A Calculus for Interrogatives Based on Their Algebraic Semantics*

Rani Nelken Nissim Francez[†]
Department of Computer Science,
Technion, Haifa 32000, Israel
{nelken|francez}cs.technion.ac.il

Abstract

We present a novel calculus for reasoning with both indicative and interrogative sentences, simultaneously modeling entailment between indicative sentences, entailment between interrogative sentences and answerhood relations. The logic is based on an interpretation of questions as entities of type **t**, the type of propositions. This is achieved by an algebraic reinterpretation of the domain of type **t** as a bilattice, rather than the standard boolean interpretation. We provide a Gentzen style axiomatization of the logic and prove its soundness and completeness with respect to the bilattice semantics. We also consider an alternative formulation using multi-valued free variable first-order tableaux, allowing for efficient algorithmic proof-search. We have implemented the tableau rules for the logic using a tableau-based theorem prover.

Keywords: bilattice, logic, natural language, tableau, theorem-prover

1 Introduction

Entailment relations obviously play a key role in the semantic analysis of natural language. The classical notion of entailment is one between propositions, corresponding to the content of indicative (declarative) sentences. Semantic entailment relations involving questions have also been discussed in both the philosophical and linguistic literature. See (Harrah, 1984) and (Groenendijk and Stokhof, 1997) (henceforth GS) for surveys. Several different semantic relations are discussed in the literature. Groenendijk and Stokhof (1997) suggest that any adequate semantic analysis of questions should account for the following relations: *Answerhood* is the relation between a question and its answers. *Interrogative entailment* is a relation between two questions. According to GS, one question entails another if whenever the answer to the first question is known, so is the answer to the second one. *Interrogative equivalence* is defined as two-way interrogative entailment. Wisniewski (1995) is concerned with the further relation of of a question *a*rising from a set of indicative sentences and the closely related notion of the presuppositions of a question.

Hintikka (1992) and Groenendijk (1999) independently propose that semantic relations involving questions can be used as governing the rules of (variants) of an idealized interrogation language game. This game is played by two players, an *interrogator* and a *witness*. The interrogator may ask non-superfluous questions, which the witness is required to answer by giving credulous pertinent answers. Hintikka (1992) requires that a question can only be addressed if its presupposition has been established. Groenendijk (1999) shows how Gricean maxims of cooperative conversation may be spelled out in terms of the semantic relations discussed above. For instance, they may be used to verify that a given answer does indeed stand in the answerhood relation to the asked question and is not entailed or contradicted by previous discourse.

*This work was carried out as part of the research project "Semantics of Natural Language Temporal Questions and Interfaces to Temporal Database Systems" sponsored by the Fund for interdisciplinary research, administered by the Israeli Academy of Science. We thank Stefan Gerberding for help with the Deep Thought theorem prover.

[†]The work of the second author was partially supported by the fund for the promotion of research in the Technion.

Groenendijk's conception of the entailment relations involving interrogatives is based on the highly influential *partition* theory of the semantics of questions developed by (Groenendijk and Stokhof, 1984). In fact, Groenendijk and Stokhof (1997) show that their own theory has a distinct advantage in comparison to other existing theories with respect to certain reasonable requirements from such semantic entailment relations. However, Groenendijk's formulation of these semantic relations is purely model-theoretical. Thus in order to check whether a particular move in a given instance of the interrogation game complies with the rules one has to reason using quite complex and highly abstract model-theoretic entities, e.g. sets of possible worlds. In contemplating a logic of questions, one would certainly hope for a syntactic, proof-theoretic formulation bundled with an effective proof-search procedure to complement the semantic model-theoretic formulation. Indeed, Hintikka (1992) sketches how a tableau-based reasoning method for first-order logic can be extended for checking presuppositions of questions. A natural question is therefore whether this approach can be extended to provide adequate proof-theoretical derivations corresponding to the required semantic relations. Unfortunately, such a proof-theoretic formulation does not seem particularly forthcoming using the GS semantics due to the inherent complexity of the model-theoretic entities involved. For instance, according to GS, a question denotes a partition on the set of possible worlds. A sentence answers the question if its intension is wholly included in one of the equivalence classes of the partition denoted by the question. It is hard to imagine a reasonable notion of derivation that is based on this notion.

In (Nelken and Francez, 1999, 2000a,b) we present an alternative extensional theory of the semantics of questions. Our theory is based on interpreting questions as entities of type **t**, the type of propositions. This is achieved by reinterpreting the domain of type **t** as an algebraic structure called a bilattice. Bilattices were pioneered by (Ginsberg, 1988), who suggested them as a generalization of the four-valued logic of (Belnap, 1977), as a uniform basis for different applications in AI. A bilattice is an algebraic structure composed of an arbitrary number of truth values and two separate partial order relations imposed on them, usually thought of as the order of truth and the order of knowledge. Fitting (1991) used bilattices as a framework for reasoning about the semantics of logic programs and Arieli and Avron (1996, 1998) have studied logical systems based on bilattices. Bilattices have been previously proposed for linguistic purposes by Muskens (1989) for handling propositional attitudes, and by Schoeter (1996) for pragmatics. Their application to questions is new. A more detailed exposition of our framework including its deeper linguistic significance, a compositional method for arriving at question meanings and various applications to pertinent issues in the semantics of natural language questions can be found in the papers cited above.

Our new algebraic interpretation paves the way towards a calculus of questions. We provide such a calculus and prove its soundness and completeness with respect to the bilattice semantics. We show an effective tableau-based proof procedure for formulae of this logic and discuss an implementation of this logic using an automated theorem prover.

The paper is structured as follows. In Section 2 we present our logic for basic questions, and its bilattice semantics. In Section 3 we present a sequent formulation of inference rules for the logic. In Section 4 we present an alternative formulation of the inference rules using tableaux. In Section 5 we extend the logic to include more complex questions, allowing coordination and quantification over questions. Finally, in Section 6, we discuss an implementation of the logic using an automated theorem prover for multi-valued first-order logic.

2 A logic for basic questions

In this section we briefly introduce a logic for basic questions, presenting its syntax and algebraic semantics.

2.1 Syntax

We use a formal language called QL, related to the one introduced by Groenendijk (1999):

Definition 1 *Let* PL *be a language of first order predicate logic with equality.* QL *is the minimal set such that:*

- *If* $\varphi \in \mathrm{PL}$ *then* $\varphi \in \mathrm{QL}$.
- *If* $\varphi \in \mathrm{PL}$ *then* $?(\varphi) \in \mathrm{QL}$.
- *If* $\varphi \in \mathrm{QL}$ *then* $?x(\varphi) \in \mathrm{QL}$.

We distinguish two *flavors* of QL formulae. Formulae that contain an occurrence of one of the interrogative operators, are called *interrogative*, and are use to represent the meanings of questions. Other formulae are called *indicative*, and are used to represent the meanings of indicative sentences.

We call '?' the *interrogative operator* and '$?x$' the *binding interrogative operator*. The free variables of a QL formula φ, written $\mathrm{FR}(\phi)$, are determined as for PL with the addition that the binding interrogative operator is true to its name, i.e. $\mathrm{FR}(?x(\varphi)) = \mathrm{FR}(\varphi) \setminus \{x\}$.

As examples of meaning representations of natural language questions, (1) represents the meaning of a yes/no question and (2) represents the meaning of a wh-question. A Montague-style compositional construction method for question meanings using a higher-order language is given in (Nelken and Francez, 2000a).

(1) $[\![$ Did Mary kiss John? $]\!] = ?(\textsc{kiss}(\textsc{mary}, \textsc{john}))$

(2) $[\![$ Which woman kissed John?$]\!] = ?x(\textsc{woman}(x) \wedge \textsc{kiss}(x, \textsc{john}))$.

QL is similar to the language defined by Groenendijk (1999), but whereas we use two different interrogative operators, Groenendijk (1999) introduces a single operator $?\bar{x}(\varphi)$ for interrogative formulae, where $\bar{x}$ is a sequence of $n \geq 0$ variables and φ is a PL formula. We allow the use of $?\bar{x}(\varphi)$ as syntactic sugar in QL for $?(\varphi)$ for $n = 0$ and for an iteration of n adjacent $?x_i$ operators otherwise.

We now turn to the semantics of QL.

2.2 Semantics - Motivation

We begin our presentation of QL semantics with an informal discussion. We assign both indicative and interrogative QL formulae truth values. Of course, we cannot assign interrogative formulae the value *true* (t) or *false* (f), since it is meaningless to say a question is either true or false. Instead, we assign interrogative formulae one of two *interrogative values*: *resolved* (r) or *unresolved* (ur). See (Ginzburg, 1995) for the significance of resolvedness to the semantics of questions. We would like the truth value of an interrogative formula $?\bar{x}(\varphi)$ to be defined as a function of the truth value of the embedded indicative formula φ. Our analysis is based on assigning indicative formulae one of three *indicative values*: *known to be false* (f), *unknown* (uk) or *known to be true* (t). In a given structure, atomic formulae are assigned one of these indicative values. Thus the structure determines for each atomic formula whether it is known to be true, known to be false or unknown. The agent of this knowledge remains implicit. The truth values of complex indicative formulae are computed according to strong Kleene three-valued logic.

Consider first a yes/no question, represented using a formula of the form $?(\varphi)$. We assign such a formula the value r if φ is either known to be true or known to be false. Otherwise, if φ is assigned uk, then we assign $?(\varphi)$ the value ur. For instance, (1) is resolved if it is known whether Mary kissed John or not. Otherwise, it is unresolved.

The meaning of a formula of the form $?x(\varphi)$ such as (2) is computed similarly to quantified sentences, as a function of the truth values of the corresponding open sentence φ for every possible value of the free variable x. If φ is indicative, then $?x(\varphi(x))$ is assigned r in case φ is assigned either f or t. For (2), this requires that for each possible domain element d, it is known whether d is a woman kissed John. This implements GS's *strong exhaustivity* requirement on answers. We do not implement the uniqueness presupposition usually attributed to which phrases with a singular noun (Ginzburg and Sag, pear).

We compute the truth value of an iteration of binding interrogative operators incrementally. For instance, for $?y?x(\varphi)$, we first compute the value of the embedded interrogative formula $?x(\varphi)$ as before, and then compute the truth value of $?y?x(\varphi)$ to be r if for every possible value of y, $?x(\varphi)$ is r, and to be ur otherwise. It is easy to verify that by this definition, $?\bar{x}(\varphi)$ is assigned r iff for every possible value of the variables in the sequence $\bar{x}$, φ is either known to be true or known to be false.

We now turn to formalize this system using a bilattice.

2.3 Semantics - formalization

In the informal presentation above, we have introduced five truth values. It is convenient to impose a bilattice structure on these values. A bilattice is a set B, which is simultaneously viewed as two complete lattices and contains a negation operation. Each lattice induces a separate partial order on B. These are usually referred to as the order of truth ($\leq_t$) and the order of knowledge ($\leq_k$):

Definition 2 (Bilattice) *A bilattice is a structure:*
$(B, \wedge_t, \vee_t, \wedge_k, \vee_k, \neg, f, t, \bot, \top)$*, such that:*

1. $(B, \wedge_t, \vee_t, f, t)$ *and* $(B, \wedge_k, \vee_k, \bot, \top)$ *are both complete lattices where* f, t *and* $\bot, \top$ *are respectively the bottom and top elements of the two lattices, and*
2. $\neg : B \to B$ *is a mapping with:*
   (a) *For any* $b \in B$, $\neg(\neg b) = b$, *and*
   (b) $\neg$ *is a lattice homomorphism from* $(B, \wedge_t, \vee_t, f, t)$ *to* $(B, \vee_t, \wedge_t, t, f)$ *and from* $(B, \wedge_k, \vee_k, \bot, \top)$ *to itself.*

Since we use a specific system of five values, we will use a particular bilattice called **FIVE**. We prefer to think of the knowledge dimension as one of *resolvedness*, changing the notation accordingly, yielding $\leq_r$, $\wedge_r$ and $\vee_r$ instead of the corresponding operators with the k subscript and ur and r, instead of $\bot$ and $\top$. **FIVE** is depicted in the double Hasse diagram of Figure 1. The two induced partial orders are reflected by the horizontal and vertical arrows.

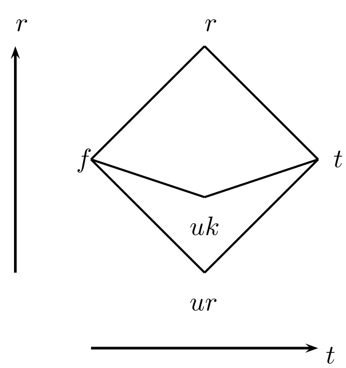

Figure 1: The bilattice **FIVE**

We model the logical operators on the three indicative values using **FIVE**'s operations on the truth dimension. Thus QL's conjunction, disjunction and negation syntactic operations will be computed using **FIVE**'s $\wedge_t$, $\vee_t$ and $\neg$ operations restricted to the indicative values. This yields the truth values corresponding to Kleene's logic. Universal and existential quantification are defined as infinitary versions of $\wedge_t$ and $\vee_t$. Thus, $\forall x(\varphi(x))$ is assigned t iff for every assignment of a domain element for x, $\varphi(x)$ is assigned t. Conversely, $\forall x(\varphi(x))$ is assigned f if the corresponding open formula is assigned f for any possible domain element. Otherwise, if there is at least one domain element for which the open formula is assigned uk, then the whole formula is also assigned uk. Similar remarks apply to existential quantification.

For indicative implication we do not use the standard definition using negation and disjunction (used in Nelken and Francez (1999)). Instead, we follow Avron (1991) in defining:

$$a \supset_t b =_{def} \begin{cases} b & \text{if } a = t \\ t & \text{otherwise} \end{cases}$$

The difference lies in the case where a is uk. Under the standard definition, the value may be either uk or t, depending on b. Under the revised definition, it is always t. This implication connective has the advantage of satisfying both modus ponens and the deduction theorem (once a notion of derivability is introduced).

For computing the truth values of interrogatives, we define the following semantic operation:

$$?a =_{def} \begin{cases} r & \text{if } a \in \{f, t, r\} \\ ur & \text{otherwise} \end{cases}$$

The interrogative operator is modeled by this unary operator. The construction is closely related to the definition of modal operators over bilattices by Ginsberg (1990). The binding interrogative operator is modeled using an infinitary version of the same operator, when operating on a set S of values, it yields r if all the values in the set are included in $\{f, t, r\}$. Otherwise, i.e. if uk or ur are in S, it yields ur:

$$?S = \begin{cases} r & \text{if for all } a \in S,\ a \in \{f, t, r\} \\ ur & \text{otherwise.} \end{cases}$$

For now, we do not make any use of the operators of the resolvedness dimension. These will come in handy in Section 5 where we augment QL, allowing logical operations to apply to interrogative formulae as well as to indicative formulae.

Arieli and Avron (1996) define a notion of consequence for bilattices. In multi-valued logics, it is common to choose a subset $\mathcal{D}$ of the values as *designated*. We choose $\mathcal{D} = \{t, r\}$ for **FIVE**. Consequence is defined as follows:

Definition 3 $\varphi \models \psi$ *iff whenever* φ *is assigned a designated value, so is* ψ.

As it turns out, the entailment relations discussed by Groenendijk and Stokhof (1997) are instances of this more general consequence relation. When both φ and ψ are indicative, consequence is regular indicative entailment. When both are interrogative, we get interrogative entailment. When φ is indicative and ψ is interrogative, we get answerhood. We extend the consequence relation to *finite* sets of QL formulae (of any flavor) $\Gamma \models \Delta$ in the usual way. $\Gamma \models \Delta$ iff whenever all the formulae in Γ are designated, at least one of the formulae in Δ is also designated. In particular, Γ or Δ may also be empty. The restriction to finite sets is not essential, but is sufficient for the study of entailment relations in a discourse of finite length. Some examples are the following:

(3) [Mary kissed John] $= \text{KISS}(\text{MARY}, \text{JOHN}) \models$
$?(\text{KISS}(\text{MARY}, \text{JOHN}))$ = [Did Mary kiss John?]

Answerhood holds since whenever the indicative is assigned t, the interrogative is assigned r. The negative answer also stands in the answerhood relation to the question:

(4) [Mary did not kiss John] $= \neg\text{KISS}(\text{MARY}, \text{JOHN}) \models$
$?(\text{KISS}(\text{MARY}, \text{JOHN}))$ = [Did Mary kiss John?]

An example of an answer to a wh-question is:

(5) [Only Mary kissed John.] $= \text{KISS}(\text{MARY}, \text{JOHN}) \wedge \forall x(\text{KISS}(x, \text{JOHN}) \supset x = \text{MARY}) \models$
$?x(\text{KISS}(x, \text{JOHN}))$ = [Who kissed John?]

This is because whenever the indicative formula is true, for each value of x we know whether x kissed John, in which case, the interrogative is resolved. If we were to omit the contribution of the restriction, only, answerhood fails to hold:

(6) $[\textsf{Mary kissed John}] = \textsc{kiss}(\textsc{mary}, \textsc{john}) \quad \not\models ?x(\textsc{kiss}(x, \textsc{john})) = [\textsf{Who kissed John?}]$

This is due to the exhaustiveness requirement. While the answer asserts that Mary kissed John, it does not provide the required information about other possible values of x, for some of which it may be unknown whether they kissed John or not.

There is an interesting connection between our interpretation and intuitionistic logic. As an illustration, consider the entailment pattern $\varphi \vee \neg\varphi \models ?\varphi$, which is valid in our system. Unlike in classical logic, the left hand side is not tautologically true. It may be *uk* if φ is assigned *uk*. However, if it is assigned t, then one of the disjuncts must be assigned t as well, in which case $?\varphi$ is resolved, independently of which disjunct was assigned t. This represents a deviation from the empirical notion of answerhood, since we standardly view such disjunctions as tautological, and hence not as informative answers. However, under our stronger interpretation of the value t as *known to be true* and f as *known to be false*, the disjunction is known to be true only if we really know which disjunct is true. This is reminiscent of the intuitionistic interpretation of disjunction, where in order to prove a disjunction, we must really know which disjunct is true. Ranta (1994) advocates a similarly intuitionistic view on questions. We discuss this issue as well as an alternative classical interpretation in (Nelken and Francez, 2000b). However, the connection with intuitionistic logic is not complete, since by the bilattice definition, for any truth value v in our system, $\neg\neg v = v$.

Turning over to interrogative entailment, it is easy to verify that the following entailments hold:

(7) $?\varphi \models ?\neg\varphi$

(8) $?x(\varphi(x)) \models ?(\exists x(\varphi(x)))$

Notice that the general consequence relation $\varphi \models \psi$ admits a fourth form, where φ is interrogative and ψ is indicative. This relation is closely connected to the relation of a question arising from an indicative sentence (Wisniewski, 1995) and the relation of an indicative sentence being a presupposition of a question (Hintikka, 1992). For instance, if one asserts in discourse the disjunction $\phi \vee \neg\phi$ (which is not tautologically true in our system), then the question $?(\varphi)$, which asks which alternative actually holds, naturally arises. This is reflected by the valid entailment pattern $?(\varphi) \models \varphi \vee \neg\varphi$, since if the left hand side is resolved, then the right hand side is known to be true, implying knowledge of the truth of one disjunct.

This constitutes the core of our model-theoretic interpretation of LQ. In the next section we present sequent rules for the logic.

3 Sequent rules

Interpreting questions using truth values lends itself to a natural proof-theoretic notion of derivation using sequent rules. We propose different sequent rules for different flavors of QL formulae. Sequent rules for indicatives are taken to be the ones given by (Avron, 1991; Arieli and Avron, 1996) for Kleene's logic. The basic axioms and logical rules for conjunction, disjunction, implication and quantification are the same as in classical logic and are given in Table 1, in which Γ and Δ are finite sets of indicative formulae.

In these rules, "a" stands for a new parameter, i.e. it does not appear in any of the other formulae in either the premises or the consequence of the sequent rule, and "t" stands for an arbitrary ground term. We use the notation $\varphi[x/\mathrm{t}]$ to denote the substitution of the term t for any free occurrence of the variable x in φ. The substitution may need to rename bound variables so that no variable free in t is captured by a quantifier in φ.

Once negation is allowed, the classical sequent rules can no longer be used. This is because whereas the sequent $\Rightarrow A, \neg A$ is classically valid, it ceases to be valid in Kleene's logic. As a result, the single compact pair of (left and right) negation introduction rules of Gentzen's classical system can no longer be used. Instead, Avron (1991) shows that a single axiom is needed together which a pair of rules (left and right) for the introduction of negation applied to each operator. In Table 2 we give the axiom and just a taste of the rules, the rules for applying negation to conjunction and

Axiom:

$$\frac{}{\Gamma, A \Rightarrow A, \Delta}\ (id)$$

Logical rules:

$$\frac{\Gamma, A, B \Rightarrow \Delta}{\Gamma, A \wedge B \Rightarrow \Delta}\ (\wedge \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma, \Rightarrow \Delta, A \wedge B}\ (\Rightarrow \wedge)$$

$$\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \vee B \Rightarrow \Delta}\ (\vee \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B}\ (\Rightarrow \vee)$$

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma, B \Rightarrow \Delta}{\Gamma, A \supset B \Rightarrow \Delta}\ (\supset\Rightarrow) \qquad \frac{\Gamma, A \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \supset B}\ (\Rightarrow\supset)$$

$$\frac{\Gamma, A[x/\mathrm{a}] \Rightarrow \Delta}{\Gamma, \exists x A \Rightarrow \Delta}\ (\exists x \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A[x/\mathrm{t}]}{\Gamma \Rightarrow \Delta, \exists x A}\ (\Rightarrow \exists x)$$

$$\frac{\Gamma, A[x/\mathrm{t}] \Rightarrow \Delta}{\Gamma, \forall x A \Rightarrow \Delta}\ (\forall x \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A[x/\mathrm{a}]}{\Gamma \Rightarrow \Delta, \forall x A}\ (\Rightarrow \forall x)$$

Table 1: Sequent rules for indicatives (without negation)

existential quantification. The rules combining negation with other operators are omitted as they are rather bulky and are of only secondary interest for us here.

Axiom:

$$\frac{}{\Gamma, A, \neg A \Rightarrow \Delta}\ (neg)$$

Logical rules:

$$\frac{\Gamma, \neg A \Rightarrow \Delta \quad \Gamma, \neg B \Rightarrow \Delta}{\Gamma, \neg(A \wedge B) \Rightarrow \Delta}\ (\neg\wedge \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, \neg A, \neg B}{\Gamma \Rightarrow \Delta, \neg(A \wedge B)}\ (\Rightarrow \neg\wedge)$$

$$\frac{\Gamma, \neg A[x/\mathrm{t}] \Rightarrow \Delta}{\Gamma, \neg\exists x A \Rightarrow \Delta}\ (\neg\exists x \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A[x/\mathrm{a}]}{\Gamma \Rightarrow \Delta, \neg\exists x A}\ (\Rightarrow \neg\exists x)$$

Table 2: Sequent rules for negating indicatives

Avron (1991) shows that the system is sound and complete relative to Kleene's logic and enjoys cut-elimination. Since the indicative fragment of QL operates according to Kleene's logic, these results transfer directly to our system.

To complete the possible derivations available in QL we add the sequent rules for interrogatives in Table 3, where now Γ and Δ are finite sets of QL formulae of any flavor. Note that the rules for the binding interrogative operator replace a premise formula of the form $?(\varphi)$ with a consequent formula of the form $?x(\varphi)$. As defined above, $?(\varphi)$ is well-formed only if φ is indicative. If φ is already interrogative, then we allow the same sequent rules without the unnecessary interrogative operator of the premise formula.

Logical rules:

$$\frac{\Gamma, A \Rightarrow \Delta \quad \Gamma, \neg A \Rightarrow \Delta}{\Gamma, ?(A) \Rightarrow \Delta} \; (? \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, A, \neg A}{\Gamma \Rightarrow \Delta, ?(A)} \; (\Rightarrow ?)$$

$$\frac{\Gamma, ?(A)[x/\mathrm{t}] \Rightarrow \Delta}{\Gamma, ?x(A) \Rightarrow \Delta} \; (?x \Rightarrow) \qquad \frac{\Gamma \Rightarrow \Delta, ?(A)[x/\mathrm{a}]}{\Gamma \Rightarrow \Delta, ?x(A)} \; (\Rightarrow ?x)$$

Structural Rules:

$$\frac{\Gamma \Rightarrow \Delta, A \quad \Gamma, A \Rightarrow \Delta}{\Gamma \Rightarrow \Delta} \; (cut)$$

Table 3: Sequent rules for interrogatives

Definition 4 *Δ is derivable from Γ, written: $\Gamma \vdash \Delta$, if $\Gamma \Rightarrow \Delta$ is provable.*

A very important property of these sequent rules is that they are *invertible* (Kleene, 1967):

Definition 5 *A rule is invertible iff the provability of the consequent sequent implies the provability of all the premise sequents.*

For the quantifier rules, invertibility means that if the consequent sequent is valid, then the premise sequent is valid for *every* term t and for *some* new parameter a.

Lemma 1 *The sequent rules for* LQ *are invertible.*

The following theorem, based on (Avron, 1991; Arieli and Avron, 1996) establishes soundness, completeness and cut-elimination for the combined system.

Theorem 1

- *Soundness and completeness: $\Gamma \models \Delta$ iff $\Gamma \vdash \Delta$.*
- *Cut-elimination: if $\Gamma \Rightarrow \Delta$ is provable, then it also has a proof with no application of the cut rule.*

Proof:

- Soundness is straightforwardly proved by induction on the structure of the proof.
- Completeness and cut-elimination are proved together. We show that if $\Gamma \Rightarrow \Delta$ has no cut-free proof, then $\Gamma \not\models \Delta$. The proof is by induction on the structure of $\Gamma \Rightarrow \Delta$.

  Base case:
  If $\Gamma \cup \Delta$ consists only of literals then the proof reduces to that of Avron (1991) for Kleene logic.

  Induction step:
  For any QL connective, assume $\phi \in \Gamma \cup \Delta$ is a formula containing that connective. Consider the corresponding (left or right) introduction rule for that connective. By the induction hypothesis either all the premise sequents of the rule have a cut-free proof or there is at

least one sequent that is refutable. If the former holds, we construct a cut-free proof of the consequent sequent from the cut-free proofs of the premise sequents using the rule itself. Otherwise, by invertibility, since one of the premise sequents is refutable, the same valuation must also refute the consequent sequent.

■

Combining completeness with invertibility leads to the following proof methodology. To prove a semantic entailment $\Gamma \models \Delta$, by completeness, it is sufficient to prove the validity of the sequent $\Gamma \Rightarrow \Delta$. To do so, choose any compound formula $\varphi \in \Gamma \cup \Delta$. Assume the main connective of φ is $*$. By invertibility, we can apply the appropriate (i.e. left for $\varphi \in \Gamma$ and right for $\varphi \in \Delta$) introduction rule backwards. The result is a simpler sequent for each premise of the rule used. Continuing backwards in this way, we can reduce the sequent to a finite set of sequents in which there are only literals (i.e. atoms or negated atoms). If each of the reduced sequents is an instance of one of the axioms, we have completed the proof.

As an illustration, consider first the simple answerhood relation in (3), repeated here:

(9) $\textsc{kiss}(\textsc{mary}, \textsc{john}) \models ?(\textsc{kiss}(\textsc{mary}, \textsc{john}))$

In order to prove it, we will prove the corresponding sequent in Figure 2 by a single application of the ($\Rightarrow$?) rule, and an application of the axiom (id).

$$\cfrac{\overline{\textsc{kiss}(\textsc{mary}, \textsc{john}) \Rightarrow \textsc{kiss}(\textsc{mary}, \textsc{john}), \neg\textsc{kiss}(\textsc{mary}, \textsc{john})}\ (id)}{\textsc{kiss}(\textsc{mary}, \textsc{john}) \Rightarrow ?(\textsc{kiss}(\textsc{mary}, \textsc{john}))}\ (\Rightarrow ?)$$

Figure 2: Proof of $\textsc{kiss}(\textsc{mary}, \textsc{john}) \Rightarrow ?(\textsc{kiss}(\textsc{mary}, \textsc{john}))$

In order to prove the answerhood relation of (5), we have to be able to reason with equality. In general, equality adds a heavy burden of proof. For the sake of simplicity, we do not introduce the required axioms here.

The above methodology is not algorithmic. The main source of difficulty is the rules that allow the instantiation of a quantifier or binding interrogative operator to any term t. The ability to reach axioms at the end of the process may depend on a proper choice of these terms. A related problem concerns the order in which the formulae are reduced. Since some of the quantifier rules instantiate a bound variable to a fresh parameter, they must sometimes be applied first. For instance, consider the proof given in Figure: 3 of the interrogative entailment example (8), repeated here:

(10) $?x(\varphi(x)) \models ?(\exists x(\varphi(x)))$

We begin (at the bottom) with the corresponding sequent. We choose to reduce the right hand side first using ($\Rightarrow$?). We now reduce the negated existential using the rule ($\Rightarrow \neg\exists$), which replaces the existential quantification with a negated atom with a new parameter, a. Next, we reduce the existential quantifier on the right hand side. Since we are allowed to use any term, we are free to use a in this reduction as well. The order of the last two steps cannot be switched, since if we were to reduce the existential first using a, we could not have used a again when reducing the negated existential. We now move to the left hand side, first reducing the binding interrogative operator to the interrogative operator, once again choosing a as our parameter, and then reducing the interrogative operator itself, reaching instances of the axiom (id) on both branches.

Thus whereas completeness guarantees that if an entailment or answerhood relation is semantically valid then the corresponding sequent is provable, the above methodology requires making informed choices. Since we are also interested in the more practical aspects of computing such deductions and in particular in mechanizing them, we consider a more effective and efficient deduction method. Haehnle and Escalada-Imaz (1997) provide a survey of deduction methods for multiple-valued logics. One of the most effective methods is the use of tableaux.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi(a) \Rightarrow \varphi(a), \neg\varphi(a)}\,(id) \qquad \dfrac{}{\neg\varphi(a) \Rightarrow \varphi(a), \neg\varphi(a)}\,(id)}{?(\varphi(a)) \Rightarrow \varphi(a), \neg\varphi(a)}\,(? \Rightarrow)}{?x(\varphi(x)) \Rightarrow \varphi(a), \neg\varphi(a)}\,(\Rightarrow \exists)}{?x(\varphi(x)) \Rightarrow \exists x(\varphi(x)), \neg\varphi(a)}\,(\Rightarrow \neg\exists)}{?x(\varphi(x)) \Rightarrow \exists x(\varphi(x)), \neg\exists x(\varphi(x))}\,(\Rightarrow \exists)}{?x(\varphi(x)) \Rightarrow ?(\exists x(\varphi(x)))}\,(\Rightarrow ?)$$

Figure 3: Proof of $?x(\varphi(x)) \Rightarrow ?(\exists x(\varphi(x)))$

4 Tableaux

Tableaux methods, originally introduced for classical logic (Smullyan, 1968; Fitting, 1996) have also been extended for multiple-valued logics (Haehnle, 1990, 1999). See (D'Agostino et al., 1999) for a recent comprehensive handbook. For classical logic there is a direct connection between invertible sequent rules and tableaux (Smullyan, 1968). Tableaux can be represented using trees of sets of *signed formulae*, which are formulae preceded by their truth value. A proof begins by assuming that the conclusion sequent $\Gamma \Rightarrow \Delta$ is not valid, i.e. all the formulae in Γ are true and all the formulae in Δ are false. This is represented by placing the set of signed formulae where all the formulae in Γ are preceded by t and all the formulae in Δ are preceded by f at the root of the tableau. By the soundness of the sequent rules, if a formula is true, then all the premises of the corresponding introduction rule must also be true. We thus expand the current branch by introducing the set of signed formulae corresponding to the premise sequents. Likewise, if a formula is false, then at least one of the premises of the introduction rule must be false as well. In this case, we introduce a separate branch for each premise. Going backwards in this fashion, one simplifies the sequents until all branches are *closed* by reaching a contradiction to an axiom. A tableau is closed iff all of its branches are closed. The tableau proof method is sound and complete.

To be sure, the same problem of properly instantiating quantifiers that we have encountered in the previous section arises for Smullyan's tableau method as well. However, there is a modification of the method using free variables (see (Letz, 1999) for a summary). In this method, free variables are allowed as placeholders. The substitution of values for the free variables is delayed until a branch can be immediately closed. In addition, instead of introducing new constants, Skolemization is necessary. Closing off branches is done by solving a *unification* problem. A branch is closed by finding two atomic formulae on it with the same predicate but different signs, and applying a unifying substitution to both.

The connection between sequent rules and tableaux is not as straightforward for multi-valued logics. In finitely-valued systems, any truth value is allowed as a sign. If a sequent $\Gamma \Rightarrow \Delta$ is not valid, there are more possibilities to consider. Each formula in Γ may be assigned one of the designated values, and each formula in Δ may be assigned one of the non-designated values. Since we must take into account all the possible values, the sequent rules of the form above become insufficient, as they only distinguish between designated and non-designated formulae, regardless of the actual truth values. However, Haehnle (1990) proves a fundamental theorem showing that for each connective θ, each assignment of a value v to a formula $\varphi = \theta(\varphi_1, \ldots, \varphi_m)$, written $< v, \varphi >$ can be reduced to an equivalent finite DNF form, corresponding to a meta-disjunction $\bigvee C_j$ of meta-conjunctions of value assignments $C_j = \bigwedge < v_{j,i}, \varphi_i >$ to the constituent formulae φ_i. Such DNF forms can be computed from the truth tables of the connectives.

There is a direct connection between the DNF forms and tableau rules. A node with the signed formula $v\varphi$, is expanded using the equivalent DNF form $\bigvee C_j$, by adding a branch for each disjunct C_j, in which all of the signed formulae of the conjunction C_j are added. A tableau branch is closed if the same formula appears on the branch preceded by two different signs, or if a formula is assigned a value that does not occur in the range of its main connective.

Haehnle (1990) suggests that efficiency can be gained through grouping several cases together, by using sets of truth values as signs. Signs are chosen from a family $\mathbf{S} \subseteq \mathcal{P}(\mathbf{FIVE})$. A sufficient condition for choosing $\mathbf{S}$ is that for any truth value $v \in \mathbf{FIVE}$, the singleton $\{v\} \in \mathbf{S}$. For concreteness, we choose the family $\mathbf{S}$ to include all the singletons as well as the set of designated values: $\{r, t\}$ the set of non-designated values: $\{uk, f, ur\}$ and the set of non-designated indicative values: $\{uk, f\}$.

Definition 6 *A signed* QL *formula is a formula of the form* $S\varphi$ *where* $S \in \mathbf{S}$ *and* $\varphi \in$ QL*.*

To illustrate this method, we give the proof of $?x(p(x)) \Rightarrow ?(\exists x(p(x)))$ using the tableau in Figure 4. We begin at the top, by placing the set of signed formulae corresponding to the sequent at the root of the tableau. The formula on the left hand side is preceded by the interrogative designated value, r, and the formula on the right hand side is preceded by the non-designated interrogative sign, ur. We begin by reducing the right formula. By the truth tables of the interrogative operator, if the formula is assigned ur, then the embedded indicative formula is assigned uk. Next, by the truth definition of existential quantification, in order for an existential formula $\exists x(\varphi(x))$ to be assigned uk, two conditions must hold. For any possible value of x, $\varphi(x)$ must be either f or uk (this is represented by the signed formula with a new free variable x_1), and in addition, there must be some Skolem constant, s_1, for which $\varphi(s_1)$ is uk. We now expand the left formula. In order for $?x(\varphi(x))$ to be assigned r, the embedded indicative formula must be either t or f. Hence we branch off. Each of the two branches may be closed by substituting s_1 for x_2. For the left branch this substitution unifies the two signed formulae $\{uk\}\varphi(s_1)$ and $\{t\}\varphi(x_2)$. Similarly for the right branch.

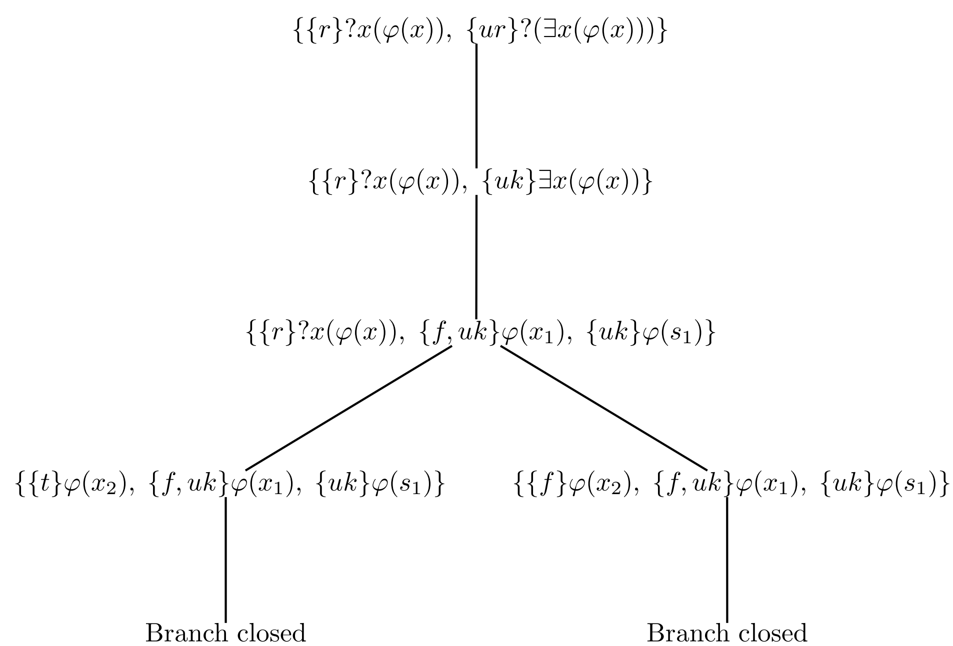

Figure 4: Tableau proof for $?xp(x) \Rightarrow ?\exists xp(x)$

The tableau method described above is general and can be applied to any finitely valued system. As a result, there is some overhead in having to introduce more rules, corresponding to each possible value. However, the technique can be fully automated. We discuss such an implementation in Section 6.

5 Extending the logic

In QL which we have considered so far, the logical operations of disjunction, conjunction, implication and quantification are restricted to indicative formulae. In (Nelken and Francez, 2000b) we show that the bilattice interpretation provides a natural interpretation for applying logical operations to questions as well as combining indicatives and interrogatives within a single combined formula. We thus extend the logic to the following:

Definition 7 EQL *is the minimal set such that:*

- *If* $\varphi \in$ PL *then* $\varphi, ?(\varphi) \in$ EQL.
- *If* φ, ψ *are* EQL *formulae, then* $\varphi \wedge \psi, \varphi \vee \psi, \varphi \supset \psi, \forall x(\varphi), \exists x(\varphi), ?x(\varphi)$ *are in* EQL

We do not allow negation to apply to interrogative formulae as natural language apparently does not allow such an operation. For EQL too we define the *interrogative* formulae to be those that contain an occurrence of one of the interrogative operators. Any other formula is called indicative.

Since both indicatives and interrogatives are interpreted as denoting type **t** entities, the result of applying operations to questions can be computed in a direct manner. Recall that we have interpreted the logical operations on indicative formulae using the bilattice operators of the truth dimension. We interpret the logical operations on the purely interrogative formulae using the operations of the resolvedness dimension. For instance, consider conjunction of a pair of questions as in (11).

(11) Who walks? And who talks?

The truth value of such a conjunction is computed by applying $\wedge_r$ to the truth values of the two conjuncts. Thus the conjunction is resolved iff both the conjunct questions are resolved. An answer to such a question must therefore answer both conjuncts. Disjunction is interpreted similarly using $\vee_r$. By contrast, the analysis of coordination in Groenendijk and Stokhof (1989) is much more complex. Conjunction is interpreted as a point-wise intersection of two partitions. However, a similar approach fails to work for disjunction since the point-wise union of two partitions fails to yield a partition. GS solve this by applying a type-shifting operation. The meaning of each disjunct question in a given possible world is shifted from a class of possible worlds to sets of sets of classes of worlds before applying set-union. It is even harder to imagine a proof-theoretical account involving such complex objects.

Nevertheless, the empirical data on coordination is not that crisp. Several authors have rejected this view of disjunction. For instance, Szabolcsi (1997), citing Hungarian evidence, views disjunction as a discourse device that cancels the first question for the second. Our analysis sides with that of Groenendijk and Stokhof (1989), while we believe that both views have empirical merit. We use a single conjunction operator that simulates $\wedge_t$ for a conjunction of indicatives and simulates $\wedge_r$ for a conjunction of interrogatives. Similarly for disjunction and implication.

In EQL quantification can also be defined over interrogatives. Such quantification is useful for the analysis of questions with quantifiers such as:

(12) Which professor recommends each candidate?

Such questions are ambiguous (Chierchia, 1995) between an *individual reading*, in which the expected answer is an individual and a *pair-list reading*, in which the expected answer is a specification for each candidate which professor recommended her. Intuitively, the two readings correspond to the two possible scope orderings of the quantifier and the interrogative operator. A notoriously difficult issue is making such quantification over questions formally respectable (Szabolcsi, 1997). The problem stems from the fact that questions are usually seen as denoting entities for which quantification is not well defined. Our interpretation, which assigns questions truth values makes such quantification particularly straightforward. The individual reading is represented in (13) and the pair-list reading in (14).

(13) $?x(\textsc{professor}(x) \wedge \forall y(\textsc{candidate}(y) \supset \textsc{recommend}(x,y)))$.

(14) $\forall y(\textsc{candidate}(y) \supset ?x(\textsc{professor}(x) \wedge \textsc{recommend}(x,y)))$.

Semantically, we define universal and existential quantification over EQL formulae as infinitary versions of the generalized conjunction and disjunction operators respectively. This interpretation yields intuitively appropriate answerhood relations. In order to answer (13) an exhaustive specification of the professors that recommended all the candidates is required.

An example entailment is the following, which asserts that for each professor there is at least one candidate that she did not recommend. Such an assertion answers the individual reading of the question by asserting that there is no such professor. In addition, it is required that the question of who is a professor is resolved. This extra requirement is needed because of the changed definition of implication, $\supset$.

(15) $?x(\textsc{professor}(x)),\ \forall x(\textsc{professor}(x) \supset \exists y(\textsc{candidate}(y) \wedge \neg\textsc{recommend}(x,y)))$
$\models ?x(\textsc{professor}(x) \wedge \forall y(\textsc{candidate}(y) \supset \textsc{recommend}(x,y)))$

In order to answer (14), for each candidate y, an exhaustive specification must be given who are all the professors x that recommended y. This is precisely the required pair-list interpretation. Here is an example answerhood relation:

(16) $?x(\textsc{professor}(x)), \forall y(\textsc{candidate}(y) \supset \forall x(\textsc{professor}(x) \supset \textsc{recommend}(x,y))) \models$
$\forall y(\textsc{candidate}(y) \supset ?x(\textsc{professor}(x) \wedge \textsc{recommend}(x,y)))$.

Once again, note that the empirical facts regarding quantification over questions are not as simple. Szabolcsi (1997) shows that the availability of pair-list readings depends not only on the quantifier but also on whether the question is embedded or not and even on the embedding verb. In (Nelken and Francez, 2000b) we discuss some insights into this problem gained from our analysis.

Note that EQL allows not only operations combining purely indicative formulae or purely interrogative formulae but also combinations. Such a mixed implication between an indicative and interrogative appears in (14). By definition, such combinations are always interrogative. We extend the generalized version of implication to also account for such combinations as follows:

$$a \supset b =_{def} \begin{cases} b & \text{if } a \in \mathcal{D} \\ t & \text{if } a \notin \mathcal{D} \text{ and } a \supset b \text{ is indicative} \\ r & \text{otherwise} \end{cases}$$

We similarly extend the other generalized operators. For instance, a mixed conjunction is interrogative. It is assigned r if both disjuncts are designated, and ur otherwise.

Since we have increased the power of the language additional sequent rules are required. The required rules are just the combined rules of Table 1 and Table 3, where we now allow them to apply to any EQL formulae. The axiom and rules for negation continue to be restricted just to indicative formulae.

It is easy to verify that soundness, completeness and cut-elimination continue to hold as before. The main observation is that the sequent rules are only concerned with whether a formula is designated, regardless of its flavor and of the particular designated value it is assigned. For instance, the actual truth value assigned to a conjunction depends on the values assigned to the conjuncts. However, a conjunction will be designated iff both conjuncts are designated, regardless of their flavor.

We illustrate EQL derivations for the two entailment relations above. Figure 5 gives a proof of (15). To save space, we have shortened predicate names to their first initial. We begin at the bottom with the consequent sequent. Applying ($\Rightarrow ?x$), we reduce the binding interrogative operator on the right hand side, replacing x by a new parameter a. We then reduce the universal quantification on the left hand side using ($\forall x \Rightarrow$), replacing x once again with a, and then reduce the binding interrogative on the left hand side, once again using a. We now reduce the implication on the left hand side. This immediately leads to an instance of (id) on the left branch. We split the proof for the right branch for reasons of space, continuing just above. We then reduce the interrogative operator on the right hand side, and are left with a sequent consisting of just indicative formulae. We omit the remainder of the proof, which is straightforward.

Figure 6 gives a proof of (16). Once again, we begin at the bottom with the consequent sequent. Applying ($\Rightarrow \forall$), we replace y by the new parameter b. Applying ($\Rightarrow \supset$) we move $c(b)$ to the left hand side. Applying ($\Rightarrow ?x$), we replace x with the new parameter a. We now reduce the universal quantifier on the left hand side, using ($\forall \Rightarrow$), substituting the term b for y. Next, we

$$\dfrac{p(a),\ \exists y(c(y) \wedge \neg r(a,y)) \Rightarrow p(a) \wedge \forall y(c(y) \supset r(a,y)),\ \neg(p(a) \wedge \forall y(c(y) \supset r(a,y)))}{p(a),\ \exists y(c(y) \wedge \neg r(a,y)) \Rightarrow ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}\ (\Rightarrow ?)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{p(a) \Rightarrow p(a), ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}\ (id) \qquad p(a),\ \exists y(c(y) \wedge \neg r(a,y)) \Rightarrow ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}{p(a),\ p(a) \supset \exists y(c(y) \wedge \neg r(a,y)) \Rightarrow ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}\ (\supset\Rightarrow)}{?x(p(x)),\ p(a) \supset \exists y(c(y) \wedge \neg r(a,y)) \Rightarrow ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}\ (?x \Rightarrow)}{?x(p(x)),\ \forall x(p(x) \supset \exists y(c(y) \wedge \neg r(x,y))) \Rightarrow ?(p(a) \wedge \forall y(c(y) \supset r(a,y)))}\ (\forall x \Rightarrow)}{?x(p(x)),\ \forall x(p(x) \supset \exists y(c(y) \wedge \neg r(x,y))) \Rightarrow ?x(p(x) \wedge \forall y(c(y) \supset r(x,y)))}\ (\Rightarrow ?x)$$

Figure 5: Proof of $?x(p(x)),\ \forall x(p(x) \supset \exists y(c(y)\ \wedge \neg r(x,y))) \Rightarrow ?x(p(x) \wedge \forall y(c(y) \supset r(x,y)))$

reduce the implication on the left hand side. The left branch is an instance of (id). For reasons of space, we split the proof, giving the right branch separately. For the right branch, we first reduce the universal quantifier followed by the binding interrogative operator on the left hand side, introducing the term a. We reduce the two remaining interrogative operators and are left with a sequent that consist purely of indicative formulae. We forego the details of the remaining proof, which is once again straightforward.

$$\frac{p(a),\ p(a) \supset r(a,b),\ c(b) \Rightarrow p(a) \land r(a,b), \neg(p(a) \land r(a,b)) \qquad \neg p(a),\ p(a) \supset r(a,b),\ c(b) \Rightarrow p(a) \land r(a,b), \neg(p(a) \land r(a,b))}{?(p(a)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow p(a) \land r(a,b), \neg(p(a) \land r(a,b))}\ (? \Rightarrow)$$

$$\frac{?(p(a)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow p(a) \land r(a,b), \neg(p(a) \land r(a,b))}{?(p(a)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (\Rightarrow ?)$$

$$\frac{?(p(a)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (?x \Rightarrow)$$

$$\frac{?x(p(x)),\ p(a) \supset r(a,b),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (\forall x \Rightarrow)$$

$$\frac{\dfrac{}{?x(p(x)),\ c(b) \Rightarrow c(b),\ ?(p(a) \land r(a,b))}\ (id) \qquad ?x(p(x)),\ \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ c(b) \supset \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (\supset \Rightarrow)$$

$$\frac{?x(p(x)),\ c(b) \supset \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ c(b) \supset \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (\forall x \Rightarrow)$$

$$\frac{?x(p(x)),\ c(b) \supset \forall x(p(x) \supset r(x,b)),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}\ (\forall x \Rightarrow)$$

$$\frac{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))),\ c(b) \Rightarrow ?(p(a) \land r(a,b))}{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))),\ c(b) \Rightarrow ?x(p(x) \land r(x,b)}\ (\Rightarrow ?x)$$

$$\frac{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))),\ c(b) \Rightarrow ?x(p(x) \land r(x,b)}{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))) \Rightarrow c(b) \supset ?x(p(x) \land r(x,b)}\ (\Rightarrow \supset)$$

$$\frac{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))) \Rightarrow c(b) \supset ?x(p(x) \land r(x,b)}{?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))) \Rightarrow \forall y(c(y) \supset ?x(p(x) \land r(x,y)))}\ (\Rightarrow \forall x)$$

Figure 6: Proof of $?x(p(x)),\ \forall y(c(y) \supset \forall x(p(x) \supset r(x,y))) \Rightarrow \forall y(c(y) \supset ?x(p(x) \land r(x,y)))$

Of course, the tableau rules which compute the actual truth values become more complex. Such rules have to take into account the actual truth values assigned to formulae. We have implemented this extended system as discussed in the next section.

6 The implementation

In order to mechanize theorem proving using the logic EQL, we have used the multiple-valued tableau-based theorem prover "Deep Thought" (DT) of Gerberding (1996). DT implements a free-variable tableau method using sets of values as signs, using several optimizations and heuristics. It is implemented in "C" on Unix. DT allows for different logical systems to be defined. We have defined rules for EQL, and tested the theorem prover with several instances of answerhood and entailment relations. DT works not with sequents but rather with just a single formula at a time. In order to prove an entailment relation of the form $\varphi \models \psi$, we prove the equivalent $\models \varphi \supset \psi$.

7 Conclusion

Both Hintikka (1992) and Groenendijk (1999) have suggested a logic of questions as a tool for analyzing an interrogative language game. In order to realize such a project, a proof-theoretic formulation is needed in addition to a well-motivated semantics. In order to realize this goal, we have suggested a logic that captures several of the entailment relations that drive such a language game. This logic is made possible by our algebraic bilattice-based interpretation. As we have seen, using this interpretation leads to rather straightforward sequent rules that allow for such derivations to be computed. In addition to accounting for basic entailment relations, it also naturally extends to complex questions involving coordination and quantification. It is only through the uniform interpretation of both indicative and interrogative sentences as entities of type **t**, that such a project becomes reasonable, as it is hard to see how previous approaches that interpret questions as more complex objects may lead to such a logic. Our logic provides an approximation of some of the entailment relations that drive conversation. Further research is needed in order to tighten the fit with actual conversation.

References

Arieli, O. and Avron, A. (1996). Reasoning with logical bilattices. *Journal of Logic, Language and Information, Kluwer academic publishers, the Netherlands*, 5(1):25–63.

Arieli, O. and Avron, A. (1998). The value of the four values. *Artificial Intelligence*, 102(1):97–141.

Avron, A. (1991). Natural 3-valued logics–characterization and proof theory. *Journal of Symbolic Logic*, 56(1):276–294.

Belnap, N. D. (1977). A useful four-valued logic. In Dunn, J. M. and Epstein, G., editors, *Modern uses of Multiple-valued logic*, pages 8–37. D. Reidel Publishing Company, Dordrecht.

Chierchia, G. (1995). *Dynamics of Meaning: Anaphora, Presupposition and the Theory of Grammar*. The University of Chicago Press, Chicago.

D'Agostino, M., Gabbay, D. M., Haehnle, R., and Posegga, J., editors (1999). *Handbook of Tableau Methods*. Kluwer, Dordrecht.

Fitting, M. (1991). Bilattices and the semantics of logic programming. *The Journal of Logic Programming*, 11:91–116.

Fitting, M. (1996). *First-Order Logic and Automated Theorem Proving*. Springer - Verlag, Berlin. First edition, 1990.

Gerberding, S. (1996). DT - an automated theorem prover for multiple-valued first order predicate logic. In *proceedings of the 26th International Symposium on Multiple-valued Logic*, pages 284–289, Santiago de Compostela, Spain.

Ginsberg, M. L. (1988). Multi-valued logics: a uniform approach to inference in artificial intelligence. *Computational Intelligence*, 4(3):265–316.

Ginsberg, M. L. (1990). Bilattices and modal operators. *Journal of Logic and Computation, Oxford University Press*, 1(1):41–69.

Ginzburg, J. (1995). Resolving questions I & II. *Linguistics and Philosophy*, 18:459–527,567–609.

Ginzburg, J. and Sag, I. (To appear). *English Interrogative Constructions*. CSLI publications.

Groenendijk, J. (1999). The Logic of Interrogation - Classical Version. In Matthews, T. and Strolvitch, D., editors, *Proceedings of the Ninth Conference on Semantics and Linguistic Theory, SALT9*, Santa Cruz. CLC Publications.

Groenendijk, J. and Stokhof, M. (1984). *The semantics of questions and the pragmatics of answers*. PhD thesis, University of Amsterdam.

Groenendijk, J. and Stokhof, M. (1989). Type-shifting rules and the semantics of interrogatives. In Chierchia, G., Partee, B. H., and Turner, R., editors, *Properties, Types and Meaning, Vol. 2: Semantic Issues*, pages 21–68. Kluwer, Dordrecht.

Groenendijk, J. and Stokhof, M. (1997). Questions. In Benthem, J. V. and Meulen, A. T., editors, *Handbook of Logic and Language*, pages 1055–1124. Elsevier Science B.V., Amsterdam.

Haehnle, R. (1990). Towards an efficient tableau proof procedure for multiple-valued logics. In *Proceedings of the Workshop on Computer Science Logic, LNCS 533*, pages 248–260, Heidelberg. Springer.

Haehnle, R. (1999). Tableaux for many-valued logics. In D'Agostino, M., Gabbay, D. M., Haehnle, R., and Posegga, J., editors, *Handbook of Tableau Methods*, pages 529–580. Kluwer, Dordrecht.

Haehnle, R. and Escalada-Imaz, G. (1997). Deduction in many-valued logics: a survey. *Mathware & Soft Computing*, iv(2):69–97.

Harrah, D. (1984). The logic of questions. In Gabbay, D. and Guenthner, D., editors, *Handbook of Philosophical Logic*, volume 2, pages 715–764. D. Reidel Publishing Company, Dordrecht.

Hintikka, J. (1992). The interrogative model of inquiry as a general theory of argumentation. *Communication and Cognition*, 25(2):221–242.

Kleene, S. C. (1967). *Mathematical Logic*. J. Wiley and sons Inc Publishing Company, New York.

Letz, R. (1999). First-order tableaux methods. In D'Agostino, M., Gabbay, D. M., Haehnle, R., and Posegga, J., editors, *Handbook of Tableau Methods*, pages 125–196. Kluwer, Dordrecht.

Muskens, R. A. (1989). Going Partial in Montague Grammar. In Bartsch, R., van Benthem, J. F. A. K., and van Emde Boas, P., editors, *Semantics and Contextual Expression*. Foris, Dordrecht.

Nelken, R. and Francez, N. (1999). The algebraic semantics of questions. In *Proceedings of MOL6, The Sixth Meeting on Mathematics of Language*, pages 167–182.

Nelken, R. and Francez, N. (2000a). The algebraic semantics of interrogative NPs. To appear in Grammars.

Nelken, R. and Francez, N. (2000b). Bilattices and the semantics of natural language questions. To appear in Linguistics and Philosophy.

Ranta, A. (1994). *Type-theoretical Grammar*. Oxford University Press.

Schoeter, A. (1996). Evidential bilattice logic and lexical inference. *Journal of Logic, Language and Information, Kluwer academic publishers, the Netherlands*, 5(1):65–105.

Smullyan, R. M. (1968). *First-Order Logic*. Springer, Berlin.

Szabolcsi, A. (1997). Quantifiers in pair-list readings. In Szabolcsi, A., editor, *Ways of Scope Taking*, pages 311–347. Kluwer, Dordrecht.

Wisniewski, A. (1995). *The Posing of Questions - Logical Foundations for Erotetic Inferences*. Kluwer Academic Publishers, Dordrecht.