# PREDICATION GRAPHS AS CANONICAL REPRESENTATION OF QUERY SENTENCES

Honkela, T., Lehtola, A. and Valkonen, K.
KIELIKONE-project, SITRA Foundation
P.O.Box 329, SF-00121 Helsinki
Finland
tel. intl + 358 0 641 877

This paper surveys problems encountered in studying the logico-semantic form and discourse problems of Finnish query sentences. We call the logico-semantic form a predication graph. The basic framework we use to represent the logical form of Finnish query sentences is an annotated logical tree transformed from the dependency parse tree using graph transformations of the AWARE-system. Examples of analysing elliptic and anaphoric expressions are given. Finally, some critical points of computational semantics are discussed.

Keywords: semantics, predication, discourse analysis

## 1 CANONICAL STRUCTURE

One of the main objectives of the logico-semantic level is to give those utterances that differ only syntactically a uniform representation. This representation is a predication structure with predicates and their arguments. For example, the expressions "valtiot, jotka tuottavat laivoja" (countries that produce vessels) and "laivoja tuottavat valtiot" (countries producing vessels) would lead into a single logico-semantic form something like as follows:

```
          PRODUCE
             !
    +--------+--------+
    !                 !
 COUNTRY           VESSEL
   PL                PL
```

The arguments are kind of typed variables. The expression could also be stated: produce(X,Y) and country(X) and vessel(Y). An argument may further be another predication. If the word 'countries' were replaced by 'nordic countries' then the resulting structure would be:

```
               PRODUCE
                  !
         +--------+--------+
         !                 !
     SITUATION          VESSEL
         !                PL
    +----+-----+
    !          !
 COUNTRY     NORDIC
   PL
```

Predication Graphs as Canonical Representation of Query Sentences
Timo Honkela, Aarno Lehtola, K. Valkonen
Proceedings of NODALIDA 1987, pages 69-77

This could be also expressed with a clause: produce(X,Y) and country(X) and situation(X,nordic) and vessel(Y). Our representational form makes the variables in the logical form implicit thus making it more readable. The choise of the variable to be passed to an upper predication is demand driven. The solution is based on the ideas of polymorphism.

## 2 CHOICE OF PREDICATES

There are two essential decisions to be made when the predicates for logico-semantic form are selected. Firstly, does the system interpret the semantic content of utterances strongly thus making possibly also strong reduction or does the system rely on the original form by using the verbs with their valences as predicates with arguments? Secondly, one must decide whether the predicates are general, specific or both.

### 2.1 GENERAL PREDICATIONS IN NLI FOR DATABASES

The selection of predications is not determined by the features of the AWARE-system. General predicates as well as specific ones may be used. The degree of canonization depends on the person(s) who makes the semantic modeling, too. In the natural language interface for databases, SUOMEX (Jäppinen & al. 1988) we use general predicates. At conceptual level these predicates reflect the entity-attribute-relationship (EAR) approach for conceptual modeling. This is motivated by the fact that these predicates reflect the conceptual models of the databases.

Let's assume that we have companies with a branch of business and certain properties (or attributes). Here we have an example of using AWARE transformation rules to create predication structure for expressions like "Anna metsäalan yhtiöiden liikevaihto!" (Give the turnover of the companies in forestry).

The first rule instance matches to dependency structures (for more about dependency parsing see Lehtola et al.,1985 or Valkonen et al.,1987) where the dependent is restricted to be a particular branch and the regent is any synonym for company. This very simple rule contains only semantic conditions in addition to the dependency structure specified. Further syntactic checks could be added to avoid overgeneration.

```
              X2
       (EQUALTO: company)
               !
       +-------+
       !
      X1
  (AKO: branch)

  =>
         'HAVE-PROPERTY
               !
       +-------+-------+
       !               !
   ROOT-ONLY          X1
      X2
```

The second example is an instance of a rule for covering expressions stating an entity to have a certain property. The entity here could be a company, companies, companies in certain branch etc. The expression is allowed to have an specification of point or interval of time.

```
              X2
       (AKO: attribute)
           ANYORDER
            ANYOPTS
   (REST (AKO: time_expression))
               !
       +-------+
       !
      X1
  (AKO: entity)
    GenAttr
    ANYDEPS

  =>
        'HAVE-PROPERTY
               !
       +-------+-------+-------+
       !               !       !
      X1           ROOT-ONLY REST
                      X2
```

The predication structure of the expression "Näytä metsäalan yritysten liikevaihto vuodelta 1983!" (Show the turnover of the companies in forestry in 1983) is shown below.

```
                       HAVE-PROPERTY
                             !
            +----------------+------+--------+
            !                       !        !
      HAVE-PROPERTY            TURNOVER    YEAR
            !                                !
    +-------+-------+                        +-------+
    !               !                                !
 COMPANY          BRANCH                            1983
   PL            forestry
```

To emphasize the generality of AWARE, one must state that the choice of predicates such as 'HAVE-PROPERTY' follows from its use as a part of a database interface.

## 2.2 CONCEPTUAL HIERARCHY

Nodes in graph transformation rules may contain semantic restrictions. For each restriction a proper level of generality is needed. Information about conceptual classes is in a form of hierarchy (compare to Grozs et al, 1987). The use of semantic restrictions and their relation to the conceptual hierarchy could be exemplified with pair of expressions like (1) "Peter's car" versus (2) "Peter's wife". The first expression could be transformed into predication 'OWN' but the latter one presumably not. The classification into living and non-living objects can be used to refine the transformation to match appropriately.

## 3 DISCOURSE ANALYSIS WITH GRAPH TRANSFORMATIONS

In many cases it is not possible to interpret a sentence without solving references to the other sentences of the discourse. The AWARE-system makes it possible to analyze also the context of an utterance rather than only a single dependency structure. The expressions of the discourse are gathered under a single node called 'Discourse Node' (DN).

```
      Discourse_node
              !
     +--------+--------+
     !        !        !
 ... series of expressions ...
```

The transformations may refer to DN giving a convenient possibility to handle anaphoric and elliptic utterances.

## 3.1 SOLVING ANAPHORA

In a sentence a word may refer backwards to another word, group of words or a whole sentence replacing it. Pronouns are the most typical case. Here we give some examples of dependency structures with anaphoric reference.

(1) "Luettele yritykset, joilla on yli 1000 työntekijää!"
(Give a list of the companies that have over 1000 employees!)

```
Luettele-+
Predicate!
         !
         +--yritykset-+
            Object    !
                      !
                      !              +---,
                      !              !   Comma
                      !              !
                      !   +--joilla--+
                      !   !  Adverbial
                      !   !
                      +-on-+
                       RelAttr
                          !
                          !                  +-yli
                          !                  ! QuantorAdvAttr
                          !                  !
                          !             +1000-+
                          !             !Attr
                          !             !
                          +--työntekijää+
                             Subject
```

In the example a relative pronoun refers to an antecedent inside the same sentence. A pronoun may also refer to an antecedent in a previous sentence.

(2) "Millä yrityksillä on yli 1000 työntekijää?"
(Which companies have over 1000 employees)
"Mikä on niiden vuosittainen liikevaihto?"
(What is their annual turnover?)

```
                       +Millä
                       !Attr
                       !PreQP
                       !
        +---yrityksillä-+
        !   Adverbial
        !   SimpleSP
        !
on------+
Predicate
CopVP   !
        !
        !                    +-yli
        !                    ! QuantorAdvAttr
        !                    !
        !              +1000-+
        !              !Attr
        !              !
        +--työntekijää--+
           Subject

        +--Mikä
        !  Subject
        !  PreQP
        !
on------+
Predicate
CopVP   !
        !
        !              +niiden
        !              !GenAttr
        !              !PreQP
        !              !
        +--liikevaihto--+
           Complement
           SimpleSP
```

A pair of expressions given here and especially the anaphoric reference can be analysed with the rule given below.

```
                     Discourse_node
                           !
          +----------------+---------------+
          !                                !
         X3                               X5
   'HAVE-PROPERTY                   'HAVE-PROPERTY
          !                                !
   +------+--------+              +--------+--------+
   !               !              !                 !
   !               !             SE                X4
(EQUALTO: company) ANYDEPS     GenAttr     (AKO: attribute)
     PL                          PL            ANYDEPS
  ANYDEPS

=>
                     Discourse_node
                           !
          +----------------+---------------+
          !                                !
         X3                            ROOT-ONLY
                                          X5
                                           !
                                  +--------+--------+
                                  !                 !
                                 X3                X4
```

The node with "niiden" (plural and genetive of "se") is replaced with reference to the structure bound to variable X3.

## 3.2 ELLIPSIS

Often an elliptical sentence is preceeded by a complete sentence, which contains the lexical entities left out from the elliptical sentence.

The first problem is to verify whether an expression is elliptical or not. Some heuristic rules exist but generally the decision cannot be made deterministically by analysing the sentence itself. Those heuristics might include:

- a certain expression is used together with elliptical utterance. ("Entäs ...","What about ...")
- the case of nominal phrase is other than nominative. ("Annen?", "Ann's?")
- nominal phrase is in comparative ("Enemmän kuin Helsingissä?","More than in Helsinki?")
- transitive verb has no object ("Myyty vuonna 1983?","Sold year 1983?")

The expression (1) "Turnover of Nokia?" could be understood as "Give me turnover of Nokia!". If the previous expression were (2) "Is the number of employees of Huhtamäki greater than the average in forestry" it would most propably be understood as elliptic.

Our tranformation rule example for analysis of anaphora had the discourse history as set of logico-semantic forms. AWARE-system may also be used to match parts of current expression towards the dependency structure of previous utterances.

The discourse structure of expressions (1 & 2) is shown below:

```
                    'Discourse_node
                           !
     +---------------------+----------------------+
     !                                            !
   'OLLA                                   'LIIKEVAIHTO
  'Predicate                                   'Head
     !                                            !
     +-------+-------------+              +-------+
             !             !              !
     'TYÖNTEKIJÄMÄÄRÄ   'SUURI          'NOKIA
        'Subject     'Complement      'GenAttr
             !             !
     +-------+             +-------+----------+
     !                             !          !
  'HUHTAMÄKI                     KUIN     'METSÄALA
   'GenAttr                  AdjConjAttr 'AdjPostAttr
                                              !
                                              +-------+
                                                      !
                                               'KESKIMÄÄRIN
                                                'Adverbial
```

Let's consider a rule for handling an ellipsis like this. The right hand side of the rule would contain a reference from the node 'LIIKEVAIHTO' (ie. turnover) to the node 'TYÖNTEKIJÄMÄÄRÄ' (ie. number of employees) thus producing a directed acyclic graph (DAG). DAGs are usually formed when elliptical and anaphoric expressions are analyzed.

## 4 SEMANTIC MODELING

### 4.1 TRUTHCONDITIONS

The view applied in this representation is that the sense of an utterance depends on its truthconditions. This assumption is to be seen as a basis for the way how we handle the logical aspects of natural language. Though we are aware about the limitations of this approach in accordance to other aspects of natural language.

## 4.2 DEPTH OF SEMANTIC ANALYSIS

The conceptual size of the domain a NLU system is developed for largely determines the semantic modeling needed. This could trivially be understood as an linear relation between the size of domain and the semantic model. Actually, if new domain areas are introduced part of the preceeding semantic modeling has to be corrected.

Let's consider again different ways of expressing 'possession'. In most of the cases 'to own' and 'to belong to' could be canonized. Compare examples below:

"This car belongs to my father"
"My father owns this car"

This general rule does not hold in all of the cases, though. Consider for example the following sentences:

"My heart belongs to my daddy"
"My daddy owns my heart"

Such examples are not just peculiarities but show the inherent character of natural language. One important consequence from this is that the methods and tools for semantic analysis should take into account these features (see e.g. Michalski, 1987) including induction and analogy. As a human being inductively infers general "rules" for her own use she also notes the exceptions for their usage.

The phenomenon known as 'the knowledge principle' in the field of artificial intelligence is analogical to the need of large amount semantic modelling for NLU systems. To get results in practical work one must have efficient tools for knowledge acquisition. The AWARE-system takes into account these needs with its graphical representation, rule generator and powerful rulebase maintenance tools. Further plans for research include development of near match analysis and use of machine learning methods.

References

Grosz,B.J.,Appelt,D.E.,Martin,P.A.,Pereira,F.C.N.:
TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces. Artificial Intelligence, Volume 32, Elsevier Science Publishers, 1987, pp. 173-243.

Ishikawa,H. et al.:
A Knowledge-Based Approach to Design a Portable Natural Language Interface to Database Systems. Proceedings of the International Conference on Data Engineering, IEEE Computer Society, Los Angeles, California, 1986, pp. 134-143.

Hafner,C.D. and Godden,K.:
Portability of Syntax and Semantics in Datalog. ACM Transactions on Office Information Systems, Vol.3,No.2, pp. 141-164, 1985.

Hirst, G.:
Anaphora in Natural Language Understanding: A Survey. Springer-Verlag, Berlin 1981, 128 p.

Jäppinen, H., Honkela, T., Lehtola, A. and Valkonen, K.:
Hierarchical Multilevel Processing Model for Natural Language Database Interface. Proceedings of the 4th IEEE Conference on Artificial Intelligence Applications, San Diego, California, 1988, 6p (in print).

Lehtola, A., Jäppinen, H. and Nelimarkka, E.:
Language-based Environment for Natural Language Parsing. Proceedings of the 2nd European Conference of ACL, Geneve, 1985, pp. 98-106.

Michalski, R.S.:
How to Learn Imprecise Concepts: A Method for Employing a Two-Tiered Knowledge Representation in Learning. Proceedings of the 4th International Workshop on Machine Learning, Irvine, California, 1987, pp. 50-58.

Moore, R. C.:
Problems in Logical Form. Proceedings 19th Annual Meeting of the Association for Computational Linguistics, Stanford, The association for computational linguistics, 1981, pp.117-124.

Valkonen, K., Jäppinen, H. and Lehtola, A.:
Blackboard-based Dependency Parsing. 10th International Joint Conference on Artificial Intelligence, Milano, 1987, pp. 700-702.