Tenses and truth-conditions: a plea for if-then-else

Marie Duží*

December 6, 2009

1 Introduction

Sentences in the present, past and future tenses obviously have different truth-conditions. This fact has been observed by numerous logicians, and many variants of so-called temporal logic have been developed. These formal systems are mostly viewed as a special case of modal logic interpreted by means of Kripkean possible-world semantics. The term temporal logic is broadly used to cover all approaches to the representation of the temporal dimension within a logical framework. More narrowly, it is also used to refer to a particular modal system of temporal propositional logic that Arthur Prior introduced in (1957, 1962 and 1967) under the name '*tense logic*'.

The logical language of Prior's tense logic contains, in addition to the usual truth-functional operators, four modal operators whose intended meanings are:

P "It has at some time been the case that ..."
F "It will at some time be the case that ..."
H "It has always been the case that ..."
G "It will always be the case that ..."

P and F are known as the *weak tense operators*, while H and G are known as the *strong tense operators*. Prior developed a formal system of tense logic with axioms like

$Gp \rightarrow Fp$ "What will always be will be";
$G(p \rightarrow q) \rightarrow (Gp \rightarrow Gq)$ "If p will always imply q, then if p will always be the case, so will q";
$Fp \rightarrow FFp$ "If it will be the case that p, it will be the case that it will be that p";
$\neg Fp \rightarrow F\neg Fp$ "If it will never be that p then it will be that it will never be that p".

*This work has been supported by the Grant Agency of the Czech Republic, project No. GACR 401/07/0451, 'Semantization of Pragmatics', and by the internal grant agency of FEECS VSB-TU Ostrava - IGA 22/2009, 'Modelling, simulation and verification of software processes'.

Similarly for the past operators P, H; e.g., $Hp \rightarrow Pp$, "What has always been has been". Subsequently, systems of temporal logic have been further developed by computer scientists, notably Zohar Manna and Amir Pnueli,[1] and widely used for formal verification of programs and for encoding temporal knowledge within artificial intelligence. In addition to Prior's future and past operators Manna and Pnueli have introduced modal operators like *Since* and *Until* that are provably more expressive than ordinary modal operators and are usually interpreted by (labelled) transition systems of program states that are pivotal to the operational semantics of programs.

These logics are undeniably simple, elegant and logically convenient. However, simplicity and convenience do not always go hand in hand with *logical adequacy.* Despite the great applicability of particular variants of tense logic in the semantics of programming languages, the systems just mentioned suffer a drawback when applied to the semantics of natural language. The drawback is their inability to adequately analyse sentences indicating a point of reference referring to the interval when the sentence was or will be true. Such sentences come attached with a *presupposition* under which a sentence is true or false. To illustrate the problem, consider the sentences

"Tom is sick".
"Tom has been sick".
"Tom was sick throughout October 2009".
"Tom will be sick the whole day on April 1st, 2010".

The first two sentences do not come with a presupposition. They ascribe to Tom the property of being sick and of having been sick, respectively. They are true or false according as Tom has the relevant property. However, the truth-condition of the third sentence depends not only on whether Tom has the property of being sick throughout October 2009, but also on the time at which the sentence is evaluated. If T is the time of evaluation, then the truth-conditions are specified as follows:

If $T \leq$October 31, 2009, 24:00, then **no truth-value**, else **True** or **False** according as Tom was sick at all times during October 2009.

Similarly, the fourth sentence comes attached with a presupposition that the time T of evaluation comes before April 1st, 2010.

Our analysis must respect these truth-conditions. To this end we apply the rich system of Tichý's Transparent Intensional Logic (TIL).[2] Tichý put forward his solution in (1980). However, this solution is difficult to understand, because Tichý applies the *singulariser* function to a singleton typed as containing a truth-value in order to make the set fail to deliver a truth-value in case the associated presupposition is not satisfied. Tichý's analysis is analogous to what the computer scientist would call an *imperative* rather than *declarative* analysis. The downside

[1] See (Manna & Pnueli, 1992).
[2] See (Tichý, 1988)

to an imperative analysis is that it may conceal flaws that rear their head only when the analysis is applied to extreme situations. Yet there is an elegant alternative that makes use of the 'if-then-else' connective, which I am going to introduce in Section 3 of this paper.[3]

There has been much dispute over the semantics of 'if-then-else' in the logic of computer science. It cannot be adequately analyzed by means of material implication. The reason is this. The application of *if-then-else* to a condition P and two formulae F_1 and F_2 is not improper (by failing to provide a truth-value) even when F_2 is improper whenever P is true or when F_1 is improper whenever P is false. However, the regimentation of "If P then F_1, else F_2" in propositional logic that takes the form $[(P \supset F_1) \wedge (\neg P \supset F_2)]$ is improper by failing to produce a truth-value whenever F_1 or F_2 is improper regardless the condition P. Thus it is often said that *if-then-else* is a non-strict function that does not behave in compliance with the compositionality principle. Yet there is no cogent reason to settle for non-strictness.

In what follows I am going to show that the *procedural* semantics of TIL enables us to specify a strict definition of *if-then-else* that meets the compositionality constraint. The definition of "If P then F_1, else F_2" is a procedure that decomposes into two phases. First, on the basis of the condition P, select one of F_1, F_2 as the procedure to be executed. Second, execute the selected procedure. Thus, for instance, if P is true then F_1 is executed rather than F_2, and the possible improperness of F_2 does not matter. After setting out the definition, I specify a general schema in which to couch the analysis of sentences that come attached with a presupposition, in particular sentences in the past and future tenses.

2 Method of analysis

TIL operates with a single *procedural semantics* for all kinds of logical-semantic context, whether extensional, intensional or hyper-intensional.[4] It means that it explicates the meaning of an expression as an abstract *procedure* encoded by the expression. Such procedures are rigorously defined as TIL *constructions* and we assign them to expressions as their *context-invariant* meanings. From the formal point of view, TIL is a hyper-intensional, partial, typed λ-calculus. Hyper-intensional, because the terms of the TIL formal language in which constructions are encoded are interpreted as *procedures* (generalized algorithms) rather than their *products*; partial, because the primitive notion of TIL is function understood as a partial mapping that assigns to each element of its domain *at most one* element of its range; and typed, because all the entities of TIL ontology, including constructions, receive a type.

Intuitively, construction C is a *procedure* (a generalised algorithm). Constructions are *structured* in the following way. Each construction C consists of sub-

[3] I am grateful to Nikola Ciprich for drawing my attention to this option.

[4] In this section, the philosophy and basic notions of TIL are only briefly summarized. For details, see (Duží, Jespersen, & Materna, forthcomming).

constructions (*constituents*), each of which needs to be executed when executing C. Specification of a construction can be viewed as an instruction on how to proceed in order to obtain the output entity given some input entities. In this way a construction constructs a function understood as a mapping from the set of input entities into a set of output entities.

There are two kinds of constructions, atomic and compound (molecular). Atomic constructions (*Variables* and *Trivializations*) do not contain any other constituent but themselves; they specify objects (of any type) on which compound constructions operate. The *variables* $x, y, p, q, \ldots,$ construct objects dependently on a valuation; they v-construct. The *Trivialisation* of an object X (of any type, even a construction), in symbols ${}^0X$, constructs simply X without the mediation of any other construction. *Compound* constructions, which consist of other constituents as well, are *Composition* and *Closure*. The *Composition* $[F\, A_1 \ldots A_n]$ is the operation of functional application. It v-constructs the value of the function f (*valuation-*, or v-, -constructed by F) at a tuple argument A (v-constructed by $A_1, \ldots, A_n$), if the function f is defined at A, otherwise the Composition is v-*improper*, i.e., it *fails* to v-construct anything.[5] The *Closure* $[\lambda x_1 \ldots x_n\, F]$ spells out the instruction to v-construct a function by abstracting over the values of the variables $x_1, \ldots, x_n$ in the ordinary manner of the λ-calculi.[6] Finally, higher-order constructions can be used twice over as constituents of composite constructions. This is achieved by a fifth construction called *Double Execution*, ${}^2X$, that behaves as follows: If X v-constructs a construction Y, and Y v-constructs an entity Z, then ${}^2X$ v-constructs Z; otherwise ${}^2X$ is v-*improper*, failing as it does to v-construct anything.

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* (o-truth values, ι-individuals, τ-time moments/real numbers, ω-possible worlds), the induction rule for forming functional types is applied: where $\alpha, \beta_1, \ldots, \beta_n$ are types of order 1, the set of partial mappings from $\beta_1 \times \cdots \times \beta_n$ to α, denoted '$\alpha\, \beta_1 \ldots \beta_n$', is a type of order 1 as well. Constructions that construct entities of order 1 are *constructions of order 1.* They belong to a *type of order 2*, denoted '$\star_1$'. The type $\star_1$ together with atomic types of order 1 serve as a base for the induction rule: any collection of partial mappings, type $(\alpha\, \beta_1 \ldots \beta_n)$, involving $\star_1$ in their domain or range is a *type of order 2.* Constructions belonging to a type $\star_2$ that v-construct entities of order 1 or 2, and partial mappings involving such constructions, belong to a *type of order 3.* And so on *ad infinitum.*

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

[6] Comparison with programming languages might be helpful: λ-Closure corresponds to a declaration of a procedure F with formal parameters $x_1, \ldots, x_n$; Composition corresponds to calling the procedure F with actual values $A_1, \ldots, A_n$ of parameters.

The sense of an empirical expression is a *hyperintension*, i.e., a construction that produces a (possible world) *(α-)intension*; α-intensions are members of type $(\alpha\omega)$, i.e., functions from possible worlds to an arbitrary type α. On the other hand, *(α-)extensions* are members of a type α, where α is not equal to $(\beta\omega)$ for any β, i.e., extensions are not functions whose domain are possible worlds. Intensions 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 $(\alpha\tau)$.

Some important kinds of intensions are:

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

Properties of members of a type α, or simply *α-properties*, type $(o\alpha)_{\tau\omega}$.[7] General terms, some substantives, intransitive verbs ('student', 'walks') denote properties, mostly of individuals.

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

***α-roles*,** also ***α-offices*,** type $\alpha_{\tau\omega}$, where $\alpha \neq (o\beta)$; frequently $\iota_{\tau\omega}$. They are often denoted by concatenation of a superlative and a noun ('the highest mountain').

Notational conventions. An object A of a type α is denoted 'A/α'. That a construction $C/\star_n$ v-constructs an object of type α is denoted '$C \rightarrow_v \alpha$'. We use variables $w, w_1, \dots$ as v-constructing elements of type ω (possible worlds), and $t, t_1, \dots$ as v-constructing elements of type τ (times). If $C \rightarrow_v \alpha_{\tau\omega}$ v-constructs an α-intension, the frequently used Composition of the form $[[Cw]t]$, the intensional descent of the α-intension, is abbreviated 'C_{wt}'.

Quantifiers, $\forall^\alpha$ (the general one) and $\exists^\alpha$ (the existential one), are of types $(o(o\alpha))$, i.e., sets of sets of α-objects. $[^0\forall^\alpha \lambda x\, A]$ v-constructs the truth-value $\mathbf{T}$ iff $[\lambda x A]$ v-constructs the whole type α, otherwise $\mathbf{F}$; $[^0\exists^\alpha \lambda x A]$ v-constructs $\mathbf{T}$ iff $[\lambda x A]$ v-constructs a non-empty subset of the type α, otherwise $\mathbf{F}$. We write '$\forall x\, A$', '$\exists x\, A$' instead of $[^0\forall^\alpha \lambda x\, A]$, $[^0\exists^\alpha \lambda x A]$, respectively, when no confusion can arise. In the effort of easier reading we will also use an infix notation without trivialisation when using constructions of truth-value functions of type (ooo), i.e., $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication), $\equiv$ (equivalence), and negation $(\neg)$ of type (oo), and when using constructions of common relations like identities, less than $(<)$, greater than $(>)$, etc.

We invariably furnish expressions with their procedural structured meanings, which are explicated as TIL constructions. The analysis of a sentence thus consists in discovering the logical construction encoded by a given sentence. The TIL compositional *method of analysis* is driven by Carnap's *principle of subject matter*, which says, roughly, that only those entities that receive mention in a sentence

[7]We model α-sets and $(\alpha_1 \dots \alpha_n)$-relations by their characteristic functions of type $(o\alpha)$, $(o\alpha_1 \dots \alpha_n)$, respectively. Thus an α-property is an empirical function that dependently on states-of-affairs $(\tau\omega)$ picks-up a set of α-individuals, the population of the property.

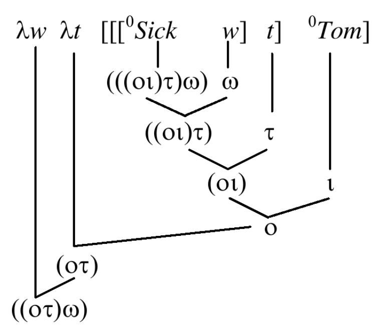
can serve as constituents of the sentence meaning.[8] The method decomposes into three phases:

1. *Type-theoretical analysis*, i.e., assigning types to the objects that receive mention in the analysed sentence.
2. *Synthesis*, i.e., combining the constructions of the objects *ad* (1) in order to construct the proposition of type $o_{\tau\omega}$ denoted by the whole sentence.
3. *Type-theoretical checking*

To illustrate the method let us analyse the sentence "Tom is sick".
Ad (1) Tom/ι; $Sick/(o\iota)_{\tau\omega}$: the property of individuals (of being sick).
Ad (2) Since we aim at discovering the *literal* analysis of the sentence, objects denoted by semantically simple expressions 'Tom', 'is sick' are constructed by their Trivialisations: ${}^0Tom$, ${}^0Sick$. Now we need to predicate the property *Sick* of the individual *Tom*. But we cannot apply the property directly to *Tom*. The individual *Tom* is not a type-theoretically proper object to serve as an argument of a property, because property is an *intension*. It must be extensionalized first, i.e., applied to a given world w and time t of evaluation, and only then applied to Tom. This is achieved by three Compositions $[{}^0Sick\, w] \rightarrow_v ((o\iota)\tau)$, $[[{}^0Sick\, w]t] \rightarrow_v (o\iota)$, abbreviated as ${}^0Sick_{wt}$, and finally $[[[{}^0Sick\, w]t]{}^0Tom] \rightarrow_v o$, or $[{}^0Sick_{wt}\, {}^0Tom]$ for short.[9] Since we are going to construct a proposition, i.e., an intension, we must abstract from values of w and t:

$$\lambda w \lambda t[{}^0Sick_{wt}\, {}^0Tom].$$

This construction is the literal analysis of the sentence "Tom is sick".
Ad (3) *Type-theoretical checking:*



The role of Trivialisation and empirical parameters, that is modal $w \rightarrow \omega$ and temporal $t \rightarrow \tau$ ones, can be elucidated as follows. When evaluating the truth-condition of the sentence, then the upper index '^{0}' serves as a marker of the primitive concept that supplies an object to be operated on. The lower index '${}_{wt}$' can be understood as an instruction to execute an *empirical inquiry (search)* in

[8] See (Carnap, 1947, §24.2, §26)
[9] For details on predicating properties of individuals, see (Jespersen, 2008)

order to check whether the object in question (here Tom) satisfies the condition specified by the intension (here sickness).

So much for the TIL method of analysis. Now I am going to apply this method to analyze sentences in past and future.

3 Sentences in past and future tenses

Consider the sentence in present perfect

(1) "Tom has been sick".

It informs us that Tom has the property of being sick not only in the present time t of evaluation but also in an interval that runs from the past up to t (and possibly beyond). Thus the analysis of the sentence comes down to the Closure

(1') $\lambda w \lambda t \, \exists t_1 \, [\forall t_2 \, [t_1 < t_2 \leq t] \supset [{}^0Sick_{wt_2} \, {}^0Tom]]$.

A similar sentence in simple past, which is "Tom was sick", seems to be incomplete, because one is tempted to ask "When was Tom sick?" This is because sentences in past should contain an indication of *when* something happened, for instance

(2) "Tom was sick throughout October 2009".

As mentioned above, such a sentence not only entails but also *presupposes* that the time t of evaluation comes after October 2009. The difference between presupposition and mere entailment can be schematically demonstrated as follows:

(i) P is a *presupposition* of S, iff: $(S \models P)$ and $(non\text{-}S \models P)$

Corollary: If *non-P* then neither S nor *non-S* is true, i.e., S does not have a truth-value.

(ii) Mere entailment: $(S \models P)$ and neither $(non\text{-}S \models P)$ nor $(non\text{-}S \models non\text{-}P)$

More precisely, the entailment relation $\models$ obtains between hyperpropositions P, S, i.e., the meaning of P is entailed or presupposed by the meaning of S. And since we work with *partial* functions, we can smoothly analyse sentences associated with a presupposition. If the proposition constructed by P does not take the truth-value **T** at a given $\langle w, t \rangle$-pair, then the proposition constructed by S has a truth-value gap at this $\langle w, t \rangle$.

Denoting the interval October 2009 by *'Oct09'*, the presupposition of (2) is

$$\lambda w \lambda t \forall t_1 \, [[{}^0Oct09 \; t_1] \supset [t_1 < t]]$$

Gloss: In any world w at any time t it holds for all times $t_1 \rightarrow \tau$ belonging to the interval $Oct09/(o\tau)$ that $t_1 < t$. In other words, the entire month October 2009 precedes the time t.

Now the schematic analysis of (2) is this:

(2^s) $\lambda w \lambda t$ If $\forall t_1\, [[^0 Oct09\, t_1] \supset [t_1 < t]]$ then $\forall t'\, [[^0 Oct09\, t'] \supset [^0 Sick_{wt'}\, ^0 Tom]]$ else *Fail*.

To complete the analysis, we must define the *If-then-else* function. Here is how. The instruction encoded by "If $P(\rightarrow o)$ then $C(\rightarrow \alpha)$, else $D(\rightarrow \alpha)$" behaves as follows:

a. If P v-constructs **T** then execute C (and return the result of type α, provided C is not v-improper).
b. If P v-constructs **F** then execute D (and return the result of type α, provided D is not v-improper).
c. If P is v-improper then fail to produce the result.

Hence, *if-then-else* is seen to be a function of type $(\alpha o \star_n \star_n)$, and its definition decomposes into two phases.

First, select a construction to be executed on the basis of a specific condition P. The choice between C and D comes down to this Composition:

$$[^0 The\_only\, \lambda c\, [[P \supset [c = {}^0 C]] \wedge [\neg P \supset [c = {}^0 D]]]]$$

Types: $P \rightarrow_v o$ v-constructs the condition of the choice between the execution of C or D, $C/\star_n$, $D/\star_n$; $c \rightarrow_v \star_n$; $The\_only/(\star_n(\star_n))$: the singularizer function that associates a singleton set of constructions with the only construction that is an element of this singleton, and is otherwise (i.e., if the set is empty or many-valued) undefined. If P v-constructs **T** then the variable c v-constructs the construction C, and if P v-constructs **F** then the variable c v-constructs the construction D. In either case, the set constructed by

$$\lambda c\, [[P \supset [c = {}^0 C]] \wedge [\neg P \supset [c = {}^0 D]]]$$

is a singleton and the singularizer *The_only* returns as its value either the construction C or the construction D. Note that in this phase constructions C and D are not constituents to be executed; rather they are mere objects to be supplied by the variable c. This is to say that without *hyperintensional* approach we would not be able to define the function *If-then-else*.

Second, the selected construction is executed; therefore, Double Execution must be applied:

$$^2[^0 The\_only\, \lambda c\, [[P \supset [c = {}^0 C]] \wedge [\neg P \supset [c = {}^0 D]]]]$$

As a special case of P being a presupposition, *no* construction D is to be selected whenever P is not satisfied. Thus the analysis of

"If (presupposition) P then $C \rightarrow o$ else *Fail* (to produce a truth-value)" comes down to the Double Execution

$$^{2}[^{0}The\_only\,\lambda c\,[[P \supset [c = {}^{0}C]] \wedge [\neg P \supset^{0} \mathbf{F}]]]$$

Gloss: If $\neg P$ v-constructs $\mathbf{T}$ then $[\neg P \supset {}^{0}\mathbf{F}]$ v-constructs $\mathbf{F}$ and the set v-constructed by the Closure $\lambda c\,[[P \supset [c = {}^{0}C]] \wedge [\neg P \supset {}^{0}\mathbf{F}]]$ is empty. Thus the singulariser *The_only* does not return any construction and the Double Execution does not obtain an argument to execute; hence it is v-improper, that is fails to produce a truth-value.

Back to the analysis of (2). Applying this schematic definition to the construction (2^s), that is, substituting $\forall t_1\,[[^{0}Oct09\,t_1] \supset [t_1 < t]]$ for P and $[\forall t'\,[[^{0}Oct09\,t'] \supset [^{0}Sick_{wt'}\,{}^{0}Tom]]]$ for C, we obtain the final analysis of the sentence (2):

(2∗) $\lambda w \lambda t\, ^{2}[^{0}The\_only\,\lambda c\,[[\forall t_1 [[^{0}Oct09\,t_1] \supset [t_1 < t]] \supset$
$[c = {}^{0}[\forall t'\,[[^{0}Oct09\,t'] \supset [^{0}Sick_{wt'}\,{}^{0}Tom]]]]] \wedge [\exists t_1\,[[^{0}Oct09\,t_1] \wedge [t_1 \geq t]] \supset {}^{0}\mathbf{F}]]]$

Since such an analysis is not easy to read and the *If-then-else* function has been defined, in what follows I will use the schematic analysis of the form

$$\text{``}\lambda w \lambda t\, \text{If}\, P_{wt}\, \text{then}\, S_{wt}\, \text{else}\, Fail\text{''}$$

rather than the full-fledged

$$\text{``}\lambda w \lambda t\, ^{2}[^{0}The\_only \lambda c[[P_{wt} \supset [c = {}^{0}[S_{wt}]]] \wedge [\neg P_{wt} \supset {}^{0}\mathbf{F}]]]\text{''}.$$

The sentences in past often indicate as a reference point not only an interval when something happened but also a frequency of it, like once, twice, often, or throughout (as is the case of (2)). To adduce another example, consider the sentence

(3) "Tom was sick (just) twice in October 2009".

The presupposition of (3) is again the proposition that the entire month October 2009 precedes time t of evaluation: $\lambda w \lambda t\, \forall t_1\,[[^{0}Oct09\,t_1] \supset [t_1 < t]]$. The schematic analysis of (3) comes down to:

(3^s) $\lambda w \lambda t\,\text{If}\,\forall t_1\,[[^{0}Oct09\,t_1] \supset [t_1 < t]]\,\text{then}\,[[^{0}Twice_w \lambda w \lambda t[^{0}Sick_{wt}\,{}^{0}Tom]]\,{}^{0}Oct09]$ else $Fail$.

The frequency modifier *Twice* denotes a world-dependent function that takes a proposition $p \to o_{\tau\omega}$ to the class of those intervals $d \to (o\tau)$ which are contained in the chronology of p (i.e. $p_w \to (o\tau)$). This class of intervals d that have a non-empty intersection with a reference interval c is of cardinality two. Thus the application of *Twice* of type $((o(o\tau))o_{\tau\omega})_\omega$ to a proposition p and reference interval c comes down to this Composition:

$$[[^{0}Twice_w\,p]c] = [^{0}Card \lambda d\,[\forall t\,[[d\,t] \supset p_{wt}] \wedge \exists t\,[[d\,t] \wedge [c\,t]]] = {}^{0}2]$$

In our case the interval c is *Oct09* and the proposition p is $\lambda w \lambda t\,[^{0}Sick_{wt}\,{}^{0}Tom]$, and we have

$$[[^{0}Twice_w\,\lambda w \lambda t\,[^{0}Sick_{wt}\,{}^{0}Tom]]\,{}^{0}Oct09] =$$

$$[{}^0\!Card\,\lambda d\,[\forall t\,[[d\,t] \supset [{}^0\!Sick_{wt}\,{}^0\!Tom]] \wedge \exists t\,[[d\,t] \wedge [{}^0\!Oct09\,t]]] = {}^0\!2]$$

Thus we can refine the analysis (3^s) like this:

$$(3^*)\quad \lambda w\lambda t\,\text{If}\,\forall t_1\,[[{}^0\!Oct09\,t_1] \supset [t_1 < t]]$$
$$\text{then}\,[{}^0\!Card\,\lambda d\,[\forall t\,[[d\,t] \supset [{}^0\!Sick_{wt}\,{}^0\!Tom]] \wedge \exists t\,[[d\,t] \wedge [{}^0\!Oct09\,t]]] = {}^0\!2]$$
$$\text{else}\,Fail.$$

Our resources up to now make it possible to define a general schema of the analysis of a sentence S in past tense with a reference interval $In\_Time/(o\tau)$ and a modifier $Frequency/((o(o\tau))o_{\tau\omega})_\omega$. Let *Past* be a time-dependent function that takes a class of o-chronologies (the intervals in which a given proposition is true) together with an (implicit or explicit) reference interval and returns **T**, **F** or no value, according as the interval serving as point of reference belongs to the respective class of o-chronologies *and* precedes the time T at which the proposition denoted by the sentence is being evaluated. Thus, *Past* is typed as $((o(o(o\tau))(o\tau))\tau)$. Let $\leq_\tau /(o(o\tau)\tau)$ mean that the reference interval In_Time is prior to time t. Then the **general schema of sentences in past** is:

$$\lambda w\lambda t\,[{}^0\!Past_\tau\,[{}^0\!Frequency_w\,S]\,{}^0\!In\_Time] =$$

$$\lambda w\lambda t\,\text{If}\,[{}^0\!In\_Time \leq_\tau t]\,\text{then}\,[[{}^0\!Frequency_w\,S]{}^0\!In\_Time]\,\text{else}\,Fail.$$

For instance, our sentence (2) receives the literal analysis

$$\lambda w\lambda t[{}^0\!Past_t\,[{}^0\!Throughout_w\,\lambda w\lambda t\,[{}^0\!Sick_{wt}\,{}^0\!Tom]]\,{}^0\!Oct09]$$

and the sentence

(4) "Tom was sick at least once before October 2009"

is analysed by the Closure

$$\lambda w\lambda t\;\; [{}^0\!Past_t\,[{}^0\!At\_least\_once_w\lambda w\lambda t\,[{}^0\!Sick_{wt}\,{}^0\!Tom]]\lambda t'[{}^0\!Before\,t'\,{}^0\!Oct09]] =$$
$$\lambda w\lambda t\;\; \text{If}\,[\lambda t'\,[{}^0\!Before\,t'\,{}^0\!Oct09] \leq_\tau t]$$
$$\text{then}[[{}^0\!At\_least\_once_w\lambda w\lambda t\,[{}^0\!Sick_{wt}\,{}^0\!Tom]]\lambda t'[{}^0\!Before\,t'\,{}^0\!Oct09]]$$
$$\text{else}\,Fail.$$

Here the point of reference is specified as *any time before Oct09.* To analyse 'before October 2009', we have to define the type of the object denoted by 'before'. Given a time t' and a τ-class c, the time t' is prior to c if t' is prior to every element of c. Thus $Before/(o\tau(o\tau))$ receives the definition ${}^0\!Before = \lambda t'c[\forall t\,[c\,t] \supset [t' < t]]$, and 'before October 2009' expresses the Closure $\lambda t'[{}^0\!Before\,t'\,{}^0\!Oct09]$ which is equivalent to $\lambda t'[\forall t\,[{}^0\!Oct09\,t] \supset [t' < t]]$.

The definition of $At\_least\_once/((o(o\tau))o_{\tau\omega})_\omega$ is easy:

$${}^0\!At\_least\_once = \lambda w\lambda p\lambda c\exists t\,[[c\,t] \wedge p_{wt}].$$

The truth-condition is that a proposition p be true at least once in a world w in an interval c if there is at least one time t in c at which p is true in w. Thus the Composition $[{}^0At\_least\_once_w\, \lambda w \lambda t\, [{}^0Sick_{wt}\, {}^0Tom]]$ v-constructs the class $S/(o(o\tau))$ of intervals in which Tom is sick at least once.

The **general schema of sentences in future** is similar to the analytic schema of sentences in past:

$$\lambda w \lambda t\, [{}^0Future_t\, [{}^0Frequency_w\, S]{}^0In\_Time] =$$

$$\lambda w \lambda t\, \text{If}\, [{}^0In\_Time \geq_\tau t]\, \text{then}\, [[{}^0Frequency_w\, S]\, {}^0In\_Time]\, \text{else}\, Fail.$$

Here $\geq_\tau$ means that the reference interval In_Time comes after time t, $Future$ receives the same type as $Past$, that is $((o(o(o\tau))(o\tau))\tau)$. For instance, the sentence

(5) "Tom will be sick the whole day on April 1st 2010"

expresses as its sense ($April1/(o\tau)$: the day April 1st, 2010)

(5^s) $\lambda w \lambda t\, [{}^0Future_t\, [{}^0The\_whole_w\, \lambda w \lambda t [{}^0Sick_{wt}\, {}^0Tom]]\, {}^0April1]$

which is equivalent to

$$\begin{gathered} \lambda w \lambda t\, \text{If}\, [{}^0April1 \geq_\tau t] \\ \text{then}\, [[{}^0The\_whole_w \lambda w \lambda t\, [{}^0Sick_{wt}\, {}^0Tom]]\, {}^0April1] \\ \text{else}\, Fail. \end{gathered}$$

This analysis can be refined to this Closure:

$$\begin{gathered} \lambda w \lambda t\, \text{If}\, \forall t_1 [[{}^0April1\, t_1] \supset [t_1 > t]] \\ \text{then}\, \forall t' [[{}^0April1\, t'] \supset [{}^0Sick_{wt'}\, {}^0Tom]] \\ \text{else}\, Fail. \end{gathered}$$

4 Topic-focus ambiguities

Now I am going to heed the ambiguities pivoted on the difference between topic and focus articulation.[10] As an example, consider the sentence

(6) "All Toms children were sick last week".

There are two non-equivalent readings of this sentence. To illustrate, imagine two scenarios.
(i) The sentence is an answer to the question "What about Tom's children"? Then 'Tom's children' is the topic of the sentence and to each of these children the property of being sick last week (the focus) is ascribed. In such a situation the sentence not only entails but also *presupposes* that Tom has children *in the*

[10] For a linguistic analysis of this difference see (Hajičová, 2008)

present time.

(ii) Another possible scenario is this. The question is "What was going on *last week*"? And the answer, "Oh, all Tom's children were sick *last week*". In this situation 'last week' is the topic and the sentence *only entails* but does not presuppose that Tom had children *last week*.

Since the sentence is ambiguous, we are actually going to analyse *two* non-equivalent sentences, which might be paraphrased as follows:

(6i) "Each of present Tom's children was sick (throughout) last week".
(6ii) "(Throughout) last week each of the children Tom had was sick".

Let $Last\_week/((o\tau)\tau)$ be the function that associates a given time t with an interval that is last week with respect to t. The analyses come down to these Closures:

(6$i*$) $\lambda w \lambda t \,\text{If}\, [^0Has_{wt}\, {}^0Tom\, {}^0Children]$
$\text{then} \forall t^* \,[[[^0Last\_weekt]\, t^*] \supset [[^0All\, [^0Children\_of_{wt}\, {}^0Tom]]\, {}^0Sick_{wt^*}]]$
$\text{else}\, Fail.$

(6$ii*$) $\lambda w \lambda t \forall t^* \,[[[[^0Last\_week\, t]t*] \supset [^0Has_{wt^*}\, {}^0Tom\, {}^0Children]] \wedge$
$[[^0Has_{wt^*}\, {}^0Tom\, {}^0Children] \supset [[^0All\, [^0Children\_of_{wt^*}\, {}^0Tom]]\, {}^0Sick_{wt^*}]]]$

Types: $All/((o(o\iota))(o\iota))$: the restricted quantifier that associates a set M of individuals with the set of supersets of M. $Has/(o\iota(o\iota)_{\tau\omega})_{\tau\omega}$: the relation-in-intension between an individual and a property (of having instances of the property); $Children/(o\iota)_{\tau\omega}$; $Children\_of/((o\iota)\iota)_{\tau\omega}$;

Note that indeed (6ii^*) only entails that Tom had children in all times t^* belonging to the last week, but does not presuppose it. This is because (6ii^*) constructs a proposition that takes value **F** in those $\langle w, t\rangle$-pairs where either Tom did not have children in times t^* or Tom had children at that time but some of them were not sick.

5 Concluding remarks

In this paper I demonstrated the method of analysis of sentences in past and future. Moreover, I also presented the general analytic schema for sentences that come associated with a presupposition. To this end I utilized a strict definition of the *If-then-else* function that complies with the compositionality constraint. Last but not least, the semantic character of the ambivalence concerning the topic-focus articulation of sentences was analysed.

Logical analysis cannot disambiguate any sentence, because it presupposes full linguistic competence. Yet, our fine-grained method can contribute to a language disambiguation by making these hidden features *explicit* and *logically tractable*. In case there are more non-equivalent senses of a sentence we furnish the sentence with different TIL constructions. Having a formal fine-grained encoding of a sense, we can then *infer the relevant consequences*.

To sum up, I am convinced that if any logic can serve to solve such hard problems like fine-grained analysis of tenses, topic-focus ambiguities, and many others that natural language can produce, then it must be a logic with *hyper-intensional (most probably procedural) semantics*, such as TIL.

Marie Duží
Department of Computer Science
VSB-Technical University Ostrava
17. listopadu 15
708 33 Ostrava, Czech Republic
marie.duzi@vsb.cz
http://www.cs.vsb.cz/duzi

References

Carnap, R. (1947). *Meaning and Necessity.* Chicago: Chicago University Press.

Duží, M., Jespersen, B., & Materna, P. (forthcomming). *Procedural Semantics for Hyperintensional Logic; Foundations and Applications of Transparent Intensional Logic*. Berlin: Springer.

Hajičová, E. (2008). What we are talking about and what we are saying about it. *Computational Linguistics and Intelligent Text Processing*, *4919*, 241-262.

Jespersen, B. (2008). Predication and extensionalization. *Journal of Philosophical Logic*, *37*, 479–499.

Manna, Z., & Pnueli, A. (1992). *The Temporal Logic of Reactive and Concurrent Systems: Specification.* New York: Springer-Verlag.

Manna, Z., & Pnueli, A. (1995). *Temporal Verification of Reactive Systems: Safety.* New York: Springer-Verlag.

Prior, A. N. (1957). *Time and Modality.* Oxford: Oxford University Press.

Prior, A. N. (1962). Tense logic and the continuity of time. *Studia Logica*, *13*, 133–148.

Prior, A. N. (1967). *Past, Present and Future.* Oxford: Clarendon Press.

Tichý, P. (1980). The logic of temporal discourse. *Linguistics and Philosophy*, *3*, 343–369. (reprinted in (Tichý, 2004))

Tichý, P. (1988). *The Foundations of Frege's Logic.* Berlin: New York: De Gruyter.

Tichý, P. (2004). *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.