

Information about a given entity:
from semantics towards automated deduction

Robert Demolombe
Institut de Recherche en Informatique de Toulouse
France
email: robert.demolombe@orange.fr

Luis Fariñas del Cerro
Institut de Recherche en Informatique de Toulouse
France
email: luis.farinass@irit.fr

Abstract

The standard method to retrieve information can be formally defined as follows. To ask a query, one gives the properties of the entities to be retrieved, and the answer is the set of all the entities that satisfy the query. Another method, is to ask the overall information about a given entity, and the answer is the corresponding information. An example of the first kind of query is: "*who are the persons who have had a car accident?*", an example of the second kind is: "*what is the overall information about a given person?*".

This latter method has deserved very few researches though it has great potential practical applications. However, it raises many non trivial issues which are investigated here. The first one is to find a precise definition of the fact that a piece of information "*is about*" a given entity. We propose a new formal definition of this notion of aboutness for query languages in First Order Logic with function symbols, and the main properties that follow from this definition are presented. The second one is to define a bridge between this abstract semantic definition and automated deduction methods based on Resolution Principle. Deduction strategies are defined in this direction and it is proved that they are complete.

1 Introduction

Since the beginning of the seventies many research works have been devoted to define theoretical foundations and to develop tools to retrieve data in the form of so called Relational Databases. In these databases the information is formally organized with a rather limited number of predicates whose extensions may be very large. This approach requires to define a priori the predicates and to store the information in this predefined form. This is not a constraint for applications in the field of management because the predicates are rarely changed. These predicates are known by users and they are used to express queries in formal languages like SQL. The retrieved information must satisfy exactly these formulas.

Now, the information which can be accessed via Internet is not formally structured in that way. Rather, in most cases, it is represented in natural languages and queries are no more expressed in a formal language but by keywords. These keywords denote either given entities or topics which are formally represented, from a logical point of view, either by constant symbols or by predicate symbols.

The work presented in this paper proposes theoretical foundations for these new kinds of queries, in particular to characterize the information provided by answers which is about a given entity. This characterization is based on previous works by philosophers and logicians [12, 16, 7, 1, 14]. From a practical point view it is not enough to have a clear definition of the meaning of the answers, we also need automated deduction methods in order to be able to generate the answers. Our contribution in this direction is based on previous works in the field of automated deduction (see [18, 2, 13, 11, 9, 10, 18]). However, our objective is only to show that the proposed theoretical definitions can lead to automated deduction methods that are more sophisticated than blind researches. We do not pretend that these methods could be directly implemented in efficient algorithms. It is a just a preliminary step and the design of implemented algorithms requires further works.

In the following section 2 we have classified the different approaches to store and retrieve information using examples. In section 3 are presented the theoretical foundations that allow to characterize the information which is about a given entity. The next section is devoted to the presentation of automated deduction methods to retrieve this information. Finally, in section 5 are presented our conclusions and some open questions that deserve further researches.

2 Selecting information about a given entity: a new perspective for information retrieval

The **standard approach** to retrieve information from a knowledge base (KB) is to request all the entities that satisfy given properties. If this approach is formalized in classical logic, KB is represented by a set of formulas (many of them represent atomic facts), and the query is represented by a formula of the form $F(x)$ ¹.

The answer which is obtained by **deduction** is the set of entities a defined by: $\{ a : \vdash KB \rightarrow F(a) \}$.

For instance, in a knowledge base about car accidents we may have the information that: if someone is driving and is drunk then he may have an accident, if someone is driving and the road is icy then he may have an accident, Smith and Dupont are driving, and Smith may have an accident. This information is formally represented by:

$$KB = \{ \forall x(drunk(x) \wedge driving(x) \rightarrow accident(x)), \forall x(icy \wedge driving(x) \rightarrow accident(x)), driving(Smith), driving(Dupont), accident(Smith) \}$$

Then, the query is represented by:

$$F(x) = driving(x) \wedge accident(x),$$

and the answer is: $\{Smith\}$.

If we want to know whether Dupont may have an accident the query is represented by: $F = accident(Dupont)$, and the answer which is obtained from a standard knowledge base is: *unknown*; because from KB we cannot infer that this fact is true, and we cannot infer that it is false.

Now, if we want to know in which circumstances Dupont may have an accident we have to ask another kind of query that gives in the answer these circumstances. In formal terms this answer is obtained reasoning by **abduction**, that is by looking for the minimal assumptions that must be added to KB to derive the fact that Dupont may have an accident.

The general formalization of these answers are defined as follows. The query is represented by a formula of the form: F , and the answer is the set of formulas H which is minimal, in some formal sense like subsumption, such that:

$$\{ H : \vdash KB \rightarrow (H \rightarrow F) \}.$$

Any formula H in this set which is added to KB allows to derive F .

For instance, for the query: $F = accident(Dupont)$,

¹In general the formula may have zero or several free variables.

the answer is: $\{drunk(Dupont), icy\}$, because, if either $drunk(Dupont)$ or icy is added to KB it is possible to derive F .

A **new approach** to retrieve information is to focus on the entities rather than on their properties. That leads to give the priority to the notion of aboutness.

According to this approach, a query is defined by the entity t we are interested in, and the answer is defined by the set of formulas F that are about this entity t and that can be derived by **deduction** from KB . That is:

$$\{ F : \vdash KB \rightarrow F \text{ and } About(F, t) \},$$

where $About(F, t)$ means that the formula F is about the entity represented by the term t .

For instance, if the query is to know everything about Dupont that can be inferred from KB , the answer is:

$$\{ drunk(Dupont) \rightarrow accident(Dupont), icy \rightarrow accident(Dupont), driving(Dupont) \}.$$

In the case of answers defined by **abduction**, the answer is defined in a similar way as in the standard approach, except that we only want to get the assumptions that are about a given entity t .

Then, the answer is formally defined by:

$$\{ H : \vdash KB \rightarrow (H \rightarrow F) \text{ and } About(H, t) \},$$

where H is minimal in the same sense as above.

For instance, if the query is: $F = accident(Dupont)$, the answer about Dupont is: $\{ drunk(Dupont) \}$. The formula icy has been removed from the answer because it is not about Dupont.

The different kinds of queries and answers can be summarized in formal terms as follows.

Standard approach.

Deduction.

$$\text{Query: } F(x). \text{ Answer: } \{ a : \vdash KB \rightarrow F(a) \}.$$

Abduction.

$$\text{Query: } F. \text{ Answer: } \{ H : \vdash KB \rightarrow (H \rightarrow F) \}.$$

New approach.

Deduction.

$$\text{Query : } t. \text{ Answer: } \{ F : \vdash KB \rightarrow F \text{ and } About(F, t) \}.$$

Abduction.

$$\text{Query: } \langle t, F \rangle. \text{ Answer: } \{ H : \vdash KB \rightarrow (H \rightarrow F) \text{ and } About(H, t) \}.$$

The new approach shows that we need a very clear and formal definition of the fact that an information is about a given entity t . That is the purpose of the next section.

3 Formal characterization of sentences that inform about a given entity

A formal definition of sentences that are not about a given entity and of sentences that are about a given entity has been defined in the semantics in [5]. The same definition is adopted in this paper, except that the language is extended to function symbols and that it is not indexed by a given entity.²

Sentences are represented by formulas of a first order language L which is defined as follows.

Definition 1. *Syntactical definition of language L .*

L is defined by the following rules.

1. *If t is a constant symbol or a variable symbol, then t is a term. If f is an n -ary function symbol and t is a n -tuple of terms, then $f(t)$ is a term.*
2. *If p is an n -ary predicate and t is a n -tuple of terms, then $p(t) \in L$.*
3. *If $F \in L$ and $G \in L$, then $(\neg F) \in L$ and $(F \vee G) \in L$.*
4. *if $F \in L$, then $(\exists x F) \in L$.*
5. *All the sentences in L are defined by rules 1, 2 and 3.*

As usual we adopt the following notations: $p \wedge q \stackrel{\text{def}}{=} \neg((\neg p) \vee (\neg q))$, $p \rightarrow q \stackrel{\text{def}}{=} (\neg p) \vee q$, $p \leftrightarrow q \stackrel{\text{def}}{=} (p \rightarrow q) \wedge (q \rightarrow p)$ and $\forall x F \stackrel{\text{def}}{=} \neg(\exists x \neg F)$.

Definition 2. *Interpretation.*

Let's consider a language L as defined in Definition 1. An interpretation M of L is a tuple $M = \langle D, i \rangle$ such that

- *D is a non empty set of individuals,*
- *i is a function that assigns*
 - *to each variable symbol and constant symbol an element of D ,*
 - *to each function symbol symbol of arity n and to each element of D^n an element of D ,*

²We have slightly changed the presentation of the definition of the variants of an interpretation. This modification makes the definition easier to understand but it does not change its meaning.

– to each predicate symbol of arity n a subset of D^n ,

In the following D will be called the domain of the interpretation, and i will be called the interpretation function, or, for short, the interpretation.

Notation: the domain of M will be denoted by D_M and the interpretation function of M will be denoted by i_M .

The satisfiability conditions of formulas are defined as usual.

Definition 3. Satisfiability conditions.

Let M be an interpretation of the language L . The fact that a formula F of L is true in M is denoted by $M \models F$, and is inductively defined as follows.

- If F is an atomic sentence of the form $p(t)$, where t is a tuple of terms, we have $M \models F$ iff $i_M(t) \in i_M(p)$.
- $M \models \neg F$ and $M \models F \vee G$ are defined from $M \models F$ and $M \models G$ as usual.
- $M \models \exists x F$ iff there exists an interpretation $M_{x/d}$ that only differs from M by the interpretation of variable symbol x , such that $i_{M_{x/d}}(x)$ is the element d of $D_{M_{x/d}}$ and $M_{x/d} \models F$.

A formula F is a valid formula iff for every interpretation M we have $M \models F$. This is denoted by $\models F$.

For technical reasons, it is easier to characterize first sentences that are not about a given entity. The characterization of sentences that are about a given entity easily follows by taking the negation of the previous one.

The intuitive idea in the semantical characterization of sentences that are not about the entity denoted by a term t is that the truth value of such sentences should remain unchanged if we change the truth value of the atomic facts that are about t .

From a technical point of view, these atomic facts are represented in the interpretation of the predicates by the tuples of D^n which have a component which is the interpretation of a term which contains t and which is not the interpretation of another term that does not contain t . The motivation of the second condition is that if an element d of D is the interpretation of a term that does not contain t the truth value of the facts represented with d must remain unchanged even if d is also the interpretation of a term that contains t .

Let's consider, for example, the predicate $french(x)$ which means that x 's nationality is french, and the function symbols $father(x)$ and $son(x, y)$

which respectively denote x 's father and x and y 's son. If we want to check whether a sentence is about *Dupont* we have to change the truth value of atomic facts represented by tuples of D^n which contain a component which is the interpretation of *Dupont*, or the interpretation of other terms like $father(Dupont)$ or $sister(Dupont)$ which contain *Dupont*. However, if some of these tuples have a component which is, for instance, the interpretation of *Smith*, the corresponding atomic facts must remain unchanged.

In a similar way, if we want to check whether a sentence is about $father(Dupont)$, that is *Dupont*'s father, we have to change the truth value of atomic facts represented by tuples of D^n which contain a component which is the interpretation of $father(Dupont)$, or the interpretation of other terms, like $father(father(Dupont))$ or $son(father(Dupont), Smith)$, which contain the sub-term $father(Dupont)$.

Notice that the truth value of the other atomic facts remain unchanged, in particular those which only contains elements which are the interpretation of *Dupont*. Indeed, in that case we are interested in *Dupont*'s father which is another person than *Dupont* himself.

Definition 4. Elements of a domain which are only about a term.

Let M be an interpretation of the language L . We adopt the following notations.

$A^t(D_M)$: set of elements in D_M which are about the term t .

$AA^t(D_M)$: set of elements of D_M which are about another term than t .

$OA^t(D_M)$: set of elements of D_M which are only about the term t .

These sets are defined as follows.

- if $d \in D_M$ and $d = i_M(t)$, then $d \in A^t(D_M)$,
- if f is an n -ary function symbol and $\delta = \langle d_1, \dots, d_n \rangle \in (D_M)^n$ and there is an element d_j in δ such that $d_j \in A^t(D_M)$ and $d' = i_M(f)(\langle d_1, \dots, d_n \rangle)$, then $d' \in A^t(D_M)$,
- if t' is a ground term such that t is not a subterm of t' and $d = i_M(t')$, then $d \in AA^t(D_M)$,
- if f is an n -ary function symbol and $\delta = \langle d_1, \dots, d_n \rangle \in (D_M)^n$ and there is no element in δ which is in $A^t(D_M)$ and there is an element d_j in δ such that $d_j \in AA^t(D_M)$ and $d' = i_M(f)(\langle d_1, \dots, d_n \rangle)$, then $d' \in AA^t(D_M)$.

An element d of D_M is in $OA^t(D_M)$ iff $d \in A^t(D_M)$ and $d \notin AA^t(D_M)$. In short terms we have $OA^t(D_M) = A^t(D_M) \setminus AA^t(D_M)$.

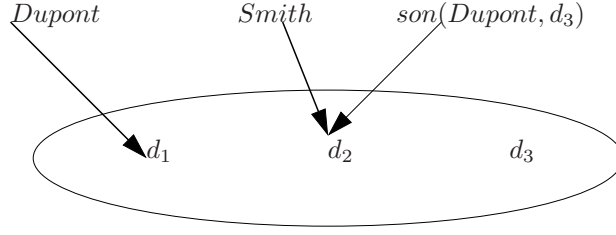


Figure 1: Terms interpretation.

This definition can be instantiated on the following example.

Let us assume that $D_M = \{d_1, d_2, d_3\}$ and $t = Dupont$.

Let us assume that the interpretation of constant symbols and function symbols are such that:

$$d_1 = i_M(Dupont), d_2 = i_M(Smith), d_2 = i_M(son)(\langle d_1, d_3 \rangle).$$

Intuitively, d_1 "is" *Dupont*, d_2 "is" *Smith* and "is" also the son of *Dupont* and of someone else d_3 which has no name in the language (see figure 1).

Then, we have: $A^t(D_M) = \{d_1, d_2\}$, $AA^t(D_M) = \{d_2\}$ and $OA^t(D_M) = \{d_1\}$. Notice that d_3 is neither about t nor about another term than t . That is why d_3 is neither in $A^t(D_M)$ nor in $AA^t(D_M)$.

We also have $D_M/OA^t(D_M) = \{d_2, d_3\}$.

From an intuitive point of view the set of elementary facts whose truth values should remain unchanged are those which are represented only by tuples of elements in $D_M/OA^t(D_M)$, that is elements that are not only about t (in particular d_5).

Definition 5. Variants of an interpretation with regard to an entity.

Let L be a language as defined in Definition 1. We call variants of M with regard to the ground term t the set M^t of interpretations M' defined from M in the following way.

- $D_{M'} = D_M$
- $i_{M'} = i_M$ for every variable symbol, constant symbol and function symbol,
- $i_{M'}$ is defined from i_M for each predicate symbol as follows. Let p be a predicate symbol of arity n . If a tuple $\langle d_1, \dots, d_n \rangle$ of $(D_M)^n$ is such that no element in this tuple is in $OA^t(D_M)$, then $\langle d_1, \dots, d_n \rangle \in i_{M'}(p)$ iff $\langle d_1, \dots, d_n \rangle \in i_M(p)$.

In the case where there is no function symbol in L and t is the constant symbol c the only element in $A^c(D_M)$ is the interpretation of c , and $AA^c(D_M)$ is the set of elements which are the interpretation of some constant symbol different of c . In that case the definition of variants is equivalent to the definition given in [5].

Notice that M belongs to M^t , and that, if M' belongs to M^t , M belongs to M'^t too.

Definition 6. Sentences that are not about an entity.

Let t be a ground term. Let F be a sentence of language L . We say that F is not about an entity named by the term t iff for every interpretation M , if $M \models F$, then for every interpretation M' in M^t we have $M' \models F$.³

The fact that F is not about entity t is denoted by $NAbout(F, t)$. In short we have:

$NAbout(F, t)$ holds iff $\forall M$ (if $M \models F$ then $(\forall M' \in M^t M' \models F)$)

We say that a formula F is about the entity t , if it is not the case that it is about the entity t . This fact is denoted by $About(F, t)$. In short terms we have:

$About(F, t)$ holds iff $\exists M(M \models F$ and $(\exists M' \in M^t M' \not\models F))$

We can check that we have $About(F, t)$ iff we do not have $NAbout(F, t)$.

It can be easily checked that the following properties hold:

$About(\text{french}(\text{Dupont}), \text{Dupont}),$
 $About(\text{french}(\text{father}(\text{Smith})), \text{Smith}),$
 $About(\text{french}(\text{sister}(\text{father}(\text{Dupont}))), \text{father}(\text{Dupont})),$
 $NAbout(\text{french}(\text{Dupont}), \text{father}(\text{Dupont}))$ and
 $NAbout(\text{french}(\text{Dupont}) \vee \neg \text{french}(\text{Dupont}), \text{Dupont}).$

Not surprisingly we have: $About(\forall x(\text{french}(x) \rightarrow \text{drunk}(x)), \text{Dupont})$. The intuitive justification is that from $\forall x(\text{french}(x) \rightarrow \text{drunk}(x))$ we can infer $\text{french}(\text{Dupont}) \rightarrow \text{drunk}(\text{Dupont})$, and we accept that this sentence is about Dupont because, if we know that $\text{french}(\text{Dupont})$, we can infer $\text{drunk}(\text{Dupont})$, and it is clear that $\text{drunk}(\text{Dupont})$ is an information about Dupont .

However, the fact that we have $About(\forall x(\text{french}(x) \rightarrow \text{drunk}(x)), \text{Smith})$ is not intuitive if we know that it is not the case that Smith is french. It seems to be more difficult to accept that, for instance, the sentence "every men is mortal" is about Toulouse if we know that Toulouse is a city.

³In the particular case where t is a constant symbol c this definition is logically equivalent to the definition given in [5] because M is in M^c . Indeed, if for every interpretation M' in M^c we have $M' \models F$, we have $M \models F$.

Nevertheless, according to our definitions we have $About(\forall x(man(x) \rightarrow mortal(x)), Toulouse)$ because in the definition of $About$ there is no information about what denotes Toulouse. If we have in mind the fact that there are people whose surname is Toulouse this result is acceptable, it just point out that we have to make explicit what we know about a Toulouse.

Indeed, if we assume that we know that $Toulouse$ is not a man, that is $\neg man(Toulouse)$, if we know that every men is mortal, that is $\forall x(man(x) \rightarrow mortal(x))$, we can infer $man(Toulouse) \rightarrow mortal(Toulouse)$, but we do not know anything new about $Toulouse$ from this sentence because $man(Toulouse) \rightarrow mortal(Toulouse)$ is a logical consequence of $\neg man(Toulouse)$.⁴

That is why we have extended the aboutness definition in order to explicitly represent what is known about an entity in the form of a theory K .

The notation $About(F, t, K)$ will be used to express the fact that the information represented by the formula F is about the entity denoted by t in a context where we know a theory represented by K . $NAbout(F, t, K)$ is used to represent the fact that it is not the case that $About(F, t, K)$.

We shall use the following notations:

M_K : set of models of the theory K .

M_K^t : set of models of the theory K which are in M^t .

Definition 7. Sentences that are not about an entity in the context of a theory.

Let t be a ground term. Let F be a sentence of language L . We say that F is not about an entity named by the term t in the context of the theory K iff for every model M of K , if $M \models F$, then for every interpretation M' in M_K^t we have $M' \models F$.

In formal terms we have:

$NAbout(F, t, K)$ holds iff $\forall M \in M_K (M \models F \Rightarrow (\forall M' \in M_K^t M' \models F))$

From the definition of the relationship between $About(F, t, K)$ and $NAbout(F, t, K)$ we have:

$About(F, t, K)$ holds iff $\exists M \in M_K (M \models F \text{ and } (\exists M' \in M_K^t M' \not\models F))$

If we apply these definitions to the previous examples we get the following results.

Let K_1 be the theory represented by the formulas: $\{\neg man(Toulouse), \neg french(Smith)\}$, we have:

$NAbout(\forall x(man(x) \rightarrow mortal(x)), Toulouse, K_1)$

$NAbout(\forall x(french(x) \rightarrow drunk(x)), Smith, K_1)$

⁴As a matter of simplification belief and knowledge are not distinguished here.

$About(\forall x(\text{french}(x) \rightarrow \text{drunk}(x)), \text{Dupont}, K_1)$

It is worth noting that even if we have $About(\text{french}(\text{Dupont}), \text{Dupont})$ we have $NAbout(\text{french}(\text{Dupont}), \text{Dupont}, \text{french}(\text{Dupont}))$, because in the context of the theory $\text{french}(\text{Dupont})$ the fact $\text{french}(\text{Dupont})$ is already known.

The most important properties about the notion of aboutness that have been proved in [5] holds in the extension of the language to symbol functions.

Theorem 1. *Let F and G be formulas of the language L . Let t be a ground term. We have the following properties.*

1. *If $\models F \leftrightarrow G$, then we have $NAbout(F, t)$ iff $NAbout(G, t)$.*
2. *We have $NAbout(F, t)$ iff $NAbout(\neg F, t)$.*
3. *If $NAbout(F, t)$ and $NAbout(G, t)$, then $NAbout(F \vee G, t)$.*
4. *If $NAbout(F, t)$ and $NAbout(G, t)$, then $NAbout(F \wedge G, t)$.*
5. *If $\models F \leftrightarrow G$, then we have $About(F, t)$ iff $About(G, t)$.*
6. *If $About(F \vee G, t)$, then $About(F, t)$ or $About(G, t)$.*
7. *If $About(F \wedge G, t)$, then $About(F, t)$ or $About(G, t)$.*
8. *If $\models F \rightarrow G$, then it is **not** the case that if we have $NAbout(F, t)$, then we have $NAbout(G, t)$.*
9. *If $\models F \rightarrow G$, then it is **not** the case that if we have $NAbout(G, t)$, then we have $NAbout(F, t)$.*
10. *If $\models F \rightarrow G$, then it is **not** the case that if we have $About(F, t)$, then we have $About(G, t)$.*
11. *If $\models F \rightarrow G$, then it is **not** the case that if we have $About(G, t)$, then we have $About(F, t)$.*

Proof. *The proofs of these properties are the same as the proofs given in [5].*

Properties 1 and 5 are rather trivial.

Properties 3, 4, 6 and 7 show how the notions $About$ and $NAbout$ "propagate" from simple formulas to complex formulas.

Property 2 is the most significant one. Of course, we have the same property for $About$. They show that the concept of aboutness is independent of the truth value of the formula F .

The "negative" properties 8, 9, 10 and 11 are not trivial. They can be intuitively understood with the following examples. For Properties 8 and 11 take: $F = q(a)$ and $G = q(a) \vee p(t)$, for 9 and 10 take: $F = p(t)$ and $G = p(t) \vee \neg p(t)$.

4 Automated deduction methods to retrieve information about a given entity

In order to design, in a further step, performant automated deduction tools to retrieve information we define automated deduction methods based on Resolution [17]. These methods require to formally represent sentences in clausal form. Clausal form restrict slightly the expressive power of the language but for a large number of real applications that is not a practical limitation. Then, in the following it will be assumed that the overall information is represented in clausal form.

In the next subsections we present first a classical abduction method, in the second subsection a deduction method to retrieve information about a given entity is defined, and in the third subsection we extend the classical abduction method to retrieve information about a given entity.

4.1 Classical abduction

Many strategies have been defined to compute answers in the standard approach when the answers are obtained by deduction.

When the answers are obtained by abduction the number of strategies is rather limited. There is, for instance, the SOL-resolution defined by Inoue in [8, 9, 10], or the method defined by Cialdea and Pirri in [15] and the l-inference defined by Demolombe and Fariñas del Cerro in [4].

For doing abduction with the l-inference the basic idea is to "define" a literal L that characterizes the formula F for which we are looking for additional assumptions in the context of a given theory. If X is the tuple of the free variables in F , L is a literal of the form $L = l(X)$ such that the predicate symbol l does not occur in the set of clauses that represents the theory and such that we have $\vdash \forall X(l(X) \leftrightarrow F(X))$. Then, the clauses obtained from the formula $\forall X(l(X) \leftrightarrow F(X))$ are added to the theory and the abduction problem is to find the consequences of the theory of the form $L' \vee C$ where L' is an instance of L and which are minimal for the subsumption. The negation of the clauses like C give the assumptions we are looking for.

For instance, if we refer to the example at the beginning of section 1, if

one is interested in the assumptions that we have to add to the theory KB to know people who have an accident and who are not drunk, the following "definition" is added to KB : $\forall x(l(x) \leftrightarrow accident(x) \wedge \neg drunk(x))$.

Then, to apply the l-inference defined in Definition 10 this formula is transformed into its clausal form and we can derive, for instance, the clauses: (1) $\neg icy \vee \neg driving(x) \vee drunk(x) \vee l(x)$ and (2) $\neg icy \vee drunk(Smith) \vee l(Smith)$. These clauses respectively are of the form $C_1 \vee l(x)$ and $C_2 \vee l(Smith)$, and we have $\neg C_1 = icy \wedge driving(x) \wedge \neg drunk(x)$ and $\neg C_2 = icy \wedge \neg drunk(Smith)$.

The intuitive meaning of $\neg C_2$, for instance, is that it is sufficient to add the assumption that it is icy and Smith is not drunk to infer from KB that Smith has an accident.

We briefly resume below what is the l-inference.

Definition 8. Clause.

A clause is a first order formula of the form: $L_1 \vee L_2 \vee \dots \vee L_n$, where each L_i is a literal. A literal is an atomic formula or the negation of an atomic formula of the language L .

The free variables of a clause are implicitly universally quantified.

Definition 9. l-Clause.

A clause C is a l-clause iff there is an atomic formula in C with l as predicate symbol.

Definition 10. l-Inference.

A resolvent C from C_1 and C_2 by Resolution Principle is obtained by a l-inference iff C is a l-clause.

If C is a l-clause one of the parent clauses C_1 or C_2 is a l-clause.

Definition 11. l-Deduction. *Let S be a set of clauses. A l-deduction of C_n from S is a finite sequence $C_0 \dots C_n$ of clauses such that : each C_i is either a clause in S or there are C_{i_1} and C_{i_2} in the l-deduction such that $i_1 < i$, $i_2 < i$ and C_i is the l-resolvent of C_{i_1} and C_{i_2} .*

Definition 12. R-Deduction. *A R-deduction of C_n from S is a finite sequence $C_0 \dots C_n$ of clauses such that : each C_i is either in S or there are C_{i_1} and C_{i_2} in the R-deduction such that $i_1 < i$, $i_2 < i$ and C_i is the resolvent by Resolution Principle of C_{i_1} and C_{i_2} .*

Theorem 2. *Let S be a set of clauses, if C is a clause derivable from S , there is a clause C' , subsuming C , such that C' is derivable from S by a R-deduction.*

The Theorem 2 has been proved by Lee in [13].

let S_1 be, for example, the following set of clauses:

$$S_1 = \{(1)\neg p(x, y) \vee q(y) \vee l(x), (2)\neg p_1(x, z) \vee \neg p_2(z, y) \vee p(x, y), (3)\neg p_3(x, y) \vee p_1(x, y), (4)\neg p_1(x, y) \vee r(x) \vee s(y), (5)p_1(b, c), (6)p_2(c, a), (7)p_1(b, a), (8)p_2(a, d), (9)p_3(d, c)\}$$

From S_1 we can draw the following l-deduction:

- (L1) $\neg p_1(x, z) \vee \neg p_2(z, y) \vee q(y) \vee l(x)$ [from (1) and (2)]
- (L2) $\neg p_3(x, z) \vee \neg p_2(z, y) \vee q(y) \vee l(x)$ [from (L1) and (3)]
- (L3) $\neg p_3(x, c) \vee q(a) \vee l(c)$ [from (L2) and (6)]

We can also draw from S_1 the following R-deduction:

- (R1) $\neg p_3(x, z) \vee \neg p_2(z, y) \vee p(x, y)$ [from (2) and (3)]
- (R2) $\neg p_3(x, c) \vee p(x, a)$ [from (R1) and (6)]
- (R3) $\neg p_3(x, c) \vee q(a) \vee l(c)$ [from (R2) and (1)]

This R-deduction derives the same consequence (R3) as the previous l-deduction, and, since this R-deduction is not an l-deduction, a strategy which draws only l-deductions does not explore this R-deduction. This strategy also discard all the R-deductions which use clauses (2) to (9). In the practical applications where the number of ground atoms is very large, the number of R-deductions which are discarded is extremely large.

Theorem 3. *Let S be a set of clauses and l a given predicate. If there is a R-deduction of the l -clause C , then there is a l-deduction of C .*

The proof of Theorem 3 can be found in [4]⁵.

The practical interest of this result is that we can cut the deductions that contain clauses which are not l -clauses. That significantly reduces the search space.

4.2 Deduction restricted to an entity

We present now an automated deduction method to derive answers which are about a given entity denoted by a term t .

In a first step the notion of formula about an entity t is applied to the context of formulas represented in a clausal form. That leads to the syntactical definition of t -clauses and we prove that this definition characterizes all the clauses that are about t .

In a second step a linear strategy is proposed to derive the t -clauses. An interesting property of this strategy is that only t -clauses are generated and it is complete up to subsumsion.

⁵The Theorem given in [4] is more general in the sense that the l-inference is defined as an hyperresolution. That gives a deduction methods which is more efficient.

4.2.1 Syntactical characterization of clauses about t

Definition 13. t -Clause.

Let t be a ground term. A t -clause is a clause which is not a tautology such that there is a literal in the clause and a term t' in this literal such that there is a sub term of t' which is either t or a variable.

For instance, if t is the constant a , $p(x) \vee p(b)$ and $p(x) \vee p(a) \vee q(b)$ are t -clauses, while $p(a) \vee \neg p(a)$ is not a t -clause because it is a tautology.

Definition 14. Minimal clause. A clause C is minimal iff there is no sub clauses C' and C'' such that $C = C' \vee C''$ and there exists a substitution σ such that the set of literals in $C'.\sigma$ is included into the set of literals in C'' .

For instance, the clause $p(x) \vee q(b)$ is minimal, while the clauses $p(x) \vee p(b)$ and $p(x) \vee p(a) \vee q(b)$ are not minimal.

For the characterization of the clauses which are about t we have to define t -clauses which are minimal in a particular sense. They must be minimal in the sense that the set of literals C'_1 which are about t does not imply the set of other literals C''_1 , but there may be another set of literals C'_2 in the same clause which implies its complement C''_2 .

For instance, the clause $p(x) \vee p(a) \vee q(b)$ is minimal in this weaker sense because the set of literals about t : $p(x) \vee p(a)$, does not implies $q(b)$, and the clause $p(x) \vee p(b)$ is not minimal in this weaker sense because $p(x)$ implies $p(b)$.

The intuitive motivation for this weaker definition of minimality is that, even if $p(x) \vee p(a) \vee q(b)$ is not minimal in the sense of Definition 14 it is about t . Indeed, we can find a model M where $p(x) \vee p(a)$ is true and $q(b)$ is false and it is possible to find a variant M' in M^a where $p(x) \vee p(a)$ is false. Then, in this variant the global clause is false because the truth value of $q(b)$ does not change in the variants in M^a . That means that $p(x) \vee p(a) \vee q(b)$ is about a .

At the opposite, if the clause $p(x) \vee p(b)$, which is not minimal in the weaker sense, is true in some model M , necessarily $\forall xp(x)$ and $p(b)$ are true in M (if $p(b)$ is false, $\forall xp(x)$ cannot be true, and if $\forall xp(x)$ is true, $p(b)$ is also true). Since the truth value of $p(b)$ does not change in the variants of M in M^a , the clause $p(x) \vee p(b)$ remains true in these variants. That means that $p(x) \vee p(b)$ is not about a .

In the following, as a matter of simplification, we call minimal t -clauses the t -clauses which are minimal in this weaker sense.

Definition 15. Minimal t -Clause.

Let t be a ground term. Let C be a t -clause of the form $C = C' \vee C''$ such that C' contains all the literals whose a term has a sub term which is either t or a variable. C is a minimal t -clause iff we have $\not\vdash C' \rightarrow C''$. In the following we adopt the notations:

$$C' = L'_1 \vee \dots \vee L'_n, \text{ where } n > 0,$$

$$C'' = L''_1 \vee \dots \vee L''_m,$$

$$C = C' \vee C''.$$

Definition 16. Separated interpretation.

Let t be a ground term. Let M be an interpretation as defined in Definition 2. M is a separated interpretation iff $A^t(D_M) \cap AA^t(D_M) = \emptyset$.

Lemma 1. If M is an interpretation which satisfies a set of literals, then there exists a separated interpretation M_s which satisfies this set of literals.

Proof. The separated interpretation M_s is defined from M in such a way that the only differences are defined as follows.

Let τ be a tuple of arguments of some literal L in this set. Let t' be a component of τ .

If $i_M(t') \in A^t(D_M) \cap AA^t(D_M)$, then a new element d is added to D_{M_s} and i_{M_s} is defined such that $i_{M_s}(t') = d$, else $i_{M_s}(t') = i_M(t')$.

Then, i_{M_s} is defined such that: $i_{M_s}(\tau) \in i_{M_s}(L)$ iff $i_M(\tau) \in i_M(L)$.

The truth values of the literals is the same in M_s than in M and we have $A^t(D_{M_s}) \cap AA^t(D_{M_s}) = \emptyset$.

Theorem 4. Let C be a clause which is not a tautology. We have $\text{About}(C, t)$ iff C is a minimal t -clause.

Proof. We first prove that if C is a minimal t -clause, then we have $\text{About}(C, t)$.

Since C is minimal there exists an interpretation M such that we have $M \models C'$ and $M \models \neg C''$. Else we would have $\vdash C' \rightarrow C''$, which contradicts the fact that C is minimal. From Lemma 1 it can be assumed that M is a separated interpretation.

The interpretation M' is defined from M in such a way that the only differences are defined as follows.

Let σ_t be an interpretation of the variables in C' which assigns the element $d = i_M(t)$ to all the variables.

If $i_M(L'_i, \sigma_t)$ is true, then $i_{M'}(L'_i, \sigma_t)$ is false.

Then, we have $M' \models \neg(C'.\sigma_t)$ and $M' \models \neg C''$, since the truth value of each L''_j is the same in M' than in M . Therefore we have $M' \models \neg C$.

For the literals of the type L'_i either there exists an argument t' which has a sub term which is a variable and $i_M(t'.\sigma_t)$ is in $OA^t(D_M)$, or t is a sub term of t' and $i_M(t')$ is in $OA^t(D_M)$ (because M is a separated interpretation).

Since the only tuples that have changed the truth value of some atomic fact are those which have at least one component in $OA^t(D_M)$, M' is in M^t .

Now, we prove that if we have $About(C, t)$, then C is a minimal t -clause.

The proof is by refutation. It is assumed that we have $About(C, t)$ and C is not a minimal t -clause.

If C is a t -clause which is not minimal, for every interpretation M if we have $M \models C$, we also have $M \models C''$ because if C is not minimal we have $\vdash C \leftrightarrow C''$. Since there is no occurrence of t nor variable in C'' , from the definition of M^t for every M' in M^t we have $M' \models C''$, and then we have $M' \models C$. Therefore we have $NAbout(C, t)$, which contradicts the assumption.

If C is not a t -clause we have $C = C''$ and it can be proved like above that we have $NAbout(C, t)$, which also contradicts the assumption.

The Theorem 4 shows that a complete characterization of clauses such that $About(C, t)$ is given by the minimal t -clauses. If minimal t -clauses would be defined as clauses which are t -clauses and which are minimal (in the sense of Definition 14) we would have an incomplete characterization. For example, the clause $p(x) \vee p(a) \vee q(b)$ is a t -clause for $t = a$, while it is not minimal in the sense of Definition 14.

4.2.2 Derivation of clauses about t

Definition 17. Minimal R-Deduction. A minimal R -deduction of C from S is an R -deduction such that all the clauses in the deduction are minimal clauses.

Lemma 2. If $Res(S)$ is the set of clauses obtained by an R -deduction from S , there exists a set of clauses S' logically equivalent to S such that for every clause C in $Res(S)$ there exists a minimal deduction of C' from $S' = S \cup S_\tau$, where S_τ is the set of clauses of the form: $\neg L \vee L$ such that the literal L occurs in some clause in S , such that C' subsumes C .

Proof. The proof is by induction on the length n of the proofs of clauses C in $Res(S)$.

Induction hypothesis. If C is in $Res(S)$ and the length of the proof of C is less or equal to n , there exists $S' = S \cup S_\tau$ logically equivalent to S such that there exists a minimal R -deduction of C' from S' and C' subsumes C .

Let C be a clause in $Res(C)$ such that the length of the proof δ of C is $n + 1$.

If δ is a minimal R -deduction, then $C' = C$ and $S' = S$.

If δ is not a minimal R-deduction, let C_1 and C_2 be the two clauses in δ such that C is the resolvent of C_1 and C_2 .

Since the length δ_1 (resp. δ_2) of the proof of C_1 (resp. of C_2) is less or equal to n , by induction hypothesis there exists S_1 (resp. S_2) logically equivalent to S such that there exists a minimal R-deduction δ'_1 (resp. δ'_2) of D_1 from S_1 (resp. of D_2 from S_2) and D_1 subsumes C_1 (resp. D_2 subsumes C_2).

Let D be the resolvent by Resolution of D_1 and D_2 .

If D is a minimal clause, we have with δ_1 and δ_2 a minimal R-deduction of D from $S_1 \cup S_2$ which is equivalent to S and D subsumes C .

If D is not minimal, D is of the form $D = C' \vee C''$ and there exists a substitution σ such that the set $C'.\sigma$ is included in C'' . The clause C' is of the form $C' = L_1 \vee \dots \vee L_n$. Then, for each L_i there is a literal L'_i in C'' such that $L'_i = L_i.\sigma$. Moreover, L_i is an instance of some literal that occurs in some clause in S . Then, there is a clause in S_τ which has an instance of the form $\tau_i = \neg L_i.\sigma \vee L'_i$ for i in $[1, n]$.⁶ Since all the clauses in S_τ are tautologies the set $S_1 \cup S_2 \cup S_\tau$ is logically equivalent to S .

For each L_i there is literal L''_i either in D_1 or D_2 such that L_i is an instance of L''_i . This literal can be "replaced" by L'_i in D_1 or D_2 , or their consequences generated by the same method, by a resolution with the clause $\tau_i = \neg L_i.\sigma \vee L'_i$. Since, D_1 and D_2 are minimal, these consequences are also minimal.

At the end of these resolutions we get the clauses E_1 from D_1 , and E_2 from D_2 , that contain no more occurrences of literals of the type of L''_i . Then, E_1 and E_2 can be resolved on the same literals as the literals used for the resolution of D_1 and D_2 , and the resolvent is C'' , which is minimal, subsumes C .

The following example can help to have a more concrete view of the proof.

Let us consider the following clauses : $D_1 = l(a) \vee p(x, y) \vee q(u) \vee r(c)$ and $D_2 = \neg l(z) \vee p(x, z) \vee r(v) \vee q(b) \vee s(d)$.

We have: $C = p(x, y) \vee q(u) \vee r(v) \vee p(x, a) \vee q(b) \vee r(c) \vee s(d)$.

If $\sigma = \{y/a, u/b, v/c\}$, $C' = p(x, y) \vee q(u) \vee r(v)$ and $C'' = p(x, a) \vee q(b) \vee r(c) \vee s(d)$, we have $C'.\sigma = p(x, a) \vee q(b) \vee r(c)$ which is included in C'' .

Then, we have: $\tau_1 = \neg p(x, a) \vee p(x, a)$, $\tau_2 = \neg q(b) \vee q(b)$ and $\tau_3 = \neg r(c) \vee r(c)$.

We define the following deductions.

$$R(D_1, \tau_1) = E_1 = l(a) \vee p(x, a) \vee q(u) \vee r(c)$$

⁶If L_i is a negative literal the double negation is removed.

$$\begin{aligned}
R(E_1, \tau_2) &= F_1 = l(a) \vee p(x, a) \vee q(b) \vee r(c) \\
R(D_2, \tau_3) &= E_2 = \neg l(z) \vee p(x, z) \vee r(c) \vee q(b) \vee s(d) \\
R(F_1, E_2) &= p(x, a) \vee q(b) \vee r(c) \vee s(d)
\end{aligned}$$

Finally, we have: $R(F_1, E_2) = C''$.

From an intuitive point of view, the role of the tautologies like τ_1 is to derive instances of a given clause like D_1 . Indeed, with the R-inference we cannot directly derive E_1 from D_1 . Let's assume, for example, that a set of clause contains only the clause: $q(x) \vee q(b)$. With the R-inference we cannot infer $q(b)$. However, if we add the tautology $\neg q(b) \vee q(b)$, we can.

The technical interest of Lemma 2 in the following proofs is that to prove that R-deductions can be transformed into minimal t -deductions we only have to consider R-deductions that are minimal.

Definition 18. t -Inference. *An inference of C from C_1 and C_2 by Resolution Principle is a t -inference iff the resolvent C of C_1 and C_2 is a t -clause.*

Definition 19. Minimal t -Inference. *Let t be a ground term. An inference of C from C_1 and C_2 by Resolution Principle is a minimal t -inference iff the resolvent C of C_1 and C_2 is a minimal t -clause.*

Definition 20. t -Deduction. *A t -deduction of C_n from S is a finite sequence of clauses $C_0 \dots C_n$ such that each C_i is either a clause in S or there are C_{i_1} and C_{i_2} in the t -deduction, with $i_1 < i$ and $i_2 < i$, such that C_i is obtained by a t -inference from C_{i_1} and C_{i_2} .*

C_0 is called the top clause.

Definition 21. Minimal t -Deduction. *A minimal t -deduction of C_n from S is a finite sequence of clauses $C_0 \dots C_n$ such that each C_i is either a clause in S or there are C_{i_1} and C_{i_2} in the minimal t -deduction, with $i_1 < i$ and $i_2 < i$, such that C_i is obtained by a minimal t -inference from C_{i_1} and C_{i_2} .*

C_0 is called the top clause.

To prove Lemma 4 we shall use Lemma 3 and to prove Lemma 5 we shall use Lemma 4. Finally, to prove Theorem 5 we shall use both Lemma 2 and Lemma 5.

Lemma 3. *If C is obtained by a t -inference from C_1 and C_2 , and C_2 is a t -clause, and C_1 is the resolvent by Resolution Principle of the two clauses E_1 and E_2 , then there exists a t -inference of E_2 and C_2 whose resolvent is F , and there exists a t -inference of E_1 and F whose resolvent is C .*

If C_1, C_2, E_1, E_2 and C are minimal clauses, then F is a minimal clause.

Proof. Let L_2 be the literal in C_2 which is resolved with a literal in C_1 . Without loss of generality we can assume that this literal in C_1 is an instance of a literal in E_2 which is called L'_1 .

Let M_2 be the literal in E_2 which is resolved with some literal M_1 in E_1 . Then, E_1 and E_2 have the following form:

$$\begin{aligned} E_1 &= M_1 \vee e_1 \\ E_2 &= M_2 \vee L'_1 \vee e_2 \end{aligned}$$

Let σ_1 be the mgu of M_1 and M_2 , and σ_2 be the mgu of $L'_1\sigma_1$ and L_2 . then the clauses C_1 , C_2 and C have the form:

$$\begin{aligned} C_1 &= L'_1\sigma_1 \vee e_1\sigma_1 \vee e_2\sigma_1 \\ C_2 &= L_2 \vee c_2 \\ C &= e_1\sigma_1\sigma_2 \vee e_2\sigma_1\sigma_2 \vee c_2\sigma_2 \end{aligned}$$

Since $L'_1\sigma_1$ and L_2 can be unified there exists a mgu σ'_1 of L'_1 and L_2 . Let F be the resolvent by Resolution Principle of E_2 and C_2 . F has the form:

$$F = M_2\sigma'_1 \vee e_2\sigma'_1 \vee c_2\sigma'_1$$

The literals M_1 and $M_2\sigma'_1$ can be unified by the mgu σ'_2 . Let C' be the resolvent by Resolution Principle of E_1 and F . Then, C' has the form:

$$C' = e_1\sigma'_2 \vee e_2\sigma'_1\sigma'_2 \vee c_2\sigma'_1\sigma'_2$$

It can be proved that F and C' are t -clauses. Then, they are obtained by a t -inference.

It can also be proved that the clause C' is the same clause as C .

We can easily check that if C_1 , C_2 , E_1 , E_2 and C are minimal clauses, then F is a minimal clause.

Lemma 4. *If there is a minimal R-deduction of C from S and the minimal clause C_2 such that C is a minimal t -clause and C is the resolvent of the clauses C_1 and C_2 , then there exists a minimal t -deduction of C from S and C_2 , such that C_2 is the top clause.*

Proof. *The proof is by induction on the length n of the R-deduction of C_1 .*

Induction hypothesis. If there is a minimal R-deduction of C from S and the minimal clause C_2 such that: C is a minimal t -clause, C is the resolvent of the clauses C_1 and C_2 and the length of the R-deduction of C_1 is $\leq n$, then there exists a minimal t -deduction of C from S and C_2 such that C_2 is the top clause.

For $n = 0$ the induction hypothesis is true (trivial).

Assumption: there is a minimal R-deduction of C from S and the minimal clause C_2 such that: C is a minimal t -clause, C is the resolvent of the clauses C_1 and C_2 and the length of the minimal R-deduction of C_1 is $n + 1$.

Let E_1 and E_2 be the two clauses whose inference by Resolution principle is C_1 . The length of their minimal R-deductions is $\leq n$.

From Lemma 3 the deduction of C_1 from E_1 and E_2 , and of C from C_2 can be transformed into a deduction of F from E_2 and C_2 and of C from E_1 and F , where C and F are obtained by minimal t -inference.

From the induction hypothesis, there exists a minimal t -deduction δ_1 of F from S and C_2 . From the induction hypothesis we can also infer that there exists a minimal t -inference δ_2 of C from S and F .

Therefore the sequence $\delta_1\delta_2$ is a minimal t -deduction of C from S and C_2 .

Lemma 5. *If there is a minimal R -deduction of C from S such that C is a minimal t -clause, then there exists a minimal t -deduction of C from S .*

Proof. *The proof is by induction on the length n of the minimal R -deduction of C .*

Induction hypothesis. If there is a minimal R -deduction of length $\leq n$ of C from S such that C is a minimal t -clause, then there exists a minimal t -deduction of C from S .

Assumption. There is a minimal R -deduction of length $n + 1$ of C from S such that C is a minimal t -clause.

Let C_1 and C_2 be the clauses whose inference by Resolution principle is C . Therefore either C_1 or C_2 is a minimal t -clause. Let C_2 be that minimal t -clause. Since the length of the minimal R -deduction of C_2 is $\leq n$, by induction hypothesis there exists a minimal t -deduction δ_1 of C_2 from S .

From Lemma 4 there exists a minimal t -deduction of C δ_2 from S and C_2 .

Therefore the sequence $\delta_1\delta_2$ is a minimal t -deduction of C from S .

Theorem 5. *If there is an R -deduction of C from S such that C is a minimal t -clause, then there exists a minimal t -deduction of C' , such that C' subsumes C , from $S \cup S_\tau$, where S_τ is the set of clauses of the form: $\neg L \vee L$ such that the literal L occurs in some clause in S .*

Proof. *From Lemma 2, if there is an R -deduction of C from S , there is a minimal R -deduction of C' , such that C' subsumes C , from $S \cup S_\tau$, and from Lemma 5 there is a minimal t -deduction of C' from $S \cup S_\tau$.*

Notice that, from an implementation point of view, there is no need to explicitly represent the set S_τ .

If the term t is the constant a , we can draw from the example S_1 the following minimal t -deduction:

- (T1) $\neg p_1(x, c) \vee p(x, a)$ [from (2) and (6)]
- (T2) $\neg p_3(x, c) \vee p(x, a)$ [from (T1) and (3)]
- (T3) $p(d, a)$ [from (T2) and (9)]

All the R-deductions that derive ground clauses which does not mention the constant a are discarded by a strategy which only generates t-deductions. The number of discarded R-deductions is of the same order of magnitude as the number of ground clauses without a .

4.3 Abduction restricted to an entity

To only retrieve answers that are assumptions about an entity we have defined an abduction methods which is more specific than the SOL-deduction⁷ or the l-deduction.

Definition 22. *lt-Clause.* *A clause is a lt-clause iff it is both a l-clause and a t-clause.*

Definition 23. *lt-Inference.* *A lt-inference is an inference which is both a l-inference and a t-inference.*

It is worth noting that there are lt-clauses that are R-deductibles from a given set of clause S , such that it does not exist a R-deduction in which each resolvent is a lt-clause. Let's consider, for instance, the set of clauses: $S = \{\neg q \vee r, q \vee l, \neg r \vee p(t)\}$ and the lt-clause: $l \vee p(t)$.

Definition 24. *lt-Deduction.* *A lt-deduction of C_n from S is a finite sequence of clauses $C_0 \dots C_n$ such that there exists i , $0 < i < n$, such that the sequence $C_0 \dots C_i$ is a l-deduction, and the sequence $C_{i+1} \dots C_n$ is a t-deduction, and C_n is a lt-clause.*

The strategy to prove that to find lt-clauses we can restrict the deduction to those that only contain l-clauses and then only minimal t-clause is very close to the strategy we have adopted to prove Theorem 5.

Lemma 6. *If C is obtained by a lt-inference from C_1 and C_2 , and C_1 is a t-clause and C_2 is a l-clause, and C_1 is the resolvent by Resolution Principle of the two clauses E_1 and E_2 , then there exists a l-inference of E_2 and C_2 whose resolvent is F , and there exists a t-inference of E_1 and F whose resolvent is C .*

If C_1, C_2, E_1, E_2 and C are minimal clauses, then F is a minimal clause.

Proof. *The proof is very close to the proof of Lemma 3.*

⁷The SOL-deduction [8] cannot be directly applied because the property of being a c-clause does not define a stable production field.

Lemma 7. *If there is a R-deduction of C from S such that C is the resolvent of C_1 and C_2 , and:*

- C is a l-clause, C_1 is a c-clause and C_2 is a l-clause,
- there exists a t-deduction of C_1 from S ,
- there exists a l-deduction of C_2 from S ,

then there exists a lt-deduction of C from S .

Proof. *The proof is by induction on the length of the t-deduction of C_1 .*

Induction hypothesis. If there is a R-deduction of C from S such that C is the resolvent of C_1 and C_2 , and:

- C is a lt-clause, C_1 is a c-clause and C_2 is a l-clause,
- C_1 is obtained by a t-deduction from S whose length is $\leq n$,
- there exists a l-deduction of C_2 from S

then there exists a lt-deduction of C from S .

Assumption. There is a R-deduction of C from S such that C is the resolvent of C_1 and C_2 , and:

- C is a lt-clause, C_1 is a c-clause and C_2 is a l-clause,
- C_1 is obtained by a t-deduction from S whose length is $n + 1$,
- there exists a l-deduction of C_2 from S .

Let E_1 and E_2 be the two clauses such that C_1 is their resolvent by t-inference. Either E_1 or E_2 can be resolved with the clause C_2 (see the proof of Lemma 6). Without loss of generality it can be assumed that this clause is E_2 .

From Lemma 6 the R-deduction of C can be transformed as follows: a l-inference infers the clause F from E_2 and C_2 , and a t-inference infers the clause C from E_1 and F . In this transformation the l-deduction of C_2 and the R-deductions of E_1 and E_2 remain unchanged.

Since F is a l-clause, from Theorem 3 there exists a l-deduction δ_1 of F from S .

Since C is a t-clause, either E_1 or F is a t-clause.

Case 1. *F is a t-clause. From Lemma 4 (replacing C_1 by E_1 , and C_2 by F), there exists a t-deduction δ_2 from S and F whose top clause is F . Therefore the sequence $\delta_1\delta_2$ is a lt-deduction of C from S .*

Case 2. E_1 is a t -clause. Since C_1 is obtained by a t -deduction of length $n + 1$, the t -deduction of E_1 is of length n . Then, by induction hypothesis, there exists a lt -deduction of C from S .

Lemma 8. *If there is a R -deduction of C from S such that C is a lt -clause, then there exists a lt -deduction of C from S .*

Proof. *The proof is by induction on the length of the t -deduction of C .*

Induction hypothesis. If there is a R -deduction of C from S of length $\leq n$ such that C is a lt -clause, then there exists a lt -deduction of C from S .

Assumption. There is a R -deduction of C from S of length $n + 1$ such that C is a lt -clause.

Case 1. *Either C_1 or C_2 is a lt -clause.*

Let's assume that C_2 is a lt -clause. The length of the R -deduction of C_2 is n . Then by induction hypothesis there is a lt -deduction of C_2 from S . Since C and C_2 are t -clauses, from Lemma 4 there is a t -deduction of C from S whose top clause is C_2 . Therefore the lt -deduction of C_2 and this deduction make a lt -deduction of C .

Case 2. *Neither C_1 nor C_2 is a lt -clause.*

Therefore C_1 is a t -clause and C_2 is a l -clause (or vice versa). Then, from Theorem 3 there is a l -deduction of C_2 from S and from Lemma 5 there is a t -deduction of C_1 from S . Therefore, from Lemma 7 there is a lt -deduction of C from S .

Definition 25. Minimal lt -Clause. *A clause is a minimal lt -clause iff it is both a l -clause and a minimal t -clause.*

Definition 26. Minimal lt -Inference. *A lt -inference is an inference which is both a l -inference and a minimal t -inference.*

Definition 27. Minimal lt -Deduction. *A lt -deduction of C_n from S is a finite sequence of clauses $C_0 \dots C_n$ such that there exists i , $0 < i < n$, such that the sequence $C_0 \dots C_i$ is a l -deduction, and the sequence $C_{i+1} \dots C_n$ is a minimal t -deduction, and C_n is a minimal lt -clause.*

Theorem 6. *If there is an R -deduction of C from S such that C is a minimal lt -clause, then there exists a minimal lt -deduction of C' , such that C' subsumes C , from $S \cup S_\tau$, where S_τ is the set of clauses of the form: $\neg L \vee L$ such that the literal L occurs in some clause in S .*

Proof. *The proof is very close to the proof of Theorem 5.*

We have shown that from the set of clauses S_1 we can draw the l-deduction: L1,L2,L3, where L3 is the clause: $\neg p_3(x, c) \vee q(a) \vee l(c)$. Then, this l-deduction can be continued with the following t-deduction, for $t = a$: (T4) $q(a) \vee l(c)$ [from (L3) and (9)]

Then, the deduction L1,L2,L3,T4 is an lt-deduction. This short example shows that a strategy which only draws lt-deductions combines the benefits of l-deductions and t-deductions.

5 Conclusion

It has been shown in section 2 that a new approach to information retrieval requires to define what does it mean that information represented by a formula in a language of first order logic is about a given entity.

This formal definition has been presented in section 3 in terms of variants of a given interpretation like in [5]. However, the definition given in [5] has been extended into two directions. The first one is to accept function symbols in the language. and to characterize the terms which are about an entity (denoted by A^t) and the terms which are only about an entity (denoted by OA^t). The second one is to explicitly represent with a theory K the information we know about an entity. This extension, formally represented by $About(F, t, K)$, allows to prevent unexpected results when we have $About(F, t)$ and, according to our knowledge about t , the formula F is not about t .

In the perspective of designing tools to automatically retrieve information about an entity we have defined deduction strategies based on Resolution Principle. That needs to have formulas in clausal form and we have given a syntactical characterization of all the clauses C such that we have $About(C, t)$.

These strategies generate only consequences that are about an entity. They cover both the generation of consequences for deduction answers, with the minimal t -deductions, and for abduction answers, with the minimal lt-deductions, and it has been proved that they are complete up to clause subsumption. At the present time we have not found a specific strategy which only derives consequences that are about an entity in the context of a given theory (i.e. such that we have $About(F, t, K)$). A possible method, which is far to be optimal, could be to check whether the consequences satisfying $About(F, t)$ are consequences of K .

The presented strategies have not been implemented and their implementation requires further researches if we want to have acceptable performances.

Nevertheless, they define a bridge between abstract semantic definitions and effective automated deduction methods.

The approach and the techniques presented in this paper are rather new and there are many related topics that deserves further researches. Some of them are listed below.

Language extension to equality. There are many applications where the background theory contains information about equality and the definition of aboutness has to be revised if equality is added to the language. For instance, if in the theory Smith's father is Dupont, in formal terms: $Dupont = father(Smith)$, the sentence $accident(father(Smith))$ is clearly about Dupont.

Equality also raises difficult problems in designing efficient automated deduction methods. Indeed, a brute force application of paramodulation rule leads to extremely expensive computations. It is possible to find heuristics to reduce the problem, but there are few works in this direction.⁸

Sentences about a given topic. It may be that the information about a given an entity is too large to be efficiently exploited. For instance, if one asks the overall information about a given drug the answer may be extremely large. In that case it can be more convenient to select the information about that drug which is about a given topic, like, for instance, toxicity. In [3] a logic has been proposed for reasoning about sentences that are about a given topic. The combination of this logic with the definition of sentences that are about a given entity would lead to a powerful query language.

Acknowledgements. Comments of an anonymous referee have largely contributed to the improvement of this paper. We are also grateful to Marta Cialdea Mayer and Katsumi Inoue for their significant comments. Of course, if there are mistakes they are the only responsibility of the authors.

References

- [1] R. Carnap. The logical syntax of language. 1937.
- [2] C.L. Chang and R.C.T. Lee. *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.
- [3] R. Demolombe and A.J.I. Jones. On sentences of the kind "sentence "p" is about topic "t": some steps toward a formal-logical analysis. In H-J. Ohlbach and U. Reyle, editor, *Logic, Language and Reasoning. Essays in Honor of Dov Gabbay*. Kluwer Academic Press, 1999.

⁸See, for instance, [6] where SOL deduction has been extended to equality.

- [4] R. Demolombe and L. Fariñas del Cerro. An Inference Rule for Hypothesis Generation. In *Proc. of International Joint Conference on Artificial Intelligence*, Sydney, 1991.
- [5] R. Demolombe and L. Fariñas del Cerro. Towards a logical characterisation of sentences of the kind “sentence p is about object c”. In S. Holldobler, editor, *Intellectics and Computational Logic. Papers in Honor of Wolfgang Bibel*. Kluwer Academic Press, 2000.
- [6] R. Demolombe and M. P. Pozos Parra. An extension of sol-resolution to theories with equality. In *Proceedings of the International Joint Conference on Automated Reasoning*, 2001.
- [7] N. Goodman. About. *Mind*, LXX(277), 1961.
- [8] K. Inoue. Consequence-Finding Based on Ordered Linear Resolution. In *Proc. of International Joint Conference on Artificial Intelligence*, Sydney, 1991.
- [9] K. Inoue. Linear Resolution for Consequence Finding. *Artificial intelligence, an International Journal*, 56, 1992.
- [10] K. Inoue. *Studies on Abductive and Nonmonotonic Reasoning*. PhD thesis, Kyoto University, 1992.
- [11] R. Kowalski and D. Kuhner. Linear resolution with selection function. *Artificial Intelligence*, 2:227–260, 1971.
- [12] L. Fariñas del Cerro and V. Lugardon. Sequents for dependence logic. *Logique et Analyse*, 133-134, 1994.
- [13] R.C.T. Lee. *A completeness theorem and a computer program for finding theorems derivable from given axioms*. PhD thesis, Univ. of California at Berkeley, 1967.
- [14] D. K. Lewis. Relevant implication. *Theoria*, LIV(3), 1988.
- [15] M. Cialdea Mayer and F. Pirri. Abduction is not Deduction-in-Reverse. *Journal of the Interest Group in Pure and Applied Logics (IGPL)*, 4(1):1–14, 1996.
- [16] H. Putnam. Formalization of the concept “About”. *Philosophy of Science*, XXV:125–130, 1958.

- [17] J. A. Robinson. A machine-oriented logic based on the resolution principle. *JACM*, 12:23–41, 1965.
- [18] John Alan Robinson and Andrei Voronkov, editors. *Handbook of Automated Reasoning (in 2 volumes)*. Elsevier and MIT Press, 2001.