or change some assumptions of the old. For example, our language of adjacency of physical objects may implicitly assume a gravitational field, e.g. by having relations of *on* and *above.* We may not have encapsulated these relations in a context. One use of transcendence is to permit relaxing such implicit assumptions.

The formalism might be further extended to provide so that in c_{-1} the whole set of sentences true in c_0 is an object $truths(c0)$.

Transcendence in this formalism is an approach to formalizing something that is done in science and philosophy whenever it is necessary to go from a language that makes certain assumptions to one that does not. It also provides a way of formalizing some of the human ability to make assertions about one's own thoughts.

The usefulness of transcendence will depend on there being a suitable collection of non-monotonic rules for *lifting* sentences to the higher level contexts.

As long as we stay within a fixed outer context, it seems that our logic could remain ordinary first order logic. Transcending the outermost context seems to require a changed logic with what Tarski and Montague call *reflexion principles.* They use them for sentences like $true(p*) \equiv p$, e.g " 'Snow is white.' is true if and only if snow is white."

The above discussion concerns the epistemology of transcending contexts. The heuristics of transcendence, i.e. when a system should transcend its outer context and how, is entirely an open subject.

9 Relative Decontextualization

Quine [1969] uses a notion of "eternal sentence", essentially one that doesn't depend on context. This seems a doubtful idea and perhaps incompatible with some of Quine's other ideas, because there isn't any language in which eternal sentences could be expressed that doesn't involve contexts of some sort. We want to modify Quine's idea into something we can use.

The usefulness of eternal sentences comes from the fact that ordinary speech or writing involves many contexts, some of which, like pronoun reference, are valid only for parts of sentences. Consider, "Yes, John McCarthy is at Stanford University, but he's not at Stanford today". The phrase "at Stanford" is used in two senses in the same sentence. If the information is to be put (say) in a book to be read years later by people who don't know McCarthy or Stanford, then the information has to be decontextualized to the extent of replacing some of the phrases by less contextual ones.

The way we propose to do the work of "eternal sentences" is called *relative decontextualization.* The idea is that when several contexts occur in a discussion, there is a common context above all of them into which all terms and predicates can be lifted. Sentences in this context are "relatively eternal", but more thinking or adaptation to people or programs with different presuppositions may result in this context being transcended.

10 Mental States as Outer Contexts

A person's state of mind cannot be adequately regarded as the set of propositions that he believes—at least not if we regard the propositions as sentences that he would give as answers to questions. For example, as we write this we believe that George Bush is the President of the United States, and if we were entering information in a database, we might write

$$president(U.S.A.) = George.Bush.$$

However, my state of mind includes, besides the assertion itself, my reasons for believing it, e.g. he has been referred to as President in today's news, and we regard his death or incapacitation in such a short interval as improbable. The idea of a TMS or reason maintenance system is to keep track of the pedigrees of all the sentences in the database and keep this information in an auxiliary database, usually not in the form of sentences.

Our proposal is to use a database consisting entirely of *outer* sentences where the pedigree of an *inner* sentence is an auxiliary parameter of a kind of modal operator surrounding the sentence. Thus we might have the outer sentence

$$believe(president(U.S.A.) = George.Bush, because \ldots),$$

where the dots represent the reasons for believing that Bush is President.

The use of formalized contexts provides a convenient way of realizing this idea. In an outer context, the sentence with reasons is asserted. However, once the system has committed itself to reasoning with the proposition that Bush is President, it enters an inner context with the simpler assertion

$$president(U.S.A.) = George.Bush.$$

If the system then uses the assertion that Bush is President to reach a further conclusion, then when it leaves the inner context, this conclusion needs to acquire a suitable pedigree.

Consider a belief revision system that revises a database of beliefs solely as a function of the new belief being introduced and the old beliefs in the system. Such systems seem inadequate even to take into account the information used by TMS's to revise beliefs. However, it might turn out that such a system used on the outer beliefs might be adequate, because the consequent revision of inner beliefs would take reasons into account.

11 Remarks

1. Guha has put contexts into Cyc, largely in the form of microtheories. The *above-theory* example is a microtheory. See [Guh91] for some of the details.

2. We have mentioned various ways of getting new contexts from old ones: by specializing the time or place, by specializing the situation, by making abbreviations, by specializing the subject matter (e.g. to U.S. legal history), by making assumptions and by specializing to the context of a conversation. These are all specializations of one kind or another. Getting a new context by transcending an old context, e.g. by dropping the assumption of a gravitational field, gives rise to a whole new class of ways of getting new contexts.

   These are too many ways of getting new contexts to be treated separately.

3. We have used natural language examples in this article, although natural language is not our main concern. Nevertheless, we hope that formalizing context in the ways we propose may be useful in studying the semantics of natural language. Natural language exhibits the striking phenomenon that context may vary on a very small scale; several contexts may occur in a single sentence.

   Consider the context of an operation in which the surgeon says, "Scalpel". In context, this may be equivalent to the sentence, "Please give me the number 3 scalpel".

4. $ist(c, p)$ can be considered a modal operator dependent on c applied to p. This was explored in [Sho91].

5. It would be useful to have a formal theory of the natural phenomenon of context, e.g. in human life, as distinct from inventing a form of context useful for AI systems using logic for representation. This is likely to be an *approximate theory* in the sense described in [McC79a]. That is, the term "context" will appear in useful axioms and other sentences but will not have a definition involving "if and only if".

6. Useful nonmonotonic rules for lifting will surely be more complex than the examples given.

Acknowledgments

The development of these ideas has benefitted from discussions with Tom Costello, Mike Genesereth, Fausto Giunchiglia, R. V. Guha, Ian Mason, and Carolyn Talcott. Guha wrote his thesis [Guh91] while this article was going through many versions as the ideas developed, and the mutual influences cannot be specified.

This work was partly supported by DARPA contract NAG2-703 and ARPA/ONR grant N00014-94-1-0775.

References

[AS] Giuseppe Attardi and Maria Simi. A formalization of viewpoints. *Fundamenta Informaticae*. To appear. Also technical report TR-93-062 at the International Computer Science Institute, Berkeley, California 94704-1105.

[BBM] Saša Buvač, Vanja Buvač, and Ian A. Mason. Metamathematics of contexts. *Fundamenta Informaticae*. To appear.

[BM93] Saša Buvač and Ian A. Mason. Propositional logic of context. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, 1993.

[GS86] Barbara J. Grosz and Candace L. Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[Guh91] R. V. Guha. *Contexts: A Formalization and Some Applications*. PhD thesis, Stanford University, 1991. Also published as technical report STAN-CS-91-1399-Thesis, and MCC Technical Report Number ACT-CYC-423-91.

[KW84] Jussi Ketonen and Joseph S. Weening. EKL—an interactive proof checker: User's reference manual. Technical report, Computer Science Department, Stanford University, Stanford, California, 1984.

[McC79a] John McCarthy. Ascribing mental qualities to machines. In Martin Ringle, editor, *Philosophical Perspectives in Artificial Intelligence*. Humanities Press, 1979. Reprinted in [McC90b].

[McC79b] John McCarthy. First order theories of individual concepts and propositions. In Donald Michie, editor, *Machine Intelligence*, volume 9. Edinburgh University Press, Edinburgh, 1979. Reprinted in [McC90b].

[McC87] John McCarthy. Generality in artificial intelligence. *Comm. of ACM*, 30(12):1030–1035, 1987. Also in *ACM Turing Award Lectures, The First Twenty Years*, ACM Press, 1987; and reprinted in [McC90b].

[McC90a] John McCarthy. An example for natural language understanding and the AI problems it raises. In *Formalizing Common Sense: Papers by John McCarthy*. Ablex Publishing Corporation, 355 Chesnut Street, Norwood, NJ 07648, 1990.

[McC90b] John McCarthy. *Formalizing Common Sense: Papers by John McCarthy.* Ablex Publishing Corporation, 355 Chesnut Street, Norwood, NJ 07648, 1990.

[McC93] John McCarthy. Notes on formalizing context. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 1993.

[MH69] John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, Edinburgh, 1969. Reprinted in [McC90b].

[Nay94] P. Pandurang Nayak. Representing multiple theories. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, 1994.

[Sho91] Yoav Shoham. Varieties of context. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy.* Academic Press, 1991.

/@sail.stanford.edu:/u/buvac/e94/context.tex: created 1991 winter, latexed 1994 Oct 9 at 9:07 p.m.