KYBERNETIKA — VOLUME *25* (1989), NUMBER 6

SEMANTIC ANALYSIS OF TOPIC AND FOCUS

TOMÁŠ VLK

The paper deals with a logical (semantic) interpretation of disambiguated linguistic meanings of natural language sentences. A semantic analysis of topic and focus as two parts of underlying representations (meanings) by means of the transparent intensional logic (TIL) is presented. An informal short description of an algorithm handling the topic/focus articulation in the translation of underlying representations into the constructions of TIL is added.

1. UNDERLYING REPRESENTATIONS

The disambiguated underlying representation in the style of Sgall and Hajičová [9] is basically a dependency tree, where the left-to-right ordering of the nodes corresponds to the scale of dynamism rather than immediately to the surface word order. Informally, the topic of a sentence is what the sentence talks about ("given information"), and the focus is what the sentence says about the topic ("new information"). A formal definition of topic and focus as two parts of the disambiguated or tectogrammatical representation (TR) can be found in Sgall et al. [9]. The lexical tokens belong either to the topic or to the focus of the sentence. The boundary between topic and focus is always placed in such a way that there is a borderline such that every item of the underlying representation which is less (more) dynamic belongs to the topic (focus).

The topic/focus articulation (TFA) plays a crucial role in the analysis of presuppositions, of the scope of negation and also of the so called exhaustive listing [1, 2, 9].

Let us remark that A is a presupposition of B if B entails A and not-B also entails A, see Hajičová [1, 2]. Propositions are understood as $(o\omega)$-objects in TIL. Let A, B be $(o\omega)$-objects. We say that A is a presupposition of B if for all possible worlds w it holds that if $[B\ w]$ is defined then $[A\ w]$ is true (and if $[A\ w]$ is not true then $[B\ w]$

is undefined). The presupposition A determines all possible worlds where B is defined (has a truth-value). To ensure the presence of the desired presupposition means to restrict the set of possible worlds where B is defined.

2. TRANSPARENT INTENSIONAL LOGIC

Sufficient means for the semantic analysis of natural language are given by Tichý's Transparent intensional logic (TIL). Referring as for exact definitions to Tichý [10] and Materna [3] we reproduce here only a brief characterization of TIL.

Let $o = \{T, F\}$ be a set of truth-values, let ι be a set of individuals (the universe of discourse) and let ω be a set of possible worlds (the logical space). Then $B = \{o, \iota, \omega\}$ is an *epistemic basis*. Then

(i) any member of B is a *type over* B,

(ii) if $\eta, \xi_1 \dots \xi_n$ are types over B, then
$(\eta\xi_1 \dots \xi_n)$ is a *type over* B, where
$(\eta\xi_1 \dots \xi_n)$ is the set of (total and partial) functions from $\xi_1 \times \dots \times \xi_n$ to η.

(iii) the types over B are just those introduced in (i), (ii).

Any member of type η is called an *object of type* η, or an η-object. An object is an η-object for any η. For every type a denumerably infinite set of η-variables is at our disposal.

The *constructions* are the ways in which objects can be given. They are defined inductively:

(i) any η-objects, and also any η-variable, is an η-*construction* (called the atomic construction).

(ii) let F be a $(\eta\xi_1 \dots \xi_n)$-construction, X_i a ξ_i-construction for $i = 1, \dots, n$. Then the application $[F\, X_1\, X_2 \dots X_n]$ of F to $X_1, X_2, \dots, X_n$ is an η-*construction*.

(iii) let Y be an η-construction and $x_1, x_2, \dots, x_n$ distinct variables of types $\xi_1, \dots, \xi_n$. respectively. Then the abstraction $[\lambda x_1\, x_2 \dots x_n\, Y]$ of Y on $x_1, x_2, \dots, x_n$ is a $(\eta\xi_1 \dots \xi_n)$-*construction*.

(iv) there are no constructions except those defined in (i)–(iii).

Let us characterize some important objects of TIL. For every type ξ we have Σ^ξ, Π^ξ of the type $(o(o\xi))$, such that (i) and (ii) hold:

(i) $[\Sigma^\xi X] =$ **if** X is the empty class **then** F
else T

(ii) $[\Pi^\xi X] = [\sim[\Sigma^\xi \lambda y[\sim[X\, y]]]]$

For every type ξ we have the ξ-singularizer I^ξ of the type $(\xi(o\xi))$, which is defined on single-element ξ-classes only and returns the single element of the respective class. Propositions are objects of the type $(o\omega)$.

The following notation will be used throughout the paper. The outermost parentheses and brackets will be sometimes omitted. Furthermore, a dot will represent a left bracket whose corresponding right bracket is to be imagined as far to the right as is compatible with other pairs of brackets. The notation with an apostrophe will be used in the following meaning:

$$X' = \begin{cases} [X\ w] & \text{if } X \text{ is of type } (\eta\omega) \text{ for any } \eta \\ X & \text{otherwise} \end{cases}$$

where X is a construction and w is a particular ω-variable. We write $\exists x\ .\ Y$ in place of $[\Sigma^{\xi}\ \lambda x\ Y]$ and $\forall x\ .\ Y$ in place of $[\Pi^{\xi}\ \lambda x\ Y]$, $\iota x\ .\ Y$ in place of $[I^{\xi}\ \lambda x\ Y]$, X/η in place of "X is of the type η". Logical connectives and identity will be written in the standard way, e.g. $a\ \&\ b$, $a = b$ in place of $[\&\ a\ b]$, $[= ab]$ respectively.

Generalized quantifiers So (some), Ev (every) are defined as follows:

$$\text{So} = \lambda x\ \lambda y[\exists z\ .\ [x\ z]\ \&\ [y\ z]]$$
$$\text{Ev} = \lambda x\ \lambda y[\forall z\ .\ [x\ z] \Rightarrow [y\ z]]$$
$$x,\ y/(o\xi)\quad z/\xi$$
$$\text{So, Ev}/(o(o\xi))\ (o\xi)$$

A class of objects is identified with its characteristic function. For example, the function constructed by the $(o\iota)$-construction $\lambda x\ [x = A \vee x = B]$ can also be viewed as the class $\{A, B\}$.

3. INNER PARTICIPANTS

In this section we discuss examples in which the dividing edges between topic and focus are edges having inner participants as their functors, dependent on the verb. We restrict ourselves in this paper to sentences having just one dividing edge.

In all the examples, that part of sentence which belongs to the focus (on the reading discussed) is written in italics.

(1) Tom *sells Jim a car.*

(2) Tom sells a car *to Jim.*

(3) Tom sells Jim *a car.*

Case (1) brings no problem to the semantic analysis. It corresponds to the "normal" (preferred, primary) reading of the sentence. We can formulate the constructions corresponding to the topic and focus as follows:

$$\text{Topic} = \text{Tom}$$

$$\text{Focus} = \lambda w\ \lambda y\ .\ [\text{So Car'}]\ [\lambda x\ .\ \text{Sell'}\ y\ \text{Jim}\ x]$$

Now we can formulate the constructions (they are reduced step by step through the lambda-reduction):

(1′) $\lambda w[\text{Focus' Topic'}]$

$\lambda w[[\lambda y\ .\ [\text{So Car'}]\ [\lambda x\ .\ \text{Sell'}\ y\ \text{Jim}\ x]]\ \text{Tom}]$

$$\lambda w.[\text{So Car'}]\,[\lambda x.\text{Sell' Tom Jim } x]$$
$$\lambda w.\exists x.[\text{Car' } x]\ \&\ [\text{Sell' Tom Jim } x]$$
$$\text{Sell}/(o\iota\iota\iota)\,\omega \qquad \text{So}/(o(o\iota))\,(o\iota)$$
$$\text{Tom, Jim, } x/\iota \quad w/\omega$$
$$\text{Car}/(o\iota)\,\omega$$

Case (2) is more complicated. Here, the verb belongs to the topic of the sentence. The sentence can be paraphrased in the following way (with sentential stress on Jim):

The individual such that Tom sells a car to him is *Jim*.

The topic of the sentence determines a class of individuals such that Tom sells a car to them. The sentence triggers a presupposition that Tom sells a car to somebody, i.e. that this class is non-empty. The focus is an exhaustive listing of such individuals whenever the verb belongs to the topic [5, 9].

To ensure the presence of the desired presupposition we define the function

$$\text{Ne} = \lambda c\,.\,\imath y[y = c\ \&\ \exists x[y\ x]]$$
$$\text{Ne}/(o\iota)\,(o\iota) \quad y, c/(o\iota) \quad x/\iota$$

The function Ne is an identical function defined only on nonempty classes.

The following constructions reflect the desired properties:

$$\text{Topic} = \lambda w.\text{Ne}\ \lambda x.[\text{So Car'}]\,[\lambda y.\text{Sell' Tom } x\ y]$$
$$\text{Focus} = \lambda z.z = \lambda x.x = \text{Jim}$$

(2′) $$\lambda w[\text{Focus' Topic'}]$$
$$\lambda w.[\text{Ne}\ \lambda x.[\text{So Car'}]\,[\lambda y.\text{Sell' Tom } x\ y]] = \lambda x.x = \text{Jim}$$

The verb of sentence (3) again belongs to the topic. The topic of (3) determines a class of individuals that Tom sells to Jim. Again, the presupposition of (3) is that the class is non-empty. The focus of (3) is an exhaustive listing of such individuals (because the verb belongs to the topic). We feel that (3) is not (fully) true if Tom sells Jim a car and a house or if Tom sells Jim two cars. Focus of (3) says that the class determined by the topic has only one element and this element is a car. We need a function "One" dividing a class X into a set of single-element subclasses of X. It should hold that

$$[[\text{One } X]\ Y] \text{ iff } Y \text{ is single-element and } Y \cap X \neq \varnothing$$

For every type η we define

$$\text{One}^{\eta} = \lambda p.\lambda y.[\text{So } p]\,[\lambda z.y = [\lambda x.x = z]]$$
$$x, z/\eta \quad p, y/(o\eta) \quad \text{So}/(o(o\eta))\,(o\eta)$$

The constructions corresponding to the topic and the focus of (3) are

$$\text{Topic} = \lambda w[\text{Ne}\ \lambda x.\text{Sell' Tom Jim } x]$$
$$\text{Focus} = \lambda w[\text{One Car'}]$$

and to sentence (3)

(3′) $$\lambda w.[\text{One Car'}]\,[\text{Ne}\,\lambda x.\text{Sell' Tom Jim}\ x]$$

The proposition constructed by (3′) is defined only in those possible worlds where Tom sells something to Jim (because only in such possible worlds function Ne is defined), and it is true in those possible worlds where Tom sells Jim just one thing and this thing is a car. An alternative analysis of (3) can be found in Materna et al. [4].

The negative counterparts of (2), (3) have two readings each, in accordance with the contextual boundness of the operator of negation, see Hajičová [1, 2]) Let us take the negative counterparts of (2) for illustration.

(4) Tom *does not* sell a car *to Jim* .

(5) Tom does not sell a car *to Jim* .

In (4), it is the focus which constitutes the scope of operator of negation. The topic is the same as in (2):

$$\text{TopicA} = \lambda w.\text{Ne}\,\lambda x.[\text{So Car'}]\,[\lambda y.\text{Sell' Tom}\ x\ y]$$

$$\text{FocusA} = \lambda z.z \neq \lambda x.x = \text{Jim}$$

(4′) $$\lambda w[\text{FocusA}'\ \text{TopicA}']$$

$$\lambda w.[\text{Ne}\,\lambda x.[\text{So Car}']\,[\lambda y.\text{Sell' Tom}\ x\ y]] \neq \lambda x.x = \text{Jim}$$

Construction (4′) ensures the same presupposition as (2′), it is defined only in those possible worlds where Tom sells a car to somebody, and it asserts that the class containing just Jim is not an exhaustive listing of individuals whom Tom sells a car to. (Tom may sell a car to John or to Jim, John and Mary, etc.)

In the following example, where the negation is contextually bound, it is presupposed that there exists a non-empty class of persons to whom Tom does not sell a car, and it is asserted that this class has Jim as its single element.

$$\text{TopicB} = \lambda w.\text{Ne}\,\lambda x.[\text{So Car'}]\,[\lambda y.\sim[\text{Sell' Tom}\ x\ y]]$$

$$\text{FocusB} = \lambda z.z = \lambda x.x = \text{Jim}$$

(5′) $$\lambda w[\text{FocusB' TopicB'}]$$

$$\lambda w.[\text{Ne}\,\lambda x.[\text{So Car'}]\,[\lambda y.\ \sim[\text{Sell}'\ \text{Tom}\ x\ y]]] = \lambda x.x = \text{Jim}$$

4. CAUSE AND AIM

In this section we discuss examples in which the dividing edges between topic and focus are edges having an adverbial complementation of 'Cause', 'Aim', or 'Condition' as their functors. They are analyzed rather as relation-in-intension between propositions, as was pointed out by Materna and Sgall [6].

Let us take the following examples

(6) Because Mary was ill, *Charles came.*

(7) Charles came *because Mary was ill.*

Both the sentences have one dividing edge with the functor "Cause". The sentences have come as their main verb (the root of the dependency tree). In these examples "Cause" is the relation-in-intension between two propositions. The first proposition is in the topic and the second one is in the focus of the sentence. The proposition in the topic is a presupposition of the sentence. Thus (6) has the presupposition that Mary was ill, (7) presupposes that Charles came. To ensure such presuppositions we need a function Tr that would be a "filter" or a "sieve" for true propositions. It should hold for every possible world W and proposition P that

$$[[\mathrm{Tr}\ W]\ P] = \begin{cases} P & \text{if } [P\ W] = T\ (P \text{ is true in } W) \\ \text{undefined otherwise} \end{cases}$$

$$\mathrm{Tr}/((o\omega)\,(o\omega))\,\omega \quad P/(o\omega) \quad W/\omega$$

This requirement is fulfilled by the definition:

$$\mathrm{Tr} = \lambda w\, \lambda p[\imath q\,.\, q = p\, \&\, [q\ w]]$$

$$\mathrm{Tr}/((o\omega)\,(o\omega))\,\omega \quad p, q/(o\omega) \quad w/\omega$$

If $\mathrm{Caus}/(o(o\omega)\,(o\omega))\,\omega$ realizes the relationship "Cause", so that $\lambda w[\mathrm{Caus'}\ \mathrm{P1}\ \mathrm{P2}]$ means that P2 causes P1, the constructions corresponding to (6) and (7) are as follows:

(6′) $\lambda w.\mathrm{Caus'}\ [\lambda w.\mathrm{Come'}\ \mathrm{Charles}]\ [\mathrm{Tr'}\ \lambda w.\mathrm{Ill'}\ \mathrm{Mary}]$

(7′) $\lambda w.\mathrm{Caus'}\ [\mathrm{Tr'}\ \lambda w.\mathrm{Come'}\ \mathrm{Charles}]\ [\lambda w.\mathrm{Ill'}\ \mathrm{Mary}]$

5. TFA ALGORITHM

In this section we describe an algorithm handling the TFA. For the sake of simplicity, only examples with one dividing edge between the topic and focus were presented. A general case (with more than one dividing edge) requires the introduction of additional functions [11]. Here we present a simplified version of the algorithm, which reflects only the phenomena discussed in this paper.

First we summarize the functions defined in the previous sections.

$$\mathrm{Tr} = \lambda w.\lambda p[\imath y.y = p\,\&\,[y\ w]]$$

$$\mathrm{Ne} = \lambda p.\imath y[y = p\,\&\,\exists x_1 \ldots x_n[y\ x_1 \ldots x_n]]$$

$$\mathrm{One} = \lambda p.\lambda y.[\mathrm{So}\ p]\,[\lambda z.y = \lambda x.x = z]$$

$$\mathrm{Tr}/((o\omega)\,(o\omega))\,\omega$$

$$\mathrm{Ne}/(o\xi_1 \ldots \xi_n)\,(o\xi_1 \ldots \xi_n)$$

$$\mathrm{One}/(o(o\eta))\,(o\eta)$$

The following functions are used in the description:

CB:	DepTree	→ Bool
NB:	DepTree	→ Bool
NBNeg:	DepTree	→ Bool
Tree:	Edge	→ DepTree
Fun:	Edge	→ Functor
M:	Functor	→ Construction
R-Edge:	Edge	→ Bool
A-Edge:	Edge	→ Bool
DivEdge:	DepTree	→ Edge
DelEdge:	DepTree Edge	→ DepTree
PutVar:	DepTree Edge	→ DepTree
Translate:	DepTree	→ Construction
GetTyp:	Construction	→ Type

The meanings of the functions are as follows:

CB(dt) returns true iff the root of dt is contextually bound. NB(dt) returns true iff CB(dt) returns false (NB(dt) = $\sim$CB(dt)). NBNeg(dt) returns true iff the contextually nonbound operator of negation is connected with the root of dt (contextually bound operator of negation is handled by the function Translate, see [11]).

Tree(e) returns the dependency tree suspended on edge e. Fun(e) returns the functor of edge e. $M(f)$ returns the object of TIL realizing relationship f ('Cause', 'Aim'). R-Edge(e) returns true iff e is an R-Edge ('Cause', 'Aim', 'Condition'). A-Edge(e) returns true iff e is an A-Edge (an inner participant). DivEdge(dt) returns the dividing edge between the topic and the focus of dt.

The functions DelEdge and PutVar realize dividing of the dependency tree. DelEdge(dt, e) returns the dependency tree dt without edge e (edge e is removed from dt). PutVar(dt, e) replaces the tree suspended on edge e in tr by a variable and returns the resulting dependency tree.

Translate(dt) returns the construction of TIL corresponding to the "primary reading" of the underlying structure dt (i.e. without taking into account the topic/focus articulation). The function Translate is described in [11]. GetTyp(c) returns the type of construction c.

Now we can describe the following procedures:

TFA – the main procedure (function)
FA – verb in the focus, dividing A-edge
TA – verb in the topic, dividing A-edge
FR – verb in the focus, dividing R-edge
TR – verb in the topic, dividing R-edge

TFA: DepTree → Construction

TFA(dt) =

$$
\begin{array}{l}
\textbf{let } e = \mathrm{DivEdge}(dt) \textbf{ in} \\
(\text{A-Edge}(e) \,\&\, \mathrm{NB}(dt) \rightarrow \mathrm{FA}(dt)\,, \\
\ \text{A-Edge}(e) \,\&\, \mathrm{CB}(dt) \rightarrow \mathrm{TA}(dt)\,, \\
\ \text{R-Edge}(e) \,\&\, \mathrm{NB}(dt) \rightarrow \mathrm{FR}(dt)\,, \\
\ \text{R-Edge}(e) \,\&\, \mathrm{CB}(dt) \rightarrow \mathrm{TR}(dt) \\
)\,;
\end{array}
$$

If the dividing edge is an A-edge and the verb belongs to the focus the tree is handled by the function FA. The tree suspended on the dividing edge is replaced by a variable, the topic and focus are translated separately and the resulting construction is put together. F is the construction corresponding to the focus and T is the construction corresponding to the topic.

FA: DepTree → Construction

FA(dt) =

$$
\begin{array}{l}
\textbf{let } e = \mathrm{DivEdge}(dt) \textbf{ in} \\
\textbf{let } F = \mathrm{Translate}\ (\mathrm{PutVar}(dt, e))\,, \\
\quad T = \mathrm{Translate}\ (\mathrm{Tree}(e)) \\
\textbf{in} \\
\quad \textbf{if } \mathrm{NBNeg}(dt) \textbf{ then } [\lambda w.\,{}^{\sim} [F'\ T']] \\
\qquad\qquad\qquad \textbf{else } [\lambda w[F'\ T']]
\end{array}
$$

If the dividing edge is an A-edge and the verb belongs to the topic the tree is handled by the function TA. The tree is divided in the same manner as in FA. The resulting construction is more complicated than in TA because it has to reflect presuppositions and exhaustive listing.

TA: DepTree → Construction

TA(dt)

$$
\begin{array}{l}
\textbf{let } e = \mathrm{DivEdge}(dt) \textbf{ in} \\
\textbf{let } T = \mathrm{Translate}\ (\mathrm{PutVar}(dt, e))\,, \\
\quad F = \mathrm{Translate}\ (\mathrm{Tree}(e)) \\
\textbf{in} \\
\textbf{let} \\
\quad Y = (\textbf{if } \mathrm{GetTyp}(F') = \mathrm{GetTyp}(T') \\
\qquad\qquad\qquad \textbf{then } [[\mathrm{One}\ F']\ [Ne\ T']] \\
\qquad\qquad\qquad \textbf{else } [[\lambda y\,.\,y = F'] = [Ne\ T']]) \\
\textbf{in} \\
\quad \textbf{if } \mathrm{NBNeg}(dt) \textbf{ then } [\lambda w.\,{}^{\sim}\ Y] \\
\qquad\qquad\qquad \textbf{else } [\lambda w.\ Y]\,;
\end{array}
$$

If the dividing edge is an R-edge and the verb belongs to the focus the tree is translated by the function FR. Here the dividing edge is removed from the tree and the functor of the dividing edge determines a relationship between the topic and the focus. The proposition in the focus is presupposed, the presupposition is ensured by the function Tr. The relationship between the topic and the focus is not within the scope of negation.

FR: DepTree → Construction

FR(dt) =
let e = DivEdge(dt) **in**
let F = Translate (DelEdge (dt, e)),
T = Translate (Tree(e)),
P = M(Fun(e))
in
if NBNeg(dt) **then** $[\lambda w.[P'[\lambda w.{}^{\sim} F'] [\mathrm{Tr}'\ T]]]$
else $[\lambda w[P'\ F[\mathrm{Tr}'\ T]]]$;

If the dividing edge is an R-edge and the verb belongs to the topic, the tree is translated by function TR. The tree is divided in the same manner as in FR. The relationship between the topic and focus is within the scope of negation here.

TR: DepTree → Construction

TR(dt) =
let e = DivEdge(dt) **in**
let T = Translate (DelEdge(dt, e)),
F = Translate (Tree(e)),
P = M(Fun(e))
in
if NBNeg(dt) **then** $[\lambda w.{}^{\sim} [P'[\mathrm{Tr}'\ T]\ F]]$
else $[\lambda w[P'\ [\mathrm{Tr}'\ T]\ F]]$;

ACKNOWLEDGEMENT

The author wishes to thank Prof. Petr Sgall for his support of this work and valuable discussions.

(Received October 13, 1988.)

REFERENCES

[1] E. Hajičová: Meaning, presupposition and allegation. Philologica Pragensia *17* (1974), 18—25.
[2] E. Hajičová: Presupposition and allegation revisited. Journal of Pragmatics *8* (1984), 155—167.
[3] P. Materna: "Linguistic constructions" in the transparent intensional logic. PBML *43* (1985), 5—24.

[4] P. Materna, E. Hajičová and P. Sgall: Redundant answers and topic/focus articulation. Linguistics and Philosophy *10* (1987), 101—113.
[5] P. Materna and P. Sgall: Functional sentence perspective, the question test and intensional semantics. SMIL 1—2 (1980), 141—160.
[6] P. Materna and P. Sgall: Optional participants in semantic interpretations (arity of predicates and case frames of verbs). In: [8], 51—62.
[7] M. Plátek, J. Sgall and P. Sgall: A dependency base for linguistic descriptions. In: [8], 63—97.
[8] P. Sgall (ed.): Contributions to Functional Syntax, Semantics and Language Comprehension. Prague, Academia 1984.
[9] P. Sgall, E. Hajičová and J. Panevová: The Meaning of the Sentence in Its Semantic and Pragmatic Aspects. Academia, Prague 1986.
[10] P. Tichý: The logic of temporal discourse. Linguistics and Philosophy *3* (1980), 343—369.
[11] T. Vlk: Towards a transduction of underlying structures into intensional logic. PBML *50* (1988), 35—70.

RNDr. Tomáš Vlk, Výpočetní středisko Poldi — SONP, (Computer Center, Poldi — United Steelworks) 27262 Kladno. Czechoslovakia.