First-Order Inference and the Interpretation of Questions and Answers

Johan Bos & Malte Gabsdil

Computerlinguistik
Universität des Saarlandes
Im Stadtwald, Postfach 151150
66041 Saarbrücken, Germany
{bos,gabsdil}@coli.uni-sb.de

Abstract

Building on work by Groenendijk and Stokhof, we develop a theory of question and answer interpretation for first-order formalisms. The proposed framework is less fine-grained than its higher-order ancestor, but instead offers attractive implementational properties as it deals with the combinatorial explosion problem underlying Groenendijk and Stokhof's original theory. To incorporate the treatment of questions and answers in a larger setting, we use an extension of Discourse Representation Theory to cover typical contextual phenomena such as anaphora and presupposition. The actual interpretation of the dialogue representation is done via a translation to first-order logic. A prototype implementation, using state-of-the-art theorem proving and model building facilities, supports the idea that this first-order approximation of the interpretation of questions and answers is indeed a useful one.

1. Introduction

This paper discusses the treatment of questions and answers in automatic dialogue understanding. Questions require answers, so an inference mechanism that determines whether an utterance is an appropriate answer to a question under discussion should be an elementary part of a dialogue system. Such a component obviously improves human-machine conversation.

We describe a theoretical account and its computational implementation of the interpretation of questions and answers in dialogue. Our first aim is to arrive at a formal definition of what counts as a proper answer to a posed question. Our second aim is to transfer the analysis of questions and answers into a framework that deals with other context-sensitive phenomena, such as pronouns and presuppositions. Our third aim, finally, is to implement these ideas in a prototype dialogue system.

More precisely, we show how first-order logic can be used to model questions and answers (building on work by Groenendijk and Stokhof), and present a computational framework, where state-of-the-art theorem provers and model builders perform the inferences required to determine the appropriateness of (possible) answers to questions. The entire framework is embedded in an extension of Discourse Representation Theory.

2. Modeling Questions

Questions are traditionally analyzed as sets of their possible answers (Hamblin, 1973; Karttunen, 1977). For instance, the question 'Who likes Paris', in a domain with two individuals Tim and Kim, denotes the set containing the answers:

(1) { 'Tim likes Paris', 'Tim does not like Paris',
'Kim likes Paris', 'Kim does not like Paris' }

These approaches are too weak to capture certain aspects of quantification, as they do not contain answers like 'Everybody likes Paris', or 'Only Kim likes Paris' (see Higginbotham (1996) for further discussion). Groenendijk and Stokhof (1984) argue that questions partition the logical space into mutually exclusive and jointly exhaustive sets of possible worlds which represent the different ways in which a question can be answered. Under this view, questions denote *sets of sets* of propositions. For 'Who likes Paris?', we have the answer set:

(2) { { 'Tim likes Paris', 'Kim likes Paris' },
{ 'Tim does not like Paris', 'Kim does not like Paris' },
{ 'Tim likes Paris', 'Kim does not like Paris' },
{ 'Tim does not like Paris', 'Kim likes Paris' } }

This approach adds more structure to the interpretation of questions and therefore offers a more sophisticated way for classifying answers. Following Groenendijk and Stokhof, answers remove those fields in the partition of a question with whom they are inconsistent. A *partial answer* is an answer inconsistent with at least one member of the question's partition, and consistent with all others. For instance, the answer 'Kim likes Paris' is only inconsistent with two members of the partition in (2). However, 'Tim has red hair' is not an answer as it is consistent with all fields in (2). A question is finally *resolved* (using Ginzburg's (1995) terminology) if only one consistent field is left. The answers 'Only Tim likes Paris' or 'Nobody likes Paris' are resolving answers, because they are consistent with only one field of partition (2).

We will use Groenendijk and Stokhof's approach for implementing questions and answers but modify it on two points. First, because we want to make use of first-order inference, we take propositions to denote truth-values instead of functions from states to truth-values. Second, computing the partitions in Groenendijk and Stokhof's theory is subject to a combinatorial explosion problem. To deal with this, we simplify the structure of partitions.

2.1. The Combinatorial Explosion Problem

From a computational perspective, the original approach of Groenendijk and Stokhof faces a serious problem. The size of partitions of single wh-questions grows exponentially in the size of the question's domain (by domain

	Consistency Checks				
Example	(3)	(4)	(5)	(6)	Result
‘Where do you want to go? I go everywhere. ’	yes	yes	no	no	proper answer
‘Where do you want to go? I don’t go to Paris.’	no	yes	yes	yes	proper answer
‘Where do you want to go? I go to Paris’	yes	yes	yes	no	proper answer
‘Where do you want to go? I go nowhere’	no	no	yes	yes	proper answer
‘Where do you want to go? I go somewhere’	yes	yes	yes	no	proper answer
‘Where do you want to go? I start in London’	yes	yes	yes	yes	improper answer
‘Where do you want to go? Paris is beautiful’	yes	yes	yes	yes	improper answer
‘Where do you want to go? Paris is a country’	no	no	no	no	improper answer

Figure 1: Illustration for determining proper answers to wh-questions.

we understand the individuals syntactically determined by the wh-clause, such as locations for ‘where’, persons for ‘who’, and so on). To check for consistency with every field of a wh-partition would mean making 2^n inferences (n being the size of the domain) which is computationally not feasible, except for toy domains.

To solve this problem, we assume that wh-questions are—much like quantifiers in natural language—segmented into a domain and a body. For wh-questions the domain is defined as above and is assumed to be nonempty, and the body is the property that should hold for the inquired members of the domain.

The general strategy pursued for interpreting answers is as follows: By using the domain and body of a question Q, it is possible to construct formulas of first order logic that coarsely describe Q’s partition. Taking these formulas to represent the “semantics” of Q, we then determine whether a proposition A is a *proper answer* by checking for consistency of A and Q. The next sections describe this in more detail.

2.2. First-Order Semantics of Questions

Suppose that a wh-question Q is translated into a formula with domain D, body B, and principal variable referent x. Then an answer A is defined as proper for a question Q if at least one of the propositions (3)–(6) is consistent, and at least one of them is inconsistent.

(3) $\forall$x[D(x) $\rightarrow$ B(x)] & A

(4) $\exists$x[D(x) & B(x)] & A

(5) $\exists$x[D(x) & $\neg$B(x)] & A

(6) $\neg\exists$x[D(x) & B(x)] & A

Compared to the original ideas in Groenendijk and Stokhof (1984), these four formulas reduce arbitrarily large partitions for single wh-questions to partitions with only three fields. Formula (3) characterizes the answer that the body of the question holds for every individual in the question domain, (4) and (5) are compatible with possible answers that state that at least one individual either does or does not have this property, and (6) represents the answer that no individual in the question domain has the property expressed by the body (Figure 1 gives some examples). Thus, a proper answer resembles a partial answer (in the sense of Groenendijk and Stokhof) on such a reduced partition. Assume, for example, a model with three individuals a, b, and c and a unary property P. The partition for the question ‘Who is P?’ would have eight fields standing for the possibilities that either a, b, and c are P, only a, b, or c is P, nobody is P or that P holds of either a and b, a and c, or b and c. A graphical representation of such a partition is depicted in the left part of Figure 2. The four first-order formulas in (3)–(6), on the other hand, describe a partition of three fields for the same question and domain, as illustrated by the right part of Figure 2.

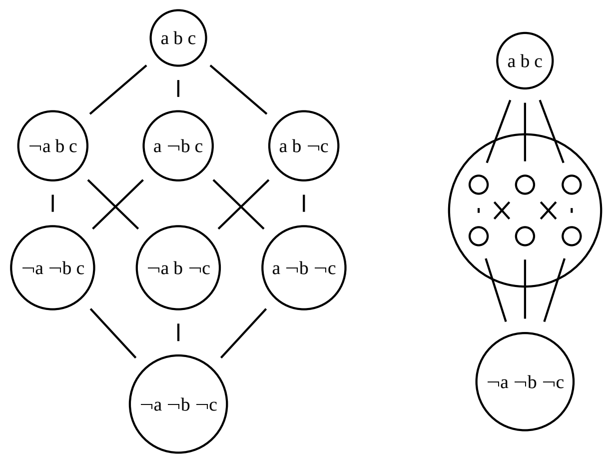


Figure 2: Two partitions for a wh-question with domain size 3, according to Groenendijk and Stokhof (left), and our simplified analysis (right). Note that since the two partitions have the same general structure, the right partition is included in the left partition by means of Groenendijk and Stokhof’s partition-inclusion operator $\sqsubseteq$.

Of course, the modified analysis means a loss of fine-grainedness with respect to Groenendijk and Stokhof’s original work, because it is not able to determine strong exhaustiveness of answers for wh-questions. For instance, our analysis would classify both ‘Only a is P’ and ‘a is P’ as proper answers to the question ‘Who is P?’, but does not recognize that the former is strongly exhaustive, and the latter is not. Whether such a fine distinction is required is debatable. Ginzburg discusses several examples where strong exhaustiveness seems to be an inappropriate resolvedness criterion (Ginzburg, 1996).

The important feature of our new analysis is that it copes with the combinatorial explosion problem, and thereby opens the way to computational implementation. Moreover, we believe that the approach naturally extends to yes/no and choice-questions. For yes/no-questions, the four formulas in (3)–(6) collapse into two by equivalence. The remaining formulas represent the answers 'yes' and 'no', which in turn model the bipartitions for yes/no-questions assumed by Groenendijk and Stokhof. This means that in the case of yes/no-questions, a proper answer can be identified with a strongly exhaustive answer.

3. Questions and Answers in DRT

The previous section outlined our analysis of questions and the interpretation of answers. This section describes how we can embed it in a framework for dialogue analysis, namely Discourse Representation Theory (Kamp and Reyle, 1993). This shift does not mean that we say farewell to first-order logic. In fact, for the interpretation of Discourse Representation Structures (DRSs, the representations used in DRT), we use a translation to first-order formulas.

Originally, DRT focuses on discourse, and puts forward concrete proposals to deal with anaphora and presupposition. To deal with specific dialogue phenomena, we use some of the extensions to DRT as proposed by (Poesio and Traum, 1998), to wit the integration of dialogue acts in DRSs and operations on DRSs for modeling *grounding acts*. The implementation of grounding slightly deviates from Poesio & Traum (see below). The treatment of questions and answer is a novel extension to Poesio & Traum's model.

3.1. Defining Dialogue Representations

Using DRSs for the analysis of dialogue requires at least three simple extensions to the basic syntax of DRSs. First, we have DRS-merging for two DRSs K and K′ resulting in a new DRS (K;K′). Second, questions are represented by the DRS (K?K′), where K and K′ are DRSs, representing the domain and body of a question, respectively. Third, DRS-conditions can be formed by τ:K, where τ is a discourse referent and K a DRS. This latter extension associates discourse referents with DRSs, and hence allows us to connect discourse referents with questions and answers. A sortal ontology on discourse referents assures that discourse referents for questions or answers are disjoint from other entities in the domain of interpretation. This ontological information, and other supportive background knowledge, is assumed to be part of the main DRS in the following discussion.

3.2. Building Dialogue Representations

A new utterance is translated to a DRS K and appended to the dialogue representation by conditions 'x:K' and 'P(x)', where x is a fresh discourse referent associated with K, and P a relation symbol specifying sortal information (e.g., whether it is a question or a proposition). Further, there is a condition 'under-discussion(x)' that marks that x is currently under discussion. Additional conditions are introduced stating the dialogue act associated with the utterance. As basic dialogue acts we have 'ask', 'reask', 'check', and 'assert'.

Context-dependent phenomena are dealt with as described in (Blackburn et al., 1999), using Van der Sandt's resolution algorithm (Van der Sandt, 1992). Anaphoric and presuppositional elements are resolved with respect to the DRS of the dialogue so far, generating a set of potential readings. From this set those readings are chosen that obey the acceptability constraints: they should be consistent and informative.

Consider as example the dialogue in (7) with two participants A and B.

(7) A: 'Where do you want to start?'
B: 'I am leaving from Paris'.

The DRS for this mini-dialogue, from A's perspective, after hearing B's answer, is given in (8), where discourse referent y maps to participant A and x to B.

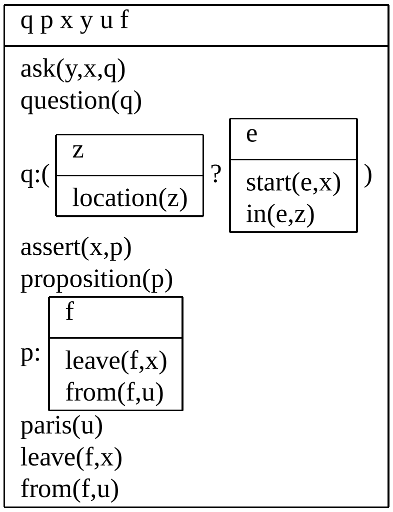
```
     ___________________________________________
    | q p x y u                                 |
    |___________________________________________|
    | ask(y,x,q)                                |
    | question(q)                               |
    | obliged-to-address(x,q)                   |
    |                          _____________    |
    |       _____________     | e           |   |
    |      | z           |    |_____________|   |
    | q:(  |_____________| ?  | start(e,x)  | ) |
    |      | location(z) |    | in(e,z)     |   |
(8) |      |_____________|    |_____________|   |
    | assert(x,p)                               |
    | proposition(p)                            |
    | paris(u)                                  |
    |     ____________                          |
    |    | f          |                         |
    | p: |____________|                         |
    |    | leave(f,x) |                         |
    |    | from(f,u)  |                         |
    |    |____________|                         |
    | under-discussion(p)                       |
    |___________________________________________|
```

The main DRS in (8) contains two conditions of the form τ:K, where the first occurrence represents the question, and the second an assertion.[1] These embedded DRSs are subordinated to the main DRS. This means that free variables appearing in them are actually bound by discourse referents occurring in the main DRS. To illustrate this idea, the proper name 'Paris' caused global accommodation of its discourse referent u (following Van der Sandt), which appears in the main DRS. This discourse referent binds the free occurrence of u in the DRS annotated by p in (8).

The main DRS represents the 'common ground', and is therefore subject to the process known as *grounding* (Traum, 1994). An optimistic instance of grounding is one where the hearer assumes that (s)he understood the utterance in the way it was intended, and as a result takes this information for granted (provided no contradictions arise). A pessimistic (or cautious) grounding scenario is one where the hearer is not sure what (s)he heard, does not accept the new information, and perhaps starts a clarification dialogue. This grounding behavior involves the speaker in a similar way.

[1]Note that temporal and modal information is left out from these examples. Events are represented by discourse referents.

Technically, the content of grounded utterances is accommodated to the main DRS, resembling an acceptance of the utterance. Coming back to our example (7), the DRS in (9) shows the situation after grounding the assertion 'I am leaving from Paris'.

(9) 

The content of ungrounded utterances stays at the subordinate level until its status is clarified. Note that the actual content of questions is never 'grounded', only the content of propositions undergoes this kind of accommodation. This model of grounding is less elaborated than the one proposed by Poesio and Traum (1998) for DRT, but it suffices for our purposes.

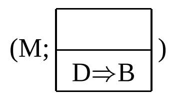
Assuming that each new utterance is constructed by assigning fresh occurrences of discourse referents, it can be shown that clashes of duplicate discourse referents will never appear. As free variables in ungrounded DRSs are already bound by discourse referents declared in the universe of the main DRS, grounding will never introduce new free occurrences. Hence this grounding mechanism is safe from a computational semantic perspective.

3.3. Interpreting Dialogue Representations

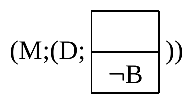
The representations for dialogues that we use form an intermediate level of representation required for its interpretation, according to the principles of DRT. Interpretation of DRSs is required in our model to implement the consistency tests as formulated for the rules for questions and answers, but also for applying acceptability constraints put forward by the resolution algorithm for pronouns and presuppositions. To perform these inferences on DRSs, we appeal to the standard translation to expressions of first-order logic (Kamp and Reyle, 1993; Blackburn et al., 1999).[2] The translation is defined by the function $(.)^{fo}$ by the clauses presented in Figure 3.

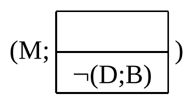
This translation is meaning preserving (a DRS is *consistent* or *inconsistent* if and only if its first-order translation has the same property) and the computational overhead involved in translation is negligible (the translation is linear in the size of the input).

The standard translation is not defined for our three extensions of the DRS language, i.e. conditions of the form τ:K, and DRSs of the form (K;K′), or (K?K′). First, for DRSs of the form (K;K′) we use merge-reduction along the lines in (Muskens, 1996) to obtain standard DRSs. Merge-reduction is the process of combining the universes of K and K′ and their conditions respectively. This can be done safely as long as the intersection of the universes of K and K′ are disjoint and no free variables in K are bound by discourse referents in K′. Second, DRS-conditions of the form τ:K represent ungrounded utterances. We do not want to include ungrounded information in any inference tasks, and therefore do not need to extend $(.)^{fo}$ for this type of condition. Third, DRSs of the form (K?K′) represent questions, and are only interpreted with respect to possible answers. This is done by transferring the insights on question partitions for first-order logic to DRSs. Recall from the previous discussion that possible answers are grounded before they are subject to the process whether they constitute a proper answer. Hence, given an answer A and main DRS M, M contains the information of A after grounding. For a question (D?B), we then arrive at the following four consistency tests:

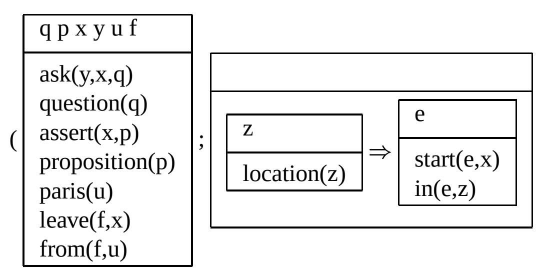
(10) 

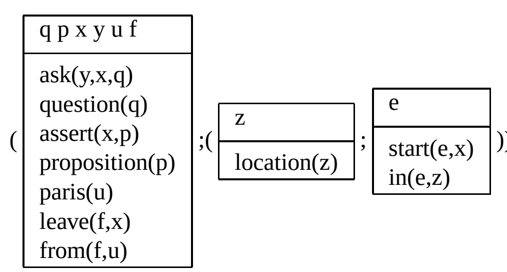
(11) (M;(D;B))

(12) 

(13) 

Applying these schemata to (9), where we want to check whether the proposition associated with discourse referent p is a proper answer to the question annotated by marker q, we get the following instantiations for (10)–(13):

(14) 

(15) 

[2]Some alternative translation are available (Van Eijck and De Vries, 1992; Muskens, 1996).

$$
\begin{aligned}
\left(\begin{array}{|l|}\hline \mathrm{x}_1\cdots\mathrm{x}_n\\\hline \gamma_1\\ \cdot\\ \cdot\\ \gamma_m\\\hline\end{array}\right)^{fo} &= \exists\mathrm{x}_1\cdots\exists\mathrm{x}_n\,((\gamma_1)^{fo}\wedge\cdots\wedge(\gamma_m)^{fo})\\
(R(\mathrm{x}_1,\ldots,\mathrm{x}_n))^{fo} &= R(\mathrm{x}_1,\ldots,\mathrm{x}_n)\\
(\mathrm{x}_1=\mathrm{x}_2)^{fo} &= \mathrm{x}_1=\mathrm{x}_2\\
(\neg K)^{fo} &= \neg(K)^{fo}\\
(K_1\vee K_2)^{fo} &= (K_1)^{fo}\vee(K_2)^{fo}\\
\left(\begin{array}{|l|}\hline \mathrm{x}_1\cdots\mathrm{x}_n\\\hline \gamma_1\\ \cdot\\ \cdot\\ \gamma_m\\\hline\end{array}\Rightarrow K\right)^{fo} &= \forall\mathrm{x}_1\cdots\forall\mathrm{x}_n(((\gamma_1)^{fo}\wedge\cdots\wedge(\gamma_m)^{fo})\rightarrow(K)^{fo})
\end{aligned}
$$

Figure 3: Translation of DRS to first-order expressions (Blackburn et al., 1999).

(16) $$\left(\begin{array}{|l|}\hline \text{q p x y u f}\\\hline \text{ask(y,x,q)}\\ \text{question(q)}\\ \text{assert(x,p)}\\ \text{proposition(p)}\\ \text{paris(u)}\\ \text{leave(f,x)}\\ \text{from(f,u)}\\\hline\end{array};\left(\begin{array}{|l|}\hline \text{z}\\\hline \text{location(z)}\\\hline\end{array};\begin{array}{|l|}\hline \\\hline \neg\begin{array}{|l|}\hline \text{e}\\\hline \text{start(e,x)}\\ \text{in(e,z)}\\\hline\end{array}\\\hline\end{array}\right)\right)$$

(17) $$\left(\begin{array}{|l|}\hline \text{q p x y u f}\\\hline \text{ask(y,x,q)}\\ \text{question(q)}\\ \text{assert(x,p)}\\ \text{proposition(p)}\\ \text{paris(u)}\\ \text{leave(f,x)}\\ \text{from(f,u)}\\\hline\end{array};\begin{array}{|l|}\hline \\\hline \neg\left(\begin{array}{|l|}\hline \text{z}\\\hline \text{location(z)}\\\hline\end{array};\begin{array}{|l|}\hline \text{e}\\\hline \text{start(e,x)}\\ \text{in(e,z)}\\\hline\end{array}\right)\\\hline\end{array}\right)$$

After applying merge-reduction to (14)–(17) these resulting DRSs can be fed to the standard translation arriving at ordinary first-order representations. For instance, the DRS (14) is reduced to (18) and translated to the first-order formula (19):

(18) $$\begin{array}{|l|}\hline \text{q p x y u f}\\\hline \text{ask(y,x,q)}\\ \text{question(q)}\\ \text{assert(x,p)}\\ \text{proposition(p)}\\ \text{paris(u)}\\ \begin{array}{|l|}\hline \\\hline \begin{array}{|l|}\hline \text{z}\\\hline \text{location(z)}\\\hline\end{array}\Rightarrow\begin{array}{|l|}\hline \text{e}\\\hline \text{start(e,x)}\\ \text{in(e,z)}\\\hline\end{array}\\\hline\end{array}\\ \text{leave(f,x)}\\ \text{from(f,u)}\\\hline\end{array}$$

(19) ∃q(∃p(∃x(∃y(∃u(∃f(ask(y,x,q) & (question(q) &
(assert(x,p) & (proposition(p) & (paris(u) &
(∀z(location(z)→ ∃e(start(e,x)&in(e,z))) &
(leave(f,x) & from(f,u)))))))))))))

If the resulting first-order formula is satisfiable, the DRS is consistent. If the negation of the resulting formula is a theorem, the DRS is marked as inconsistent. Of course, these inferences have to be supported by background knowledge. This background knowledge is a set of axioms derived from ontological information (an isa-hierarchy of concepts in the domain) and domain knowledge (such as leaving from a location implies starting in that location). Given the proper background knowledge, we find out that (14), (15), and (16) are consistent, and (17) is inconsistent. As at least one of the tests is consistent, and one of them is inconsistent, the question has been properly answered.

4. Implementation

The ideas presented above are implemented in one of the research prototypes developed in the Trindi Project (Traum et al., 1999a). MIDAS, as the system is called, covers the domain of route services, and aims to provide the user with a description of a route on the basis of a starting point, destination, and time. The implementation follows the model of dialogue moves and information state revision (Traum et al., 1999b). Utterances are treated as updates to the current state of the dialogue. This section describes the basic architecture of the system, how the information state (the DRS of the dialogue) is updated during dialogue processing, and how the inference tasks are implemented.

4.1. Basic Architecture

The system components of MIDAS are a parser, semantic construction, dialogue move engine, generator, and synthesizer. The start of the system initializes the information state, by loading a plan with actions it intends to perform. In the route service domain, these are questions that ask the user where (s)he wants to go, when (s)he wants to go, where (s)he wants to start, and so on. New utterances are analyzed by the parser, on the basis of which the semantic construction component builds a DRS. This DRS is integrated within the actual information state, by resolving pronouns, ellipsis, and presuppositions. Next, the dialogue move engine updates the current information state by applying a set of update rules to it. On the basis of this new information state, the system generates new utterances and feeds these to the synthesizer. Then the system waits for input of the user and the whole process is repeated until all intended actions are performed.

4.2. Information State Updates

The dialogue move engine from MIDAS changes the information state by either adding or removing information from it. These changes are triggered by *update rules*. Update rules consist of a name and three parts: a set of binders, a set of preconditions, and a set of effects. For any binding such that all the preconditions of an update rule holds, the dialogue move engine applies the effects to the information state. This iterative process continues until no further rules apply.

Figure 4 shows some of the update rules of MIDAS. Here we use the flat notation for DRSs, where $[U|C]$ stands for a DRS with discourse referents U and conditions C. 'K::D' binds the discourse referents in D with those in K. Further, 'K$\supseteq$ C' is short for 'DRS K contains conditions C' under the current binding, and 'K$\not\supseteq$ C' is short for 'DRS K does not contain conditions C'. The operations K+=C add conditions C to DRS K, and the operation K$-$=C remove conditions C from DRS K. Note that these operations are in the scope of the binders. The function 'consistent' returns true if its argument (a DRS) is consistent and false if it is inconsistent. $\oplus$ maps a list of boolean values to true if at least one member of the list is true, and one is false. The function 'consistent' calls the external inference component (see next section).

The rule for *optimistic grounding* has as preconditions that there is an asserted proposition under discussion, and that MIDAS is in optimistic mood. The effects add the content of the proposition to the main DRS, and cancel the status of being under discussion: after grounding the assertion, it is being dealt with. The rule for *pessimistic grounding* activates a check-question. The other rules deal with updating the intentions, and asking, addressing, or repeating questions.

The last update rule in Figure 4 deals with answer determination. One of the preconditions for this rule is the presence of a question that the user is obliged to answer. The other preconditions determine whether the main DRS contains a proper answer, by appealing to the consistency checks. If this is the case, the obligation expires. The next section describes how the consistency-checks are done in practice.

4.3. Inference

To implement inference, MIDAS makes use of current automated deduction techniques for first-order logic. As these, mostly, do not work on DRSs directly, we use the translation to first-order predicate logic with equality (Figure 3). The basic kind of inference we are interested in is checking for consistency. A formula is consistent if it is satisfiable, or if its negation is a theorem. Therefore, not only theorem provers are useful, but also model builders for detecting satisfiability.

MIDAS requires inference at two stages within processing the dialogue. First, the resolution component (dealing with anaphora and presupposition) generally produces several analyses, of which only the consistent ones are taken for further consideration (this is done by using the same techniques as in the DORIS system (Blackburn et al., 1999)). Second, the update rule for answer determination requires consistency checking of four formulas. So, generally, for both of these stages, we have many independent inference tasks for which we want an answer soon (to meet real-time constraints). Moreover, for each problem we need to find out whether it is a theorem or whether it is satisfiable, so it makes sense to call a theorem prover and a model builder in parallel.[3]

By making use of MathWeb (Franke and Kohlhase, 1999), inference problems can be solved in parallel in a competitive distributive framework, using local intra-nets or the Internet to spread the inference tasks on different machines. Currently, MathWeb runs inference services via the Internet at around 20 machines in Saarbrücken, Edinburgh, and Budapest. Of the inference arsenal offered by MathWeb, MIDAS uses the theorem provers Bliksem (De Nivelle, 1998), SPASS (Weidenbach et al., 1996), FDPLL (Baumgartner, 2000), and Otter (McCune and Padmanabhan, 1996), and the model builder MACE (McCune, 1998). It should be noted here that FDPLL and SPASS handle satisfiable problems, too.

As for answer determination, each question-answer pair results in four inference problems that are send to MathWeb. Computing times vary from 300–5000 msecs on each problem, where non-answers (300–1200 msecs) take less effort than proper answers (500–5000 msecs). For a set of four problems, MathWeb uses in average a total time of around 1.7 times the time of one problem (including Internet latency times, which are very low in general). These results clearly show the benefits of the MathWeb concept to distributed inference.

5. Conclusions and Future Work

We avoid the inherent combinatorial explosion in Groenendijk and Stokhof's theory of questions and answers by reformulating their approach in first-order logic and taking questions to denote partitions with only three different fields. We have illustrated this approach for wh-questions, and future work aims at extending the approach to deal with yes/no-questions and choice-questions.

The steps for deciding the properness of an answer are entirely based on first-order inference. This has computational advantages: many state-of-the art first-order inference services, such as theorem provers and model builders, offer high speed and coverage. The price for such a reduced analysis obviously is a loss of fine-grainedness in answer evaluation (i.e. if we wanted to we would have to find other means to detect (strongly) exhaustive answers), but so far we are not aware of important practical consequences.

[3]Incidentally, first-order logic is not decidable. In theory, this means that there is a chance that theorem provers for some input (when given enough resources) will never return with an answer. In practice, one uses time-constraints.

Name:	**Optimistic Grounding**
Binders:	M::[X,Y,P,K]
Preconditions:	M $\supseteq$ [user(X), assert(X,Y,P), proposition(P), P:K, under-discussion(P), midas(Y), optimistic(Y)]
Effects:	M+= K
	M$-$=[][under-discussion(P)]

Name:	**Pessimistic Grounding**
Binders:	M::[X,Y,P,K]
Preconditions:	M $\supseteq$ [user(X), assert(X,Y,P), proposition(P), P:K, under-discussion(P), midas(Y), pessimistic(Y)]
Effects:	M+= [Q][question(Q), obliged-to-address(X,Q), Q:K, check(Y,X,Q), under-discussion(Q)]
	M$-$= [][under-discussion(P)]

Name:	**Update Intentions**
Binders:	M::[X,Q]
Preconditions:	M $\supseteq$ [question(Q), intend-to-ask(X,Q), answered(Q)]
Effects:	M$-$=[][intend-to-ask(X,Q)]

Name:	**Ask a Question**
Binders:	M::[X,Y,Q]
Preconditions:	M $\supseteq$ [midas(X), user(Y), question(Q), intend-to-ask(X,Y,Q), unanswered(Q)]
	M $\not\supseteq$ [obliged-to-address(Y,Q)]
Effects:	M+= [][ask(X,Y,Q), obliged-to-address(Y,Q), under-discussion(Q)]

Name:	**Repeat a Question**
Binders:	M::[X,Y,Q]
Preconditions:	M $\supseteq$ [midas(X), user(Y), question(Q), obliged-to-address(Y,Q)]
	M $\not\supseteq$ [under-discussion(Q)]
Effects:	M+= [][reask(X,Y,Q), under-discussion(Q)]

Name:	**Address a Question**
Binders:	M::[X,Y,Q]
Preconditions:	M $\supseteq$ [midas(X), user(Y), ask(X,Y,Q), question(Q), under-discussion(Q)]
Effects:	M+= [][obliged-to-address(Y,Q)]
	M$-$= [][under-discussion(Q)]

Name:	**Determine Answer to Question**
Binders:	M::[Q,D,B,Y]
Preconditions:	M $\supseteq$ [question(Q), Q:D?B, user(Y), obliged-to-address(Y,Q)]
	$\oplus$ (consistent(M;[|D$\Rightarrow$B]), consistent(M;(D;B)), consistent(M;(D;[|$\neg$B])), consistent(M;[|$\neg$(D;B)]))
Effects:	M$-$= [][obliged-to-address(Y,Q)]
	M+= [][answered(Q)]

Figure 4: Example Update Rules from MIDAS. Note that 'M' stands for the main DRS.

We implemented the ideas in a prototype system under the name of MIDAS. This dialogue system uses Discourse Representation Structures to serve as intermediate structures to model the ongoing dialogue. Questions are indirectly interpreted with their (possible) answers by translating them from the intermediate structure to first-order logic. The required inference tasks are carried out by the MathWeb society of inference agents.

Ginzburg (Ginzburg, 1995) has pointed out that to decide whether a question is finally resolved for a dialogue participant, one also has to take additional criteria into account. These constitute the goals associated with a question and the questioner's view of the world. In our approach, these factors can be integrated by performing additional inference tasks or by providing further axioms when checking for consistency. In particular, the mental state of a dialogue participant can be modeled by first order formulas that are send as additional axioms to the inference machinery. According to Ginzburg, a resolving answer has to entail the goals that a dialogue participant associates with a certain question. Consider:

(20) A: 'Where do you want to start?'
B: 'Germany.'

In our approach, B would properly answer A's question, because Germany is a location, and that's what A was asking for. But if A's goal was to find out in which *city* B intends to start, then B's answer does not provide the information A was looking for, although it is still a partial answer to A's question. In our view, an entailment relation between the questioner's goals and an answer should be modeled as an *additional* constraint on determining resolving answers. We argue that it is necessary to make a first coarse classification of whether a proposition addresses a question under discussion, before finally checking for resolvedness. Such a second step, taking into account the goals of the questioner for determining answers, would obviously improve our analysis and is currently under investigation.

6. Acknowledgments

This work has been carried out as part of the Trindi (Task Oriented Instructional Dialogue) EC Project LE4-8314. We thank Alexander Koller and Kristina Striegnitz, our colleagues from the Trindi project, as well as two anonymous reviewers, for valuable comments on earlier versions of this paper.

7. References

Peter Baumgartner. 2000. FDPLL – A First-Order Davis-Putnam-Logeman-Loveland Procedure. In David McAllester, editor, *CADE-17 – The 17th International Conference on Automated Deduction*, Lecture Notes in Artificial Intelligence. Springer. To appear.

Patrick Blackburn, Johan Bos, Michael Kohlhase, and Hans de Nivelle. 1999. Inference and Computational Semantics. In Bunt and Thijsse, editors, *IWCS-3*, Tilburg, NL.

Hans De Nivelle. 1998. A resolution decision procedure for the guarded fragment. In *CADE 15*. Springer Verlag.

Andreas Franke and Michael Kohlhase. 1999. System description: Mathweb, an agent-based communication layer for distributed automated theorem proving. In *16th International Conference on Automated Deduction CADE-16*.

Jonathan Ginzburg. 1995. Resolving Questions, I. *Linguistics and Philosophy*, 18(5):459–527.

Jonathan Ginzburg. 1996. Interrogatives: Questions, Facts and Dialogue. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 385–422, Oxford, UK/Cambridge, USA. Blackwell.

Jeroen Groenendijk and Martin Stokhof. 1984. On the Semantics of Questions and the Pragmatics of Answers. In Fred Landman and Frank Veltman, editors, *Varieties of Formal Semantics*, Groningen-Amsterdam Studies in Semantics (GRASS) 3, pages 143–170, Dordrecht/Cinnaminson, U.S.A. Foris Publications.

C.L. Hamblin. 1973. Questions in Montague Grammar. *Foundations of Language*, 10:41–53.

James Higginbotham. 1996. The Semantics of Questions. In Shalom Lappin, editor, *The Handbook of Contemporary Semantic Theory*, pages 385–422, Oxford, UK/Cambridge, USA. Blackwell.

Hans Kamp and Uwe Reyle. 1993. *From Discourse to Logic*. Studies in Linguistics and Philosophy 42. Kluwer Academic Publishers, Dordrecht/Boston/London.

Lauri Karttunen. 1977. Syntax and Semantics of Questions. *Linguistics and Philosophy*, 1:3–44.

W. McCune and R. Padmanabhan. 1996. *Automated Deduction in Equational Logic and Cubic Curves*. Lecture Notes in Computer Science (AI subseries). Springer-Verlag.

W. McCune. 1998. Automatic Proofs and Counterexamples for Some Ortholattice Identities. *Information Processing Letters*, (65):285–291.

Reinhard Muskens. 1996. Combining montague semantics and discourse representation. *Linguistics and Philosophy*, 19:143–186.

Massimo Poesio and David Traum. 1998. Towards an Axiomatization of Dialogue Acts. In J. Hulstijn and A. Nijholt, editors, *Formal Semantics and Pragmatics of Dialogue, Proceedings of Twendial '98*, pages 207–221, Universiteit Twente, Enschede.

David Traum, Peter Bohlin, Johan Bos, Staffan Larsson, Ian Lewin, Colin Matheson, and David Milward. 1999a. Dialogue Dynamics and Levels of Interaction. Technical Report Deliverable D3.1p, Trindi.

David Traum, Johan Bos, Robin Cooper, Staffan Larsson, Ian Lewin, Colin Matheson, and Massimo Poesio. 1999b. A model of dialogue moves and information state revision. Technical Report Deliverable D2.1, Trindi.

David R. Traum. 1994. *A Computational Theory of Grounding in Natural Language Conversation*. Ph.D. thesis, University of Rochester, Department of Computer Science, Rochester.

Rob A. Van der Sandt. 1992. Presupposition Projection as Anaphora Resolution. *Journal of Semantics*, 9(4):332–378.

Jan Van Eijck and Fer-Jan De Vries. 1992. Dynamic interpretation and hoare deduction. *Journal of Logic, Language and Information*, 1(1):1–44.

Christoph Weidenbach, Bernd Gaede, and Georg Rock. 1996. Spass & flotter, version 0.42. In *13th International Conference on Automated Deduction, CADE-13*. Springer. To appear.

8. URLs

MathWeb: `www.mathweb.org`
MIDAS: `www.coli.uni-sb.de/˜bos/midas`
Trindi: `www.ling.gu.se/research/projects/trindi/`