Extending Tableaux Calculus with Limited Regular Expression for Role Path: an Application to Natural Language Processing

David Rudloff, François De Beuvron and **Michael Schlick**
{ rudloff, beuvron, schlick}@eric.u-strasbg.fr

Abstract

The main challenge in a natural language interface for databases is to provide easy portability and fast customization for a new database. In this focus, we try to design a simple syntactic analyser that could be plugged to a database model with minimum efforts. We use consistency test and classification capabilities of description logics to solve ambiguities and semantic shortcuts. The introduction of limited regular expressions in Tableaux Calculus is studied for this purpose.

1 Introduction

A natural language interface for databases is quite different from a general natural language translation processing. The domain is limited and defined in an entity/relationship model. The vocabulary is predictible in a large proportion and queries have often the same construction. However the customization of a natural language interface for a new database is not straightforward because the connection between a lexical base and the conceptual model depends on the naming convention in the model. Another issue concerns the query construction. Actually the end user often does not know exactly the model structure. Actually, he may write semantic shortcuts not directly matching to the model and needing deductions. The query formulation in natural language usually describes conceptual information with approximative relation specification. Lastly, a parser often requires semantic information embeded in it. We want, when it is possible, to clearly separate the syntactic parsing phase and the logical coreference resolution. In this way, a description logic-based system will give a complete algorithm to produce all possible consistent interpretations of the natural language query. We designed also a basic portable syntactic analyser that simply extracts concept and role names. These unary and binary predicates have to be unified in a single conceptual expression. The semantic reconstruction can be done during the consistency test in CICLOP[1], a Tableaux Calculus based system. In this purpose, we introduced an anonymous role operator close to regular expression. These ideas are integrated in the CICLOP description-logic system.

2 A basic syntactic analysis

The query translation involves two tasks: (1) to find the main concept described by the query, (2) to find the elements of this main concept required in the result. In this paper, we only address the first task without studying the specification of result elements. We want to find a whole concept that contains every relation refered by the query formulation. The syntactic analysis will consist of finding concept and relation names. Our general assumption is to consider nouns as concept names and verbs as relation names. We claim that this simplification is efficient enough for most cases. In other more complexe cases we are working on relation reification in our system. Then, the modeling methodology is done under the following guidelines:

- concept names are nouns;
- role names are verbs in the infinitive form;
- a synonym set is attached to each concept and role name. Synset intersections are detected and shown to the designer;

[1] CICLOP is the description logics system developped by the LIIA research group (Strasbourg, France). It is used in the CALIS project, concerning Natural Language Interface for Database, funded by the computering company Neurocim.

- the model is based on a simple ontology that guides the choice of role name to inherit automatically a predefined lexical base like in WordNet [1].

The domain currently studied is an enterprise model. In the following example, we will only show a simplified representation of an enterprise database involving client, article, supplier and invoice. This model (Table 1) is designed using the description logic $\mathcal{ALC}$. The predefined concept $Value$ is a set. It represents host types, that are immediate values with a viewable representation. Two subtypes are defined: $Number$ and $String$. In our terminology, only instances of $Value$ can be used in the definition of a concept, because they are never reclassified. We will extend the semantic with value restriction for these types.

Person	$\sqsubseteq$ (AND (DEF[a] toName Name)
Client	$\sqsubseteq$ (AND Person (DEF toLive Address))
Supplier	$\sqsubseteq$ (AND Person (DEF toOwn Telephon)
Article	$\doteq$ (AND (DEF toName Designation) (DEF toCost Price)) (DEF toComeFrom Supplier))
Invoice	$\doteq$ (AND (DEF toDate Date) (DEF toComeFrom Client) (DEF toConcern Article) (DEF toCount Quantity))
Name	$\sqsubseteq$ String
Adress	$\sqsubseteq$ String
Designation	$\sqsubseteq$ String
Price	$\sqsubseteq$ Number
Quantity	$\sqsubseteq$ Number
Value	$\sqsubseteq$ Top
String	$\sqsubseteq$ Value
Number	$\sqsubseteq$ Value

[a] The operator `DEF` is a shortcut for both ALL and SOME restriction. `(DEF R C)`is equivalent to `(ALL R C) AND (SOME R C)`. This means that the role `r` is defined for the concept with at least one occurence.

Table 1: *Modelisation methodology*

Queries are done upon this model. For example:

(1) SHOW THE NAME OF THE ARTICLE THAT COSTS 20$

The syntactic analyser will use local grammar to identify a price (20$), a date or a string value. Nouns and verbs are matched to the lexical base and the model.

This query sentence can be represented in a first order predicate logic:

$$Name(x_1) \wedge Article(x_2) \wedge toCost(x_3, x_4) \wedge Price(x_5) \wedge (x_5 = \text{"20\$"})$$

We note that the syntactic analyser may not be able to do a choice concerning the connections between the predicates. In this worst case, we want the description logic-based system to handle this clause and to try to produce a single description.

3 Description Logic expression reconstruction

The final query representation is a conceptual expression. We have to link each part of the logical form to obtain a single description. This semantic query reconstruction idea has some links with works on *spreading activation*. There were based on psychological theory of the mind and led to sense reconstruction algorithms [2]. The final conceptual query expression should be:

$$Article \sqcap (\exists\ toName\ Name) \sqcap (\exists\ toCost\ (Price \sqcap (= 20))$$

To verify the query consistency, the terminology has to be closed. We use the terminology closure definition given by Weida in [3]. After the terminology containing the model is closed, the following rules are defined:

- a query concept is inconsistent if it defines a new role. New constraints on existing restricted roles are allowed;
- role names are exhaustivly known and represented by the set $\Re$. We will see farther that for each concept name C_N it is possible to calculate the $\mathcal{PR}(C_N)$ set of possible role names.
- two concepts with no common subsumee are considered as disjoint.

We added to the ($\mathcal{ALC}$) logic the host value restrictions (like $=$, $<$, $>$) (noted v), inverse role (noted $\mathcal{I}$) and a anonymous existential role restriction (noted ρ) giving the $\rho\mathcal{ALCI}v$ logic. The semantic of this ρ operator is as follows

$$\{d \in \Delta^{\mathcal{I}} \mid \exists R \in \Re, R^{\mathcal{I}}(d) \cap C^{\mathcal{I}} \neq \emptyset\}$$

A relation link involves a domain concept, a codomain concept and a role. In most cases, the role name is omitted. Sometimes only the role name and the codomain concept name are known. The resolution procedure implies the following operations:

- variable unification;
- role name insertion;
- domain or codomain concept name insertion;
- path of role insertion.

We want to connect the concepts quoted in the natural language query in order to construct a more general meaning. In other words, the semantic query reconstruction try to find relation paths in the model that are consistent and to link the concepts of the query.

4 Regular Expression in Tableaux Calculus

Our description logic system is based on the Tableaux Calculus technic [4]. Our idea is to use this consistency test procedure to try to construct the possible unified descriptions. In this purpose, we add the anonymous operators to the logic syntax. They will be used to represent a link path between two concepts. Such operators increase considerably the algorithm complexity. To handle this, we propose the following restrictions:

- the model design is only based on the $\mathcal{ALC}$ logic;
- the terminology is closed;
- only queries are based on the $\rho\mathcal{ALCI}v$ logic.

We define also the ρ^* operator with the following constraint propagation rules:

$$\begin{array}{lll} x : \rho^*\ C & := & \mid x : C \\ & & \mid x : \rho\ C \\ & & \mid x : \rho\ (\rho^*\ C) \end{array}$$

This operator is close to the one defined in converse-PDL which is decidable and EXPTIME complete [5]. However we have to verify that it is really equivalent as the transitive closure for this ρ operator may involve different role names at each ρ expansion. In the hope to handle complexity issues, we actually block the ρ^* deepness to a limit value compatible with the complexity of the model. Thus, we will call it ρ^n to be clearer with n set at the beginning of the consistency test.

The propagation rule for ρ is special. ρ has to be replaced by role name depending on the restricted role attached to the embedding expression. In a first approach, we could consider the whole closed set of role names $\Re$. In fact, we will reduce this set by calculating what are really the role names usable for the ρ expansion.
Let be $\mathcal{R}(C_N)$ the set of role names restricted in the description of the concept name C_N.

$$\mathcal{R}(C_N) = \mathcal{NR}(C_N) \cup \bigcup_{N \in Subsumer(C_N)} \mathcal{R}(N)$$

Let be $\mathcal{PR}(C_N)$ the set of role names that can be used to constrain C_N. This set is intended to allow the use of role names defined by its subsumees.

$$\mathcal{PR}(C_N) = \mathcal{R}(C_N) \cup \bigcup_{N \in Subsumee(C_N)} \mathcal{PR}(N)$$

In this definition, we note that $\mathcal{PR}(\top) = \Re$ that is the whole set of role name from the closed terminology. When expanding the constraint $(x : \rho\ C)$, we consider every constraints of the type $(x : D)$ from the constraint set $\mathcal{S}$ where D is a concept name of the terminology. The possible role names are then those of each D. Let be $\mathcal{VR}(x)$ the set of role names that can be used for the expansion of $(x : \rho\ \ldots)$.

$$\mathcal{VR}(x) = \bigcup_{\{D \mid (x:D) \in \mathcal{S}\}} \mathcal{PR}(D)$$

It is then possible to define propagation rules for ρ

$$\begin{array}{lll} x : \rho\ C & := & \mid x : \exists\ R_1^x\ C \\ & & \mid \ldots \\ & & \mid x : \exists\ R_n^x\ C \\ & & where\ R_1^x\ to\ R_n^x\ are\ in\ \mathcal{VR}(x) \end{array}$$

With these definitions, a concept that uses a role name only defined in one of its subconcept can be consistent. For example, the concept $Person \sqcap (\exists toLive\ Adress))$ is consistent because $\mathcal{PR}(Person)$ contains the role name $toLive$ given by its subconcept `Client`. When expanding a constraint $(x : \rho C)$, if no constraint $(x : D)$ have already been expanded, we can assume the constraint $(x : \top)$ and then $\mathcal{VR}(x) = \mathcal{R}(\top) = \Re$. We remind that all these assumptions are done in a closed terminology where two concepts with no common subsumee are assumed to be disjoint.

Let us take a detailled example:

(2) SHOW THE ARTICLE OF THE CLIENT X

The basic syntactic analysis will produce:

$$Article(x_1) \wedge Client(x_2) \wedge (x_2 = "X")$$

corresponding to the following conceptual expression:

$$(\rho^n Article) \sqcap (\rho^n (Client \sqcap (= "X")))$$

The following constraint system is then constructed:

$$\boxed{\begin{array}{l} x : (\rho^n Article) \\ x : (\rho^n (Client \sqcap (= "X")) \end{array}} \rightarrow$$

$$\begin{array}{ll}
\boxed{\begin{array}{l} x : Article \\ x : (\rho^n (Client \sqcap (= "X")) \end{array}} \rightarrow & \\
\quad \boxed{\begin{array}{l} x : Article \\ x : (Client \sqcap (= "X"))) \end{array}} & \text{(a)} \\
\quad \boxed{\begin{array}{l} x : Article \\ x : \rho(Client \sqcap (= "X"))) \end{array}} & \text{(b)} \\
\quad \boxed{\begin{array}{l} x : Article \\ x : \rho(\rho^n (Client \sqcap (= "X")))) \end{array}} & \text{(c)} \\
\boxed{\begin{array}{l} x : \rho Article \\ x : (\rho^n (Client \sqcap (= "X")) \end{array}} \rightarrow \ldots \text{(d)} &
\end{array}$$

The case (a) leads to a *Clash* because *Article* and *Client* are disjoint. The case (b) leads to a *Clash* because no direct role of *Article* have a codomain subsumed by *Client*. In the case (d), the same pathes are constructed but starting from *Client*. By following the (c) branch,

$$\boxed{\begin{array}{l} x : Article \\ x : \rho(\rho^n (Client \sqcap (= "X")))) \end{array}} \text{(c)}$$

$$\rightarrow \ldots \rightarrow$$

$$\boxed{\begin{array}{l} x : Article \\ x : toConcern^{-1}(\rho^n (Client \sqcap (= "X")))) \end{array}}$$

$$\rightarrow \ldots \rightarrow$$

$$\boxed{\begin{array}{l} x : Article \\ x : toConcern^{-1}\ y \\ y : Invoice \\ y : toComeFrom\ z \\ z : Client \sqcap (= "X") \end{array}}$$

We note that inverse role are used to go up to the *Invoice* concept.

At this point, each regular expression is determined, we obtain finally the following expression:

$$Q' : Article \sqcap (\exists\ toConcern^{-1}(Invoice \sqcap (\exists toComeFrom\ (Client \sqcap (= "X")))))$$

To avoid reverse role, the query can be formulated as

$$Q'' : Invoice \sqcap (\exists\ toConcernArticle) \sqcap (\exists toComeFrom\ (Client \sqcap (= "X")))$$

The query should then be displayed in natural language to the user for verification:

(3) Do you mean: SHOW THE ARTICLE CONCERNED BY THE INVOICE COMING FROM THE CLIENT X ?

5 Conclusion

In order to provide a simple portable natural language query parser, we worked on the separation of the syntactic analysis and semantic one. In the context of database query, the conceptual model is limited and organized in a entity-relationship model. We tried to use our description logic based system CICLOP, to produce each consistent interpretation of the basic parsing result. Each interpretation consists of a semantic reconstruction by linking separate conceptual informations into one terminological concept. This paper presented a very pragmatic extension of the $\mathcal{ALC}$ description logics to recompose separated semantic elements of this natural language query. We introduced a new operator and new propagation rules in the Tableaux Calculs algorithm to handle anonymous role name. Because of complexity issues, we impose some limitations. However we claim this method is efficient and useful for a database domain normally limited. The main feature we want to provide is the portability and the simplicity of the syntactic analyser required by the system.

References

[1] Georges A. Miller, Rochard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. Introduction to wordnet: An on-line lexical database. Technical report, 1993.

[2] M. Ross Quillian. Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, (12):410–430, 1967. Also edited in BraLe90: Readings in Knowledge Representation.

[3] Robert Anthony Weida. *Closed Terminologies and Temporal Reasoning in Description Logic for Concept and Plan Recognition.* PhD thesis, Columbia University, 1996.

[4] Paolo Bresciani, Enrico Franconi, and Serge Tessaris. Implementing and testing expressive description logics: a preliminary report. In IRST, editor, *Proceedings of the 1995 International Workshop on Description Logics*, Rome, 1995.

[5] Giuseppe De Giacomo and Maurizio Lenzerini. Tbox and abox reasoning in expressive description logics. Technical report, La Sapienza, 1997.