

Natural Language Dialogue Formalization: From Hyperintensional Logic to Linear Logic

Zuzana Bilanová ^[0000-0001-6347-2409] and Ján Perháč ^[0000-0002-0128-5111]

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Košice, Slovakia
{zuzana.bilanova,jan.perhac}@tuke.sk

Abstract. This paper deals with a formalization of a natural language dialogue using transparent intensional logic and predicate linear logic. In the beginning, it is necessary to choose the dialogue in a natural language. Subsequently, this dialogue is analyzed by transparent intensional logic, that allows it to capture the meaning of sentences in logical structures. At this stage, a three-step method of an analysis will be applied to the dialogue: at first, there will be a type analysis of the lexical units, which is the base for the logical constructions that capture the meaning of the sentences and then we apply a type checking to verify the correctness of the solution. In the second part of the paper, the same dialogue is analyzed by the Ludics theory, which represents the superstructure of the predicate linear logic. In a technical presentation of Ludics theory, the dialogue is placed in a logical space and time what allows model interactions between its actors.

Keywords: Logical Analysis of Natural Language · Ludics Theory · Predicate Linear Logic · Transparent Intensional Logic.

1 Introduction

Computer programs require people to communicate with them in a formal programming language that is accurate, unambiguous, and structured. Human speech does not have these attributes. The natural language has not been discovered, but gradually developed spontaneously over the centuries in communicative practice and adapted to changing conditions in the society. This causes that people mostly understand themselves, but sometimes their understanding of meaning is different. The meaning of the natural language expressions is often ambiguous because its logical structure is not identical to the grammatical structure. Explaining the meaning of natural language expressions is dealt by the logic analysis of natural language (LANL), which also allows to discover the mistakes that arise from misinterpretations of the premises.

The most important requirement for a logical system examining the natural language semantics is the expressivity. An extensional logic systems (propositional logic, predicate logic...) are not enough expressive - for LANL needs it is necessary to work with an intensional logical systems. This paper is built on the following research results:

- in [1][2][3] was presented that the most expressive LANL system is a transparent intensional logic (TIL), which we will deal with in this paper,
- TIL has not been fully automated for use in the computer science yet, which is related to its extraordinary expressive force. The solution would be to link its apparatus with another logical system that does not have this deficiency - in [4][5] was presented that a suitable logic could be predicate linear logic (PLL). PLL is not commonly used in LANL, but this paper will describe how it can analyze the natural language dialogues.

In the future research, the goal is to create the formal specification of a new logic linking TIL with PLL. It will be the basis for the implementation of a semantic machine [6] that will be able to automate the logical analysis of natural language. A prerequisite for its implementation is to provide the necessary architectural [7][8][9] and security requirements [10][11][12]. Previously it will be necessary to explore the possibilities of TIL and PLL in formalizing the meaning of sentences from several points of view. The aim of this paper is to compare how mentioned logic can analyze natural language dialogues.

2 Transparent intensional logic

The founder of TIL is Pavel Tichý [13] who (same as Frege) claimed extended Church's simple theory of types (containing a set of individuals and truth values) by modal and temporal parameters. TIL is high-order logic based on the concept of possible worlds. Language of TIL is a modified version of the typed object-oriented lambda calculus with a ramified hierarchy of types.

2.1 Basic objects in TIL

Logical analysis of natural language is related to a specific object base. The basic objects of TIL which represent non-functional element types are:

- set of truth value (true, false) o ,
- set of individuals ι ,
- set of time points (real numbers) τ ,
- set of logically possible worlds ω .

Functional types are sets partial functions $(\alpha\beta_1,\dots,\beta_m)$, with arguments b_1,\dots,b_m of type β_1,\dots,β_m and values in α . Intensions are objects of type $(\alpha\omega)$, i.e. functions from a possible worlds for any type α . Most often, these are functions of type $((\alpha\tau)\omega)$, for simplicity it is stated in TIL as $\alpha_{\tau\omega}$. Some examples of frequently used intensions are shown in Table 1.

2.2 Constructions in TIL

Tichý introduced the concepts of construction [14] as a non-linguistic abstract procedures which expressed the meaning of natural language expressions. The constructions can be:

Table 1. Intensions in TIL

Type intension	Description	Example
$(o\alpha)$	object type α	prime number($\sigma\tau$)
$(o(o\alpha))$	quantifiers for any types α	$\forall \exists$
(ooo)	binary logical operators	$\vee, \wedge, \Leftrightarrow$
(oo)	unary logical operator	\neg
$\iota_{\tau\omega}$	individual roles	president of SR
$(o\alpha)_{\tau\omega}$	properties of type objects α	surprise
$o_{\tau\omega}$	propositions	Peter is looking for a princess.

- variable x , constructs objects based on their valuation,
- trivialization 0X , constructs an object X without changes,
- composition $[X_1 \dots X_m]$, constructs the result of an application of function to arguments,
- closure $\lambda x.X$ (based on λ -abstraction), constructs function.

2.3 Application of TIL to natural language dialogue

Consider the following example:

Example 1. Mother: “If you behave well, you’ll either be playing at home with ball or toy, or going out to the cinema or shopping. If you want to go out, I will decide where to go.”

Son: “I want to go out.”

Mother: “Then we will go to the cinema.”

In TIL, these sentences are analyzed in three steps - type analysis (to each object of sentence is assigned a corresponding type), synthesis (create a construction which represents the meaning of a sentence) a type checking (verify if the synthesis was correctly done).

Let’s analyze the first sentence: “If you (*son*) behave well, you’ll either be (*choose*) playing at *home* with ball or toy, or going *out* to the cinema or shopping.”¹

1. Type analysis:
son/ ι , behave_well/ $(ool)_{\tau\omega}$, \Rightarrow/ooo , choose/ $(ou)_{\tau\omega}$, home/ $(ol)_{\tau\omega}$, \vee/ooo , out/ $(ol)_{\tau\omega}$.
2. Synthesis:
 $\lambda w \lambda t [[{}^0\text{behave_well}_{w t} \Rightarrow [{}^0\text{choose}_{w t} [{}^0\text{home}_{w t} \vee {}^0\text{out}_{w t}]]] {}^0\text{son}]$
3. Type checking: (Fig. 1)

¹ This analysis is simplified. The disjunctions of which the objects *home* and *out* are composed were not analyzed.

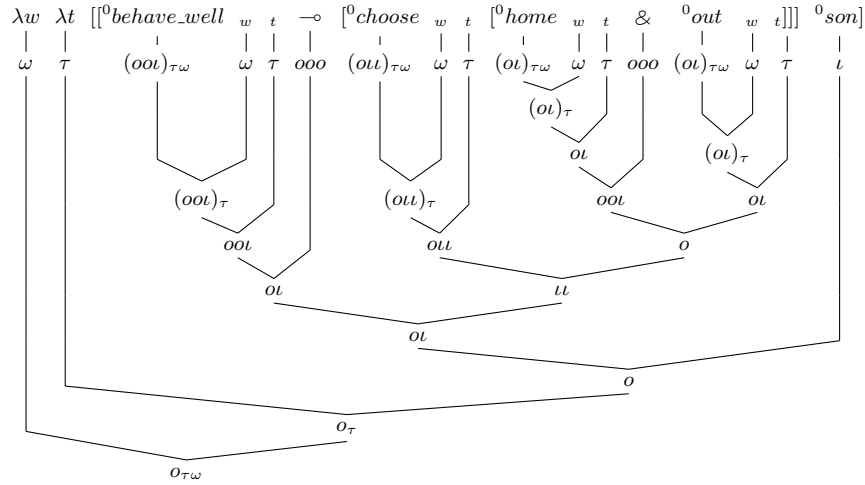


Fig. 1. Type checking of the sentence: “If you behave well, you’ll either be playing at home with ball or toy, or going out to the cinema or shopping.”

Other sentences can be analyzed in the same way. A sentence, the meaning of which is correctly expressed, is always a proposition with type $(o)_{\tau\omega}$ (same as in Fig. 1). From this example it is clear that even though TIL is an effective tool of LANL, the individual sentences of the natural language are analyzed separately, and so it is not adequate for its application on natural language dialogues. In the next section, the same dialogue will be explored through the PLL.

3 Predicate linear logic

Linear logic, introduced in 1987 by J.Y. Girard [17], has the following characteristics:

- can describe real processes in the world, because it works with resources and their consumption,
- the formula of it is an action, a source or a literal,
- introduces new linear logic connectives,
- does not contain the rules of weakness and contraction.

Since a intensions in the natural language expressions can be described by using predicates, we need to introduce PLL. The formula φ in a PLL can be formulated by the following production rule [15] [16]:

$$\varphi ::= p_n \mid 0 \mid 1 \mid \perp \mid \top \mid \varphi_1 \multimap \varphi_2 \mid \varphi_1 \otimes \varphi_2 \mid \varphi_1 \oplus \varphi_2 \mid \varphi_1 \wp \varphi_2 \mid \varphi_1 \& \varphi_2 \mid \varphi^\perp \mid \varphi^? \mid \varphi! \mid P(t, \dots, t) \mid (\forall x)\varphi \mid (\exists x)\varphi,$$

$$t ::= x \mid c \mid f(t, \dots, t),$$

(1)

where $0, 1, \top, \perp$ are logical constants (neutral elements), p_n are atomic statements, $\otimes, \&, \oplus, \wp$ are logical connectives, \multimap is linear implication, $(\cdot)^\perp$ is negation, $!, ?$ are modal operators, $P(t, \dots, t)$ is predicate, \forall, \exists are quantifiers, x is variable, c is constant, and $f(t, \dots, t)$ is the application of a functional symbol to a term t .

3.1 Ludics theory

The Ludics theory [18] is an extension of the PLL by time and space. Ludics is a theory of interaction, which can overcome the difference between grammar and semantics. The basic principle of this calculus is handling positions of PLL formulas, while ignoring their contents.

Time and space in linear logic To change the polarity of the blocks (called clusters) in proof tree means an increment of time. PLL connectives are divided into positive and negative, which also determines the polarity of the PLL formula. There are:

- positive - $\otimes, \oplus, 1, 0, !, \exists$,
- negative - $\&, \wp, \top, \perp, ?, \forall$,
- special - \multimap represents dependent polarity, $(\cdot)^\perp$ causes the flipping of polarity.

The rules of invertibility and focalization are used to make one-step from few follow-up steps in proof tree. In invertibility, a cluster of negative formulas is formed, in focalization a cluster of positive formulas is originated.

Talking about space in Ludics, the truth nor the content of formula, but its location, is essential in the proof tree, known as the ξ . The proof tree that contains only location data is called a design. Immediate subformulas of the A formula are enumerable, while the number of immediate subformulas can be labelled as $B_i, B_{ij}, B_{ijk} \dots$, where i, j, k are positive integers - biases and i, ij, ijk are concatenation of particulate biases i, j, k . If all the data, except the formula addresses, is removed in the sequences used in argument, we call it a pitchfork:

$$\xi \vdash A, \quad (2)$$

where ξ represents one address, and context A is a finite set of addresses. In pitchfork calculus, the following rules are applied:

Axiom daimon

$$\frac{}{\vdash A}^{(\text{a})} \quad (3)$$

Positive and negative rule

$$\frac{\dots \xi * i \vdash A_i \dots}{\vdash A, \xi}^{(+, \vdash, \xi, I)} \quad \frac{\dots \vdash A_I, \xi * I \dots}{\xi \vdash A}^{(-, \xi \vdash N)} \quad (4)$$

3.2 Application of Ludics theory to natural language dialogue

With PLL will be analyzed the same dialogue as with TIL. The PLL formula of the sentence: “If you *behave well* (W), you’ll either be playing at home with *ball* (B) or *toy* (T), or going out to the *cinema* (C) or *shopping*. (S)“ is:

$$W \multimap ((B\&T)\&(C \oplus S))^2 \quad (5)$$

From this PLL formula is possible to create a proof tree, where the following rules of the PLL sequent calculus are used:

$$\frac{}{A \vdash A} (id) \quad \frac{\Gamma \vdash \varphi, \Delta \quad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \& \psi, \Delta} (\&) \quad \frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \multimap B, \Delta} (\multimap) \quad (6)$$

$$\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi^\perp \vdash \Delta} ((\cdot)^\perp) \quad \frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \varphi \oplus \psi, \Delta} (\oplus \ r_1)$$

In proof tree, the change of the polarity is marked with the signs + a -, what creates the polarized tree, shown in Fig. 2.

$$\frac{\frac{\frac{}{\vdash W, W^\perp, B, B^\perp} (id)}{W, W^\perp \vdash B, B^\perp} ((\cdot)^\perp) \quad \frac{\frac{}{\vdash W, W^\perp, T, T^\perp} (id)}{W, W^\perp \vdash T, T^\perp} ((\cdot)^\perp)}{W, W^\perp \vdash (B\&T), \Gamma_3} (\&) \quad \frac{\frac{\frac{}{\vdash W, W^\perp, C, C^\perp} (id)}{\vdash W, W^\perp, C, C^\perp} ((\cdot)^\perp)}{\vdash (C \oplus S), \Gamma_2, W, W^\perp} (\oplus)}{W, W^\perp \vdash (C \oplus S), \Gamma_2} ((\cdot)^\perp)}{W, W^\perp \vdash ((B\&T)\&(C \oplus S)), \Gamma_1} (\&) \quad \frac{}{\vdash W \multimap ((B\&T)\&(C \oplus S)), \Gamma} (\multimap)$$

Fig. 2. Prof tree of the PLL formula with a marked polarity change

In this proof tree are used contexts such as:

- $\Gamma = \{W^\perp, B^\perp, T^\perp, C^\perp, S^\perp\}$,
- $\Gamma_1 = \{B^\perp, T^\perp, C^\perp, S^\perp\}$,
- $\Gamma_2 = \{C^\perp, S^\perp\}$,
- $\Gamma_3 = \{B^\perp, T^\perp\}$.

We modify the proof tree into a reduced tree, shown in Fig.3, by closing the formulas with the same polarity into the clusters. In the reduced tree, the following substitutions and contexts are used:

- $A = W \multimap ((B\&T)\&(C \oplus S))$,
- $A^1 = \{W, B, T, C, S\}$,
- $A^2 = \{\{W\}, \{B\}, \{T\}, \{C\}, \{S\}\}$,
- $D = W \multimap (C \oplus S)$,

² & - the choice depends on the son (outer non-determinism), \oplus - the choice does not depend on the son (inner non-determinism).

$$\frac{\frac{\frac{}{\vdash W, W^\perp, B, B^\perp} (\text{⊗}) \quad \frac{}{\vdash W, W^\perp, T, T^\perp} (\text{⊗}) \quad \frac{\frac{}{W, W^\perp \vdash C, C^\perp} (\text{⊗})}{\vdash (C \oplus S), \Gamma_2, W, W^\perp} (+, \vdash, D, D^1)}{\vdash (C \oplus S), \Gamma_2, W, W^\perp} (-, A, \vdash A^2)}{W, W^\perp \vdash (B \& T) \& (C \oplus S), \Gamma_1} (+, \vdash, A, A^1)}{\vdash A, \Gamma}$$

Fig. 3. Reduced tree showing incrementation of logic time

- $D^1 = \{W, C, S\}$,
- $\Gamma = \{W^\perp, B^\perp, T^\perp, C^\perp, S^\perp\}$,
- $\Gamma_1 = \{B^\perp, T^\perp, C^\perp, S^\perp\}$,
- $\Gamma_2 = \{C^\perp, S^\perp\}$.

A address tree (Fig. 4), in which the addresses are assigned to the formulas and subformulas, represents a logical space.

$$\frac{\frac{\frac{}{\vdash \xi.0.1, \xi.0.2} (\text{⊗}) \quad \frac{}{\vdash \xi.0.1, \xi.0.3} (\text{⊗}) \quad \frac{\frac{\xi.0.4.1 \vdash \xi.0.1} (\text{⊗})}{\vdash \xi.0.1, \xi.0.4} (+, \vdash, \xi.0.4, \{1\})}{\vdash \xi.0.1, \xi.0.4} (-, \xi.0, \vdash \{\{1\}, \{2\}, \{3\}, \{4\}\})}{\xi.0 \vdash} (+, \vdash, \xi, \{0\})}{\vdash \xi}$$

Fig. 4. Address tree showing logic space

From the address tree it is possible to create the dialogue strategies. The mother has three strategies - she always sets out the offer (positive action $(\xi, \{0\})$) and after its execution she is ready to allow her son to play with the ball (negative action $(\xi.0, \{1, 2\})$), or the toy (negative action $(-, \xi.0, \{1, 3\})$), or to go out with him (negative action $(-, \xi.0, \{1, 4\})$). If a son wants to go out, his mother chooses from two other strategies - cinema (positive action $(\xi.0.4, \{1\})$) or shopping (positive action $(\xi.0.4, \{2\})$).

The son has four strategies - he always accepts the mother's offer (negative action $(\xi, \{0\})$) and either wants to play with the ball (positive action $(\xi.0, \{1, 2\})$), or the toy (positive action $(-, \xi.0, \{1, 3\})$), or wants to go out and he is ready for both eventualities (two negative actions $(\xi.0.4, \{1\})$ and $(\xi.0.4, \{1\})$), or is ready only for going to the cinema (negative action $(\xi.0.4, \{2\})$).

The strategies of the dialogue actors make it possible to explore their interactions by using principles from the game semantics. PLL allows to describe natural language dialogues more effectively than TIL.

4 Conclusion

In this paper the possibilities of analyzing natural language dialogues through TIL and PLL logic systems are explored. From the presented results is clear

that each of mentioned logic systems has a certain deficiency - TIL analyzes the sentences separately, whether they are in dialogue or not; PLL is not usually used in LANL, because it does not allow fine-grained semantics analysis of natural language expressions.

In order to achieve the declared goals of the research, it will be necessary to divide it into two parts, where each represents an original contribution in the processing of the natural language. The first phase of the project will be based on existing and already published results in which a transparent intensional logic has been identified as the logical system for LANL with the highest expressive power. Because of several serious disadvantages (the absence of formal definitions of syntax and semantics of the system, the impossibility of specifying a complex deductive calculus, the impossibility of creating a fully automated practical implementation, etc.) an original logical system of source-oriented character will be created. Its ambition will be overcome mentioned disadvantages. Following the first phase of the research, its capabilities will be tested and compared with existing solutions not only through exact logical proofing, but also through its implementation of a semantic machine.

Acknowledgment

This work was supported by the following projects:

- Faculty of Electrical Engineering and Informatics, Technical University of Košice under the contract No. FEI-2018-59: Semantic Machine of Source-Oriented Transparent Intensional Logic.
- Slovak Research and Development Agency under the contract No. SK-AT-2017-0012: Semantics technologies for computer science education.

References

1. Bilanová, Z.: Applications of Transparent Intensional Logic and Montague Intensional Logic on Natural Language Sentences: A Review. *SCYR 2017. - Košice : TU*, 32–35 (2017).
2. Vokorokos, L., Bilanová, Z., Mihályi, D.: Linear logic operators in transparent intensional logic. *Informatics 2017. - Danvers : IEEE*, 420–424 (2017).
3. Bilanová, Z., Uchnár, M.: Comparison of the approaches of Montague and Tichý within a logical analysis of an English sentence. *POSTER 2017. - Prague : Czech Technical University*, 1–6 (2017).
4. Bilanová, Z., Dankovičová, Z.: From Extensional Principles in the Logical Analysis of Natural Language to Intensional Principles. *POSTER 2018. - Praha : ČVUT*, 1–4 (2018).
5. Bilanová, Z.: Increasing the expressive power of the intensional logic used for logical analysis of the natural language sentences. *SCYR 2018. - Košice : TU*, 128–129 (2017).
6. Vokorokos, L., Pekár, A., Ádam, N., Daranyi, P: Yet Another Attempt in User Authentication. *Acta Polytechnica Hungarica*, 10(3), 37–50 (2013).

7. Juhár, J., Vokorokos, L.: Separation of concerns and concern granularity in source code. *Informatics 2015. - Danvers : IEEE*, 139–144 (2015).
8. Chovanec, M., Chovancová, E., Dufala, M.: DIDS based on hybrid detection. *ICETA 2014 : 12th IEEE International Conference on Emerging eLearning Technologies and Applications*, 79–83 (2014).
9. Chovancová, E., Ádam, N., Baláz, A., Pietriková, E., Feciľak, P., Šimoňák, S., Chovanec, M.: Securing distributed computer systems using an advanced sophisticated hybrid honeypot technology. *Computing and Informatics*, 36(1), 113–139 (2017).
10. Ádam, N.: Single input operators of the DF KPI system. *Acta Polytechnica Hungarica*, 7(1), 73–86 (2010).
11. Vokorokos, L., Baláz, A., Madovs, B.: Application Security through Sandbox Virtualization. *Acta Polytechnica Hungarica*, 12(1), 83–101 (2015).
12. Vokorokos, L., Madovs, B., Ádam, N., Baláz, A.: Innovative Operating Memory Architecture for Computers using the Data Driven Computation Model. *Acta Polytechnica Hungarica*, 10(5), 63–79 (2013).
13. Tichý, P.: The Foundations of Frege’s Logic. *Berlin and New York: De Gruyter*, 1988.
14. Materna, P.: Denotation and reference. *Organon F: Medzinarodny Casopis Pre Analyticku Filozofiu*, 17(1), 3–20 (2010).
15. Vokorokos, L., Bilanová, Z., Dankovičová, Z.: Logic programming from the point of view of type theory and predicate linear logic. *International Journal of Innovative Science and Research Technology*, 3(7), 756–760 (2018).
16. Perháč, J.: Modeling Component Composition’s Synchronization Problems by Petri Nets and Linear Logic. *SCYR 2018. - Košice : TU*, 159–160 (2018).
17. Girard, J. Y.: Locus Solum: From the rules of logic to the logic of rules. *Journal Mathematical Structures in Computer Science*, 11(3), 301–506 (2001).
18. Fleury M.-R., Quatrini, M., Tronçon, S.: Dialogues in Ludics. *PRELUDE Project - 2006-2009. Revised Selected Papers, Logic and Grammar*, 138–157 (2011).