TIL as the Logic of Communication in a Multi-Agent System

Marie Duží

VSB-Technical University Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic
marie.duzi@vsb.cz

Abstract. The method of encoding communication of agents in a multi-agent system (MAS) is described. The autonomous agents communicate with each other by exchanging messages formulated in a near-to-natural language. Transparent Intensional Logic (TIL) is an expressive system primarily designed for the logical analysis of natural language; thus we make use of TIL as a tool for encoding the semantic content of messages. The hyper-intensional features of TIL analysis are described, in particular with respect to agents' attitudes and anaphoric references. We demonstrate the power of TIL to determine the antecedent of an anaphoric pronoun. By an example of a simple dialogue we illustrate the way TIL can function as a dynamic logic of discourse where anaphoric pronouns refer to entities of any type, even constructions, i.e. the structured meanings of other expressions.

1 Introduction

Multi-agent system (MAS) is a system composed of autonomous, intelligent but resource-bounded agents. The agents are active in their perceiving environment and acting in order to achieve their individual as well as collective goals. They communicate with each other by exchanging messages formulated in a standardised natural language. According to the FIPA standards[1], *message* is the basic unit of communication. It can be of an arbitrary form but it is supposed to have a structure containing several attributes. Message semantic *content* is one of these attributes, the other being for instance 'Performatives', like 'Query', 'Inform', 'Request' or 'Reply'. The content can be encoded in any suitable language. The standards like FIPA SL and KIF are mostly based on the First-Order Logic (FOL) paradigm, enriched with higher-order constructs wherever needed.[2] The enrichments extending FOL are well defined syntactically, while their semantics is often rather sketchy, which may lead to communication inconsistencies. Moreover, the bottom-up development from FOL to more complicated cases yields the versions that do not fully meet the needs of the MAS communication. In particular, agents' attitudes and anaphora processing create a problem. In the paper we are going to demonstrate the need for an expressive logical tool of Transparent Intensional Logic (TIL) in order to encode the semantic content of

[1] The Foundation for Intelligent Physical Agents, http://www.fipa.org

[2] For details on FIPA SL, see http://www.fipa.org/specs/fipa00008/; for KIF, Knowledge Interchange Format, see http://www-ksl.stanford.edu/knowledge-sharing/kif/

© A. Gelbukh (Ed.) Received 18/10/07
Advances in Natural Language Processing and Applications Accepted 07/12/07
Research in Computing Science 33, 2008, pp. 27-20 Final Version 12/01/08

messages in a near-to-natural language. Using TIL, the human-computer interface and communication is designed in a smooth way.

Transparent Intensional Logic (TIL)[3] is a system with procedural semantics primarily designed for the logical analysis of natural language. Traditional non-procedural theories of formal semantics are less or more powerful logical languages, from the extensional languages based on FOL approach, through some hybrid systems up to intensional (modal or epistemic) logics. Particular systems are suited well to analysing restricted sublanguages, and they are broadly used and well standardised. However, the logic of attitudes is a stumbling block for all of them. Moreover, if such a great variety of specialised languages were involved in order to design a communication language for a multi-agent system (MAS), the agents would have to keep switching from one logical language to another, which is certainly not a plausible solution.

On the other hand, TIL, due to its strong typing and procedural semantics, operates smoothly with the three levels of granularity: the extensional level of truth-functional connectives, the intensional level of modalities and finally the hyperintensional level of attitudes. The sense of a sentence is an algorithmically structured *construction* of a proposition denoted by the sentence. The denoted proposition is a flat mapping with the domain of possible worlds. Our motive for working 'top-down' has to do with anti-contextualism: any given unambiguous term or expression (even one involving indexicals or anaphoric pronouns) expresses the same construction as its sense (meaning) in whatever sort of context the term or expression is embedded within. And the meaning of an expression determines the respective denoted entity (if any), but not vice versa.

When assigning a construction to an expression as its meaning, we specify *procedural know-how*, which must not be confused with the respective *performancy know-how*.[4] Understanding a sentence *S* involves procedural know-how; one can spell out instructions for evaluating the truth-conditions of *S* in any state-of-affairs *w* at any time *t*. But, of course, one can know how to evaluate *S* without actually being able to do so—that is, without having the performatory skills that enable him to determine the truth-value of *S* in a particular state-of-affairs *W* at time *T*.

The paper is organised as follows. After briefly introducing TIL philosophy and its basic notions in Section 2, the following Section 3 describes the method of analysing sentences with anaphoric references occurring in any context; extensional, intensional, or even hyperintensional context of attitudes. By way of an example we demonstrate in Section 4 how TIL functions as the logic of dynamic discourse. Finally, a few notes on TIL implementation by the TIL-Script language are contained in concluding Section 5.

[3] See, for instance, [5], [10] and [11].
[4] See [7], pp.6-7.

2 Basic Principles of TIL

TIL *constructions* are uniquely assigned to expressions as their algorithmically structured meanings. Intuitively, construction is a procedure (a generalised algorithm), that consists of particular sub-constructions. It is an instruction on how to proceed in order to obtain the output entity given some input entities. Atomic constructions (*Variables* and *Trivializations*) do not contain any other constituent but itself; they supply objects (of any type) on which compound constructions operate. *Variables x, y, p, q, ...,* construct objects dependently on a valuation; they *v*-construct. *Trivialisation* of an object *X* (of any type, even a construction), in symbols $\mathbf{{}^0}\boldsymbol{X}$, constructs simply *X* without the mediation of any other construction. *Compound* constructions, which consist of other constituents, are *Composition* and *Closure. Composition* $\mathbf{[}\boldsymbol{F\ A_1 \ldots A_n}\mathbf{]}$ is the instruction to apply a function *f* (*v*-constructed by *F*) to an argument A (*v*-constructed by $A_1 \ldots A_n$).[5] Thus it *v*-constructs the value of *f* at A, if the function *f* is defined at A, otherwise the Composition is *v*-improper, i.e., it does not *v*-construct anything. *Closure* $\mathbf{[\lambda}\boldsymbol{x_1 \ldots x_n\ X}\mathbf{]}$ is the instruction to *v*-construct a function by abstracting over variables $x_1,\ldots,x_n$ in the ordinary manner of λ-calculi. Finally, higher-order constructions can be used twice over as constituents of composed constructions. This is achieved by a fifth construction called *Double Execution*, $\mathbf{{}^2}\boldsymbol{X}$**,** that behaves as follows: If *X* *v*-constructs a construction *X*', and *X'* *v*-constructs an entity *Y*, then ${}^2X$ *v*-constructs *Y*; otherwise ${}^2X$ is *v*-improper.

TIL constructions, as well as the entities they construct, all receive a type. The formal ontology of TIL is bi-dimensional; one dimension is made up of constructions, the other dimension encompasses non-constructions. On the ground level of the type-hierarchy, there are non-constructional entities unstructured from the algorithmic point of view belonging to a ***type of order 1***. Given a so-called *epistemic (or 'objectual')* ***base*** of ***atomic types*** (ο-truth values, ι-individuals, τ-time moments / real numbers, ω-possible worlds), the induction rule for forming functions is applied: where α, $\beta_1,\ldots,\beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times\ldots\times \beta_n$ to α, denoted $\boldsymbol{(\alpha\ \beta_1 \ldots \beta_n)}$, is a type of order 1 as well.[6] Constructions that construct entities of order 1 are ***constructions of order 1***. They belong to a ***type of order 2***, denoted by $*_1$. This type $*_1$ together with atomic types of order 1 serves as a base for the induction rule: any collection of partial mappings, type $\boldsymbol{(\alpha\ \beta_1 \ldots \beta_n)}$, *involving* $*_1$ in their domain or range is a *type of order 2*. Constructions belonging to a type $*_2$ that identify entities of order 1 or 2, and partial mappings involving such constructions, belong to a ***type of order 3***. And so on *ad infinitum*.

An object A of a type α is called an α-object, denoted A/α. That a construction C *v*-constructs an α-object is denoted $C \rightarrow_v \alpha$.

(α-)intensions are members of a type (αω), i.e., functions from possible worlds to the arbitrary type α. *(α-)extensions* are members of the type α, where α is not equal to (βω) for any β, i.e., extensions are not functions from possible worlds. Intensions

[5] We treat functions as mappings, i.e., set-theoretical objects, unlike the *constructions* of functions.

[6] TIL is an open-ended system. The above epistemic base {ο, ι, τ, ω} was chosen, because it is apt for natural-language analysis, but the choice of base depends on the area to be analysed.

are frequently functions of a type $((\alpha\tau)\omega)$, i.e., functions from possible worlds to *chronologies* of the type α (in symbols: $\alpha_{\tau\omega}$), where a chronology is a function of type $(\alpha\tau)$. We use variables $w, w_1, \ldots$ as v-constructing elements of type ω (possible worlds), and $t, t_1, \ldots$ as v-constructing elements of type τ (times). If $C \rightarrow \alpha_{\tau\omega}$ v-constructs an α-intension, the frequently used Composition of the form $[[Cw]t]$, the intensional descent of the α-intension, is abbreviated as C_{wt}.

Some important kinds of intensions are:

Propositions, type $o_{\tau\omega}$. They are denoted by empirical (declarative) sentences.

Properties of members of a type α, or simply *α-properties,* type $(o\alpha)_{\tau\omega}$. General terms (some substantives, intransitive verbs) denote properties, mostly of individuals.

Relations-in-intension, type $(o\beta_1\ldots\beta_m)_{\tau\omega}$. For example transitive empirical verbs, also attitudinal verbs denote these relations.

α-roles, offices, type $\alpha_{\tau\omega}$, where $\alpha \neq (o\beta)$. Frequently $\iota_{\tau\omega}$. Often denoted by concatenation of a superlative and a noun ("*the highest mountain"*).

Example: We are going to analyse the sentence "Adam is looking for a parking place". Our method of analysis consists of three steps:

1) *Type-theoretical analysis*, i.e., assigning types to the objects talked about by the analysed sentence. In our case we have:
   a) *Adam*/ι;
   b) *Look_for*/$(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$—the relation-in-intension of an individual to a property of individuals: the seeker wants to find an instance of the property;
   c) *Parking(Place)*/$(o\iota)_{\tau\omega}$—the property of individuals.

2) *Synthesis*, i.e., composing the constructions of the objects *ad* (1) in order to construct the proposition of type $o_{\tau\omega}$ denoted by the whole sentence. The sentence claims that the individual Adam has the 'seeking-property' of looking for a parking place. Thus we have to construct the individual Adam, the 'seeking-property', and then apply the latter to the former. Here is how:
   a) The atomic construction of the individual called Adam is simply ^0Adam;
   b) The 'seeking-property' has to be constructed by Composing the relation-in-intension *Look_for* with a seeker $x \rightarrow \iota$ and the property *Parking*/$(o\iota)_{\tau\omega}$ an instance of which is being sought. But the relation-in-intension cannot be applied directly to its arguments. It has to be extensionalized first: $[[^0Look\_for\ w]\ t]$, abbreviated as $^0Look\_for_{wt}$. Thus we have $[^0Look\_for_{wt}\ x\ ^0Parking]$ v-constructing a truth value. Abstracting first from x by $\lambda x\ [^0Look\_for_{wt}\ x\ ^0Parking]$ we obtain the class of individuals; abstracting from w and t we obtain the 'seeking-property':
      $\lambda w \lambda t\ [\lambda x\ [^0Look\_for_{wt}\ x\ ^0Parking]]$.
   c) Now we have to Compose the property constructed *ad* (b) with the individual constructed *ad* (a). The property has to be extensionalised first, i.e., $[\lambda w \lambda t\ [\lambda x\ [^0Look\_for_{wt}\ x\ ^0Parking]]]_{wt}$, and then Composed with the former.[7] Since we are going to construct a proposition, i.e., an intension, we finally have to abstract from w, t:
      $\lambda w \lambda t\ [[\lambda w \lambda t\ [\lambda x\ [^0Look\_for_{wt}\ x\ ^0Parking]]]_{wt}\ ^0Adam]$.

[7] For details on predication of properties of individuals, see [3].

This construction is the literal analysis of our sentence. It can still be β-reduced to the equivalent form:

$$\lambda w\lambda t\ [{}^0Look\_for_{wt}\ {}^0Adam\ {}^0Parking].$$

3) *Type-Theoretical checking*:

$$\begin{array}{ll} \lambda w\lambda t\ [{}^0Look\_for_{wt} \quad {}^0Adam \quad {}^0Parking] & \\ \quad \underbrace{(o\iota(o\iota)_{\tau\omega}) \qquad \iota \qquad (o\iota)_{\tau\omega}}_{o} & \\ o_{\tau\omega} & \end{array}$$

The role of Trivialisation and empirical parameters $w \to \omega$, $t \to \tau$ in the communication between agents can be elucidated as follows. Each agent has to be equipped with a basic ontology, namely the set of primitive concepts she is informed about. Thus the upper index '^{0}' serves as a marker of the primitive concept that the agents should have in their ontology. If they do not, they have to learn them by asking the others. The lower index '$_{wt}$' can be understood as an instruction to execute an *empirical inquiry (search)* in order to obtain the actual current value of an intension, for instance by searching agent's database or by asking the other agents, or even by means of agent's sense perception.

3 Anaphora and Meaning

The problem of an anaphoric reference to a previously used expression is a well-known hard nut of semantic analysis, because the antecedent of the anaphoric reference is often not unambiguously determined. Thus it is often said that anaphora constitutes a pragmatic problem rather than a problem of logical semantics. We agree that *logical* analysis cannot disambiguate any sentence, because it presupposes understanding and full linguistic competence. Yet our method of logical analysis can contribute to solving the problem of disambiguation in at least two respects; (a) a type-theoretical analysis often unambiguously determines which of the possible meanings is used, and (b) if there are two or more possible readings of a sentence, the logical analysis should make all of them explicit. This often concerns the distinction between *de dicto* and *de re* readings.

In this section we propose the method of *logically* analysing sentences with anaphoric references. The method consists in substituting an appropriate construction of the object to which the anaphora refers for the anaphoric variable. In other words, we perform a *semantic* pre-processing of the embedded anaphoric clause based on the *meaning* of the respective antecedent. In this sense anaphora *is* a *semantic* problem. Moreover, we are going to show that TIL strong typing often unambiguously determines the respective antecedent.

3.1 Semantic Pre-Processing of Anaphoric References

Our hyperintensional (procedural) semantics makes it possible to apply anti-contextualist and compositional analysis to anaphoric sentences. The meaning of a sentence containing a clause with an anaphoric reference is the procedure which is a two-phase instruction that comes down to this:

(i) *execute the substitution based on the meaning of the antecedent for the anaphoric variable;*

(ii) *execute the result (a propositional construction) again to obtain a proposition.*

To specify phase (i) we make use of the fact that constructions are objects *sui generis* that the other constructions can operate on. The substitution is realised by a function $Sub/(*_n *_n *_n *_n)$ that operates on constructions C_1, C_2 and C_3 yielding as output the construction C_4 that is the result of substituting C_1 for C_2 in C_3. The phase (ii) consists in executing the adjusted meaning, namely the construction pre-processed by phase (i). To this end we use the fifth construction defined above, the *Double Execution.* The method is uniquely applicable to all kinds of sentences, including those that express (*de dicto* / *de re*) attitudes to a hyperintension, attitudes to an intension, and relations (-in-intension) to extensional entities. Now we adduce examples that illustrate the method.

(A) "5 + 7 = 12, and Charles knows it."

The embedded clause "Charles knows it" does not express Charles' relation(-in-intension) to a truth-value, but to a *construction*, here the *procedure* of calculating the result of *5 + 7 = 12*. Hence $Know(ing)/(o\iota *_1)_{\tau\omega}$ is a relation-in-intension of an individual to a construction. However, the meaning of the clause is incomplete; it is an *open* construction with the free variable *it*: $\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ it]$. The variable $it/*_2 \rightarrow *_1$ is the meaning of the pronoun 'it' that in (A) anaphorically refers to the meaning of "5 + 7 = 12", i.e., the construction $[{}^0+\ {}^05\ {}^07]$. The meaning of the whole sentence (A) is, however, complete. It is the *closed* construction

$$\text{(A')}\quad \lambda w \lambda t\ [[{}^0{=}\ [{}^0+\ {}^05\ {}^07]\ {}^012] \wedge {}^{\mathbf{2}}\mathbf{[}{}^0Sub\ {}^{00}[{}^0{=}\ [{}^0+\ {}^05\ {}^07]\ {}^012]\ {}^0it\ {}^{\mathbf{0}}\mathbf{[}\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ it]\mathbf{]]}_{wt}]$$

Types: $Charles/\iota$; $Know/(o\iota *_1)_{\tau\omega}$; $Sub/(*_2 *_2 *_2 *_2)$; $it/*_2 \rightarrow *_1$; the other types are obvious.

Since (A') seems to be rather complicated, we now show that (A') is an adequate analysis meeting our three requirements of compositionality, anti-contextualism and a purely semantic solution. The argument of the second conjunct of (A'), namely

$$\text{(S)}\quad \mathbf{[}{}^0Sub\ {}^{00}[{}^0{=}\ [{}^0+\ {}^05\ {}^07]\ {}^012]\ {}^0it\ {}^{\mathbf{0}}\mathbf{[}\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ it]\mathbf{]]}_{wt}] \rightarrow *_1$$

constructs a *construction* of order 1, namely the one obtained by the substitution of the construction ${}^0[{}^0{=}\ [{}^0+\ {}^05\ {}^07]\ {}^012]$ for the variable *it* into the construction $\mathbf{[}\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ it]\mathbf{]}$. The result is the *construction*

$$\text{(S')}\quad \mathbf{[}\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ {}^0[{}^0{=}\ [{}^0+\ {}^05\ {}^07]\ {}^012]]\mathbf{]},$$

which constructs a proposition P. But an argument of the truth-value function conjunction ($\wedge$) can be neither a propositional construction, nor a proposition, but must be a truth-value. Since (S) constructs the construction (S'), and (S') constructs the proposition P, the execution steps have to be: (a) execute (S) to obtain the propositional construction (S'), (b) execute the result (S') to obtain the proposition P; hence we need the Double Execution of (S) to construct the proposition P, and then (c) P has to undergo intensional descent with respect to the external w, t in order to v-construct a truth-value.

Note that the open construction $\lambda w \lambda t\ [{}^0Know_{wt}\ {}^0Charles\ it]$ is assigned to "Charles knows *it*" invariably of a context. The variable *it* is free here either for a pragmatic valuation or for a substitution by means of the meaning of the antecedent that is referred to in a linguistic context. The object—*what* is known by Charles—can be completed by a situation of utterance or by a linguistic context. If the sentence occurs within another linguistic context, then *Sub* substitutes a different construction for the variable *it*.

The other example concerns Charles' attitude of seeking the occupant of an individual office:

(B) "Charles sought the Mayor of Dunedin but *he* did not find *him*."

Suppose now the *de dicto* reading of (B), i.e., that Charles' search concerned the office of Mayor of Dunedin and not the location of its holder. The function *Sub* creates a new construction from constructions and, so, can easily be iterated. The analysis of (B) is:

$$(\mathrm{B}^d) \quad \lambda w \lambda t\ [[{}^0Seek_{wt}\ {}^0Ch\ \lambda w \lambda t\ [{}^0Mayor\_of_{wt}\ {}^0D]] \wedge {}^{\mathbf{2}}\mathbf{[}{}^0Sub\ {}^{00}Ch\ {}^0he\ \mathbf{[}{}^0Sub\ {}^0[\lambda w \lambda t\ [{}^0Mayor\_of_{wt}\ {}^0D]]\ {}^0him\ {}^{\mathbf{0}}[\lambda w \lambda t\ \neg[{}^0Find_{wt}\ he\ him]]\mathbf{]]}_{\mathbf{wt}}].$$

Types: $Seek/(o\iota\iota_{\tau\omega})_{\tau\omega}$; $Find/(o\iota\iota_{\tau\omega})_{\tau\omega}$; $Ch(arles)/\iota$; $Mayor\_of$ (something)$/(\iota\iota)_{\tau\omega}$; D(unedin)$/\iota$; $he/*_1 \rightarrow \iota$; $him/*_1 \rightarrow \iota_{\tau\omega}$.[8]

Again, the meaning of (B) is the closed construction (B^d), and the meaning of the embedded clause "*he* did not find *him*" is the open construction[9] $\lambda w \lambda t\ \neg[{}^0Find_{wt}\ he\ him]$ with the two free variables *he* and *him*. Note that since $he \rightarrow \iota$ and $him \rightarrow \iota_{\tau\omega}$, the arguments of *Sub* function are unambiguously type-determined. The only construction of an individual to be substituted for *he* is here ${}^0Ch$; and the only construction of an individual office to be substituted for *him* is the construction of the Mayor office, namely $[\lambda w \lambda t\ [{}^0Mayor\_of_{wt}\ {}^0D]]$.

Of course, another refinement is thinkable. The variables *he* and *him*, ranging over individuals and individual offices, respectively, reduce the ambiguity of 'find' by determining that here we are dealing with finding the occupant of an individual office. But the pronouns like 'he', 'him', or 'she', 'her' also indicate that the finder as well as the occupant of the sought office are male and female, respectively. Thus, e.g., a refined meaning of "He found her" would be

[8] We often use the infix notation without Trivialisation when using constructions of truth-value functions $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication), all of the type (ooo), and negation ($\neg$) of the type (oo).

[9] Tenses are disregarded.

$\lambda w\lambda t\ [[^0Find_{wt}\ he\ her] \wedge [^0Male_{wt}\ he] \wedge [^0Female_{wt}\ her_{wt}]]$.

Additional types: *Male*, *Female*/$(o\iota)_{\tau\omega}$; *her*/$*_1 \rightarrow \iota_{\tau\omega}$.

Now perhaps a more natural *de re* reading of 'seeking sentences' like

(B^r) "Charles is looking for the Mayor of Dunedin (namely the location of him)"

is understood as uttered in a situation where Charles knows who the Mayor is, and is striving to locate this individual. Unlike the *de dicto* case, the sentence understood *de re* has an *existential presupposition*: in order that (B^r) have *any* truth value, the Mayor has to exist. Thus we must not substitute the construction of an office, but of the individual (if any) that occupies the office. To this end we use $[^0Tr\ [^0Mayor\_of_{wt}\ ^0D]]$ that fails to construct anything if $[^0Mayor\_of_{wt}\ ^0D]$ is v-improper (the Mayor does not exist), otherwise it v-constructs the Trivialisation of the occupant of the office. Using the technique of substitutions we can discover the adequate analysis of (B^r):

$$\lambda w\lambda t\ [^0Look_{wt}\ ^0Ch\ ^2[^0Sub\ [^0Tr\ [^0Mayor\_of_{wt}\ ^0D]]\ ^0him\ ^0[\lambda w\lambda t\ [^0Loc\_of_{wt}\ him]]]]$$

Additional types: *Look(_for)*/$(o\iota\mu_{\tau\omega})_{\tau\omega}$; *Tr*/$(*_1\iota)$; *him* /$*_1 \rightarrow \iota$; *Loc_of*/$(\mu\iota)_{\tau\omega}$.[10]

3.2 Donkey Sentences

The following example is a variant of the well-known problem of Peter Geach's *donkey sentence*s:

(D) "If somebody has got a new car then *he* often washes *it*."

The analysis of the embedded clause "*he* often washes *it*" containing the anaphoric pronouns 'he' and 'it' is again an open construction with two free variables *he*—*who* (washes), *it* —*what* (is washed), *he*, *it* $\rightarrow \iota$; *Wash*/$(o\iota\iota)_{\tau\omega}$:

$\lambda w\lambda t\ [^0Wash_{wt}\ he\ it]$.[11]

The problem of donkey sentences consists in discovering their logical form, because it is not clear how to understand them. Geach in [1], p.126, proposes a structure that can be rendered in 1st-order predicate logic as follows (*NC* new car):

$$\forall x \forall y\ ((NC(y) \wedge Has(x, y)) \rightarrow Wash(x, y)).$$

However, Russell objected to this analysis that the expression 'a new car' is an *indefinite description*, which is not rendered by Geach's analysis. Hence Russell proposed the analysis that corresponds to this formula of 1st-order predicate logic:

$$\forall x\ (\exists y\ (NC(y) \wedge Has(x, y)) \rightarrow Wash(x, \mathbf{y})).$$

---

[10] The type μ is the type of a location/position.

[11] If we also want to analyze the frequency of washing, i.e., the meaning of 'often', then we use the function *Freq*(*uently*)/$((o(o\tau))\tau)$. The function *Freq* associates each time T with a set of those time intervals (of type $(o(o\tau))$) that are frequent in T (for instance, once a week). The analysis of "*he* often washes *it*" is then $\lambda w\lambda t\ [^0Freq_t\ \lambda t'[^0Wash_{wt'}\ he\ it]]$.

But the last occurrence of the variable y (marked in bold) is free in this formula—out of the scope of the existential quantifier supposed to bind it.

Neale in [6] proposes a solution that combines both of the above proposals. On the one hand, the existential character of an indefinite description is saved (Russell's demand), and on the other hand, the anaphoric variable is bound by a general quantifier (Geach's solution). Neale introduces so-called *restricted quantifiers*:

[every *x*: man *x* and [a *y*: new-car *y*](*x* owns *y*)]
([whe *z*: car *z* and *x* owns *z*] (*x* often washes *z*)).[12]

The sentence (D) does not entail that if the man owns more than one new car then some of these cars are not washed by him. Hence we can reformulate the sentence into

(D_1) "Anybody who owns some new cars often washes *all of them* [each of the new cars he owns]."

However, the following sentence (D_2) means something else:

(D_2) "Anybody who owns some new cars often washes *some of them* [some of the new cars he owns]."

The TIL analysis of (D_1), which in principle corresponds to Geach's proposal, is

(D_1') $\lambda w\lambda t\ [{}^0\forall\lambda x\ [{}^0\forall\lambda y\ [[[{}^0NC_{wt}\, y] \wedge [{}^0Own_{wt}\, x\, y]] \supset {}^2\mathbf{[}{}^0Sub\ {}^0x\ {}^0he\ \mathbf{[}{}^0Sub\ {}^0y\ {}^0it\ {}^0[\lambda w\lambda t\ [{}^0\mathrm{Wash}_{wt}\ he\ it]]]\mathbf{]}_{wt}]$.

Types: $Own/(o\iota\iota)_{\tau\omega}$; $Wash/(o\iota\iota)_{\tau\omega}$; NC (being a new car)$/(o\iota)_{\tau\omega}$; $x, y, he, it \rightarrow \iota$; $\forall/(o(o\iota))$—the general quantifier: $[{}^0\forall^\alpha\ \lambda xA]$ *v*-constructs True iff $[\lambda xA]$ *v*-constructs the whole type ι, otherwise False.

But then an objection due to Neale can be levelled against these analyses, namely that in the original sentence (D) the anaphoric pronoun 'it' stands *outside* of the scope of the quantifier occurring in the antecedent. To overcome this objection, we use a different type of quantifier. Apart the common quantifiers $\forall,\exists/(o(o\iota))$ that operate on sets of individuals, we use quantifiers of another type, namely *Some* and $All/((o(o\iota))(o\iota))$. *Some* is a function that associates the argument—a set S—with the set of all those sets which have a non-empty intersection with S. *All* is a function that associates the argument—a set S—with the set of all those sets which contain S as a subset. For instance the sentence "Some students are happy" is analyzed by

$\lambda w\lambda t\ [[{}^0Some\ {}^0Student_{wt}]\ {}^0Happy_{wt}]$.

The analyses of the embedded clauses of (D_1), (D_2), namely "*he* washes all of *them*", "*he* washes some of *them*" are (the anaphoric pronoun '*them*' refers here to the *set of individuals*; we use the variable $them \rightarrow (o\iota)$ as the meaning of 'them')

$\lambda w\lambda t\ [[{}^0All\ them]\ \lambda it\ [{}^0Wash_{wt}\ he\ it]]$, $\lambda w\lambda t\ [[{}^0Some\ them]\ \lambda it\ [{}^0Wash_{wt}\ he\ it]]$

[12] Neale in [6], p. 236, takes into account that the sentence is true even if a man owns *more than one* new car. To avoid singularity he thus claims that the description used in his analysis does not have to be singular (definite) but plural: his abbreviation 'whe *F*' stands for 'the *F* or the *F*s'.

respectively. Now we need to substitute a construction of the set of new cars owned by the man for the variable *them.* Further, we have to substitute the variable x (‘anybody’) for the variable *he* (‘*who* washes’), and then the pre-processed construction has to be Double Executed. To prevent collision of variables, we rename the internal variables w, t.

(D$_1$’’) $\lambda w\lambda t\ [{}^0\forall\lambda x\ [[[{}^0Man_{wt}\ x] \wedge [{}^0\exists\lambda y\ [[{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]] \supset {}^2[{}^0Sub\ {}^0[\lambda y\ [[{}^0NC_{wt}\,y] \wedge [{}^0Own_{wt}\ x\ y]]]\ {}^0them\ [{}^0Sub\ {}^0x\ {}^0he\ {}^0[\lambda w'\lambda t'\ [[{}^0All\ them]\ \lambda it\,[{}^0Wash_{w't'}\ he\ it]]]]]_{wt}]]$.

Gloss: “For every man, if the man owns some new cars then all of them [i.e., the new cars owned] are washed by him [the man x].”

This construction can be viewed as the most adequate analysis of (D$_1$), because it meets Russell’s requirement of an indefinite description in the antecedent, while the scope of $\exists$ does not exceed the antecedent.

The second possible reading of (D) is now analyzed using *Some* instead of *All*:

(D$_2$’’) $\lambda w\lambda t\ [{}^0\forall\lambda x\ [[[{}^0Man_{wt}\ x] \wedge [{}^0\exists\lambda y\ [[{}^0NC_{wt}\ y] \wedge [{}^0Own_{wt}\ x\ y]]]] \supset {}^2[{}^0Sub\ {}^0[\lambda y\ [[{}^0NC_{wt}\,y] \wedge [{}^0Own_{wt}\ x\ y]]]\ {}^0them\ [{}^0Sub\ {}^0x\ {}^0he\ {}^0[\lambda w'\lambda t'\ [[{}^0Some\ them]\ \lambda it\,[{}^0Wash_{w't'}\ he\ it]]]]]_{wt}]]$.

Gloss: “For every man, if the man owns some new cars then some of them [i.e., the new cars owned] are washed by him [the man x].”

As we pointed out above, it is not clear how to exactly understand the sentence (D), simply because the sentence is ambiguous. We thus offered analyses that disambiguate it. Whether these readings are the only possible ones is not for us to decide. In our opinion the reading (D$_1$) is more plausible, and Neale takes into account only this one. However, our method makes it possible to easily analyse particular variants of donkey sentences like “… none of them …”, “… most of them…”, and suchlike. It might be objected, however, that in the interest of disambiguation, we actually analysed two variants of the original sentence.

Sandu formulates in [8] two principles that every compositional procedure for analysing natural language sentences should obey:

(a) there is a one-to-one mapping of the surface structure of a sentence of (a fragment of) English into its logical form which preserves the left-to-right ordering of the logical constants
(b) the mapping preserves the nature of the lexical properties of the logical constants, in the sense that an indefinite is translated by an existential quantifier, etc.

One can see that our analyses (D$_1$’’) and (D$_2$’’) obey these principles with respect to the glossed variants, but not with respect to the original sentence (D). Regardless of the disambiguation concerning some/all new cars being washed, principle (b) is violated because ‘a man’ is analysed as ‘every man’. To put our arguments on a still more solid ground, we now propose the literal analysis of the sentence (D). The analysis of the clause “A man has a new car” is as follows:

(NC) $\lambda w\lambda t\ [{}^0\exists\lambda xy\ [[{}^0Man_{wt}\,x] \wedge [{}^0NC_{wt}\,y] \wedge [{}^0Own_{wt}\ x\ y]]]$.

Additional type: $\exists/(o(o\iota\iota))$.

The consequent of (D) expresses that *all* the couples <*he*, *it*> are such that *he Washes it.* Using a variable *couples*/$*_1 \rightarrow (o\iota\iota)$, and quantifier $All^c/((o(o\iota\iota))(o\iota\iota))$, we have:

$$\lambda w \lambda t\ [[{}^0All^c\ couples]\ \lambda he\ it\ [{}^0Wash_{wt}\ he\ it]].$$

Now composing (NC) with the latter, we substitute the construction of the set of couples constructed by the Closure of (NC) for the variable *couples*:

(D') $\lambda w \lambda t\ [[{}^0\exists \lambda xy\ [[{}^0Man_{wt}\,x] \wedge [{}^0NC_{wt}\,y] \wedge [{}^0Own_{wt}\ x\ y]]] \supset$
${}^2[{}^0Sub\ {}^0[\lambda xy\ [[{}^0Man_{wt}\,x] \wedge [{}^0NC_{wt}\,y] \wedge [{}^0Own_{wt}\ x\ y]]]\ {}^0couples$
${}^0[\lambda w \lambda t\, [[{}^0All^c\ couples]\ \lambda he\ it\ [{}^0Wash_{wt}\ he\ it]]]]_{wt}]$.

As is seen, (D') is fully compositional. Our constituents operate on constructions of sets of couples of individuals, as well as particular individuals, which is impossible within a first-order theory. In this respect Hintikka is right when claiming that the compositional treatment does not work;[13] it does not work within a first-order framework. But as soon as we have a powerful higher-order system like TIL at our disposal, there is no need to give up the desirable principle of compositionality.

One pressing question is whether the anaphoric pronouns should be, in general, bound, and if so, another pressing question is whether this is to be in a standard or non-standard way. The Dynamic Predicate Logic (DPL) applies a mechanism of passing on binding.[14] Note that (D') at the same time provides the semantics of this mechanism. Indeed, the variables *he* and *it* are bound in (D'), but the binding is of another kind. They are not directly bound by the existential quantifier. Technically, they are bound by Trivialization; semantically, they are bound by the condition that the pairs of individuals they *v*-construct have to belong to the set mentioned by the antecedent clause.

4 Outline of the Implementation Method

Now we outline the method of computing the complete meaning of anaphoric sentences, i.e., the method of substituting an appropriate antecedent for an anaphoric reference. The method is similar to the one applied by Hans Kamp's Discourse Representation Theory (DRT). 'DRT' is an umbrella term for a collection of logical and computational linguistic methods developed for dynamic interpretation of natural language, where each sentence is interpreted within a certain discourse, which is a sequence of sentences uttered by the same speaker. Interpretation conditions are given *via* instructions for updating the discourse representation. DPL is a logic belonging to this group of theories. Discourse representation theory as presented in [2] addresses in particular the problem of anaphoric links crossing the sentence boundary. It is a first-order theory, and it can be proved that the expressive power of the DRT language with negation is the same as that of first-order predicate logic. Thus actually only expressions denoting individuals (indefinite or definite noun phrases) introduce the so-called discourse referents, i.e., free variables that are updated when interpreting the

[13] See [9]
[14] See [8].

discourse. Anaphoric pronouns are represented by free variables linked to appropriate antecedent discourse variables.

As we have seen above, our semantics is hyperintensional, i.e., procedural, and higher order. Thus not only individuals, but entities of any type, like properties of individuals, propositions, relations-in-intension, and even constructions (i.e. meanings of the antecedent expressions), can be linked to anaphoric variables. Moreover, strong typing makes it possible to *determine the respective type-appropriate antecedent*.

The specification of the implementation algorithm proposed here is imperative;[15] similarly as in DRT, we update the list of potential antecedents, or rather constructions expressed by them, in order to substitute the type-appropriate entities for anaphoric variables, whenever needed. For each type (ι, $(o\iota)_{\tau\omega}$, $o_{\tau\omega}$, $(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$, $(o\iota\iota)_{\tau\omega}$, $*_n$, etc.) the list of discourse variables is created. The method substitutes the content of type-appropriate discourse variables for anaphoric variables to complete the meaning of anaphoric clauses. Each closed constituent of a resulting construction becomes an updated value of the respective (type-appropriate) free discourse-referent variable. In this way the discourse variables are gradually updated.

We now illustrate the method by an example of a simple dialog between three agents, *Adam*, *Berta* and *Cecil*. The list of discourse variables for the dialog together with the types of entities constructed by their respective content is: $ind{:=}\iota$, $loc{:=}\mu$, $pred{:=}(o\iota)_{\tau\omega}$, $prof{:=}(o\iota)_{\tau\omega}$, $rel_1{:=}(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$, $rel_2{:=}(o\iota\mu)_{\tau\omega}$, $rel_3{:=}(o\iota o_{\tau\omega})_{\tau\omega}$, $prop{:=}o_{\tau\omega}$, $constr{:=}*_n$.

Adam to Cecil: "Berta is coming. ***She*** is looking for a parking".
'Inform' message content:
$\lambda w\lambda t\ [[{}^0Coming_{wt}\ {}^0Berta]$;
(Relevant) discourse variables updates:
$ind{:=}{}^0Berta$; $pred{:=}{}^0Coming$;
$prop{:=}\ \lambda w\lambda t\ [[{}^0Coming_{wt}\ {}^0Berta]$;
$\lambda w\lambda t\ {}^2[{}^0Sub\ ind\ {}^0she\ {}^0[{}^0Looking\_for_{wt}\ she\ {}^0Parking]] \Rightarrow$ (is transformed into)
$\lambda w\lambda t\ [{}^0Looking\_for_{wt}\ {}^0Berta\ {}^0Parking]$.
(Relevant) discourse variables updates:
$rel_1{:=}\ {}^0Looking\_for$; $pred{:=}{}^0Parking$;
$prop{:=}\ \lambda w\lambda t\ [{}^0Looking\_for_{wt}\ {}^0Berta\ {}^0Parking]$;
$prof{:=}\ \lambda w\lambda t\ \lambda x\ [{}^0Looking\_for_{wt}\ x\ {}^0Parking]$; ('propositional function')

Cecil to Adam: "***So*** am I."
'Inform' message content:
$\lambda w\lambda t\ {}^2[{}^0Sub\ prof\ {}^0so\ {}^0[so_{wt}\ {}^0Cecil]] \Rightarrow \lambda w\lambda t\ [{}^0Looking\_for_{wt}\ {}^0Cecil\ {}^0Parking]$
(Relevant) discourse variables updates:
$ind{:=}{}^0Cecil$;

Adam to both: "There is a free parking at p_1".
'Inform' message content: $\lambda w\lambda t\ \exists x\ [[[{}^0Free\ {}^0Parking]_{wt}\ x] \wedge [{}^0At_{wt}\ x\ {}^0p_1]]$
(Relevant) discourse variables updates:
$loc{:=}{}^0p_1$; $pred{:=}[{}^0Free\ {}^0Parking]$;
$prop{:=}\ \lambda w\lambda t\ [\exists x\ [[{}^0Free\ {}^0Parking]_{wt}\ x] \wedge [{}^0At_{wt}\ x\ {}^0p_1]]$

[15] The algorithm was first proposed in [4].

Berta to Adam: "What do you mean by free parking?"
'Query' message content: $\lambda w \lambda t\ [{}^0Refine_{wt}\ {}^0[{}^0Free\ {}^0Parking]]$
(Relevant) discourse variables updates: $constr := {}^0[{}^0Free\ {}^0Parking]$

Adam to Berta: "Free parking is a parking and some parts of it are not occupied".
'Reply' message content: $[{}^0Free\ {}^0Parking] =$
$[\lambda w \lambda t\ \lambda x\ [[{}^0Parking_{wt}\ x] \wedge \exists y\ [[{}^0Part\_of_{wt}\ y\ x] \wedge \neg[{}^0Occupied_{wt}\ y]]]]$

Berta to Adam: "I don't believe ***it***. I have just been ***there***".
'Inform' message content (first sentence):
$\lambda w \lambda t\ [{}^2[{}^0Sub\ prop\ {}^0it\ {}^0[\neg[{}^0Believe_{wt}\ {}^0Berta\ it]]] \Rightarrow$
$\lambda w \lambda t\ \neg[{}^0Believe_{wt}\ {}^0Berta\ [\lambda w \lambda t\ [\exists x\ [[{}^0Free\ {}^0Parking]_{wt}\ x] \wedge [{}^0At_{wt}\ x\ {}^0p_1]]]]],$
'Inform' message content (second sentence):
$\lambda w \lambda t\ \exists t'[[t' \leq t] \wedge {}^2[{}^0Sub\ loc\ {}^0there\ {}^0[{}^0Been\_at_{wt'}\ {}^0Berta\ there]]] \Rightarrow$
$\lambda w \lambda t\ \exists t'[[t' \leq t] \wedge [{}^0Been\_at_{wt'}\ {}^0Berta\ {}^0p_1]].$
And so on.

Note that due to the procedural semantics, our agents can learn new concepts by asking the other agents. In our example, after receiving Adam's reply Berta learns the refined meaning of the 'free parking' predicate, i.e., she updates her knowledge base by the respective composed construction defining the property of being a parking with some free parts. Moreover, though our approach is as fine-grained as the syntactic approach of languages like KIF, the content of agent's knowledge is not a piece of syntax, but its meaning. And since the respective construction is what synonymous expressions (even of different languages) have in common, agents behave in the same way independently of the language in which their knowledge and ontology is encoded. For instance, if we switch to Czech, the underlying constructions are *identical*: ${}^0[{}^0Free\ {}^0Parking] = {}^0[{}^0Volné\ {}^0Parkoviště]$.

Of course, improvements of the above method are straightforward. For instance, in the example we were substituting the last type-appropriate entity that received mention; if we wanted to take into account ambiguities of anaphoric references, we might store into the discourse-representation file more than one variable for each type, together with the other characteristics or prerequisites of entities (e.g., gender, ISA hierarchies between properties), so as to be able to generate more meanings of an ambiguous sentence, and thus to contribute to their disambiguation.

5 Concluding Remarks

The above described method is currently being implemented in the TIL-Script programming language, the computational FIPA compliant variant of TIL. It is a declarative functional language. Its only imperative feature is the *Let* command for the dynamic assignment of a construction *C* to a discourse variable in order to update its content. The command is also used for recursive definitions of functions. TIL-Script comprises all the higher-order features of TIL, as the hyperintensional logic of partial functions with procedural semantics and explicit intensionalisation and temporalisation, making thus a communication of software-agents smooth and very natural. TIL constructions are encoded by natural-language expressions in a nearisomorphic manner and for the needs of real-world human agents TIL-Script messages are presented in a standardised natural language. *Vice versa*, humans can formulate their requests, queries, etc., in the standardised natural language that is transformed into TIL-Script messages. Thus the provision of services to humans can be realised in a form close to human understanding.

Acknowledgements. This research has been supported by the program ‘Information Society’ of the Czech Academy of Sciences within the project “Logic and Artificial Intelligence for multi-agent systems”, No. 1ET101940420, and by the project “Semantization of pragmatics”, No. GACR 401/07/0451.

References

1. Geach, P.: *Reference and Generality***.** Ithaca, NY: Cornell University Press (1962)
2. Kamp, H. and Reyle, U.: *From Discourse to Logic. Introduction to Model-Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Springer (1993)
3. Jespersen, B.: ‘Predication and extensionalization’. *Journal of Philosophical Logic*, January 2008; retrievable at: http://www.springerlink.com/content/q31731r311428343/?p=60281b936ca249278691b0f25135cc0a&pi=2
4. Křetínský, J.: *Use–mention Distinction in Transparent Intensional Logic*. Bachelor thesis, Masaryk University Brno (2007); retrievable at: http://is.muni.cz/th/139914/fi_b/bachelor.pdf
5. Materna, P.: *Conceptual Systems.* Logos Verlag, Berlin (2004)
6. Neale, S.: *Descriptions*. The MIT Press, Cambridge (1990)
7. Rescher, N.: *Epistemic Logic.* Pittsburgh: University of Pittsburgh Press (2005)
8. Sandu, G.: ‘On the theory of anaphora: dynamic predicate logic vs. game-theoretical semantics’. *Linguistic and Philosophy* 20 (1997), 147-174
9. Sandu, G., Hintikka, J.: ‘Aspects of compositionality’. *Journal of Logic, Language, Information 10* (2001) 49-61
10. Tichý, P.: *The Foundations of Frege’s Logic*, Berlin, New York: De Gruyter (1988)
11. Tichý, P.: *Collected Papers in Logic and Philosophy*, V. Svoboda, B. Jespersen, C. Cheyne (eds.), Prague: Filosofia, Czech Academy of Sciences, and Dunedin: University of Otago Press (2004)